Snort

IIN Gruppenarbeit

Lukas Reusser, Christian Hauser

Inhaltsverzeichnis

1 Einführung

In dieser kurzen Einführung werden wir kurz unser IIN Projekt erläutern, sowie auf unsere Netzwerkkonfiguration eingehen.

1.1 Beschreibung des Praktikums

Dieses Dokument ist unsere Dokumentation zum 2. IIN Praktikum. Bei diesem Praktikum ging es um Realisierung eines Netzwerk basierenden Intrusion Detection System. Wir verwendeten Snort, ACID und Snortreport in den neusten zur Verfügung stehenden Versionen.

1.2 TCP/IP Konfiguration

Nachfolgend eine Übersicht über unser Netzwerk, das wir für das Praktikum benutzt haben.

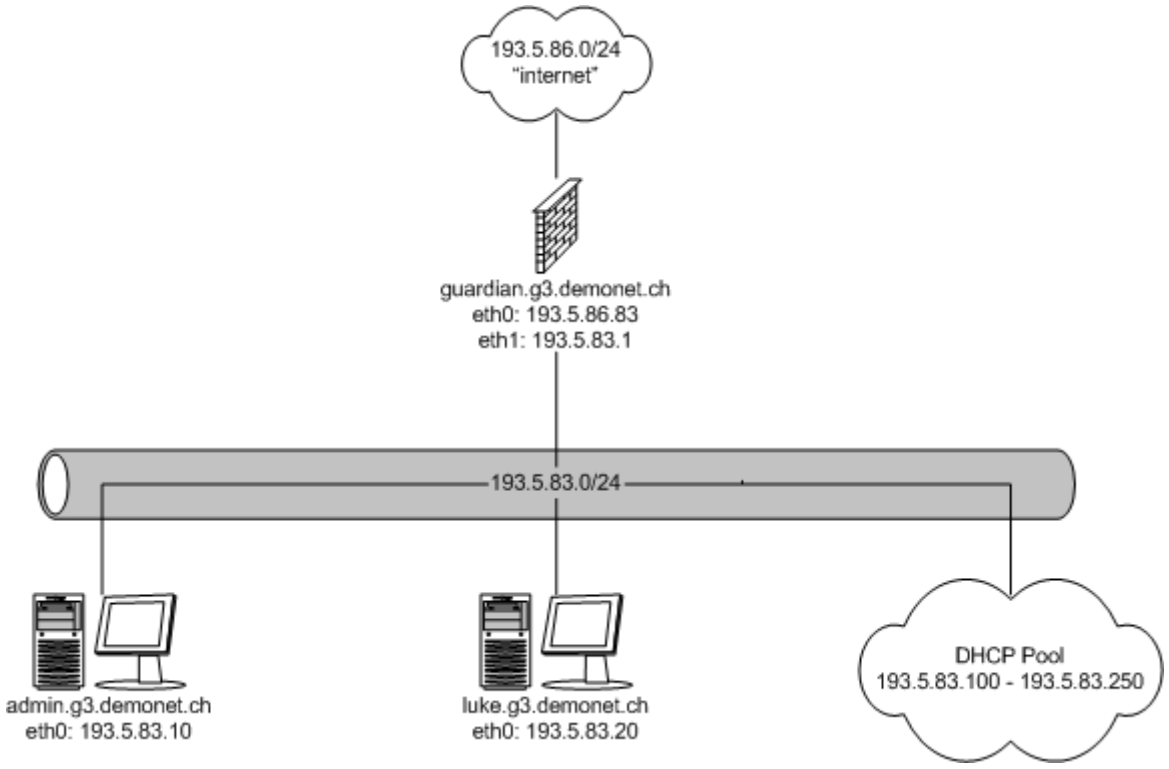


Abbildung 1: Das obige Bild zeigt den Rechner *guardian*, der als Internet-Router, Firewall und Snort-Sensor aufgesetzt wurde. Am Netzwerk angeschlossen sind noch die Rechner *admin* (DNS, DHCP, Web Server, MySQL, ACID, Snortreport) und *luke* (Client).

1.3 Verwendete Betriebssysteme

Auf allen Rechnern ist Linux (RedHat Fedora Core 1) installiert.

1.3.1 Update Hinweis zu Fedora

Die RedHat-Server welche die Updates von Fedora beherbergen, sind meistens überlastet und darum funktioniert up2date sehr langsam oder auch überhaupt nicht. Um diesen Flaschenhals zu umgehen, empfiehlt es sich, up2date so zu konfigurieren, dass es einen Mirror-Server verwendet:

[root@host]# vi /etc/yum.conf

```
[main]
cachedir=/var/cache/yum
debuglevel=2
logfile=/var/log/yum.log
pkgpolicy=newest
distroverpkg=fedora-release
tolerant=1
exactarch=1

[base]
name=Fedora Core $releasever - $basearch - Base
baseurl=http://fedora.redhat.com/releases/fedora-core-$releasever

[updates-released]
name=Fedora Core $releasever - $basearch - Released Updates
#baseurl=http://fedora.redhat.com/updates/released/fedora-core-$releasever
baseurl=ftp://sunsite.cnlab-
switch.ch/mirror/fedora/linux/core/updates/$releasever
```

[root@host]# vi /etc/sysconfig/rhn/sources

```
yum fedora-core-1 http://fedora.redhat.com/releases/fedora-core-1
yum updates-released ftp://sunsite.cnlab-
switch.ch/mirror/fedora/linux/core/updates/1/i386
```

2 Intrusion Detection Systeme

Dieses Kapitel ist eine allgemeine Einführung in Intrusion Detection Systeme. Wir erklären was für eine Funktion Intrusion Detection Systeme innerhalb eines Netzwerks erfüllen, nämlich dass sie eine Art Alarmanlage gegen Eindringlinge darstellen. Im nächsten Kapitel gehen wir dann auf die Installation von Snort ein. Snort ist ein Open-Source Intrusion Detection System.

2.1 Einleitung

Bei einem Intrusion Detection System (IDS) handelt es sich um ein System, das in der Lage ist, Benutzer zu entdecken, die sich auffällig verhalten. Man kann sich das IDS als Alarmanlage des Netzwerks vorstellen. Doch im Vergleich zu einer Alarmanlage zu Hause, muss das IDS zwischen zulässigem und unerwünschtem Datenverkehr unterscheiden können.

2.2 Wie funktioniert ein IDS?

Ein IDS kann entweder eine Software- oder Hardware-Lösung sein, die den Zweck hat, unautorisierte Benutzung eines Computersystems oder Netzwerks oder Angriffe darauf festzustellen. Das IDS hält nach unautorisierten Versuchen Ausschau, sich Zugang zu einem System zu verschaffen, zulässige Rechte auf einem autorisierten System zu überschreiten oder die Verfügbarkeit eines Systems zu verringern, sei es von innerhalb der Organisa-

tion aus oder über das Internet. Es gilt dabei zu beachten, dass ein IDS nur ein einzelnes Element in einer Sicherheitsstrategie aus miteinander verknüpften und sich überlappenden Segmenten ist.

IDS-Systeme gibt es in einer Vielzahl von Varianten, mit unterschiedlichen Überwachungs- und Analysemethoden für die verfügbaren Daten. IDS-Systeme überwachen Ereignisse auf drei verschiedenen Stufen: Netzwerk, Host und Anwendung. Sie können diese Ereignisse mithilfe zweier Techniken analysieren, durch Erkennen von Signaturen oder von Anomalien. Aktive IDS-Systeme verfügen zusätzlich über die Fähigkeit, Gegenmassnahmen bei einer Attacke zu ergreifen, wobei man sich das sehr gut überlegen sollte (siehe “Reaktion auf Angriffe” weiter unten).

2.3 Erkennungsmethoden

Wie oben bereits erwähnt gibt es zwei Techniken um die Ereignisse zu analysieren. Von diesen beiden Erkennungsmethoden wird Signatur Detection in kommerziellen IDS-Produkten am häufigsten verwendet. Anomaly Detection ist zwar jünger, wird aber immer stärker kommen.

2.3.1 Signatur-basierte Erkennung

Signatur-basierte Erkennung achtet auf Aktivitäten, die mit einer festgelegten Zeichenfolge übereinstimmen, welche eindeutig eine bekannte Angriffsmethode beschreibt. Somit müssen Signatur-basierte IDS-Systeme für jede bekannte Angriffsmethode programmiert werden. Diese Technik ist zwar äusserst effektiv gegenüber bekannten Angriffsmethoden, erfordert aber laufende Aktualisierungen, um auch gegen neuartige Attacken gewappnet zu sein.

2.3.2 Anomalie-basierte Erkennung

Anomalie-basierte IDS-Systeme erkennen einen Eindringling an seinem unüblichen Verhalten im geschützten System oder Netzwerk (Anomalien). Sie beruhen darauf, dass sich das Verhalten von normalen Anwendern von dem von Angreifern unterscheidet und dieses deshalb bis zu einem gewissen Grad entsprechend zugeordnet werden kann. Der ursprüngliche Normalzustand muss zuerst gemessen werden, indem man die Arbeitsmuster und Bandbreite der normalen Benutzung beobachtet. Die Überwachung erfolgt, indem man diese Werte kontinuierlich mit dem aktuellen Datenverkehr vergleicht. Ein generelles Problem von Anomalie-basierten IDS-Systemen besteht darin, dass ein "durchschnittlicher" Workload fast unmöglich festzustellen ist. Auf der anderen Seite bieten Anomalie-basierte IDS-Systeme die Möglichkeit, bislang unbekannte Attacken zu erkennen. Einige Signatur-basierte IDS-Systeme umfassen auch in begrenztem Umfang Funktionen zur Anomaly Detection (Snort zum Beispiel), aber nur wenige verlassen sich allein auf diese Technologie.

2.4 Welche Arten von IDS gibt es?

IDS-Systeme werden in der Regel danach unterschieden, was sie überwachen: das gesamte Netzwerk, einen bestimmten Host oder sogar nur eine einzelne Anwendung. Ein wirklich effektives IDS wird eine Kombination aus Netzwerk- und Host-basierter Intrusion Detection verwenden. Herauszufinden, wo man welche Art von IDS einsetzt und wie man die Daten integriert, ist eine entscheidende Frage, die immer wichtiger wird.

2.4.1 Netzwerk-basiert IDS-Systeme

Die meisten IDS-Systeme auf dem Markt basieren auf Network IDS-Systemen (NIDS). NIDS sammeln Daten an einem oder mehreren wichtigen Punkten im Netzwerk und schicken entsprechende Berichte an eine Verwaltungs-Konsole. Die Datensammelsysteme müssen im Netzwerk platziert sein, so dass sie allen durchfliessenden Traffic beobachten können. In einem vollständig geswitchten Netzwerk kann es unter Umständen schwierig sein, alle Daten einzufangen, es sei denn, man konfiguriert die Switches so, dass sie eine Kopie des gesamten Datenverkehrs an einen speziellen Port für das IDS schicken.

Vorteile

- Man kann ein recht grosses Netzwerk mit nur wenigen Rechnern beobachten.
- Das System ist transparent, da das Gerät Traffic-Informationen sammelt.
- Der gesamte Traffic zwischen der Konsole und dem NIDS-Collector kann verschlüsselt werden oder für vollständige Sicherheit über ein separates Netzwerk laufen.

Nachteile

- Es kann im System eine grosse Menge an Traffic geben, womöglich mehr, als das System verarbeiten kann. Dies erschwert das Entdecken von Eindringlingen, wenn die Auslastung hoch ist.
- Die Notwendigkeit, Datenpakete in kürzester Zeit zu verarbeiten, kann dazu führen, dass man einige Funktionen nicht nutzen kann, wenn das System mit der Menge an Traffic Schritt halten soll.
- Vollständig geswitchte Netzwerke können schwierig zu überwachen sein, da nicht wie bei nicht-geswitchten Netzwerken der gesamte Traffic über alle Ports repliziert wird.
- Verschlüsselter Traffic kann nicht analysiert werden.

2.4.2 Host-basierte IDS-Systeme

Ein Host-basiertes IDS (HIDS) beobachtet, was auf dem Computer passiert, auf dem es installiert ist. Dies erlaubt es dem IDS, mit Hilfe der Logdateien (und/oder internen Auditing-Systemen) sehr genau hinzuschauen, was auf diesem Rechner vor sich geht. Es gibt zwei Hauptarten von HIDS:

- Host Wrapper und Personal Firewalls
- Agenten-basierte Software

Host Wrapper oder Personal Firewalls sind so konfiguriert, dass sie alle Netzwerkpakete, Verbindungsversuche oder Login-Versuche beobachten, die den überwachten Rechner erreichen. Host-basierte Agenten verfolgen den Zugriff auf wichtige Systemdateien und Veränderungen an diesen sowie Änderungen an Benutzerberechtigungen.

Idealerweise vereinfacht das HIDS die Verwaltung mehrerer Hosts, indem die Administrationsfunktionen und Logdateien über Angriffe alle an eine zentrale Maschine geleitet werden.

Vorteile

- Erkennt eine Vielzahl lokaler Attacken.
- Verschlüsselung stellt normalerweise kein Problem dar, wenn die Daten auf dem Server entschlüsselt werden.
- Kein Problem mit geswitchten Netzwerken.

Nachteile

- Häufig muss jeder Host separat installiert und gewartet werden.
- Da sich das IDS auf dem Host befindet, kann auch das IDS attackiert und lahmgelegt werden.
- Erkennt unter Umständen keinen weit gestreuten Netzwerk-Scan.
- Kann mit einer DoS-Attacke (Denial of Service) überschwemmt werden.
- Beansprucht Rechenleistung und Netzwerkressourcen des zu schützenden Servers.

2.5 Systeme immer sofort patchen

Es gibt Tausende von Methoden, mit denen man sich unerlaubten Zugriff auf Computer verschaffen kann, und jeden Monat tauchen Dutzende neue auf, von Buffer Overflows und Directory Traversal Exploits bis zu Denial of Service (DoS) und Distributed Denial of Service (DDoS) Attacken.

Theoretisch sollten alle Systeme im Fall bekannter Sicherheitslücken gepatcht oder mit entsprechenden Workarounds versehen sein, so dass keine Notwendigkeit mehr für ein Signatur-basiertes IDS bestehen dürfte. Leider sieht die Wirklichkeit so aus, dass viele System nicht oder nicht gleich gepatcht (oder aktualisiert) werden, sobald eine Sicherheitslücke entdeckt wird. Dies wird durch die Zahl der Systeme deutlich, die jeden Tag beeinträchtigt werden, und die Tatsache, dass es sich bei den meisten der Probleme um recht alte und gut bekannte Probleme handelt, für die Fixes schon längst verfügbar sind.

2.6 Reaktion auf Angriffe

Die meisten IDS-Maschinen sind so konfiguriert, dass sie alle aufgezeichneten Daten in einer Datenbank speichern und eine E-Mail an den zuständigen Administrator schicken, sobald ein Angriff festgestellt wird. Das Problem dabei ist, dass gewisse Angriffe (von z.B. Würmern wie “Nimda” oder “Code Red”) leicht dazu führen können, dass ein Administrator mit E-Mails überschwemmt wird. Das Resultat ist ein System, das zu viele Informationen liefert, die nicht alle verarbeitet werden können. Eine praktikable Möglichkeit zur Vorbeugung besteht darin, die Schwelle für eine Benachrichtigung per E-Mail oder Pager möglichst hoch anzusetzen, aber auch die Konsole ein Alarmsignal von sich geben zu lassen, wenn sie ein Problem feststellt. Auf diese Weise hat der Administrator die auffälligen Probleme im Blick, bemerkt aber auch ernste Schwierigkeiten, wenn das System immer mehr Alarmsignale aussendet.

Falls der Administrator nicht immer zur Stelle ist oder es Grund für erhöhte Alarmbereitschaft gibt, können einige IDS-Systeme so konfiguriert werden, dass sie automatisch auf Angriffe reagieren. Dies kann wie oben beschrieben eine einfache Benachrichtigung per E-Mail oder an einen Pager sein, aber auch aktivere Massnahmen umfassen, um einen akuten Angriff zu stoppen und dann die Schwachstelle abzudichten.

Das direkte Eingreifen zur Unterbrechung der Kommunikation zwischen einem Angreifer und dem Opfer wird oft als Session Sniping oder Knockdown bezeichnet und erfolgt durch das Einfügen von Paketen, um die Verbindung zu unterbrechen, welche die Response ausgelöst hat. Die effektivste Methode, eine TCP-Verbindung abzubrechen, ist ein Reset der Verbindung mithilfe gefälschter Pakete. Hierzu muss das IDS gefälschte Pakete an eines oder beide Systeme schicken, bei denen das TCP Reset-Bit gesetzt ist.

Andere Interventionsmethoden umfassen die Neukonfigurierung der Perimeter-Router und Firewalls, um die IP-Adresse des Angreifers zu blockieren, oder das Blockieren der Protokolle, die für den Angriff verwendet werden. In hartnäckigen Fällen kann es besser sein, jegliche Kommunikation zu dem angegriffenen System zu unterbrechen, anstatt es der Gefahr einer Beschädigung auszusetzen. Weitere Reaktionsmöglichkeiten sind: Aktiv Informationen über den Host oder die Website des Angreifers herauszufinden oder sogar eine Gegenattacke zu starten. Bevor man jedoch solche Funktionen aktiviert, sollte man sich aber vorher juristisch absichern.

2.7 IDS als Teil einer Gesamt-Sicherheits-Architektur

IDS-Systeme werden von vielen Organisationen eingesetzt, egal ob gross oder klein, um ihre Netzwerke und Systeme zu schützen. Kein Unternehmen, das Sicherheit ernst nimmt, sollte auf ein IDS als Teil seines Sicherheitssystems verzichten.

Insgesamt erfüllen IDS-Systeme ihren Zweck recht gut, aber oft sind sie nicht schnell genug und bemerken einen Angriff erst, wenn es schon zu spät ist, so dass auch das Schliessen des Einfallstores nichts mehr nützt. Die Implementierung eines IDS als Schicht in einer aus mehreren Schichten bestehenden Gesamt-Sicherheits-Architektur (Firewalls, Zugangskontrollen und Authentifizierungsmechanismen, Überwachungs-Tools, Vulnerability Scanning Tools, IDS-Systeme und Sicherheitsschulungen) erschwert Angriffe von externen Eindringlingen und erleichtert die Prävention vor Angriffen und deren Erkennung.

Intrusion Detection ist erforderlich, da Firewalls in der Praxis zwar einen guten, jedoch keinen vollständigen Schutz vor Eindringlingen bieten können. Die Erfahrung lehrt, dass man sich nie auf nur eine Verteidigungsstrategie oder -technik verlassen sollte. Eine Firewall dient üblicherweise als effektiver Filter, der viele Angriffe abblockt, bevor sie das Netzwerk einer Organisation erreichen können. Allerdings sind Firewalls anfällig für Konfigurationsfehler und uneinheitliche oder nicht festgelegte Sicherheitsrichtlinien. In der Regel bieten sie keinen Schutz vor bösartigem Code, der per E-Mail verschickt wird, oder Insider-Attacken und ungesicherten internen Netzwerken und Schnittstellen. Firewalls verlassen sich darauf, dass es einen zentralen Punkt gibt, über den der gesamte Traffic fliesst.

2.8 Snort

Nachdem wir nun das Prinzip von Intrusion Detection Systemen angeschaut haben kommen wir zu Snort, demjenigen IDS, das wir für unsere Gruppenarbeit eingesetzt haben.

Snort ist ein Netzwerk IDS und wird gemeinhin als Signatur-basiertes IDS angeschaut. Aber Snort kann auch Protokoll-Anomalien mittels der Regeln-basierten Engine, dem Decoder und den Präprozessoren erkennen. Somit ist Snort auch ein Anomalie-basiertes IDS. Snort besteht also im Wesentlichen aus vier Basiskomponenten: Dem Sniffer, dem Präprozessor, der Detection-Engine und der Ausgabe. Der Sniffer leitet alle Pakete von der Netzwerkkarte zum Präprozessor weiter. Beim Präprozessor angekommen, werden die Daten vorsortiert. Man kann also beim Präprozessor die Daten durch verschiedene Plugins jagen, um zu entscheiden welche Daten überhaupt interessant sind um an die Detection-Engine weitergeleitet zu werden. Die Detection-Engine testet dann die ankommenden Daten anhand von Regeln. Snort bringt selber schon eine sehr grosse Anzahl von Regeln mit, es ist aber auch ziemlich einfach sich selber Regeln zu erstellen. Trifft eine Regel zu, wird das ganze geloggt oder auch ein Alarm generiert. Da man bei einem Alarm auch ein Shellscript starten kann, ist man wirklich frei was man bei einem Alarm machen will.

2.8.1 ACID - Analysis Console for Intrusion Databases

Snort generiert je nach Konfiguration eine wirklich grosse Anzahl von Events. Um diese Datenmenge besser analysieren zu können, bietet sich zum Beispiel ACID an. ACID ist in php geschrieben und greift via adodb auf eine Datenbank zu. Es bereitet die gesammelten Daten auf und bietet eine vielzahl von Funktionen um die Analyse zu vereinfachen.

2.8.2 Snortreport

Snortreport ist ebenfalls ein Tool um die Datenanalyse zu vereinfachen. Unserer Meinung nach bietet es noch einen besseren Grobüberblick der jeweils aktuellsten Ereignisse. Auf Snortreport greift man ebenfalls via Webbrowser zu und es ist noch etwas performanter als ACID.

3 Installation von Snort & CO

Snort kann nicht nicht einfach nur mit rpm -ivh snort-2.1.1.rpm installiert werden. Je nachdem was man alles für Ansprüche hat, muss man snort selber kompilieren. Weiter haben snort, acid und snortreport einige Abhängigkeiten, auf die nun im Folgenden eingegangen wird.

3.1 Download der benötigten Software

Download SNORT
http://www.snort.org/dl/snort-2.1.1.tar.gz

Download **PCRE** - Perl Compatible Regular Expressions
ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-4.5.tar.gz

Download The Libnet Packet Construction Library
http://www.packetfactory.net/libnet/dist/libnet.tar.gz

Download ACID
http://acidlab.sourceforge.net/acid-0.9.6b23.tar.gz

Download ADODB
http://phplens.com/lens/dl/adodb411.tgz

Download snortreport
http://www.circuitsmaximus.com/snortreport/snortreport-1.2.tar.gz

Download JPGraph
http://www.aditus.nu/jpgraph/downloads/jpgraph-1.14.tar.gz

3.2 Installation auf guardian

Normalerweise versucht man soviel als möglich als RPM zu installieren. So kann man sich das kompilieren sparen und das updaten geht später meist auch viel einfacher. Damit man nicht ständig die Cds suchen muss, empfiehlt es sich, gleich nach der Installation des Rechners alle RPMs von den Cds auf die Festplatte zu kopieren. Dazu eignet sich das Verzeichnis /var/spool/up2date ausgezeichnet.

3.2.1 GCC Installation

Um snort kompilieren zu können, bracht man natürlich einen Kompiler. Um den gcc zu installieren, werden folgende Pakete benötigt. Wichtig scheint mir auch noch die Tatsache, dass das kernel-headers Paket jetzt neu glibc-kernheaders heisst. Weiss man das nicht sucht man eine ganze Weile ;-).

[root@guardian]# rpm -ivh gcc-3.3.2-1.i386.rpm binutils-2.14.90.0.6-3.i386.rpm cpp-3.3.2-1.i386.rpm glibc-devel-2.3.2-101.i386.rpm glibc-headers-2.3.2-101.i386.rpm glibc-kernheaders-2.4-8.36.i386.rpm

3.2.2 PCRE Installation

[luke@guardian]$ cd /usr/local/src

[luke@guardian]$ tar -xvzf pcre-4.5.tar.gz

[luke@guardian]$ cd pcre-4.5

[luke@guardian]$./configure

[luke@guardian]$ make

[luke@guardian]$ su -

[root@guardian]# make install

3.2.3 LIBNET Installation

[luke@guardian]$ tar -xvzf libnet.tar.gz

[luke@guardian]$ cd libnet

[luke@guardian]$ ./configure

[luke@guardian]$ make

[luke@guardian]$ su -

[root@guardian]# make install

Wir verwendeten libnet Version 1.0.2a, wo wir noch ein File patchen mussten:

[root@guardian]# vi include/libnet.h

[vi] 87G

```
#if (!LIBNET_LIL_ENDIAN && !LIBNET_BIG_ENDIAN)
#error "byte order has not been specified, you'll \
need to #define either LIBNET_LIL_ENDIAN or LIBNET_BIG_ENDIAN.  See the \
documentation regarding the libnet-config script."
#endif
```

3.2.4 SNORT Installation

[luke@guardian]$ tar -xvzf snort-2.1.1.tar.gz

[luke@guardian]$ cd snort-2.1.1

[luke@guardian]$ ./configure --prefix=/usr --bindir=/usr/sbin --sysconfdir=/etc/snort \

  --enable-flexresp --enable-smbalerts --with-mysql –with-snmp

[luke@guardian]$ make

[luke@guardian]$ su -

[root@guardian]# make install

3.2.5 SNORT Konfiguration

Konfigurationsdateien und Regeldateien nach /etc kopieren:

[root@guardian]# cp -a /usr/local/src/snort-2.1.1/etc /etc/snort

[root@guardian]# cp -a /usr/local/src/snort-2.1.1/rules /etc/snort/rules

In der Konfigurationsdatei lassen wir für den Anfang die Defaultwerte stehen und ändern nur die Ausgabe der Daten ab. In unserem Fall schreiben wir die Daten in eine MySQL Datenbank die sich auf unserem Host admin befindet. Weiter passen wir noch den Pfad für die Regeldateien an.

[root@guardian]# vi /etc/snort/snort.conf

```
# Path to your rules files (this can be a relative path)
var RULE_PATH /etc/snort/rules

# database: log to a variety of databases
# -----------------------------------------
# See the README.database file for more information about configuring
# and using this plugin.
#
output database: log, mysql, user=snortuser password=sniffMyAss dbname=snort
host=admin
```

Damit wir snort bequemer starten und stoppen können, benötigen wir noch ein Startup-Script:

[root@guardian]# vi /etc/init.d/snort

```
#!/bin/bash
#
# Startup script for Snort NIDS on Fedora
#
#
# 2004.02.11 by luke

case "$1" in
          start)
                    echo -n "Starting snort: "
                    cd /var/log/snort
                    /usr/sbin/snort -D -c /etc/snort/snort.conf && \
                    echo " OK " || echo " Failed "
                    ;;
          stop)
                    echo -n "Stopping snort: "
                    killall snort && \
                    echo " OK " || echo " Failed "
                    echo
                    ;;
          restart)
                    $0 stop
                    $0 start
                    ;;
          status)
                    ps -ax | grep snor[t]
                    ;;
          *)
                    echo "Usage: $0 {start|stop|restart|status}"
                    exit 1
esac

exit 0
```

Mit folgenden Befehlen wird sichergestellt, das snort im Runlevel 2 auch wirklich startet.

[root@guardian]# cd /etc/init.d/

[root@guardian]# chmod 755 snort.conf

Mit dem Befehl runlevel kann überprüft werden, in welchem runlevel wir uns befinden.

[root@guardian]# runlevel

[root@guardian]# cd /etc/rc2.d

[root@guardian]# ln -s ../init.d/snort S60snort

[root@guardian]# /etc/init.d/snort start

3.2.6 Tipps und Tricks

Stunnel

Die Daten vom Host guardian werden unverschlüsselt an den Host admin übertragen. Dies ist naturelich etwas unschön. Besser wäre es, man würde das ganze mit stunnel absichern. Eine Anleitung wie dies zu realisieren ist findet man hier: http://www.stunnel.org/examples/mysql.html

Iptables und snort auf derselben Maschine

Da alle Pakete welche den Firewall passieren wirklich zuerst bei den iptables-chains vorbeikommen, kann Snort nur jene Pakete analysieren, welche der Firewall nicht gedroppt hat. Dies hat Vor- und Nachteile. Zum einen sieht man so nur Alarme, welche wirklich Dienste betreffen, die man auch anbietet. Zum anderen sieht man nicht was wirklich alles für Angriffe auf die Systeme ausgeführt wurden. Am besten installiert man sowieso einen Snort-Sensor in jeder Zone. Zum Thema iptables und snort auf der selben Maschine siehe auch Hogwash:

http://hogwash.sourceforge.net (Zur Zeit wird ein neuer Maintainer gesucht)

3.3 Installation auf admin

Auf unserem Host admin wird mysql für Datenspeicherung, und diverse Tools für die Auswertung der Daten installiert.

3.3.1 MySQL Installation

[root@admin]# rpm -ivh mysql-server-3.23.58-4.i386.rpm mysql-3.23.58-4.i386.rpm perl-DBI-1.37-1.i386.rpm perl-DBD-MySQL-2.9002-1.i386.rpm mysql-devel-3.23.58-4.i386.rpm

[root@admin]# /etc/init.d/mysql start

MySQL muss nach der Installation noch abgesichert werden. Zuerst vergeben wir ein Passwort für root und danach entfernen wir alle User ohne Benutzernamen:

[root@admin]# mysqladmin -u root -h localhost password 'xxxxx'

[root@admin]# mysqladmin -u root -h hostname password 'xxxxxx'

[root@admin]# mysql -p

mysql> use mysql;

mysql> delete from user where user='';

mysql> flush privileges;

Danach erstellen wir die snort Datenbank:

mysql> create database snort;

mysql> grant select, insert, update, delete on snort.* to snortuser identified by 'xxxxxxx'

mysql> flush privileges;

mysql> exit

Nun müssen wir nur noch die nötigen Tabellen in der Datenbank erstellen. Dafür werden uns SQL-Dateien zur Verfügung gestellt, die diese Aufgabe übernehmen.

[root@admin]# mysql -u root -p snort < /usr/local/src/snort-2.1.1/contrib/create-mysql

[root@admin]# gzip -d snortdb-extra.gz

[root@admin]# mysql -u root -p snort < /usr/local/src/snort-2.1.1/contrib/snortdb-extra

Nach dem Neustart soll mysql natürlich auch starten:

[root@admin]# chkconfig --level 2345 mysql on

Nun kann eigentlich snort auf dem Host guardian bereits gestartet werden und die Daten sollten in die Datenbank auf dem Host admin geschrieben werden.

3.3.2 Apache Installation

Die apache Installation unter Fedora gestaltet sich recht einfach. Einfach die RPMs installieren und das ganze sollte laufen.

[root@admin]# rpm -ivh httpd-2.0.47-10.i386.rpm apr-0.9.4-2.i386.rpm apr-util-0.9.4-2.i386.rpm

[root@admin]# rpm -ivh php-4.3.3-6.i386.rpm curl-7.10.6-7.i386.rpm php-mysql-4.3.3-6.i386.rpm

Einzig den Servernamen sollte man noch im httpd.conf eintragen. Wenn man schon dabei ist, kann man ja noch die Serversignatur und das Servertoken umstellen. Damit ist es nicht mehr ganz so offensichtlich was für eine Apacheversion und was für ein Betriebssystem wir verwenden.

```
ServerName admin.g3.demonet.ch
ServerSignature Off
#
# Don't give away too much information about all the subcomponents
# we are running.  Comment out this line if you don't mind remote sites
# finding out what major optional modules you are running
ServerTokens Prod
```

[root@admin]# chkconfig --level 2345 httpd on

3.3.3 ACID Installation

[root@admin]# cd /var/www/html

[root@admin]# tar –xvzf acid-0.9.6b23.tar.gz

[root@admin]# rm -rf acid-0.9.6b23.tar.gz

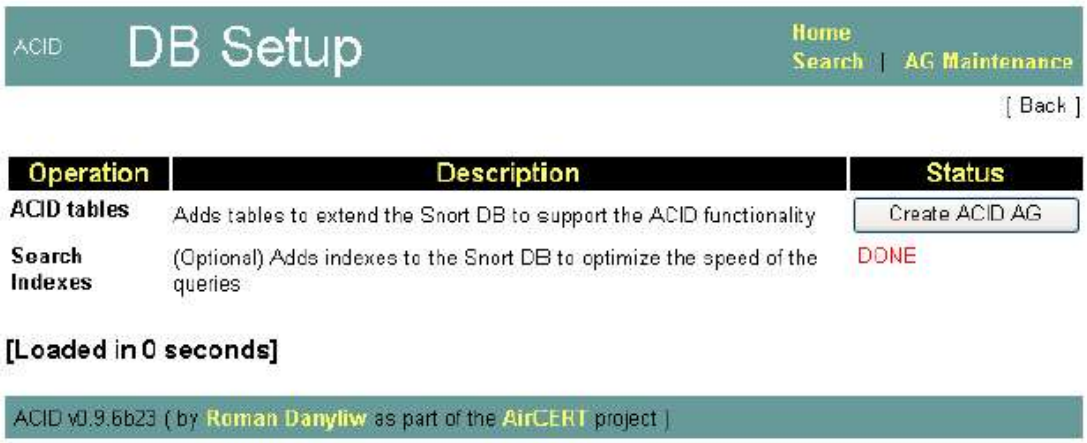
Nun muss noch die Konfigdatei angepasst werden.

[root@admin]# vi acid_conf.php

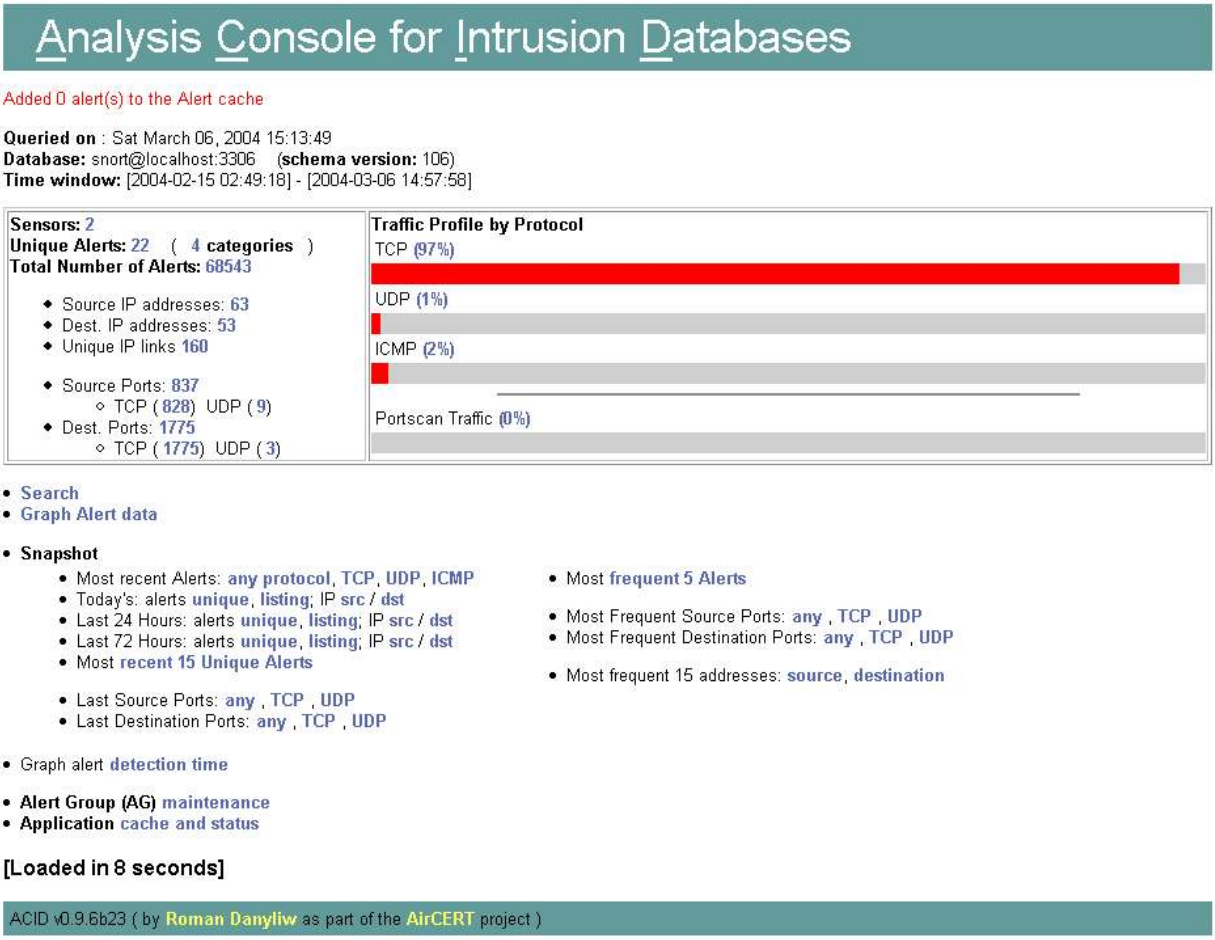
```
$DBlib_path = "/var/www/adodb";
$DBtype = "mysql";
$alert_dbname   = "snort";
$alert_host     = "localhost";
$alert_port     = "3306";
$alert_user     = "snortuser";
```

```
$alert_password = "sniffMyAss";
```

Nun sollte man mit http://admin.g3.demonet.ch/acid/ auf ACID zugreifen können. Beim ersten mal sollte folgendes erscheinen:



Mit einem Klick auf den Button “Create ACID AG” sollte alles weitere automatisch erledigt werden und die nächste Seite die man sieht sollte so aussehen:



Der Umgang mit ACID lernt man am besten durch ausprobieren.

3.3.4 ADODB Installation

ACID braucht adodb um auf die mysql Datenbank zuzugreifen.

[root@admin]# cd /var/www/

[root@admin]# tar -xzvf adodb411.tgz

[root@admin]# rm -f adodb411.tgz

3.3.5 Snortreport Installation

[root@admin]# cd /var/www/html

[root@admin]# tar -xzvf snortreport-1.2.tar.gz

[root@admin]# mv snortreport-1.2 snortreport

[root@admin]# vi /etc/srconf.php

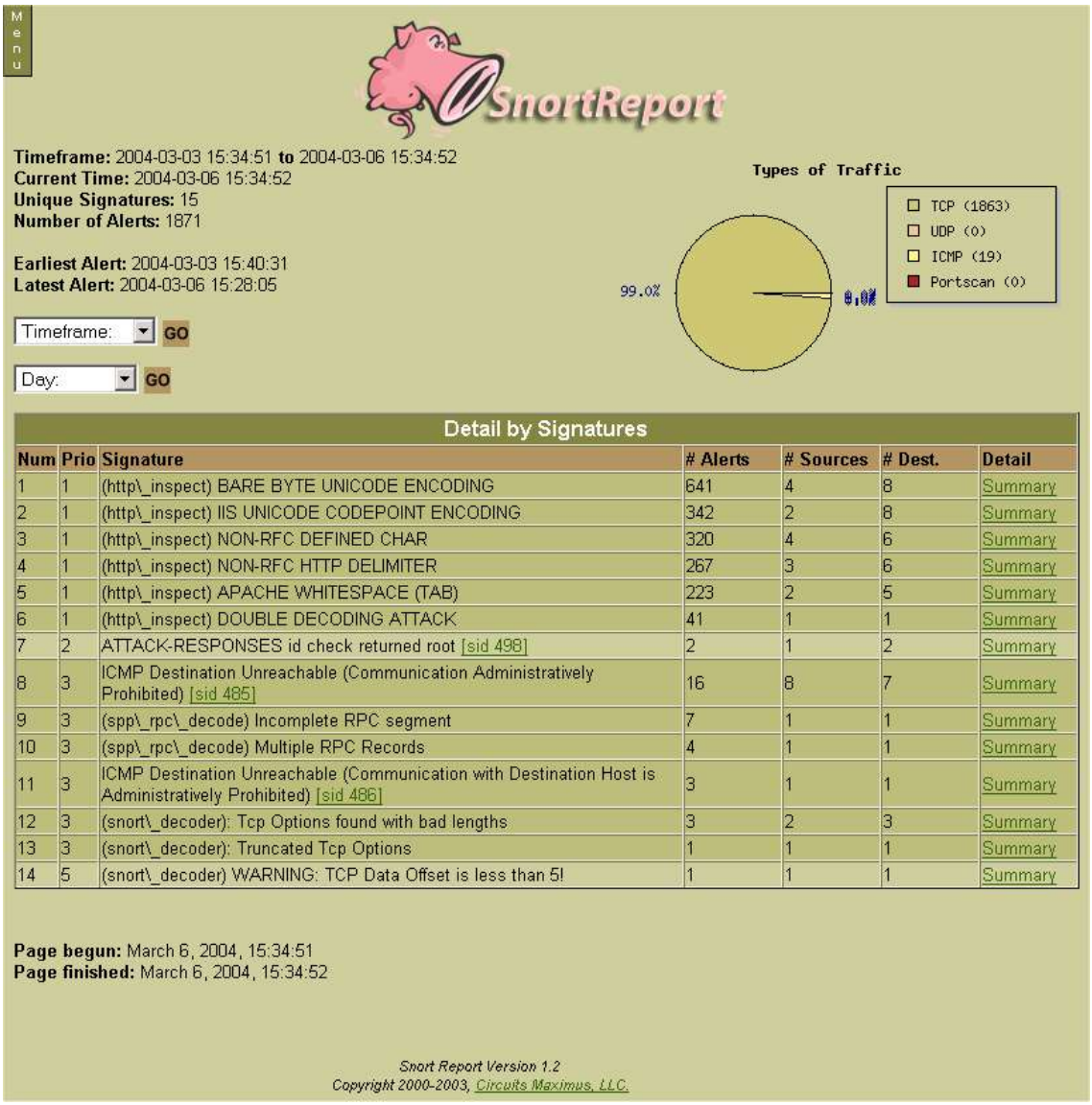
```
// Put your snort database login credentials in this section
$server = "localhost";
$user = "snortuser";
$pass = "sniffMyAss";
$dbname = "snort";

// use either "mysql" or "pgsql" below, depending on your database
$dbtype = "mysql";

// Change to FALSE if GD *and* JPGraph are not installed
$haveGD = TRUE;

// Relative path to JPGraph
// You need to have jpgraph and jpgraph_pie installed to see the chart.
// Change the variable below to reflect the location of jpgraph relative
// to Snort Report, for example "../jpgraph/", etc.
define("JPGRAPH_PATH", "../../jpgraph/");
```

So, snortreport sollte nun unter http://admin.g3.demonet.ch/snortreport erreichbar sein.



Damit auch die Grafiken korrekt ausgegeben werden, muss noch JPGraph installiert werden.

3.3.6 JPGraph Installation

[root@admin]# cd /var/www

[root@admin]# tar -xzvf jpgraph-1.14.tar.gz

[root@admin]# ln -s jpgraph-1.14.tar.gz jpgraph

[root@admin]# vi jpgraph/jpgraph.php

```
DEFINE("CACHE_DIR","/tmp/jpgraph_cache/");
DEFINE("TTF_DIR","/usr/share/fonts/msfonts/");
```

Wir hatten Probleme damit, dass jpgraph msfonts verwendet, welche natürlich auf unserem System nicht vorhanden waren. Nach einigem Suchaufwand fanden wir die Fonts hier:

http://avi.alkalay.net/software/msfonts/download/

http://webperf.org/msfonts/msfonts-1.2.1-1.noarch.rpm

Eventuell sind noch ein paar Softlinks im Fontsverezeichnis nötig, aber ansonsten sollte snortreport jetzt auch mit den Grafiken laufen.

3.3.7 Datenbank Tuning

Die Datenbank mit den Events wird sofort sehr gross und dadurch auch langsam. Die folgenden Tuningtipps erhöhen die Abfragegeschwindigkeit signifikant:

[root@admin]# mysql -p

mysql> use snort;

-- These 4 make an enormous difference as they improve several of the joins used in *every* query in alerts.php

CREATE INDEX ip_cid ON iphdr (cid);

CREATE INDEX udp_cid ON udphdr (cid);

CREATE INDEX tcp_cid ON tcphdr (cid);

CREATE INDEX icmp_cid ON icmphdr (cid);

-- More improvements by using cid indexes:

CREATE INDEX event_cid ON event (cid);

CREATE INDEX data_cid ON data (cid);

-- This one makes the two alert using queries using an index instead of a scan.

CREATE INDEX time_sig ON event (timestamp, signature, cid);

3.3.8 Tipps und Tricks

Natürlich sollte man jetzt nur via https auf den Webserver zugreifen, die Verzeichnisse mit htpasswd schützen und lokal iptables installieren (System Hardening). Die Daten die Snort sammelt sollten mit Sicherheit nicht für jedermann zugänglich sein!

4 Tuning und Pflege von Snort

Nachdem man Snort mit der Default-Konfig installiert hat, lässt man das ganze eine weile so laufen und verfolgt was passiert. Je nachdem was man mit seinem IDS erreichen will, passt man dann die Konfiguration Schritt für Schritt an. Es macht zum Beispiel keinen Sinn ein HTML-Decoding-Plugin einzusetzen, wenn man gar keine HTTP Dienste anbietet. Klar kriegt man so auch Angriffe mit die für einen Webserver bestimmt gewesen wären, aber da uns bei diesem Angriff ja nicht passieren kann, können wir ihn ignorieren. Angriffe von Würmen und Viren sind ja heutzutage etwas altägliches im Internet.

Das Herz von Snort sind eigentlich die Regeln. Man sollte diese immer auf dem neusten Stand halten, um auch aktuellste Angriffe zu erfassen. Natürlich nützen die zur Verfügung gestellten Snort-Regeln wenig, wenn es sich um einen noch unbekannten Angriff handelt. Es macht auch Sinn sich eigene, genau den Bedürfnissen angepasste Regeln zu erstellen. Schliesslich läuft alles darauf hinaus, dass man nur noch dann einen Alarm kriegt, wenn wirklich etwas los ist. Bis man aber dieses Ziel erreicht hat, kann es gut und gerne einige Monate dauern.

5 Resultate

Nachdem wir Snort im Demonet installiert hatten, hat sich unsere Datenbank sofort mit einer Vielzahl von Einträgen gefüllt. Es ist schon fast erstaunlich wie viel heutzutage los ist auf dem Internet. Wir haben auch einige Manipulationen von anderen Gruppen aus unserer Klasse aufgeschnappt, so zum Beispiel fehlerhafte Login-Versuche der Mailgruppen. Um das IDS also zu testen, braucht man es eigentlich nur ans Internet anzuschliessen. Die Penetration-Test übernimmt das Internet gleich selbst. Um aber zu sehen, ob unser Snort auch selbst herbeigeführte Ereignisse erkennt, haben wir von eine paar externen Maschinen Angriffe auf unsere Infrastruktur gestartet. Wir haben mit nmap und Nessus versucht, möglichst viel über unsere Maschinen in Erfahrung zu bringen. Das Resultat hat uns ziemlich beeindruckt. Die Angriffe wurden mit einer sehr hohen Zuverlässigkeit erkannt. ACID und Snortreport teilten uns auch gleich mit, dass es sich um nmap und Nessus Angriffe handelt. Wurden Hackertools, Viren und Würmer erkannt, kriegt man gleich einen Link auf das jeweilige Posting bei Securityfocus oder anderen sicherheitsrelevanten Webseiten.

Alles in allem machte Snort, ACID und Snortreport einen Recht guten Eindruck. Der Aufwand der betrieben werden muss, bis man ein funktionierendes Intrusion Detection System zur Verfügung hat, darf aber nicht unterschätzt werden.