Name, Vorname der Studierenden: **Bruno Cajacob, Richard Lanicca**

Studierende der Abteilung: Telecom

Thema: **Architektur eines Honeypots**

Partnerfirma: KPMG Information Risk Management
Badenerstr. 172, 8026 Zürich 4, Switzerland

Dozent: Dr. Dipl. El. Ing. ETH Bruno Studer

Ausgabe der Semesterarbeit: Dienstag, 17. April 2001

Abgabe des Berichtes: Donnerstag, 6. September 2001

Thema der Semesterarbeit	**Architektur eines Honeypots**
Abteilung	Telecom

Personalienblatt

Name, Vorname	**Cajacob Bruno**
Adresse	Scesaplanastr. 9
PLZ, Wohnort	7000 Chur
Geburtsdatum	05.05.1976
Heimatort	Sumvitg
Zivilstand	ledig
Beruf	Elektromechaniker

Name, Vorname	**Lanicca Richard**
Adresse	Scesaplanastr. 9
PLZ, Wohnort	7000 Chur
Geburtsdatum	25.6.1977
Heimatort	Sarn
Zivilstand	ledig
Beruf	Elektroniker

Wir bestätigen, dass die vorliegende Semesterarbeit in allen Teilen selbständig erarbeitet und durchgeführt wurde.

Ort und Datum	Unterschrift
_____	_____
Ort und Datum	Unterschrift
_____	_____

Semesterarbeit an der Abteilung Telecom SS2001

Thema: **Architektur eines Honeypots**

Studenten: **Herr Bruno Cajacob,** bruno.cajacob@gmx.net
Herr Richard Lanicca, lari@gmx.ch

Partner: Herr Martin Lechmann, KPMG Information Risk Management Zürich, mlechmann@kpmg.com

Dozent: Herr Dr. Bruno Studer, bruno.studer@fh-htwchur.ch

Ausgabe: Dienstag, 17. April 2001
Abgabe: Donnerstag, 6. September 2001



1 Introduction

On the page http://project.honeynet.org/ a project was launched within the security community. The goal of this project is a defensive strategy to learn the tools, tactics, and motives of the blackhat community, and share those lessons learned. A group of 30 international security professionals collects information on their own time with their own resources. The primary method for obtaining information is through the use of a Honeynet. What is a Honeynet?

A Honeynet is a network designed to be compromised. It is a highly controlled network made up of various production systems. The productions system are the same systems commonly found on the Internet today, no systems nor vulnerabilities are emulated. The risk and vulnerabilities identified within the Honeynet are the same risks commonly found on the Internet.

Any traffic that either enters or leaves the Honeynet is controlled, captured, contained and analyzed. It is designed so that systems within the Honeynet can be compromised, but they cannot be used to compromise production systems on the Internet. This information is then analyzed to learn the tools, tactics and motives of the blackhat community.

2 Tasks / Activities

- Core Installations of different UNIX flavoured operating systems (OS) or Windows.
- A statistical analysis of the scan of the month with a follow up to possibilities and reactions and effects.
- Architecture suggestion (design and solution) to implement and apply such systems.
- Application of freeware and / or commercial intrusion detection systems (ids) like (snort etc.).
- Repair of a compromised system (ghost, imaging etc.).

In a practical part of the work, experiences should be collected

- implementation of a honeypot or a honeynet
- 24 hours online use / test on the internet (more days, weeks)
- analysis of the attacks made

3 Expected Results

The students should deliver:

- a time and project-plan.
- during the work, a form of a protocoll should grow up, which shows the progress of the work (views, problems, solutions etc.) so that KPMG can follow up the process and findings the students made.
- a written report (Schlussbericht) with an appendix (look at the guidelines "Allgemeinen Richtlinien und Bestimmungen für die Semesterarbeiten").
- At the end of the work a presentation to a technical forum where the findings are discussed and questions over the execution and results are answered.

4 Recommended literature

Hacking Exposed **2nd Edition**; Network Security Secrets & Solutions; Joel Scambray, Stuart McClure, George Kurtz, Bruce Schneier; McGraw-Hill Professional Publishing; ISBN: 0072127481

Links

http://project.honeynet.org/
http://www.enteract.com/~lspitz/

Chur, im April 2001

gez. M. Lechmann und B. Studer

Kurzzusammenfassung der Honeypot Semesterarbeit

Die folgende Dokumentation enthält einen Überblick über Honeypots, welche mit kostenloser Software arbeiten und einen Überblick über kommerzielle Tools, welche einen fixfertigen Honeypot zur Verfügung stellen. Ein Honeypot ist ein ungeschütztes Netzwerk mit Standardapplikationen, welche in den Firmen für gewöhnlich laufen. Im Hintergrund des Netzwerkes läuft dann das frei verfügbare IDS namens Snort, welches alle Aktivitäten aufzeichnet. So kann Snort Angriffe von aussen erkennen und in Logfiles abspeichern. Die Kommerziellen Tools arbeiten in ähnlicher Weise. Wir haben ein Honeypot im Telecomlabor aufgebaut und alle Tools ausprobiert. Dazu kam ein 30 tägiger Test. Die Resultate wurden ausgewertet und miteinander verglichen.

Zusammenfassung

In der heutigen Zeit wird vermehrt Wert auf das Sicherheitskonzept eines Netzwerkes gelegt. Dabei muss mit mehreren verschiedenen Möglichkeiten versucht werden, Angreifer den Zugriff auf das Netzwerk zu verunmöglichen. Normalerweise wird dies durch eine Firewall realisiert. Ausserdem sollte mit Intrusion Detection Systemen versucht werden, Angriffe zu erkennen und mit geeigneten Antworten den Angriff zu stoppen. Seit kurzer Zeit wird eine weitere Möglichkeit eingesetzt. Es wird ein sogenannter Honeypot in das Netzwerk gestellt. Dieser Honigtopf hat dabei zwei Hauptziele.
Das erste Ziel ist den Angreifer abzulenken und mit sinnloser Arbeit zu beschäftigen. Ein Eingreifer wird zuerst immer versuchen, ein „Loch“ in einer der Systemen zu finden, um dieses auszunützen und in das Netzwerk eindringen zu können. Da das Honeypot dem Angreifer genau solche Einstiegsmöglichkeiten anbietet, wird der Hacker diese auch nutzen. Ist der Angreifer nun schon so weit vorangekommen, wird er mit sinnloser Arbeit bei „Laune“ gehalten. Dies wir mit virtuellen Diensten oder vorgetäuschten Passwortlisten erreicht.
Während der Hacker versucht das System zu knacken, werden alle seine Bewegungen aufgezeichnet. Dies ist das zweite Hauptziel eines Honeypots. Damit kann sich der Netzwerk-Administrator einen Überblick verschaffen, wer dieser Angreifer ist und mit welchen Tools er versucht in das System einzudringen. Anhand dieser Informationen kann nun der Administrator eine geeignete Gegenmassnahme einleiten. Ausserdem können diese aufgezeichneten Informationen zur Beweisführung dienen, falls diese später benötigt werden.
In dieser Semesterarbeit haben wir uns mit dem Thema Honeynet auseinander gesetzt. Dabei wurden verschiedene Programme getestet und beschrieben. Damit man sich ein Bild machen kann, wie diese Anwendungen funktionieren und was sie alles aufzeichnen können. Weiter wurde ein kleines Honeynet aufgebaut und in einer DMZ plaziert. An dieser Stelle wurde das System 30 Tag lang Angriffen ausgesetzt, um einen solchen Aufbau in einer realen Umgebung auszutesten. In den Auswertungen wird ersichtlich, was für Angriffe tagtäglich geschehen und wie dieses Honeynet einen Penetration-Test durchlebt hat.

Inhaltsverzeichnis

ZUSAMMENFASSUNG **1**

VORWORT **4**

1 EINLEITUNG **5**

2 HONEYPOT **6**

2.1 WAS IST EIN HONEYPOT? 6
2.2 WIESO EIN HONEYPOT? 7
2.3 PROJECT HONEYNET 9

3 SITUATIONSANALYSE **10**

4 KONZEPT UND ZIELFORMULIERUNG **11**

5 REALISIERUNG **13**

5.1 EINLEITUNG 13
5.2 FREEWARE 13
5.2.1 Snort *13*
5.2.2 Snort auf NT *20*
5.2.2.1 Installation von Windows 2000 Server 20
5.2.2.2 Installation von Win Snort 20
5.2.2.3 ACID (Analysis Console for Intrusion Database) 26
5.2.2.4 IDS Center 28
5.2.3 Snort auf Linux *29*
5.2.4 The Deception Toolkit (DTK) *29*
5.3 KOMMERZIELLE TOOLS 30
5.3.1 Cybercop Sting *30*
5.3.1.1 Kurzbeschreibung 30
5.3.1.2 Bemerkungen: 30
5.3.1.3 Installation 31
5.3.1.4 Konfiguration 31
5.3.2 Specter *38*
5.3.2.1 Kurzbeschreibung 38
5.3.2.2 Bemerkungen: 38
5.3.2.3 Installation 38
5.3.2.4 Konfiguration 38
5.3.3 ManTrap *44*
5.3.3.1 Kurzbeschreibung 44
5.3.3.2 Bemerkungen: 44

6 CORE INSTALLATION **46**

6.1 MICROSOFT WINDOWS 46
6.2 SUN SOLARIS 46
6.3 INTEGRITÄT DER SYSTEME 46

7 UNSER HONEYNET **50**

7.1 AUFBAU EINES HONEYNETZES 50
7.2 STING 50
7.3 SPECTER 51
7.4 TEST 52

8 AUSWERTUNG UND ERGEBNISSE **53**

8.1 AUSWERTUNG SNORT 53
8.1.1 Auswertung über die ganze Testdauer mit ACID 53
8.1.2 Auswertung der Angriffe von Martin Lechmann (23.07.2001) 56
8.2 AUSWERTUNG SPECTER 65
8.2.1 Auswertung über die ganze Testdauer 65
8.2.2 Auswertung der Angriffe von Martin Lechmann (23.07. 2001) 68
8.3 AUSWERTUNG CYBERCOP STING 70
8.3.1 Auswertung über die ganze Testdauer 70
8.3.2 Auswertung der Angriffe von Martin Lechmann (23.07. 2001) 72
8.4 VERGLEICH DER RESULTAT MIT DER AUSWERTUNG VOM MARTIN LECHMANN 72
8.5 INTEGRITÄT DER SYSTEME 72

SCHLUSSWORT 73

QUELLEN- UND ABBILDUNGSVERZEICHNIS 75

LITERATUR 75
INTERNET 75

ANHANG 77

Vorwort

Schon vor Abschluss der ersten Semesterarbeit war uns klar, dass wir uns weiterhin mit dem Thema Internetsicherheit befassen wollen. Dadurch konnten wir auf unser erarbeitetes Wissen aus der ersten Semesterarbeit aufbauen. Dort beschäftigen wir uns mit dem IDS von Real Secure. In der aktuellen Semesterarbeit war Snort im Einsatz, welcher frei verfügbar ist und noch andere Tools um unser Honeynet aufzubauen.

Aus der ersten Semesterarbeit entstanden gute Kontakte mit der Firma KMPG in Zürich. Es war dann auch Martin Lechmann mit der Firma KPMG, welcher uns mit der Aufgabenstellung beauftragte. Für die gute und angenehme Zusammenarbeit, wollen wir Martin Lechmann danken. Auch unserem Dozenten Dr. Bruno Studer wollen wir für seine gute Betreuung danken. Er war stets mit Tipps und Ratschlägen bereit um unsere Semesterarbeit zum Erfolg zu verhelfen. Für die Unterstützung im Telecom Labor, wollen wir Adriano Zanoli hiermit auch recht herzlich danken.







1 Einleitung

Die Sicherheit im Internet ist in letzter Zeit immer öfters ein Thema. Immer wieder gibt es Meldungen über Angriffe auf Servern oder über gehackte Seiten. Denn durch den Zuwachs von automatisierten Scannern, wie ShadowScan, CybercopScanner, usw., die bekannte und weit verbreitete Sicherheitslücken ausnutzen, wird die Absicherung von Rechnern zunehmend schwierig. So sei der durchschnittliche Computer rund acht Stunden im Internet, bevor in diesen eingebrochen würde. Noch schlimmer sei die Lage bei Universitätsrechnern; ein ungesicherter Computer "überlebt" hier nur 45 Minuten, bevor sich ein Eindringling Zugriff verschafft.
Aus diesen Gründen ist es unerlässlich sich gegen solche Angriffe zu schützen. Eine Möglichkeit ist mit Intrusion Detection Systemen in Kombination mit Firewalls. In dieser Arbeit wird nicht darauf eingegangen, da wir uns schon in der ersten Semesterarbeit damit beschäftigt haben. In dieser Arbeit geht es um die Idee des Honeypots. Ein Honeypot ist ein ungeschütztes Netzwerk mit Standardapplikationen, welche in den Firmen für gewöhnlich laufen. Im Hintergrund des Netzwerkes läuft ein frei verfügbarer IDS namens Snort, welcher alle Aktivitäten aufzeichnet. So kann Snort Angriffe von aussen erkennen und in Logfiles abspeichern.
Unsere Semesterarbeit wurde in Zusammenarbeit mit der Firma KPMG realisiert. KPMG interessiert sich für Snort, wollte aber auch einen Vergleich mit kommerziellen Tools. Darum haben wir einige solcher Honeypot Tools angeschaut. Dies sind Cybercop Sting, Deception Toolkit, Specter und ManTrap.

Die folgende Dokumentation enthält also einen Überblick über Honeypots, welche mit kostenloser Software arbeiten und einen Überblick über kommerzielle Tools, welche einen fixfertigen Honeypot zur Verfügung stellen. Wir haben ein Honeynetz im Telecom Labor aufgebaut und alle Tools ausprobiert. Dazu kam ein 30 tägiger Test, welcher wir auswerteten. Damit wollen wir die Möglichkeiten und Probleme dieser Tools aufzeigen, die damit verbunden sind.

2 Honeypot

Angriffe von Hackern und Datenpiraten kosten Behörden und Unternehmen jährlich viele Milliarden Franken. Allein der passive Schutz durch Firewalls und Virenscanner verschlingt einen beträchtlichen Teil der Investitionen. Ein weiterer enormer Kostenfaktor sind inzwischen auch die Massnahmen zur Beseitigung von entstandenen Schäden. Jetzt gehen einzelne Unternehmen dazu über, den passiven Schutz durch aktive Massnahmen zu verstärken: So sollen die Angreifer demnächst der Initiative "Honeypot" auf den Leim gehen.

http://www.dradio.de/cgi-bin/user/fm1004/es/neu-computer/503.html

2.1 Was ist ein Honeypot?

Nach einem Bericht des Wall Street Journal ist eine pakistanische Hackergruppe, die im Juni 2000 in ein amerikanisches Netzwerk eingedrungen war, dabei einem Sicherheitsteam direkt in die virtuellen Arme gelaufen. Den Pakistanern soll verborgen geblieben sein, dass ein Sicherheissystem namens Honeypot (Honigtopf) jeden ihrer Schritte aufzeichnete. Über mehrere Monate hätten Sicherheitsexperten jede einzelne Eingabe der Hacker in die Tastatur ihrer Computer aufzeichnen können. Jede eingesetzte Software konnte erkannt werden; sogar Chat-Sessions seien mitprotokolliert worden. Diese Sicherheitsexperten sind Mitarbeiter des Project Honeynet.

http://www.internetworld.de/sixcms/detail.php?id=48

Dabei wollen wir zuerst noch auf die unterschiedlichen Begriffe Honeypot und Honeynet eingehen.

- Honeypot: Im allgemeinen ist ein Honeypot ein System, welches bekannte Sicherheitsmängel und das Verhalten anderer Systeme simuliert. Gleichzeitig werden die Angriffe geloggt und analysiert. Beispiele solcher Honeypots sind: The Deception Toolkit, CyberCop Sting, Specter und Mantrap.

- Honeynet: Ist in erster Linie ein eine Möglichkeit um zu lernen. Es ist ein Netzwerk mit ganz normalen Produktionssystemen und denselben Sicherheitsstandards wie die meisten Firmen sie verwenden. Wird das Netzwerk angegriffen, werden diese Angriffe analysiert. Aus den Analysen kann man die Netzwerksicherheit verbessern. Diese Methode wird vom Project Honeynet angewendet.

In unserer Semesterarbeit haben wir beide Varianten eingesetzt aus folgenden Gründen:

1. Unser Honeynet wird mittels einem Freeware Tool aufgebaut (Snort).
2. Um die Freeware mit kommerziellen Tools vergleichen zu können, haben wir uns entschieden auch diese einzusetzen.
3. Die Kunden der Firma KPMG bevorzugen die kommerzielle Tools, weil der Support gewährleistet ist.

2.2 Wieso ein Honeypot?

Um diese Frage zu beantworten, sollte man sich zuerst bewusst werden, was für Möglichkeiten oder Fähigkeiten ein Hacker hat. Dazu wurden vier Fälle aufgelistet:

- FBI warnt vor Chinesischen Hackerangriffen auf US-Regierungsseiten

New York, 01.05.2001, Pro-Chinesische Hacker haben die Websites von zwei US-Ministerien geknackt. Die Internet-Angebote des Arbeits- und des Gesundheitsministeriums haben für mehrere Stunden gesperrt werden müssen, berichtet die New York Times.

Die Hacker stellten ein Foto des chinesischen Kampfpiloten auf die Homepage des Arbeitsministeriums, der am 1. April bei der Kollision mit einem US-Spionageflugzeug abgestürzt war. „Das ganze Land bedauert den Verlust seines besten Sohnes – Wang Wie, Du wirst uns bis zum Ende aller Tage fehlen“, stand laut Times unter dem Bild.



Amerikanische Hackergruppen riefen unterdessen zum Cyberwar auf. Zwischen 22 und 23 Schweizer Zeit wurden bereits mehr als 30 chinesische Regierungsites gehackt

http://www.computerchannel.de/news/ticker/viren_sicherheit/9720.phtml

- Microsoft Angriff

Laut Aussage eines leitenden Microsoft-Angestellten, war der Fehler eines Mitarbeiters für den jüngsten Einbruch in das Firmennetzwerk verantwortlich. Dem Hacker gelang es dabei, Zugriff auf Teile des streng geheimen Microsoft Source-Codes zu bekommen. Im ersten Schritt versandte der Eindringling eine Mail mit einem sog. 'Trojaner' an einen Angestellten. Dadurch erhielt er Zugriff auf das System. Einmal drin, brauchte er nur nach einem Rechner zu suchen, für den kein Zugangspasswort definiert wurde. Schliesslich fand er einen Server, der ursprünglich für einen Kunden vorgesehen war. Bei diesem hatte ein Microsoft-Administrator kein Passwort vergeben und so letztendlich den Zugriff möglich gemacht.

- Swissonline Hack

Die SonntagsZeitung hat am 22.07.2001 ein Datenleck bei Swissonline publik gemacht. Einem Hacker war es gelungen, den Zugang zu 250 000 E-Mail-Konten zu knacken. Der Mann übergab der SonntagsZeitung eine CD-ROM mit 185 000 E-Mail Adressen und demonstrierte den Zugriffsweg zu einer beliebigen Adresse. Noch in der Nacht auf den Sonntag bestätigte ein Sicherheitsfachmann von Swissonline das Datenleck. In der Nacht zum Donnerstag nahm Swissonline den Mailserver vom Netz, um «sicherheitsbedingte Wartungsarbeiten vorzunehmen», wie es hiess. Swissonline bedauerte gegenüber den Kundinnen und Kunden in einer

Stellungnahme, dass es einem Hacker gelungen sei, Kundendaten aufzuspüren. «Selbstverständlich haben wir sofort entsprechende Massnahmen ergriffen und diesen aussergewöhnlichen Zugriffsweg versperrt», hiess es in der Stellungnahme weiter. Der Hacker, der anonym bleiben will, erklärt in einem Interview, wie er zu diesen Kundendaten gelangte und warum er es der SonntagsZeitung geschickt hat. Dieses Interview kann im Anhang der Dokumentation nachgelesen werden.

- Hacked by Illegal Crew

Wie oben schon erwähnt, hinterlassen die Hacker gerne ihre Spuren. Oft hinterlassen sie auch Hinweise auf die Sicherheitsmängel, oder wie sie ins Netzwerk eingedrungen sind.



Orginal Seite:
http://firstrespondersupplies.com/

Seite nach dem Hacking ⟶



Aus den unterschiedlichen Artikeln ist die Gefahr, welche von Hackern ausgeht, gut zu sehen. Dagegen muss etwas unternommen werden. Eine Variante bietet das Honeynet.

2.3 Project Honeynet

Lance Spitzner ist nicht nur oberster Sicherheitsingenieur von Sun, sondern auch Gründer des Projektes Honeynet (http://www.honeynet.org). Die Honeynetler haben eine typische Netzwerkumgebung mit Switches, Routern, Firewalls, DNS-Server (Linux), Webserver (Windows NT), FTP-Server (Solaris) und herkömmlichen Applikationen ans Internet angeschlossen und mit allerlei Überwachungssoftware und Einbruchs-Erkennern (IDS) bestückt. Jeglicher Verkehr ins und aus dem Internet wird aufgezeichnet und mit parallellaufenden, sauberen Systemen verglichen. Damit kann man detailgenau herausfinden, mit welchen Techniken die Angreifer arbeiten, was sie alles so treiben und wo die Schwächen der einzelnen, mit Standardkonfigurationen arbeitenden Betriebssysteme liegen. An Honeynet arbeiten rund dreissig Sicherheitsexperten und Hacker mit, Personen also, die in nicht wenigen Fällen beide Rollen in sich vereinen.



Mit Erfolg. „Wir haben in letzter Zeit viel über die Methoden der Hacker gelernt“, so Spitzner. Ein weiteres Ergebnis: „Mit automatischen Scannern, die Server auf Schwachstellen untersuchen, ist ein Durchschnittsserver in acht Stunden gehackt“. Noch einfacher seien, so ein Sicherheitsexperte, Uni-Netzwerke zu knacken. Über einen ungesicherten Rechner hätten findige Studenten nach 45 Minuten die Herrschaft übernommen. In einem Wettbewerb schaffte es der Gewinner, in einer Minute ins Netzwerk einer Universität einzudringen, wo er sich dann eine halbe Stunde aufhielt. Um sich ein genaues Bild zu machen, was er in diesen 30 Minuten alles getrieben hat, braucht ein Wissenschaftler im Durchschnitt 34 Stunden. Ein teuerer Spass: „Eine 34-stündige Untersuchung kostet 2000 Dollar, wenn sie interne Mitarbeiter durchführen. Für externe Berater sind gar satte 22000 Dollar zu berappen“, sagt David Dittrich, Sicherheitsingenieur der Universität von Washington und Honeynet-Mitarbeiter. Die Einbrüche via Web ins Honeynet geschehen trotz Firewalls und IDS. Die Technologien und Tools haben zuletzt grössere Fortschritte gemacht. Das Problem auf Anwenderseite: Obwohl immer mehr Anwender solche Tools aus Sicherheitsgründen einsetzen, kann nur ein kleiner Bruchteil die Werkzeuge richtig handhaben und deren Daten und Informationen zutreffend interpretieren. „Die Ausbildung von Systemverwaltern und Anwendern zeigt nur zum Teil Erfolg, weil sie häufig nicht mit echten und permanenten Lernen zu bewegen sind“, sagt Martin Roesch, dessen Firma Source-Fire den bei Honeynet eingesetzten Einbruchserkenner Snort herstellt. „Sicherheit ist kein Produkt und kein Zustand, sondern ein Prozess“, ergänzt Harald von Fellenberg, Consultant von Sun Schweiz.
Der Feind ist jedoch häufig in den eigenen Reihen zu suchen: Eine gemeinsame Studie des Computer Security Institute (CSI) und des FBI aus dem Jahr 2000 kam zu dem Ergebnis, dass 70% von 640 US-Firmen einen oder mehrere nichtautorisierte Netzwerk- und Datenzugriffe durch interne Mitarbeiter zu verzeichnen hatten. Hingegen hatte nur jede dritte Firma mit externen Eindringlingen Bekanntschaft gemacht. Der Schaden, der durch externe Angreifer angerichtet wurde, betrug durchschnittlich 25‘000 Dollar. Interne Attacken kamen hingegen auf 1,8 Millionen Dollar zu stehen.

Computerworld 20. April 2001 Nr. 16/2001

3 Situationsanalyse

Die ganze Semesterarbeit teilten wir grob in drei Bereiche auf:

Phase 1:
- Einarbeitung in das Thema
- Kennenlernen von Snort
- Kommerzielle Tools suchen und testen
- Planung des Aufbaus des Honeynetzes

Phase 2:
- Ausarbeitung des Testaufbaus
- Integrität der Rechner gewährleisten
- Test
    - Eigene Tests
    - Test von Martin Lechmann
    - Öffentliche Test (z.B. externe Hacker)

Phase 3:
- Auswertung der Logfiles
- Vergleich der verwendeten Tools
- Vergleich mit den Daten von Martin Lechmann

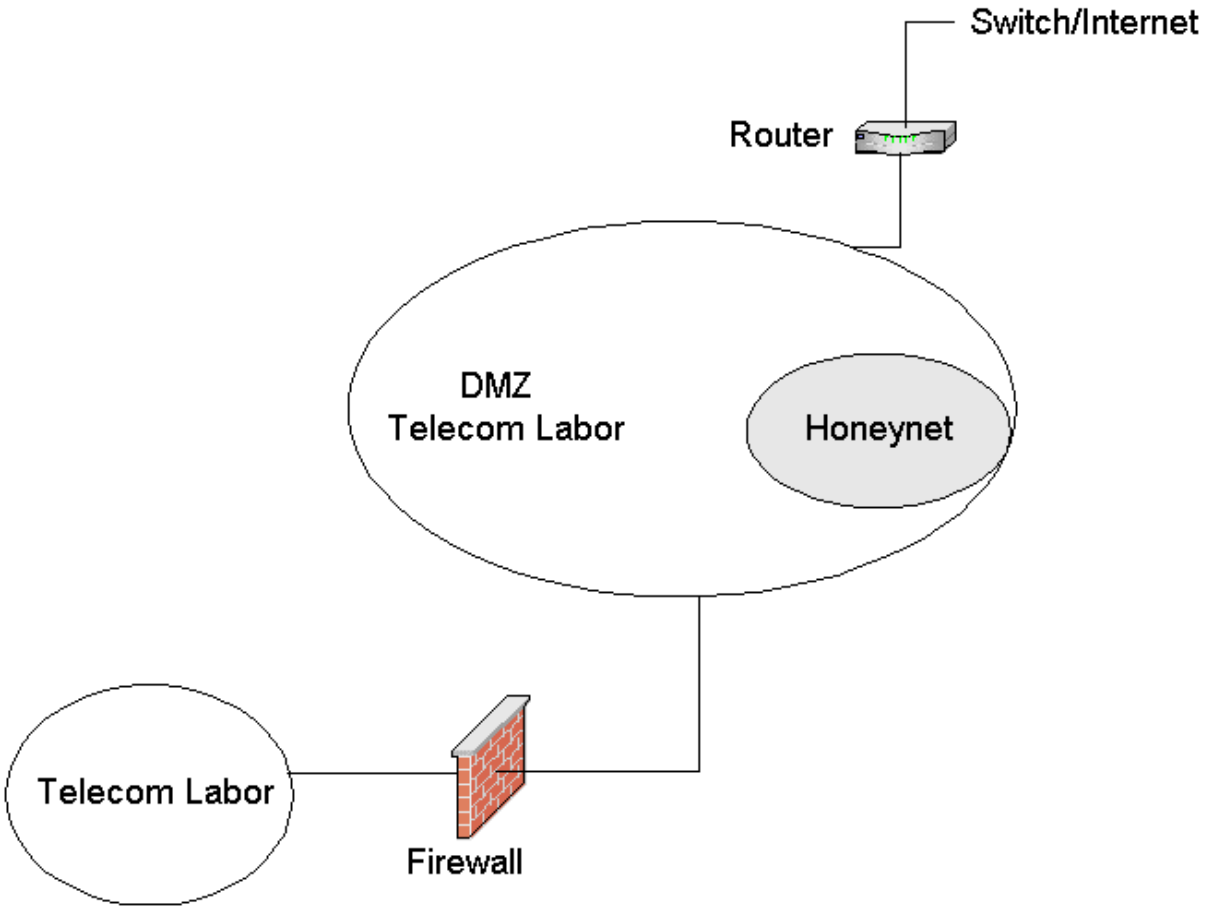
Unsere Hauptziele:
- Über das Thema Wissen aneignen
- Diverse Tools kennenlernen
- Vergleich der eingesetzten Tools
- Vor-/Nachteile eines Honeynetzes
- Einsatzgebiet für solche Programme

4 Konzept und Zielformulierung

Als erstes musste der Standort für den Aufbau eines Honeynetzes abgeklärt werden. Dabei kamen mehrere Möglichkeiten in Frage:

1. Der zugewiesene Arbeitsplatz im Telecom Labor
2. Bei unserer Partnerfirma KPMG in Zürich
3. Bei einem Provider mit viel Internetverkehr

Der Arbeitsplatz im Telecom Labor machte für uns keinen Sinn, da dieser relativ gut gegen das Internet geschützt ist und wir so keine aussagekräftigen Bewertungen machen könnten. Sofern wir an diesem Standort überhaupt irgendwelche Angriffe registrieren hätten können, ausser Angriffe von intern.
Der Standort Zürich bei unser Partnerfirma KPMG wäre sicher ein guter Ort. Aber wir müssten einen Platz vor oder neben Ihrer Firewall bekommen. Ausserdem wäre der Aufwand immer nach Zürich zu reisen sehr gross. Nach mehreren Gesprächen mit Martin Lechmann kamen wir zum Schluss, dass diese Alternative für uns als nicht optimal gewesen wäre.
Nun blieb uns noch die Idee mit einem Indernetprovider. Nach mehreren erfolglosen Telefongesprächen mit einigen Providern in der Umgebung von Chur stellten wir die Suche ein. Auch aus dem Grunde, da wir eine sehr gute neue Lösung gefunden hatten. Wir erhielten einen Platz vor der Firewall des Telecom Labors. An dieser Stelle möchten wir Herrn Zanoli vom Telecom Labor danken für die Unterstützung. Und hier nun der Aufbau:



Für das Honeynetz erhielten wir ein eigenes Subnetz, welches sich in der DMZ des Telecom Labors befand. Die Daten für das Subnetz waren:

Subnet:	195.176.242.216
Mask:	255.255.255.248

Wie zu erkennen ist, sind in diesem Subnetz nur sechs nutzbare IP-Adressen vorhanden. Zusätzlich konnten wir die IP-Adresse 195.176.242.217 auch nicht benutzen, da diese für den Übergang von unserem Honeynet zur DMZ gebraucht wurde. Also blieben für unser Netz noch genau fünf IP-Adressen. Aus diesem Grunde mussten wir genau Überlegen, wie unser Honeynet aufgebaut werden sollte. Ausserdem wurde für unser Subnetz alle Ports zum Internet gerichtet geöffnet, da die meisten standardmässig in der DMZ des Telecom Labors gesperrt sind.

Als Computer standen uns drei Dell Optiplex GX1 (450 MHz, 128 MB Ram und 8 GB HD) zur Verfügung.

5 Realisierung

5.1 Einleitung

In der ersten Phase unserer Semesterarbeit ging es darum, Programme rund um das Thema Honeypot zu suchen und auszutesten. Nach einer ausführlichen Suche in der Literatur, Computer Zeitschriften und dem Internet hatten wir folgende Programme gefunden:

- Snort (Bestandteil der Semesterarbeitaufstellung) [31]
- The Deception Toolkit (Fred Cohen & Associates) [39]
- ManTrap (Recouse Technologies) [42]
- Specter (Netsec) [40]
- CyperCop Sting (PGP Security) [41]

5.2 Freeware

5.2.1 Snort

Was ein IDS ist und wie es funktioniert, darauf wird in dieser Semesterarbeit nicht mehr eingegangen. Mehr Informationen darüber findet man in unserer letzten Semesterarbeit [25] *Sicherheit im Internet – Firewall und Intrusion Detection Systems.*

Auf dem Markt gibt es eine ganze Reihe von kommerziellen Intrusion Detection Systemen (IDS), wie der Real Secure von ISS (Internet Security Systems). Diese Systeme sind in der Regel aber sehr teuer.
Doch es geht auch anders. Snort ist ein kostenloses IDS, welches von der offiziellen Homepage [31] runtergeladen werden kann. Snort gibt es in einer Version für Linux und für Windows. Die Entwickler von Snort bezeichnen ihr Produkt als Leight weight IDS, doch mit seinen Fähigkeiten braucht es sich von den kommerziellen Tools nicht zu verstecken.



Snort arbeitet wie die kommerziellen IDS mit Angriffssignaturen, die die Eigenschaften verdächtiger IP Pakete beschreiben. Zu den Kennzeichen gehören die verwendeten Protokolle (TCP, UDP, ICMP) und eine Anzeige für die Portscans. Alle Angriffe werden mit Source- und Destinationadress beschrieben und dazu noch eine Datum und Zeitangabe.

Die Regeln müssen nicht unbedingt selber erstellt werden. Für eine spezielle Konfiguration für das eigene Netzwerk, ist es aber trotzdem ratsam eigene Regeln zu erstellen. Es gibt eine gute Anleitung dazu [32]. Vom Snort-Team werden über 1000 fertige Regeln bereitgestellt. Diese spüren von ungewöhnlichen ICMP Pakete über Backdoor Signaturen bis hin zu Angriffe auf CGI-Skripte auf.

Gründe um sich für Snort zu entscheiden sind:

- Es ist kostenlos
- Die Prgrammiersprache für die Regeln ist einfach zu erlernen
- Snort hat eine aktiv unterstützte Regel Datenbank
- Es gibt ein „Community Support“ für Snort
- SiliconDefense.com bietet einen kommerziellen Support für Snort an
- Whitehats.com zusammen mit ACID, bietet eine gute Auswertung von Snort

Sensor: Ein Sensor ist die Komponente eines IDS, welches die Überwachung des Verkehrs verwaltet. In unserem Fall ist das der Rechner, wo Snort installiert ist.

Signatur: Die Reihenfolge oder der Inhalt von IP Paketen werden gebraucht um eine Attacke zu identifizieren.

Wir setzen die Snort Version 1.7 ein. Im Laufe der Semesterarbeit kam die neue Version 1.8 heraus. Die neue Version wurde weiter verbessert und enthält jetzt neue Plugins und neue Optionen für Regeln. Für unseren Zweck machte es aber keinen Sinn auf 1.8 umzustellen, da unsere Tests schon mit 1.7 liefen.

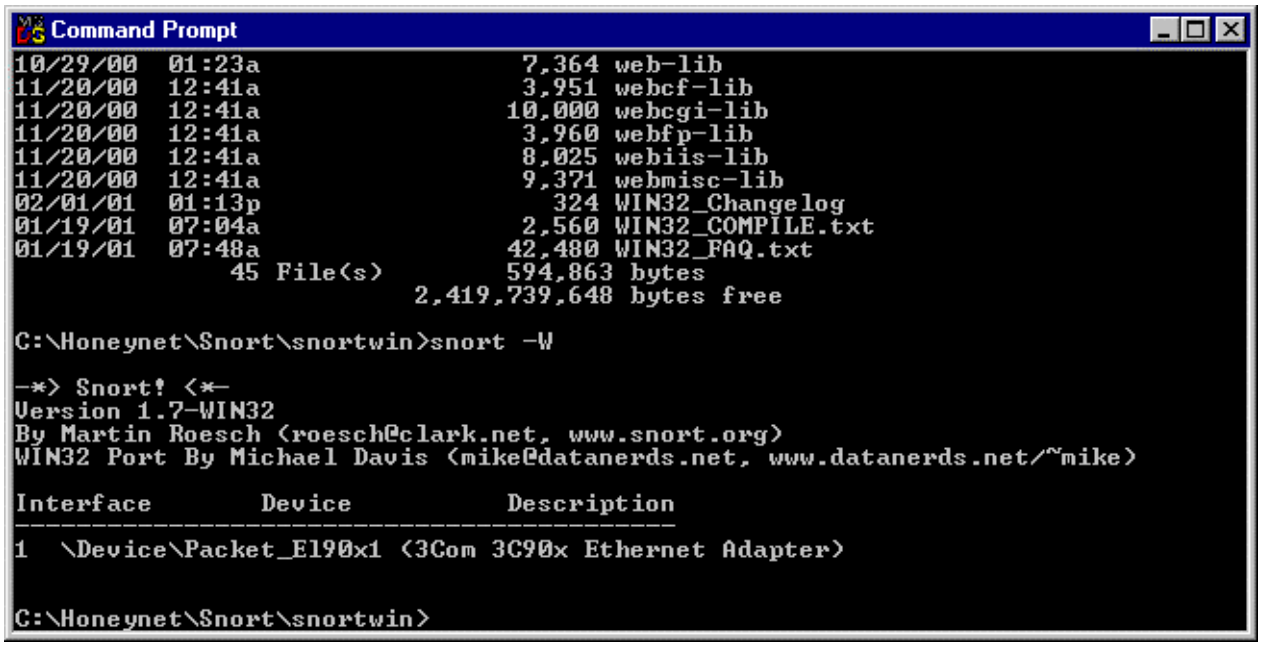
Snort kann über das Command Prompt aufgerufen werden, nachdem es installiert wurde.

- Snort Installation

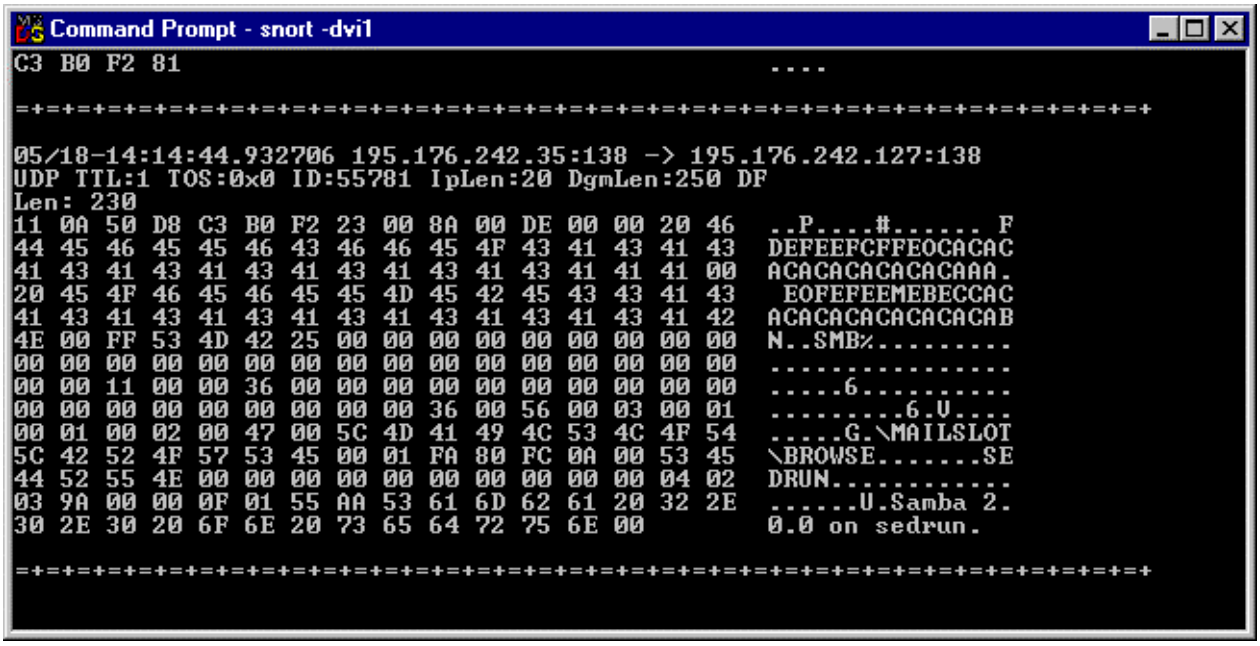
File download: Snort-1.7-win32 Binary (Windows Version)

Im Command Prompt: ins Verzeichnis gehen wo der Binary Code abgespeichert ist und snort.exe ausführen.
Beim Ausführen von snort.exe kommt die Fehlermeldung packet.dll not found. Dies bedeutet das etwas fehlt. Aus dem FAQ von snort.org ist ersichtlich, dass ein Programm namens WinPcap fehlt. WinPcap ist eine allgemeine Library, die den Hardwarezugang zur Netzwerkkarte erlaubt. Die libpcap ist Bestandteil vieler moderner Unixe. Die WinPcap ist die dazu kompatible Windows Umsetzung.
Erhältlich unter http://netgroup-serv.polito.it/winpcap/install/default.htm

Mit `Snort -W` werden die vorhandenen Interfaces des PC aufgelistet. Im Screeshot ist nur ein Interface sichtbar.

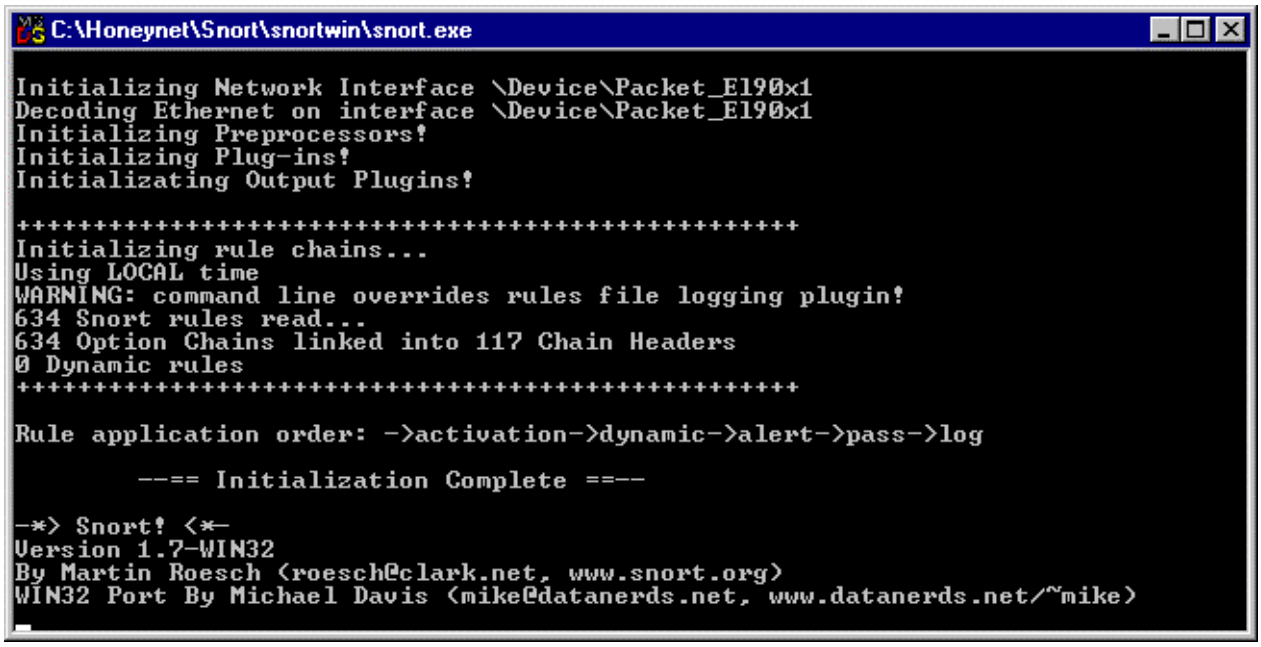


Mit `snort -dvi1` wird das Interface 1 ausgewählt, in unserem Fall das von 3Com. Nachher ist Snort schon in Betrieb. Im Command Prommt sind die Pakete sichtbar die er gerade anschaut.

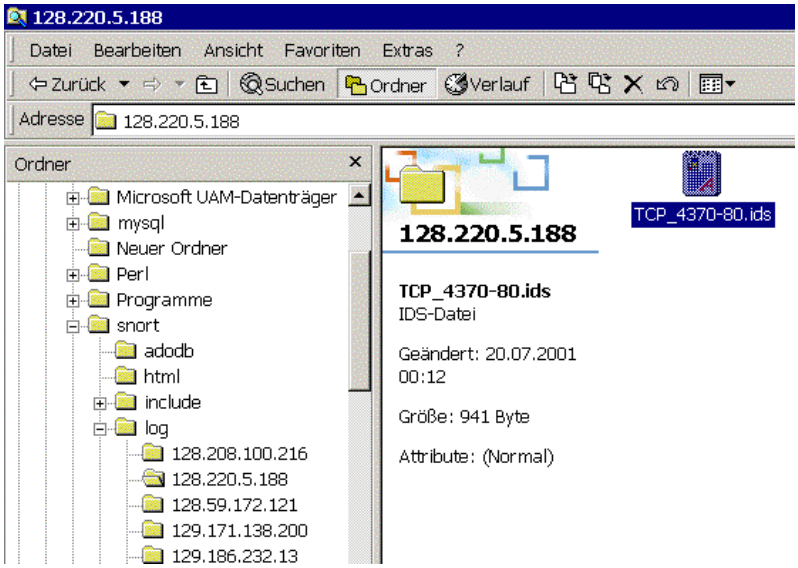


Man sieht Zeit und Datum und die IP des Rechners von welchem das Paket kommt und zu welchem Rechner es geht. Dann ist der Inhalt des Paktes in Hex kodiert und daneben auskodiert mit ACII-Zeichen.

- Snort ist am aufzeichnen:



Ordner angeben, wo die Log Dateien gespeichert werden sollen, mit dem Befehl `snort -l C:\Honeynet\Snort\snortwin\log.`



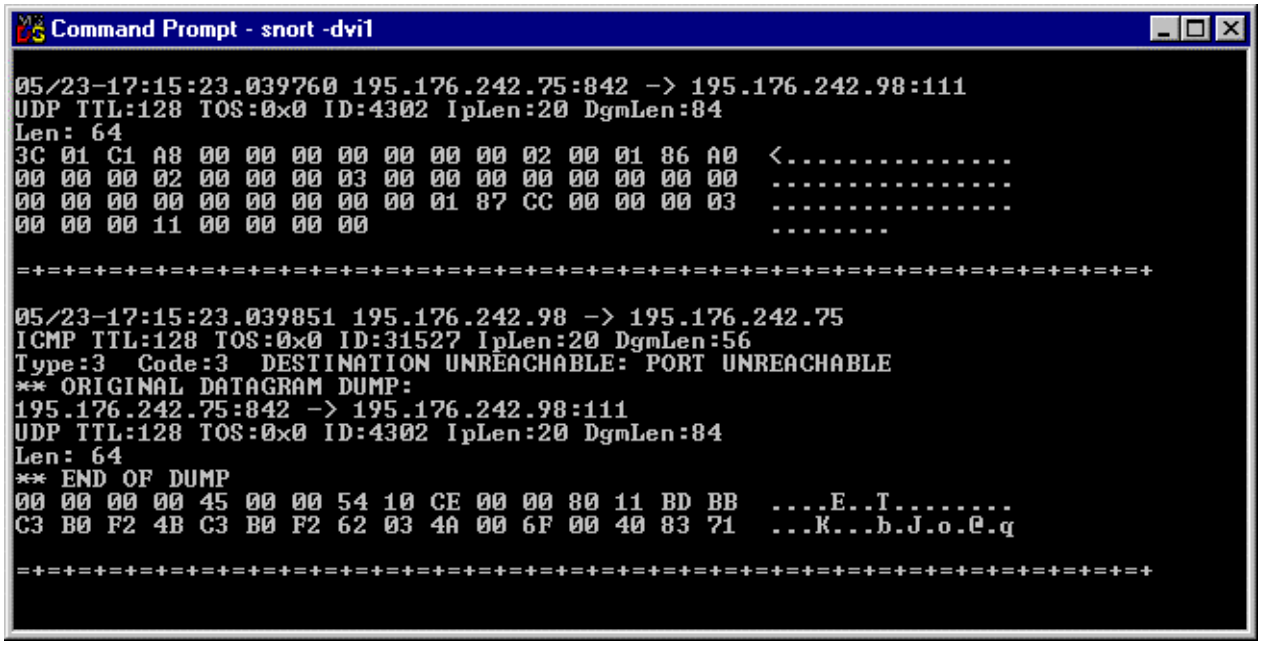
Man sieht, dass Snort für jede IP einen Ordner anlegt. Die Daten werden in Textfiles abgelegt. Diese auszuwerten ist recht schwierig. Unten ist eine solche Auswertung von einem Textfile aufgelistet.

```
[**] IDS296/web-misc_http-whisker-splicing-attack-space [**]
07/20-00:12:41.834549 128.220.5.188:4370 -> 195.176.242.219:80
TCP TTL:113 TOS:0x0 ID:46764 IpLen:20 DgmLen:44 DF
***AP*** Seq: 0x81FFC9AF  Ack: 0x3A332BC6  Win: 0x40B0  TcpLen: 20
=+=+=+
```

Um die Auswertung angenehmer zu gestalten, wurde im Kapitel 5.2.2 ein Auswertungstool namens ACID installiert. ACID holt die Daten aus einer Datenbank und stellt diese übersichtlich in Tabellen und Grafiken dar. Mehr davon im nächsten Kapitel.

- Verhalten von Snort bei einem Portscan

Durch systematisches Scannen aller Ports kann ein Angreifer feststellen, welche Dienste und Prozesse auf der Zielmaschine derzeit ablaufen. Das Scannen kann bei einigen Betriebssystemen auf Grund von Implementierungsschwächen zu einem Denial-of-Service oder sogar zu Systemabstürzen führen. Die erhaltenen Informationen können für weitere Angriffe verwendet werden (z.B. Sniffing).



Aus dem Snort Screeshot ist ersichtlich, dass er die nicht verfügbaren Ports mit PORT UNREACHEBLE definiert. Damit ist dem Angreifer klar, welche Ports offen und welche geschlossen sind.

- Snort Befehle

Eine Auflistung was mit den Snort alles gemacht werden kann und die Befehle dazu.

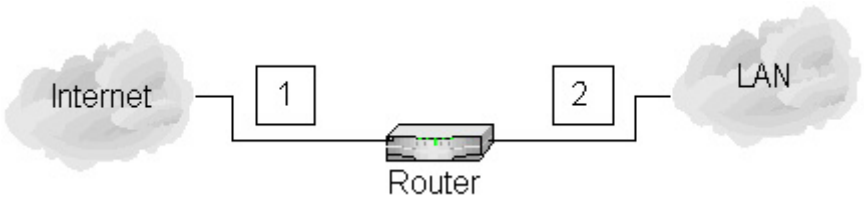
-A		Set alert mode: fast, full, or none (alert file alerts only „unsock“ enables UNIX socket logging (experimental)). *
-a		Display ARP packets
-c	<rules>	Use Rules File <rules>
-C		Print out payloads with character data only (no hex)
-d		Dump the Application Layer
-D		Run Snort in background (deamon) mode
-e		Display the second layer header info
-E		Log alert messages to NT Eventlog
-F	<bpf>	Read BPF Filters from file <bpf>
-g	<gname>	*Run snort gid as <gname> user or uid after initialization
-h	<hn>	Home network = <hn>

-i	<if>	Listen on interface <if>
-I		Add Interface name to alert output
-l	<ld>	Log to directory <ld>
-n	<cnt>	Exit after receiving <cnt> packets
-N		Turn off logging (alerts still work)
-o		Change the rule testing order to Pass|Alert|Log
-O		Obfuscate the logged IP addresses
-p		Disable promiscuous mode sniffing
-P	<snap>	Set explicit snaplen of packet (default: 1514)
-q		Quiet. Don't show banner and status report
-r	<tf>	Read and process tcpdump file <tf>
-s	<server:port>	Log alert messages to syslog server (default port: 51)
-S	<n=v>	Set rule file variable n equal to value v
-t	<dir>	Chroots process to <dir> after initialisation
-u	<uname>	Run snort uid as <uname> user (or uid) after initialisation
-U		Use UTC for timestamps
-v		Be verbose
-W		Lists available interfaces
-V		Show version number
-X		Dump the raw packet data starting at the link layer
-?		Show this information

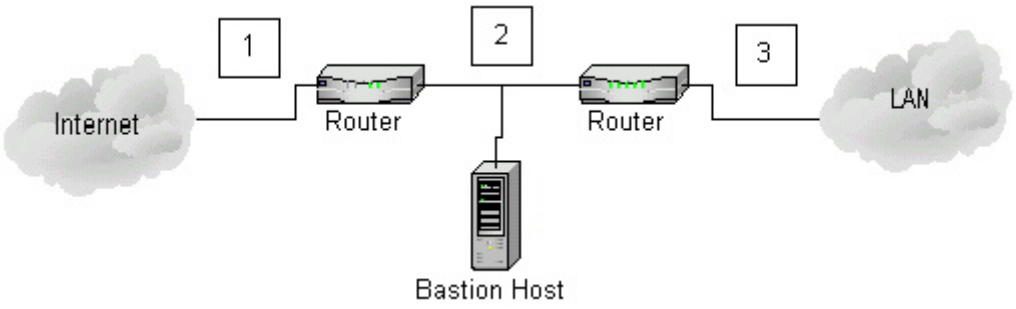
<Filter Options> are standard BPF options, as seen in TCPDump
* denotes an option that is NOT SUPPORTED in this WIN32 port of snort.

- Einsatz im Netzwerk:

In einem simplen LAN ohne DMZ hat man zwei Optionen, um den Sensor zu plazieren. Zwischen dem Router und dem Internet (1) oder zwischen dem Router und dem LAN (2). Die erste Konfiguration (1) wird die Attacken gegen den Router detektieren aber er weiss nicht genau welche Attacken der Router durchlässt und ins LAN gelangen. Mit der zweiten Variante (2) sieht man welche Attacken ins LAN eindringen.



In einem Netzwerk mit DMZ und Basition Hosts hat man drei Möglichkeiten um den Sensor zu platzieren. Die Situation ist hier ein wenig komplexer. Die Bastion Hosts können verschiedene Dienste anbieten und auf verschiedene Betriebssysteme laufen.



Wenn man den Sensor an der Stelle (1) platziert, dann kann man alle Angriffe gegen das Netzwerk mit all seinen Komponenten detektieren.
Ein Sensor platziert an der Stelle (2) wird alle Attacken erkennen, welche es durch den externen Router hindurch geschafft haben. Wenn man also alle Angriffe in die Bastion Host sehen will, muss der Sensor hier platziert werden.

An Stelle (3) werden die Attacken erfasst, welche den LAN erreichen, aber keinen Aufschluss über die Attacken gegen die Bastion Host geben.

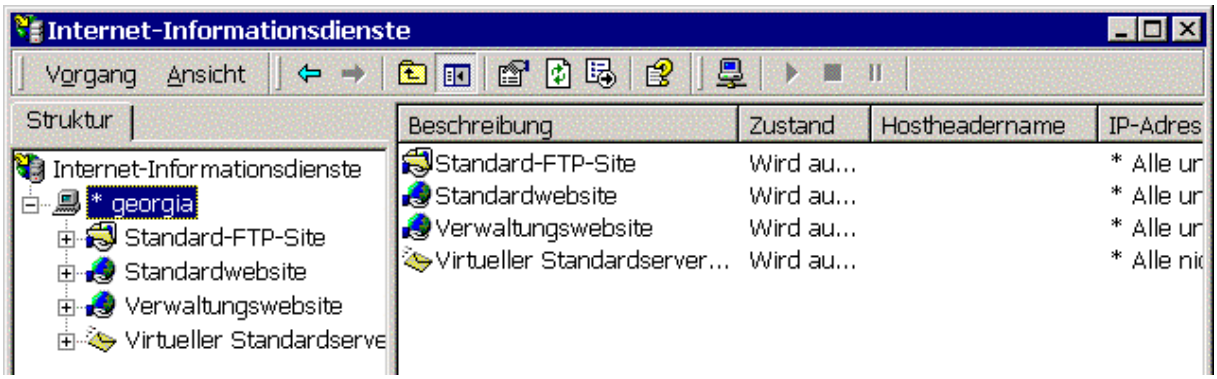
Die Position (1) des Sensors mag auf den ersten Blick nicht sinnvoll erscheinen. Wenn anstatt des Routers eine Firewall an der Position (1) steht, so kann mit einem Vergleich von Sensor (1) mit Sensor (2) verglichen werden, wie gut die Firewall konfiguriert ist.

5.2.2 Snort auf NT

Wie aus Kapitel 5.2.1 ersichtlich ist, ist das Arbeiten mit Command Prompt recht mühsam. Darum haben wir eine andere Lösung gesucht und gefunden. Die Lösung ist die Verwendung einer Datenbank zur Verwaltung der Daten und ACID (Analysis Console for Intrusion Database) um die Daten auszuwerten.
Um diese Tools zu installieren, konnten wir ein Autoinstallations File benutzen. Damit das ganze funktioniert musste ein Windows 2000 Server zur Verfügung stehen. Die ganze Installation und Angaben zu ACID werden im folgenden Kapitel beschrieben.

5.2.2.1 Installation von Windows 2000 Server

Als erstes geht es einmal darum den Windows2000 Server zu installieren. Die Installation ab CD bereitet keine grossen Probleme. Die Konfiguration des Servers ist dann etwas komplexer sowieso wenn man zum ersten mal damit arbeitet. Der Server kann über Programme->Verwaltung->Konfiguration des Servers eingestellt werden.



Für unsere Zwecke wurden hauptsächlich Änderungen im Menupunkt Webserver gemacht. Dort müssen noch Einstellungen unter dem Punkt „Haupteigenschaften des WWW-Dienstes für georgia„ (Name des Rechners im Tlab) gemacht werden, damit man direkt auf die Startseiten zugreifen kann. Man muss Default.htm, Default.asp, index.html, index.php unter „Dokumente“ hinzufügen.

5.2.2.2 Installation von Win Snort

Snort wird mit einem Installationsskript automatisch installiert, es wurde von Richard Howlett geschrieben. Das Skript wird auf der Snort Seite [31] zur Verfügung gestellt. Es muss aber noch für die eigenen Bedürfnisse angepasst werden und einige Fehler müssen behoben werden.
Das Installationsfile erleichtert die Arbeit bezüglich Download der Dateien und Konfiguration. Es garantiert aber nicht, dass es bis in aller Ewigkeit ohne Probleme ausführbar ist. Schon wir mussten das File anpassen, einige Adressen hatten z.B. geändert. Die Idee die ganze Installation zu automatisieren ist aber sehr gut. Für unsere Zwecke war es ideal. Es sollte aber für kommerzielle Zwecke anders gelöst werden.

Das Auto Installations Batch-File sollte also folgende Arbeiten ausführen:

- Download aller benötigten Files, wenn sie nicht schon existieren
- Auspacken der Files und in einem temporären Ordner kopieren
- Installation von Winpcap
- Installation Snort for Windows mit mySQL Support
- Download der aktuellen Regeln
- Installation von Activestate Perl
- Installation von Snortsnarf (Wird von uns nicht gebraucht)
- Installation von mySQL und Datenbank als Dienst einstellen
- Erstellen der Snort Datenbank, Tabellen und User in mySQL

- Installation und Konfiguration von Acid, einschliesslich php405 und Adodb

Bei der folgenden Beschreibung der Installation wird erklärt, wie wir vorgegangen sind. Man sieht, dass wir mit dem Installationsfile immer nur Schritt für Schritt vorwärts gekommen sind. D.h. wir haben autoinstall.bat ausgeführt. Das File lief bis zu einem Problem ab, dann haben wir das Problem behoben. Die Behebung des Problems ist dann immer beschrieben. So gingen wir von Problem zu Problem, bis das File ohne Fehler ausführbar war. Wir haben dies aus dem Grund so dokumentiert, da es einen besseren Einblick in unsere Arbeit gibt und zeigt auf, dass Vieles was man von Internet downloadet nicht auf Anhieb funktioniert.

Schritt 1:

Um mit diesem Installationsfile arbeiten zu können, müssen zuerst ein paar Tools zu Verfügung gestellt werden. Am besten kopiert man dies in einem separaten Ordner, bei uns heisst dieser Snortfiles.

Dies sind:
- Autoinstall.bat
- Pkzc400s.exe
- Wget5-i.6b.zip
- WinPcap.exe
- Winzip80.exe

Schritt 2:

WinZip, PkZip und WinPcap installieren.

Schritt 3:

Autoinstall.bat ausführen. Die Ordner „snort, sntemp, tmp“ werden erstellt.

Schritt 4:

Um die gewünschten Dateien aus dem Internet automatisch runterzuladen, wird das Programm Wget eingesetzt. Damit dieser auch innerhalb des Telecom Labor funktioniert, muss noch ein kleines *.ini File erstellt werden, damit Wget weiss, über welchen Rechner er gehen muss. Dieses File wurde Wget.ini benannt und sieht folgendermassen aus.

```
http_proxy = http://trimmis.tlab.ch:8080
ftp_proxy = http://trimmis.tlab.ch:8080
use_proxy = on
```

Wget entzippen und Wget.exe und Wget.ini in Ordner „sntemp“ verschieben.

Schritt 5:

Dann wieder Autoinstall.bat ablaufen lassen. Jetzt sucht das File die mit dem Pfad angegebenen Dateien und installiert diese in den vorgesehen Ordnern. Achtung! Die runtergeladenen Programme müssen dann noch von Hand installiert und ausgeführt werden. Den Download kann man gut im CommandPrompt verfolgen, die aktuell runtergeladene Software wird mit Prozentangaben versehen um den Stand anzugeben.

Schritt 6:

Die fertig runtergeladenen Dateien IDS Center und mysql installiert. Das Programm für Perl ist automatisch installiert worden, man sieht das daran, dass man bei der Installation nichts mehr auswählen kann.

Schritt 7:

Änderung an Snort1.conf

```
var INTERNAL 195.176.242.216/24
var EXTERNAL any
# !$INTERNAL
#
# add preprocessors here
preprocessor defrag
preprocessor stream: timeout 23, ports 21 23 25 80 110 143, maxbytes 16384
preprocessor http_decode: 80 2301
preprocessor portscan: $INTERNAL 5 5 c:\snort\log\portscan.log
#
# uncomment output options if you need them, or use snort command line
# output alert_syslog: LOG_AUTH LOG_ALERT
# output log_tcpdump: snort.log
output database: alert, mysql, user=snort dbname=snort host=localhost
# output database: log, postgresql, user=snort dbname=snort
# output database: log, unixodbc, user=snort dbname=snort
# output xml: log, file=/var/log/snortxml
#
include vision17.rules
```

Schritt 8:

Download von Phplot und in Ordner phplot gespeichert. Phplot ist Graphik Library um Tabellen und Grafiken in native PHP zu erstellen. Damit ACID mit MySQL zusammenarbeiten kann, müssen noch ACID Tabellen importiert werden.

Im CommandPrompt können die aktuellen Tabellen angeschaut werden.

```
c\>:mysql\bin>my sql –u snort
mysql> use snort
Database changed
mysql> show tables;
```

Mit folgendem Befehl werden die ACID-Tabellen in der Datenbank erstellt:

```
MySQL-Client
    |     Benutzer  DB
    |         |       | Import to                File
    |         |       |   |                       |
    |         |       |   | ------------------------------------------------
mysql –u snort snort < c:\sntemp\acid\acid\create_acid_tbls_mysql.sql
```

Vorher:	Nachher:
+-----------------+ | Tables_in_snort | +-----------------+ | data | | detail | | encoding | | event | | icmphdr | | iphdr | | opt | | sensor | | tcphdr | | udphdr | +-----------------+ 10 rows in set (0.00 sec)	+-----------------+ | Tables_in_snort | +-----------------+ | acid_ag | | acid_ag_alert | | data | | detail | | encoding | | event | | icmphdr | | iphdr | | opt | | sensor | | tcphdr | | udphdr | +-----------------+ 12 rows in set (0.00 sec)

Schritt 9:

Acid_conf.php

Es müssen einige Änderungen in diesem File gemacht werden. Im Kommentartext sind die Parameter beschrieben und unten sind Eintragungen, die wir gemacht haben.

```
/* Alert DB connection parameters
    - $alert_dbname   : MySQL database name of Snort alert DB
    - $alert_host     : host on which the DB is stored
    - $alert_port     : port on which to access the DB
    - $alert_user     : login to the database with this user
    - $alert_password : password of the DB user
 */

$alert_dbname   = "snort";
$alert_host     = "localhost";
$alert_port     = "3306";
$alert_user     = "snort";
$alert_password = "";
```

Schritt 10:

Damit der Rechner auch die PHP-Seiten darstellen kann die ACID erzeugt, müssen wiederum einige Einstellungen vorgenommen werden. Die Einstellungen sollten nach folgenden Punkten eingestellt werden:

IIS 4.0+ (CGI)
1. Copy the php.ini-dist to your systemroot (the directory where you installed windows), rename it to php.ini, and edit it to fit your needs *(WINNT → System32)*
2. Start the Microsoft Management Console (may appear as 'Internet Services Manager', either in your Windows NT 4.0 Option Pack branch or the Control Panel->Administrative Tools under Windows 2000).
3. Right click on your Web server node (will most probably appear as 'Default Web Server'), and select 'Properties'.
4. Under 'Home Directory', click on the 'Configuration' button. Add a new
5. entry to the Application Mappings; Use the path to php.exe as the Executable, supply .php as the extension, leave 'Method exclusions', blank, and check the Script engine checkbox.
6. Put a .php file under your Web server's document root and check if it works!

Die Datei `php_gd.dll` muss noch ins Verzeichnis von WINNT kopiert werden.

In der Datei `c:\WINNT\php.ini` muss bei `extension=php_gd.dll` der Punktstrich (;) weggenommen werden um es zu aktivieren.

Um zu kontrollieren ob die *.php Seiten im Browser überhaupt angezeigt werden, haben wir eine Testseite in php programmiert:

```
<?
phpinfo();
?>
```

Durch: http://localhost/test.php kann geschaut werden ob die Seite funktioniert. Wenn ja sind die Einstellungen richtig gemacht.

Schritt 11:

In der Datei Inetpub → wwwroot → acid → acid_conf.php müssen die gleichen Erweiterungen gemacht werden wie in Schritt 9. In diesem File müssen zusätzlich noch zwei Pfade gesetzt werden:

```
/* Path to the DB abstraction library */
$DBlib_path = "c:\sntemp\adodb";

/* Path to the PHPlot graphing library */
$ChartLib_path = "c:\sntemp\phplot";
```

Schritt 12:

Bevor Snort gestartet wird, muss noch ein Ornder „Log“ erstellt werden, damit Snort weiss wo er die Log-Files hinschreiben kann. Um Snort nun einfach zu starten, ohne spezielle Kenntnisse von Snort-Befehlen zu haben, wurde ein kleines Batch-File programmiert, welches diese Befehle ausführt:

```
c:
cd c:\snort
snort -c snort1.conf
```



Start Snort.bat

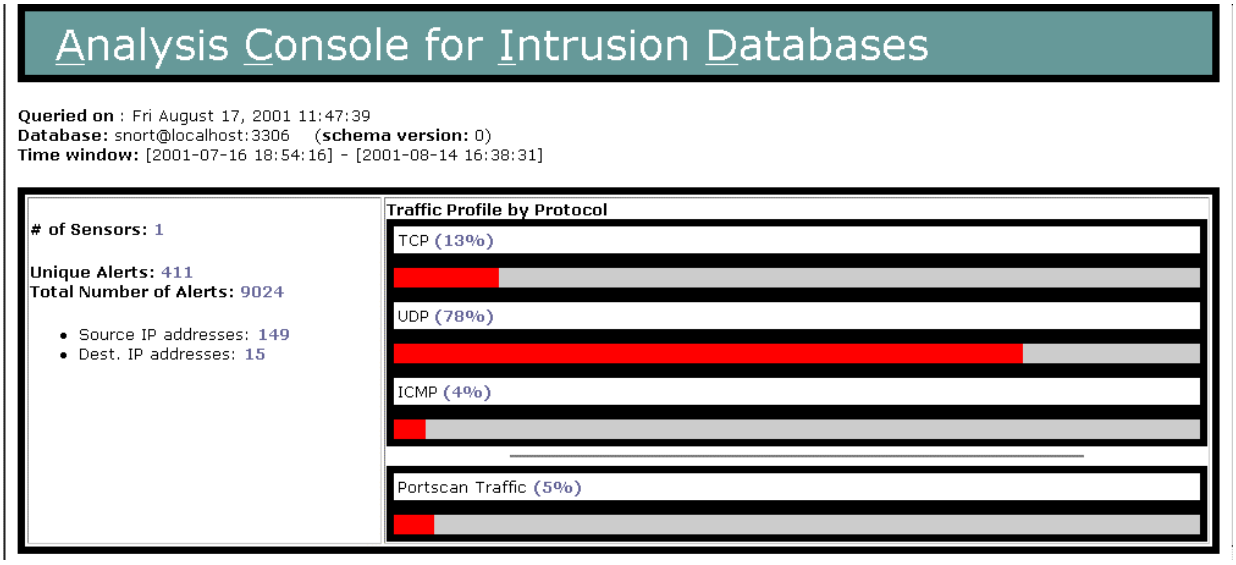
5.2.2.3 ACID (Analysis Console for Intrusion Database)

Ein Punkt, in dem sich Snort konzeptbedingt nicht mit kommerziellen Intrusion Detection Systemen messen kann, ist die Aufbereitung der Alarmmeldungen. Snort konzentriert sich ganz auf die Aufgabe, Netzwerkverkehr zu analysieren und nach entsprechenden Regeln Meldungen zu erstellen. Diese Meldungen kann es in eine Textdatei schreiben, an den syslog-Dienst weiterreichen und mittlerweile auch Datenbanken füttern (MySQL,...). Damit ist die Aufgabe von Snort eigentlich erledigt.

Für eine vernünftige Auswertung braucht man jedoch Statistiken, Querverbindungen und Suchmöglichkeiten. Dafür sind auch für Windows diverse Tools entwickelt worden. Eines davon ist das IDS Center, es soll einem das Arbeiten mit Snort erleichtern, beachten Sie dabei das folgende Kapitel 5.2.2.4. Ein etwas anderes Tool ist das Snort2HTML. Wie der Name schon sagt, bringt das Programm die gesammelten Meldungen von Snort in ein HTML Format damit diese in einem Browser lesbar sind. Wir sind darauf aber nicht näher eingegangen, weil wir ein besseres Tool gefunden haben.

Das Tool, welches wir in der Semesterarbeit eingesetzt haben ist ACID (Analysis Console for Intrusion Databases). Auch ACID ist kostenlos erhältlich. Es wurde von CERT (Computer Emergency Response Team) entwickelt. Es besteht aus einer Reihe von PHP-Seiten, die sich ihre Informationen aus einer Datenbank holen. Wir haben eine MySQL Datenbank aufgesetzt, um dies zu ermöglichen. ACID kann also die Seiten dynamisch erstellen, damit diese den aktuellen Stand anzeigen können. Durch drücken auf dem „Aktualisieren“ Button im Browser, wird jeweils aus der Datenbank gelesen und die Webseite mit den Daten erzeugt.

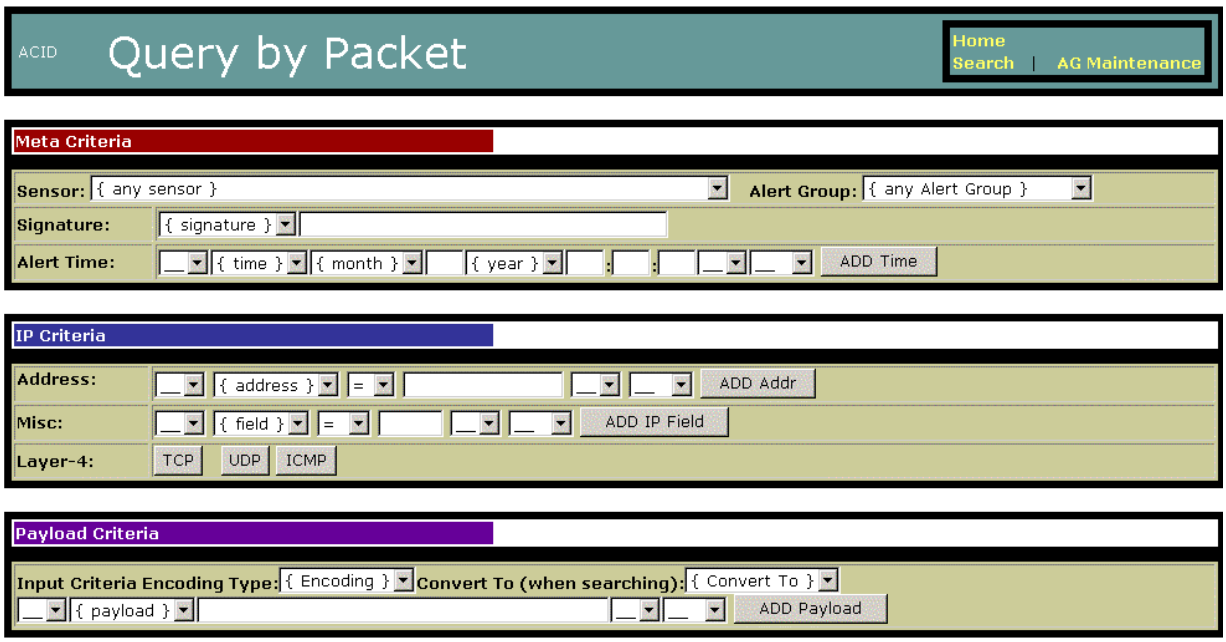
Die Auswertung über alle Alerts ist in der folgenden Grafik dargestellt. Es werden 3 Protokolle genauer untersucht, das TCP, UDP und ICMP. Dazu wird noch eine



grafische Auswertung der Portscans dargestellt. Um die genaueren Daten anzuschauen, kann auf den Prozent-Zahlen geklickt werden und die ganzen Daten werden in tabellarischer Form dargestellt. So können auch die Unique Alerts oder die totale Anzahl Alerts angeschaut werden. Diese werden in der Reihenfolge aufgelistet wie Whitehats [38] die Events bezeichnet hat. Whitehats haben die Alerts jeweils mit

einer fortlaufenden Nummer versehen. Beim Anklicken der IDS+Nummer erscheinen mehrere Informationen über diesen Alert. Mehr davon aber im Kapitel Auswertung und Ergebnisse.

ACID bietet diverse Suchoptionen. So kann man nach Alerts suchen, die in einem bestimmten Zeitraum aufgetreten sind oder nach allen Alerts einer bestimmten IP oder sogar nach Payload Kriterien. Diese Möglichkeiten erleichtern dem Benutzer die Auswertung sehr.



Die grafische Auswertung vermisst man aber schnell. Den ACID stellt alle gesuchten Meldungen in tabellarischer Form dar. Eine kleine Hilfe bietet da das „Graph Alert data“. Es ist aber erst in der Versuchsphase und ist bei weitem noch nicht ausgereift. So kann diese Erweiterung nur einfache Grafiken darstellen, nicht zu vergleichen mit den Möglichkeiten, die man mit Excel hätte. Darum haben wir bei der Auswertung der Alerts auch mit Microsoft Excel gearbeitet.

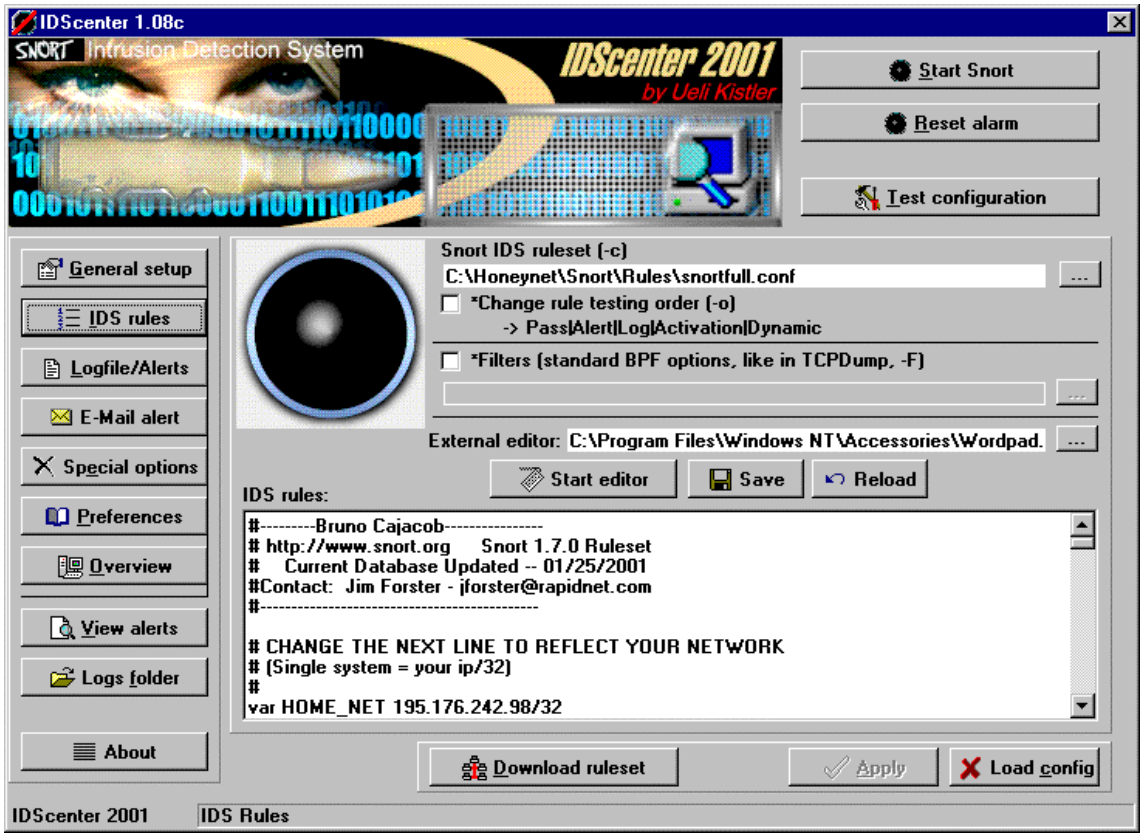
Während der Semesterarbeit gab es auch für ACID eine neue Version. Sie hat einige neue Erweiterungen, wie z.B. können die IP in den Grafiken vollständig angezeigt werden, was bei der alten Version nicht möglich ist. In der alten Version kann die IP nur auf 10 Stellen angezeigt werden. Um das Prinzip der Auswertung von ACID zu erklären genügt aber auch die alte Version. Für einen kommerziellen Einsatz, sollte aber die neue Version installiert werden.

Trotz dieser Auswertungstools weist Snort in manchen Bereichen noch Defizite auf. So kann es bisher ausschliesslich IP-Pakete analysieren – Net-BEUI oder IPX ignoriert der Wächter völlig. Ausserdem versieht Snort seine Alarmmeldungen nicht mit Prioritäten, die die Auswertung deutlich erleichtern würde.

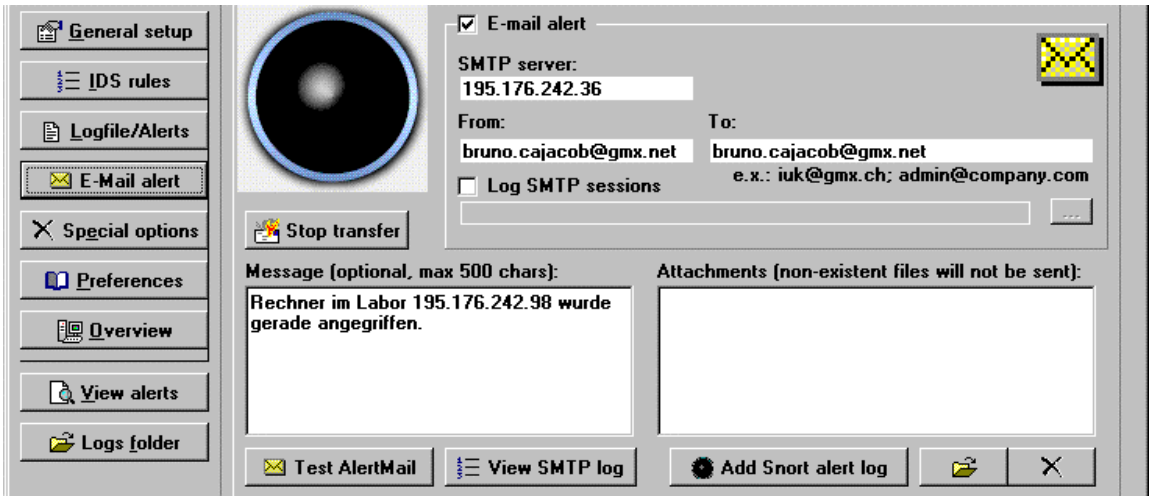
5.2.2.4 IDS Center

Auch der IDS Center erleichtert das Arbeiten mit Snort sehr. Es wurde von Ueli Kistler entwickelt. Man hat sehr viele Möglichkeiten wie: E-Mail Alerts, Speichern der Daten auf ein SysLog-Server und viele andere Optionen. Wir haben in der Anfangsphase sehr viel mit dem IDS Center gearbeitet. Auch hatten wir guten Kontakt zum Entwickler, der uns sehr viele gute Tips geben konnte. Vielen Dank an Ueli Kistler. Wir haben uns aber im Verlauf der Arbeit uns für die Lösung mit Datenbank und ACID entschieden.

- IDS Rules



- E-mail alert



5.2.3 Snort auf Linux

Eigentlich ist Snort für Linux realisiert worden. Wir haben uns aus dem Grund für die Windows Variante entschieden, weil noch eine andere Gruppe von Studenten sich mit Snort befasst.

5.2.4 The Deception Toolkit (DTK)

Das Tool “The Deception Toolkit” wird von Fred Cohen & Associates [39] allen Interessierten zur Verfügung gestellt. Das Ziel des Programmes ist es, Informationen über den Angreifer herauszufinden und somit dem Angegriffenen bei der Abwehr des Angriffes zu unterstützen. Dabei hat es folgende Schwerpunkte:

- Erhöht den Arbeitsaufwand des Angreifers, um in ein System einzubrechen, da das Programm unter anderem gefälschte Passwortlisten erzeugt
- Aufzeichnung aller Bewegungen des Angreifers, um Gegenmassnahmen einzuleiten, bevor der Angreifer eine Sicherheitslücke gefunden hat
- Vortäuschen von Systemdiensten
- Gemeinsam von den Techniken der Angreifern lernen, um sich so besser schützen zu können

Das besondere Merkmal bei Rechner mit DTK ist der Port 365. Wenn ein Rechner bei einem Verbindungsaufbau auf den Port 365 reagiert, heisst dies, dass es ein Täuschungsversuch des DTK sein könnte. Eventuell veranlasst das den Angreifer, sich einen anderen Host zu suchen und somit ist das Ziel des Verteidigers schon erreicht.

Bemerkung: Aus Zeitgründen gehen wir nicht weiter auf dieses Programm ein und wir werden es auch nicht in unserem Aufbau einsetzen. Wir mussten uns dazu entscheiden, da wir nur eine beschränkte Anzahl IP-Adressen zur Verfügung hatten. Darum gehen wir bei diesem Programm nicht mehr ins Detail, obwohl es sich um eine sehr interessante Anwendung eines Honeypots handelt.

5.3 Kommerzielle Tools

5.3.1 Cybercop Sting

5.3.1.1 Kurzbeschreibung

CyberCop Sting ist ein Lockmittelsystem, das ein virtuelles Netzwerk auf einem Rechner simuliert. Diese Technologie lenkt Hacker von kritischen Systemen ab. Der Server protokolliert den Intrusionsverkehr, um dessen Ursprung festzustellen, und sammelt Beweise gegen Angreifer. Mit Hilfe verschiedener Arten von stillen Alarmen können Eindringlinge mit CyberCop Sting entdeckt werden, bevor sie Schaden anrichten können.

- Eindringlinge können eingegrenzt und auf frischer Tat ertappt werden
- Hacker können von kritischen Systemen abgelenkt und auf vorgetäuschte Systeme aufmerksam gemacht werden
- Dieses erweiterte Tool verstärkt die Netzwerksicherheit

Merkmale:

- Vorgetäuschte Systeme für Hacker
  - Windows NT 4.0 Server
  - Solaris 2.6 Server
  - Cisco IOS 11.2 Router
- Verfolgt Eindringlinge
- Sammelt Beweise

Systemanforderungen:

- Windows NT 4.0, min Service Pack 4.0
- Pentium II Processor
- 128 MB RAM Hauptspeicher
- 150 MB verfügbarer Festplattenspeicher
- Netzwerkkarten

Version:

- 1.0.0 Evaluation für Windows NT

Preis:

- Nicht einzeln erhältlich. Nur im Gesamtpaket CyberCop Scanner.

5.3.1.2 Bemerkungen:

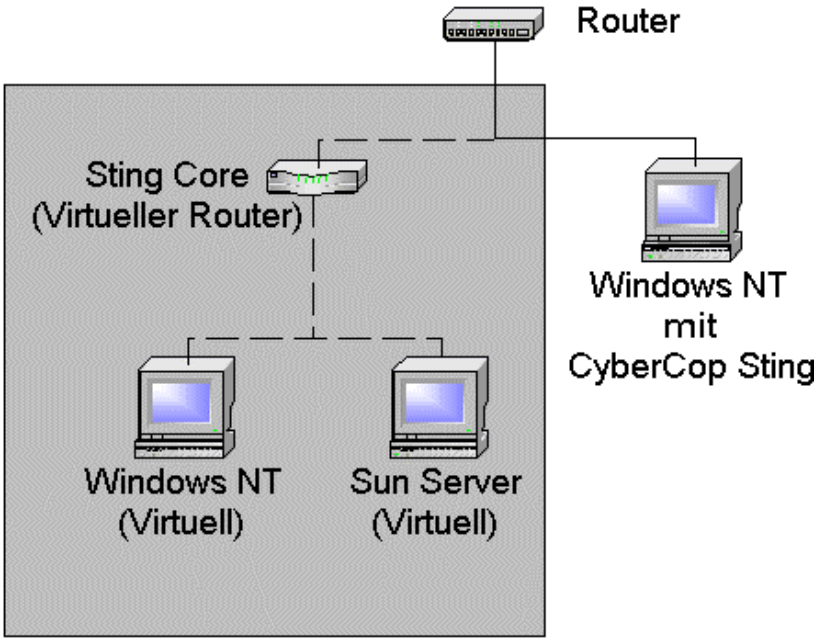
Am Anfang unserer Semesterarbeit war eine Trial Version noch frei verfügbar auf der Homepage von PGP Security [41]. Doch dies änderte sich unterdessen. Die Software ist nicht mehr verfügbar. Laut Aussagen von Thomas Stutz (Geschäftsleiter Omicron Electronics GmbH, Partner PGP Security) wird die Software nicht mehr weiterentwickelt.

5.3.1.3 Installation

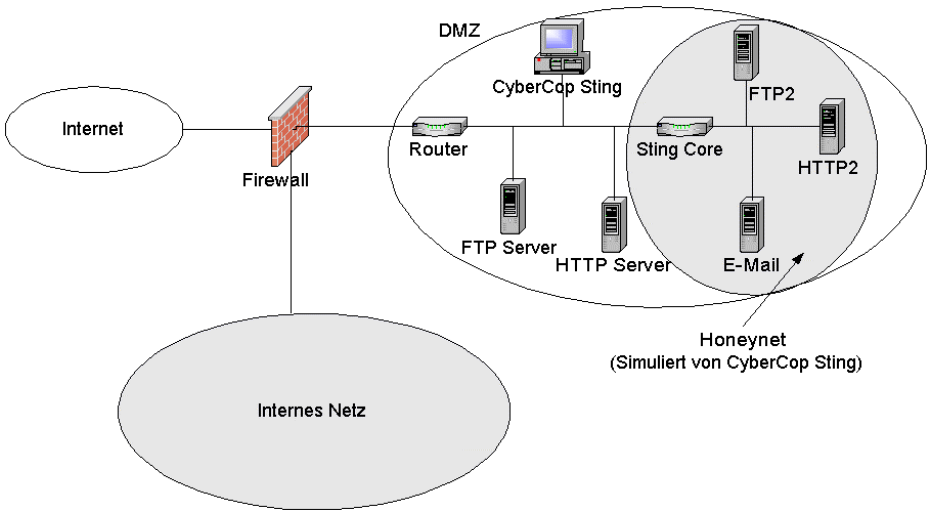
Die Installation ist sehr einfach, da die notwendigen Schritte mit einem Assistenten erleichtert werden. Die einzige Eingabe des Benutzers betrifft den Installationsort, wo dieses Programm installiert werden soll. Anschliessend muss der Rechner neu gestartet werden.

5.3.1.4 Konfiguration

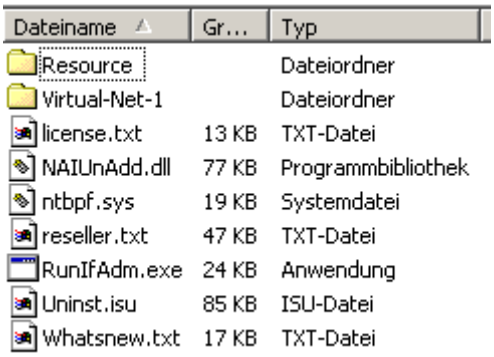
CyberCop Sting kann drei verschiedene Systeme vortäuschen. Das wären Windows NT 4.0 Server, Solaris 2.6 Server und Cisco IOS 11.2 Router. Im folgenden Bild kann man den Grundaufbau erkennen:



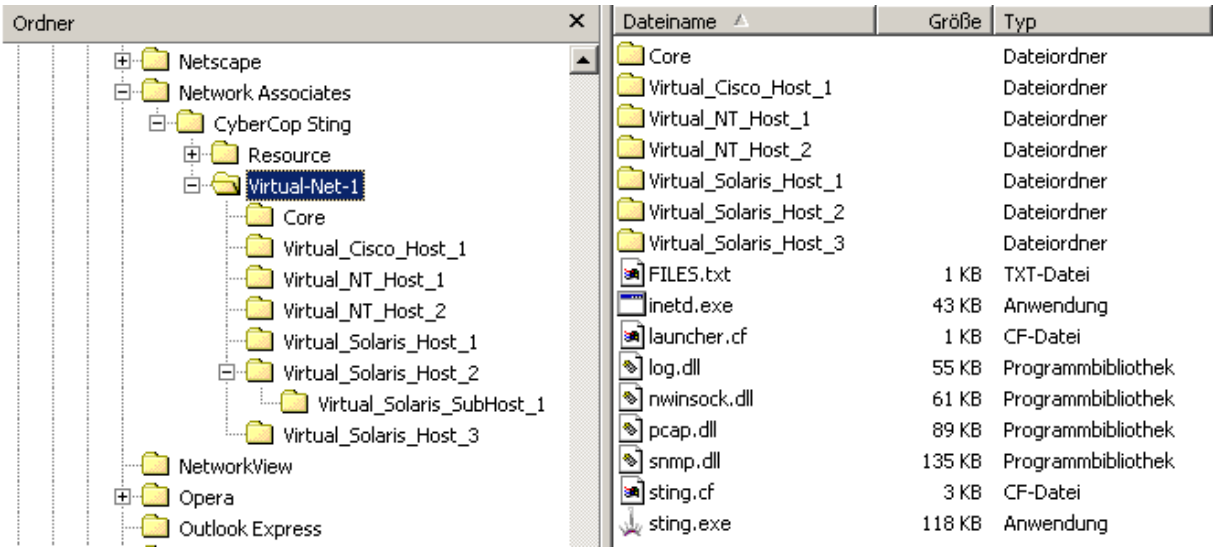
Alle Komponenten im grau markierten Bereich werden von der Software simuliert. Wie dieser Aufbau genau aussehen soll, bestimmt der Benutzer. Die einzige Komponente die im virtuellen Netz vorkommen muss, ist der Sting Core Router. Wie und wo dieses virtuelle Netz eingesetzt werden soll ist auch nicht vorgegeben und kann so vom Anwender frei gewählt werden. Hier ein solcher Aufbau eines ganzen Netzes:



Schauen wir uns die Konfiguration von CyberCop Sting etwas besser an. Erst mal zeigen wir, wie das Verzeichnis der Software nach der Installation aussieht:

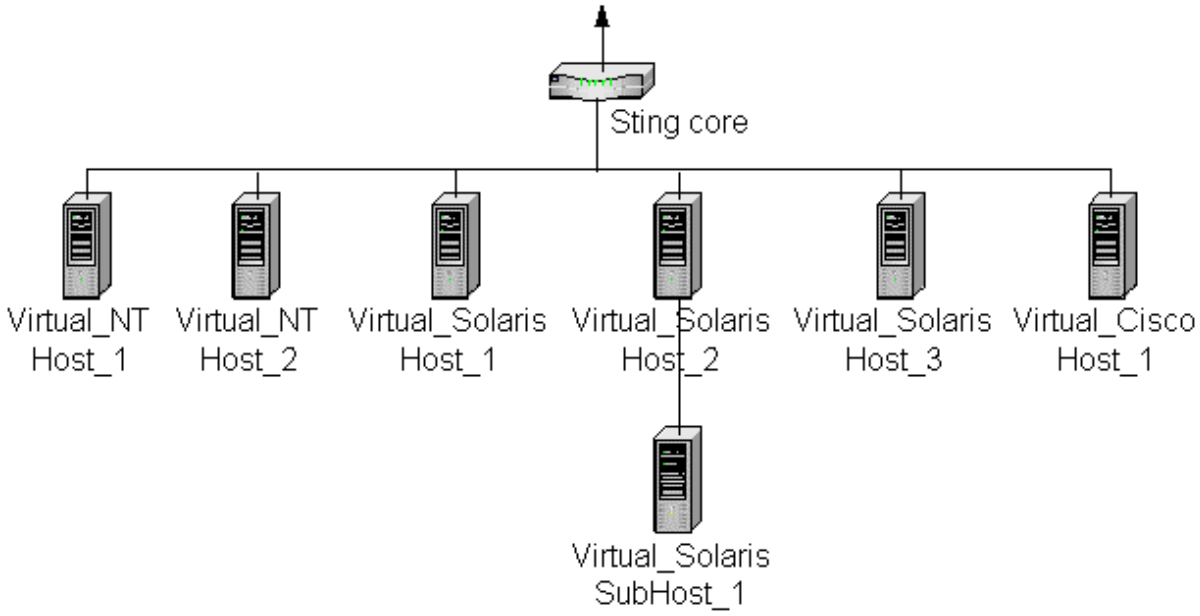


Besonders dem Ordner Virtual-Net-1 ist das Augenmerk gerichtet. Denn in diesem ist eine vorkonfiguriertes Netz eingerichtet. Wir schauen uns dieses genauer an:

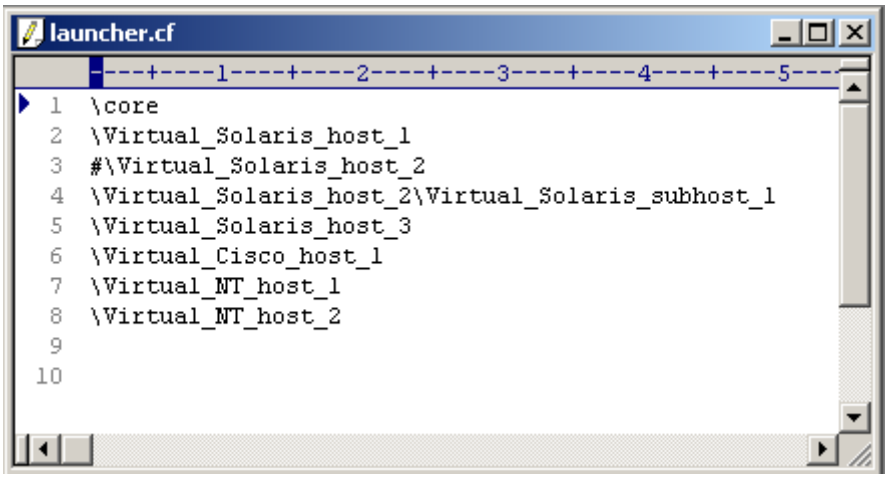


Nun können wir verschiedene Ordner sehen. Jeder dieser Ordner symbolisiert einen Rechner oder Router. Das heisst in diesem Fall besteht das virtuelle Netz aus zwei Routern (Core und Virtual_Cisco_Host_1), zwei Windows NT Server und aus vier Solaris Server, wobei *Virtual_Solaris_SubHost_1* virtuell an den Server *Virtual_Solaris_Host_2* angeschlossen ist. Alle anderen sind direkt am *Core* Router angeschlossen.

Hier das Schema dazu:

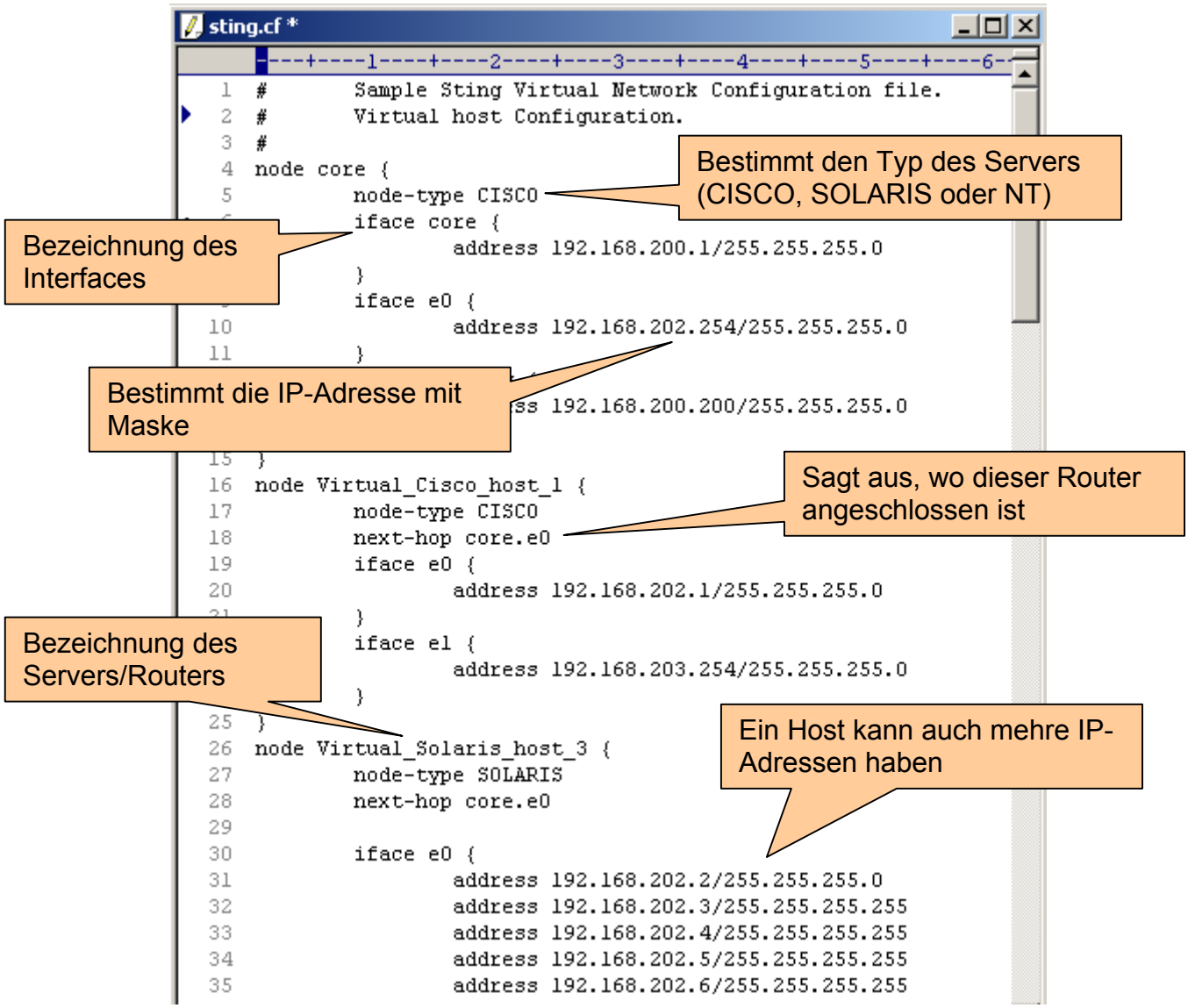


Die zwei wichtigsten Konfigurationsdateien um ein ganzes Netz mit CyberCop Sting aufzubauen. sind die Dateien *sting.cf* und *launcher.cf*. *launcher.cf* bestimmt welche virtuellen Server oder Router gestartet werden, welche sich in den Unterordner befinden.



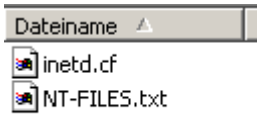
In diesem Beispiel werden alle Router (Core und Virtual_Cisco_host_1) und Server gestartet, ausser dem Virtual_Solaris_host_2.

In der Datei *sting.cf* wird bestimmt, welcher Server/Router wo angeschlossen ist und welche IP-Adressen sie besitzen:

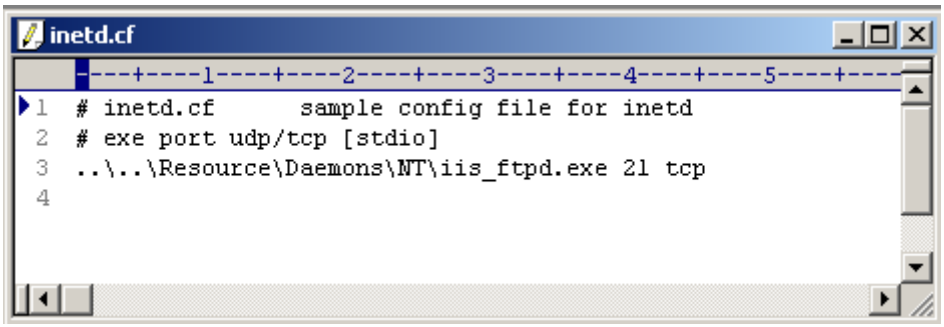



```
1  #       Sample Sting Virtual Network Configuration file.
2  #       Virtual host Configuration.
3  #
4  node core {
5          node-type CISCO
           iface core {
                   address 192.168.200.1/255.255.255.0
           }
           iface e0 {
10                 address 192.168.202.254/255.255.255.0
11         }
                   ss 192.168.200.200/255.255.255.0
15 }
16 node Virtual_Cisco_host_1 {
17         node-type CISCO
18         next-hop core.e0
19         iface e0 {
20                 address 192.168.202.1/255.255.255.0
           }
           iface e1 {
                   address 192.168.203.254/255.255.255.0
           }
25 }
26 node Virtual_Solaris_host_3 {
27         node-type SOLARIS
28         next-hop core.e0
29
30         iface e0 {
31                 address 192.168.202.2/255.255.255.0
32                 address 192.168.202.3/255.255.255.255
33                 address 192.168.202.4/255.255.255.255
34                 address 192.168.202.5/255.255.255.255
35                 address 192.168.202.6/255.255.255.255
```

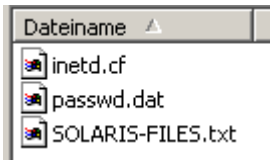
In einem Windows NT Ordner (z.B. Virtual_NT_Host_1) befinden sich diese Dateien:



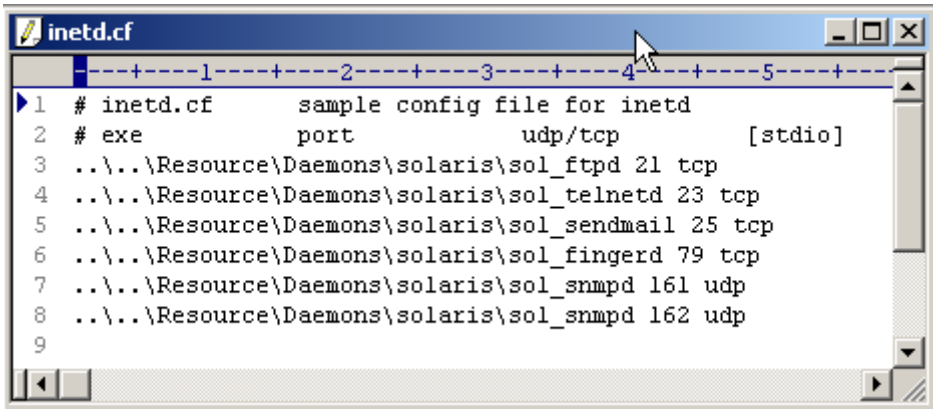
Die Datei *NT-FILES.txt* ist nicht von Bedeutung. Dagegen wird in der Datei *inetd.cf* die Dienste konfiguriert, auf welche der Virtuelle NT-Server ansprechen soll:



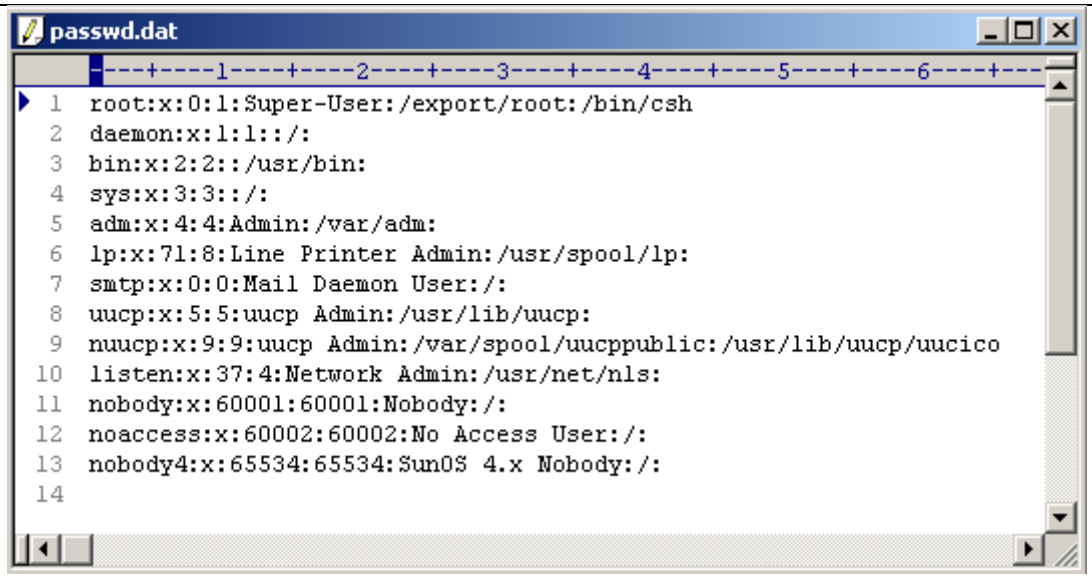
Wir können erkennen, dass hier nur der FTP Service gestartet wird. Wobei man sagen muss, dass CyberCop Sting für einen Microsoft NT Server nur diesen FTP Dienst simulieren kann.
Auch in einem Solaris Ordner sind diese Dateien vorhanden:



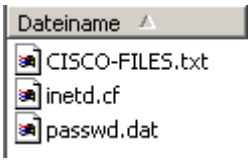
Auch hier gibt es eine *inetd.cf* Datei:

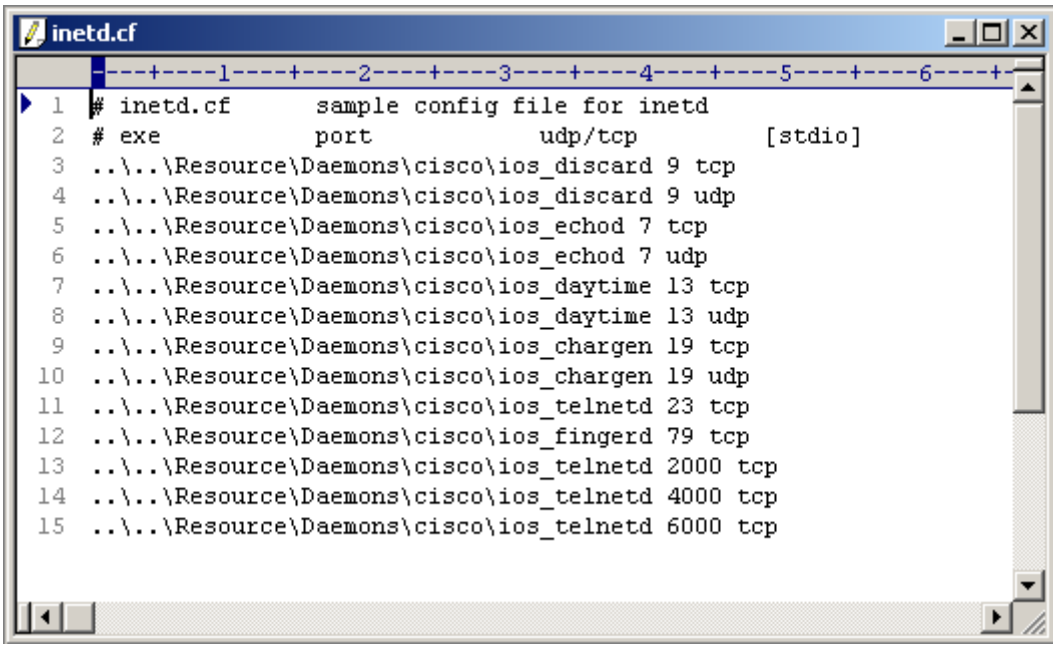


Bei Solaris können mehrere Dienste simuliert werden. Ausserdem ist im Solaris Ordner eine Datei mit dem Namen *passwd.dat* vorhanden:

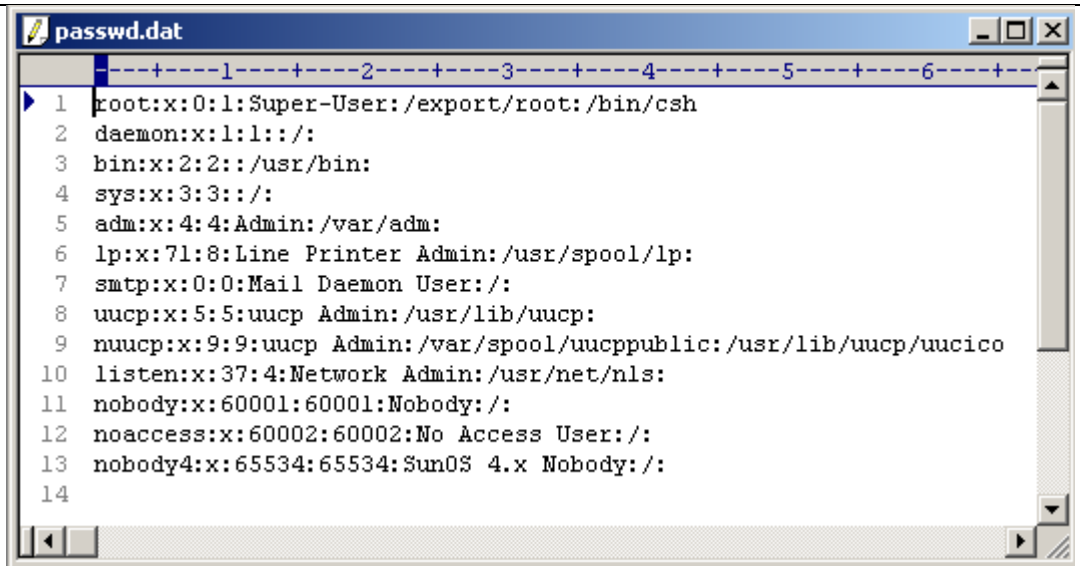


In dieser Datei sind die virtuellen Benützer dieser Solaris Servers eingetragen, eben wie das in der Wirklichkeit auch ist.
Und nun zum Cisco Ordner (z.B. Core oder Virtual_Cisco_Host_1). Hier sind die gleichen Dateien vorhanden wie beim Solaris Ordner:

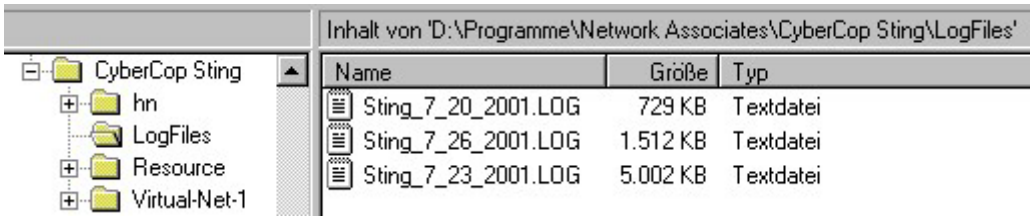




Wie ersichtlich kann CyberCop Sting auch für die virtuellen Cisco Router verschiedene Dienste simulieren. Ausserdem ist auch eine Passwort-Datei vorhanden:



CyberCop Sting schreib für jeden Tag eine Datei, an welchem Angriffe aufgezeichnet werden, auf die Festplatte:



Dabei wird für jeden Angriff folgende Daten in der Datei gespeichert:

Bezeichnung	**Beispiel**
Zeit	14:14:46
DST	195.176.242.218
SRC	192.168.200.200
DST PORT	0
SRC PORT	0
CAT	EVENTLOG_INFORMATION_TYPE
TYPE	[ICMP] Packet Sent
ITEM	ICMP
THREAT	none
DATA	type 0, seq #2048

5.3.2 Specter

5.3.2.1 Kurzbeschreibung

SPECTER [dt: Erscheinung] simuliert einen kompletten Rechner mit all den Diensten des jeweiligen Betriebsystems. Mit dem Ziel den Angreifer anzulocken, um mehr über diesen herauszufinden. Dazu kann SPECTER, noch während des Angriffes, den Angreifer selbst ausspionieren.

Merkmale:

- Täuscht Betriebsystem und Dienste vor
- Zeichnet sämtliche Schritte des Angreifers auf (Für spätere Beweisführung)
- Kann den Angreifer identifizieren und ausforschen (Whois, Port Scan, ...)
- Lockt mit zusätzlichen Programmen, wie z.B. BO2K oder Sub-7
- Alarmierung kann auf diverse Arten erfolgen (Mail, Syslog, ...)

Systemanforderungen:

- Pentium II 450 MHz
- 128 MB RAM Hauptspeicher
- Windows NT 4.0 SP6A oder Windows 2000 SP1
- Bildschirmauflösung von 1024x768

Version:

- 5.52

Preis:

- Erstlizenz: US$ 899.-
- Jede weitere Lizenz: US$ 399.-

5.3.2.2 Bemerkungen:

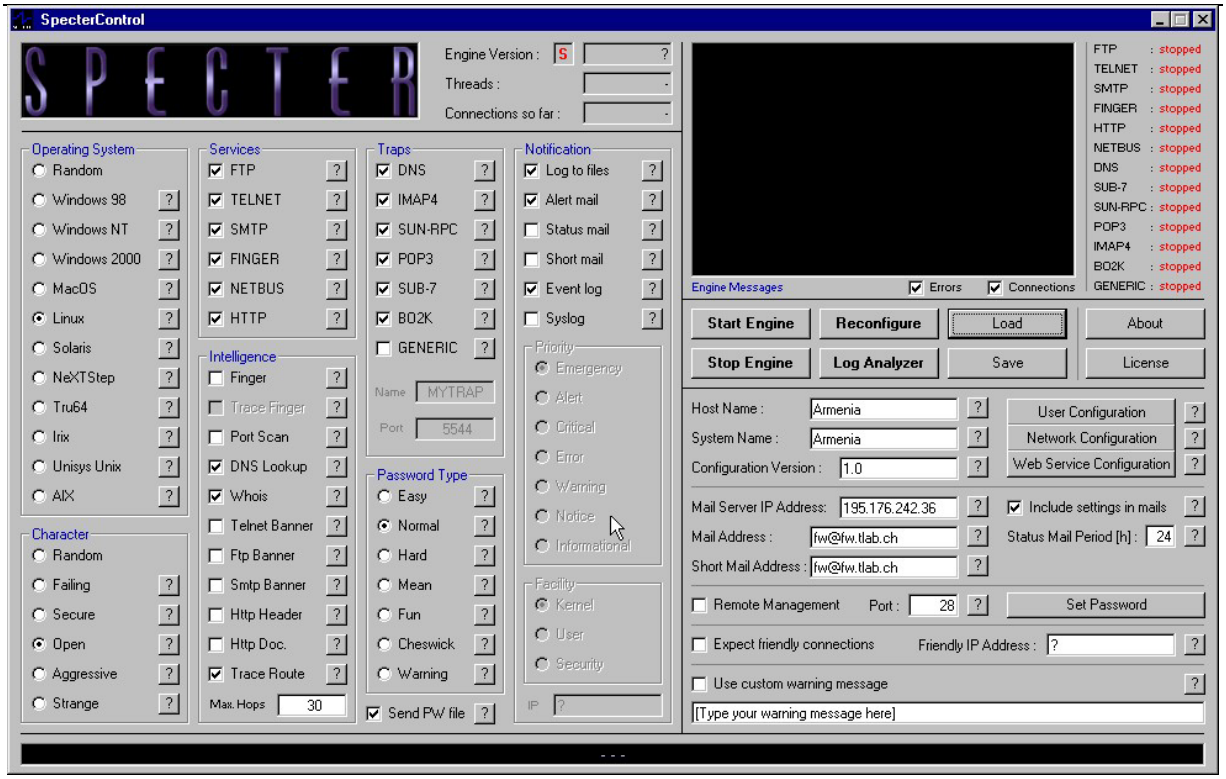
Das Programm SPECTER wird von der Schweizer Firma NETSEC entwickelt und vermarktet. An dieser Stelle möchten wir der Firma NETSEC danken, da sie uns eine zeitbegrenzte Vollversion gegeben hat. Der Dank geht auch an Martin Lechmann, der dies überhaupt ermöglicht hat.

5.3.2.3 Installation

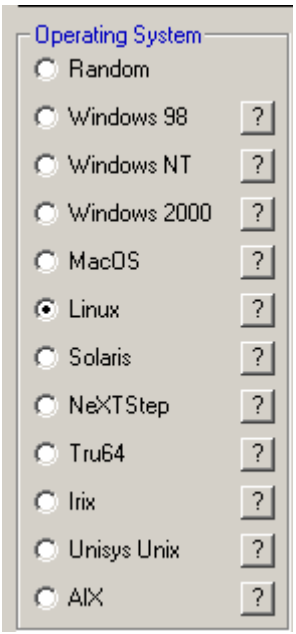
Die Installation ist sehr einfach durchzuführen, nur der Installationspfad muss angegeben werden, den Rest erledigt der Installationsassistent.

5.3.2.4 Konfiguration

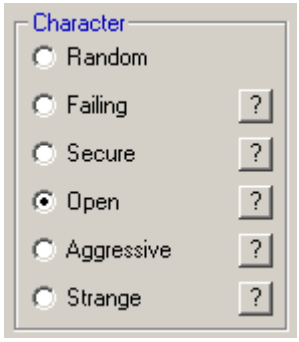
Alle möglichen Einstellungen können mit dem Programm SpecterControl gemacht werden. Da das GUI sehr übersichtlich und logisch aufgebaut ist, bereitet die Konfiguration von Specter keinen grossen Aufwand. Hier sehen sie das Hauptfenster, in dem alle wichtigen Einstellungen getätigt werden können:



Wir werden nun kurz die einzelnen Bereiche des GUI's beschreiben, obwohl man zu jedem einzelnen Eintrag eine Erklärung findet und falls diese nicht genügt, kann mittels F1 das Hilfesystem aufgerufen werden. In diesem findet man genaue Beschreibungen zu jedem Thema rund um Specter.



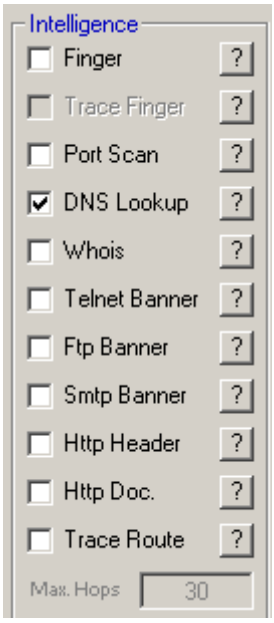
Im linken Teil der Oberfläche kann man zwischen elf verschiedenen Betriebsystemen auswählen, welches Specter simulieren soll. Es gibt sogar eine Option Random, d.h. das Programm wählt zufällig das jeweilige Betriebsystem aus.



Im Bereich Character wählt man aus, wie gut das simulierte System aufgebaut ist. Hier gibt es fünf verschiedene Sicherheitsstufen. Von einem fehlerhaften bis zu einem perfekt geschützten Rechner.

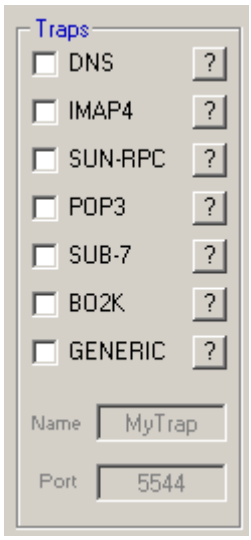


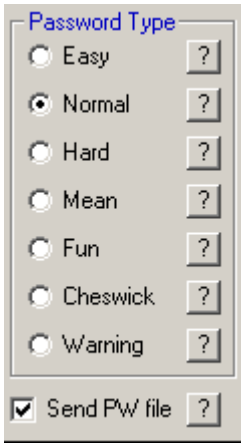
Des weiteren kann man bestimmen, welche Dienste auf dem System laufen sollen. Dabei sind sowohl die bekannten (FTP, HTTP, ...), aber auch die für den Hobby-Hacker interessante Dienste (NETBUS) verfügbar. All diese simulierten Anwendungen verhalten sich genau so, wie die originalen, nur mit dem Detail, dass der Angreifer keinen Schaden anrichten kann.



Damit man aber auch etwas mehr über den Angreifer herausfinden kann, sind im Specter diverse Tools eingebaut. Die Auswahl geht von einem simplen DNS Lookup, bis zu einem vollständigen Portscan des Angreifers. Diese Tools erlauben dem Benützer, interessante Informationen über den Angreifer herauszufinden. Wir können hier ein Beispiel nennen. Mit der Option Telnet Banner sagt man Specter, er solle bei einem Angriff versuchen, Informationen mittels Telnet über den Angreifer herauszufinden. Wenn nun der Angreifer den Dienst Telnet bei sich eingerichtet hat, meldet sich Specter dort an und empfängt die Antworten des Telnet Servers. Meistens sind in diesem Informationen der Name des Betriebsystem, des Rechners, sowie die Versionsnummer enthalten.

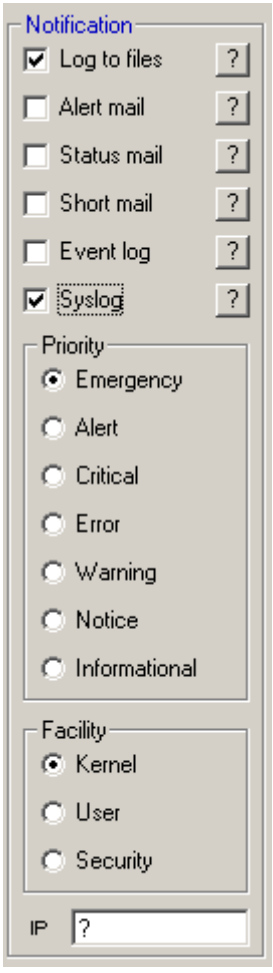
Nun kommen wir zu den Traps. Hiermit erhält man die Möglichkeit, diverse Ports zu überwachen, welche normalerweise speziellen Programmen zugeordnet sind, wie z.B. das Sub-Seven. Dies ist ein bekannter Trojaner, welcher sich bei einem ahnungslosen Benützer einnistet und dadurch erhält der Angreifer die volle Kontrolle über diesen Rechner. Specter „hört" aber nur diese Ports ab, antwortet aber nicht auf die Anfragen.



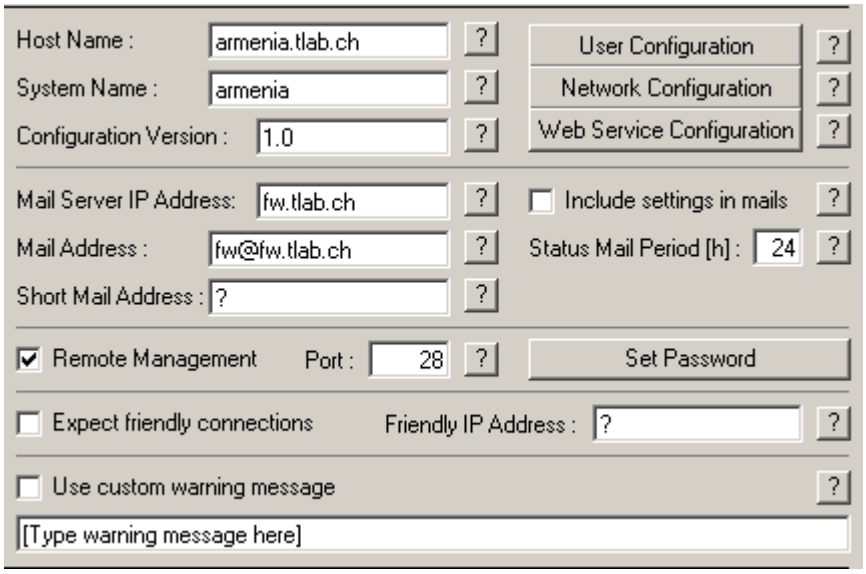


Im Bereich Password Type kann man über die virtuellen Passwörter diverse Einstellungen machen. Dabei wird bei aktivierter Option „Send PW file", dem Angreifer eine Passwortliste übermittelt, natürlich nur sobald dieser die Liste „anfordert".

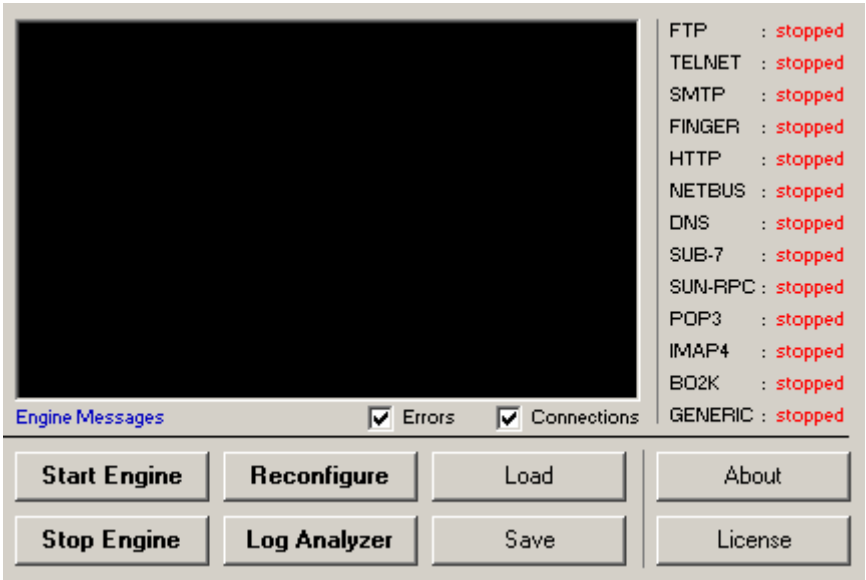
Damit bei einem Angriff alle Daten gespeichert werden, oder dass der Administrator benachrichtig wird, gibt es in Specter den Bereich Notification. Hier kann, differenziert, die gewünschte Alarmierungsoption gewählt werden. Speziell zu erwähnen ist die Option Syslog. Damit können sämtliche Logdaten einem Syslog-Server übermittelt werden, um eine bessere Auswertung/Überwachung machen zu können.



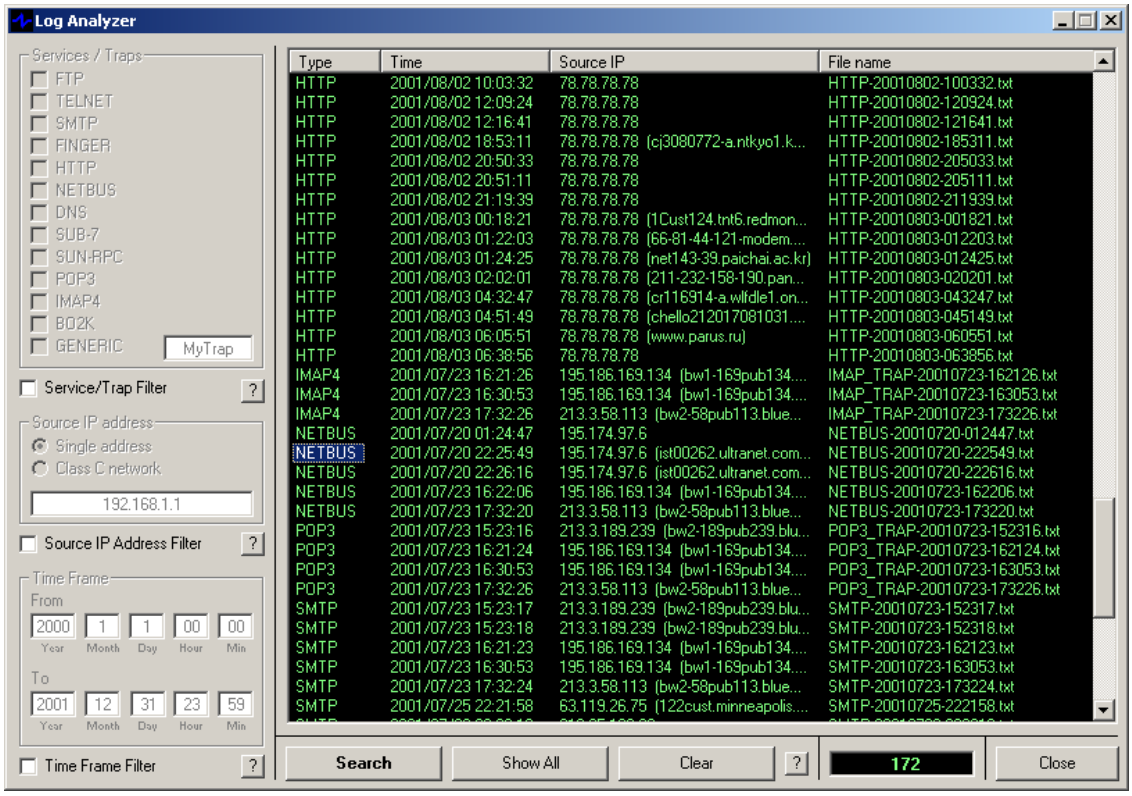
Im folgenden Bild können wir die allgemeinen Einstellungen für Specter erkennen. Hier werden verschiedene Einstellungen über den Rechner selbst, welchen Mail-Server benutzt werden kann, welche „Angreifer“ nicht beachtet werden sollen und was für eine Meldung dem Angreifer automatisch geschickt werden soll. Ausserdem lässt sich Specter fernsteuern, um das Management zu vereinfachen.



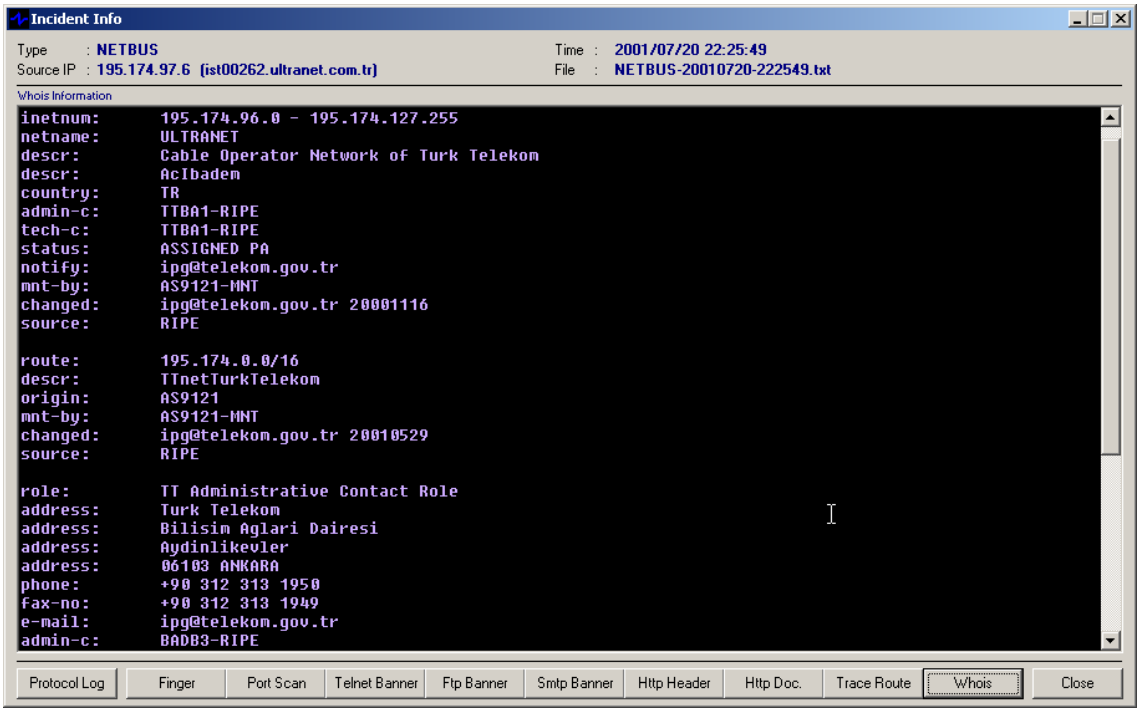
Nun kommen wir zum letzten Teil der Bedieneroberfläche.
Hier kann Specter gestartet und überwacht werden. Im schwarzen Bereich werden die Angriffe kurz aufgeschrieben, welche dann mit dem Log Analyzer genau überprüft werden können.



Im Log Analyzer können die gespeicherten Daten schnell gefunden und analysiert werden. Dafür gibt es diverse Filter, um die Suche zu vereinfachen. Hier das Hauptfenster, nachdem der Button „Show all“ gedrückt wurde:

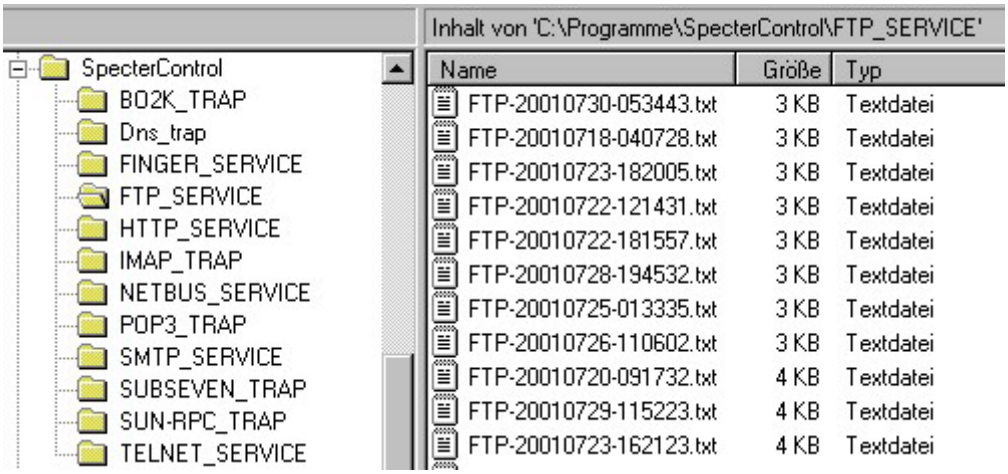


Wollen wir nun mehr über diesen Angriff (Versuch mit Netbus die Kontrolle über unser System zu erhalten) erhalten, genügt ein Doppelklick und schon wird das folgende Fenster geöffnet:



Je nach Einstellung können hier nun die Informationen über den Angreifer abgefragt werden. Im Bild oben sehen Sie, was Specter mit „Whois“ herausgefunden hat.

Specter macht für jede Falle oder Dienst einen Ordner. Für jeden Angriff wird eine eigene Datei gemacht, dessen Name aus dem Dienst, dem Datum und der Zeit des Angriffes besteht:



In dieser Datei sind folgende Daten:

- Konfiguration Specter
- Daten von Whois, Portscan, Banner...
- Loginname und Passwort (falls vom Angreifer angegeben)
- Genaue Zeitangabe

5.3.3 ManTrap

5.3.3.1 Kurzbeschreibung

Die Software versucht den Angreifer in ein simuliertes Honeypot zu locken, um den Hacker auf Schritt und Tritt zu verfolgen.

Merkmale:

- Täuscht Betriebsystem und Dienste vor
- Zeichnet sämtliche Schritte des Angreifers auf (Für spätere Beweisführung)
- Kann den Angreifer identifizieren
- Alarmierung kann auf diverse Arten erfolgen (Mail, Syslog, ...)

Systemanforderungen:

ManTrap Host:

- Sun Ultra 5 oder Sun HCL certified Intel Pentium II [43]
- Sun Solaris 2.6 oder 7
- 160 MB Arbeitsspeicher
- 2 GB Harddisk

ManTrap Administration Console

- Solaris 2.6 /7/8 oder Microsoft Windows 95/98/NT/2000
- Java 2 Runtime Environment v.1.3
- Sun HCL certified network interface card

5.3.3.2 Bemerkungen:

Um ManTrap besser kennenzulernen, versuchten wir Kontakt mit der Firma Recourse aufzunehmen. Dies war uns dann auch geglückt und wir waren zuversichtlich eine Demoversion von ManTrap zu bekommen. Wir waren schon so weit, dass Richard Galphin (Director, International Operations Recourse Technologies, Inc.) uns fragte, was für Hardware wir denn einsetzten werden, damit sie die Software darauf anpassen können. Doch dann war es leider plötzlich zu Ende. Recourse antwortete einfach nicht mehr auf unsere Mails. Darum haben wir ManTrap nicht in unserem Honeynet einsetzen können.

Ausserdem haben wir auf der Homepage von SecuriTeam einen interessanten Beitrag entdeckt:

Nasty security holes in ManTrap
ManTrap's intent is to set up a honey pot as a 'cage' which can be filled with fake information running on a system where the intruder can be lured and studied, and which is also supposed to prevent erasing of logs, etc. This 'honey-pot' method is recommended by some security experts, but this specific implementation of it is poor. Instead of running as a VMware like virtual machine, ManTrap runs as a chroot like environment with kernel patches, and thus it can easily be identified and subverted. In case an intruder breaks in and finds out that it's a cage, s/he will most likely just trash it and move on instead of fulfilling the purpose of staying long enough to be studied / identified / traced.

Determining that the machine is a honey-pot:
ManTrap hides processes from the caged intruder (or at least tries to), by hiding them from the /proc in the cage (resembles what many LKM Trojans do), which sounds fine.

However, the cage still runs the **same** kernel as the real box, and has access to all kernel memory, etc. Using a syscall (in this case kill()), we can get this information from the kernel instead of /proc. What we do is to simply send a signal (SIGCONT in this case) which hopefully shouldn't affect anything, to PID 1 to 65535 as a non-root user, and compare the results with /proc.

kill() gives EPERM and /proc/<PID> exists -> Fine
kill() gives EPERM and /proc/<PID> does not exist -> Not fine!

This can also possibly be used to detect LKM Trojans and the like. It might give a false alarm though; as some kernel patches designed to hide other user's processes might give the same result. Nevertheless, together with the other telltale signs of ManTrap it gives a very good fingerprint.

Denial-of-Service:
This looks like a result from the /proc filtering/emulation mentioned above. /proc/.. doesn't show up in any syscalls wanting to get the directory listing (such as getdents()). Also, (cd /proc/self; cd cwd; pwd) gives an error. Another interesting note is that the whole box can be made to lock up and disconnect all users for a couple of minutes by doing:

```
# cd /proc && cd self && cd cwd
# pwd <causes error response>
# cd ../../../../../
# cd proc
# cd self <should receive error response>
# ls, pwd, etc, <BOOM!>
```

More indications the machine is a honey-pot:
Since / isn't the real root of the file system, but rather /usr/rti/cage/root, the inode number is very off, usually in the 100000-200000 range instead of the normal 2. This can be checked by simply doing `ls -id /`.

Reading files:
If the intruder got root in the cage, it's very possible to read/write directly to/from /dev/mem, the raw disk devices, etc. The crash(1M) utility can be used to examine /dev/mem and get the real process listing etc, which includes all ManTrap's processes (and, yes, they can be killed...).

More serious damage can be caused using the raw disk devices, such as /dev/rdsk/c0t0d0s0. **Any** data on the system can be read/modified by the intruder, and this can be used to wipe logs and such. An utility such as fsdb(1M) can be used to view the directory listings.

Den ganzen Bericht finden Sie unter:

http://www.securiteam.com/securitynews/6H00S1P0AA.html

6 Core Installation

Ein Hauptpunkt in der Semesteraufgabenstellung ist, die Sicherheit des Betriebsystems auf ein Maximum zu erhöhen. Das heisst, es wird versucht das Operation System so zu optimieren und zu beschränken, dass es einem möglichen Angreifer sehr schwer fallen wird, in das System einzudringen.
Damit man dieses Ziel einfacher erreichen kann, werden von vielen verschiedenen Anbietern Anleitungen angeboten, welche Schritt für Schritt dem Benützer zeigen, wie das „Hardening“ gemacht wird. Wir werden hier an dieser Stelle nur auf ein paar Anleitungen verweisen, da es sonst den Umfang dieser Arbeit sprengen würde.

6.1 Microsoft Windows

Von der Firma Compass Security Network Computing AG [46] in Rapperswil gibt es ein übersichtliches Dokument:

http://www.csnc.ch/downloads/docs/hardening/WindowsNTIISHardening_CSNC.pdf

Ein weiteres ist auf der Homepage von SystemExperts [48] zu finden:

http://www.systemexperts.com/tutors/HardenW2K101.pdf

Und auch Lance Spitzner vom Projekt Honeynet hat ein Dokument geschrieben:

http://www.enteract.com/~lspitz/nt.html

6.2 Sun Solaris

Firma Compass Security Network Computing AG bietet auch für Solaris eine solche Anleitung an:

http://www.csnc.ch/downloads/docs/hardening/SolarisHardening_CSNC.pdf

Auf der Homepage von SecurityFocus [47] ist ein Dokument verfügbar:

http://www.securityfocus.com/frames/?focus=sun&content=/focus/sun/articles/harden1.html

Und das letzte Beispiel für die zahlreichen Anleitungen kommt von SANS (System Administration, Networking, and Security) Institute [49]. Auf dieser Homepage finden Sie zahlreiche Dokumente, welche jedoch kostenpflichtig sind:

http://www.sans.org/newlook/resources/hard_solaris.htm

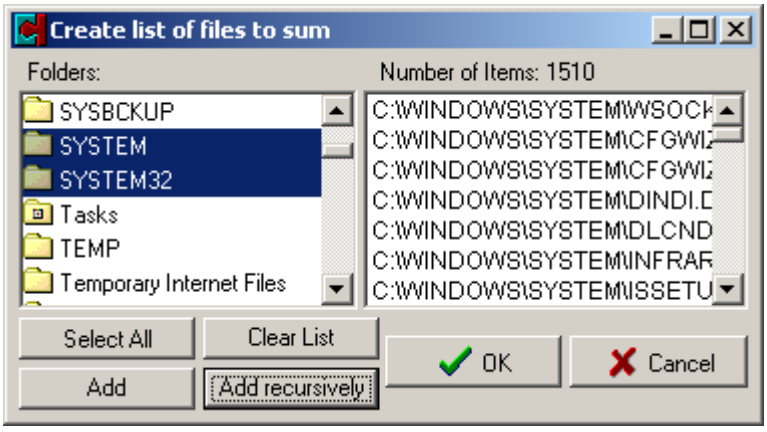
6.3 Integrität der Systeme

Nehmen wir an, es würde einem Hacker gelingen in unsere Rechner eindringen. Eventuell würde er dann Prozesse „modifizieren“, damit diese zusätzlich versteckten Funktionen ausführen würden, oder sonst irgendwelche Taten am System vornehmen. Um diese im Nachhinein erkennen zu können, mussten wir eine Lösung finden. Wir stiessen dabei auf die Software von Tripwire [50]. Doch leider ist diese nicht gratis und auch KPMG hatte keine Lizenz dafür. Eine andere Möglichkeit war eine Kopie der Harddisk zu machen und sie anschliessend zu vergleichen. Dies

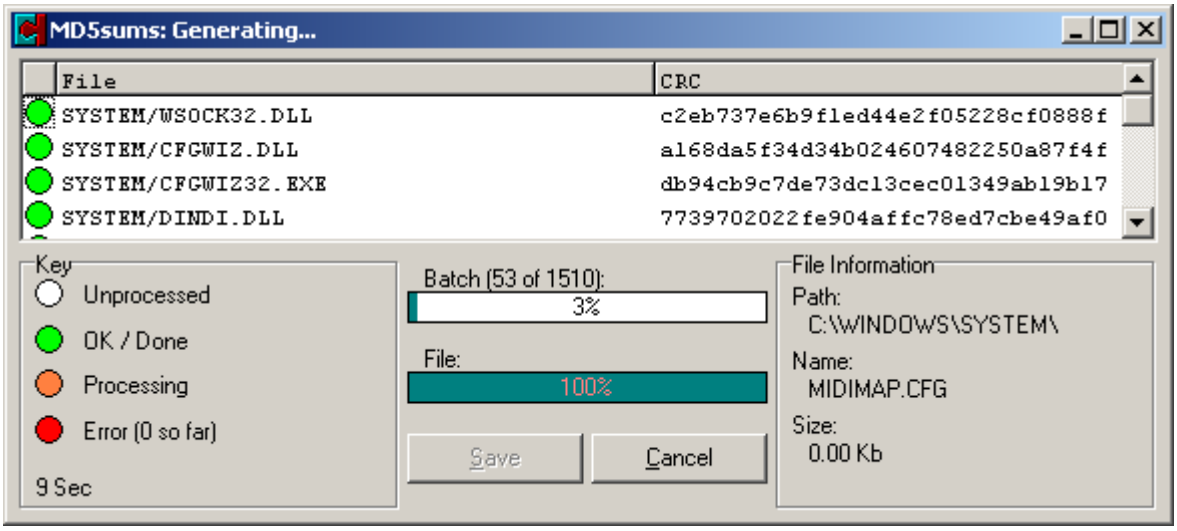
schien uns eine logische und relativ einfache Lösung zu sein. Das kopieren der Harddisk konnten wir mit Norton Ghost [52] realisieren, doch für der Vergleich mussten wir noch eine einfache Lösung finden. Aus dem privaten Gebrauch kannten wir das Programm md5sum [53]. Mit diesem Programm kann von einer Datei ein sogenannter Fingerprint oder Message-Digest angefertigt werden. Damit kann man mit einer relativ hohen Sicherheit sagen, ob diese Datei z.B. während dem Download verändert wurde oder nicht. Doch da wir ja eine ganze Festplatte vergleichen mussten, würde der Aufwand mit dem zeilenorientierten Programm md5sum einen zu grossen Arbeitsaufwand machen. Also suchten wir weiter nach einer besseren Lösung. Da wir lange nichts gefunden hatten, dachten wir nun darüber nach, selbst ein kleines Programm zu schreiben, das diese Funktion übernehmen könnte. Nachdem wir ein kleines Dos Batch-Programm geschrieben hatten, verliess uns jedoch die Motivation. Darum suchten wir nochmals intensiv im Internet. Und siehe da, wir fanden MD5summer [54]! Dieses Programm machte genau das was wir wollten. Hier ein Screenshot:



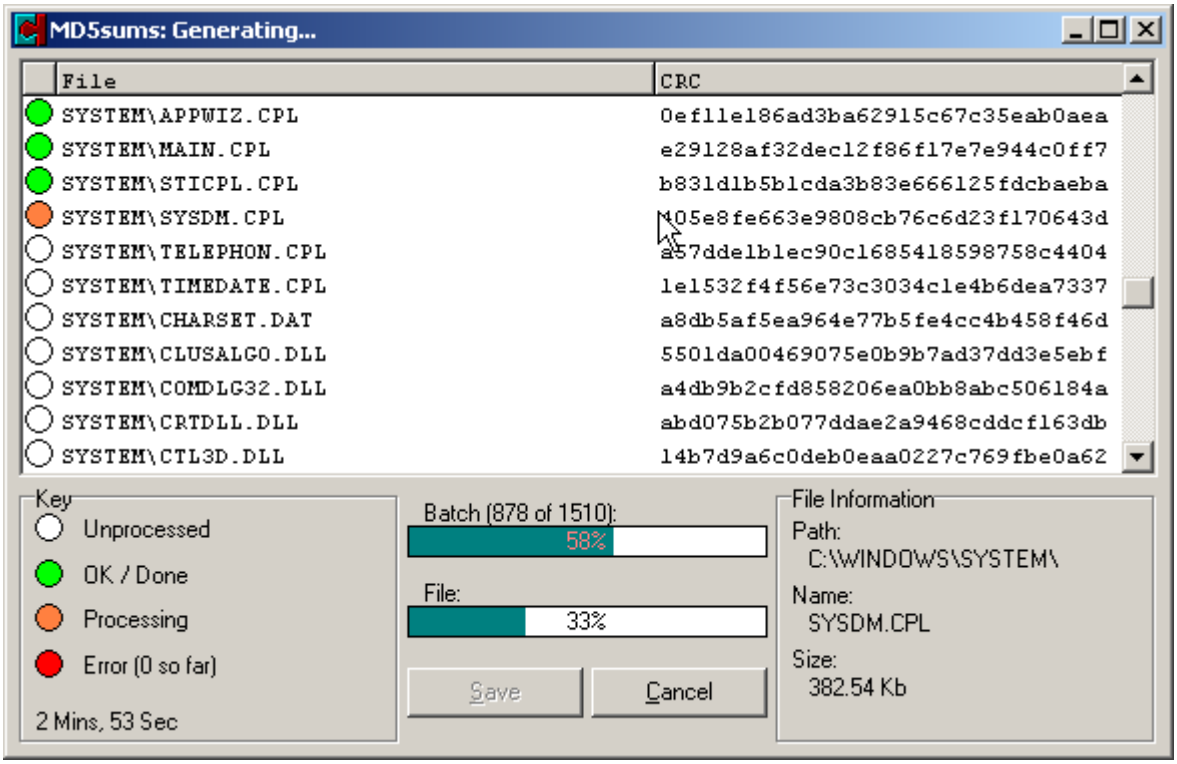
Man kann dem Programm die Dateien oder Ordner angeben und es generiert für jede einzelne Datei eine Checksumme. Diese Checksummen werden in eine Textdatei gespeichert. Hier sehen Sie ein Beispiel, wie wir die Ordner System und System32 eines Windows 98 Rechners überprüfen:



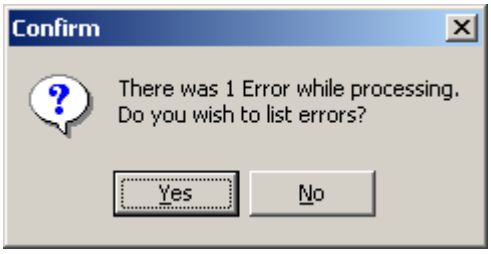
Hier ist das Programm am generieren der Checksummen:

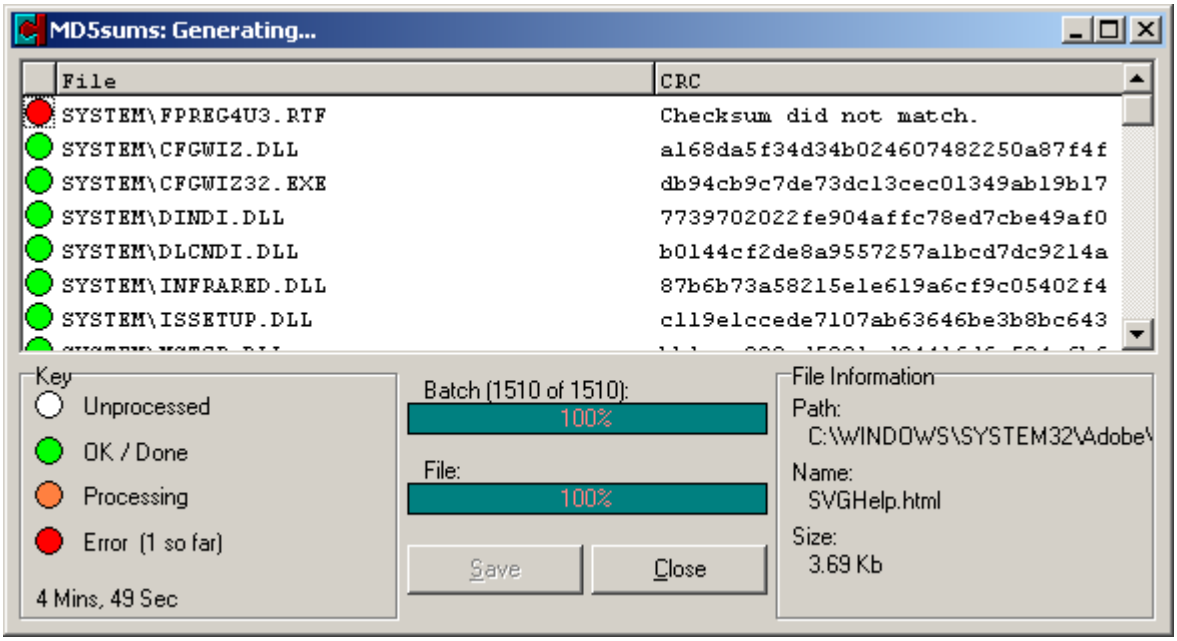


Nun veränderten wir in genau einer Datei (fpreg4u3.rtf) einen Buchstaben. Anschliessend starteten wir wieder MD5summer und liessen die zwei Verzeichnisse überprüfen (Verify sums).:



Und siehe da, das Programm erkannte die Änderung:



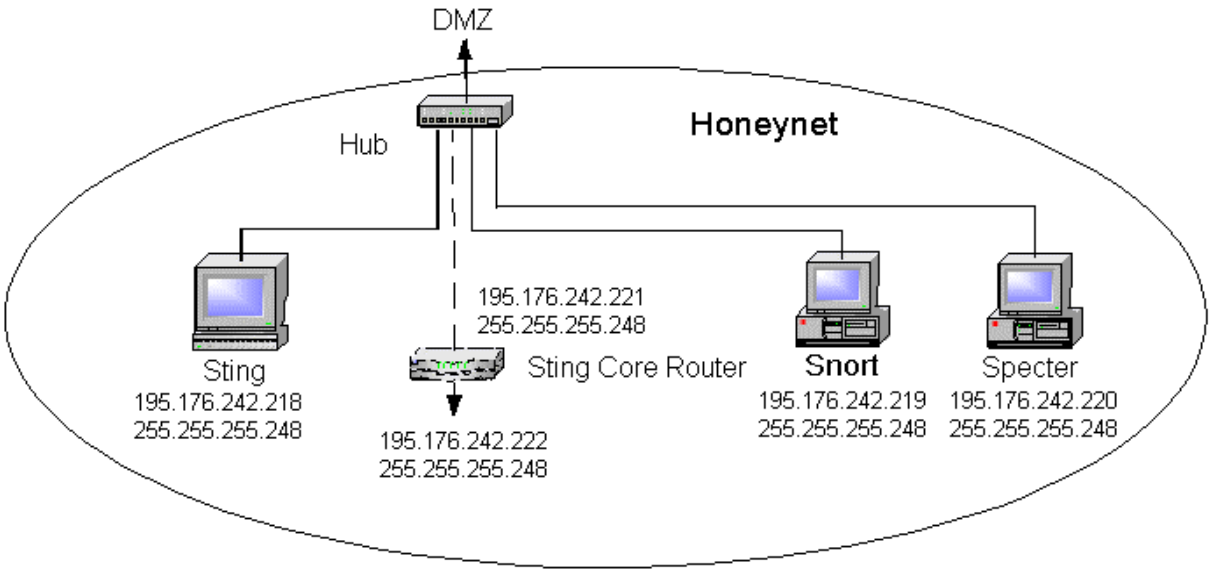


Bemerkung:
Um jetzt die Daten in einer Abbildungsdatei überprüfen zu können, muss zuerst diese Datei „entpackt“ werden. Dies kann mittels Ghost Explorer oder mit Ghost selbst geschehen. Wir hatten jedoch mit Ghost Explorer etwas Mühe, da er Dateien wie die Auslagerungsdatei von Windows nicht extrahieren konnte. Darum wählten wir die Lösung mit Ghost selbst. Das heisst, jede Abbildungsdatei wird retour auf eine Festplatte kopiert und anschliessend überprüft.

7 Unser Honeynet

7.1 Aufbau eines Honeynetzes

Der nächste Schritt unserer Semesterarbeit war die Planung eines Honeynetzes. Als erstes mussten wir entscheiden, welche Software wir einsetzen. Snort und Specter standen bereits schon fest. Nun wählten wir CyberCop Sting als dritte Software. Bei Snort nahmen wir die Version, welche unter Microsoft Windows lauffähig ist, da eine parallel laufende Semesterarbeit auch Snort einsetzte und diese Gruppe bereits Snort unter Linux im Einsatz hatte. Dies würde zusätzlich einen Vergleich zwischen der Linux- und Windowsversion ermöglichen, sofern wir und die andere Gruppe am Ende der Semesterarbeit noch Zeit dafür finden.
Unser Honeynetz sah nun folgendermassen aus:



7.2 Sting

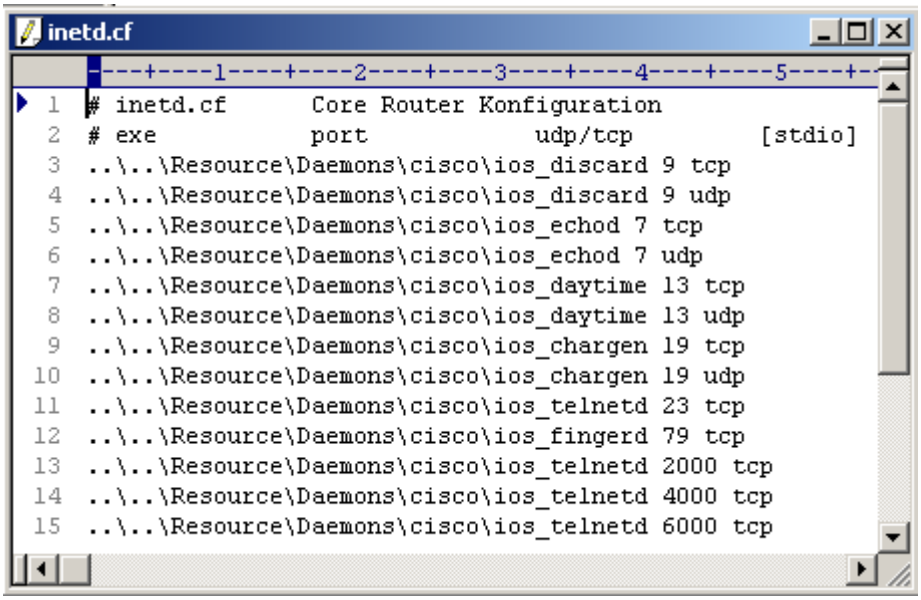
Da wir nur fünf verfügbare IP-Adressen hatten, konnten wir mit Sting nur einen virtuellen Cisco Router (Sting Core) mit zwei Schnittstellen simulieren. Die Konfigurationsdatei *sting.cf* sah so aus:

E:\Dokumente und Einstellungen\richi\Eigene Dateien\TC\Semestera...

```
#       HoneyNet Configuration
#
node core {
        node-type CISCO

        iface core {
                address 195.176.242.211/255.255.255.248
        }
        iface e0 {
                address 195.176.242.222/255.255.255.248
        }
}
```

Und das dazugehörende *inetd.cf* File (bestimmt, welche Dienste auf dem simulierten Router verfügbar sind):



7.3 Specter

Bei Specter wählten wir folgende Einstellungen:

System name :	Armenia
Config file version :	1.0
Maximum connections :	5
Connection throttle :	on
Connections/min. :	10
Flood blocking :	on
Flood limit :	100
Send status mail :	yes
Status mail period :	24 [h]
Send mails :	yes
Send short mails :	yes
Log to files :	yes
Log to event log :	yes
Log to syslog :	no
Do finger probe :	no
Do port scan :	no
Whois lookup :	yes
Log telnet banner :	no
Log ftp banner :	no
Log smtp banner :	no
Log http document :	no
Log http header :	no
Custom warning msg. :	no
Use web graphics :	no
Use custom web doc. :	no
Expect friendly con. :	no
Remote management :	no
Trace route :	yes
Maximum hops :	30
Do reverse lookup :	yes
Send password files :	yes
Password type :	normal

Activated services :	FTP TELNET SMTP NETBUS FINGER HTTP
Activated traps :	DNS SUN-RPC SUBSEVEN POP3 IMAP BO2K
Mail Server :	195.176.242.36
Mail Address :	fw@fw.tlab.ch
Short Mail Address :	fw@fw.tlab.ch
Role OS :	Linux
Role Character :	Open System
Role Hostname :	Armenia
Crowd Level :	Multiple users
User Names :	Default

Dabei ist zu erwähnen, dass wir die Optionen Portscan und die Banner Sniffer aus dem Grunde nicht ausgewählt haben, um den Angreifer nicht darauf aufmerksam zu machen, dass es sich hier bei uns um eine Falle handelt.

7.4 Test

Bevor wir mit dem Aufzeichnen beginnen konnten, mussten wir noch mittels Norton Ghost eine Kopie der drei vorbereiteten Rechner anfertigen. Nachdem wir das erledigt hatten, konnten wir loslegen. Das war am 16. Juli 2001. In der Zeit bis am 23. Juli zeichnete unser Honeynet Angriffe auf, welche „normal“ am Telecom Labor geschehen. Das heisst, wir machten keine speziellen Angriffe und wir hatten auch niemandem sonst gesagt, dass sie unser System testen sollen. So wollten wir aufzeichnen, was die Firewall des Telecom Labors tagtäglich über sich ergehen lassen muss.

Am 23. Juli stoppten wir die Rechner, um von allen Festplatten der drei Recher wieder eine Kopie zu machen. Anschliessend wurden alle Rechner wieder gestartet und die Aufzeichnungen konnten weitergehen. Denn am späten Nachmittag dieses Tages versuchte Martin Lechmann, unser System auf Herz und Nieren zu prüfen. Dabei hatten wir Martin nur unseren IP-Bereich gegeben. Martin führte seine Test etwa von 15 Uhr bis am 21 Uhr. Am 24. Juli machten wir anschliessend nochmals eine Kopie unserer Festplatten.

In der Zeit bis zum 12. August 2001 versuchten wir anschliessend, noch ein paar andere „Spezialisten“ auf unser Netz aufmerksam zu machen. Dabei suchten wir diverse Hackerboard auf, um dort die gewünschten Personen zu finden. Dies erwies sich aber schwieriger als angenommen. Denn in den Board’s konnten wir nicht einfach unseren IP-Bereich veröffentlichen, denn dies verstösst gegen die Boardregeln, falls man sich nicht an diese Regeln hält, wird der Eintrag sofort gelöscht. Aber auch nur die Suche nach Personen, die unser System testen würden, stoss auf heftigsten Widerstand. Dies ist natürlich auch verständlich, denn wir könnten ja irgendeine IP-Adresse angeben, welche uns gar nicht gehört, um so eine Firma zu schädigen, obwohl wir das natürlich nicht wollten. Nach ein paar recht „bösen“ E-Mails stellten wir die Suche nach den „Spezialisten“ ein und wir genossen unsere Sommerferien. Unser Netz zeichnete unterdessen bis am 12. August 2001 weiter auf. Denn an diesem Datum war unser Test zu Ende und wir konnten mit der Auswertung beginnen.

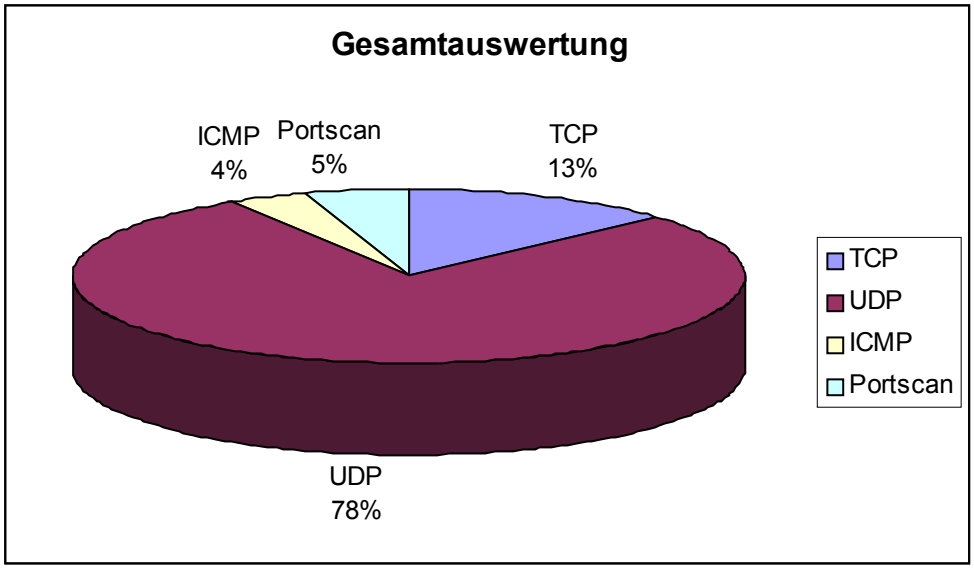
8 Auswertung und Ergebnisse

8.1 Auswertung Snort

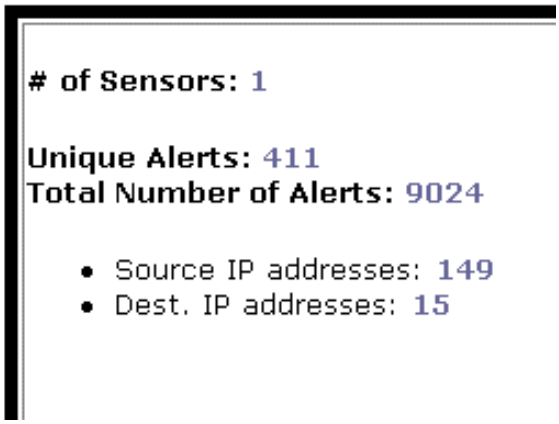
Die Auswertung mit Snort übernimmt ACID wie im Kapitel 5.2.2.3 erwähnt wurde. Die Daten werden also aus der MySQL Datenbank herausgelesen und mit ACID aufbereitet. Da die Auswertung von ACID noch zu wünschen übrig lässt, wurden zum Teil auch Grafiken mit Excel gemacht.

8.1.1 Auswertung über die ganze Testdauer mit ACID

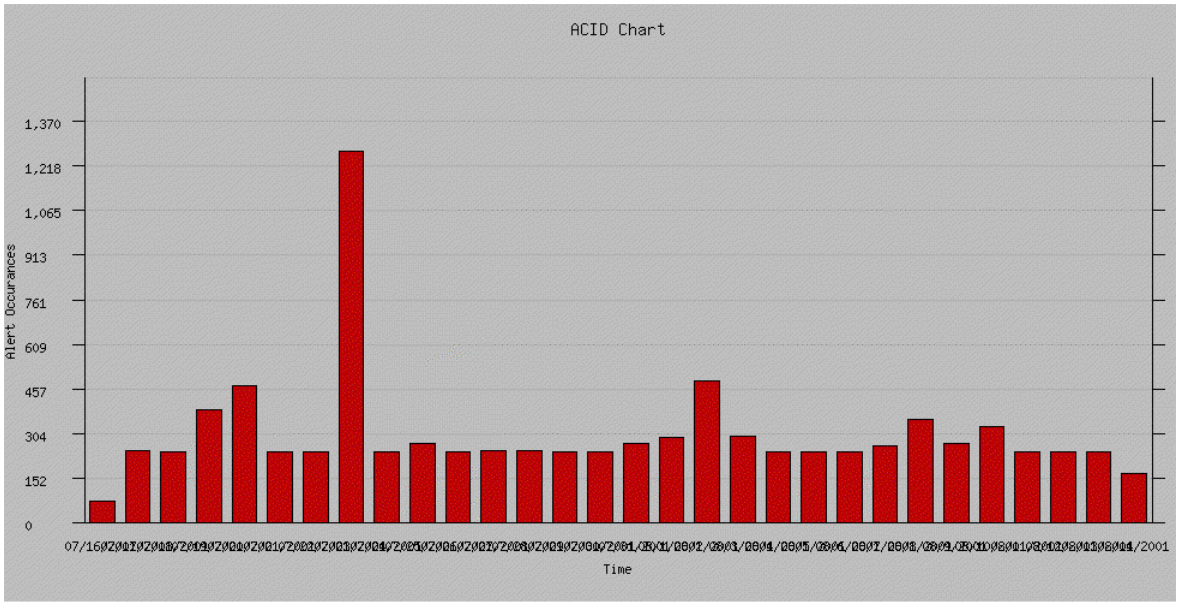
Um die Auswertung von ACID anzuschauen kann http://localhost/acid eingegeben werden. Die Grobauswertung über die ganze Testdauer von 30 Tagen, zeigt folgendes Bild. Den grössten Anteil machen die UDP Pakete aus.



In der gesamten Testphase hat Snort eine Gesamtzahl von 9024 Alerts erkannt. Davon sind 411 unterschiedliche Angriffstypen. Aufgeführt sind auch noch die Source- und Destination- IP-Adressen.



Wenn man die Anzahl Angriffe auf die 30 Testtage verteilt anschaut sieht man, dass es beim 23.07.2001 eine grosse Spitze hat. Das ist der Zeitpunkt, wo unser Honeynet besonders oft angegriffen wird. Die Erklärung ist sehr einfach. An diesem Tag hat Martin Lechmann unser Netzwerk getestet, mehr darüber aber im nächsten Kapitel.

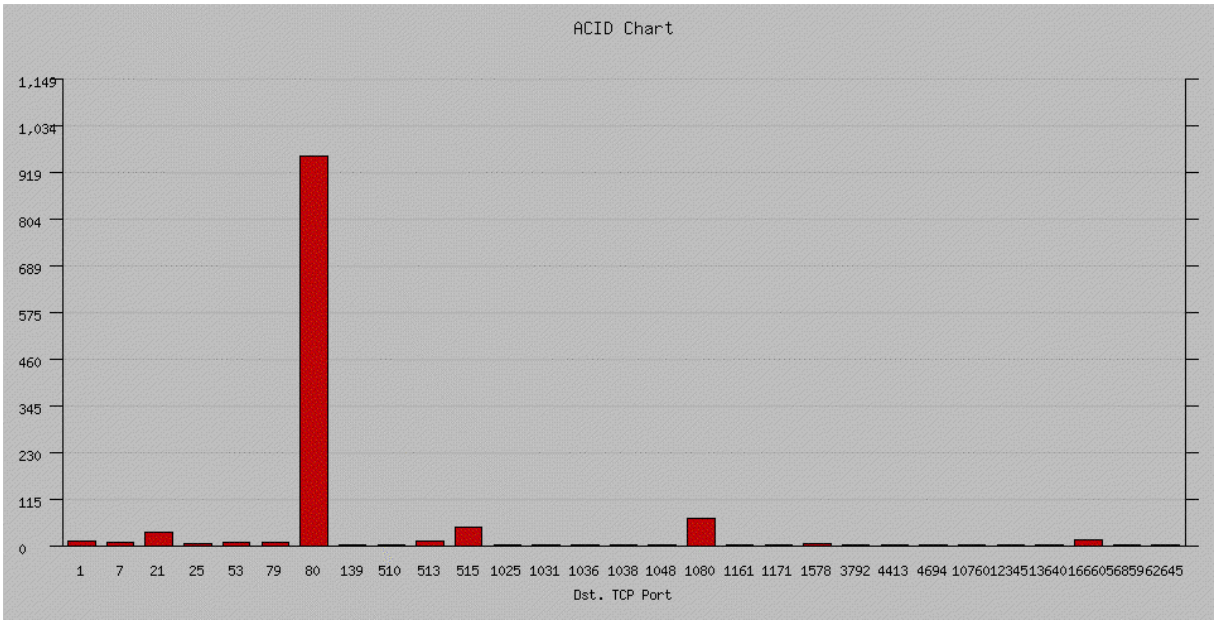


Um einen Überblick zu bekommen, was an diesem Tag alles passiert ist, kann man mit der Suchfunktion von ACID arbeiten. Hier ein kleiner Ausschnitt von einem beliebigen Tag. Nicht alle Angriffe können hier in der Dokumentation angeschaut werden, darum beschränken wir uns auf ein Beispiel, um das Prinzip und die Arbeitsweise von ACID aufzuzeigen. Alle Angaben können in der Datenbank problemlos angeschaut werden.
In der ersten Spalte ist die Nummer des Alerts angegeben. In der zweiten sind Angaben über die Signatur die angesprochen hat und einen Hinweis, was es ist. z.B IDS 115 / scan_Traceroute UDP. In der dritten Spalte sind Datum und genaue Zeit angegeben. In der vierten und fünften sind Source- und Destination-Adress aufgeführt und zuletzt noch das Protokoll.

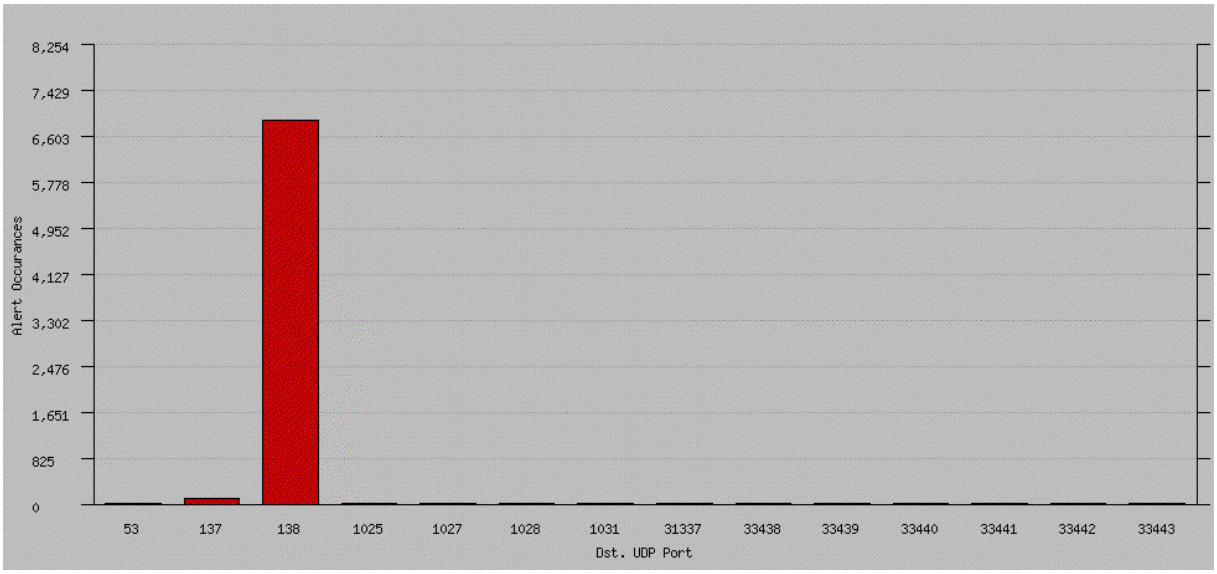
☐	#773-(1-774)	IDS115/scan_Traceroute UDP	2001-07-19 17:06:26	195.176.242.227	195.176.242.255	UDP
☐	#774-(1-775)	IDS115/scan_Traceroute UDP	2001-07-19 17:06:26	195.176.242.227	195.176.242.255	UDP
☐	#775-(1-776)	IDS296/web-misc_http-whisker-splicing-attack-space	2001-07-19 17:06:33	168.37.171.82	195.176.242.220	TCP
☐	#776-(1-777)	IDS552/web-iis_IIS ISAPI Overflow ida	2001-07-19 17:06:33	168.37.171.82	195.176.242.220	TCP
☐	#777-(1-778)	IDS552/web-iis_IIS ISAPI Overflow ida	2001-07-19 17:06:33	168.37.171.82	195.176.242.220	TCP
☐	#778-(1-779)	IDS243/web-cgi_http-cgi-pipe	2001-07-19 17:06:34	168.37.171.82	195.176.242.220	TCP
☐	#779-(1-780)	IDS243/web-cgi_http-cgi-pipe	2001-07-19 17:06:34	168.37.171.82	195.176.242.220	TCP
☐	#780-(1-781)	IDS115/scan_Traceroute UDP	2001-07-19 17:18:27	195.176.242.227	195.176.242.255	UDP

Wenn man jetzt in der zweiten Spalte auf die Numerierung der Signatur klickt, dann kommt man zu den Seiten von Whitehats.com. Dort ist dann z.B IDS 115 genauer erklärt. Man bekommt eine kleine Zusammenfassung über das Ereignis, Protokoll und wie die Regel in Snort programmiert ist. Dazu kommen viele Informationen über das Benutzte Protokoll und einige Infos über den Angriff. Diese Seite ist recht ausführlich und zusammen mit ACID sehr wertvoll für Snort.

Ausserdem können die Ports auch mit ACID angeschaut werden. Hier sind die Destination TCP Ports aufgeführt.



Der Port 80 überragt die anderen Ports bei weitem. Der Grund ist, während der Testphase mussten wir mit der Semesterarbeit weiterfahren. Dabei benutzten wir sehr oft das Internet und HTTP geht halt über Port 80. Im nächsten Kapitel haben wir dann ein Beispiel genauer angeschaut. Es soll hier nur die Möglichkeiten von ACID aufgezeigt werden.
Die gleiche Auswertung kann auch mit den UDP Destination Ports gemacht werden:



Hier werden die Ports 138 und 139 sehr viel benutzt. Diese Ports werden von NETBIOS Session Service benutzt.

Die Auflistung der IP Adressen mit dem grössten Verkehrsaufkommen ist im folgenden Bild ersichtlich. Dabei kann man mit Hilfe von ACID die IP-Adresse aufgelöst werden. Damit kann man mehr Informationen über den Angreifer herausfinden.

< IP Address >	FQDN	# of Sensors	< # of Occurances >
195.176.242.227	trimmis.tlab.ch	1	6956
213.3.189.239	bw2-189pub239.bluewin.ch	1	261
213.3.59.132	bw2-59pub132.bluewin.ch	1	192
195.186.169.134	bw1-169pub134.bluewin.ch	1	175
195.176.242.66	URUGUAY.tlab.ch	1	140
195.176.242.218	belarus.tlab.ch	1	58
195.176.242.220	armenia.tlab.ch	1	47
210.200.248.175	mse.e2100.com.tw	1	44
213.3.58.113	bw2-58pub113.bluewin.ch	1	41
195.176.242.233	felsberg.trapper.ch	1	34
210.33.68.1	*Unable to resolve address*	1	30

8.1.2 Auswertung der Angriffe von Martin Lechmann (23.07.2001)

Um unser System von einem Fachman testen zu lassen, haben wir Martin Lechmann gebeten, unser System zu prüfen. Wir einigten uns auf den 23.07.2001. Es war ein guter Zeitpunkt für den Test. Wir hatten in dieser Zeit Ferien und so hatten wir mehr Zeit, um die Tests besser zu verfolgen. Wir gaben ihm also nur unsere IP's an und er konnte dann seine Scans und Checks durchführen.

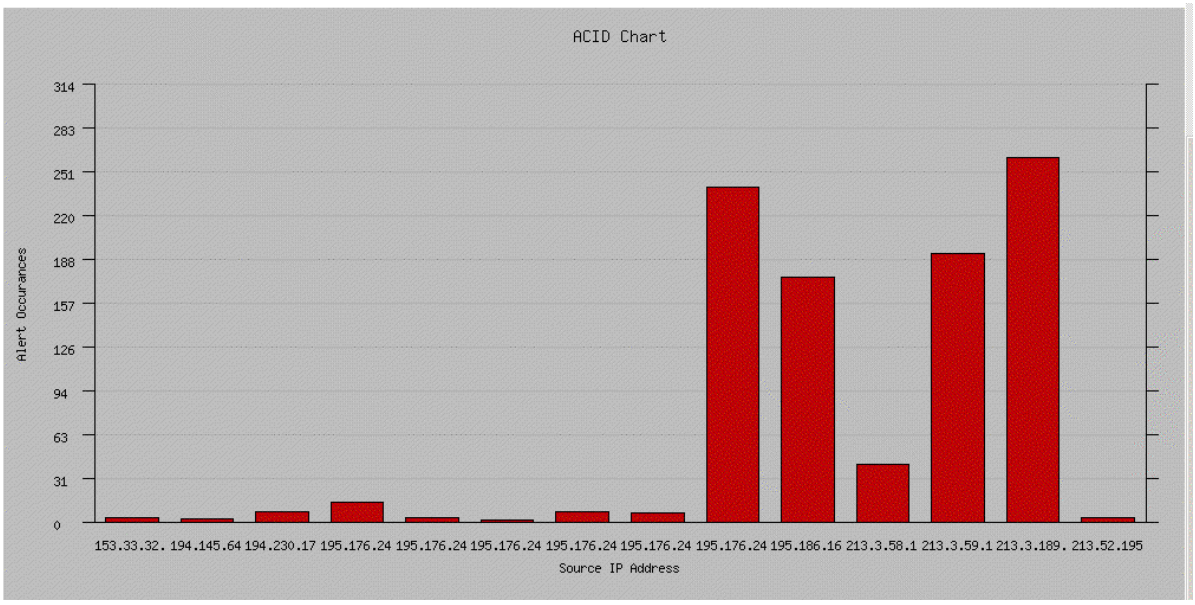
Die Angriffe seinerseits wurden auch dokumentiert, so hatten wir eine Referenz und konnten unsere Ergebnisse mit seinen Dokumenten vergleichen. Damit können wir die Zuverlässigkeit unseres Systems besser beurteilen.

In den folgenden Abschnitten wird erklärt, wie wir vorgegangen sind, um unsere Daten besser auszuwerten.

Wenn man mit ACID eine Grafik aufstellt, mit den Parametern „Angriffe verteilt auf die IP's" sieht man, dass einige Source Adressen eine sehr grosse Anzahl Angriffe im Vergleich zu anderen IP's haben. Daraus ist zu schliessen, dass diese IP's von Martin Lechmann benutzt wurden. Wenn man die Grafik mit Angabe der Pixel vergrössert, können die IP's besser abgelesen werden. Die IP's mit den meisten Angriffen sind:

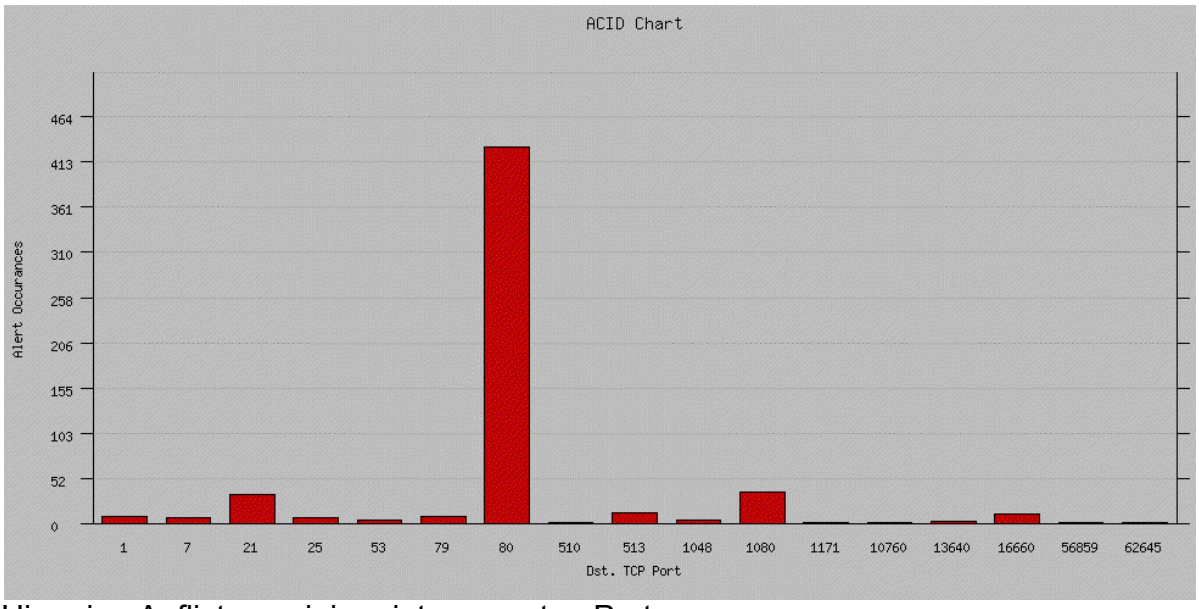
IP:	Zeit:	Anzahl Angriffe:
213.3.189.239	15.09 - 15.40 Uhr	261
195.186.169.134	16.23 - 17.45 Uhr	175
213.3.58.113	17.26 - 17.49 Uhr	41
213.3.59.132	19.19 - 19.34 Uhr	192

In der Grafik sind fünf Balken mit hohen Angriffszahlen zu sehen. In der obigen Tabelle sind nur 4 IP's aufgeführt. Die fünfte IP ist die IP des trimmis.tlab.ch, dies ist der Gateway von Telecom Labor, welcher viel Verkehr produziert.



Damit dies nicht als Angriff gewertet wird, muss bei den Regeln (Snort1.conf) die IP dieses Rechner angegeben werden. Dann wird dieser Verkehr gar nicht aufgezeichnet. Es ist aber trotzdem noch interessant, denn so sieht man, was im internen Netz so alles läuft.

Diese Grafik zeigt auf, wie die Angriffe auf die Ports verteilt sind über den ganzen 23.07.2001 gesehen.

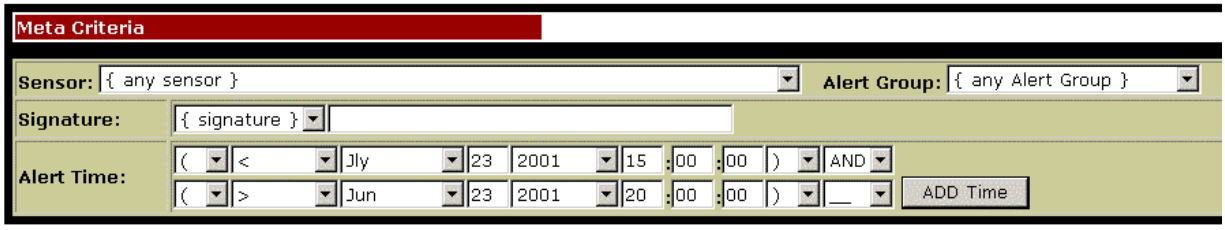


Hier eine Auflistung einiger interessanten Ports:

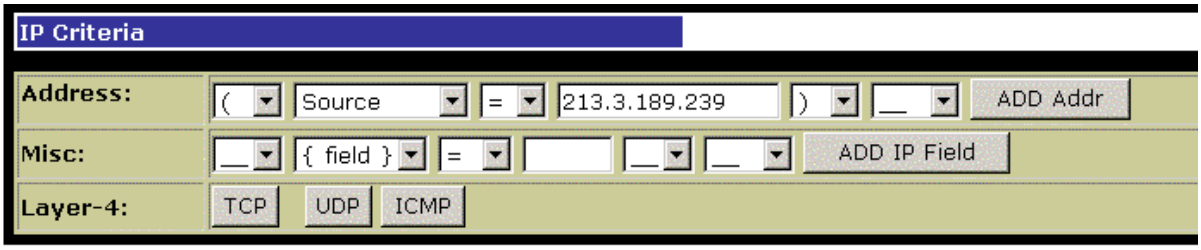
Port:	Dienst:	Port:	Dienst:
21	FTP	79	Finger
25	Email / SMTP	80	WWW
53	DNS	1080	Socks

Im Bereich von 1-1023 sind die Well-Known Ports definiert. Ab 1023 bis 49151 sind die Ports reserviert für Programme und Anwendungen. Und ab 49151 bis 65535 sind die Ports für alle frei wählbar.
Aus der Grafik ist ersichtlich, dass die Ports aus allen drei Bereichen vorkommen.

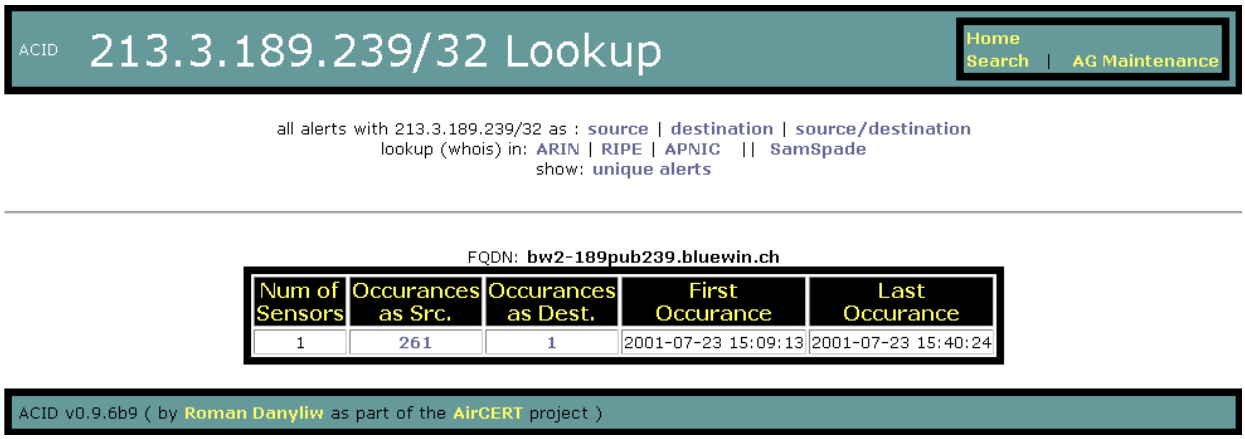
Durch die Suchfunktion kann jetzt nach Angriffen in einem bestimmten Zeitraum durch ACID angezeigt werden lassen. Durch die IP's die M. Lechmann benutzt hat, untersuchen wir den Zeitraum zwischen 15.00 Uhr und 20.00 Uhr. Wir lassen ACID alle stattgefundenen Angriffe anzeigen.



Man kann aber auch nach Source IP's die Alerts auflisten lassen. Wir wissen aus der ersten Grafik, was für IP's Adressen er hat.



Die Adressauflösung kann für jede IP über ACID einfach gemacht werden.



Man muss einfach auf die IP geklickt werden. Dabei wird aufgezeigt wieviele Alerts die IP erzeugt hat und in welchen Zeitraum dies passiert ist.
Für genauere Angaben kann noch auf der RIPE (Réseaux IP Européens) Seite nachgeschaut werden, diese Option stellt ACID auch zur Verfügung.

- Whois – Abfrage bei RIPE:

213.3.189.239



% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html

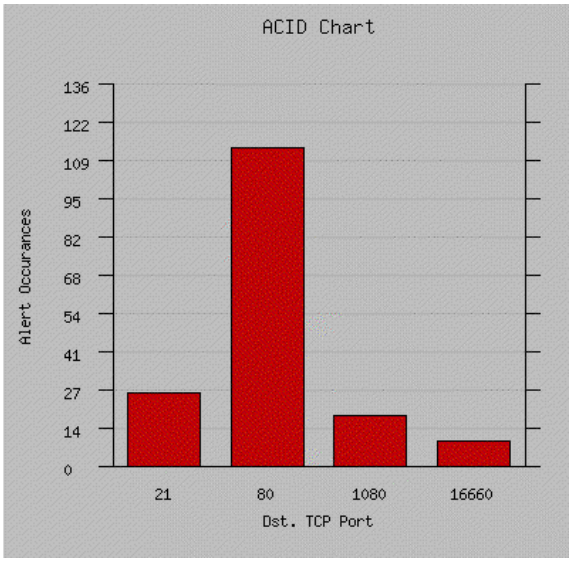
inetnum:	**213.3.128.0 - 213.3.255.255**
netname:	BLUEWINNET
descr:	Bluewin is an internet service provider in CH.
country:	CH
admin-c:	PZ1009-RIPE
tech-c:	AM1626-RIPE
tech-c:	MR1192-RIPE
rev-srv:	dns1.bluewin.ch
rev-srv:	dns2.bluewin.ch
rev-srv:	dns3.bluewin.ch
status:	ASSIGNED PA
notify:	ipmaster@bluewin.ch
mnt-by:	RIPE-NCC-NONE-MNT
changed:	ipmaster@bluewin.ch 20010528
source:	RIPE

Ersichtlich ist, dass der Angreifer Kunde von Bluewin sein muss. Man sieht aber nur den IP Bereich des Providers und keine Kundendaten.

- **Auswertung der einzelnen IP's**

195.186.169.134

IP:	Zeit:	Anzahl Angriffe:
195.186.169.134	16.23 - 17.45 Uhr	175



ACID liefert uns eine gute Auflistung der benutzten Ports.

Ports 21, 80, 1080, 16660

Die ersten zehn Angriffe die Snort von dieser IP registriert hat, wollen wir näher untersuchen.

ID	< Signature >	< TimeStamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
#0-(1-2473)	IDS175/misc_socks-probe	2001-07-23 16:23:17	195.186.169.134	195.176.242.219	TCP
#1-(1-2474)	IDS175/misc_socks-probe	2001-07-23 16:23:18	195.186.169.134	195.176.242.219	TCP
#2-(1-2475)	IDS175/misc_socks-probe	2001-07-23 16:23:19	195.186.169.134	195.176.242.219	TCP
#3-(1-2480)	IDS179/ddos_ddos-stacheldraht client	2001-07-23 16:23:41	195.186.169.134	195.176.242.219	TCP
#4-(1-2481)	IDS179/ddos_ddos-stacheldraht client	2001-07-23 16:23:42	195.186.169.134	195.176.242.219	TCP
#5-(1-2482)	IDS179/ddos_ddos-stacheldraht client	2001-07-23 16:23:43	195.186.169.134	195.176.242.219	TCP
#6-(1-2487)	IDS175/misc_socks-probe	2001-07-23 16:24:15	195.186.169.134	195.176.242.220	TCP
#7-(1-2489)	IDS175/misc_socks-probe	2001-07-23 16:24:15	195.186.169.134	195.176.242.220	TCP
#8-(1-2490)	IDS175/misc_socks-probe	2001-07-23 16:24:16	195.186.169.134	195.176.242.220	TCP
#9-(1-2495)	IDS179/ddos_ddos-stacheldraht client	2001-07-23 16:24:41	195.186.169.134	195.176.242.220	TCP
#10-(1-2496)	IDS179/ddos_ddos-stacheldraht client	2001-07-23 16:24:42	195.186.169.134	195.176.242.220	TCP
#11-(1-2497)	IDS179/ddos_ddos-stacheldraht client	2001-07-23 16:24:42	195.186.169.134	195.176.242.220	TCP

Im Feld <Signature> sind die „Arten" der Angriffe aufgelistet. Durch klicken auf z.B. IDS175 kommt man zur Homepage von Whitehats.com. Dort ist eine genaue Erklärung des Angriffes aufgeführt. IDS 175: http://www.whitehats.com/IDS/175

WHITEHATS

Summary
This event indicates that someone is scanning your system to see if it is running SOCKS. This may be a hacker that desires to "bounce" traffic through your system or a chat server (trying to determine if someone is bouncing through your system to chat anonymously).

How Specific
This event is specific to a vulnerability, but may have been caused by any of several possible exploits. Packet payload is not considered in the signatures used to detect this attack.
Trusiting The Source IP Addrss
Although this event was caused by a TCP packet, the packet is not thought to be a part of an existing TCP session. Therefor the source IP address could be easily forged. It has been noted that the intruder is likely to expect or desire a response to their packets, so it may be likely that the source IP address is not spoofed

Für alle Signaturen besteht eine solche Auswertung.

Auf der IP 195.186.169.134 haben folgende Signaturen angesprochen:

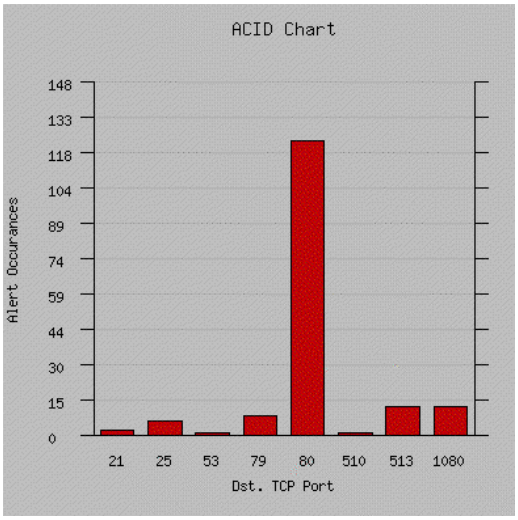
Signatur:	Erklärung:	Signatur:	Erklärung:
IDS 169	ICMP_PING-Win9x2000	IDS 228	WEB-CGI_HTTP-GUESTBOOK
IDS 175	MISC-SOCKS-PROBE	IDS 235	WEB-CGI_HTTP-HANDLER
IDS 179	DDOS-STACHELDRAHT-CLIENT	IDS 258	WEB-CGI_HTTP-GET32.EXE
IDS 205	WEB-CGI_HTTP-PHORUM-ADMIN	IDS 268	WEB-COLDFUSION-APPLICATION.CFM
IDS 207	WEB-CGI_HTTP-CODE	IDS 272	WEB-CGI_HTTP-PIRANHA-PASSWD.PHP3
IDS 218	WEB-CGI_HTTP-TEST-CGI	IDS 311	PING SCANNER-L3RETRIEVER
IDS 219	WEB-CGI_HTTP-CGI-PERL.EXE	IDS 326	FTP_FTP-USER-ROOT
IDS 221	WEB-CGI_HTTP-CGI-FINGER	IDS 469	WEB-CGI-WEBSENDMAIL
IDS 225	WEB-CGI_HTTP-CGI-ANYFORM	IDS 472	WEB-CGI_WEBGAIS

Die interessanten sind:

- MISC-SOCKS-PROBE: Scannen des Systems, um vielleicht laufende Socks zu finden.
- DDOS-STACHELDRAHT-CLIENT: Versuch, ob ein Stacheldraht Client installiert ist.
- FTP_FTP-USER-ROOT: Man hat probiert, ob man mit dem Passwort root per FTP eindringen kann.

213.3.189.239

IP:	Zeit:	Anzahl Angriffe:
213.3.189.239	15.09 - 15.40 Uhr	261



Die benutzten Ports sind:

Ports: 21, 25, 53, 79, 80, 510, 513, 1080

Signatur:	Erklärung:	Signatur:	Erklärung:
IDS 6	MISC-SOURCEPOR-TTRAFFIC-20-TCP	IDS 232	WEB-CGI-HTTP-CGI-PHP-SLASH
IDS 31	SMTP-EXPN-ROOT	IDS 243	WEB-CGI-HTTP-CGI-PIPE
IDS 32	SMTP-EXPN-DECODE	IDS 250	WEB.COLDFUSION-OPEN-FILE.CFM
IDS 128	WEB-CGI-HTTP-CGI-PHF	IDS 258	WEB-CGI_HTTP-GET32.EXE
IDS 169	ICMP_PING-Win9x2000	IDS 278	DNS-NAMED-PROBE-VERSION
IDS 171	ICMP-PING-ZEROS	IDS 292	WEB-FRONTPAGE-SHTML.DELL
IDS 218	WEB-CGI_HTTP-TEST-CGI	IDS 296	WHISKER-SPLICING-ATTACK-SPACE
IDS 219	WEB-CGI_HTTP-CGI-PERL.EXE	IDS 297	HTTP-DIRECTORY-TRAVERSAL-1
IDS 221	WEB-CGI_HTTP-CGI-FINGER	IDS 298	HTTP-DIRECTORY-TRAVERSAL-2
IDS 225	WEB-CGI_HTTP-CGI-ANYFORM	IDS 376	FINER-FINGER-ROOT
IDS 226	WEB-CGI-HTTP-CGI-FORMMAIL	IDS 469	WEB-CGI-WEBSENDMAIL
IDS 231	WEB-CGI-HTTP-CGI-WIN-C		

Die interessanten sind:

- ICMP_PING-Win9x2000

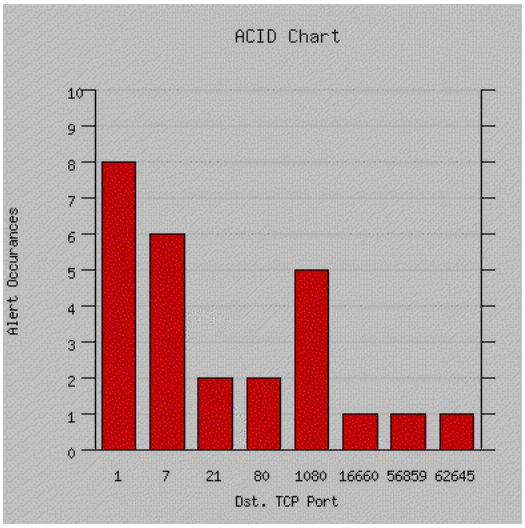
  Man hat versucht Informationen über den „root account“ herauszufinden. (Unix,...)
- DNS-NAMED-PROBE-VERSION

  Versuch um BIND im DNS Server zu beenden. Ist oft eine Vor-Attacke
- WHISKER-SPLICING-ATTACK-SPACE:

  Ein ungewöhnlich kurzes Paket welches versucht eine TCP Session zu fragmentieren.

213.3.58.113

IP:	Zeit:	Anzahl Angriffe:
213.3.58.113	17.26 - 17.49 Uhr	41



Benutzte Destination TCP Ports:

Ports: 1, 7, 21, 80, 1080, 16660, 56859, 62645

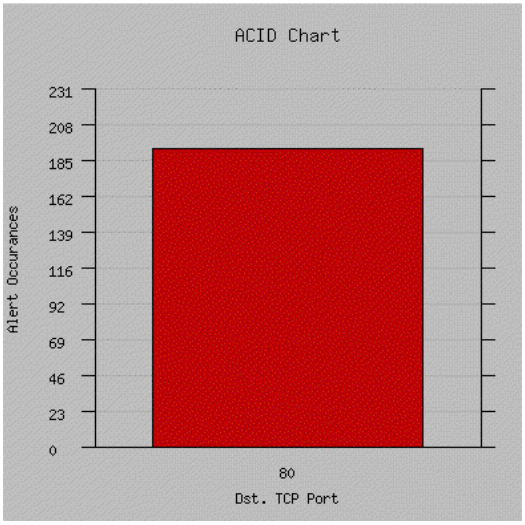
Signatur:	Erklärung:	Signatur:	Erklärung:
IDS 5	NMAP-FINGERPRINT-ATTEMPT	IDS 179	DDOS-STACHELDRAHT-CLIENT
IDS 28	NMAP-TCP-PING	IDS 188	TROJEN-PROBE-BACKORIFICE
IDS 30	PROBE-XMAS-SCAN	IDS 228	WEB-CGI_HTTP-GUESTBOOK
IDS 152	ICMP-PING BSDTYPE	IDS 272	WEB-CGI_HTTP-PIRANHA-PASSWD.PHP3
IDS 162	SCAN-PING-NMAP-ICMP	IDS 521	SYNSCAN-PORTSCAN-ID-19104
IDS 175	MISC-SOCKS-PROBE		

Die interessanten sind:

- NMAP-FINGERPRINT-ATTEMPT: Mit NMAP das System gescannt, um Angaben über OS,.. zu bekommen.
- TROJEN-PROBE-BACKORIFICE: Versuch, um über den default Port des Backorifice Trojans, ins System einzudringen
- WEB-CGI_HTTP-PIRANHA-PASSWD.PHP3: Piranha passwd.php ist ein GUI Skript. Man hat probiert ob man Zugriff zu diesem hat, um dann den Webserver mittels Shell Commands umzukonfigurieren.

213.3.59.132

IP:	Zeit:	Anzahl Angriffe:
213.3.59.132	19.19 - 19.34 Uhr	192



Mit dieser IP wurde nur ein Port benutzt.

Port: 80

Signatur:	Erklärung:	Signatur:	Erklärung:
IDS 205	WEB-CGI_HTTP-PHORUM-ADMIN	IDS 228	WEB-CGI_HTTP-GUESTBOOK
IDS 207	WEB-CGI_HTTP-CODE	IDS 235	WEB-CGI_HTTP-HANDLER
IDS 218	WEB-CGI_HTTP-TEST-CGI	IDS 258	WEB-CGI_HTTP-GET32.EXE
IDS 219	WEB-CGI_HTTP-CGI-PERL.EXE	IDS 268	WEB-COLDFUSION-APPLICATION.CFM
IDS 221	WEB-CGI_HTTP-CGI-FINGER	IDS 469	WEB-CGI-WEBSENDMAIL
IDS 225	WEB-CGI_HTTP-CGI-ANYFORM	IDS 472	WEB-CGI-WEBGAIS

Die interessanten sind:

- WEB-CGI_HTTP-CGI-PERL.EXE: Man hat probiert ob perl.exe ausführbar ist, mit Perl, könnte man Befehle Richtung Webserver ausführen
- WEB-CGI_HTTP-GET32.EXE: Das get32.exe vom Alibaba Server kann missbraucht werden um den Webserver umzukonfigurieren.
- WEB-COLDFUSION-APPLICATION.CFM: Versuch um ein Loch im ColdFusion Server 4.x ausnutzen zu können, um den Inhalt des Webservers anzeigen zu können.

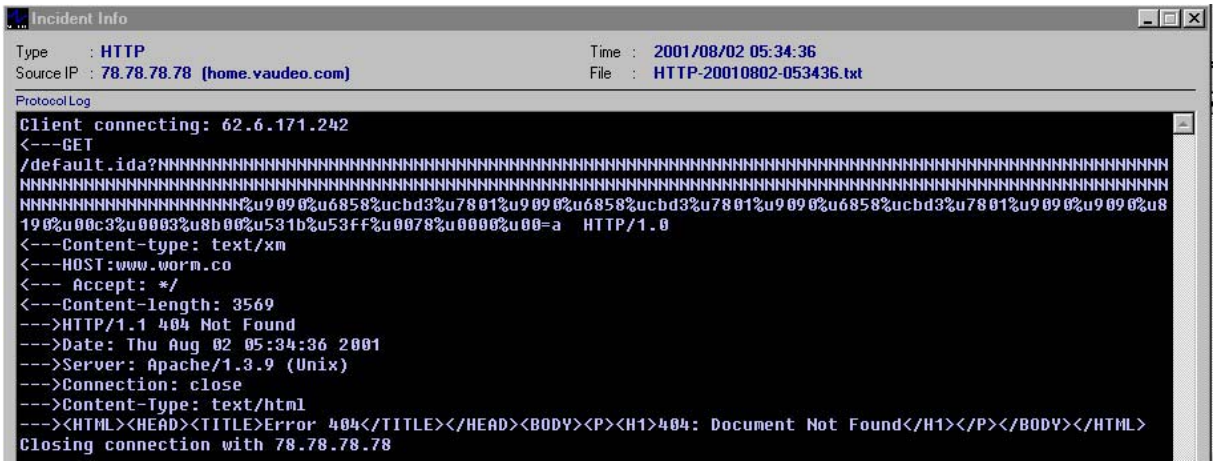
8.2 Auswertung Specter

8.2.1 Auswertung über die ganze Testdauer

Über die ganze Testdauer wurden insgesamt 172 Angriffe registriert. Davon wurden 72 von Martin Lechmann „verübt“, welche im nächsten Kapitel genauer beschrieben werden. Ausserdem erkannte Specter auch noch fünf „Angriffe“ von den anderen Rechner in unserem Honeynet. Hier ein Ausdruck der restlichen Aufzeichnungen:

FTP	2001/07/18 04:07:28	130.207.32.204 (d6eae.marc.gatech.edu)
FTP	2001/07/26 11:06:02	131.211.200.153 (jury.law.uu.nl)
HTTP	2001/07/23 13:42:21	134.68.81.93 (compeng6.sl.iupui.edu)
SUN-R...	2001/08/01 06:50:42	140.112.54.151 (hic.ch.ntu.edu.tw)
FTP	2001/07/23 18:20:05	153.33.32.222 (ts2_mod2.ltx-tr.com)
FTP	2001/07/22 18:15:57	172.185.167.188 (ACB9A7BC.ipt.aol.com)
FTP	2001/07/31 22:01:13	193.159.131.115 (pC19F8373.dip.t-dialin.net)
FTP	2001/07/29 11:52:23	193.253.185.167 (ANancy-101-1-2-167.abo.wanadoo.fr)
FTP	2001/07/29 22:21:14	193.253.185.167 (ANancy-101-1-2-167.abo.wanadoo.fr)
DNS	2001/07/30 11:20:39	194.105.9.70
FTP	2001/07/25 01:33:35	194.143.202.202 (ns.visualcom.es)
HTTP	2001/07/23 09:15:15	194.230.171.94 (pop-ls-14-4-2-dialup-94.freesurf.ch)
SUN-R...	2001/08/02 06:51:38	195.120.129.56
NETB...	2001/07/20 01:24:47	195.174.97.6
NETB...	2001/07/20 22:25:49	195.174.97.6 (ist00262.ultranet.com.tr)
NETB...	2001/07/20 22:26:16	195.174.97.6 (ist00262.ultranet.com.tr)
SUN-R...	2001/07/24 19:14:44	195.202.200.144
SUN-R...	2001/07/27 20:13:12	195.202.200.144
SUN-R...	2001/07/25 13:43:22	202.85.72.249 (ip72-249.asiaonline.net)
FTP	2001/07/24 05:20:31	203.164.95.115 (co3022884-a.rivrw1.nsw.optushome.com.au)
DNS	2001/07/28 09:32:02	209.136.35.2 (mail.planetlink.net)
FTP	2001/07/28 19:45:32	210.225.81.16 (decgt00.interzone.or.jp)
SUN-R...	2001/07/28 19:45:31	210.225.81.16 (decgt00.interzone.or.jp)
FTP	2001/07/22 12:14:31	210.254.171.129 (ns1.rdenet.com)
HTTP	2001/07/25 13:04:36	210.33.68.1
HTTP	2001/07/25 14:20:35	210.33.68.1
SUN-R...	2001/07/29 10:47:30	211.139.67.205
SUN-R...	2001/07/29 12:52:07	211.139.67.221
SUN-R...	2001/07/25 14:38:30	211.169.35.130
HTTP	2001/07/20 00:14:38	211.234.53.99
SUN-R...	2001/08/03 03:21:11	211.251.146.60
SUN-R...	2001/07/23 23:34:44	211.251.192.129
SUN-R...	2001/07/22 23:39:33	211.39.180.29
FTP	2001/07/30 05:34:43	212.161.14.89
FTP	2001/07/20 09:17:32	213.118.91.157
DNS	2001/08/01 08:18:16	213.41.85.249 (access-249.85.rev.fr.colt.net)
DNS	2001/07/23 00:31:51	213.52.195.36
SMTP	2001/07/26 02:38:16	216.25.169.23
SUN-R...	2001/07/24 10:45:33	24.7.23.200 (c828911-a.alntn1.tx.home.com)
DNS	2001/07/17 09:52:45	61.128.193.72
DNS	2001/07/17 09:52:46	61.128.193.72
SUN-R...	2001/08/03 04:13:06	61.220.247.23 (61-220-247-23.HINET-IP.hinet.net)
SMTP	2001/07/25 22:21:58	63.119.26.75 (122cust.minneapolis.mn5.dial56k.us1.rizlernet.com)

Besonders zu erwähnen sind die 53 HTTP Angriffe, welche immer als Source IP 78.78.78.78 angegeben wurde. Ausserdem wurde bei allen diesen Angriffen die genau gleiche Daten uns übermittelt:


Incident Info
Type : HTTP
Source IP : 78.78.78.78 (home.vaudeo.com)
Time : 2001/08/02 05:34:36
File : HTTP-20010802-053436.txt
Protocol Log
Client connecting: 62.6.171.242
<---GET
/default.ida?NNNN

Aus den Angriffen nehmen wir nun zwei Interessante heraus. Beim ersten Angriff handelt es sich um einen Versuch per FTP Zutritt zu unserem System zu verschaffen:

```
FTP connection
Host  : 203.164.95.115  (co3022884-a.rivrw1.nsw.optushome.com.au)
Login : anonymous
Pass  : guest@here.com
Time  : Tue Jul 24 05:20:31 2001

Log :
Client connecting: 203.164.95.115
Client tries anonymous Login
--->331 Guest login ok, send your complete e-mail address as password.
Client sent PASS 'guest@here.com'
--->230 User anonymous logged in.
Client wants to change current directory to pub/
--->200 CWD command successful.
Command not understood
--->502 'MKD 010724132628p': command not understood.
Client wants to change current directory to public/
--->200 CWD command successful.
Command not understood
--->502 'MKD 010724132630p': command not understood.
Client wants to change current directory to pub/incoming/
--->200 CWD command successful.
Command not understood
--->502 'MKD 010724132631p': command not understood.
Client wants to change current directory to incoming/
--->200 CWD command successful.
Command not understood
--->502 'MKD 010724132633p': command not understood.
Client wants to change current directory to _vti_pvt/
--->200 CWD command successful.
Command not understood
--->502 'MKD 010724132635p': command not understood.
Client wants to change current directory to
--->200 CWD command successful.
Command not understood
--->502 'MKD 010724132637p': command not understood.
Client wants to change current directory to upload/
--->200 CWD command successful.
Command not understood
--->502 'MKD 010724132639p': command not understood.
Client wants to change current directory to inbound/
--->200 CWD command successful.
Command not understood
--->502 'MKD 010724132640p': command not understood.
Client wants to change current directory to _vrt_txt/
--->200 CWD command successful.
Command not understood
--->502 'MKD 010724132642p': command not understood.
Client wants to change current directory to tagged/
--->200 CWD command successful.
Command not understood
--->502 'MKD 010724132644p': command not understood.
Client wants to change current directory to  /
--->200 CWD command successful.
Command not understood
--->502 'MKD 010724132646p': command not understood.
Connection timed out
Closing connection with 203.164.95.115
```

Und hier nochmals ein Versuch:

```
FTP connection
Host  : 193.159.131.115  (pC19F8373.dip.t-dialin.net)
Login : anonymous
Pass  : guest@here.com
Time  : Tue Jul 31 22:01:13 2001

Log :
Client connecting: 193.159.131.115
Client tries anonymous Login
--->331 Guest login ok, send your complete e-mail address as password.
Client sent PASS 'guest@here.com'
--->230 User anonymous logged in.
Client wants to change current directory to pub/
--->200 CWD command successful.
Command not understood
--->502 'MKD 010731215905p': command not understood.
Client wants to change current directory to public/
--->200 CWD command successful.
Command not understood
--->502 'MKD 010731215906p': command not understood.
Client wants to change current directory to pub/incoming/
--->200 CWD command successful.
Command not understood
--->502 'MKD 010731215908p': command not understood.
Client wants to change current directory to incoming/
--->200 CWD command successful.
Command not understood
--->502 'MKD 010731215909p': command not understood.
Client wants to change current directory to _vti_pvt/
--->200 CWD command successful.
Command not understood
--->502 'MKD 010731215909p': command not understood.
Client wants to change current directory to
--->200 CWD command successful.
Command not understood
--->502 'MKD 010731215910p': command not understood.
Client wants to change current directory to upload/
--->200 CWD command successful.
Command not understood
--->502 'MKD 010731215910p': command not understood.
Connection timed out
Closing connection with 193.159.131.115
```

Bei beiden Versuchen sehen wir, wie der „Angreifer“ sich einloggt und anschliessend in diverse Unterverzeichnisse geht, um dort eigene Ordner anzulegen. Jedoch das lässt Specter nicht zu.

8.2.2 Auswertung der Angriffe von Martin Lechmann (23.07. 2001)

Specter zeichnete von 15 Uhr bis um 21 Uhr genau 72 Angriffe auf.

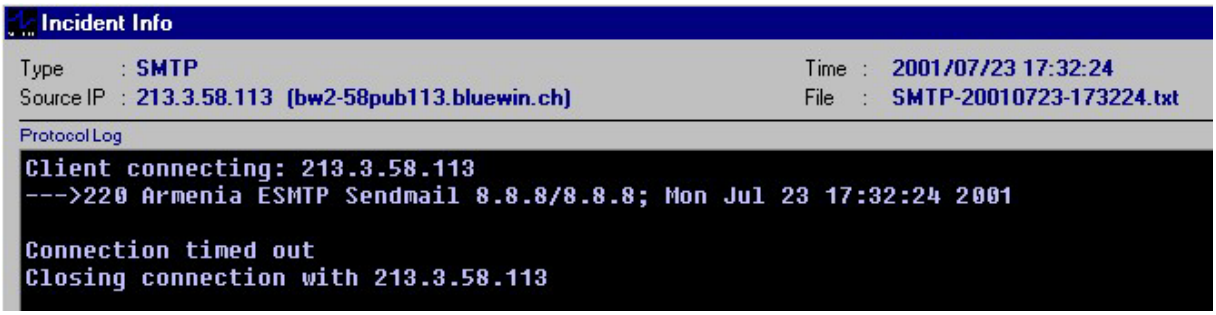
Sortiert nach Source-IP:

IP	Whois	Anzahl Angriffe
153.33.32.222	ts_mod2.ltx-tr.com	1
195.186.169.134	bw1-169pub134.bluewin.ch	35
213.3.189.239	bw2-189pub239.bluewin.ch	13
213.3.58.113	bw2-58pub113.bluewin.ch	23

Sortiert nach Angriffstyp:

IP	Type	Anzahl
153.33.32.222	FTP	1
195.186.169.134	BO2K	1
	DNS	2
	Finger	2
	FTP	2
	HTTP	17
	IMAP4	2
	NETBUS	1
	POP3	2
	SMTP	2
	SUB-7	1
	SUN-RPC	1
	TELNET	2
213.3.189.239	DNS	1
	FINGER	1
	FTP	1
	HTTP	6
	POP3	1
	SMTP	2
	SUN-RPC	1
213.3.58.113	DNS	1
	FINGER	1
	FTP	5
	HTTP	10
	IMAP4	1
	NETBUS	1
	POP3	1
	SMTP	1
	SUN-RPC	1
	TELNET	1

Das sind die Ergebnisse die Specter von den Angriffen von Martin Lechmann aufgezeichnet hat. Bei diesen Angriffen fällt etwas auf. Und zwar wird immer nur eine Verbindung aufgebaut und sofort geschlossen. Ohne das jemals ein Passwort oder Loginname übermittelt wurde. Hier ein Beispiel einer SMTP Verbindung:



Aus dieser Tatsache lässt sich schliessen, dass für die Angriffe wahrscheinlich automatisierte Tools verwendet wurden, um zu prüfen welche Dienste / Ports offen sind und welche nicht. Was für Programme verwendet wurden, lässt sich mit diesen Daten nicht sicher bestimmen.

8.3 Auswertung CyberCop Sting

8.3.1 Auswertung über die ganze Testdauer

CyberCop Sting schrieb während des ganzen Testes insgesamt über 48'800 Zeilen Logdaten auf die Festplatte!!! Davon waren aber 24'000 Einträge aus dem Telecom Labor intern gekommen (DNS, Proxy, ...). Da CyberCop String nicht selbständig die zur IP-Adresse gehörende Name (DNS) herausfindet, mussten wir dort Hand anlegen. Hier nun eine Tabelle der Angriffe mit IP-Adressen, welche mindestens vier mal in den Logdateien vorkamen:

Source	Anzahl	Whois	Nationalität
213.3.189.239	14914	The Blue Window	CH
195.186.169.134	7239	Swisscom IP-Plus	CH
195.186.0.130	533	Swisscom IP-Plus	CH
194.64.52.80	201	Focus Online GmbH	DE
195.186.0.131	139	The Blue Window	CH
195.186.0.33	56	The Blue Window	CH
198.186.203.20	29	Dandelion Digital	US
195.186.1.67	24	The Blue Window	CH
194.230.0.8	22	PLUSNET AG (Sunrise)	CH
4.24.153.165	19	BBN Planet	US
146.188.32.21	14	UUNET PIPEX	GB
216.32.74.52	14	Exodus Commnications Inc.	US
130.59.33.130	13	SWITCH Teleinformatics Services	CH
206.24.194.62	13	Cable & Wireless USA	US
62.154.17.113	13	Deutsche Telekom AG	DE
130.59.33.249	12	SWITCH Teleinformatics Services	CH
130.59.34.109	12	SWITCH Teleinformatics Services	CH

193.148.15.94	12	Amsterdam Internet Exchange	NL
195.16.167.26	12	GTE Internetworking	GB
195.16.175.205	12	Genuity Europe Aggregate	EU
195.16.175.246	12	Genuity Europe Aggregate	EU
212.1.192.169	12	TEN-155 European Backbone	GB
212.1.192.42	12	TEN-155 European Backbone	GB
212.1.193.145	12	TEN-155 European Backbone	GB
212.1.194.2	12	TEN-155 European Backbone	GB
212.133.10.38	12	Genuity Europe Aggregate	EU
212.133.10.57	12	Genuity Europe Aggregate	EU
64.58.76.227	10	Exodus Commnications Inc.	US
194.64.0.105	9	MAZ Hamburg	DE
195.179.51.102	9	IS Internet Services GmbH & Co	DE
195.206.65.50	9	European Backbone	NL
194.64.0.101	8	MAZ Hamburg	DE
212.47.109.228	8	OMNITEL Net	LT
62.2.63.27	8	Cablecom TV Provider	CH
130.59.1.30	6	SWITCH Teleinformatics Services	CH
193.253.185.167	6	France Telecom IP2000 ADSL BAS	FR
194.163.251.75	6	IS Internet Services GmbH + Co	DE
194.163.251.76	6	IS Internet Services GmbH + Co	DE
194.230.171.94	6	sunrise, Switzerland	CH
209.217.125.238	6	Magma Communications Ltd.	US
216.32.74.51	6	Exodus Commnications Inc.	US
217.12.6.16	6	Yahoo! Europe	GB
64.58.76.226	6	Exodus Communications Inc.	US
145.41.7.53	5	SURFnet IP carrier network	NL
212.74.67.194	5	COLT Backbone addresses in FR and NL	GB
213.220.0.250	5	EncomIX InterComputer-Soft	ES
64.58.76.176	5	Exodus Communications Inc.	US
64.58.76.223	5	Exodus Communications Inc.	US
193.159.131.115	4	Deutsche Telekom AG	DE
200.188.80.30	4	Comite Gestor da Internet no Brasil	BR
212.26.18.129	4	Internet Service Unit, KACST	SA

Was das für Angriffe waren, kann man aber mit den Logdaten von CyberCop Sting nicht sagen.

8.3.2 Auswertung der Angriffe von Martin Lechmann (23.07. 2001)

CyberCop Sting erkannte nur zwei IP-Adressen von Martin Lechmann. Dies waren:

Source	Anzahl	Whois	Nationalität
213.3.189.239	14914	The Blue Window	CH
195.186.169.134	7239	Swisscom IP-Plus	CH

Aber leider können anhand der Aufzeichnungen von CyberCop Sting keine weitere Aussagen über diese Angriffe getätigt werden.

8.4 Vergleich der Resultat mit der Auswertung vom Martin Lechmann

Als Martin Lechmann seine Angriffe machte, führte er ein genaues Zeitdiagramm. Das heisst, der schrieb immer auf, mit welchem Tool er welchen Rechner überprüfte. Diese Dokumentation schickte er uns, um einen Vergleich machen zu können. Wir werden aber hier keinen vollständigen Vergleich machen, da dies den Umfang dieser Arbeit bei weitem überschreiten würde. Wir werden aber ein Beispiele herausgreifen.

Es handelt sich um den von Specter simulierten Netbus Server. Aus dem Bericht von CyberCop Scanner geht hervor, dass auf dem Rechner mit der IP-Adresse 195.176.242.220 ein Netbus Server laufe. Wir konnten dabei mit Specter genau erkennen, dass Martin auf den offenen Port 12345 stiess. Dieser Port ist gewöhnlicherweise offen, wenn der Netbus Trojaner auf dem Rechner aktiv ist.
Snort hingegen erkannte den Angriff als „DDoS-Stacheldraht Client“. Normal ist bei einem Rechner mit einem DDoS-Stacheldraht Client der Port 16772 offen, kann aber auch auf diversen anderen Ports sein. Somit kann nicht bestimmt gesagt werden, ob jetzt Snort oder Specter recht haben. Man müsste überprüfen, wie das Programm CyberCop Scanner diese Überprüfung durchgeführt hat, um zu erkennen, dass ein Netbus Server auf dem Rechner ist. Tatsache ist, das beide Programme den Angriff erkannt haben und auf ein ähnliches Resultat kommen. CyberCop Sting kann hier nicht mithalten. Mit dessen Logdaten kann ein solcher Angriff nicht erkannt werden.

8.5 Integrität der Systeme

Als der Test zu Ende war, hatten wir für jeden Rechner drei Ghost-Dateien:

1. Datei: Vor dem Test
2. Datei: Vor den Angriffen von Martin Lechmann
3. Datei: Nach dem Test

Diese drei Dateien mussten nun miteinander verglichen werden. Dazu verwendeten wir MD5summer [54]. Auf allen Stationen stellten wir keine ungewöhnlichen Veränderungen fest. Es wurden zwar diverse Dateien verändert, jedoch waren das nur Logdaten, Auslagerungsdateien oder ähnliche Files. Das wichtigere waren die Anwendungen (exe, com) und von diesen wurde kein einziges verändert.
Wir schliessen daraus, dass es niemandem gelungen ist, in unser System zu kommen und dieses zu verändern.

Schlusswort

Zum Schluss kann man sicher sagen, dass uns die Semesterarbeit wirklich gefallen hat und wir sehr viel Neues lernen konnten. Das Thema Internet Sicherheit finden wir sehr wichtig und interessant und darum wollen wir auch bei der Diplomarbeit diese eingeschlagene Richtung beibehalten.
Das Thema Honeynet oder Honeypot wird man in Zukunft vermehrt begegnen, denn es bietet eine sehr gute Möglichkeit Angreifer aufzuspüren und zu verfolgen. Dennoch muss gesagt werden, dass die Sicherheit eines Netzes nur so gut ist, wie die schwächste Stelle des gesamten Systems. Und diese Systeme werden in Zukunft sicher immer schwieriger zu überblicken sein, da fast jedes Gerät einen Internetzugang besitzen wird. Hier könnte man das Beispiel mit dem Kühlschrank nehmen, welcher automatisch die benötigten Lebensmittel über das Internet bestellen kann.
Wir hoffen mit dieser Semesterarbeit einen interessanten und wertvollen Beitrag zur Förderung des Sicherheitsdenkens für jede Leserin und Leser gegeben zu haben.
Nun möchten wir Ihnen noch alles Gute wünschen und dass Sie niemals selbst in einen Honeypot geraten!

Glossar

ACID	Analysis Console for Intrusion Databases
CERT	Computer Emergency Response Team
DDoS	Distributed Denial of Service Attacks
DNS	Domain Name System
DoS	Denial of Service Attacks
DOS	Disk Operating System
IDS	Intrusion Detection Systems
ISS	Internet Security Systems
MDAC	Microsoft Database Access Component
OS	Operating System (Betriebssystem)
RAS	Remote Access Service
RIPE	Réseaux IP Européens

Quellen- und Abbildungsverzeichnis

Literatur

[21] C't Magazin für Computertechnik, 8.Ausgabe 2001, Verlag Heinz Heise GmbH & Co Kg, Hannover.

[22] Norbert Pohlmann:Firewall-Systeme 3.Aktuelle und erweiterte Auflage MITP-Verlag, Bonn, 2000.

[23] B. Cajacob, R.Lanicca, *Sicherheit im Internet – Firewall und Intrusion Detection Systems*, Semesterarbeit an der HTW Chur, Abteilung Telecom, Wintersemester 2000/2001.

Internet

[31] Snort Homepage
http://www.snort.org

[32] Writing Snort Rules
http://www.snort.org/writing_snort_rules.htm

[33] Projekt Honeynet Homepage (Spitzner)
http://project.honeynet.org/

[34] IDS Center Homepage (Ueli Kistler)
http://www.diax.ch/users/rkistler/main.htm

[35] Swissonline Hack
http://www.pctip.ch/webnews/wn/18928.asp
http://www.sonntagszeitung.ch

[36] Neu Snort Homepage (alt: snort.org)
http://snort.sourcefire.com/

[37] Autoinstallations File:
http://snort.certworks.net/Files/autoinstall.bat

[38] Whitehats Page
http://www.whitehats.com/

[39] The Deception Toolkit (Fred Cohen Associates)
http://www.all.net/dtk/

[40] Specter (Netsec)
http://www.specter.com

[41] CyperCop Sting (PGP Security)
http://www.pgp.com/international/germany/products/cybercop-sting/default.asp

[42] ManTrap (Recourse Technologies)
http://www.recourse.com/products/mantrap/trap.html

[43] Sun HCL
http://soldc.sun.com/support/drivers/hcl/index.html

[44] Securiteam
http://www.securiteam.com/securitynews/6H00S1P0AA.html

[45] Omicron Electronics GmbH
http://www.omicron.ch

[46] Compass Security Network Computing AG
http://www.csnc.ch

[47] SecurityFocus Press Center
http://www.securityfocus.com

[48] SystemExperts
http://www.systemexperts.com

[49] SANS (System Administration, Networking, and Security) Institute
http://www.sans.org

[50] TripWire
http://www.tripwire.com

[51] SecuriTeam
http://www.sans.org

[52] Norton Ghost
http://www.symantec.com/ghost

[53] md5sum
http://www.etree.org/md5com.html
http://hegel.ittc.ukans.edu/topics/linux/man-pages/man1/md5sum.1.html

[54] md5summer
http://homepages.ihug.co.nz/~floydian/md5

Anhang

Das Interview mit dem Swissonline Hacker
Datum: 29.07.2001/19:05

Der Hacker, der das Leck bei Swissonline entdeckte, über seine Motive, die Schweizer Hackerszene und die Banken

VON OLIVER ZIHLMANN

Die SonntagsZeitung hat vor einer Woche ein Datenleck bei Swissonline publik gemacht. Einem Hacker war es gelungen, den Zugang zu 250 000 E-Mail-Konten zu knacken. Der Mann übergab der SonntagsZeitung eine CD-ROM mit 185 000 E-Mail-Adressen und demonstrierte den Zugriffsweg zu einer beliebigen Adresse. Noch in der Nacht auf letzten Sonntag bestätigte ein Sicherheitsfachmann von Swissonline das Datenleck. In der Nacht zum Donnerstag nahm Swissonline den Mailserver vom Netz, um «sicherheitsbedingte Wartungsarbeiten vorzunehmen», wie es hiess. Swissonline bedauerte gegenüber den Kundinnen und Kunden in einer Stellungnahme, dass es einem Hacker gelungen sei, Kundendaten aufzuspüren. «Selbstverständlich haben wir sofort entsprechende Massnahmen ergriffen und diesen aussergewöhnlichen Zugriffsweg versperrt», hiess es in der Stellungnahme weiter. Der Hacker, der anonym bleiben will, erklärt in einem Interview, wieso er mit seiner Entdeckung an die SonntagsZeitung gelangte.

SonntagsZeitung: Sie haben ein Datenleck bei Swissonline entdeckt und publik gemacht. Warum sind Sie nicht direkt zum Mailanbieter gegangen?

Hacker: Aus einer Art Solidarität mit denen, die nicht wussten, dass ihre Mails offen waren, und sich deshalb nicht wehren konnten. Ich wollte, dass das Loch behoben wird - natürlich. Aber wenn ich es nur dem Mailanbieter gesagt hätte, wäre es niemals publik geworden, hunderttausende Mailkonten wären offen gewesen, und niemand hätte es je gewusst. Oder glauben Sie, ein Mailanbieter würde so etwas von sich aus publik machen?

Sie hatten also Erfolg.

Hacker: Swissonline konnte nicht stillschweigend das Loch stopfen, sondern musste den Mailserver letzten Mittwoch neu aufsetzen.

Der Anbieter sagt, die Gefahr sei nicht so gross gewesen, weil nur ein Profi durch das Leck hätte einsteigen können.

Hacker: Sie hatten vergessene Ware auf ihren Computern, die praktisch jeder nutzen konnte, um einzudringen.

Vergessene Ware?

Hacker: Alte Skripts - Programmfetzen -, die nicht mehr in Gebrauch sind und vergessen wurden. Das ist, als ob man einen Universalschlüssel in der Ramschkiste

liegen lässt und später vergisst. Wer den Schlüssel findet, kommt in Minuten überall rein.

Wie kann ich mich als Kunde gegen Hackerattacken schützen?

Hacker: Gehen Sie auf die Seite www.pgp.com, und installieren Sie das Sicherheitsprogramm «Pretty Good Privacy». Das ist ein Weltstandard, der E-Mails absolut sicher verschlüsselt. Einfach, sicher, gratis.

Diese Lösung gibts schon lange. Offenbar ist den Menschen der Aufwand zu gross.

Hacker: Tja. Wenn ein Internet- oder Mailanbieter auf den Markt kommt, der wirklich Sicherheit verspricht, müsste er für sein System eine Jahresgebühr von 1000 Franken verlangen. Und sicher wäre er überhaupt nicht.

Warum?

Hacker: Wenn ein Anbieter absolute Sicherheit verspricht, wird er sofort zum bevorzugten Ziel der Hacker. Deswegen läuft jeder, der sich Sicherheit auf die Fahnen schreibt, erst recht Gefahr, gehackt zu werden.

Wann lohnt es sich für einen Hacker überhaupt, einen Mailanbieter anzugreifen?

Hacker: Ein Profihacker wird versuchen, sich auf einem Mailprovider sauber einzunisten. Dann benutzt er dessen Rechner als Ausgangsbasis für Hacks in gefährlichere Systeme. Vom Rechner eines gehackten Mailanbieters aus könnte man bequem einen Angriff starten - etwa auf eine Grossbank.

Sind denn schon viele Systeme so infiltriert?

Hacker: Ich kann mir gut vorstellen, dass es viele Kisten gibt, die längst gehackt wurden, ohne dass es bekannt ist. Es gibt viele in der Szene, die darauf angewiesen sind, völlig anonym zu agieren. Als Hacker hast du einfach deine Örtchen, die du für dich behältst.

Das scheint wie eine Sucht zu sein.

Hacker: Hacker und Geheimdienste, die gezielt sensible Daten für Polit- und Industriespionage besorgen, gehen über unzählige Stationen, um ihren Weg zu verschleiern. Die grossen Hacks - wie beim Pentagon - waren monatelange Übungen. Die Hacker waren 15 Stunden pro Tag am Werk. Eine Menge anderer Systeme, die sie als Vorstationen gebraucht hatten, mussten dran glauben. Ein Mailprovider eines neutralen Landes wie der Schweiz wäre hervorragend für so was.

Wie beschreiben Sie die Hackerszene?

Hacker: Es gibt zwei Sorten. Die Profis mit den ausgetüftelten Hacks sind meistens Fachstudenten, die das Knowhow haben. Sie beschäftigen sich systematisch mit Netzwerken - Ingenieure, Elektrotechniker, Informatiker. Andere surfen sich ihr Wissen zusammen. Sie laden sich Hackerprogrämmchen von Profis herunter und legen los. Wenn sie einen Treffer haben, jubeln sie, aber sie bringen keine

Eigenleistung.

Wie sicher sind Banken vor Hacks?

Hacker: Jede Bank hat Firewallschleusen, die funktionieren wie der Eingang einer italienischen Bank: Nach dem Einloggen gehen hinter einem die elektronischen Türen zu; erst dann gehen die inneren Pforten auf. Drinnen fangen die Sicherheitsmassnahmen richtig an. Eine grosse Schweizer Bank zum Beispiel speichert alle Kontendaten auf einem einzigen Mainframe, einem Grosscomputer, einer Hochburg. Um den Frame gruppiert sich eine riesengrosse Landschaft von Systemen. Der Mainframe ist selbst innerhalb des Firmennetzes so abgeschottet, dass sogar für die wenigen zutrittsberechtigten Bankangestellten die Identifikation mühsam ist. So, dass sie jeden Kontakt mit dem System scheuen.

Also kommt man in Banken nicht rein?

Hacker: Stellen Sie sich vor, Sie hätten alle Telefonnummern der Bank mit dem Modem angewählt, hätten alle Leitungen getestet, fänden eine direkte, hackbare Linie und hätten tatsächlich Zugriff auf Kundenkonti. Ab jetzt wird jede Ihrer Mausbewegungen kontrolliert. Sie müssten jetzt viel Geld in der ganzen Welt herumschieben, und eine Sekunde nach dem Hack müsste jemand auf den Bahamas das Geld abheben. Probieren Sie das mal. Oft gibt es systemeigene Grenzen, die das verhindern. Bei einer bekannten Bank zum Beispiel macht der Computer ab 50 000 Franken immer eine Rückfrage. Das System selber verlangt, das man nochmals anruft. Es ist ein grundlegender Unterschied zu einem Mailanbieter.

Nämlich?

Hacker: Bei der Bank gehts um Sein oder Nichtsein. Angenommen, ein Hacker könnte sämtliche Server einer grossen Bank löschen: Es wäre nicht mehr möglich, innert nützlicher Frist aus Papierbelegen alle Konti und Geldflüsse zu rekonstruieren. Die Geschwindigkeit der elektronischen Buchgeldverschiebung und der Aktiensturz durch den Vertrauensverlust der Analysten in eine solche Bank würden sie innert Stunden in den Konkurs treiben. Milliarden an Geldwert würden vernichtet. Alle Kontoinhaber gingen leer aus. Eine Bank muss sich vor Hackern schützen, nicht weil ihr Geld gestohlen werden könnte, sondern weil sie durch einen Angriff auf ihren Computer vernichtet werden kann. Deshalb gilt für Bankensoftware das Gleiche wie für Software der Raumfahrtindustrie: Sie ist bombensicher.

Was geschähe, wenn ein Staat von einer koordinierten Hackerattacke aus dem Ausland angegriffen würde?

Hacker: Es gäbe ernsthafte Schäden. In erster Linie an der Infrastruktur. In der Schweiz würde man als Erstes das Kernforschungszentrum Cern angreifen.

Warum?

Hacker: Weil 90 Prozent des ganzen Internetverkehrs der Schweiz über diese Server laufen. Wenn sie flach lägen, wäre die ganze Schweiz vom Netz.

Wie schützt sich ein Staat vor dem Informationskrieg?

Hacker: Das Hacken lebenswichtiger Computersysteme ist die grosse Gefahr unserer vernetzten Welt. Das müssen wir in den Griff kriegen. Als Erstes sollten wir konsequent alle lebenswichtigen Systeme physisch vom Netz nehmen. Es darf buchstäblich kein Kabel in solche Systeme führen. Es gibt in der Schweiz Zentralleitsysteme von Krankenhäusern, die immer noch am Internet hängen. Da werden Dosierungen von gefährlichen Medikamenten an Patienten gespeichert. Die Hacker sind dabei wie Seismografen: Sie warnen vor Anfälligkeiten der global vernetzten Welt.

Quelle: http://www.sonntagszeitung.ch/