Clam AntiVirus 0.71
Benutzerhandbuch

von Tomasz Kojm
Deutsche Übersetzung: uplink coherent solutions™
`http://www.uplink.at`

Contents

1 Einführung

Clam AntiVirs ist ein Antivirus Werkzeug für UNIX Systeme. Haupteinsatzgebiet ist die Integration mit Mailservern (Filtern von Mail-Anhängen). Dieses Paket bietet einen flexiblen und skalierbaren Server, einen Kommandozeilen- Scanner und ein Programm zum automatischen Aktualisieren via Internet. Die Programme sind mittels der shared library, die mit Clam AntiVirus ausgeliefert wird, realisiert. Diese shared library können Sie in Ihrer eigenen Software verwenden.

1.1 Funktionen

- Lizenz: GNU General Public License, Version 2
- POSIX konform, portabel
- Extrem schnelles Scannen
- Scannen beim Datei-Zugriff[1] (nur bei Linux und FreeBSD)
- Erkennt derzeit über 20.000 Viren, Würmer und Trojaner
- Unterstützung für Archive und komprimierte Dateien
- Kann Microsoft Office (OLE2) Dateien entpacken
- Eingebaute Unterstützung für RAR (2.0), Zip, Gzip, Bzip2
- Eingebauter Schutz vor sog. Archiv-Bomben[2]
- Eingebaute Unterstützung für Mbox, Maildir und Mails im Rohformat
- Ein Programm zum Aktualisiern der Signatur-Datenbank mit Unterstützung für digitale Signaturen wird mitgeliefert

1.2 Mailinglisten

Diese vier Mailinglisten sind verfügbar:

- **clamav-announce*lists.sf.net** - Informartionen über neue Versionen (auch über Debian-Pakete), modieriert[3].

[1] On-access scanning
[2] z.B. ein winziges Archiv, das beim Entpacken auf die tausendfache Grösse anwächst
[3] D.h. Abonnenten können diese Liste nur lesen, aber keine Mails an die Liste schicken

- **clamav-users*lists.sf.net** - Fragen der Clam AntiVirus Nutzer
- **clamav-devel*lists.sf.net** - Entwicklung
- **clamav-virusdb*lists.sf.net** - Bekanntmachung von Datenbank-Aktualisierungen

Unter der folgenden Adresse können Sie diese Mailinglisten abonnieren und die Archive der Listen durchstöbern: `http://www.clamav.net/ml.html`

1.3 Virus übermitteln

Sie haben einen Virus entdeckt, der von ClamAV mit der aktuellsten Datenbank nicht erkannt wurde? Wir brauchen jetzt Ihre Hilfe: bitte verwenden Sie unseren *ClamAV Online Specimen Scanner*:

`http://www.gietl.com/test-clamav`

und senden den neuen Virus dann über unsere Webseite ein:

`http://www.clamav.net/cgi-bin/sendvirus.cgi`

Danke!

2 Installation

2.1 Unterstützte Plattformen

Clam AntiVirus ist zur Installation auf den folgenden Betriebssystemen/ Plattformen gedacht (getestete Plattformen in Klammern):

- GNU/Linux - alle Versionen und Plattformen
- Solaris - alle Versionen und Plattformen
- FreeBSD - alle Versionen und Plattformen
- OpenBSD 3.0/1/2 (Intel/SPARC)
- AIX 4.1/4.2/4.3/5.1 (RISC 6000)
- HPUX 11.0
- SCO UNIX

- IRIX 6.5.20f
- Mac OS X
- BeOS
- Cobalt MIPS boxes (RAQ1, RAQ2, QUBE2)
- Windows/Cygwin

Einige Funktionen stehen auf Ihrem Betriebssystem unter Umständen nicht zur Verfügung. Wenn Sie ClamAV auf einem System einsetzen, das hier noch nicht genannt ist, lassen Sie es uns wissen!

2.2 Einsatzfertige (binäre, übersetzte) Pakete - stabile Versionen

- **Debian**
  Das Paket wird von Stephen Gran und Thomas Lamy verwaltet. ClamAV ist seit dem Sarge Release offiziell in der Debian Distribution enthalten. Mit dem Befehl

  ```
  apt-cache search clamav
  ```

  können Sie die verfügbaren Paketnamen und Versionen anzeigen lassen. Inoffizielle Pakete für Woody und Sarge sind erhältlich und meist neuer als die offiziellen. Fügen Sie Ihrer `/etc/apt/sources.list` einfach die folgenden Zeilen hinzu:

  ```
  stable/woody (i386):
  deb http://people.debian.org/~sgran/debian woody main
  deb-src http://people.debian.org/~sgran/debian woody main
  testing/sarge (i386):
  deb http://people.debian.org/~sgran/debian sarge main
  deb-src http://people.debian.org/~sgran/debian sarge main
  ```

  Alternativ können Sie auch unter `http://www.apt-get.org` nach ClamAV Paketen suchen.

- **RedHat - Fedora**
  Die Pakete werden von Petr Kristof verwaltet und sind unter `http://crash.fce.vutbr.cz/crash-hat/1/clamav/` verfügbar. Bitte folgen Sie den Instruktionen unter `http://crash.fce.vutbr.cz/yum-repository.html` und führen dann Folgendes aus:

```
yum update clamav
```

oder

```
up2date -u clamav
```

- **PLD Linux Distribution**
  Die RPM Pakete für die polnische Linux Distribution werden von Arkadiusz Miskiewicz unter der Adresse `http://www.pld-linux.org` verwaltet.

- **Mandrake**
  Die RPM Pakete für Mandrake sind direkt über die Mandrake Mirror-Server zu haben (siehe `http://www.rpmfind.net/linux/RPM/cooker/contrib/i586/CByName.html`). Das aktuellste Mandrake RPM Paket finden Sie unter `ftp://ftp.neocat.org/pub/`. Es wird von Bill Randle verwaltet.

- **FreeBSD**
  Die offizielle FreeBSD Portierung wird von Masahiro Teramoto verwaltet. Es sind 2 Versionen erhältlich: clamav und clamav-devel. Beide ports finden Sie im Port-System under FreeBSD im Verzeichnis `/usr/ports/security/`

- **OpenBSD**
  Eine inoffizielle OpenBSD Portierung finden Sie unter `http://www.fatbsd.com/openbsd/`

- **NetBSD**
  Die offizielle Portierung vom ClamAV ist verfügbar.

- **AIX**
  Fertige Pakete für AIX finden Sie in der AIX PDSLIB, UCLA unter `http://aixpdslib.seas.ucla.edu/packages/clamav.html`

- **MS Windows**
  Alle Hauptfunktionen von ClamAV sind unter Win32 bereits unter Verwendung des Cygwin Kompatibilitätslayers implementiert. Sie können ein selbstinstallierendes Paket unter dieser Adresse beziehen: `http://www.sosdg.org/clamav-win32/index.php`

- **MS Windows - Version mit grafischer Benutzeroberfläche**
  Eine eigenständige GUI Version ist ebenfalls verfügbar. Details finden Sie unter *ClamWin* im Abschnitt *Software von Drittanbietern* (5.39).

2.3 Fertige Pakete - Momentaufnahmen (snapshots)

Dank Fajar A. Nugraha können Sie täglich aktualisierte Programmversionen (daily builds from daily snapshots) für die folgenden Betriebssysteme laden:

- SPARC Solaris 8/9
- DEC OSF (kompiliert unter Tru64 UNIX V5.0A)
- AIX (kompiliert unter AIX Version 5.1)
- Linux i386 mit glibc 2.3 (kompiliert unter Fedora Core 1, funktioniert auch unter RH $\geq$ 8)
- Win32/Cygwin (kompiliert unter XP)

Besuchen Sie dazu `http://clamav.or.id`

2.4 Systemanforderungen

Sie benötigen zumindest die folgende Software, um ClamAV zu kompilieren:

- zlib und zlib-devel Pakete
- gcc compiler Suite (sowohl 2.9x, als auch 3.x werden unterstützt)

Des weiteren sind die folgenden Pakete optional, aber **sehr zu empfehlen**:

- bzip2 und bzip2-devel Bibliotheken
- GNU MP 3
  Die Installation des GMP Pakets ist sehr wichtig! Es ermöglicht der Komponente freshclam, digitale Signaturen der Virendatenbank zu prüfen.
  Wenn freshclam ohne GMP Unterstützung kompiliert wird, erhalten Sie bei jedem Update die Meldung

  ```
  SECURITY WARNING: NO SUPPORT FOR DIGITAL SIGNATURES
  ```

  Sie bekommen GNU MP unter der Adresse `http://www.swox.com/gmp`.
  Ein Hinweis für Nutzer von Solaris: Sie sollten die Systemvariable *ABI* auf 32 setzen (z.B. `setenv ABI 32`), bevor Sie `configure` für GMPstarten [4].

[4] Dank an Ed Phillips

2.5 Systembenutzer und -gruppe hinzufügen

Wenn Sie ClamAV das erste Mal auf Ihrem System installieren, müssen Sie zuerst einen dedizierten UNIX Benutzer und eine eigene Gruppe anlegen: [5]

```
# groupadd clamav
# useradd -g clamav -s /bin/false -c "Clam AntiVirus" clamav
```

Diese Methode funktioniert unter Linux und Solaris. Sollten Sie die Programme *groupadd, useradd* nicht in Ihrem System vorfinden, schlagen Sie bitte im Systemhandbuch nach. Wenn Sie ClamAV hingegen nur für einen einzelnen Benutzer installieren, können Sie diesen Schritt auslassen und stattdessen dem `configure` Script die Option `--disable-clamav` übergeben:

```
$ ./configure --disable-clamav
```

Damit verhindern Sie, dass ClamAV beim Starten nach dem *clamav* Benutzer und der Gruppe sucht. **clamscan benötigt aus Sicherheitsgründen immer einen unprivilegierten Benutzer, um im Root-Modus zu arbeiten!** Das Kennwort für diesen Benutzer sollte in der */etc/passwd* bzw. */etc/shadow* gesperrt sein.

2.6 Kompilieren

Wenn Sie die clamav Benutzer und Gruppe angelegt haben, entpacken Sie als nächsten Schritt das Archiv:

```
$ zcat clamav-x.yz.tar.gz | tar xvf -
$ cd clamav-x.yz
```

Wir gehen davon aus, dass Sie die Konfigurationsdateien nach `/etc` installieren möchten. Rufen Sie das Script `configure` daher folgendermassen auf:

```
$ ./configure --sysconfdir=/etc
```

Dezeit verwendet ClamAV *gcc* zum Kompilieren. Die Unterstützung von alternativen Compilern wird in naher Zukunft hinzugefügt werden.

[5] Hinweis für Cygwin Nutzer: wenn Sie keine /etc/passwd verwenden, können Sie diesen Schritt überspringen

```
$ make
$ su -c "make install"
```

Im letzten Schritt wird die Software standardmässig nach `/usr/local` kopiert und alle Konfigurationsdateien in das Verzeichnis `/etc`. **ACHTUNG: Aktivieren Sie NIEMALS das SUID oder SGID Bit für die Clam AntiVirus Programmdateien.**

2.7 Konfiguration

Wenn Sie den Serverprozess (daemon) nutzen möchten, müssen Sie das Programm zuerst konfigurieren. Mit den Standard-Einstellungen verweigert `clamd` die Zusammenarbeit:

```
$ clamd
ERROR: Please edit the example config file
       /etc/clamav.conf.
```

Aus der obigen Meldung ersehen Sie auch gleich den Speicherort der Konfigurationsdatei. Format und verfügbare Optionen sind zur Gänze in der *clamav.conf(5)* manpage beschrieben. Die Konfiguration von clamd ist sehr einfach, da die Konfigurationsdatei gut kommentiert ist. Bitte vergessen Sie nicht, die ”Example” Direktive in der `clamav.conf` zu löschen!
Eine weitere Funktion des clamd ist das Scannen von Dateien beim Zugriff[6] auf Basis des Dazuko Moduls, das via `http://dazuko.org` erhältlich ist. **Diese Software ist zum Betrieb von clamd nicht nötig, des weiteren sollte Dazuko (noch) nicht auf Produktionssystemen eingesetzt werden**. Ein spezieller Programmteil von clamd ist für die Kommunikation mit Dazuko verantwortlich. Diesen haben wir ”Clamuko” getauft (”Clamuko” deshalb, weil ”Dazuko” ein witziger Name ist. Wir wissen nicht, was ”Clamuko” eigentlich bedeutet...). Claumko wird unter Linux und FreeBSD unterstützt. Um dazuko zu kompilieren, führen Sie folgende Kommandos aus:

```
$ tar zxpvf dazuko-a.b.c.tar.gz
$ cd dazuko-a.b.c
$ make dazuko
```

oder

[6] on-access scanning

```
$ make dazuko-smp (for smp kernels)
$ su
# insmod dazuko.o
# cp dazuko.o /lib/modules/‘uname -r‘/misc
# depmod -a
```

Je nach der von Ihnen eingesetzten Linux Distribution müssen Sie einen eigenen Eintrag fuer dazuko in Ihrer `/etc/modules` machen oder einfach den Befehl

```
modprobe dazuko
```

einem der Start-Scripts hinzufügen, damit dazumo beim Systemstart geladen wird. Das Kompilieren unter FreeBSD ist sehr ähnlich; Sie müssen hier noch den Device-Eintrag `/dev/dazuko` erstellen:

```
$ cat /proc/devices | grep dazuko
254 dazuko
$ su -c "mknod -m 600 /dev/dazuko c 254 0"
```

Nun konfigurieren Sie Clamuko bitte in der *clamav.conf* und lesen im Kapitel 3.3 weiter.

2.8 Testen

Als ersten Test empfehlen wir, rekursiv das Quellverzeichnis durchsuchen zu lassen:

```
$ clamscan -r -l scan.txt clamav-x.yz
```

ClamAV sollte einige Test-Viren im Verzeichnis `clamav-x.yz/test` finden. Das Resultat des Scans wird mit dem obigen Befehl in die Datei `scan.txt` gespeichert[7]. Um clamd zu testen, verfahren Sie wie folgt: starten Sie den Server-Prozess clamd und führen dann *clamscan* aus (Alternativ können Sie auch direkt zum clamd Prozess via telnet verbinden und das Kommando `SCAN` eingeben):

```
$ clamdscan -l scan.txt clamav-x.yz
```

Der Inhalt der Logdatei `scan.txt` sollte ähnlich der von unserem Beispiel mit *clamscan* weiter oben sein.

[7] Mehr Information zu den ClamAV Optionen finden Sie hier: **man clamscan**

2.9 freshclam: Einrichten der automatischen Updates

freshclam ist ein Aktualisierungs-Programm fuer die Datenbank von Clam AntiVirus. Es kann auf zwei Arten eingesetzt werden:

- interaktiv - von der Kommandozeile, mit detaillierter Ausgabe
- daemon - eigenständig, im Hintergrund als Serverprozess

Wird *freshclam* vom Superuser gestartet (was Standard ist), verwirft es die (root-) Benutzer-Privilegien und schaltet auf den *clamav* Benutzer zurück. *freshclam* verwendet den `database.clamav.net` round-robin DNS, wodurch vollautomatisch ein Datenbank-Mirror (2.10) zum Update gewählt wird. Freshclam ist ein hochentwickeltes Werkzeug: es unterstützt Proxy Server (mit Authentifizierung), das Verifizieren digitaler Signaturen und geht mit Fehlern während dem Update tolerant um. **Ein schneller Test: starten Sie *freshclam* (als Superuser) ohne Parameter und prüfen Sie die Ausgabe.** Wenn *freshclam* meldet, dass alles in Ordnung ist, erstellen Sie ein passendes Logfile unter `/var/log` (Besitzer: *clamav* oder jener Benutzer, mit dessen Privilegien freshclam ausgeführt wird (`--user`):

```
# touch /var/log/clam-update.log
# chmod 600 /var/log/clam-update.log
# chown clamav /var/log/clam-update.log
```

Nun *sollten* Sie die Konfigurationsdatei editieren (üblichweise *freshclam.conf*) und die Direktive *UpdateLogFile* so einstellen, daš sie auf das gerade erstellte Logfile zeigt (was wir sehr anraten). Optional können Sie den Pfad zum Logfile auch manuell beim Prorgammstart mit `-l` übergeben. Abschliessend starten Sie den freshclam Serverprozess wie folgt:

```
# freshclam -d
```

Eine andere Methode ist, den *cron* Dienst zu verwenden. Dazu fügen Sie diese Zeile zur `crontab` des **root** oder **clamav** Benutzers hinzu:

```
N * * * * /usr/local/bin/freshclam --quiet
```

...um die Datenbank jede Stunde auf Aktualität zu prüfen. **N ist eine Zahl zwischen 1 und 59. Bitte verwenden Sie keine Vielfachen von 10, da diese Zeiten bereits von zu vielen Servern verwendet werden.** Proxy Server können Sie ausschliesslich direkt in der Konfigurationsdatei einstellen (das hat den Sinn, dass Sie ein Passwort Ihres Proxy-Servers mittels UNIX permissions entsprechend schützen können):

```
HTTPProxyServer meinproxyserver.org
HTTPProxyPort 1234
HTTPProxyUsername meinbenutzername
HTTPProxyPassword meinkennwort
```

2.10 Datenbank Mirrors

Freshclam lädt die jeweils aktuelle Datenbank von der Adresse `http://database.clamav.net`. Dank eines ”round-robin” Eintrags wird der Datenverkehr dabei gleichmässig zwischen allen Servern aufgeteilt:

Mirror	IP	Location	Administrator
clamav.man.olsztyn.pl	213.184.16.3	Olsztyn, Poland	Robert d'Aystetten <dart*man.olsztyn.pl>
avmirror1.prod.rxgsys.com	64.74.124.90	USA	Graham Wooden <graham*rxgsys.com>
avmirror2.prod.rxgsys.com	207.201.202.73	USA	Graham Wooden <graham*rxgsys.com>
clamav.e-admin.de	212.162.12.159	Dusseldorf, Germany	Andreas Gietl <a.gietl*e-admin.de>
clamav.essentkabel.com	195.85.130.84	Netherlands	Chris van Meerendonk <mirror*essentkabel.com>
clamav.inet6.fr	62.210.153.201 62.210.153.202	France	Lionel Bouton <clamavdb*inet6.fr>
clamav.netopia.pt	193.126.14.29	Portugal	Miguel Bettencourt Dias <mbd*netopia.pt>
clamav.sonic.net	209.204.175.217	USA	Kelsey Cummings <kgc*sonic.net>
clamav.gossamer-threads.com	64.69.64.158	Canada	Alex Krohn <mirrors*gossamer-threads.com>
clamav.catt.com	64.18.100.4	USA	Mike Cathey <mirrors*catt.com>
clamav.datahost.com.ar	200.32.4.47	Argentina	Federico Omoto <federico.omoto*datahost.com.ar>
clamav.antispam.or.id	202.134.0.71	Indonesia	Fajar Nugraha <fajar*telkom.co.id>
clamav-du.viaverio.com	199.239.233.95	USA	Scott Wiersdorf <scott*perlcode.org>
clamav-sj.viaverio.com	128.121.60.235	USA	Scott Wiersdorf <scott*perlcode.org>
clamavdb.heanet.ie	193.1.219.100	Ireland	Colm MacCarthaigh <mirrors*heanet.ie>
clamav.crysys.hu	152.66.249.132	Hungary	Bencsath Boldizsar <boldi*mail2004.crysys.hit.bme.hu>

Mirror	IP	Location	Administrator
clamav.rockriver.net	209.94.36.5	Illinois, USA	Thomas D. Harker <tom*rockriver.net>
clamav.xmundo.net	200.68.106.40	Argentina	Cristian Daniel Merz <mirrors*xmundo.net>
clamav.infotex.com	66.139.73.146	Texas, USA	Matthew Jonkman <matt*infotex.com>
clamav.santafesolutions.com	196.40.71.226	Costa Rica	Gregory Cascante Avils <gregory*emailcr.com>
clamav.mirror.transip.nl	80.69.67.3	The Netherlands	Walter Hop <walter*transip.nl>
clamavdb.osj.net	218.44.253.75	Japan	Masaki Ikeda <masaki*orange.co.jp>
clamav.ialfa.net	210.22.201.152	People's Republic of China	Alfa Shen <alfa*ialfa.net>
clamavdb.ikk.sztaki.hu	193.225.86.3	Hungary	Gabor Kiss <kissg*debella.ikk.sztaki.hu>
clamav.mirrors.nks.net	24.73.112.74	Florida, USA	James Neal <clam-admin*nks.net>
clamav.kratern.se	212.31.160.239	Sweden	Emil Ljungdahl <emil*kratern.se>
clamav.dif.dk	193.138.115.108	Denmark	Jesper Juhl <juhl*dif.dk>
clamav.dbplc.com	217.154.108.81	United Kingdom	Simon Pither <simon*digitalbrain.com>
clamav.unet.brandeis.edu	129.64.99.170	USA	Rich Graves <rcgraves*brandeis.edu>
clamav.im1.net	65.77.42.207	Florida, US	Dmitri Pavlenkov <dmitri*im1.com>
clamav.elektrotech-ker.hu	80.95.80.7	Hungary	Bodrogi Zsolt <odin*szilank.hu>
clamav.stockingshq.com	212.113.16.74	United Kingdom	<dave*stockingshq.com>
clamav.acnova.com	203.81.40.167	Singapore	Lennard Seah <myself*lennardseah.com>
clamdb.prolocation.net	213.73.255.243	The Netherlands	Raymond Dijkxhoorn <raymond*prolocation.net>
clamav.xyxx.com	65.75.154.69	San Francisco/Palo Alto California, USA	Myron Davis <myrond*xyxx.com>
clamav.walkertek.com	38.136.139.7	USA	Stephen Walker <swalker*walkertek.com>
clamav.mirror.cygnal.ca	24.244.193.21 24.244.193.22	Burlington, Ontario, Canada	Rafal Rzeczkowski <mirrors*cygnal.ca>
clamav.mirrors.ilisys.com.au	203.202.10.60	Australia	David Wilcox <mirrors*ilisys.com.au>

Mirror	IP	Location	Administrator
clamav.securityminded.net	209.8.40.140	Ashburn, USA	Thomas Petersen <tomp*securityminded.net>
clamav.island.net.au	203.28.142.36	Sydney Australia	Hugh Blandford <hugh*island.net.au>
clamav.iol.cz	194.228.2.38	Czech Republic	Pavel Urban <pavel.urban*imaginet.cz>
clamav.securitywonks.net	66.197.159.213	USA	D. Raghu Veer <clamav*zyserver.net>
clamav.pcn.de	213.203.254.4	Hamburg, Germany	Karsten Gessner <karsten*pcn.de>
clamav.enderunix.org	193.140.143.23	Turkey	Omer Faruk Sen <ofsen*enderunix.org>
clamav.ovh.net	213.186.33.38 213.186.33.37	France	Germain Masse <germain.masse*ovh.net>
clamav.spod.org	195.92.99.99	United Kingdom	Ian Kirk <blob*blob.co.uk>
clamav.intercom.net.ua	195.13.43.28	Ukraine	Artie Missirov <kadjy*intercom.net.ua>
clamav.mirror.vutbr.cz	147.229.3.16	Czech Republic	Tomas Kreuzwieser <mirror-adm*cis.vutbr.cz>
database.clamav.ps.pl	212.14.28.36	Poland	Adam Popik <adam*popik.pl>
clamav.fx-services.com	69.93.108.98	USA	Robin Vley <robin*fx-services.com>
clamav.univ-nantes.fr	193.52.101.131	France	Yann Dupont <yann.dupont*univ-nantes.fr>
clamav.blackroute.net	64.246.44.108	Texas, USA	Maarten Van Horenbeeck <maarten*daemon.be>

In der Konfigurationsdatei finden Sie die Direktive *DatabaseMirror*, mit der Sie festsetzen können, wohin freshclam automatisch verbindet (bis zu *MaxAttempts* Mal). Wenn Sie mehr als eine *DatabaseMirror* Zeile angeben, verwendet das Programm automatisch den nächsten verfügbaren Server, wenn ein Verbindungsproblem auftritt.

3 Verwendung

3.1 Clam Server

clamd ist ein multi-threaded Server und verwendet *libclamav*, um Dateien auf Viren zu prüfen. Das Programm verwendet einen der zwei folgenden Modi zur Kommunikation:

- Unix (local) socket
- TCP socket

Der Server-Prozess ist zur Gänze über die Datei *clamav.conf* konfigurierbar. Sie finden eine Beschreibung aller Direktiven in der *clamav.conf(5)* manpage. *clamd* versteht die folgenden Kommandozeilenparameter:

- **PING**
  Testet den Status des Servers (die Antwort sollte ”PONG” sein).
- **VERSION**
  Liefert Versionsinformationen zurück.
- **RELOAD**
  Lädt die Datenbanken neu.
- **SHUTDOWN**
  Beendet den Server.
- **SCAN Datei/Verzeichnis** Scan einer einzelnen Datei oder eines Verzeichnisses (rekursiv) mit Unterstützung von Archiven. Es muss der volle Pfad angegeben werden (kein relativer!)
- **RAWSCAN Datei/Verzeichnis** Scan einer einzelnen Datei oder eines Verzeichnisses (rekursiv) mit Unterstützung von Archiven. Es muss der voller Pfad angegeben werden (kein relativer!)
- **CONTSCAN Datei/Verzeichnis** Scan einer einzelnen Datei oder eines Verzeichnisses (rekursiv) mit Unterstützung von Archiven. Der Scanvorgang stoppt nicht, wenn ein Virus gefunden wird. Es muss der voller Pfad angegeben werden (kein relativer!)
- **STREAM** Scannt einen Datenstrom - clamd gibt eine neue Port-Nummer zurück, wohin der zu scannende Datenstrom geschickt werden muss. *Das verwendete Protokoll ist überholt und es wird sehr bald durch eine neue Version ersetzt. Die alte Version des Protokolls wird aber weiterhin verwendet werden können.*
- **SESSION, END** Starten/Beenden einer clamd Sitzung. Sie können mehr als ein Kommando pro TCP Sitzung abgeben. (ACHTUNG: aufgrund des Designs von clamd wird die Sitzung bei Eingabe des Befehls **RELOAD** unterbrochen!)

Clamd erkennt die folgenden drei speziellen Signale:

- **SIGTERM** - beendet das Programm normal
- **SIGHUP** - öffnet das Logfile erneut
- **SIGUSR2** - lädt die Datenbank neu

3.2 Clamdscan

clamdscan ist ein einfacher Client für clamd. In vielen Fällen können Sie clamdscan als Alternative zu clamscan verwenden, nur ist dabei folgendes zu beachten:

- Es benötigt beim Ausführen nur clamd.
- Obwohl clamdscan dieselben Kommandozeilenparamter akzeptiert, wie clamscan, werden die meisten davon ignoriert. Zum Beispiel: clamdscan akzeptiert `--mbox`, aber der Befehl hat keinerlei Auswirkung auf den Programmablauf - Sie müssen die Option mittels `ScanMail` in clamd freischalten, um Mail-Dateien scannen zu können.
- clamdscan muss Zugriff auf die zu scannenden Dateien haben.
- clamdscan kann externe Pack-Programme nicht verwenden.

3.3 Clamuko

Clamuko ist ein spezieller Programmteil vom *clamd*, der Dateien direkt beim Zugriff scannt - unter Linux und FreeBSD. Es ist ausschliesslich als Teil von clamd konzipliert und kann aufgrund der Dazuko Implementation nicht von einem clamd Client verwendet werden. Die aktuelle Implementierung bietet allerdings einige Vorteile: clamuko greift auf dieselbe interne Virendatenbank zu, wie clamd. Ausserdem gilt der RELOAD Befehl auch für clamuko. **Sie müssen einige wichtige Grundsätze beachten, wenn Sie clamuko verwenden:**

- Beenden Sie den Server immer normal - mit dem `QUIT` Befehl oder dem `SIGTERM` Signal. Es kann sonst passieren, dass Sie auf geschützte Dateien sonst keinen Zugriff mehr haben, bis das System neu gestartet ist!
- Verwenden Sie clamuko niemals, um ein Verzeichnis zu schützen, das Ihre mail-Scanner Software zum Entpacken von Mail Anhängen verwendet. Der Zugriff auf alle infizierten Dateien wird sonst verweigert und der Scanner (sogar clamd) hat keine Gelegenheit, einen Virus zu erkennen. **Alle infizierten Mails werden zugestellt!**

Sie müssen clamuko in der *clamav.conf* aktivieren. Um beispielsweise das Verzeichnis `/home` zu schützen, verwenden Sie folgenden Befehl:

```
ClamukoIncludePath /home
```

Zum Schutz des gesamten Systems hingegen fügen Sie der Konfigurationsdatei dieses hinzu:

```
ClamukoIncludePath /
ClamukoExcludePath /proc
ClamukoExcludePath /temporaeres/verzeichnis/Ihrer/email/Scanner/Software
```

Sie können clamuko auch verwenden, um Ihre via Samba/Netatalk verwalteten Dateien zu schützen (wobei eine weitaus bessere und sicherere Methode die Verwendung des ausgezeichneten **samba-vscan** Moduls (siehe 5.16) ist). NFS wird nicht unterstützt (Dazuko fängt die speziellen NFS Befehle nicht ab). Eine weitere Idee: füllen Sie eine Datenbank mit Signaturen von populären Exploits und konfigurieren Sie clamd so, dass Ihr Server vor script-kiddies geschützt ist.

3.4 Archive und komprimierte Dateien

Alle ClamAV Scanner Komponenten sind von der zentralen Bilbliothek LibClamAV abhängig. Diese hat Unterstützung für die folgenden Formate bereits fest integriert:

- Zip
- Gzip
- Bzip2
- RAR (nur Version 2.0)

Archivtypen werden von LibClamAV anhand von magic number Tests bestimmt. [8] Sie benötigen die `zlib` Bibliothek für Zip/Gzip Unterstützung. Auf Zip Archive wird mit Hilfe der `zziplib` Bibliothek von Guido Draheim und Tomi Ollila zugegriffen. Unterstützung für RAR Archive basiert auf der `UniquE RAR File` Bibliothek von Christian Scheurer und Johannes Winkelmann. Beide Bibliotheken sind in den Quelldateien von ClamAV in leicht modifizierter Form inkludiert. `Unrarlib` unterstützt derzeit nur RAR 2.0 Archive. Christian meint dazu, dass das neue Format (mit WinRAR 3.0 eingeführt) auch in Zukunft von seiner Bibliothek nicht angeboten werden wird. Dennoch kann ClamAV WinRAR 3.0 Archive lesen - mehr dazu weiter unten. Aufgrund von Sicherheitüberlegungen scannt clamd nur jene Archive, die es mit ”Bordmitteln” lesen kann und hat keine Unterstützung für externe Programme. Clamscan ist hier unabhängiger und kann auf einen externen Entpacker zurückgreifen, wenn die eingebaute Dekompression versagt: the external unpacker when the built-in decompresor fails:

```
$ clamscan --unrar rarfail.rar
/home/zolw/Clam/test/rarfail.rar: RAR module failure.
```

[8] Das funktioniert ähnlich dem bekannten file(1) Befehl.

```
UNRAR 3.00 freeware      Copyright (c) 1993-2002 Eugene Roshal


Extracting from /home/zolw/Clam/test/rarfail.rar

Extracting  test1                                                     OK
All OK
/tmp/44694f5b2665d2f4/test1: ClamAV-Test-Signature FOUND
/home/zolw/Clam/test/rarfail.rar: Infected Archive FOUND
```

TIP: *Sie können clamscan zwingen, Viren in allen von Archiven anzuzeigen, indem Sie die Parameter –disable-archive verwenden (damit werden alle eingebauten Entpacker deaktiviert) und externe Entpacker manuell spezifizieren: –unzip –unrar... .*

Wird der Scanner mit Superuser-Privilegien (als root) gestartet, werden externe Programme unter dem *clamav* Benutzer ausgeführt, was das Entpacken weitaus sicherer macht. Damit ist ausserdem sichergestellt, dass der *clamav* Benutzer Lese-Zugriff auf alle Dateien hat. Sie müssen rekursives Scannen mittels der Option `-r` bzw. `--recursive` aktivieren, um den gesamten Inhalt eines Archives inklusive aller Unterverzeichnisse zu durchsuchen. Diese Option wird normalerweise auch benötigt, um verschachtelte Archive zu durchsuchen.
Es werden die folgenden externen Entpacker unterstützt:

–unzip: Üblicherweise benötigen Sie diese Option nicht, da das Zip Format von libclamav unterstützt wird. Es kann allerdings in Situationen hilfreich sein, in denen libclamav ein Zip Archiv nicht öffnen kann. clamscan wurde getestet mit *UnZip 5.41 vom 16.04.2000 von Info-ZIP*.

–unrar: Getestet mit *UNRAR 3.00 freeware*.

–unace: Verwendet die Optionen von *UNACE v1.2 public version*, was wir zwar nicht testen konnten, es sollte aber funktionieren.

–arj: Getested mit *arj 3.10b*.

–zoo: Getested mit *zoo 2.1*.

–lha: Getested mit *LHa for Unix V 1.14e*.

–jar: clamscan verwendet *unzip* zum Entpacken von .jar Dateien. Getestet mit *UnZip 5.41 vom 16.04.2000 von Info-ZIP*.

–tar: Diese Option bietet Unterstützung für unkomprimierte Archive. Getestet mit *GNU tar 1.13.17*.

–deb: Mit dieser Option schalten Sie die Unterstützung von Debian Paketen ein. Getestet mit *GNU ar 2.12.90.0.14*. Impliziert –tgz, kollidiert aber nicht mit –tgz=FULLPATH

–tgz: Mit dieser Option können .tar.gz und .tgz Dateien gescannt werden. Sie benötigen dazu *GNU tar*, das Sie auf nicht-Linux Systemen vermutlich als *gtar* finden werden. clamscan sucht im *$PATH* nach einem geeigneten tar Programm. Im Zweifelsfall legen Sie daher mit –tgz=gtar fest, dass clamscan *gtar* statt *tar* verwenden soll. Andernfalls übergeben Sie einen vollen Pfad zu Ihrem tar Programm.

3.5 Mail Dateien

Unterstützung für Mail-Dateien ist standardmässig deaktiviert. Um diese zu aktivieren, verwenden Sie bitte den Parameter `--mbox` in clamscan und unkommentieren gleichzeitig die Direktive `ScanMail` in `clamav.conf` (für clamd). Alle gängigen Mail-Formate (Mbox, Maildir, etc.) werden unterstützt. Die Funktion Mail Scan ist noch in aktiver Entwicklung und kann u.U. zu Stabilitätseinbussen führen. Wenn Sie damit Probleme beobachten, schicken Sie bitte email-Beispiele, die zum Fehlverhalten führen, an Nigel Horne `<njh*clamav.net>`. Sie benötigen diese Funktion nicht, wenn Sie bereits einen wrapper vie AMaViS verwenden, denn dieser übernimmt bereits die MIME Dekodierung für uns.

3.6 Format der Ausgabe

clamd verwendet ein Ausgabeformat, dass zu clamscan kompatibel ist:

```
zolw@Wierszokleta:~$ telnet localhost 3310
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
SCAN /home/zolw/infected
/home/zolw/infected/sobre.com: W32.Magistr.B FOUND
Connection closed by foreign host.
```

Im Modus **SCAN** wird die Verbindung sofort geschlossen, wenn der erste Virus identifiziert ist. Im Falle von Archiven ist die Ausgabe exakt gleich aufgebaut, da der eingebaute Archivsupport transparent ist:

```
SCAN /home/zolw/Clam/test/test2.zip
/home/zolw/Clam/test/test2.zip: ClamAV-Test-Signature FOUND
```

CONTSCAN zeigt alle gefundenen Viren an.
Fehlermeldungen werden im folgenden Format ausgegeben:

```
SCAN /no/such/file
/no/such/file: Can't stat() the file ERROR
```

und können somit einfach analysiert werden.

clamscan schreibt alle Nachrichten nach **stderr** (nur die Hilfe wird standardmässig nach **stdout** geschrieben). Sie können diese Ausgabe natürlich an beliebige Stelle umleiten - das wird vom Parameter `--stdout` unterstützt. Ein Beispiel für die Ausgabe von clamscan ist:

```
/tmp/test/removal-tool.exe: Worm.Sober FOUND
/tmp/test/md5.o: OK
/tmp/test/blob.c: OK
/tmp/test/message.c: OK
/tmp/test/error.hta: VBS.Inor.D FOUND
```

Findet clamscan einen Virus, schreibt er dessen Bezeichnung zwischen die Einträge `Dateiname:` und das Wort `FOUND`. Wenn ein Virus in einem Archiv gefunden wird, das von einem externen Entpacker geoeffnet wurde, zeigt clamscan das mit `Infected Archive` an. Mit "Infected Archive" gekennzeichnete Treffer werden nicht als infizierte Dateien gezählt - nur die enthaltenen Datein. Bitte beachten Sie hier den feinen Unterschied zum Scan von Archiven, die vom eingebauten Entpacker geoeffnet werden (dieser Prozess geschieht vollkommen transparent): clamscan weiss nicht, welche Datei genau infiziert ist und markiert das gesamte Archiv als infiziert.

3.7 Signatur Werkzeug

Sigtool automatisiert die Erstellung neuer Signaturen. Wenn Sie eine infizierte Datei finden, die von ClamAV noch nicht erkannt wird, von einer anderen Antivirus-Software aber schon, hilft Ihnen dieses Werkzeug, eine Signatur für den Virus automatisch zu erstellen.[9] *Sigtool* ist nur teilweise andwendbar, da es nur den letzten Teil einer echten Signatur erkennt. Bei mehrteiligen Signaturen (und insbesonders bei polymorphen Viren) wird die Erstellung einer brauchbaren Signatur fehlschlagen. Anwendungsbeispiel: erstellen Sie eine Datei (mit willkürlichem Inhalt) und fügen Sie an beliebiger Stelle den Text `test/test1` in die Datei ein. Wir verwenden im Beispiel nun *clamscan*, um eine Signatur zu erstellen. Bitte beachten Sie, dass das nur ein Beispiel ist - im realen Leben

[9] Stellen Sie sicher, daš die Lizenz Ihres kommerziellen Scanners die Verwendung von SigTool nicht verbietet!

benötigen Sie natürlich keine derartigen Tricks, sondern nur die infizierte Datei. Scannen Sie es nun mit `clamscan --stdout testfile` - die Ausgabe sollte in etwa so aussehen:

```
testfile: ClamAV-Test-Signature FOUND

----------- SCAN SUMMARY -----------
Known viruses: 21074
Scanned directories: 0
Scanned files: 1
Data scanned: 0.95 MB
Infected files: 1
I/O buffer size: 131072 bytes
Time: 1.245 sec (0 m 0 s)
```

Die eindeutige Textfolge in der Ausgabe ist ”ClamAV-Test-Signature”. Starten Sie daher *sigtool* wie folgt:

```
$ sigtool -c "clamscan --stdout" -f testfile -s "ClamAV-Test"
```

Das Programm wird die beiden Parameter für `-c (--command)` und `-f (--file)` verbinden, daher ist die Reihenfolge, in der Sie Parameter an sigtool übergeben, besonders wichtig. Im Endeffekt wird die Datei *testfile.sig* erstellt, die genau 100 Byte gross sein sollte (in unserem Beispiel). Diese Datei enthält die richtige Signatur.

```
Detected, decreasing end 20051 -> 16040
Detected, decreasing end 16040 -> 12029
Detected, decreasing end 12029 -> 8018
Not detected at 8018, moving forward.
Detected, decreasing end 10024 -> 8018
Not detected at 8018, moving forward.
Detected, decreasing end 9021 -> 8018
Not detected at 8018, moving forward.
Not detected at 8520, moving forward.
Detected, decreasing end 8771 -> 8520
Not detected at 8520, moving forward.
Not detected at 8646, moving forward.
Not detected at 8709, moving forward.
Detected, decreasing end 8741 -> 8709
Not detected at 8709, moving forward.
Not detected at 8725, moving forward.
```

```
Detected, decreasing end 8733 -> 8725
Not detected at 8725, moving forward.
Not detected at 8729, moving forward.
Detected, decreasing end 8731 -> 8729
Not detected at 8729, moving forward.
Detected, decreasing end 8730 -> 8729
Not detected at 8729, moving forward.
Increasing end 8729 -> 8730
 *** Signature end found at 8730
Detected at 8680, moving forward.
Detected at 8680, moving forward.
Not detected, moving backward 8693 -> 8680
Detected at 8680, moving forward.
Not detected, moving backward 8687 -> 8680
Detected at 8680, moving forward.
Not detected, moving backward 8684 -> 8680
Detected at 8680, moving forward.
Not detected, moving backward 8682 -> 8680
Detected at 8680, moving forward.
Not detected, moving backward 8681 -> 8680
Detected at 8680, moving forward.
Not detected, moving backward 8681 -> 8680
Detected at 8680, moving forward.
Moving forward 8680 -> 8681
 *** Signature start found at 8681

The scanner was executed 33 times.
The signature length is 49 (98 hex)
Saving signature in testfile.sig file.
Saving binary signature in testfile.bsig file.
```

Um die neue Signatur fertigzustellen, fügen Sie jetzt bitte noch die `VirusName=` Zeile am Anfang der hexadezimalen Signatur in der Datei `testfile.sig` hinzu.

TIP: *ClamAV Scanner lesen alle* .db *Dateien im Datenbank Verzeichnis. Sie können daher auf die Beschriebene Art mühelos eigene Datenbanken (z.B.* local.db*) erstellen, die beim Update durch freshclam nicht modifiziert werden!*

4 Problemlösung

4.1 Rückgabewerte (return codes)

Rückgabewerte sind zur Programmierung von Scripts sehr praktisch. Sie können den Rückgabewert von *clamscan* prüfen, indem Sie die Systemvariable $? befragen:

```
$ clamscan; echo Return code: $?
```

Hier ist die komplette Liste aller Rückgabewerte von *clamscan*:

0: Es wurde kein Virus gefunden.

1: Zumindest ein Virus wurde gefunden.

40: Eine unbekannte Option wurde an *clamscan* übergeben. Bitte prüfen Sie die Hilfeseite clamscan –help oder das Handbuch für eine Liste aller gültigen Optionen.

50: Fehler beim Intitialisieren der Virendantenbank. Vielleicht existiert diese nicht am Standard-Speicherort oder aber die Datenbank selbst ist ungültig (vielleicht aufgrund einer kaputten digitalen Signatur).

52: Dateityp wird nicht unterstützt - *clamscan* kann reguläre Dateien, Verzeichnisse und Symlinks verarbeiten.

53: Verzeichnis kann nicht geöffnet werden.

54: Datei kann nicht geöffnet werden.[7]

55: Eingabe/Ausgabe Fehler während des Lesens. [10]

56: Die Datei oder das Verzeichnis, das Sie scannen möchten, existiert nicht.

57: Kann den absoluten Pfad des Arbeitsverzeichnisses nicht bestimmen. Vermutlich ist der aktuelle Pfadname länger als 200 Zeichen; das ist schlecht - bitte kompilieren Sie ClamAV erneut.

58: Eingabe/Ausgabe Fehler. Bitte prüfen Sie das Dateisystem.

59: Kann Informationen zum aktuellen Benutzer (jener, der *clamscan* aufruft) nicht bestimmen.

60: Kann Informationen zum Benutzer *clamav* nicht bestimmen. Der UNIX-Benutzer *clamav* existiert vermutlich nicht in der /etc/passwd.

61: Cannot fork. Kann keinen neuen Prozess erzeugen - bitte prüfen Sie die Systemlimits.

[10] Wird nur im Einzeldatei-Modus gemeldet (beim rekursiven Scannen werden derartige Fehler ignoriert).

63: Kann keine temporären Dateien/Verzeichnisse erzeugen. Bitte prüfen Sie die Permissions von /tmp oder verwenden Sie –tempdir.

64: Kann in das temporäre Verzeichis nicht schreiben. Bitte geben Sie ein anderes an.

70: Kann keinen Arbeitsspeicher zuordnen bzw. freigeben. Das ist ein kritischer Fehler, bitte prüfen Sie Ihr System!

71: Kann keinen Arbeitsspeicher zuordnen. Siehe oben.

5 Software von Drittanbietern

Es existieren viele Projekte, die unseren Scanner unterstützen. Hier finden Sie eine Liste von Programmen, deren Zusammenspiel mit ClamAV getestet und für gut befunden worden ist.

5.1 clamav-milter

Website: Teil des ClamAV Pakets
Unterstützt: clamd

clamav-milter ist ein sehr schneller Mail Scanner, der von Nigel Horne für *sendmail* geschrieben wurde. Das Projekt ist vollständig in C programmiert und verwendet ClamAV's internen mail Scanner (der /"ubrigens ebenfalls von Nigel geschrieben wurde).

Installation:
Sie benötigen die *libmilter* Entwicklungsdateien. Konfigurieren Sie ClamAV mit

```
$ ./configure --enable-milter
```

und rekompilieren die Software. Das Programm wird dann als `/usr/local/sbin/clamav-milter` in Ihrem System installiert. Die folgende Anleitung stammt ursprünglich aus Nigel's INSTALL Datei:
Fügen Sie der Datei `/etc/mail/sendmail.mc` folgende Zeilen hinzu:

```
INPUT_MAIL_FILTER('clmilter','S=local:/var/run/clmilter.sock,
F=, T=S:4m;R:4m')dnl
define('confINPUT_MAIL_FILTERS', 'clmilter')
```

Prüfen Sie die folgenden Einträge in der `clamav.conf`:

```
LocalSocket /var/run/clamd.sock
ScanMail
StreamSaveToDisk
```

Starten Sie *clamav-milter*:

```
/usr/local/sbin/clamav-milter -lo /var/run/clmilter.sock
```

und starten schlussendlich *sendmail* neu.

5.2 IVS Milter

Website: `http://ivs-milter.lbsd.net`
Unterstützt: clamd

IVS Milter ist ein milter Scanner gegen SPAM und Viren. Der Name stehe für ”Industrial Virus + Spam milter”. Das Projekt wurde geschaffen, um in Umgebungen jeder Grösse eingesetzt zu werden - vom Heimanwender bis zu grossen ISPs.

5.3 smtp-vilter

Website: `http://www.etc.msys.ch/software/smtp-vilter`
Unterstützt: clamd

smtp-vilter ist ein hochperformanter Inhalts-Filter (content scanner) für *sendmail*, der die milter API verwendet. Die Software scannt Mail Nachrichten auf Viren und markiert oder verwirft infizierte Nachrichten. ClamAV ist standardmässig als Scanner voreingestellt.

5.4 mod_clamav

Website: `http://software.othello.ch/mod_clamav`
Unterstützt: libclamav, clamd

mod_clamav ist ein Virus Filter für den populären *Apache* Webserver. Das Projekt stammt von Andreas Müller, der auch die laufende Pflege der Software übernimmt. Das Projekt ist sehr gut dokumentiert und die Installation gestaltet sich sehr einfach.

5.5 AMaViS - ”Next Generation”

Website: `http://sourceforge.net/projects/amavis`
Unterstützung: clamscan

AMaViS-ng ist ein vollständig überarbeitete, modularere Version des Projekts *amavis-perl/amavisd*. Entwickelt wurde die Software von Hilko Bengen. Ein kurzer Auszug von der Website des Projekts:

Installation:

Bitte laden Sie die neueste Version der Software (zumindest 0.1.4). Nach der recht einfachen Installation entkommentieren Sie bitte die folgende Zeile in der `amavis.conf`:

```
virus-scanner = CLAM
```

und passen bei Bedarf den Pfad zu *clamscan* unter `[CLAM]` an:

```
[CLAM]

clamscan = /usr/local/bin/clamscan
```

5.6 amavisd-new

Website: `http://www.ijs.si/software/amavisd`
Unterstützt clamd, clamscan

amavisd-new ist eine von Mark Martinec neu geschriebene Version von *amavis*.

Installation:
clamscan wird automatisch aktiviert, wenn das entsprechende Programm beim Start der Software gefunden wurd. *clamd* wird aktiviert, indem der zuhgehörige Eintrag in `@av_scanners` in der Konfigurationsdatei `/etc/amavisd.conf` entkommentiert wird.

5.7 Qmail-Scanner

Website: `http://qmail-scanner.sf.net`
Unterstützt clamscan

Bitte erhöhen Sie den `softlimit` Wert, wenn Sie dieses Projekt mit *clamscan* verwenden möchten.

5.8 Sagator

Website: `http://www.salstar.sk/sagator`
Unterstützt clamscan, clamd, libclamav

Sagator ist ein Mail AntiVirus- und AntiSPAM-Gateway. Es arbeitet als Schnittstelle

zu *postfix* (oder einem anderen smtpd) und ruft AntiVirus- und AntiSPAM Programme bei Bedarf modular auf. Es kann je nach Konfiguration jede beliebige Kombination an AntiVirus- und AntiSPAM-Software eingesetzt werden.

5.9 ClamdMail

Website: `http://clamdmail.sf.net`
Unterstützt clamd

Ein Mail Filter für ClamAV. Klein, schnell und leicht zu installieren.

5.10 BlackHole

Website: `http://iland.net/~ckennedy/blackhole.shtml`
Unterstützt clamscan, clamd

BlackHole ist ein fortschrittlicher SPAM-/Virusfilter für Qmail, Postfix, Sendmail, Exim und Courier. Dieses Tool, das von Chris Kennedy entworfen wurde, richtet sich an den geübten Administrator (in anderen Worten: die Installation ist komplex...).

5.11 MailScanner

Website: `http://www.mailscanner.info`
Unterstützt clamscan

Mailscanner scannt Mails nach Viren, SPAM, Attacken und Sicherheitsrisken. Es ist nicht von einem bestimmten AntiVirus Scanner abh¨angig, sondern kann mit einer Kombination von bis zu 14 verschiedenen Scannern verwendet werden, sodass Sie die für Sie ideale Kombination selbst wählen können.

5.12 MIMEDefang

Website: `http://www.roaringpenguin.com/mimedefang`
Unterstützt clamscan, clamd

MIMEDefang ist ein effizienter Mail Scanner für *Sendmail/Milter*.

5.13 exiscan

Website: `http://duncanthrax.net/exiscan`
Unterstützt clamscan, clamd

exiscan ist ein Patch für *exim* Version 4, der Unterstützung für das Scannen von Mail

Kopfzeilen enthält. Vier verschiedene Methoden zum Filtern könenn verwendet werden: antivirus, antispam, Reguläre Ausdrücke, Dateierweiterungen.

5.14 scanexi

Website: `http://w1.231.telia.com/~u23107873/scanexi.html`
Unterstützt clamscan, clamd

scanexi ist ein Plugin für *exim* Version 4.14 mit dem dlopen Patch. Es bietet Unterstützung für den Scan von Mail Inhalten.

5.15 Mail::ClamAV

Website: `http://cpan.gossamer-threads.com/modules/by-authors/id/S/SA/SABECK/`
Unterstützt libclamav

Perl Erweiterung für das *libclamav* Bilbliothek.

5.16 OpenAntiVirus samba-vscan

Website: `http://www.openantivirus.org/projects.php#samba-vscan`
Unterstützt clamd

samba-vscan provides on-access scanning of Samba shares. It supports Samba 2.2.x/3.0 with working virtual file system (VFS) support.

5.17 Sylpheed Claws

Website: `http://claws.sylpheed.org`
Unterstützt libclamav

Sylpheed Claws ist ein topaktueller Entwicklungszweig von *Sylpheed*, einem leichtgewichtigen Mailclient für UNIX. Das Programm filtert Attachments in Mails, die es via POP, IMAP oder über einen lokalen Account. Optional löscht *Sylpheed Claws* gefundene Mails oder speichert diese in einen Ordner.

5.18 nclamd

Website: `http://www.kyzo.com/nclamd/`
Unterstützt libclamav

nclamd, *nclamav-milter* und *nclamdscan* sind neu geschriebene Versionen der originalen Werkzeuge. Sie verwenden Prozesse statt threads und implementieren *ripMIME* anstatt des in *ClamAV* eingebauten MIME Dekoders.

5.19 cgpav

Website: `http://program.farit.ru`
Unterstützt clamd

Bei *cgpav* handelt es sich um ein perfomantes (in C programmiertes) Plugin für *CommuniGate Pro* mit Unterstützung für *clamd*.

5.20 j-chkmail

Website: `http://j-chkmail.ensmp.fr`
Unterstützt libclamav, clamd

j-chkmail ist ein schneller (in C programmierter) Filter für Sendmail. Es filtert gefährliche Inhalte (Viren) und SPAM mithilfe vom *ClamAV*. Das Programm bietet eine Vielzahl an Optionen und wurde für den Einsatz auf Servern mit hoher Last konzipiert. *j-chkmail* ist Open Source Software und für den nichtkommerziellen Einsatz durch registrierte Anwender frei verfügbar.

5.21 qscanq

Website: `http://budney.homeunix.net:8080/users/budney/software/qscanq/index.html`
Unterstützt clamscan

qscanq ist ein Ersatz für *qmail-queue*. Es startet den Scan eingehender Mails mithilfe von clamscan oder clamdscan und meldet den exit status des Scanners oder von *qmail-queue* an den Aufrufer zurück.

5.22 clamavr

Website: `http://raa.ruby-lang.org/list.rhtml?name=clamavr`
Unterstützt libclamav

Ruby binding für *ClamAV*.

5.23 DansGuardian Anti-Virus Plugin

Website: `http://www.pcxperience.org/dgvirus/`
Unterstützt clamscan

DG AVP ist ein Zusatz zum filternden Web-Proxy *DansGuardian* und ermöglicht dort das Filtern von Viren. Das Programm wurde unter der GPL lizensiert.

5.24 Viralator

Website: `http://viralator.sourceforge.net/`
Unterstützt clamscan

Viralator, in PERL geschrieben, ist ein Programm, das HTTP downloads, die den Squid Proxy passieren, auf Viren prüft.

5.25 TrashScan

Website: clamav-sources/support/trashscan
Unterstützt: clamscan

TrashScan ist ein auf procmail basierender Scanner der Firma Trashware, der besonders einfach zu installieren ist. Allerdings ist er nur für Endanwender gedacht und nicht so effizient, wie Scanner, die direkt mit dem MTA zusammenarbeiten.

5.26 ClamAssassin

Website: `http://drivel.com/clamassassin/`
Unterstützt clamscan

Bei *ClamAssassin* handelt es sich um ein einfaches Skript zum Filtern von Viren mittels clamscan. Es arbeitet ähnlich dem beliebten *spamassassin* und wurde für die Verwendung mit procmail konzipiert.

5.27 clamscan-procfilter

Website: `http://www.virtualblueness.net/~blueness/clamscan-procfilter/`
Unterstützt: clamscan

Ein Filter für procmail, der die Zusammenarbeit mit clamscan ermöglicht. Ein neues Mail-Header-Feld, `X-CLAMAV`, das eine Liste aller gefundenen Viren enthält, wird in verseuchte Mails eingefügt.

5.28 MyClamMailFilter

Website: `http://muncul0.w.interia.pl/projects.html#myclammailfilter`
Unterstützt: clamscan

MyClamMailFilter ist ein Mail-Filter für procmail oder maildrop. Sobald es einen Virus

findet, wird der Mail-Anhang umbenannt, sowie die Betreff-Zeile modifiziert. Der Filter bietet darüberhinaus die Möglichkeit, potentiell gefährliche Mail-Anhaenge (diese werden anhand der Datei-Erweiterung erkannt) umzubenennen. Die Software ist simpel, performant und einfach anzupassen.

5.29 Gadoyanvirus

Website: `http://oss.mdamt.net/gadoyanvirus/`
Unterstützt libclamav

Gadoyanvirus ist ein weiterer Viren-Blocker für qmail. Es ersetzt das originale Programm qmail-queue und filtert eingehende emails mithilfe der Antivirus-Bibliothek von *ClamAV*. Verdächtige Mails werden in Quarantäne gestellt, und der ursprüngliche Empfänger wird darüber (optional) benachrichtigt. Standardmässig benötigt *Gadoyanvirus* eine mit QMAILQUEUE gepatchte qmail Installation.

5.30 OpenProtect

Website: `http://opencompt.com/`
Unterstützt ClamAV via MailScanner

OpenProtect ist eine serverseitige Mail-Schutzeinrichtung, die aus *MailScanner*, *Spamassasin* und *ClamAV* besteht und Unterstützung für *Sendmail*, *Postfix*, *Exim* und *qmail* bietet. Das Programm besteht des weiteren aus einer vollständig automatisierten (De-) Installationsroutine, die alles automatisch konfiguriert und sich auch um die Installation von PERL Modulen und dem Konfigurieren der Virenscanner kümmert.

5.31 RevolSys SMTP kit for Postfix

Website: `http://smtp.revolsys.org/`
Unterstützt: ClamAV via amavisd-new

Das *RevolSyS SMTP kit for Postfix* installiert AntiSPAM und AntiVirus Werkzeuge wie *amavisd-new*, *Spamassassin*, *ClamAV* und *Razor*. Das Ziel dieses Kits ist es, vorhandene Postfix Mail-Server zu um die beschriebenen Funktionen zu erweitern.

5.32 POP3 Virus Scanner Daemon

Website: `http://p3scan.sourceforge.net/`
Unterstützt clamscan

Der *POP3 Virus Scanner Daemon* ist ein transparenter Proxy Server für POP3 Clients,

der unter Linux mit iptables (für die Port-Umleitung) läuft. Das Programm kann verwendet werden, um den Mail-Verkehr via POP3 aus dem Internet in interne Netzwerke abzusichern.

5.33 mailman-clamav

Website: `http://www.tummy.com/Software/mailman-clamav`
Unterstützt clamd

Das Modul *mailman-clamav* erweitert *GNU Mailman* um die Funktion, eingehende Nachrichten mittels *ClamAV* auf Viren zu prüfen. Das Programm erlaubt es, *Mailman* so zu konfigurieren, dass verseuchte Nachrichten entweder verworfen oder festgehalten werden können. Besonders nützlich ist die Option zum Verwerfen von Nachrichten - Listen-Administratoren müssen sich so nicht mehr händisch um die Bearbeitung von Viren in Nachrichten kümmern.

5.34 wbmclamav

Website: `http://wbmclamav.labs.libre-entreprise.org/`
Unterstützt ClamAV

wbmclamav ist ein Modul für *Webmin*, um *Clam Antivirus* zu verwalten. Es wurde von Emmanuel Saracco geschrieben.

5.35 Scan Log Analyzer

Website: `http://pandaemail.sourceforge.net/av-tools/`
Unterstützt ClamAV

Mit *Scan Log Analyzer* können Sie eine graphische Repräsentation der Logdateien erstellen und ansehen. Es unterstützt *RAV*, *ClamAV* und *Vexira*.

5.36 mailgraph

Website: `http://people.ee.ethz.ch/~dws/software/mailgraph/`
Unterstützt clamd

mailgraph ist ein einfaches Statistikpake für *RRDtool*. Es unterstützt *Postfix* und produziert tägliche, wöchentliche, monatliche und jährliche Graphen der ein- und ausgehenden bzw. der gebouncten und abgelehnten Mails (SMTP Datenverkehr).

5.37 INSERT

Website: http://www.inside-security.de/INSERT_en.html
Unterstützt ClamAV

INSERT (the Inside Security Rescue Toolkit) zielt darauf ab, ein multifunkionelles, universelles Werkzeug zur disaster recovery und Netzwerk-Systemanalyse zu bieten. Das System bootet von einer kreditkartengrossen CD-ROM und besteht im Prinzip aus einer abgespeckten Version von *Knoppix*. *INSIDE* bietet gute Hardware-Erkennung, fluxbox, emelfm, links-hacked, ssh, tcpdump, nmap, chntpwd und viele weitere Programme. Weiters ermöglicht es vollen Lese- und Schreibzugriff auf NTFS Partitionen (mittels *captive*) und es enthält den *ClamAV* Virenscanner (inclusive der Signaturdatenbank).

5.38 Local Area Security

Website: http://www.localareasecurity.com/
Unterstützt ClamAV

Local Area Security Linux ist eine Live CD Distribution mit Schwerpunkt auf sicherheitsrelevante Werkzeuge und geringer Gesamtgrösse. Die Distribution bietet unter anderem die Möglichkeit, *ClamAV* direkt von CD-ROM zu starten.

5.39 ClamWin

Website: http://clamwin.sourceforge.net/
Unterstützt: clamscan, freshclam

ClamWin verleiht dem Clam AntiVirus Scanner eine graphische Benutzeroberfläche. Es ermöglicht das Auswählen und Scannen einzelner Dateien und ganzer Verzeichnisse, die Konfiguriation vieler Parameter, sowie das Update der Virus-Datenbank. Letzteres auch vollautomatisch nach dem Anmelden des Benutzers am Windows-System. Weiters ist ein Symbol für die Windows Taskbar tray (rechts unten neben der Uhrzeit-Anzeige) inkludiert. ClamWin erweitert den Windows Explorer um kontextsensitive Einträge im Menu, das bei einem Rechtsklick erscheint. So kann man z.B. den Scan einer Datei direkt aus dem Windows Explorer heraus starten. Die Software wird mit einem Installationsprogramm angeboten, das mittels *InnoSetup* erstellt wurde. Benötigte Cygwin DLL Dateien sind inkludiert.

5.40 KlamAV

Website: http://sourceforge.net/projects/klamav/
Unterstützt: ClamAV

Eine Sammlung an graphischen Werkzeugen (GUI tools), um ClamAV unter KDE einzusetzen. *Klamscan*, eine KDE Oberfläche für clamscan, ist via CVS erhältlich. In der absehbaren Zukunft wird KlamAV voraussetzen, daš ClamAV auf Ihrem Computer bereits installiert ist. Wir hoffen, daš KlamAV bald auch freshklam, ein sigtool Werkzeug, enthalten wird, sowie ein Interface für clamuko (das eine Art von 'auto-protect' Scannen ermöglichen sollte).

6 LibClamAV

libclamav kann dazu verwendet werden, um Ihre Software um Antivirus-Funktionalität zu erweitern. Die Bibliothek ist thread-safe, kann Archive, Mail-Dateien, sowie MS Office Dokumente transparent erkennen und durchsuchen,

6.1 API

Jedes auf *libclamav* basierte Programm muss die Header-Datei `clamav.h` einbinden:

```
#include <clamav.h>
```

Ein erster Schritt besteht darin, die scanning engine zu initialisieren. Es stehen drei Funktionen zur Verfügung:

```
int cl_loaddb(const char *filename, struct cl_node **root,
int *virnum);

int cl_loaddbdir(const char *dirname, struct cl_node **root,
int *virnum);

const char *cl_retdbdir(void);
```

`cl_loaddb()` lädt eine bestimmte Datenbank, `cl_loaddbdir()` lädt alle *.cvd* (und die älteren *.db*, .db2) Datenbanken aus dem Verzeichnis `dirname`. `cl_retdbdir()` liefert einen fixen (hardcoded) Verzeichnispfad für Datenbanken zurück. Die initiale interne Datenbank (Aho-Corasick tree, trie; siehe 6.3) wird unter `root` gespeichert, und diverse Signaturen werden nach `virnum` **hinzugefügt** werden.[11] Der Zeiger zum trie muss anfäglich auf NULL zeigen. Wenn Sie die Anzahl aller gelandenen Signaturen nicht kümmert, übergeben Sie einfach NULL als drittes Argument. `cl_loaddb` gibt 0 bei Erfolg und einen anderen Wert bei Fehler zurück.

[11] Vergessen Sie nicht, die Variable virus counter mit 0 zu initialiseren.

```
    struct cl_node *root = NULL;
    int ret;

ret = cl_loaddbdir(cl_retdbdir(), &root, NULL);
```

Es gibt einen eleganten Weg, um die Fehlercodes von libclamav anzeigen zu lassen:

```
const char *cl_strerror(int clerror);
```

`cl_strerror()` returns a (statically allocated) string describing a `clerror` code:

```
if(ret) {
    printf("cl_loaddbdir() error: %s\n", cl_strerror(ret));
    exit(1);
}
```

Sobald zumindest eine Datenbank geladen ist, erstellen Sie den fertigen trie wie folgt:

```
int cl_buildtrie(struct cl_node *root);
```

In unserem Beispiel:

```
if((ret = cl_buildtrie(root)))
    printf("cl_buildtrie() error: %s\n", cl_strerror(ret));
```

Jetzt können Sie einen Puffer, einen descriptor oder eine Datei scannen:

```
int cl_scanbuff(const char *buffer, unsigned int length,
const char **virname, const struct cl_node *root);

int cl_scandesc(int desc, const char **virname, unsigned
long int *scanned, const struct cl_node *root, const
struct cl_limits *limits, int options);

int cl_scanfile(const char *filename, const char **virname,
unsigned long int *scanned, const struct cl_node *root,
const struct cl_limits *limits, int options);
```

All diese Funktionen speichern den Namen eines etwaigen Virus unter dem Zeiger `virname`. Dieser zeigt auf den Namen in der trie Struktur und kann daher nicht direkt freigegeben werden. `cl_scandesc()` und `cl_scanfile()` können den Wert `scanned` erhöhen, und zwar in `CL_COUNT_PRECISION` Einheiten. Diese beiden unterstützen auch Archiv Limits:

```
struct cl_limits {
    int maxreclevel; /* maximal recursion level */
    int maxfiles; /* maximal number of files to be
   * scanned within an archive
   */
    int maxratio; /* maximal compression ratio */
    short archivememlim; /* limit memory usage for bzip2 (0/1) */
    long int maxfilesize; /* files in an archive larger than
   * this value will not be scanned
   */
};
```

Das letzte Argument in der `cl_scan` Familie konfiguriert die scan engine. Es unterstützt die folgenden flags:

- **CL_RAW**
  macht nichts. Bitte verwenden Sie diese flag (als einzige), wenn Sie keine speziellen Dateien durchsuchen möchten.
- **CL_ARCHIVE**
  Diese flag schaltet das transparente Durchsuchen von Archiven ein.
- **CL_DISABLERAR**
  Deaktiviert den eingebauten RAR Entpacker, der gerne memory leaks erzeugt.
- **CL_ENCRYPTED**
  Damit werden verschlüsselte Archive als verseucht markiert (Encrypted.Zip, Encrypted.RAR).
- **CL_MAIL**
  Benötigt, um die diversen Arten von Mail zu filtern.
- **CL_OLE2**
  Aktiviert die Unterstützung von Micro$oft Office Dateien.

Alle Funktionen liefern den Wert 0 (`CL_CLEAN`) zurück, wenn die Datei ”sauber” ist. Ansonsten wird `CL_VIRUS` zurückgegeben, wenn ein Virus erkannt wurde bzw. ein andere Wert im Falle eines Fehlers.

```
    struct cl_limits limits;
    const char *virname;

memset(&limits, 0, sizeof(struct cl_limits));
/* maximal number of files in archive */;
limits.maxfiles = 1000
/* maximal archived file size == 10 MB */
limits.maxfilesize = 10 * 1048576;
/* maximal recursion level */
limits.maxreclevel = 5;
/* maximal compression ratio */
limits.maxratio = 200;
/* disable memory limit for bzip2 scanner */
limits.archivememlim = 0;

if((ret = cl_scanfile("/home/zolw/test", &virname, NULL, root,
&limits, CL_ARCHIVE | CL_MAIL | CL_OLE2)) == CL_VIRUS) {
    printf("Detected %s virus.\n", virname);
} else {
    printf("No virus detected.\n");
    if(ret != CL_CLEAN)
        printf("Error: %s\n", cl_strerror(ret));
}
```

Geben Sie den trie frei, wenn Sie ihn nicht länger benötigen:

```
void cl_freetrie(struct cl_node *root);
```

Sie finden ein Beispiel für einen Scanner in den Quelldateien von *ClamAV* (/example). Alle Programme, die auf *libclamav* basieren, müssen wie folgt gelinkt werden:

```
gcc -Wall ex1.c -o ex1 -lclamav
```

Viel Spass !

6.2 Datenbank erneut laden

Eine der wichtigsten Aufgaben ist es, die Instanz der internen Datenbank auf aktuellem Stand zu halten. Sie können Änderungen an dieser mittels der Familie der `cl_stat` Funkionen verfolgen:

```
int cl_statinidir(const char *dirname, struct cl_stat *dbstat);
int cl_statchkdir(const struct cl_stat *dbstat);
int cl_statfree(struct cl_stat *dbstat);
```

Initialisierung:

```
    struct cl_stat dbstat;

memset(&dbstat, 0, sizeof(struct cl_stat));
cl_statinidir(dbdir, &dbstat);
```

Um zu prüfen, ob eine Veränderung stattgefunden hat, rufen Sie einfach `cl_statchkdir` auf:

```
if(cl_statchkdir(&dbstat) == 1) {
    reload_database...;
    cl_statfree(&dbstat);
    cl_statinidir(cl_retdbdir(), &dbstat);
}
```

Vergessen Sie nicht, die Struktur nach einem reload neu zu initialisieren.

6.3 Scan engine

Neue Versionen von *Clam AntiVirus* verwenden eine Variante des Aho-Corasick pattern matching Algorithmus. Dieser Algorithmus basiert auf einem Automaton des finite state Algorithmus und ist eine Generaliserung des berühmten Knuth-Morris-Pratt Algorithmus. Bitte werfen Sie einen Blick auf die Definition der Datentypen in `matcher.h`. Die Automation wird durch einen trie repräsentiert. Es ist ein rooted tree mit einigen speziellen Eigenschaften (properties) [2]. Jede node des tries repräsentiert ein bestimmtes Stadium des Automaton. In unserer Implementation werden diese nodes wie folgt festgelegt:

```
struct cl_node {
    short int islast;
    struct cli_patt *list;
    int maxpatlen;
    struct node *next[NUM_CHILDS], *trans[NUM_CHILDS], *fail;
};
```

[Fortsetzung folgt...]

6.4 CVD Format

Beim Format *CVD (ClamAV Virus Database)* handelt es sich um ein digital signiertes TAR Archiv, das eine oder mehrere Datenbank(en) enthält. Sie finden einige praktische Informationen im ASCII Header der Datei. Dieser ist ein 512 Byte langer string mit den folgenden Komma-separierten Feldern:

```
ClamAV-VDB:build time:version:number of signatures:functionality
level required:MD5 checksum:digital signature:builder name
```

Den Header können Sie somit einfach in Ihren Skripten erfassen oder mittels `sigtool --info` anzeigen lassen. Es gibt zwei CVD Datenbanken in *ClamAV*: *main.cvd* und *daily.cvd* für die täglichen Updates. Verwenden Sie *sigtool*, um ein CVD zu entpacken (`--unpack`) und (`--list-sigs`), um eine Liste der Viren zu erhalten.

7 Credits

7.1 Kontributoren

Die folgenden Personen haben zu unserem Projekt beigetragen (mit patches, Fehlerberichten, technischer Unterstützung, Dokumentation, guten Ideen,...):

- Sergey Y. Afonin <asy*kraft-s.ru>
- Robert Allerstorfer <roal*anet.at>
- Claudio Alonso <cfalonso*yahoo.com>
- Kamil Andrusz <wizz*mniam.net>
- Jean-Edouard Babin <Jeb*jeb.com.fr>
- Marc Baudoin <babafou*babafou.eu.org>
- Rolf Eike Beer <eike*mail.math.uni-mannheim.de>
- Rene Bellora <rbellora*tecnoaccion.com.ar>
- Hilko Bengen <bengen*vdst-ka.inka.de>
- Patrick Bihan-Faou <patrick*mindstep.com>
- Oliver Brandmueller <ob*e-Gitt.NET>

- Igor Brezac <igor*ipass.net>
- Brian Bruns <bruns*2mbit.com>
- Len Budney <lbudney*pobox.com>
- Matt Butt <mattb*cre8tiv.com>
- Eric I. Lopez Carreon <elopezc*technitrade.com>
- Andrey Cherezov <andrey*cherezov.koenig.su>
- Alex Cherney <alex*cher.id.au>
- Tom G. Christensen <tgc*statsbiblioteket.dk>
- Nicholas Chua <nicholas*ncmbox.net>
- Chris Conn <cconn*abacom.com>
- Christoph Cordes <ib*precompiled.de>
- Eugene Crosser <crosser*rol.ru>
- Damien Curtain <damien*pagefault.org>
- Krisztian Czako <slapic*linux.co.hu>
- Diego d'Ambra <da*softcom.dk>
- Michael Dankov <misha*btrc.ru>
- Maxim Dounin <mdounin*rambler-co.ru>
- Alejandro Dubrovsky <s328940*student.uq.edu.au>
- Magnus Ekdahl <magnus*debian.org>
- Jens Elkner <elkner*linofee.org>
- Fred van Engen <fred*wooha.org>
- Jason Englander <jason*englanders.cc>
- Oden Eriksson <oden.eriksson*kvikkjokk.net>
- Andy Fiddaman <af*jeamland.org>
- Edison Figueira Junior <edison*brc.com.br>

- David Ford <david+cert*blue-labs.org>
- Brian J. France <list*firehawksystems.com>
- Free Oscar <freeoscar*wp.pl>
- Martin Fuxa <yeti*email.cz>
- Piotr Gackiewicz <gacek*intertele.pl>
- Jeremy Garcia <jeremy*linuxquestions.org>
- Michel Gaudet <Michel.Gaudet*ehess.fr>
- Philippe Gay <ph.gay*free.fr>
- Nick Gazaloff <nick*sbin.org>
- Luca 'NERvOus' Gibelli <nervous*nervous.it>
- Wieslaw Glod <wkg*x2.pl>
- Stephen Gran <steve*lobefin.net>
- Matthew A. Grant <grantma*anathoth.gen.nz>
- Hrvoje Habjanic <hrvoje.habjanic*zg.hinet.hr>
- Michal Hajduczenia <michalis*mat.uni.torun.pl>
- Jean-Christophe Heger <jcheger*acytec.com>
- Anders Herbjornsen <andersh*gar.no>
- Paul Hoadley <paulh*logixsquad.net>
- Robert Hogan <robert*roberthogan.net>
- Przemyslaw Holowczyc <doozer*skc.com.pl>
- Thomas W. Holt Jr. <twh*cohesive.net>
- James F. Hranicky <jfh*cise.ufl.edu>
- Douglas J Hunley <doug*hunley.homeip.net>
- Kurt Huwig <kurt*iku-netz.de>
- Andy Igoshin <ai*vsu.ru>

- Jay <sysop-clamav*coronastreet.net>
- Stephane Jeannenot <stephane.jeannenot*wanadoo.fr>
- Dave Jones <dave*kalkbay.co.za>
- Jesper Juhl <juhl*dif.dk>
- Alex Kah <alex*narfonix.com>
- Stefan Kaltenbrunner <mm-mailinglist*madness.at>
- Kazuhiko <kazuhiko*fdiary.net>
- Tomasz Klim <tomek*euroneto.pl>
- Robbert Kouprie <robbert*exx.nl>
- Martin Kraft <martin.kraft*fal.de>
- Petr Kristof <Kristof.P*fce.vutbr.cz>
- Henk Kuipers <henk*opensourcesolutions.nl>
- Nigel Kukard <nkukard*lbsd.net>
- Dr Andrzej Kurpiel <akurpiel*mat.uni.torun.pl>
- Thomas Lamy <Thomas.Lamy*in-online.net>
- Marty Lee <marty*maui.co.uk>
- Dennis Leeuw <dleeuw*made-it.com>
- Martin Lesser <admin-debian*bettercom.de>
- Peter N Lewis <peter*stairways.com.au>
- Mike Loewen <mloewen*sturgeon.cac.psu.edu>
- David S. Madole <david*madole.net>
- Thomas Madsen <tm*softcom.dk>
- Bill Maidment <bill*maidment.com.au>
- Joe Maimon <jmaimon*ttec.com>
- Andrey V. Malyshev <amal*krasn.ru>

- Stefan Martig <sm*officeco.ch>
- Serhiy V. Matveyev <matveyev*uatele.com>
- Reinhard Max <max*suse.de>
- Brian May <bam*debian.org>
- Ken McKittrick <klmac*usadatanet.com>
- Chris van Meerendonk <cvm*castel.nl>
- Andrey J. Melnikoff <temnota*kmv.ru>
- Damian Menscher <menscher*uiuc.edu>
- Arkadiusz Miskiewicz <misiek*pld-linux.org>
- Mark Mielke <mark*mark.mielke.cc>
- Jo Mills <Jonathan.Mills*frequentis.com>
- Dustin Mollo <dustin.mollo*sonoma.edu>
- Doug Monroe <doug*planetconnect.com>
- Alex S Moore <asmoore*edge.net>
- Dirk Mueller <mueller*kde.org>
- Flinn Mueller<flinn*activeintra.net>
- Hendrik Muhs <Hendrik.Muhs*student.uni-magdeburg.de>
- Farit Nabiullin http://program.farit.ru
- Nemosoft Unv. <nemosoft*smcc.demon.nl>
- Wojciech Noworyta <wnow*konarski.edu.pl>
- Jorgen Norgaard <jnp*anneli.dk>
- Fajar A. Nugraha <fajar*telkom.co.id>
- Joe Oaks <joe.oaks*hp.com>
- Washington Odhiambo <wash*wananchi.com>
- Masaki Ogawa <proc*mac.com>

- Phil Oleson <oz*nixil.net>
- Martijn van Oosterhout <kleptog*svana.org>
- OpenAntiVirus Team (http://www.OpenAntiVirus.org)
- Tomasz Papszun <tomek*lodz.tpsa.pl>
- Eric Parsonage <eric*eparsonage.com>
- Oliver Paukstadt <pstadt*stud.fh-heilbronn.de>
- Christian Pelissier <Christian.Pelissier*onera.fr>
- Rudolph Pereira <r.pereira*isu.usyd.edu.au>
- Ed Phillips <ed*UDel.Edu>
- Andreas Piesk <Andreas.Piesk*heise.de>
- Alex Pleiner <pleiner*zeitform.de>
- Ant La Porte <ant*dvere.net>
- Sergei Pronin <sp*finndesign.fi>
- Thomas Quinot <thomas*cuivre.fr.eu.org>
- Ed Ravin <eravin*panix.com>
- Rupert Roesler-Schmidt <r.roesler-schmidt*uplink.at>
- David Sanchez <dsanchez*veloxia.com>
- David Santinoli <david*santinoli.com>
- Vijay Sarvepalli <vssarvep*office.uncg.edu>
- Martin Schitter
- Enrico Scholz <enrico.scholz*informatik.tu-chemnitz.de>
- Karina Schwarz <k.schwarz*uplink.at>
- Scsi <scsi*softland.ru>
- Dr Matthew J Seaman <m.seaman*infracaninophile.co.uk>
- Hector M. Rulot Segovia <Hector.Rulot*uv.es>

- Sergey <a_s_y*sama.ru>
- Kevin Spicer <kevin*kevinspicer.co.uk>
- Ole Stanstrup <ole*stanstrup.dk>
- Adam Stein <adam*scan.mc.xerox.com>
- Steve <steveb*webtribe.net>
- Richard Stevenson <richard*endace.com>
- Matt Sullivan <matt*sullivan.gen.nz>
- Dr Zbigniew Szewczak <zssz*mat.uni.torun.pl>
- Joe Talbott <josepht*cstone.net>
- Gernot Tenchio <g.tenchio*telco-tech.de>
- Masahiro Teramoto <markun*onohara.to>
- Ryan Thompson <clamav*sasknow.com>
- Michael L. Torrie <torriem*chem.byu.edu>
- Trashware <trashware*gmx.net>
- Daniel Mario Vega <dv5a*dc.uba.ar>
- Laurent Wacrenier <lwa*teaser.fr>
- Charlie Watts <cewatts*brainstorminternet.net>
- Nicklaus Wicker <n.wicker*cnk-networks.de>
- David Woakes <david*mitredata.co.uk>
- Troy Wollenslegel <troy*intranet.org>
- Dale Woolridge <dwoolridge*drh.net>
- Takumi Yamane <yamtak*b-session.com>
- Youza Youzovic <youza*post.cz>
- Leonid Zeitlin <lz*europe.com>
- ZMan Z. <x86zman*go-a-way.dyndns.org>
- Andoni Zubimendi <andoni*lpsat.net>

7.2 Spender

Finanzielle Unterstützung haben wir erhalten von (dies ist keine vollständige Liste, da wir nur Personen anführen, die mit der Veröffentlich einverstanden sind):

- ActiveIntra.net Inc. (`http://www.activeintra.net`)
- Anonymous donor from Colorado, US
- AWD Online (`http://www.awdonline.com`)
- Norman E. Brake, Jr.
- Cheahch from Singapore
- Joe Cooper
- Steve Donegan (`http://www.donegan.org`)
- Dynamic Network Services, Inc (`http://www.dyndns.org`)
- Electric Embers
- Epublica
- Bernhard Erdmann
- David Eriksson (`http://www.2good.nu`)
- Explido Software USA Inc. (`http://www.explido.us`)
- David Farrick
- Petr Ferschmann (`http://petr.ferschmann.cz/`)
- Andries Filmer (`http://www.netexpo.nl`)
- Jack Fung
- Jeremy Garcia (`http://www.linuxquestions.org`)
- GBC Internet Service Center GmbH (`http://www.gbc.net`)
- Todd Goodman
- Bill Gradwohl (`http://www.ycc.com`)
- Grain-of-Salt Consulting

- Invisik Corporation (http://www.invisik.com)
- Keith (http://www.textpad.com)
- Brad Koehn
- Logic Partners Inc. (http://www.logicpartners.com)
- Luke Reeves (http://www.neuro-tech.net)
- Midcoast Internet Solutions
- Mimecast (http://www.mimecast.com)
- Paul Morgan
- Michael Nolan (http://www.michaelnolan.co.uk)
- Oneworkspace.com (http://www.oneworkspace.com)
- Origin Solutions (http://www.originsolutions.com.au)
- outermedia GmbH (http://www.outermedia.de)
- Roaring Penguin Software Inc. (http://www.roaringpenguin.com/)
- Seattle Server (http://www.seattleserver.com)
- Solutions In A Box (http://www.siab.com.au)
- Stephane Rault
- Fernando Augusto Medeiros Silva (http://www.linuxplace.com.br)
- Brad Tarver
- Jeremy Vanderburg (http://www.jeremytech.com)
- Webzone Srl (http://www.webzone.it)
- Nicklaus Wicker

8 Autoren

8.1 Pflege der Viren-Datenbanken

Die Viren-Datenbanken sind das Herzstück jeder Antivirus-Software. Die folgenden Personen kümmern sich darum, dass *ClamAV*'s Herz in einer ausgezeichneten Verfassung bleibt:

- aCaB <acab*clamav.net>
- Christoph Cordes <ccordes*clamav.net>
- Diego D'Ambra <diego*clamav.net>
- Jason Englander <jason*clamav.net>
- Tomasz Kojm <tkojm*clamav.net>
- Trog <trog*clamav.net> (macro viruses)
- Denis De Messemacker <ddm*clamav.net>
- Tomasz Papszun <tomek*clamav.net>

Unsere Datenbank beinhaltet die Viren-Datenbank von OpenAntiVirus.org, die derzeit etwa 5.000 Signaturen umfasst.

8.2 Netzwerk Management

Dank Luca 'NERvOus' Gibelli <nervous*clamav.net> können Sie unser Datenbank von allen mirror Servern, die in 2.10 aufgelistet sind, beziehen. Luca ist ebenfalls für unsere Hauptseite www.clamav.net, Mailinglisten und den Mechanismus zum Eintragen neuer Viren zuständig.

8.3 Grafiken

Die Autoren des netten *ClamAV* Logos (auf der Titelseite) sind Mia Kalenius und Sergei Pronin <sp*finndesign.fi>.

8.4 Haupt-Entwickler

Nigel Horne <njh*clamav.net> ist ein sehr aktiver Entwickler des *ClamAV* Projekts und verantwortlich für den mbox code, sowie clamav-milter. Trog <trog*clamav.net> programmiert den den OLE2 code und den neuen thread manager in clamd. Thomas Lamy ist ein grossartiger memory leak killer und code stabilizer. Tomasz Kojm <tkojm*clamav.net> managt das Projekt und behält jeden Aspekt im Auge 8-)

References

[1] Cormen, Leiserson, Rivest: *Introduction to Algorithms*, Chapter 34, MIT Press.

[2] http://www-sr.informatik.uni-tuebingen.de/~buehler/AC/AC.html: Beschreibung des Aho-Corasick Algorithmus