Projekt „Schul-IT“

Anti-Viren und Anti-Spam-Lösung

Dozent:

Herr Prof. Dr. Hellberg

Autoren:

Florian Bredlow

Rainer Irion

Sebastian Breithor

Studienkurs:

Betriebssysteme und Netze

4. Theoriequartal

HFI407

Abgabedatum:

06.07.2009

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

ClamAV…… Clam Antivirus
DCC…….. Distributed Checksum Clearinghouse
DNS…..…… Domain Name System
EICAR……..……. European Institute for Computer Antivirus Research
E-Mail…… Electronic Mail
FTP……..…….. File Transfer Protocol
GNU……... GNU's Not Unix
GPL…….. GNU General Public License
GTUBE……. Generic Test for Unsolicited Bulk Email
GUI……. Graphical User Interface
HELO/EHLO..…… Hello / Extended Hello
HTTP……. Hypertext Markup Language
IMAP……. Internet Message Access Protocol
IP……... Internet Protocol
IT…… Information Technology
KDE…….. Kool Desktop Environment
LDAP……. Lightweight Directory Access Protocol
MDA……... Mail Delivery Agent
MRA…….. Mail Retrieval Agent
MTA…….. Mail Transfer Agent
MUA…….. Mail User Agent
Phishing……..……... Password Harvesting Fishing
POP……... Post Office Protocol
RBL……Realtime Blackhole List
RFC……..… Request For Commons
RPM…….. Red Hat Package Manager
SASL……. Simple Authentication and Security Layer
SMTP…… Simple Mail Transfer Protocol
SMTP-AUTH…… Simple Mail Transfer Protocol Authentication
SSL…….. Secure Sockets Layer
TLS…….. Transport Layer Security
UBE…….. Unsolicited Bulk E-Mail
UCE…….. Unsolicited Commercial E-Mail
URL……. Uniform Resource Locator
VM…… Virtual Machine
YAST……. Yet Another Setup Tool

1. Einleitung

Das Projekt „Schul-IT“ wird durch die Fachhochschule der Wirtschaft (FHDW), maßgeblich durch Herrn Professor Dr. Hellberg, betrieben. Es verfolgt die Aufgabe, eine einfache, kostengünstige und wirksame IT-Infrastruktur in Schulen und Fachhochschulen zu schaffen, siehe [HEL09a].

Eine besondere Rolle in diesen Infrastrukturen spielt Sicherheit. Werden die Netzwerke mit einem Mailserver betrieben, können sich Schadprogramme aus dem Internet insbesondere via E-Mail einschleusen. Es müssen geeignete Gegenmaßnahmen getroffen werden.

Das Ziel dieses Projektes ist es, einen virtuellen Mailserver einzurichten und diesen mit einem zentralen Schutz gegen Viren auszustatten, so dass zwischengespeicherte E-Mails auf dem Server gelesen, Dateianhänge dekodiert und als infiziert erkannte Mails erst gar nicht weitergeleitet werden.

Ein weiteres Hauptaugenvermerk ist auf Spam-Mails gerichtet, die mittlerweile einen Großteil des E-Mail-Verkehrs ausmachen. Durch den Einsatz von Spam-Filtern lässt sich das Mailvolumen reduzieren, so können Spam-Mails bereits bei der Zustellung gestoppt werden, so dass sie erst gar nicht in die Postfächer der Empfänger gelangen.

Der Kostenpunkt spielt in diesem Projekt grundsätzlich eine tragende Rolle, da bei einer Schule mit über 1000 Schülern eine enorme Summe an Lizenzgebühren entstehen würden, wenn keine Open-Source-Software genutzt wird – Geld, dass z.B. in bessere Hardware investiert werden kann.

Die Motivation des Projektes ist es somit, eine flexible, wirksame und kostenfreie Anti-Viren- bzw. Anti-Spam-Lösungen für das Projekt „Schul-IT“ einzurichten. Die Effektivität dieser Anti-Viren- / Anti-Spam-Lösung wird durch bewährte Methoden getestet.

Schließlich wird auf die Perspektiven der gewählten Anwendungsprogramme eingegangen, um geplante Features in der Zukunft und ein mögliches weiteres Vorgehen mit den Softwarelösungen zu skizzieren.

2. Theoretischer Hintergrund

Dieses Kapitel umfasst die theoretischen Hintergründe der Projektarbeit und die Anforderungen, die an dieses Projekt gestellt werden.

2.1 Projekt Schul-IT

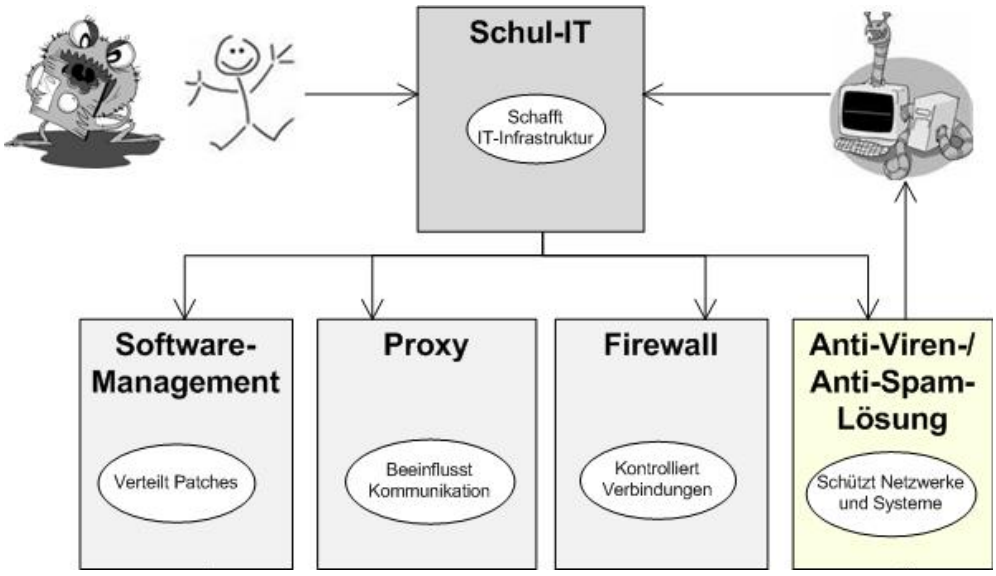

Abb. 1 - Umgebung des Schul-IT Projektes

Das Projekt “Schul-IT” verfolgt das Konzept, eine einfache IT-Infrastruktur in Schulen und Fachhochschulen bereitzustellen, siehe [HEL09a]. Die Netzwerke sollen durch virtuelle Server (Mailserver, LDAP-Server, DNS-Server,...) betrieben werden. Dadurch ist es möglich, einzelne Themenstellungen des Projektes „Schul-IT“, wie in Abb.1 dargestellt, unabhängig von anderen Teilprojekten zu realisieren.

Das Thema dieses Projektes umfasst die „Anti-Viren/Anti-Spam-Lösung“, die die Einrichtung eines E-Mailservers benötigt, auf dem die Software angewandt werden soll. Während sich der Spam-Filter auf den E-Mail-Verkehr beschränkt, soll die Anti-Viren-Lösung hauptsächtlich auf dem virtuellen Server laufen.

Die zur Verfügung gestellten Rechner für die Klienten sind auch für den privaten Gebrauch vorgesehen. Daher werden Anti-Viren-Programme auf diesen Maschinen separat laufen.

Die Schnittstellen zu den anderen Teilprojekten müssen wie folgt berücksichtigt werden:

- SOFTWARE MANAGEMENT: sollen Anti-Viren-Programme bzw. Virensignaturen mittels Patchverteilung aktualisiert werden?
- PROXY: soll der Zugriff auf den Mailserver über den Proxy-Server laufen
- FIREWALL: welche Ports dürfen nicht geblockt werden, um einen reibungslosen E-Mail-Verkehr zu gewährleisten

2.2 Mailserver

Zum Testen der Anti-Viren bzw. Anti-Spam-Lösung ist ein Mailserver erforderlich. Zur Einrichtung eines funktionsfähigen Mailservers werden folgende Dienste benötigt:

- Mail User Agent (MUA), zum Abruf und Versand von E-Mails
- Mail Delivery Agent (MDA), zur Ablage von E-Mails in den Postfächern, der IMAP bzw. POP3 verwendet
- Mail Retrieval Agent (MRA), holt E-Mails vom MDA ab und speichert sie auf dem lokalen Rechner
- Mail Transfer Agent (MTA), zum Transport von E-Mails, der das SMTP benutzt

Der Transport von E-Mails wird somit über mehrere Schnittstellen realisiert. Die aufgeführten Dienste sollen zusammen mit Mailfiltern, i.e. Blacklists, Spam-Filtern und Anti-Virus-Programmen betrieben werden. Der Zusammenspiel der Dienste ist in Abb. 2 veranschaulicht.

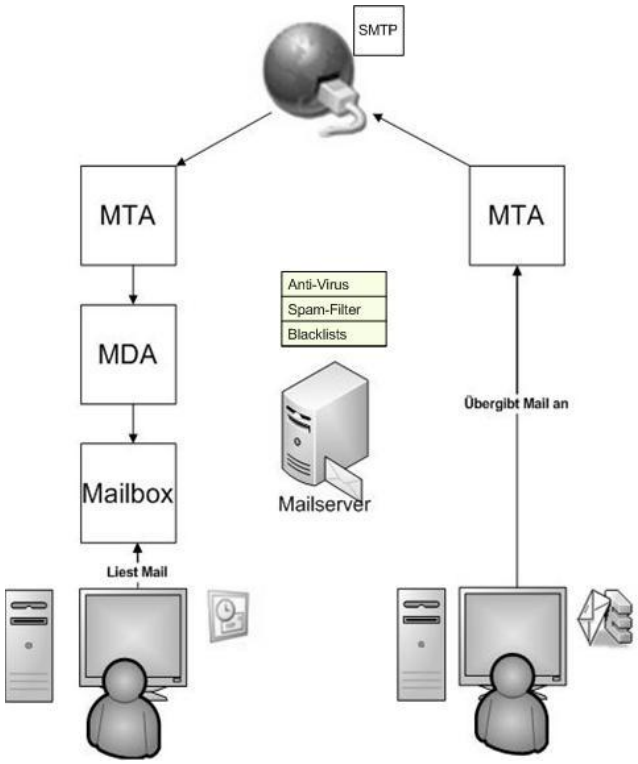

Abb. 2 – Zentrale Dienste eines Mailservers

Wichtig ist es, dass der Server ausschließlich Mails an eigene User oder von eigenen Usern annimmt. Anderenfalls würde man ein sogenanntes „OpenRelay“ betreiben, einen klassischen Spam-Server, der Mails von fremden Usern an andere fremde User weiterleitet.

2.3 Anti-Spam-Lösung

Lästige Spam-Mails, die unnötig Ressourcen blockieren und verbrauchen, stellen im Lauf der letzten Jahre ein immer größeres Problem dar.

Microsoft-Gründer Bill Gates erklärte nach einer ersten Spam-Schwemme im Jahr 2003, innerhalb von 2 Jahren sei das Spamproblem grundsätzlich gelöst [ATI09]. Das Unternehmen MessageLabs ermittelte bis 2006 jedoch einen Anstieg des Spams-Durchschnitts auf ca. 59% des Mailaufkommens. Die Fakultät für Informatik der Uni Karlsruhe verzeichnete dann jedoch schon 190 000 Spammails bei einem Gesamtvolumen von 240 000 Mails pro Woche (ca. 80%). Auch zeitweise sinkende Spamraten wie ab Sommer 2007 können nicht den Trend nachhaltig beeinflussen, siehe [HEI09].

In einem Artikel auf Zdnet.de vom 27.05.2009 überschritt das weltweite Spamaufkommen laut einer Untersuchung von Symantec im Mai`09 die 90% Marke und stieg gegenüber dem Durchschnitt des letzten halben Jahres um 12,5 Prozentpunkte an.

Die folgende Graphik zeigt die Entwicklung des Anteils der Spammails im Verhältnis zu den „richtigen“ Mails (Ham) und zum Gesamtaufkommen an der Uni Karlsruhe seit 2003 pro Woche. Die grüne Kurve stellt die gewünschten E-Mails dar, die blaue Kurve indes den Spam-Anteil, i.e. die versandten unerwünschten E-Mails. Die rote Kurve zeigt das Gesamtvolumen aller empfangenen E-Mails.

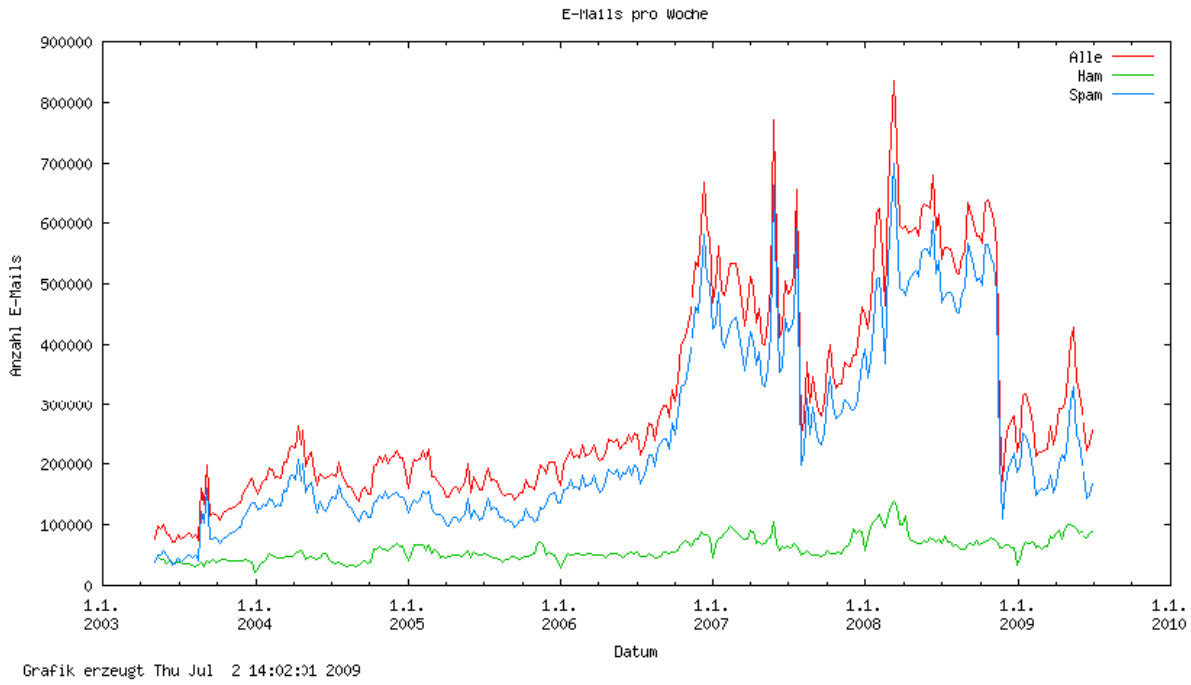

Abb. 3 – Spam-Aufkommen seit 2003

Dienstleister, wie z.B. die Abteilung Technische Infrastruktur (ATIS) der Universität Karlsruhe, kontrollieren den zentralen Mailserver der Fakultät für Informatik mit der Open-Source-Lösung Spamassassin auf Spam.

Um das Mailvolumen eines Mailservers zu reduzieren, können Spam-Filter eingesetzt werden. Ein solches Filter-Programm ist „SpamAssassin“, mit dem Spam automatisch aussortiert werden kann [SPA09].

SpamAssassin kann an jeder Stelle der Mailverarbeitungskette eingesetzt werden, z.B. auf Benutzer-Ebene als Plugin im E-Mail Programm. SpamAssassin schätzt nach bestimmten Regeln Punkte, wie hoch eine Spamwahrscheinlichkeit ist. Bei Überschreiten wird eine E-Mail als Spam markiert und kann direkt gelöscht, annahmeverweigert, in spezielle Spamordner abgelegt oder mit Warn-Betreff versehen werden. Diese Regeln basieren auf:

- Abfrage von Blacklists von spam-versendenden Servern, Realtime Blackhole Lists (RBLs)
- Abfrage von auf Prüfsummen basierten Filtern wie z.B. DCC oder Razor
- Integrierter „Bayesscher Filter“, der aufgrund der Einteilung der bisher empfangen Mails statistisch die Wahrscheinlichkeit berechnet, ob die Mail erwünscht ist oder nicht

2.4 Anti-Viren-Lösung

Die Nutzung des Internets ist heute nicht mehr ohne ausreichenden Virenschutz möglich, da schon Sekunden nach dem Einloggen von Hackern versucht wird, über fremde PC`s die Kontrolle zu erlangen. Bei Erfolg werden dann von diesen aus wieder weitere Rechner attackiert.

„ClamAV“ ist ein unter der GPL stehender Virenscanner und Phishing-Filter [CLA09]. Er kann in eigene Software integriert werden und den Zugang zu Daten automatisch sperren oder zulassen. Somit kann das kostenlose ClamAV auch auf E-Mail-Systemen mit openSUSE zur Ausfilterung von Mailwürmern und Phishing zum Einsatz kommen und eignet sich daher besonders für dieses Projekt.

Während der Spam-Filter ausschließlich im Mailverkehr angewandt werden soll, kann die Anti-Viren-Software auch für andere Zwecke in Betracht gezogen werden, z.B. für den manuellen Scan von Dateien, Programmen, dem Arbeitsspeicher und evtl. dem HTTP Verkehr.

3. Installation und Konfiguration

Dieses Kapitel beschreibt die Einrichtung der virtuellen Umgebung und die des Mailservers. Darüberhinaus werden die notwendigen Schritte aufgeführt, um die Anti-Viren-Software „ClamAV“ und den Spam-Filter „SpamAssassin“ in dieser Umgebung zu integrieren.

3.1 Virtuelle Maschine

Eine Virtual Machine (VM) ist ein virtueller Computer, der auf einem realen Computer betrieben werden kann. Dazu wird zunächst eine Virtualisierungssoftware benötigt. In den verschiedenen Teilprojekte wurde die Applikation „VMWare Server v1.0.8“ [VMW09] verwendet, um Konflikte bei einer abschließenden Integration zu vermeiden.

Die Server des Projektes „Schul-IT“ basieren auf einer 64-Bit-Version der openSUSE Linux Distribution 11.0. Sie basieren nicht auf 32-Bit-Systemen, da in diesen nur maximal 4 GB Arbeitsspeicher (= 2 Byte ^ 32) adressiert werden können. Für den Prototypen dieses Projektes reicht ein Server auf 32-Bit-Basis jedoch aus.

Zur Installation des virtuellen Servers wurden die vereinbarten Installationsanweisungen befolgt, vgl. [HEL09b]. Als grafische Benutzerfläche wurde KDE 3.5 gewählt, da es weniger Ressourcen als KDE 4.0 in Anspruch nimmt und stabil ist. Nach der Grundinstallation mit Kernelquellen, Entwicklungssystem, Netzwerkserver, etc. wurden die zur Verfügung stehenden Online-Updates durchgeführt.

Für Testzwecke und zur Überprüfung des Mailverkehres wurden mehrere Benutzerkonten lokal eingerichtet: Dazu lässt man in der Konsole folgende Befehle laufen:

```
su
```

- um Prozesse mit den Rechten des Superusers „root“ durchzuführen, sowie:

```
yast2
```

- um YAST, ein Installations- und Konfigurationswerkzeug, zu starten.

Unter dem YAST-Menüpunkt „`Sicherheit und Benutzer >> Benutzer- und Gruppenmanagement`“ lassen sich neue Benutzerkonten einrichten. Das Benutzerkonto des Administrators lautet „`admin`“. Lokale Testkonten sind user „`bernie`“ und user „`ert`“. Das Passwort für jedes Konto lautet „`aaaaa`“.

Mittels des YAST-Menüpunktes „`Netzwerkgeräte >> Netzwerkverbindungen`“ wurde dann, wie vereinbart [HEL09c], eine feste IP-Adresse (10.0.0.101) und Subnetzmaske (255.255.255.0) eingerichtet.

3.2 Mail Transfer Agent (Postfix)

Für die Installation des MTAs kommen in openSUSE Linux v.a. Sendmail und Postfix in Frage. Da Postfix schneller und einfacher zu administrieren ist, und einen zusätzlichen Sicherheitsaspekt bietet,siehe [POS09], fiel die Entscheidung auf diesen Agenten. Über Postfix wird E-Mail lokal den entsprechenden Postfächern zugeordnet und extern über SMTP versandt.

Zur Installation von Postfix wird in der Konsole unter Administratorprivilegien der folgende Befehl eingegeben:

```
yast2 -i postfix
```

Damit Postfix bei jedem Systemstart automatisch aufgerufen wird, wird ein System Startup Link wie folgt hinzugefügt:

```
chkconfig –add postfix
/etc/init.d/postfix start
```

Die Datei „`/etc/postfix/main.cf`“ dient zur Konfiguration des MTAs. Infos über Einstellungen, die man vornehmen kann, findet man darüberhinaus mit dem Befehl:

```
Postconf
```

Die wichtigsten Einträge in „`/etc/postfix/main.cf`“ sind:

- `smtpd_client_restrictions = hash:/etc/postfix/access`
  beschränkt die Klienten, von denen man SMTP akzeptiert. Standard: alle SMTP Verbindungen sind erlaubt
- `smtpd_sender_restrictions = hash:/etc/postfix/access`
  verweist auf „`/etc/postfix/access`“, um Absenderadressen, die das System akzeptiert, einzuschränken
- `smtpd_recipient_restrictions = check_relay_domains`
  schränkt die akzeptierten Rezipienten des Systems ein

Grundsätzlich lassen sich hier schon etwaige Einstellungen vornehmen. Festlegen lassen sich Einschränkungen grundsätzlich durch die Funktionen „`reject`“ und „`permit`“. Ein Eintrag wie gefolgt

```
smtpd_client_restrictions = reject quotesbymail.com,
                            permit fhdw.lan,
                            hash:/etc/postfix/access
```

würde die Domäne “`fhdw.lan`” als SMTP-Klient akzeptieren, die Endung „`quotesbymail.com`“ abweisen.

Dies lässt sich testen in dem man ein HELO/EHLO (Hello / Extended Hello) Kommando absetzt. Durch HELO/EHLO wird eine SMTP-Sitzung gestartet und der Klient am Server identifiziert. Dafür muss man in der Konsole folgenden Befehl laufen lassen:

```
telnet localhost 25
```

In Abb. 3 wird ein solches HELO „quotesbymail.com“ abgesetzt. Wie zu sehen ist, wird die Weiterleitung der E-Mail bereits durch die vorgenommenen Einstellungen verhindert:

Abb. 4 – Verweigerte Adressen

Weitere Einträge von Bedeutung in „/etc/postfix/main.cf“ sind:

- `smtpd_helo_restrictions =`
  verwaltet die Hosts, die ein HELO / EHLO Kommando senden können. Hier kann durch Einschränkungen SPAM-Software bereits geblockt werden. Standardmäßig wird von allen Hosts ein HELO akzeptiert.
- **`smtpd_helo_required`** `= yes`
  Legt fest, ob ein HELO / EHLO Kommando am Anfang einer SMTP Session benötigt wird. Standardmäßig ist „no“ eingestellt. Jedoch sollte der Wert auf „yes“ gesetzt werden, um SPAM-Software ohne HELO Kommando zu blocken.
- `transport_maps = hash:/etc/postfix/transport`
  verweist auf „/etc/postfix/transport“ zum Einrichten von Routen zu Mailrelays
- `message_size_limit = 2048000`
  setzt die maximale Größe einer E-Mail zu setzen (Standard: 1 MB), der vorgeschlagene Wert ist 2 MB, damit auch größere Dateien (z.B. Word, Bitmap,...) in der Schul-Infrastruktur versandt werden können.
- `mailbox_size_limit = 10240000`
  legt die maximale Größe der Postfächer fest (Standardwert `0` = unbegrenzt). Es empfiehlt sich, diese Größe zu limitieren. Vorschlag: 10 MB für jeden Account, um einer Maximallast von über 1000 Schülern standhalten zu können.

3.3 Filtern von E-Mails

Inhalte und Header von E-Mails lassen sich bereits durch Postfix mit „Header checks“ bzw. „Body checks“ filtern. Dazu werden zunächst zwei Dateien

- „/etc/postfix/header_checks“
- „/etc/postfix/body_checks“

erstellt, wenn sie nicht bereits vorhanden sind. In „/etc/postfix/header_checks“ lassen sich explizit Betreff- und Absenderangaben ausschließen. In „/etc/postfix/body_checks“ kann man festlegen, dass die Inhalte einer Mail auf bestimmte Worte oder ganze Sätze durchsucht werden. Bei Vorhandensein wird die E-Mail dann komplett geblockt oder nur mit einer Warnung versehen weitergeleitet wird.

Begriffe in „body_checks“ werden folgendermaßen ausgedrückt:

```
/^Subject: { Begriff }/
```

Dabei können auch Wildcards mit „.*“ eingesetzt werden. So sperrt man sämtliche Mails, die das Wort zwischen den Wildcards enthalten. Klein- und Großschreibung spielen dabei keine Rolle. Hinter den regulären Ausdrücken steht die entsprechende Aktion:

- WARN – nimmt die Mail an, gibt sie aber mit einer Warnung versehen weiter
- REJECT – verweigert die Annahme komplett und kann mit angehängtem Text an den vermeintlichen Spammer zurückgegeben werden. Hinter dem REJECT steht dann die auszugegebende Textzeile.

Beispiele für eine body_check – Dateien, die eine E-Mail untersucht, sind z.B.:

```
/ Zum downloaden hier klicken / WARN
/ Virus-Warnung / REJECT
/ Sex / REJECT Nicht hier!
```

Analog zu den body_checks, funktioniert eine header_checks-Datei wie folgt:

```
/Begriff/
```

Damit kann der gesamte Kopf einer Mail (Adresse, Betreff, usw.) nach den eingetragenen Begriffen durchsucht und im Vorfeld abgewiesen werden. Somit kann z.B. nach bestimmten Schlagwörtern in einer Betreffzeile gesucht. Ein Beispiel für eine solche header_checks Datei sieht folgendermaßen aus:

```
/^Subject: .*Sex.*/ REJECT
/^Subject: .*Virus.* /WARN
```

Im Internet kann man durch die Schlagwortsuche nach header_checks und body_checks schon bestimmte standardmäßige „Blacklists“ finden. Diese enthalten Einträge, um Formulierungen herauszufiltern, wie sie beispielsweise häufig in unerwünschten Nachrichten verwendet werden.

Damit Postfix die `header_checks` bzw. `body_checks` auch berücksichtigt, muss auf die Dateien in der „`/etc/postfix/main.cf`“ Konfigurationsdatei verwiesen werden:

```
header_checks = regexp:/etc/postfix/header_checks
body_checks = regexp:/etc/postfix/body_checks
```

Sollen IP-Adressen gesperrt werden, empfiehlt es sich hingegen, den folgenden Eintrag in der Hauptkonfigurationsdatei vorzunehmen:

```
smtpd_recipient_restrictions = check_client_access
hash:/etc/postfix/ip-block
```

Die Datei „**`/etc/postfix/ip-block`**“ kann dann folgende Einträge beinhalten:

```
10.0.8.15   REJECT
10.1.2.3    REJECT
```

Analog funktioniert das auch mit Sperren von E-Mail-Adressen. Trägt man in der Konfigurationsdatei von Postfix den folgenden Eintrag ein:

```
check_client_access = hash:/etc/postfix/client_restrictions
```

Lassen sich in „`/etc/postfix/client_restrictions`“ einzelne Adressen oder sogar ganze Länderkennungen blocken. Abb. 4 stellt eine mögliche „`/etc/postfix/client_restrictions`“-Datei dar:

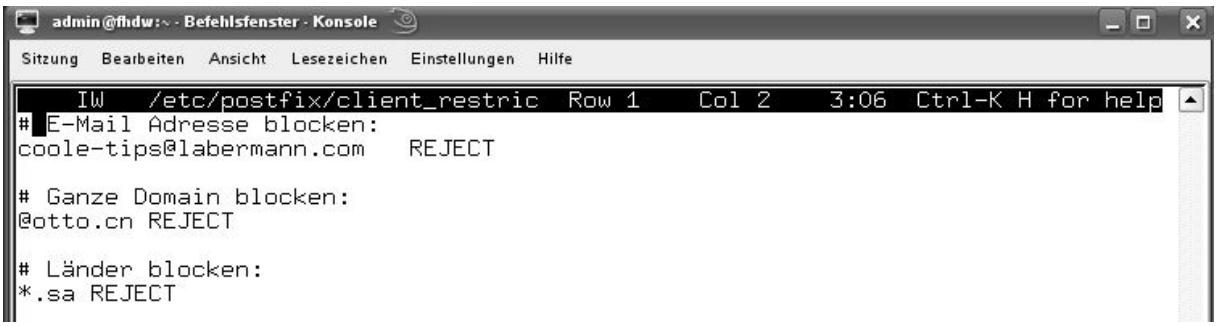

Abb. 5 – Adressen und Adressräume sperren

Somit lassen sich schon ohne Spam-Filter und Virenscanner viele Einschränkungen vornehmen.

3.4 Mail Retrieval Agent (Cyrus-SASL)

Zusätzlich zum MTA wird ein MRA benötigt. Die Wahl fiel auf cyrus, ein Open-Source-E-Mail-Server, der IMAP und POP3, sowie das Simple Authentication and Security Layer (SASL) Framework zur Authentifizierung im Internet unterstützt. Um Cyrus-SASL, sowie die bekanntesten SASL-Mechanismen zu installieren, werden folgende Befehle unter Administratorprivilegien im Terminal eingegeben:

```
yast2 -i cyrus-sasl cyrus-sasl-crammd5 cyrus-sasl-digestmd5 cyrus-
sasl-gssapi cyrus-sasl-otp cyrus-sasl-plain cyrus-sasl-saslauthd
```

Für Cyrus-SASL wird ein System Startup Link eingetragen:

```
chkconfig -add saslauthd
/etc/init.d/saslauthd start
```

Bevor die SASL Authentifizierungsmechanismen genutzt werden kännen, müssen nun noch Transport Layer Security (TLS)-Zertifikate erstellt und Postfix für SMTP-AUTH und TLS konfiguriert werden. Dazu dienen folgende Befehle:

```
mkdir /etc/postfix/ssl
cd /etc/postfix/ssl
openssl genrsa -des3 -rand /etc/hosts -out smtpd.key 1024
chmod 600 smtpd.key
openssl req -new -key smtpd.key -out smtpd.csr
```

Für das Passwort des SMTPD Schlüssel wurde „aaaaa" gewählt. Angaben für das TLS-Zertifikat wurden getroffen, wie in Abb. 5 dargestellt.

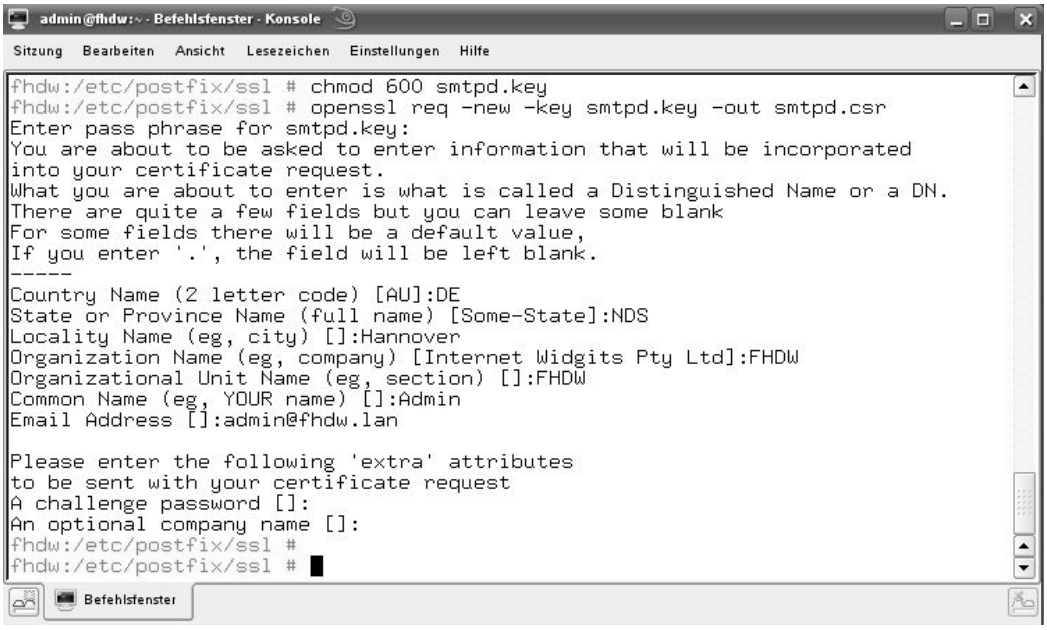

Abb. 6 – Getroffene Angaben im TLS-Zertifikat

Mit den folgenden Befehlen wird die Erstellung der Zertifikate schließlich abgeschlossen:

```
openssl x509 -req -days 3650 -in smtpd.csr -signkey smtpd.key -out
smtpd.crt
openssl rsa -in smtpd.key -out smtpd.key.unencrypted
mv -f smtpd.key.unencrypted smtpd.key
openssl x509 -req -days 3650 -in smtpd.csr -signkey smtpd.key -out
smtpd.crt
```

Um TLS Verbindungen in Postfix zu aktivieren, muss die Datei `/etc/postfix/master.cf` so bearbeitet werden, dass die Zeile `tlsmgr` nicht mehr auskommentiert ist:

```
[...]
tlsmgr      unix -       -    n        1000?   1      tlsmgr
[...]
```

TLS Einstellungen. Damit Postfix mit TLS-Unterstützung betrieben werden kann, müssen in der „`/etc/postfix/main.cf`“ Postfix Konfigurationsdatei entsprechenden Einträge vorgenommen werden.

```
smtpd_use_tls = yes
smtpd_tls_auth_only = yes
smtpd_tls_key_file = /etc/certs/key.pem
smtpd_tls_cert_file = /etc/certs/cert.pem
smtpd_tls_CAfile = /etc/certs/cert.pem
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
tls_random_source = dev:/dev/urandom
smtp_use_tls = yes
smtp_tls_key_file = /etc/certs/key.pem
smtp_tls_cert_file = /etc/certs/cert.pem
smtp_tls_CAfile = /etc/certs/cert.pem
```

Postfix ist dann TLS-fähig. Um die Konfiguration zu überprüfen, lässt man die in Abb. 6 dargestellten Befehle laufen. Wie in dieser Abbildung zu erkennen, ist TLS aktiv.

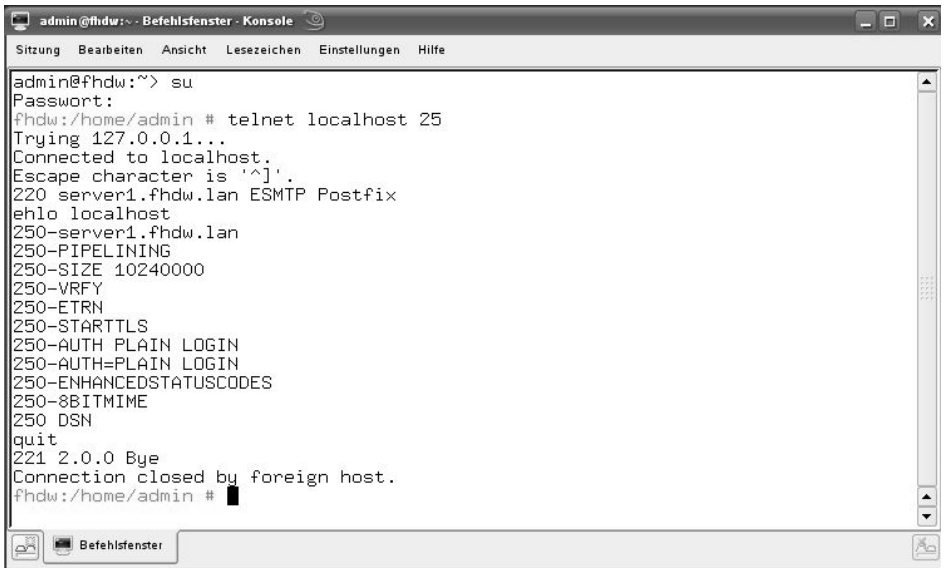

Abb. 7 – Überprüfung der Mailserver Dienste

3.5 Courier-IMAP/POP3

Zur Unterstützung von IMAP bzw. POP3 wurde der Courier-IMAP bzw. Courier-POP3 Daemon installiert, der „Maildir“-Unterstützung hat. Courier kann E-Mails in die Postfächer von lokalen Benutzerkonten legen. Zur Installation lässt man im Terminal als Superuser folgenden Befehl laufen:

```
yast2 –i courier-imap fam-server courier-authlib expect tcl
```

Analog zu Postfix und cyrus wurden dann für den IMAP bzw. POP3 Daemon System Startup Links hinzugefügt:

```
chkconfig –add fam
chkconfig –add courier-authdaemon
chkconfig –add courier-pop
chkconfig –add courier-imap
/etc/init.d/courier-pop start
/etc/init.d/courier-imap start
chkconfig –add courier-pop-ssl
chkconfig –add courier-imap-ssl
/etc/init.d/courier-pop-ssl start
/etc/init.d/courier-imap-ssl start
```

Damit Postfix so konfiguriert ist, dass Nachrichten in nutzerspezifische Verzeichnisse abgelegt werden, wie sie Courier-IMAP benötigt, ist folgender Verweis in der „`/etc/postfix/main.cf`“ – Datei nötig:

- `home_mailbox = Maildir/`

In den home Verzeichnissen der lokalen Benutzer wurde dann mittels des Befehles

```
Maildirmake Maildir
```

Ein Maildir-Ordner erstellt. Wird nun eine E-Mail an z.B. „`ert@fhdw.lan`“ versandt, landet sie in dem Verzeichnis des lokalen Benutzers `ert`.

3.6 Mail Delivery Agent (Procmail)

Bevor der Spamfilter „SpamAssassin“ eingerichtet wird, werden die Dienste eines MDA benötigt. Dafür eignet sich Procmail, das durch Postfix aufgerufen werden kann. Durch diesen MDA lassen sich E-Mails filtern und der Spamfilter „SpamAssassin“ aktivieren. Für die Installation von Procmail werden folgende Befehle im Terminal eingegeben:

```
yast2 -i procmail
```

3.7 SpamAssassin

Um „SpamAssassin“ zu installieren wird im Terminal unter Administratorprivilegien folgender Befehl eingegeben:

```
yast2 –i perl-HTML-Parser perl-Net-DNS-perl-Digest-SHA1
```

Der Packetmanager installiert dann die PERL-Module, die von „SpamAssassin“ benötigt werden. Seine Konfigurationsdateien sucht SpamAssassin in `/etc/mail/spamassassin`. Die wichtigste Datei ist `/etc/mail/spamassassin/local.cf`, in der alle nötigen Einstellungen vorgenommen werden können:

- `required_hits = 5.0`
  legt fest, wieviele Hits benötigt werden, um eine Nachricht als Spam auszulegen
- `rewrite_header Subject [*****SPAM******]`
  beschreibt den Header einer E-Mail neu, wenn es sich dabei um Spam handelt
- `report_safe 1`
  erkennt Spam in Dateianhängen
- `use_bayes 1`
  aktiviert den Bayesschen Filter
- `bayes_auto learn 1`
  aktiviert das automatische Lernen des Bayesschen Filters
- `bayes_path /home/spamd/`
  legt den Pfad des Bayesschen Filters fest
- `bayes_file_mode 066`
- `skip_rbl_checks 0`
  prüft Realtime Blackhole Lists
- `use_razor2 1, use_dcc 1, use_pyzor 1`
  Aktiviert Razor, DCC bzw. Pyzor
- `whitelist_from @drhellberg.de`
  Legt Whitelist für E-Mail-Domänen fest.

Danach wird der Filter für Postfix integriert. Dafür ändert man /etc/postfix/master cf wie folgt:

```
#smtp      inet  n       -       n       -       -       smtpd
smtp      inet  n    -   n  -   -  smtpd -o content_filter=filter:
[...]
#SPAMASSASSIN
Filter unix – n n - - pipe user=filter
argv=/home/filter/sc/filter.sh –f $ {sender} -- $ { recipient }
```

Nun wurde SpamAssassin in Postfix integriert.

3.8 ClamAV

Für die Installation von „ClamAV“ wird zunächst die Datei „`clamav-0.88.tar.gz`“ herunterladen. Dazu wird im Terminal der folgenden Befehl angeben:

```
wget http://prdownloads.sourceforge.net/clamav/clamav-0.88.tar.gz
tar xfz clamav-0.88.tar.gz
```

Im Anschluss darauf wird in das Verzeichnis gewechselt und „ClamAV“ kompiliert:

```
cd  clamav-0.88
./configure
make
make install
```

Hier gilt es zu beachten, dass es ein Benutzer „clamAv“ angelegt sein muss, unter dessen Account Konfigurationen jederzeit vorgenommen werden können. Vor der ersten Inbetriebnahme sollte ClamAV auf die neueste Version gepatcht und aktuelle Virensignaturen heruntergeladen werden. Ein automatisches Update der Software bzw. von Virensignaturen kann ebenso aktiviert werden und ist dringend zu empfehlen. kann. Beide Auswahlmöglichkeiten sind nach der Grundinstallation standardmäßig deaktiviert. Aktivieren lässt sich ein automatisches Viren-Update mittels des Befehles:

```
/usr/bin/freshclam --quiet
```

Schließlich sollte geprüft werden, ob clamAV läuft. Diese Prüfung ist in Abb. 8 dargestellt. ClamAV ist nun konfiguriert, um Virenmails zu filtern, vgl. [LAI09].

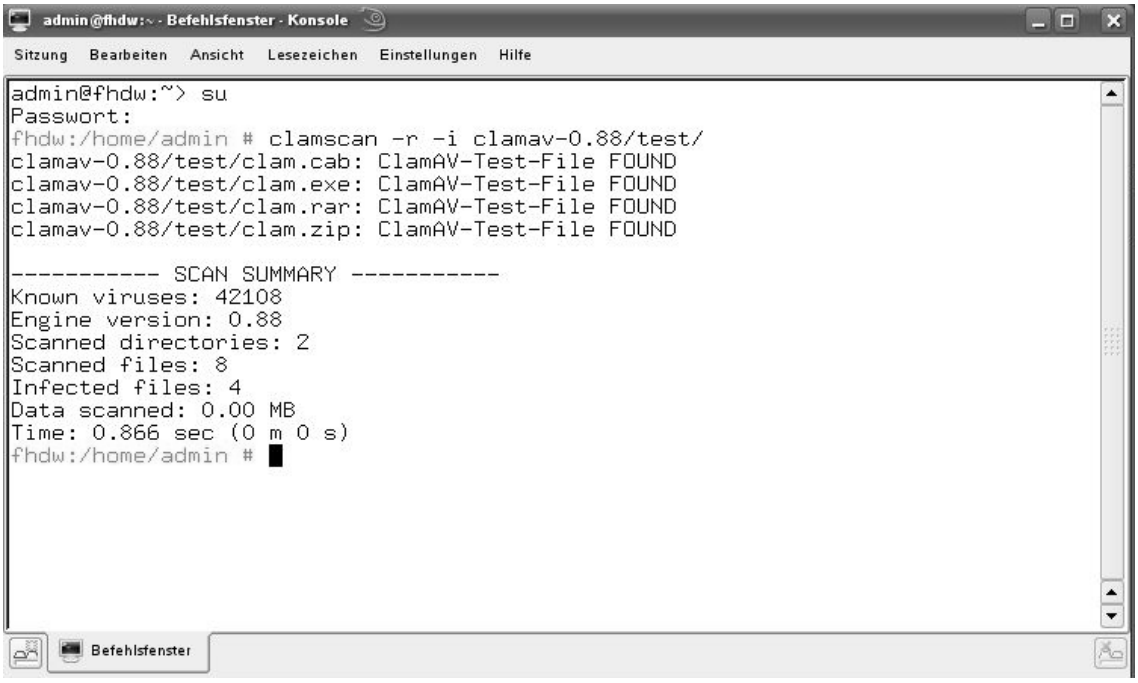

Abb. 8 – Funktionsüberprüfung von clamAV

3.9 Mail User Agents

Um den Mailverkehr zu testen, werden schließlich auch MUAs benötigt. Die Standardinstallation von openSUSE Linux bietet hierfür die Dienste des Anwendungsprogrammes KMail an. In KDE 3.5 ist es aufrufbar unter „`Start >> Anwendungen >> Internet >> Kmail`“. Sobald KMail zum ersten Mal aufgerufen wird, lassen sich Konfigurationen zum Senden und zum Versand von Nachrichten vornehmen. Dies lässt sich in der Anwendung unter „`Einstellungen >> Kmail einrichten...`“ zu einem späteren Zeitpunkt korrigieren.

Zum Empfang von Nachrichten wird POP3, für den Versand von Nachrichten SMTP ausgewählt und die statische IP-Adresse des Servers angegeben. Wird ein DNS-Server betrieben, lässt sich für diese Adressierung auch eine Namensauflösung definieren.

Um die Funktionalität des virtuellen Mailservers auch auf anderen Betriebssystemen zu testen, wurde schließlich eine zweite VM, ein „Windows XP Client“ eingerichtet. Auf diesem Betriebssystem wurden die MUAs Mozilla Thunderbird version 2.0.0.21 und Microsoft Outlook 2007 zu Testzwecken installiert. Die Einstellungen zum Empfang und Versenden von Nachrichten waren analog zu denen in KMail.

Vorausgesetzt die lokalen Benutzerkonten sind auf dem Mail-Server gespeichert, kann nun der E-Mail-Verkehr getestet werden.

Die Funktionalität für den externen Mail-Verkehr kann nur bedingt getestet werden, da der eingerichtete Mail-Server von Anbietern wie z.B. GMX, Web oder Freemail beim Versenden als Spam-Server deklariert wird.

4. Anwendung

Dieses Kapitel befasst sich mit den Szenarien, um die gewählte Anti-Viren- bzw. Anti-Spam-Lösung mit bewährten Techniken zu testen. Des weiteren wird auf die Perspektiven der Softwarelösungen eingegangen und die Schnittstellen dieses Projektes in Bezug auf die anderen Teilprojekte untersucht.

4.1 Testen der Blacklists

Zur Prüfung der `header_checks` und `body_checks` (vgl. 3.3), wurden E-Mails an die lokale E-Mail-Adressen versendet. Enthielt eine versandte E-Mail eines der vordefinierten Schlagwörter, kam sie entweder zurück (REJECT) oder wurde mit einem Warnbetreff versendet (WARN).

Abb. 9 stellt eine E-Mail dar, die in Folge der header_checks bzw. `body_checks` nicht versendet werden konnte.

Abb. 9 – Abgelehnte Nachrichten des Mailservers

Analog können auch ganze Adressräume und IP-Adressen gesperrt werden.

4.2 Testen von SpamAssassin

Um die Installation von SpamAssassin auf seine Funktionsweise zu überprüfen, bietet sich der Testtext Generic Test for Unsolicited Bulk Email (GTUBE) an. GTUBE ist ein 68 Zeichen langer Test-String, der eine gängige Methode bietet, um Anti-Spam-Lösungen zu überprüfen. Der Inhalt des Test-Strings ist in Abb. 10 angegeben. Er sollte als Text in eine Test-E-Mail eingefügt werden, ohne Leerzeichen oder Zeilenumbrüche.

Abb. 10 – Testen von Spam mit dem GTUBE-Test-String

Eine E-Mail im Request For Commons (RFC)-822-Format mit diesem Inhalt wurde an ein lokales Konto versandt. SpamAssassin erkannte und kennzeichnete die E-Mail als Spam . Bei dem Empfänger der Nachricht kam die E-Mail dann an, siehe Abb. 11.

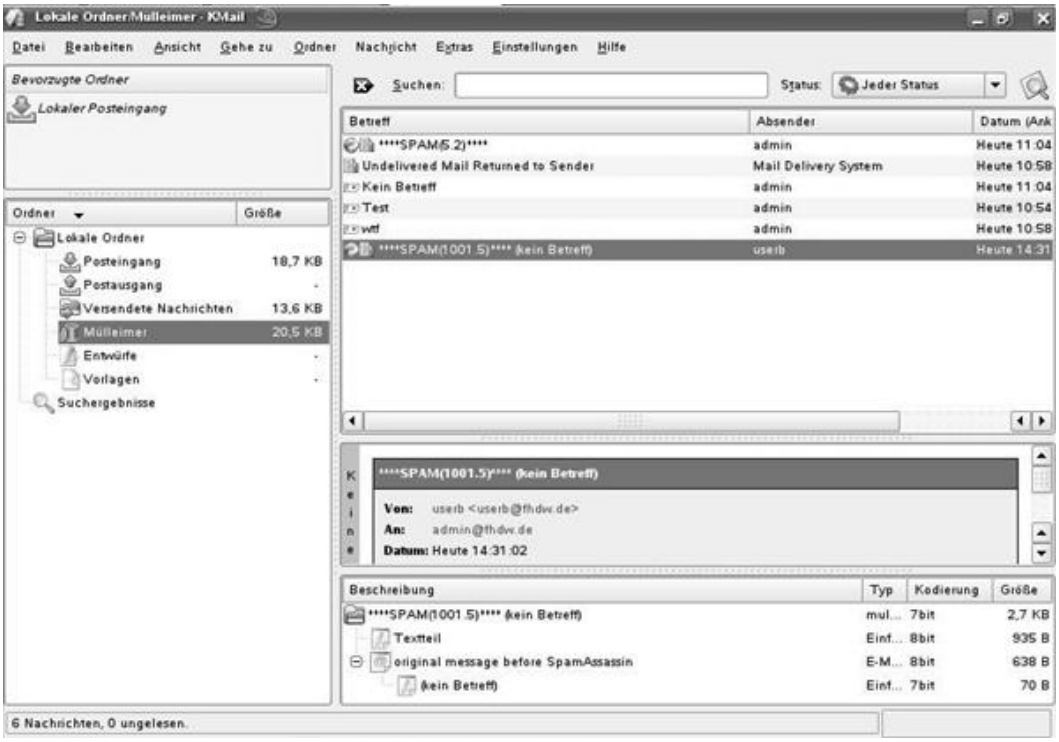

Abb. 11 – **Behandlung des GTUBE-Test-Strings mit SpamAssassin**

Der Empfänger der Nachricht kann dann selbst entscheiden, ob er die Spam-Mail den Sicherheitsvorkehrungen zum Trotz liest oder sie vollständig entfernt.

4.3 Testen von ClamAV

Zum Testen von Anti-Viren-Software wird allgemein die European Institute for Computer Antivirus Research (EICAR)-Testdatei verwendet. Mit diesem Testmuster können Funktionen von Anti-Viren-Programmen getestet werden [WIK09a]. Zu Testzwecken wurde die folgende Zeichenkette an eine lokale E-Mail-Adresse versendet:

```
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-
FILE!$H+H*
```

„ClamAV“ erkannte diese Signatur unter dem Namen „Eicar-Test-Signature“ und legte die offensichtlich infizierte E-Mail automatisch im Ordner „Spamverdacht“ ab, siehe Abb. 12.

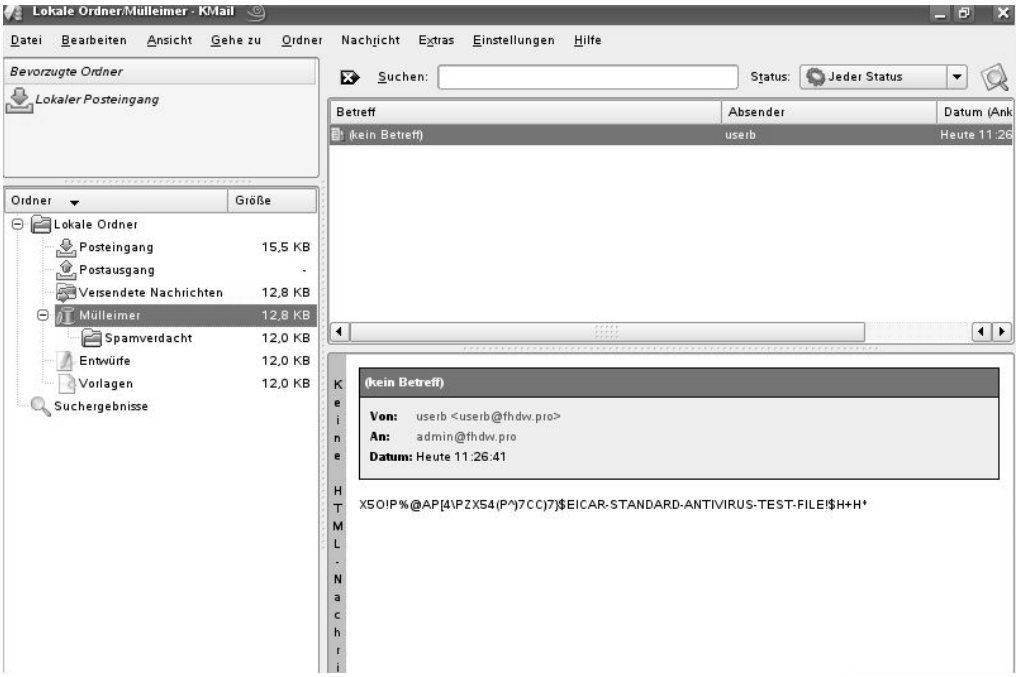

Abb. 12 – Behandeln des EICAR-Testdatei

Um auch Dateianhänge der Virusprüfung zu überziehen, wurden schließlich EICAR-Testdateien als Dateianhang versandt. Diese enthielten die Testsignatur z.B. als einfach oder mehrfach gezippte Datei. Auch diese Dateianhänge wurden erkannt und führten zu dem erwünschten Ergebnis.

5. Schnittstellen

Die Schnittstellen dieses Projektes beziehen sich auf die anderen Teilprojekte:

- Patchverteilung / Softwaremanagement
- Proxy
- Firewall

Mit dem Projekt „Softwaremanagement“ hat dieses Projekt nur geringfügig zu tun, denn die Updates von Anti-Viren-Software bzw. Virensignaturen sollen den Klienten überlassen werden. Aus dem Grund, dass die Klienten privaten Zugriff auf die Maschinen haben, muss Anti-Viren-Software ohnehin für den Privatgebrauch zugänglich sein. Ein Grund, der außerdem gegen eine automatische Patchverteilung für Anti-Viren-Software oder Virensignaturen spricht, ist, dass der Patch-Server zu Stoßzeiten überladen werden kann.

Der Zugriff auf den Mailserver kann über den „Proxy“-Server laufen. Doch das ist unnötig kompliziert, denn das Teilprojekt „Proxy“ müsste dann eine Schnittstelle für externe Zugriffe auf den Mailserver anbieten. Daher ist es angedacht, den Mailserver 'vor' dem Proxy laufen zu lassen.

Im Gegensatz zu diesen beiden Projekten, spielt die 'Firewall' eine entscheidende Rolle, denn sie sollte, den Mailverkehr nach innen und außen nicht unterbinden, i.e. die für den Mailverkehr gängigen Ports nicht sperren:

- Port 25 – SMTP bzw. 465 für SSL verschlüsseltes SMTP (SMTPS)
- Port 110 – POP3 bzw. 995 für SSL verschlüsseltes POP (POP3SSPOP)
- Port 143 – IMAP bzw. 993 für SSL verschlüsseltes IMAP (IMAPS)

6. Zusammenfassung

Im Laufe der letzten Jahre haben die Angriffe durch Viren und die Ausbremsung von Systemleistung durch Spam-Mails, dramatisch zugenommen, so dass allein der Anteil an Spam auf schätzungsweise über 90 - 95% des gesamten Mailverkehrs zugenommen hat.

Das Ziel dieses Projektes war es daher, eine Anti-Viren bzw. Anti-Spam-Lösung für das Projekt „Schul-IT“ zu installieren, zu konfigurieren und zu testen. Dazu wurde ein Mailserver in einer virtuellen Umgebung eingerichtet und mit dem Spam-Filter „Spam-Assassin“ ausgestattet.

Durch die Integration von SpamAssassin wurde es möglich, einen effektiven Filtermechanismus einzubauen. Testszenarien, mit z.B. dem GTUBE-String, bestand der Spam- Filter problemlos.

Als weiteres Ziel war es angedacht, eine effektive Anti-Viren-Lösung zu integrieren. Es wurde die Software „ClamAV“ gewählt, die problemlos in den Mailserver integriert. Die Testergebnisse liefen zufriedenstellend, so konnte eine zeit geplante Überprüfung, automatische Virenupdates und ein automatischer Virenscan mit relativ leichten Mitteln eingerichtet werden.

Die beiden gewählten Softwarelösungen eignen sich daher für das Projekt „Schul-IT“. In Bezug auf die geplanten Funktionserneuerungen für ClamAV und die allgemeine Anwendbarkeit des Spam-Filter SpamAssassin, sind beiden Anwendungsprogramme auch im Hinblick auf ein weiteres Vorgehen durchaus zu empfehlen.

Jedoch bedeuten die in dieser Dokumentation beschriebenen Vorgehensweisen und Anwendungen nur einen ersten Schritt in Richtung eines sicheren Schul-Server-Systems. Das komplexe Thema Virenschutz und Anti-Spam in diesem Projekt, aufgrund des stark begrenzten Zeitraums für die Erstellung und Konfiguration des Mailservers und der Schutzmechanismen, noch bei weitem nicht ausreichend erfasst und abgedeckt wird.

Auch muss z.B. bei den Mechanismus des `header-` und `body_checks` noch genau abgeklärt werden, wie jeweils die rechtlichen Grundlagen sind. Dabei ist zu klären, ob es zulässig ist, identifizierte Emails mit bestimmten Text oder bestimmter Herkunft gleich zu blocken oder gar zu löschen oder inwieweit die freie Selbstbestimmung der Person dadurch beeinträchtigt werden könnte.

Quellen

[ATI09] www.ATIS.uka.de, Abteilung Technische Infrastruktur – Universität Karlsruhe, offizielle Homepage, aufgerufen am 02.07.09

[CLA09] ClamAV – offizielle Homepage, http://www.ClamAV.net/, Homepage, aufgerufen am: 19.05.09

[HEL09a] „Kurzbeschreibung: Projekt-Schul-IT“, Prof. Dr. Günter Hellberg, Stand: April 2009, URL: http://www.drhellberg.de/FHDW/Betriebssysteme/2Quartal2009/Kurzbeschreibung- Projekt-Schul-IT-04-2009-2.pdf, PDF-Dokument, aufgerufen am: 08.05.09

[HEL09b] „Screenshots für die komplette Grundinstallation OpenSuse 11.0 32 Bit mit allen Updates (7z-File)“ URL: http://www.drhellberg.de/FHDW/Betriebssysteme/2Quartal2009/OS11.0.7z, Screenshots archiviert als 7z-Datei, aufgerufen am : 19.05.09

[HEL09c] "Aufbau Netzwerk (Grenznetz) und IP-Adressvergabe", http://www.drhellberg.de/FHDW/Betriebssysteme/2Quartal2009/Visio-IP-Adressierung_FW_Projekt.pdf, aufgerufen am 30.06.09

[LAI09] „Postfix-Cyrus-Procmail-SpamAssassin HowTo“, David Lais, 2009, URL: http://www.postfix-howto.de/installation/clamav.htm, aufgerufen am 05.07.09

[OSU09] ISO-Datei, „openSUSE 11.0“, last modified: 10-Jun-2008 17:42, URL: http://download.opensuse.org/distribution/11.0/iso/dvd/openSUSE-11.0-DVD-i386.iso aufgerufen am 15.04.09

[POS09] Postfix- offizielle Homepage, http://www.postfix.org/, aufgerufen am 30.06.09

[SPA09] SpamAssassin - offizielle Homepage, „The Apache SpamAssassin Project“, URL: http://spamassassin.apache.org/, aufgerufen am: 19.05.09

[VMW09] VMWare Server, „Download Vmware Server (for Windows and Linux Systems) VMWare-server-installer-1.0.8-126538, Version: 1.0.8 | 2008/11/06 | Build: 126538, URL: http://register.vmware.com/content/eula-108.html, aufgerufen am: 19.05.09

[WIK09a] Wikipedia – Freie Enzyklopädie, Artikel: „EICAR-Testdatei“, URL: http://de.wikipedia.org/wiki/EICAR-Testdatei, aufgerufen am: 19.05.09

[WIK09b] Wikipedia – Freie Enzyklopädie, Artikel: „GTUBE“, URL: http://de.wikipedia.org/wiki/GTUBE, aufgerufen am: 19.05.09