Praktikum IIN Gruppe 3

Lukas Reusser, Serge Hauser, 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 1. IIN Praktikum. Bei diesem Praktikum ging es um die Realisierung einer Internetanbindung (Router) für das zugeteilte IP-LAN. Des weiteren musste ein DNS-Server für die zugeteilten DNS-Zonen konfiguriert werden. Ebenfalls musste ein DHCP-Server für lokale DHCP-Clients konfiguriert werden. Die Konfigurationen mussten mit vorhandenen Hilfsmitteln (Utilities, Netzwerkmonitoren) überprüft und dokumentiert werden. Auch mussten die Tests dokumentiert werden. Optional konnte noch eine Firewall aufgesetzt sowie weitere Funktionen, wie z.B. Mail oder Webserver, konfiguriert werden.

1.2 TCP/IP Konfiguration

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

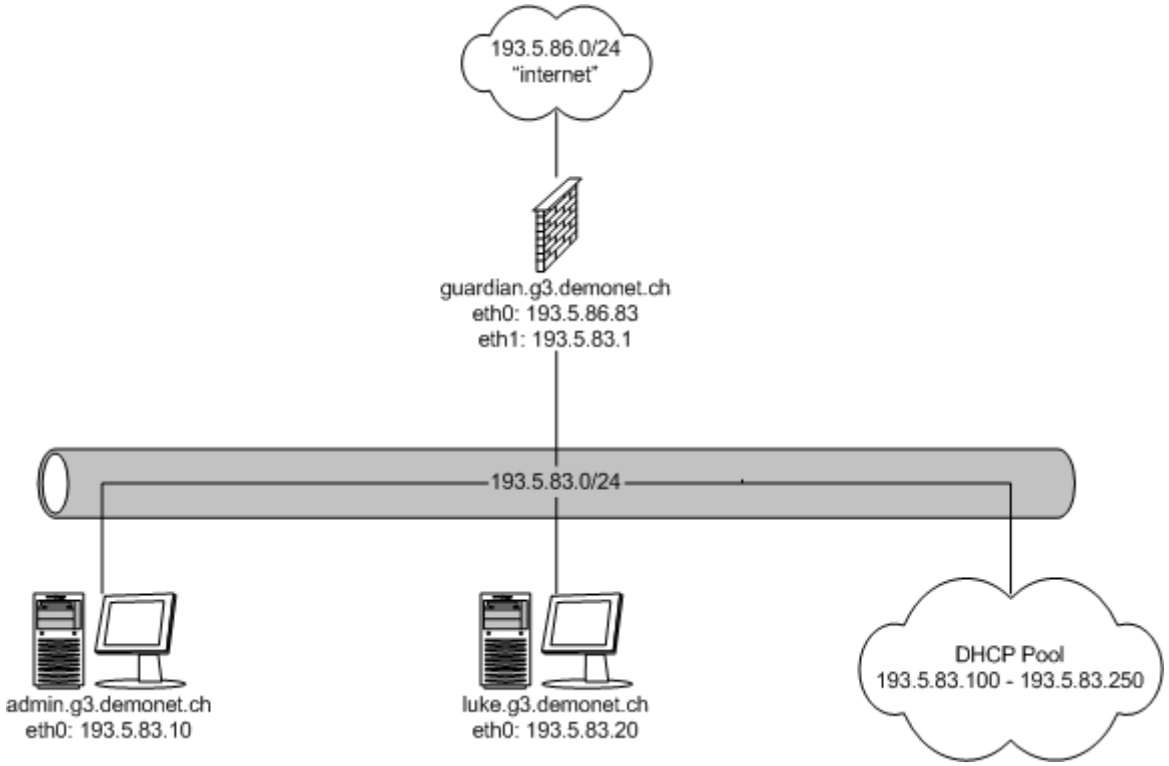

Abbildung 1: Das obige Bild zeigt den Rechner *guardian*, der als Internet-Router und Firewall aufgesetzt wurde. Am Netzwerk angeschlossen sind noch die Rechner *admin* (DNS, DHCP, Web Server) und *luke* (Client).

1.3 Verwendete Betriebssysteme

Wir hatten die Wahl zwischen Linux und Windows als Betriebssysteme, haben jedoch auf allen Rechnern Linux (RedHat Fedora Core 1) installiert. Zum testen der Konfigurationen haben wir jedoch auch Windows Clients und Knoppix Linux eingesetzt.

2 Installation Firewall und Internet-Router

In diesem Kapitel erklären wir die Installation unseres Rechners *guardian*, der als Internet-Router und Firewall dient. Wir setzen dabei das Schwergewicht der Ausführungen auf die Konfiguration der Firewall.

2.1 Vorbereitungen

Um einen Firewall, oder allgemein einen Server zu installieren, sollte man diesen aus Sicherheitsgründen vom Netzwerk trennen. Erst so hat man die Sicherheit, dass sich während oder kurz nach der Installation keine Viren, Würmen oder sonstige Malware auf dem Rechner einnistet. Installiert man zum Beispiel Windows auf einem ungeschützden Netzwerk, dauert es meist nur Sekunden, bis der neu installierte Rechner infiziert ist.

Weiter sollte man sich bereits im klaren darüber sein, welche IP-Adressen und Netze zur Verfügung stehen und wie man die Disks partitionieren will.

2.2 Booten ab CD-ROM

Um einen Server oder in unserem Fall einen Firewall zu installieren, sollte man den Textmode verwenden. Dieser ist zum einen schneller und in den meisten Fällen auch zuverlässiger und man hat keine Probleme mit den Grafikkartentreibern. Beim Bootprompt also folgendes eingeben:

```
# linux text
```

Die meisten folgenden Eingaben sind selbsterklärend. Interessanter wird es erst bei der Partitionierung der Festplatten, die wir im nächsten Abschnitt anschauen.

2.3 Partitionieren der Festplatte

Aus Sicherheitsgründen empfielt es sich für einen Firewall, mindestens folgende Partitionen zu erstellen:

/	512MB
/boot	128MB
/usr	>= 2GB
/var	>= 4GB (je nachdem was alles geloggt wird etc..)
/tmp	512MB sollten in jedem Fall ausreichen

Es empfielt sich, ein paar Gigabyte auf der Disk frei zu lassen. So kann man später bei Diskplatzknappheit schnell eine neue Partition erstellen und diese am richtigen Ort einhängen. Hat man zum Beispiel das /var Directory zu klein gewählt, und die Logfiles haben jetzt die Partition gefüllt, so kann man einfach die neue Partition auf /var/log mounten und man hat wieder genügend Platz.

Auf einem Server wo mehrere Benutzer Zugriff haben, empfielt es sich weiter noch eine /home Partition zu erstellen. Auf einem Web- oder FTP-Server ist ein /var/www oder ein /var/ftp sicherlich auch eine gute Sache. Denn so kann es nicht passieren dass eine gefüllte Partition das ganze System lahmlegt.

2.4 Auswahl der Pakete

Bei der Auswahl der Pakete gilt: Weniger ist Mehr! Diese Aussage trifft vorallem auf Systeme zu, die möglichst sicher sein sollten. Es empfielt sich, das absolute Minimum an Paketen auszuwählen und zu installieren. Die Pakete iptables, iproute sollten auf jeden Fall angewählt sein, was aber eigentlich auch immer der Fall ist.

2.5 Default Firewall Einstellungen

Fedora Linux 1.0 bietet die Option, bereits bei der Installation einen Firewall zu konfigurieren. Von dieser Option sollte man auf jeden Fall Gebrauch machen! Am besten erlaubt man vorerst keine eingehenden Verbindungen vom Internet. Erachtet man das interne Netz als sicher, so kann man Verbindungen vom internen Netz aufs Internet zulassen.

2.6 Konfiguration nach Reboot

2.6.1 Iptables überprüfen

Nachdem die Installation abgeschlossen ist, wird der Rechner neu gebootet. Bereits beim Bootvorgang sollte man auf eventuell auftretende Fehler achten. Das Starten des Netzwerkes sollte eigentlich einen Fehler verursachen, da das Ethernetkabel hoffentlich noch nicht eingesteckt ist.

Verlief der Bootvorgang sonst störungsfrei, kann man sich als root anmelden. Als erstes sollte man sich davon überzeugen, dass der Default-Firewall korrekt gestartet hat:

[root@guardian]# service iptables status

Sieht die Ausgabe korrekt aus, kann man nun die Netzwerkkabel einstecken.

Richtige Sicherheitsgurus stecken natürlich erst nach dem Systemhardening und Updaten den Netzwerkstecker ein. Will man jedoch ein wirklich 100% sicheres System, darf man den Netzwerkstecker aber nie einstecken!

2.6.2 Routing überprüfen

Um das Routing zu testen, kann man nun von einem internen Rechner aus versuchen, eine Verbindung ins Internet herzustellen. Dies sollte eigentlich schon funktionieren. Tut es dies aber nicht, so sollte man das Routing Flag überprüfen und gegebenenfalls mit dem folgenden Befehl aktivieren:

[root@guardian]# echo "1" > /proc/sys/net/ipv4/ip_forward

Um das Routing dauerhaft zu aktivieren, muss */etc/sysctl.conf* wie folgt editiert werden:

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

```
# Controls IP packet forwarding
net.ipv4.ip_forward = 1
```

2.7 System Hardening

2.7.1 Unnötige Pakete entfernen

Ganz nach dem Leitspruch; “Weniger ist Mehr”, entfernen wir jetzt noch unnötige Pakete. Mit folgendem Befehl kann man alle installierten Pakete durchsehen:

[root@guardian]# rpm -qa | more

Weder haben wir einen ISDN Anschluss, noch verwenden wir Yellow Pages oder NFS. Man sollte wirklich alle nicht benötigten Pakete entfernen, installieren kann man sie jederzeit wenn man sie wirklich braucht. Folgende Pakete können zum Beispiel bedenkenlos deinstalliert werden:

```
# rpm -e wireless-tools isdn4k-utils ypbind irda-utils kernel-pcmcia-cs yp-tools nfs-utils wvdial
```

2.7.2 Unnötige Dienste stoppen

Mit folgendem Befehl können alle aktiven Ports angezeigt werden:

[root@guardian]# netstat -na

Auf einem Firewall sollte höchstens ein SecureShell Daemon auf dem Management-Interface auf eingehende Verbindungen warten. Alle anderen Dienste haben dort nichts verloren, so harmlos sie auch sein mögen! Um zum Beispiel Port 111 dicht zu machen, gibt man folgendes Kommando ein:

[root@guardian]# /etc/init.d/portmap stop

Nun kann man wieder mit netstat überprüfen, ob weiter an diesem Port gelauscht wird.

Damit der Dienst auch nach dem Reboot abgeschaltet bleibt, empfiehlt sich folgendes Kommando:

[root@guardian]# chkconfig --level 2345 portmap off

Nun kann man auch noch die anderen ungewünschten Dienste nach /dev/null schicken:

[root@guardian]# chkconfig --level 2345 netfs off

[root@guardian]# chkconfig --level 2345 sendmail off

Netstat sollte jetzt wirklich keine offenen Ports anzeigen, ausser vielleicht sshd.

2.7.3 Unprivilegierte Benutzer erstellen

Da man sich auch auf dem Firewall nicht remote via root einloggen soll, erstellen wir unprivilegierte Benutzer:

[root@guardian]# adduser luke

Damit man den Account verwenden kann, muss man gleich noch ein Passwort vergeben:

[root@guardian]# passwd luke

Um sich mit su Rootrechte verschaffen zu können, muss der Benutzer zusätzlich noch in der*wheel* Gruppe eingetragen werden:

[root@guardian]# vi /etc/group

```
wheel:x:10:root,luke
```

Nun sollte man natürlich noch das Einloggen als root von einem anderen Rechner aus verbieten:

[root@guardian]# vi /etc/ssh/sshd_config

```
PermitRootLogin no
```

2.7.4 host, hosts.allow, hosts.deny konfigurieren

Weiter sollte man noch die hosts.allow und hosts.deny Files mit nützlichen Informationen füttern:

[root@guardian]# vi /etc/hosts.deny

```
ALL: ALL
```

Mit diesem Eintrag verbietet man allen Rechnern auf irgend einen lokalen Service zuzugreifen. Da man jetzt aber zum Beispiel sshd erlauben will, ist noch ein Eintrag im hosts.allow File nötig:

[root@guardian]# vi /etc/hosts.allow

```
sshd: admin.g3.demonet.ch, luke.g3.demonet.ch
```

Falls man auf dem Firewall keine DNS konfigurieren will, aus welchen Gründen auch immer, kann das *hosts* File gute Dienste leisten:

[root@guardian]# vi /etc/hosts

```
127.0.0.1 guardian.g3.demonet.ch guardian localhost.localdomain localhost
193.5.83.10         admin.g3.demonet.ch admin
193.5.83.20         luke.g3.demonet.ch luke
```

Die Syntax lautet: `IP-Adresse FQDN Alias`

2.7.5 Logging konfigurieren

Auf einem Firewall sollte man unbedingt seine Logdatein etwas länger aufbewahren als auf einer Workstation. Mit folgender Konfiguration teilt man logrotate mit, dass es die Logdateien 1 Jahr aufbewahren soll:

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

```
# keep 52 weeks worth of backlogs
rotate 52
```

Weiter kann man logrotate noch anweisen, die archivierten Logdateien zu komprimieren:

```
# uncomment this if you want your log files compressed
compress
```

Bei einem Einbruch werden aber zuerst immer die Spuren in den Logdateien verwischt, da nützt auch das längere Aufbewahren nichts. Um diese Problem zu umgehen, sollte man einen Loghost an einem sicheren Ort aufstellen. Dieser sammelt dann die Logging-Informationen von allen Servern im Netzwerk und man kann diese so an einem zentralen Ort auswerten.

Hat man den Loghost installiert, braucht man nur folgende Zeile in die syslog Konfiguration einzufügen:

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

```
# loghost (syslog server)
*.*       @admin.g3.demonet.ch
```

Mit diesem Eintrag werden alle Loginformationen sowohl lokal gespeichert, wie auch an den Loghost weitergeleitet. In syslog.conf kann man noch eine Menge weiterer nützlicher Einträge machen. Mehr Informationen hierzu liefert *man syslog.conf*.

Der Loghost ist sehr einfach installiert. Man braucht nur den syslog daemon mit der Option -r für Remote zu starten und schon können die Clients auf den Loghost schreiben:

[root@admin]# vi /etc/init.d/syslog

```
SYSLOGD_OPTIONS="-r -m 0"
```

man syslogd liefert noch weitere Informationen zu den Parametern.

2.7.6 Zeitserver konfigurieren

Um eventuell einen Einbruch rekonstruieren zu können, sollte jedes System die selbe Systemzeit haben. Dafür wurde das Network Time Protocol (NTP) entwickelt. Am besten hat man also einen internen Zeitserver mit einer Funkuhr oder man verwendent einen Zeitserver aus dem Internet. Um die Abfragen ins Internet aber möglichst klein zu halten, sollte man einen eigenen Zeitserver installieren und diesen so konfigurieren, dass er die Zeit im Internet abgleicht. Man sollte aber nur sichere NTP Server verwenden und ihre Idendität bei jedem Update auch prüfen. Ein gewiefter Hacker könnte sonst falsche Zeitangaben in den Datenstrom einschleusen und so die Daten in den Logdateien verfälschen.

Um die Zeit vom Timeserver abzuholen, konfiguriert man am besten einen cronjob:

[root@guardian]# vi /etc/cron.daily/ntpdate

```
#!/bin/sh
ntpdate 193.5.83.20
```

[root@guardian]# chmod 755 /etc/cron.daily/ntpdate

Natürlich muss man vorher noch das Paket installieren, welches ntpdate enthält. In diesem Fall wäre das ntp.

2.8 System aktualisieren mit up2date

up2date ist ein mächtigens Tool von RedHat, um ein System auf dem aktuellen Stand zu halten. Freundlicherweise haben sie es auch dem Fedora Projekt zur Verfügung gestellt, obwohl die Verbindungsgeschwindigkeit zu wünschen übrig lässt. Dies würde sich jedoch schnell ändern, wenn man mit up2date auf einen Sunsite-Mirror zugreifen könnte.

Um up2date zu konfigurieren, wird folgendes Kommando benutzt:

[root@guardian]# up2date --nox --configure

Das --nox bedeutet no X, also kein X-Window. Ansonsten kriegt man eine Fehlermeldung dass up2date keine display gefunden hat. Um up2date zu verwenden, braucht man noch den gpg-key von Fedora:

[root@guardian]# rpm --import /usr/share/rhn/RPM-GPG-KEY-fedora

Um dann das System auf den aktuellen Stand zu bringen, braucht man dann nur noch folgenden Befehl abzusetzen:

[root@guardian]# up2date --nox -u

Der Parameter -u steht für Update. *man up2date* zeigt weitere nützliche Parameter an.

2.9 Gefahren und wie man sie minimieren kann

2.9.1 Von Script-Kiddies bis Senior Guru Cracker

Wenn man sein System auf dem aktuellen Stand hält, hat man schon ca. 99% von allen Gefahren abgewendet (sofern man nichts falsch konfiguriert hat;-). Die Script-Kiddies verwenden veröffentlichte Exploits von bereits bekannten Sicherheitslücken. Hat man die Software auf dem neusten Stand, haben die Kiddies keine Chance. Auch Viren und Würmer basieren meist auf bekannten Sicherheitslücken. Hat es aber wirklich ein Senior Guru Cracker auf ein System abgesehen, wird es schon etwas schwieriger. Dieser hat vielleicht einen Exploit parat, bevor ein Update zur Verfügung steht. Um auch diesen Angreifer abzuwehren, sollte man bei der Sicherheit immer sehr viel Redundanz haben. Dazu gehört zum Beispiel Checksums, File-Flags, Securitylevels, Systrace Policies, Real-Time Systrace Monitoring, Non-Executable Stack, Read-Only Segements und Propolice. Diese Fea-

tures werden aber nur Teils vom aktuellen Linux Kernel unterstützt. Weitere Informationen zu den aufgezählten Securityfeatures finden man auf folgender Homepage: http://www.openbsd.org

2.9.2 Intrusion Detection Systems

Zumindest sollte man ein lokales Intrusion Detection System (IDS) installieren, welches mindestens MD5 Summen von allen relevanten Files anlegt und auch sonst nach irgendwelchen Ungereimtheiten sucht. *Logcheck* zum Beispiel prüft die Logfiles auf Abnormalitäten, *Integrit* hingegen legt MD5 Summen aller konfigurierten Files an und Vergleicht diese immer wieder. Stimmern zwei Summen nicht mehr überein, schlägt er alarm. Dieses Thema füllt jede Menge dicker Bücher und bei Google findet man auch tausende von Seiten darüber.

2.9.3 Wurde die Kiste geknackt?

Schlägt irgendwann ein lokales IDS Alarm und es sieht nicht nach einem Fehlalarm aus, sollte man sofort den Netzwerkstecker rausziehen! Nach einem erfolgreichen Einbruch, installiert der Hacker/Cracker meistens ein Rootkit. Handelt es sich bei diesem Rootkit um eine weit verbreitete Version, kann man es mit *rootcheck* aufspüren. Bei rootcheck handelt es sich um ein Perlscript, welches nach Anzeichen aller gängiger Rootkits sucht. Es läuft unter praktisch jeder Linux Distribution sowie OpenBSD und sogar SunOS. Es kann unter folgender URL runtergeladen werden: http://www.ossec.net/rootcheck/

Ein weiteres Tool mit nahezu derselben Funktionalität ist *chkrootkit*. Hierbei handelt es sich um ein Shellscript, welches unter praktisch jedem UNIX laufen sollte. Es kann hier heruntergeladen werden: http://www.chkrootkit.org/

2.10 Firewall Regeln konfigurieren

2.10.1 Einleitung

Hat man die grundlegende Installation abgeschlossen, kann man sich daran machen, die Firewall Regeln aufzusetzen. Hier sollte man sich auf jeden Fall ein Konzept ausdenken und nicht einfach loslegen. Man muss sich wirklich vorher im Klaren darüber sein, welchen Verkehr den Firewall passieren darf und welchen nicht. Wenn man das ganze einmal dokumentiert hat, ergeben sich später auch keine bösen Überraschungen wenn zum Beispiel einmal die Festplatte mit der gespeicherten Konfiguration ausfällt und man per Zufall vergessen hat ein Backup zu erstellen. Das passiert jedem Systemadministrator nur einmal...

2.10.2 FWBuilder

Sobald man mehrere Firewalls administrieren muss, oder der Regelsatz umfangreicher wird, empfiehlt es sich ein Hilfswerkzeug zur Regelverwaltung zu verwenden. Sehr empfehlenswert ist *fwbuilder* von Vadim Kurland. Die Files sowie super Anleitungen und Dokumentationen können unter folgendem Link runtergeladen werden: http://www.fwbuilder.org/

Eingefleischte hardcore UNIX Gurus verwenden natürlich weiterhin den Vi.

2.10.2.1 Installation von fwbuilder

FWBuilder wird nicht auf dem Firewall selber, sondern auf einer Admin-Workstation installiert. Die Admin-Workstation sollte über ein X-Window System verfügen und selber sehr sicher konfiguriert sein. Die Kette der Sicherheit ist nur so stark wie ihr schwächstes Glied! Es empfiehlt sich zudem der Admin-Workstation einen unauffälligen Namen zu geben. Der Name “test3” erregt sicher viel weniger Aufmerksamkeit als der Name “admin”.

Die folgenden Pakete werden verwendet und können auf der fwbuilder Homepage runtergeladen werden:

- libfwbuilder-1.0.2-1.fdr1.i386.rpm
- fwbuilder-1.1.1-1.fdr1.i386.rpm
- fwbuilder-ipt-1.1.1-1.fdr1.i386.rpm

Will man auch Regeln für andere Paketfilter erstellen können, braucht man natürlich noch den jeweiligen Compiler-Regelsatz:

- fwbuilder-ipf-1.1.1-1.fdr1.i386.rpm
- fwbuilder-ipfw-1.1.1-1.fdr1.i386.rpm
- fwbuilder-pf-1.1.1-1.fdr1.i386.rpm

Zur Zeit werden OpenBSD PF, Cisco PIX, Linux IPTables und IPFilter unterstützt.

Unter Linux braucht man noch folgende Pakete um fwbuilder erfolgreich installieren zu können:

- gtkmm-1.2.10
- libsigc++10-1.0

Man findet sie ebenfalls auf der fwbuilder Homepage. Findet man dort einmal ein RPM nicht, so sind folgende Suchmaschinen sicher eine grosse Hilfe:

http://www.rpmfind.net/

http://www.rpmseek.com/index.html

Konnten alle Pakete mit *rpm -ivh* fehlerfrei installiert werden, kann man die Applikation mit *fwbuilder* starten.

2.10.2.2 Verwendung von fwbuilder

Mit den nötigen Netzwerkkentnissen ist fwbuilder eigentlich selbsterklärend. Für die meisten Dienste gibt es vorgefertigte Objekte und man braucht bloss noch seine eigenen Netzwerke und Rechner einzutragen. Nun kann man die Objekte einfach mit der Maus in die richtigen Felder ziehen und schon ist die Firewall-Regel erstellt:

Firewall Regeln von guardian.g3.demonet.ch

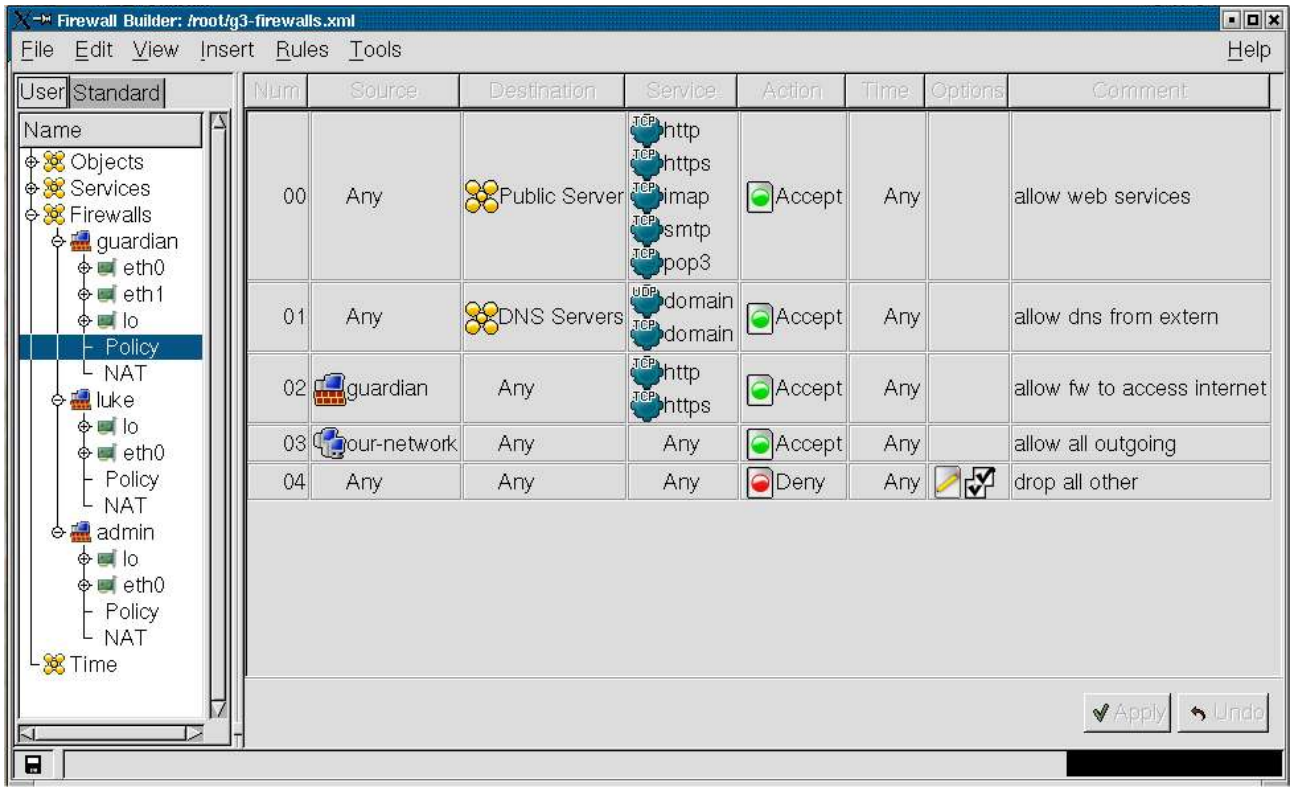

Abbildung 2: FWBuilder Policy

In der obigen Abbildung 2 sehen wir den Regelsatz von unserem Firewall mit dem Namen Guardian. Die Regel 00 erlaubt jedem Rechner auf unseren öffentlichen Server zuzugreifen. Jedoch nur auf die Dienste http, https, imap, pop und smtp. Regel 01 erlaubt jedermann auf unsere DNS Server zuzugreifen. Dies via TCP als auch UDP. Regel 02 erlaubt es unserem Firewall eine Verbindung ins Internet herzustellen und zwar via http als auch via https. Dies ist nötig um die Updates abzuholen. Regel 03 erlaubt es unserem ganzen internen Netzwerk, eine Verbindung zum Internet herzustellen. Es sind alle Dienste erlaubt. Regel 04 dropt den Rest und macht einen Logfile Eintrag.

Mit den Ausdrücken *Any* muss man sehr aufpassen, dass man keine Fehler macht. Man sollte sie wenn möglich vermeiden. Normalerweise verbietet man jeden Verkehr und gibt nur die gewünschten Services frei. Der Einfachheit halber und weil wir uns keine Restriktionen auferlegen wollten, haben wir aber fürs interne Netzwerk alle Dienste erlaubt (Rule 03).

Um bessere Sicherheit zu gewährleisten, haben wir auf jedem unserer Rechner einen Firewall konfiguriert. Dies ist eigentlich auch sinnvoll, denn wurde einmal eine Maschine in einer DMZ kompromittiert, kommt der Eindringling nicht so leicht auf all die anderen Rechner.

Im fwbuilder kann man auch alle möglichen Optionen bequem via GUI einstellen:

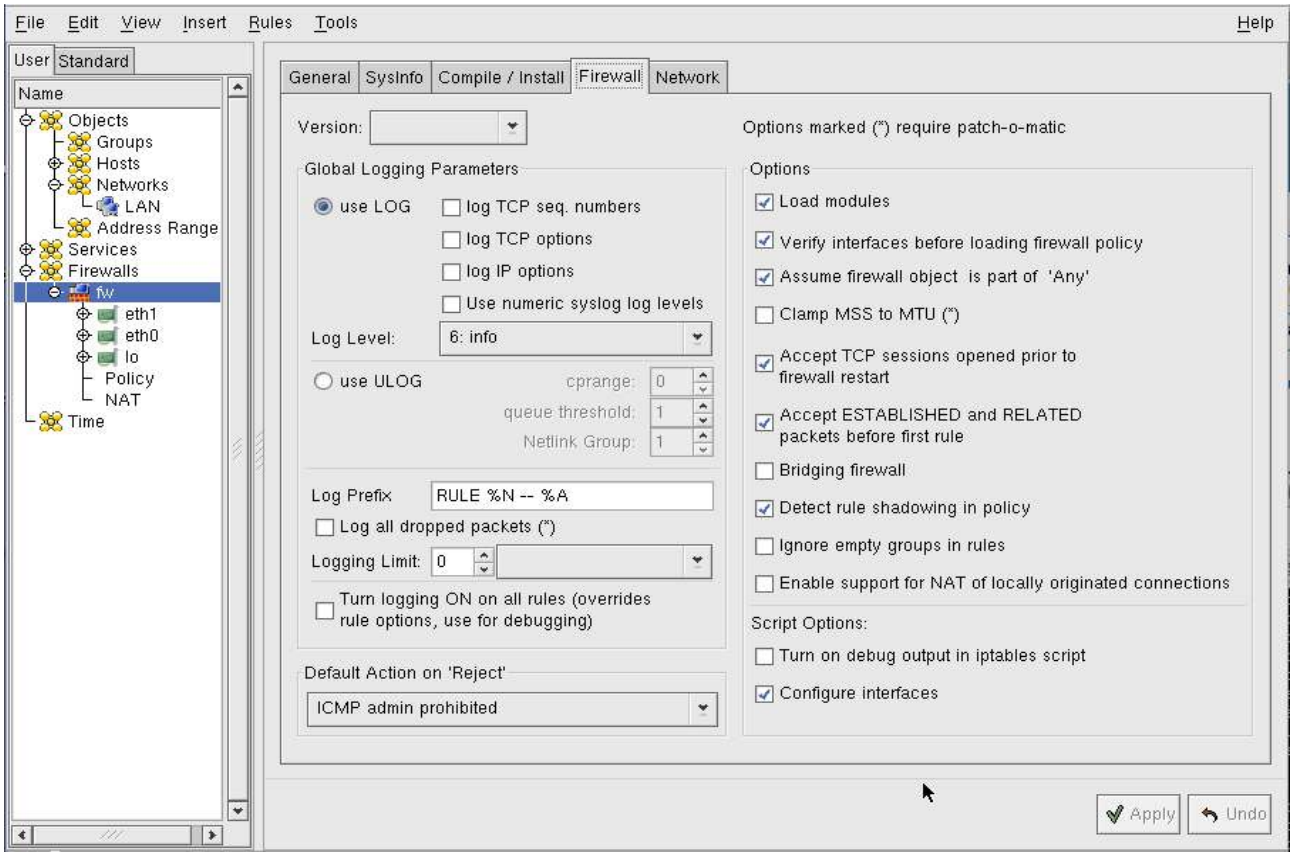

Abbildung 3: FWBuilder Firewall Optionen

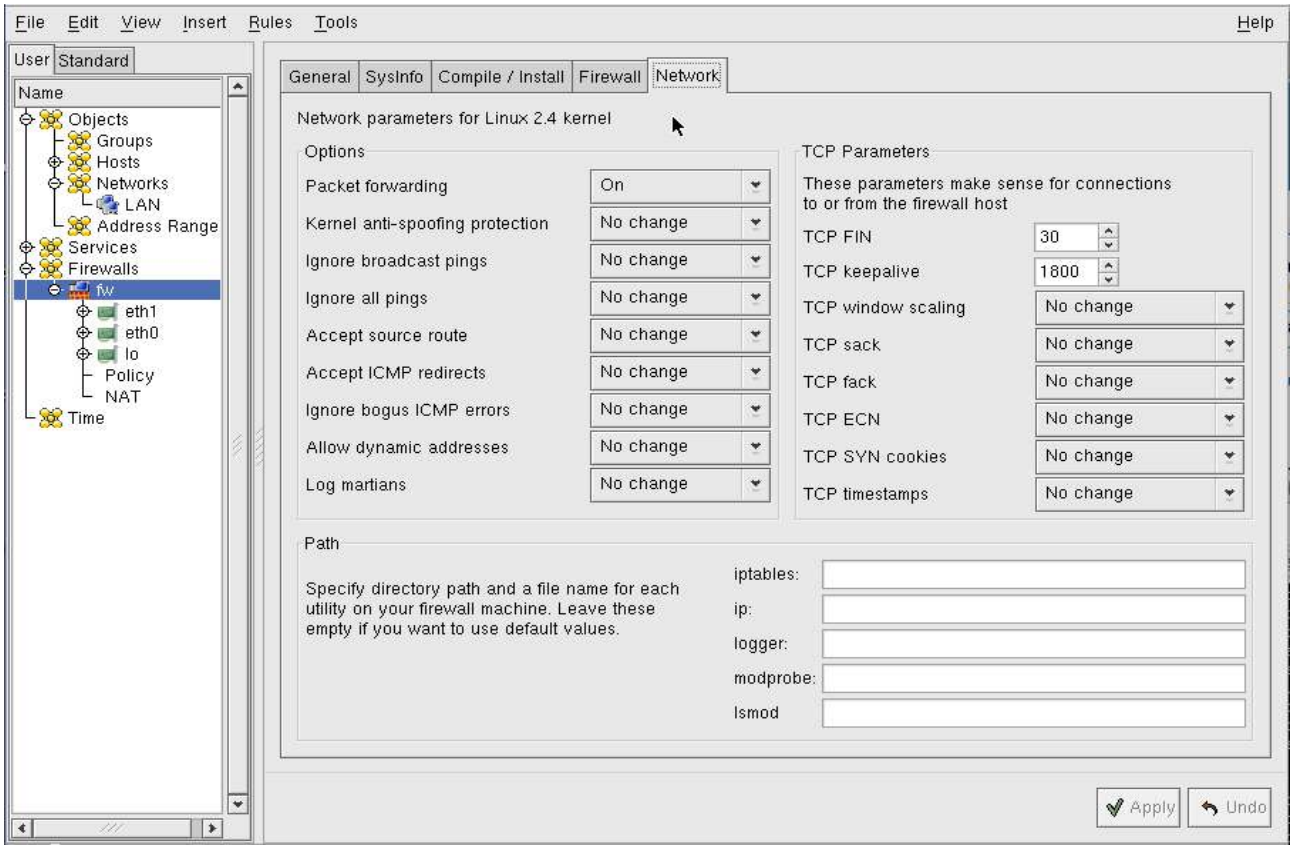

Abbildung 4: FWBuilder Network Optionen

Um zum Beispiel Pings, also ICMP Antworten zu unterbinden, braucht man nur ein Flag zu setzen. Auch die sehr nützlichen TCP Syn Cookies kann man sehr einfach aktivieren. Sie verhindern einen TCP-SYN DOS Angriff.

Hat man sich die gewünschten Regeln zusammengestellt, kann man den Regelsatz kompilieren. War dies erfolgreich, hat man ein fertiges Shellscript, welches man nun auf dem Firewall starten kann.

2.10.2.3 Regelsatz auf Firewall installieren

Um den Regelsatz nun auf der Firewall zu installieren gibt es mehrere Möglichkeiten. Man kann eine ssh Verbindung aufbauen und das Script so auf die Firewall kopieren. Dort muss man die vorhandene Version mit der neuen ersetzen. Danach kann man den Firewall mit *service iptables restart* neu starten und die neuen Regeln sollten aktiv sein.

Viel bequemer geht das aber mit dem *fwb_install* Script. Eine gut gelungene Anleitung dazu findet man hier: http://www.fwbuilder.org/archives/cat_howtos.html#000095

Hält man sich an die Anleitung, braucht man nach dem Kompilieren der Regeln nur noch das Menü Rules -> Install anzuwählen. Nach Eingabe des Passwortes vom Private-Key wird das Script auf den Firewall kopiert und die neuen Regeln werden sofort aktiv!

Es empfiehlt sich, auf dem Firewall noch folgende Änderung vorzunehmen:

[root@guardian]# vi /etc/sysconfig/iptables-config

```
# Save current firewall rules on stop.
# Value: yes|no,  default: no
IPTABLES_SAVE_ON_STOP="yes"
```

Dadurch wird sichergestellt, dass die vom fwb_install Script installierten Regeln bei einem Reboot gespeichert und anschliessend auch wieder geladen werden. Man kann natürlich auch das fwb_install Script so anpassen,

dass die Konfiguration gleich von Anfang an in der richtigen Form in die richtige Datei kopiert wird. Diese Lösung hat aber den Vorteil, dass sie bei jeder Distribution läuft, die das Speichern der Regeln beim stoppen zulässt. Man sollte aber noch prüfen, ob tatsächlich ein *service iptables stop* beim Herunterfahren ausgeführt wird.

2.10.2.4 FWBuilder Schlusswort

FWBuilder ist ein sehr mächtiges Werkzeug, welches sich vor einer kommerziellen Lösung nicht zu verstecken braucht. Mit fwbuilder ist es möglich eine ganze Heerschar von Firewalls verschiedener Hersteller zu administrieren, ohne dabei die Übersicht zu verlieren. Es wird ständig weiterentwickelt und hat mittlerweile eine sehr grosse Anhängerschaft. Definitiv eines der besten OpenSource Tools im Sicherheitsbereich auf dem Markt!

2.11 Firewall Installation Schlusswort

Dies war nur eine kurze Zusammenfassung zur Firewall Installation. Es gibt noch Dutzende andere Punkte die man beachten sollte, dies füllt aber einige dicke Bücher und würde den Rahmen dieser Dokumentation bei weitem sprengen. Wenn aber all diese Tipps beachtet werden, steht man sicher nicht so schlecht da.

3 Konfiguration DNS-Server

In den nächsten Abschnitten werden wir unsere DNS Server Konfiguration erläutern. Wir haben DNS auf unserem Rechner *admin* installiert.

3.1 Allgemein

Als Nameserver wurde das mitgelieferte Bind Paket verwendet.

Den Dienst kann man mit

```
/etc/init.d/named [start|stop|restart]
```

manuell kontrollieren.

Um den Nameserver beim Booten automatisch zu starten reicht ein einmaliges

```
chkconfig --add named
```

3.2 /etc/named.conf

Forwarder ns.isbe.ch hinzufügen:

```
options {
          directory "/var/named";
          forwarders { 193.5.80.20; };
          notify no;
};
```

Hinzufügen von zwei neuen Zonen für g3 und den Reverselookup:

```
zone "g3.demonet.ch" {
          type master;
          file "g3.demonet.ch.zone";
};
zone "83.5.193.in-addr.arpa" {
          type master;
          file "83.5.193.in-addr.arpa.zone";
};
```

3.3 Zonefiles

Erstellen einer Zone für g3.demonet.ch, dabei wurden Einträge für die 3 Rechner gemacht, sowie eine Reihe von Aliases die der Bequemlichkeit dienen:

```
$TTL 86400
g3.demonet.ch.    IN      SOA       ns.g3.demonet.ch.    root.g3.demonet.ch. (
                                    2003120208 ; serial
                                    10800      ; refresh
                                    3600       ; retry
                                    604800     ; expire
                                    86400      ; ttl
                                    )

                  IN      NS        ns.g3.demonet.ch.
                  IN      NS        ns.isbe.ch.
                  IN      MX  10    ns.g3.demonet.ch.

ns                IN      A         193.5.83.10
luke              IN      A         193.5.83.20
guardian          IN      A         193.5.83.1
fw                IN      CNAME     guardian.g3.demonet.ch.
gw                IN      CNAME     guardian.g3.demonet.ch.
www               IN      CNAME     ns.g3.demonet.ch.
mail              IN      CNAME     ns.g3.demonet.ch.
smtp              IN      CNAME     ns.g3.demonet.ch.
imap              IN      CNAME     ns.g3.demonet.ch.
admin             IN      CNAME     ns.g3.demonet.ch.
```

Für den Reverselookup wurde ebenfalls eine Zone eingerichtet:

```
$TTL 86400
@         IN      SOA         ns.g3.demonet.ch.      root.g3.demonet.ch   (
                                        2003120201 ; serial
                                        10800 ; refresh
                                        3600 ; retry
                                        604800 ; expire
                                        86400 ; ttl
                                        )

          IN      NS          ns.g3.demonet.ch.
          IN      NS          ns.isbe.ch.

1         IN      PTR         guardian.g3.demonet.ch.
10        IN      PTR         ns.g3.demonet.ch.
20        IN      PTR         luke.g3.demonet.ch.
```

3.4 DNS Tests

3.4.1 Generelle Tests

Test 1	
Host:	**luke**
Test:	*nslookup admin.g3.demonet.ch*
Ergebnis:	**FAILED**
Fix:	Nameserver läuft mit einem changeroot unter /var/named/chroot. Wir haben also die falschen Files editiert.
Test:	*nslookup admin.g3.demonet.ch*
Ergebnis:	**OK**
Server:	ns.g3.demonet.ch
Address:	193.5.83.10#53
admin.g3.demonet.ch	Canonical name = ns.g3.demonet.ch
Name:	ns.g3.demonet.ch
Address:	193.5.83.10

Test 2	
Host:	**luke**
Test:	*nslookup 193.5.83.10*
Ergebnis:	**OK**
Server:	ns.g3.demonet.ch
Address:	193.5.83.10#53
10.83.5.193.in-addr.arpa	name = ns.g3.demonet.ch

3.4.2 DNS Sleuth Resultat

Check results for g3.demonet.ch

Starting zone checks for g3.demonet.ch
Checking zone name
Checking existence of zone

```
    -> g3.demonet.ch.   86400   IN      SOA     ns.g3.demonet.ch. root.g3.demonet.ch. (
                                        2003120208      ; Serial
                                        10800   ; Refresh
                                        3600    ; Retry
                                        604800  ; Expire
                                        86400 ) ; Minimum TTL
```

Checking NS records

```
    -> g3.demonet.ch.   86400   IN      NS      ns.g3.demonet.ch.
    -> g3.demonet.ch.   86400   IN      NS      ns.isbe.ch.
```

Checking nameserver authority and synchronization

```
    -> admin.g3.demonet.ch.     86400   IN      CNAME   ns.g3.demonet.ch.


    -> ns.g3.demonet.ch.        86400   IN      A       193.5.83.10
    -> 10.83.5.193.in-addr.arpa.        86400   IN      PTR     ns.g3.demonet.ch.
```

Probing name server admin.g3.demonet.ch (193.5.83.10)

```
    -> g3.demonet.ch.   86400   IN      SOA     ns.g3.demonet.ch. root.g3.demonet.ch. (
                                        2003120208      ; Serial
                                        10800   ; Refresh
                                        3600    ; Retry
                                        604800  ; Expire
                                        86400 ) ; Minimum TTL
    -> g3.demonet.ch.   86400   IN      NS      ns.g3.demonet.ch.
    -> g3.demonet.ch.   86400   IN      NS      ns.isbe.ch.
    -> ns.g3.demonet.ch.        86400   IN      A       193.5.83.10
```

Probing name server ns.g3.demonet.ch (193.5.83.10)

```
    -> g3.demonet.ch.   86400   IN      SOA     ns.g3.demonet.ch. root.g3.demonet.ch. (
                                        2003120208      ; Serial
                                        10800   ; Refresh
                                        3600    ; Retry
                                        604800  ; Expire
                                        86400 ) ; Minimum TTL
    -> g3.demonet.ch.   86400   IN      NS      ns.g3.demonet.ch.
    -> g3.demonet.ch.   86400   IN      NS      ns.isbe.ch.
    -> ns.isbe.ch.      86400   IN      A       193.5.80.20
    -> 20.80.5.193.in-addr.arpa.        86400   IN      PTR     ns.isbe.ch.
    -> 20.80.5.193.in-addr.arpa.        86400   IN      PTR     www4you.isbe.ch.
    -> www4you.isbe.ch. 86400   IN      CNAME   ns.isbe.ch.
    -> ns.isbe.ch.      86400   IN      A       193.5.80.20
```

Probing name server ns.isbe.ch (193.5.80.20)

```
    -> g3.demonet.ch.   86400   IN      SOA     ns.g3.demonet.ch. root.g3.demonet.ch. (
                                        2003120208      ; Serial
                                        10800   ; Refresh
                                        3600    ; Retry
                                        604800  ; Expire
                                        86400 ) ; Minimum TTL
    -> g3.demonet.ch.   86400   IN      NS      ns.g3.demonet.ch.
    -> g3.demonet.ch.   86400   IN      NS      ns.isbe.ch.
```

Decided to use admin.g3.demonet.ch (193.5.83.10) for zone check
Checking whether admin.g3.demonet.ch knows an A record for its own name

```
    -> admin.g3.demonet.ch.     86400   IN      CNAME   ns.g3.demonet.ch.


    -> ns.g3.demonet.ch.        86400   IN      A       193.5.83.10
```

Checking whether admin.g3.demonet.ch is able to reverse-map its own IP address
Checking connectivity with other nameservers

```
    -> www.ucw.cz.      84945   IN      CNAME   jabberwock.ucw.cz.
    -> jabberwock.ucw.cz.       84945   IN      A       212.71.128.53
    -> 53.128.71.212.in-addr.arpa.      84946   IN      PTR     jabberwock.ucw.cz.
    -> atrey.karlin.mff.cuni.cz.        600     IN      A       195.113.31.123
    -> 123.31.113.195.in-addr.arpa.     54868   IN      PTR     atrey.karlin.mff.cuni.cz.
    -> metalab.unc.edu. 84947   IN      A       152.2.210.81
    -> 81.210.2.152.in-addr.arpa.       9347    IN      PTR     metalab.unc.edu.
```

Checking mapping of localhost

```
    -> localhost.       86400   IN      A       127.0.0.1
    -> 1.0.0.127.in-addr.arpa.  86400   IN      PTR     localhost.
```

Fetching zone data for g3.demonet.ch
Parsing zone data
Checking consistency of zone records

```
    g3.demonet.ch.      86400   IN      SOA     ns.g3.demonet.ch. root.g3.demonet.ch. (
                                        2003120208      ; Serial
                                        10800   ; Refresh
                                        3600    ; Retry
                                        604800  ; Expire
                                        86400 ) ; Minimum TTL
    -> ns.g3.demonet.ch.        86400   IN      A       193.5.83.10
```

Hostmaster e-mail address is root@g3.demonet.ch

```
    -> g3.demonet.ch.   86400   IN      MX      5 ns.demonet.ch.
    -> ns.demonet.ch.   86400   IN      A       193.5.86.86
    -> 86.86.5.193.in-addr.arpa.        86400   IN      PTR     gate.demonet.ch.
    -> 86.86.5.193.in-addr.arpa.        86400   IN      PTR     ns.demonet.ch.
    -> gate.demonet.ch. 86400   IN      A       193.5.86.86
    -> gate.demonet.ch. 86400   IN      A       193.5.80.80
    -> ns.demonet.ch.   86400   IN      A       193.5.80.80
    -> 80.80.5.193.in-addr.arpa.        86400   IN      PTR     gate.demonet.ch.
    -> 80.80.5.193.in-addr.arpa.        86400   IN      PTR     ns.demonet.ch.
    -> gate.demonet.ch. 86400   IN      A       193.5.86.86
    -> gate.demonet.ch. 86400   IN      A       193.5.80.80
```

Warning: Expire time should be between 2 and 4 weeks [see RFC1912:2.2 for details]

```
    g3.demonet.ch.      86400   IN      NS      ns.g3.demonet.ch.
    -> ns.g3.demonet.ch.        86400   IN      A       193.5.83.10
    g3.demonet.ch.      86400   IN      NS      ns.isbe.ch.
    -> ns.isbe.ch.      86400   IN      A       193.5.80.20
    g3.demonet.ch.      86400   IN      MX      5 ns.demonet.ch.
    -> ns.demonet.ch.   86400   IN      A       193.5.86.86
    -> ns.demonet.ch.   86400   IN      A       193.5.80.80
    admin.g3.demonet.ch.        86400   IN      CNAME   ns.g3.demonet.ch.
    -> ns.g3.demonet.ch.        86400   IN      A       193.5.83.10
    fw.g3.demonet.ch.   86400   IN      CNAME   guardian.g3.demonet.ch.
    -> guardian.g3.demonet.ch.  86400   IN      A       193.5.83.1
    -> 1.83.5.193.in-addr.arpa. 86400   IN      PTR     guardian.g3.demonet.ch.
    guardian.g3.demonet.ch.     86400   IN      A       193.5.83.1
    gw.g3.demonet.ch.   86400   IN      CNAME   guardian.g3.demonet.ch.
    -> guardian.g3.demonet.ch.  86400   IN      A       193.5.83.1
    imap.g3.demonet.ch. 86400   IN      CNAME   ns.g3.demonet.ch.
    -> ns.g3.demonet.ch.        86400   IN      A       193.5.83.10
    luke.g3.demonet.ch. 86400   IN      A       193.5.83.20
    -> 20.83.5.193.in-addr.arpa.        86400   IN      PTR     luke.g3.demonet.ch.
    mail.g3.demonet.ch. 86400   IN      CNAME   ns.g3.demonet.ch.
    -> ns.g3.demonet.ch.        86400   IN      A       193.5.83.10
    ns.g3.demonet.ch.   86400   IN      A       193.5.83.10
    smtp.g3.demonet.ch. 86400   IN      CNAME   ns.g3.demonet.ch.
    -> ns.g3.demonet.ch.        86400   IN      A       193.5.83.10
    www.g3.demonet.ch.  86400   IN      CNAME   ns.g3.demonet.ch.
    -> ns.g3.demonet.ch.        86400   IN      A       193.5.83.10
```

4 Konfiguration DHCP-Server

4.1 Allgemein

Als DCHP-Server wurde der mitgelieferte dhcpd verwendet.

Den Dienst kann man mit

```
/etc/init.d/dhcpd [start|stop|restart]
```

manuell kontrollieren.

Um den Dienst beim Booten automatisch zu starten reicht ein einmaliges

```
chkconfig --add dhcpd
```

4.2 /etc/dhcpd.conf

```
authoritative;

# netzconfig
option broadcast-address 193.5.83.255;
option subnet-mask 255.255.255.0;
# default gw
option routers 193.5.83.1;

# namserver unseres netzwerks
option domain-name-servers 193.5.83.10;
option domain-name-servers 193.5.83.20;

ddns-update-style none;

# dynamische adressen
subnet 193.5.83.0 netmask 255.255.255.0 {
  range 193.5.83.100 193.5.83.250;
  default-lease-time 86400;
  max-lease-time 86400;
}

# statische adressen
group {
  use-host-decl-names on;

  host guardian {
    hardware ethernet 00:02:B3:0B:81:34;
    fixed-address 193.5.83.1;
  }

  host admin {
    hardware ethernet 00:90:27:57:1C:80;
    fixed-address 193.5.83.10;
  }
```

```
  host luke {
    hardware ethernet 00:90:27:57:1C:8C;
    fixed-address 193.5.83.20;
  }
}
```

Achtung: Es gilt zu beachten, dass die statischen Adressen auch im /etc/hosts eingetragen sein müssen, da der DHCP Dienst sonst die IPs nicht zuweisen kann.

Bemerkung: DynDNS haben wir nicht implementiert. Dies hat zur Folge, dass wir diejenigen Hosts, denen wir eine dynamische IP Adresse zuweisen nicht via Hostname, sondern nur via IP-Adresse ansprechen können. Für weitere Informationen zu DynDNS verweisen wir auf *man dhcpd.conf* und *man named.conf*.

4.3 DHCP Tests

Test 1	
Host:	Notebook von Christian Hauser (Windows XP)
Test:	Einstecken am vierten Netzwerk Port und IP-Adresse mittels DHCP beziehen. Ergebnis: **OK**
Ausgabe in /var/log/messages:	DHCPREQUEST for 192.168.123.107 from 00:09:6b:8d:f6:10 via eth0: wrong network. DHCPNAK on 192.168.123.107 to 00:09:6b:8d:f6:10 via eth0 DHCPDISCOVER from 00:09:6b:8d:f6:10 via eth0 DHCPOFFER on 193.5.83.248 to 00:09:6b:8d:f6:10 (t40) via eth0 DHCPREQUEST for 193.5.83.248 (193.5.83.10) from 00:09:6b:8d:f6:10 (t40) via eth0 DHCPACK on 193.5.83.248 to 00:09:6b:8d:f6:10 (t40) via eth0
tcpdump Ausgabe bei “ipconfig /release”:	193.5.83.248.netbios-ns > 193.5.83.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST 193.5.83.248.netbios-ns > 193.5.83.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST 193.5.83.248.netbios-ns > 193.5.83.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST 193.5.83.248.bootpc > admin.g3.demonet.ch.bootps: xid:0xaf096c52 flags:0x8000 C:193.5.83.248 file ""[|bootp]
tcpdump Ausgabe bei “ipconfig /renew”:	0.0.0.0.bootpc > 255.255.255.255.bootps: xid:0x9945512f file ""[|bootp] arp who-has 193.5.83.248 tell admin.g3.demonet.ch admin.g3.demonet.ch.bootps > 193.5.83.248.bootpc: xid:0x9945512f Y:193.5.83.248 S:admin.g3.demonet.ch file ""[|bootp] [tos 0x10] 0.0.0.0.bootpc > 255.255.255.255.bootps: xid:0x9945512f file ""[|bootp] admin.g3.demonet.ch.bootps > 193.5.83.248.bootpc: xid:0x9945512f Y:193.5.83.248 S:admin.g3.demonet.ch file ""[|bootp] [tos 0x10] arp who-has 193.5.83.248 tell 193.5.83.248 admin.g3.demonet.ch > 193.5.83.248: icmp: echo request (DF) 193.5.83.248 > admin.g3.demonet.ch: icmp: echo reply (DF) arp who-has 193.5.83.248 tell 193.5.83.248 arp who-has 193.5.83.248 tell 193.5.83.248 193.5.83.248.4777 > 239.255.255.250.1900: udp 133 [ttl 1] 193.5.83.248 > IGMP.MCAST.NET: igmp v3 report, 1 group record(s) [ttl 1] 193.5.83.248.4780 > 239.255.255.250.1900: udp 133 [ttl 1] 193.5.83.248.netbios-ns > 193.5.83.255.netbios-ns: NBT UDP PACKET(137): REGI-STRATION; REQUEST; BROADCAST 193.5.83.248.netbios-ns > 193.5.83.255.netbios-ns: NBT UDP PACKET(137): REGI-STRATION; REQUEST; BROADCAST 193.5.83.248 > IGMP.MCAST.NET: igmp v3 report, 1 group record(s) [ttl 1] 193.5.83.248.netbios-ns > 193.5.83.255.netbios-ns: NBT UDP PACKET(137): REGI-STRATION; REQUEST; BROADCAST 193.5.83.248.netbios-ns > 193.5.83.255.netbios-ns: NBT UDP PACKET(137): REGI-STRATION; REQUEST; BROADCAST 193.5.83.248.4780 > 239.255.255.250.1900: udp 133 [ttl 1] 193.5.83.248.netbios-ns > 193.5.83.255.netbios-ns: NBT UDP PACKET(137): REGI-STRATION; REQUEST; BROADCAST 193.5.83.248.netbios-ns > 193.5.83.255.netbios-ns: NBT UDP PACKET(137): REGI-STRATION; REQUEST; BROADCAST

5 Konfiguration Mail

5.1 Allgemein

Als Mailserver haben wir *Postfix* in Verbindung mit *imapd* verwendet. Auf POP3 Unterstützung haben wir bewusst verzichtet, da dieser Dienst unserer Meinung nach nur Nachteile bringt.

Fedora verwendet dabei die Login-Passwörter auch für den Imapdienst, somit muss nur ein Benutzer auf dem Mailserver (Rechner *admin*) erfasst sein.

5.2 Postfix

Die Konfiguration von Postfix ist sehr einfach, aber doch umfangreich, da der Server schon von Haus aus mit einer soliden Konfiguration kommt (kein Relay möglich) beschränken wir uns hier auf das Minimum :

Die Hauptdatei */etc/postfix/main.cf* wurde dabei nur minimalst geändert:

```
myhostname = smtp.g3.demonet.ch     ; Hostname setzen
mydomain = g3.demonet.ch            ; Maildomain setzen
mynetworks_style = subnet           ; Erlaubt relay auch für
                                    ; andere Computer im selben Netz
```

Des weiteren wurden noch gewisse Antispam-Checks bei eingehenden Mails implementiert, deren Erläuterung allerdings den Rahmen dieses Dokumentes sprengen würde. (Siehe Abschnitt UCE in*/etc/postfix/main.cf*)

Da per Default bei Fedora eine Sendmail Installation die Maileraufgaben übernimmt, muss man unter X erst bei den Server Services das Mailsystem auf “postfix” umschalten, bevor dieses aktiv ist.

Ein */etc/init.d/postfix start* fährt den SMTP-Dienst hoch.

5.3 Imapd

Um den Imapdienst zum laufen zu kriegen muss nur die Konfiguration des SuperServers (inetd) angepasst werden.

Dabei muss in */etc/xinet.d/imap* “**disable = no**” gesetzt werden:

```
service imap
{
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /usr/sbin/imapd
        log_on_success          += HOST DURATION
        log_on_failure          += HOST
        disable                 = no
}
```

Ein simples */etc/init.d/xinetd restart* lässt die Änderungen wirksam werden, da der Imapdienst über den Super-Server gestartet wird.

5.4 Mail Tests

Im folgenden werden wir ein paar Tests machen und dabei auch zeigen, was für TCP Pakete bei den Protokollen SMTP und IMAP ausgetauscht werden.

5.4.1 Tests mit Mail-Client

Test 1	
Host:	**luke**
Test:	Ximian starten, Mail Konfiguration für Christian Hauser <hausi@g3.demo-net.ch> erstellen und ein Mail an shauser@g3.demonet.ch senden.
Ergebnis:	**OK**

Test 2	
Host:	**luke**
Test:	Ximian starten, Mail Konfiguration für Serge Hauser <shauser@g3.demo-net.ch> auf neue Mails prüfen und das empfangene Mail an hausi@g3.demo-net.ch zurückschicken.
Ergebnis:	**OK**

Test 3	
Host:	**luke**
Test:	Ximian starten, Mail Konfiguration für Christian Hauser <hausi@g3.demo-net.ch> wählen, auf neue Mails prüfen. Das Mail von Serge Hauser muss empfangen worden sein.
Ergebnis:	**OK**

Test 4	
Host:	**plausch.szene.ch**
Test:	*tais42@plausch:~> telnet 193.5.83.10 smtp* Trying 193.5.83.10... Connected to 193.5.83.10. Escape character is '^]'. 220 smtp.g3.demonet.ch ESMTP Postfix helo smtp.g3.demonet.ch 250 smtp.g3.demonet.ch mail from: tais42@szene.ch 250 Ok rcpt to: shauser@surfkitchen.com 554 <shauser@surfkitchen.com>: Relay access denied
Ergebnis:	**OK**

5.4.2 tcpdump von SMTP

```
13:51:11.686945 193.5.83.20.1086 > 193.5.83.10.smtp: S
3429009090:3429009090(0) win 16384 <mss 1300,nop,nop,sackOK> (DF)
13:51:11.687055 193.5.83.10.smtp > 193.5.83.20.1086: S
3980582445:3980582445(0) ack 3429009091 win 5840 <mss 1460,nop,nop,sackOK>
(DF)
13:51:11.706440 193.5.83.20.1086 > 193.5.83.10.smtp: . ack 1 win 17640 (DF)
13:51:11.751956 193.5.83.10.smtp > 193.5.83.20.1086: P 1:41(40) ack 1 win
5840 (DF)
13:51:11.781247 193.5.83.20.1086 > 193.5.83.10.smtp: P 1:13(12) ack 41 win
17600 (DF)
13:51:11.781329 193.5.83.10.smtp > 193.5.83.20.1086: . ack 13 win 5840 (DF)
13:51:11.781751 193.5.83.10.smtp > 193.5.83.20.1086: P 41:147(106) ack 13
win 5840 (DF)
13:51:11.838927 193.5.83.20.1086 > 193.5.83.10.smtp: P 13:44(31) ack 147
win 17494 (DF)
13:51:11.870073 193.5.83.10.smtp > 193.5.83.20.1086: . ack 44 win 5840 (DF)
13:51:11.907182 193.5.83.10.smtp > 193.5.83.20.1086: P 147:155(8) ack 44
win 5840 (DF)
13:51:11.938109 193.5.83.20.1086 > 193.5.83.10.smtp: P 44:73(29) ack 155
win 17486 (DF)
13:51:11.938221 193.5.83.10.smtp > 193.5.83.20.1086: . ack 73 win 5840 (DF)
13:51:16.940078 193.5.83.10.smtp > 193.5.83.20.1086: P 155:222(67) ack 73
win 5840 (DF)
13:51:16.964774 193.5.83.20.1086 > 193.5.83.10.smtp: P 73:79(6) ack 222
win 17419 (DF)
13:51:16.964837 193.5.83.10.smtp > 193.5.83.20.1086: . ack 79 win 5840 (DF)
13:51:16.965239 193.5.83.10.smtp > 193.5.83.20.1086: P 222:230(8) ack 79
win 5840 (DF)
13:51:17.005539 193.5.83.20.1086 > 193.5.83.10.smtp: P 79:85(6) ack 230
win 17411 (DF)
13:51:17.006182 193.5.83.10.smtp > 193.5.83.20.1086: P 230:239(9) ack 85
win 5840 (DF)
13:51:17.007006 193.5.83.10.smtp > 193.5.83.20.1086: F 239:239(0) ack 85
win 5840 (DF)
13:51:17.045642 193.5.83.20.1086 > 193.5.83.10.smtp: . ack 240 win 17402 (DF)
13:51:22.828190 193.5.83.20.1086 > 193.5.83.10.smtp: F 85:85(0) ack 240
win 17402 (DF)
13:51:22.828280 193.5.83.10.smtp > 193.5.83.20.1086: . ack 86 win 5840 (DF)
```

5.4.3 tcpdump von IMAP

```
13:57:31.920040 193.5.83.10.imap > 193.5.83.20.1091: F 18030361:18030361(0)
ack 3508316703 win 5840 (DF)
13:57:33.281120 193.5.83.20.1094 > 193.5.83.10.imap: S 3524497935:3524497935
(0) win 16384 <mss 1300,nop,nop,sackOK> (DF)
13:57:33.281219 193.5.83.10.imap > 193.5.83.20.1094: S 89846252:89846252(0)
ack 3524497936 win 5840 <mss 1460,nop,nop,sackOK> (DF)
13:57:33.302943 193.5.83.20.1094 > 193.5.83.10.imap: . ack 1 win 17640 (DF)
13:57:33.398601 193.5.83.10.imap > 193.5.83.20.1094: P 1:142(141) ack 1 win
5840 (DF)
13:57:33.445086 193.5.83.20.1094 > 193.5.83.10.imap: P 1:20(19) ack 142 win
17499 (DF)
13:57:33.445372 193.5.83.10.imap > 193.5.83.20.1094: . ack 20 win 5840 (DF)
13:57:33.445791 193.5.83.10.imap > 193.5.83.20.1094: P 142:329(187) ack 20 w
in 5840 (DF)
13:57:33.472029 193.5.83.20.1094 > 193.5.83.10.imap: P 20:51(31) ack 329 win
```

```
 17312 (DF)
13:57:33.510025 193.5.83.10.imap > 193.5.83.20.1094: . ack 51 win 5840 (DF)
13:57:33.533295 193.5.83.10.imap > 193.5.83.20.1094: P 329:481(152) ack 51 w
in 5840 (DF)
13:57:33.557584 193.5.83.20.1094 > 193.5.83.10.imap: P 51:71(20) ack 481 win
 17160 (DF)
13:57:33.557785 193.5.83.10.imap > 193.5.83.20.1094: . ack 71 win 5840 (DF)
13:57:33.558279 193.5.83.10.imap > 193.5.83.20.1094: P 481:507(26) ack 71 wi
n 5840 (DF)
13:57:33.602520 193.5.83.20.1094 > 193.5.83.10.imap: P 71:139(68) ack 507 wi
n 17134 (DF)
13:57:33.603359 193.5.83.10.imap > 193.5.83.20.1094: P 507:615(108) ack 139
win 5840 (DF)
13:57:33.644602 193.5.83.20.1094 > 193.5.83.10.imap: P 139:162(23) ack 615 w
in 17026 (DF)
13:57:33.645263 193.5.83.10.imap > 193.5.83.20.1094: P 615:853(238) ack 162
win 5840 (DF)
13:57:33.670640 193.5.83.20.1094 > 193.5.83.10.imap: P 162:188(26) ack 853 w
in 16788 (DF)
13:57:33.670954 193.5.83.10.imap > 193.5.83.20.1094: P 853:895(42) ack 188 w
in 5840 (DF)
13:57:33.691952 193.5.83.20.1094 > 193.5.83.10.imap: P 188:203(15) ack 895 w
in 16746 (DF)
13:57:33.692303 193.5.83.10.imap > 193.5.83.20.1094: P 895:977(82) ack 203 w
in 5840 (DF)
13:57:33.694044 193.5.83.10.imap > 193.5.83.20.1094: F 977:977(0) ack 203 wi
n 5840 (DF)
13:57:33.718997 193.5.83.20.1094 > 193.5.83.10.imap: F 203:203(0) ack 977 wi
n 16664 (DF)
13:57:33.719084 193.5.83.10.imap > 193.5.83.20.1094: . ack 204 win 5840 (DF)
13:57:33.950033 193.5.83.10.imap > 193.5.83.20.1094: F 977:977(0) ack 204 wi
n 5840 (DF)
13:57:34.430018 193.5.83.10.imap > 193.5.83.20.1094: F 977:977(0) ack 204 wi
n 5840 (DF)
13:57:35.390023 193.5.83.10.imap > 193.5.83.20.1094: F 977:977(0) ack 204 wi
n 5840 (DF)
13:57:37.310025 193.5.83.10.imap > 193.5.83.20.1094: F 977:977(0) ack 204 wi
n 5840 (DF)
13:57:39.800027 193.5.83.10.imap > 193.5.83.20.1092: F 55403872:55403872(0)
ack 3518013593 win 5840 (DF)
13:57:41.150021 193.5.83.10.imap > 193.5.83.20.1094: F 977:977(0) ack 204 wi
n 5840 (DF)
13:57:48.830027 193.5.83.10.imap > 193.5.83.20.1094: F 977:977(0) ack 204
win 5840 (DF)
```

5.4.4 SMTP in Aktion

Lukas Reusser hat uns noch folgendes Bild (Abbildung 5) zur Verfügung gestellt, das zwar nicht auf unserem Netzwerk für dieses IIN Projekt erstellt worden ist, jedoch sehr schön das SMTP Protokoll in Aktion zeigt.

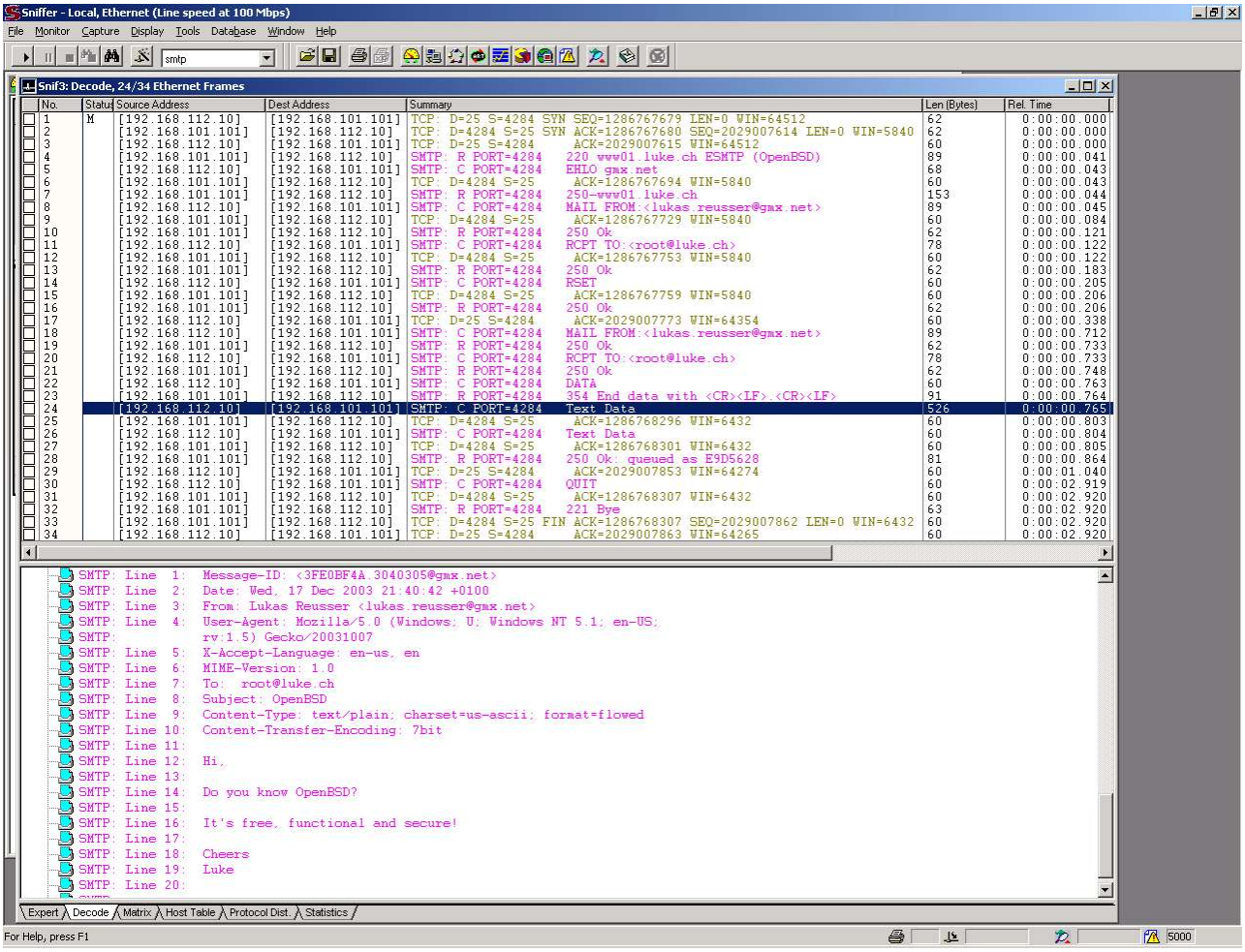

Abbildung 5: SMTP in Aktion mit SnifferPro 4.7 gesnifft

6 Konfiguration Apache Web-Server

Auf unserem Rechner *admin* haben wir noch Apache 2 (*httpd*) als Webserver aufgesetzt, jedoch ohne Virtual Hosts. Im Folgenden erläutern wir unser Vorgehen, möchten jedoch noch darauf hinweisen, dass wir von Apache sprechen und eigentlich *httpd* (also Apache 2) meinen.

6.1 Apache Installation

Da wir schon während der Installation von Fedora Linux den Webserver Apache ausgewählt haben, mussten wir diesen nicht mehr installieren.

6.2 /etc/httpd/conf/httpd.conf

Wir mussten in der Datei **/etc/httpd/conf/httpd.conf** ein paar Veränderungen vornehmen, die wir im untenstehenden Ausschnitt zeigen.

```
...

# ServerAdmin: Your address, where problems with the server should be
# e-mailed.  This address appears on some server-generated pages, such
# as error documents.  e.g. admin@your-domain.com
#
ServerAdmin webmaster@g3.demonet.ch

# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If this is not set to valid DNS name for your host, server-generated
# redirections will not work.  See also the UseCanonicalName directive.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
# You will have to access it by its address anyway, and this will make
# redirections work in a sensible way.
#
ServerName www.g3.demonet.ch:80

...

# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/var/www/html"

...
```

6.3 Web-Content: /var/www/html

Damit der Webserver auch etwas anzeigen kann, müssen entsprechende HTML (oder sonstige) Dateien im Verzeichnis **/var/www/html** gespeichert werden. In unserem Projekt haben wir eine einfache HTML Datei **index.html** erstellt und im Verzeichnis /var/www/html gespeichert.

6.4 Apache Webserver starten und stoppen

Der Apache Webserver kann wie folgt gestoppt, gestartet oder neu gestartet werden:

```
service httpd [start|stop|restart]
```

6.5 Apache Webserver testen

Zum Testen haben wir uns an einen anderen Computer (in einem anderen VLAN) gesetzt und im Browser die URL zu unserem Webserver eingegeben: http://www.g3.demonet.ch und erhielten prompt die Seite angezeigt.

6.5.1 HTTP in Aktion: HTTP-Request

Lukas Reusser hat uns noch folgendes Bild (Abbildung 6) zur Verfügung gestellt, das einen HTTP-Request zeigt:

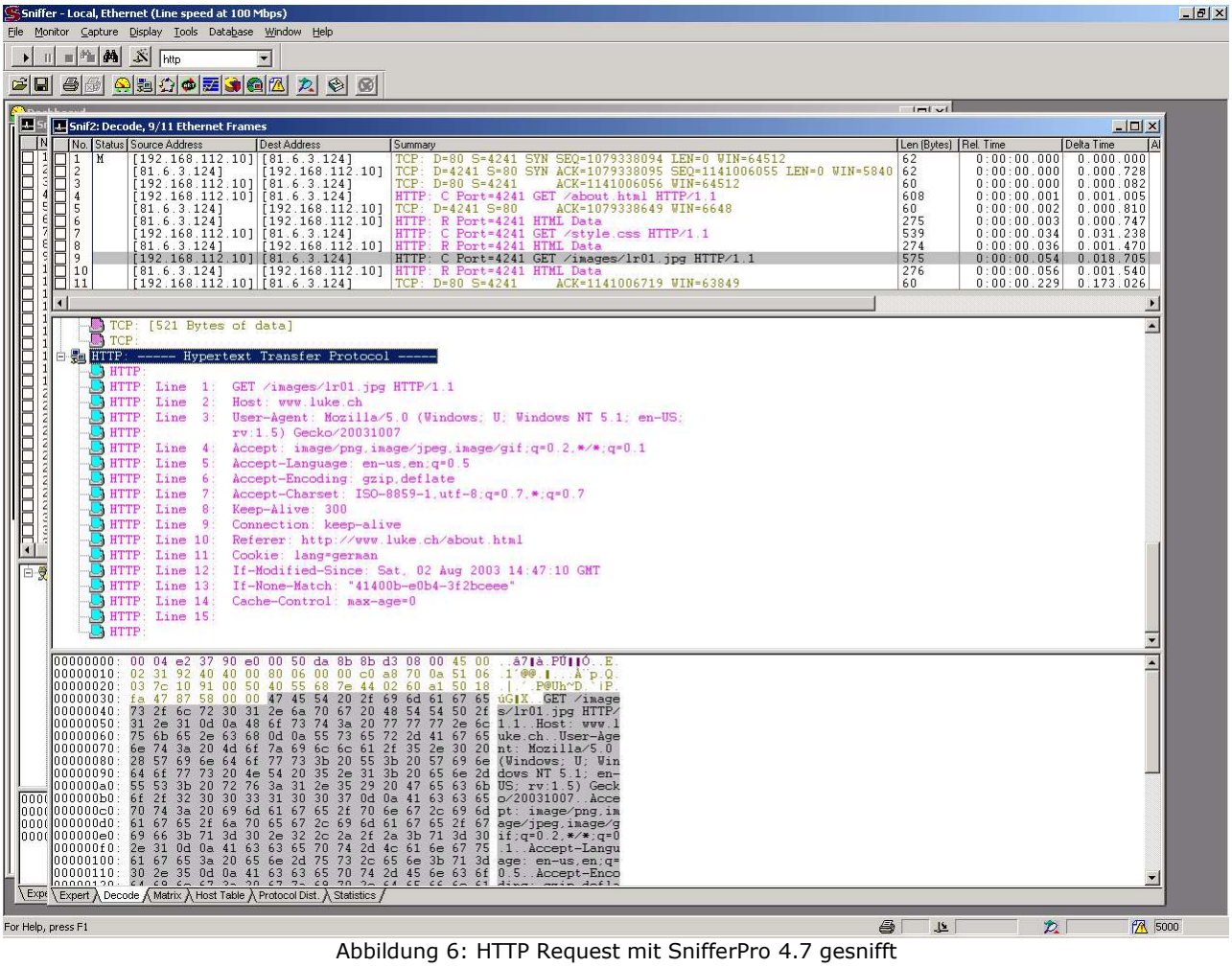

Abbildung 6: HTTP Request mit SnifferPro 4.7 gesnifft

7 Installation Java, Ant und Tomcat

Es gibt diverse Gründe, auf einem Server Java zu installieren. Wir haben uns entschieden Java zu installieren, weil wir Tomcat aufsetzen wollten.

7.1 Java

Java ist eine mächtige, objektorientierte und plattformunabhängige Programmiersprache. Für weitere Informationen verweisen wir auf *http://java.sun.com/*.

7.1.1 Vorbereitung

Wir wollen Java im Verzeichnis **/opt/java** installieren. Dafür öffnen wir mit dem Befehl “su -” eine root-Shell und wechseln ins Verzeichnis /opt:

[root@admin]# cd /opt

Daraufhin erstellen wir ein Verzeichnis /opt/java:

[root@admin:/opt]# mkdir java

7.1.2 Download von Java

1.4.2 (aktuellste Version beim Schreiben dieses Dokuments: 1.4.2_03) von *http://java.sun.com/* herunterladen. Für die Linux Plattform ist dies konkret das self-extracting File *j2sdk-1_4_2_03-linux-i586.bin* (ca. 35MB), das heruntergeladen werden muss.

j2sdk-1_4_2_03-linux-i586.bin ins Verzeichnis /opt/java kopieren und als root mit execute Rechten versehen:

[root@admin:/opt/java]# chmod u+x ./j2sdk-1_4_2_03-linux-i586.bin

7.1.3 Installation von Java

Als root User im Verzeichnis /opt/java die bin Datei ausführen:

[root@admin:/opt/java]# ./j2sdk-1_4_2_03-linux-i586.bin

Es kommt ein License Agreement, bei dem man mit der Leertaste nach unten geht und am Schluss "yes" eingibt, sodass die Dateien in ein Verzeichnis "j2sdk1.4.2_03" ausgepackt werden.

Daraufhin sollte man einen symbolischen Link mit dem Namen java erstellen, der auf das aktuellste JDK zeigt:

[root@admin:/opt/java]# ln -s j2sdk1.4.2_03 java

Original-Datei löschen:

[root@admin:/opt/java]# rm -f j2sdk-1_4_2_03-linux-i586.bin

Nun muss noch die Datei */etc/profile* angepasst werden.

Als erstes muss die globale Environment-Variable JAVA_HOME gesetzt werden. Dann muss noch der Pfad so ergänzt werden, dass die binary files von Java gefunden werden können:

[root@admin:/opt/java]# vi /etc/profile

```
export JAVA_HOME=/opt/java/java
export PATH=$PATH:$JAVA_HOME/bin
```

7.1.4 Java testen

Um zu testen, ob Java läuft, folgendes eingeben:

[root@admin]# java -version

Die Ausgabe sollte in etwa so aussehen:

```
java version "1.4.2_03"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.2_03-b02)
Java HotSpot(TM) Client VM (build 1.4.2_03-b02, mixed mode)
```

Um zu Testen, ob der Java Compiler läuft, folgendes eingeben:

[root@admin]# javac -help

Die Ausgabe sollte in etwa so aussehen:

```
Usage: javac <options> <source files>
where possible options include:
  -g                         Generate all debugging info
  -g:none                    Generate no debugging info
  -g:{lines,vars,source}     Generate only some debugging info
  -nowarn                    Generate no warnings
  -verbose                   Output messages about what the compiler is doing
  -deprecation               Output source locations where deprecated APIs are
used
  -classpath <path>          Specify where to find user class files
  -sourcepath <path>         Specify where to find input source files
  -bootclasspath <path>      Override location of bootstrap class files
  -extdirs <dirs>            Override location of installed extensions
  -d <directory>             Specify where to place generated class files
  -encoding <encoding>       Specify character encoding used by source files
  -source <release>          Provide source compatibility with specified
release
  -target <release>          Generate class files for specific VM version
  -help                      Print a synopsis of standard options
```

7.2 Ant

Ant ist ein sehr mächtiges Build-Tool (wie z.B. *make*), das vorallem in der Java-Welt sehr weit verbreitet ist.

7.2.1 Installation von Ant

Mit “su -” als root einloggen und ins Verzeichnis /opt wechseln.

Dort ein Verzeichnis ant erstellen:

[root@admin:/opt]# mkdir ant

Die binary Distribution von Ant von *http://ant.apache.org/* downloaden.

Aktueller Download beim Schreiben dieser Dokumentation ist Version 1.5.4.
Filename: *apache-ant-1.5.4-bin.tar.gz*

Die Datei apache-ant-1.5.4-bin.tar.gz ins Verzeichnis /opt/ant kopieren

Die Datei auspacken:
[root@admin:/opt/ant]# tar -xzvf apache-ant-1.5.4-bin.tar.gz
Es wird ein Verzeichnis apache-ant-1.5.4 erstellt.

Nun noch einen symbolischen Link auf dieses Verzeichnis erstellen:
[root@admin:/opt/ant]# ln -s apache-ant-1.5.4 ant

Die Original-Datei kann nun gelöscht werden:
[root@admin:/opt/ant]# rm -f apache-ant-1.5.4-bin.tar.gz

Wieder die Datei */etc/profile* anpassen:
[root@admin:/opt/ant]# vi /etc/profile

```
export ANT_HOME=/opt/ant/ant
export PATH=$PATH:$ANT_HOME/bin
```

7.3 Tomcat

Tomcat ist ein (in Java geschriebener) Web-Container, der mit mit Apache verbunden werden kann und dann die dynamischen Web-Inhalte (JSPs und Servlets) abarbeiten kann.

7.3.1 Download von Tomcat

Tomcat binary release von *http://jakarta.apache.org/tomcat/* downloaden.

Wir haben uns für die aktuellste Version 5.0.16 entschieden und somit das File *jakarta-tomcat-5.0.16.tar.gz* heruntergeladen.

7.3.2 Installation von Tomcat

Mit “su -” als root einloggen und ins Verzeichnis /opt wechseln. Ein Verzeichnis tomcat erstellen und in dieses Verzeichnis wechseln:

[root@admin]# cd /opt

[root@admin:/opt]# mkdir tomcat

[root@admin:/opt]# cd tomcat

Dann die heruntergeladene Datei ins aktuelle (/opt/tomcat) Verzeichnis kopieren und mit folgendem Befehl auspacken:

[root@admin:/opt/tomcat]# tar -xzvf jakarta-tomcat-5.0.16.tar.gz

Daraufhin wird ein neues Verzeichnis jakarta-tomcat-5.0.16 erstellt. Die ausgepackte Datei kann daraufhin gelöscht werden:

[root@admin:/opt/tomcat]# rm -f jakarta-tomcat-5.0.16.tar.gz

Nun noch einen symbolischen Link erstellen:

[root@admin:/opt/tomcat]# ln -s jakarta-tomcat-5.0.16 tomcat

Wiederum die Datei */etc/profile* editieren:

[root@admin:/opt/tomcat]# vi /etc/profile

```
export PATH=$PATH:/opt/tomcat/tomcat/bin
```

7.3.3 Tomcat testen

Testen, ob der Tomcat läuft:

Ins Verzeichnis /opt/tomcat/tomcat/bin wechseln und Tomcat wie folgt starten:

[root@admin]# cd /opt/tomcat/tomcat/bin

[root@admin:/opt/tomcat/tomcat/bin]# ./startup.sh

Danach einen Browser öffnen und auf die folgende URL gehen:

http://localhost:8080/

Wenn eine Seite angezeigt wird, läuft der Server einwandfrei. Falls dies nicht der Fall ist, sollte eigentlich eine Fehlermeldung angezeigt worden sein beim starten des Tomcat.

Um zu testen, ob auch wirklich dynamischer Content angezeigt werden kann, empfiehlt es sich eine kleine Web-Applikation mit einer einfachen JSP Seite zu erstellen.

Dazu müssen wir ein neues Verzeichnis für die Web-Applikation erstellen, gemäss J2EE Spezifikation und dann dort drin ein JSP File unterbringen:

/opt/tomcat/tomcat/webapps/test/jsp/test.jsp

Web-Applikationen benötigen noch einen Deployment Descriptor (web.xml):

/opt/tomcat/tomcat/webapps/test/WEB-INF/web.xml

Und hier noch das detailierte Vorgehen:

[root@admin]# cd /opt/tomcat/tomcat/webapps/

[root@admin:/opt/tomcat/tomcat/webapps]# mkdir test

[root@admin:/opt/tomcat/tomcat/webapps]# cd test

[root@admin:/opt/tomcat/tomcat/webapps/test]# mkdir jsp

[root@admin:/opt/tomcat/tomcat/webapps/test]# vi jsp/test.jsp

```
<%@ page import="java.util.*, java.text.*" %>
<html>
<head><title>Aktuelles Datum und  Zeit</title></head>
<body>
<%
  SimpleDateFormat datumFormat = new SimpleDateFormat("dd.MM.yyyy");
  SimpleDateFormat zeitFormat = new SimpleDateFormat("HH:mm:ss");
  Date date = new Date();
  String datum = datumFormat.format(date);
  String zeit = zeitFormat.format(date);
%>
<h3>Datum: <%= datum %></h3>
<h3>Zeit: <%= zeit %></h3>
</body>
</html>
```

[root@admin:/opt/tomcat/tomcat/webapps/test]# mkdir WEB-INF

[root@admin:/opt/tomcat/tomcat/webapps/test]# vi WEB-INF/web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee web-app_2_4.xsd"
    version="2.4">

    <description>
            Test Web Applikation
    </description>
    <display-name>
            Test Web Applikation
    </display-name>

</web-app>
```

Daraufhin muss der Tomat neu gestartet werden:

[root@admin]# /opt/tomcat/tomcat/bin/shutdown.sh

[root@admin]# /opt/tomcat/tomcat/bin/startup.sh

Das Testen der Web-Applikation ist denkbar einfach. Einfach in einem Browser die folgende URL eingeben:

http://localhost:8080/test/jsp/test.jsp

Es sollte eine HTML-Seite angezeigt werden, die das aktuelle Datum anzeigt.

7.3.4 Verbinden von Apache httpd und Tomcat

Das Problem das wir jetzt haben ist jedoch, dass wir nun zwar Apache (httpd) auf Port 80 und Tomcat auf Port 8080 am laufen haben, doch was wir wirklich wollen wäre eine Kombination beider Server auf Port 80. Wobei der Apache den statischen und der Tomcat den dynamischen (Servlets, JSPs) Web-Content liefern sollte.

Um dies zu erreichen gibt es den **mod_jk2** Connector. Der jedoch sehr schwierig einzurichten ist. Selbst mit einer exzellenten Dokumentation gelang es uns bislang nicht mod_jk2 zum laufen zu bringen.

Es gab noch einen anderen Connector: **mod_webapp**. Dieser ist zwar viel einfacher zu installieren, jedoch nicht so performant und bietet auch kein load-balancing. Da mod_webapp mit erscheinen von Tomcat 4.1.23 deprecated wurde sollte er sowieso nicht mehr verwendet werden.

Das heisst konkret, dass wir im Praktikum keine Verbindung von Apache und Tomcat realisiert haben.

8 Referenzen und Verweise

Hier auf die wichtigsten RFCs oder sonstigen Online-Dokumente verweisen.

DNS Sleuth:

http://atrey.karlin.mff.cuni.cz/~mj/sleuth/

RedHat Fedora Project:

http://fedora.redhat.com/

Java:

http://java.sun.com/

Ant:

http://ant.apache.org/