



Clam AntiVirus 0.70 User Manual

by Tomasz Kojm

Contents

1	Introduction	4
1.1	Features	4
1.2	Mailing lists	4
1.3	Virus submitting	5
2	Installation	5
2.1	Supported platforms	5
2.2	Binary packages - stable versions	6
2.3	Binary packages - snapshots	7
2.4	Requirements	7
2.5	New system user and group	8
2.6	Compilation	8
2.7	Configuration	9
2.8	Testing	10
2.9	freshclam: Setting up auto-updating	10
2.10	Database mirrors	11
3	Usage	13
3.1	Clam daemon	13
3.2	Clam <code>d</code> scan	14
3.3	Clamuko	15
3.4	Archives and compressed files	16
3.5	Mail files	17
3.6	Output format	17
3.7	Signature Tool	19
4	Problem solving	21
4.1	Return codes	21
5	ClamAV-certified software	22
5.1	clamav-milter	22
5.2	IVS Milter	23
5.3	smtp-vilter	23
5.4	mod_clamav	23
5.5	TrashScan	23
5.6	AMaViS - "Next Generation"	24
5.7	amavisd-new	24
5.8	Qmail-Scanner	24

5.9	Sagator	24
5.10	ClamdMail	25
5.11	BlackHole	25
5.12	MailScanner	25
5.13	MIMEDefang	25
5.14	exiscan	25
5.15	scanexi	26
5.16	Mail::ClamAV	26
5.17	OpenAntiVirus samba-vscan	26
5.18	Sylpheed Claws	26
5.19	nclamd	26
5.20	cgpav	27
5.21	j-chkmail	27
5.22	qscanq	27
5.23	clamavr	27
5.24	DansGuardian Anti-Virus Plugin	27
5.25	ClamAssassin	27
5.26	Gadoyanvirus	28
5.27	OpenProtect	28
5.28	POP3 Virus Scanner Daemon	28
5.29	mailman-clamav	28
5.30	wbmclamav	29
5.31	Scan Log Analyzer	29
5.32	mailgraph	29
5.33	INSERT	29
5.34	Local Area Security	29
6	LibClamAV	30
6.1	General API	30
6.2	Database reloading	33
6.3	Scan engine	34
6.4	CVD format	34
7	Credits	35
7.1	Contributors	35
7.2	Donors	40
8	Authors	41
8.1	Virus Database Maintainers	41
8.2	Network management	42
8.3	Graphics	42

8.4 Core developers	42
-------------------------------	----

Clam AntiVirus is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

1 Introduction

Clam AntiVirus is an anti-virus toolkit for UNIX. The main purpose of this software is integration with mail servers (attachment scanning). The package provides a flexible and scalable multi-threaded daemon, a command line scanner, and a tool for automatic updating via Internet. The programs are based on a shared library distributed with the Clam AntiVirus package, which you can use with your own software.

1.1 Features

- Licensed under the GNU General Public License, Version 2
- POSIX compliant, portable
- Very fast scanning
- On-access scanning (Linux and FreeBSD only)
- Detects over 20000 viruses, worms and trojans
- Supports archives and compressed files
- Can decompress Microsoft Office (OLE2) files
- Built-in support for RAR (2.0), Zip, Gzip, Bzip2
- Built-in protection against archive bombs
- Built-in support for Mbox, Maildir and raw mail files
- Includes a database updater with support for digital signatures

1.2 Mailing lists

There are four mailing lists available:

- **clamav-announce@lists.sf.net** - info about new versions (including debian package releases), moderated¹.
- **clamav-users@lists.sf.net** - user questions
- **clamav-devel@lists.sf.net** - development
- **clamav-virusdb@lists.sf.net** - database update announcements

¹That means the subscribers are not allowed to write into the mailing list

You can subscribe and check the mailing list archives at: <http://www.clamav.net/ml.html>

1.3 Virus submitting

If you have got a virus that is not detected by your ClamAV with latest databases, please check it with the *ClamAV Online Specimen Scanner*:

<http://www.gietl.com/test-clamav>

and then submit it on our website:

<http://www.clamav.net/cgi-bin/sendvirus.cgi>

2 Installation

2.1 Supported platforms

Clam AntiVirus is prepared for the installation on the following operating systems / architectures (tested platforms in brackets):

- GNU/Linux - all versions and platforms
- Solaris - all versions and platforms
- FreeBSD - all versions and platforms
- 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

Some features may not be available on your operating system. If you are running Clam AntiVirus on some system not listed above please let us know.

2.2 Binary packages - stable versions

- **Debian**

The package is maintained by Stephen Gran and Thomas Lamy. ClamAV has been officially included in the Debian distribution starting from the Sarge release. Run `apt-cache search clamav` to find the name of the packages available for installation. Unofficial packages for Woody and Sarge are available and they are usually more recent than official ones. Add the following lines to your `/etc/apt/sources.list`:

```
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
```

Feel free to search for clamav on <http://www.apt-get.org> too.

- **RedHat - Fedora**

The packages are maintained by Petr Kristof and available at <http://crash.fce.vutbr.cz/crash-hat/1/clamav/> Please follow the instructions at <http://crash.fce.vutbr.cz/yum-repository.html> and then run:

```
yum update clamav
or
up2date -u clamav
```

- **PLD Linux Distribution**

The RPM packages for the Polish(ed) Linux Distribution are maintained by Arkadiusz Miskiewicz (visit <http://www.pld-linux.org>).

- **Mandrake**

The RPM package for Mandrake is available on Mandrake's mirrors, see <http://www.rpmfind.net/linux/RPM/cooker/contrib/i586/CByName.html>. The latest Mandrake RPM packages are available at <ftp://ftp.neocat.org/pub/> and maintained by Bill Randle.

- **FreeBSD**

The official FreeBSD port is maintained by Masahiro Teramoto. There are two version available: `clamav` and `clamav-devel`. You can find both of them under `/usr/ports/security/`

- **OpenBSD**

The unofficial port for OpenBSD is available at: <http://www.fatbsd.com/openbsd/>

- **NetBSD**

The official port is available.

- **AIX**

The binary packages for AIX are available in AIX PDSLIB, UCLA <http://aixpdslib.seas.ucla.edu/packages/clamav.html>

- **MS Windows**

All major features of ClamAV are implemented under Win32 using the Cygwin compatibility layer. You can download a self installing package at <http://www.sosdg.org/clamav-win32/index.php>

2.3 Binary packages - snapshots

Thanks to Fajar A. Nugraha you can download daily builds (from daily snapshots) for the following operating systems:

- SPARC Solaris 8/9
- DEC OSF (built on Tru64 UNIX V5.0A)
- AIX (built on AIX Version 5.1)
- Linux i386 with glibc 2.3 (compiled on Fedora Core 1, works on RH \geq 8)
- Win32/Cygwin (compiled on XP)

Please visit <http://clamav.or.id>

2.4 Requirements

The following elements are required to compile ClamAV:

- zlib and zlib-devel packages
- gcc compiler suite (both 2.9x and 3.x are supported)

The following packages are optional but **highly recommended**:

- bzip2 and bzip2-devel library

- GNU MP 3

It's very important to install the GMP package because it allows freshclam to verify the digital signature of the virus database. If freshclam was compiled without GMP support it will display "SECURITY WARNING: NO SUPPORT FOR DIGITAL SIGNATURES" on every update. You can download GNU MP at <http://www.swox.com/gmp> A note for Solaris users: you should set the *ABI* system variable to 32 (e.g. `setenv ABI 32`) before running the GMP's configure script ².

2.5 New system user and group

If you are installing ClamAV for the first time, you have to add a new user and group to your system: ³

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

The above method works on Linux and Solaris - in case you don't have *groupadd*, *useradd* please consult the system manual. If you are installing ClamAV on a user account you may omit this step with the `--disable-clamav` option passed to the configure script:

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

It disables testing for existence of the *clamav* user and group. **clamscan still requires the unprivileged user and group to work in the superuser mode.** The password for that account should be locked in */etc/passwd* or */etc/shadow*.

2.6 Compilation

Once you have created the clamav user and group, please extract the archive:

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

Assuming you want to install the configuration files in */etc*, configure the package as follows:

²Thanks to Ed Phillips

³Cygwin note: If you don't have */etc/passwd* you don't need that.

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

Currently *gcc* is required for the compilation. Support for other compilers will be added in the near future.

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

In the last step software is installed in the `/usr/local` directory and the config file goes to `/etc`. **WARNING: Never enable the SUID or SGID bits in Clam AntiVirus binaries.**

2.7 Configuration

If you are going to use the daemon you have to configure it because it won't run with default settings:

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

This shows a location of the configuration file. The format and options of this file are fully described in the *clamav.conf(5)* manual. `clamd` configuration is very easy because the config file is well commented. Remember - you must remove the "Example" directive.

Another feature of `clamd` is on-access scanning based on the Dazuko module, available from <http://dazuko.org>. **This is not required to run clamd - furthermore, you shouldn't run Dazuko on production systems.** A special thread in `clamd` responsible for a communication with Dazuko is called "Clamuko" (it's due to the funny name of Dazuko - we don't know what Clamuko means). Clamuko is supported on Linux and FreeBSD only. To compile dazuko execute:

```
$ tar xzpvf dazuko-a.b.c.tar.gz
$ cd dazuko-a.b.c
$ make dazuko
or
$ make dazuko-smp (for smp kernels)
$ su
# insmod dazuko.o
# cp dazuko.o /lib/modules/`uname -r`/misc
# depmod -a
```

Depending on your Linux distribution you have to add a "dazuko" entry to */etc/modules* or something like:

```
modprobe dazuko
```

to some startup file in order to load dazuko at a boot time. Compilation on FreeBSD is very similar. You must also create the */dev/dazuko* device:

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

Now just configure Clamuko in *clamav.conf* and see the 3.3 section.

2.8 Testing

Try to scan recursively the source directory:

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

It should find some test viruses in the *clamav-x.yz/test* directory. The scan result is saved in the *scan.txt* log file.⁴ To test clamd: start it and use *clamdscan* (you can also connect directly to clamd and run the SCAN command):

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

The output and the logfile should be similar to those of *clamscan*.

2.9 freshclam: Setting up auto-updating

freshclam is a default database updater for Clam AntiVirus. It can work in two modes:

- interactive - from command line, verbosly
- daemon - alone, silently

⁴More info on clamscan options: **man clamscan**

When started by a superuser (by default) it drops privileges and switches to the *clamav* user. *freshclam* uses the database.clamav.net round-robin DNS which automatically selects a database mirror2.10. *freshclam* is an advanced tool: supports proxy servers (with authentication), digital signature verification and various error scenarios. **Quick test: run *freshclam* (as superuser) with no parameters and check the output.** If everything is OK you may create the log file in */var/log* (owned by *clamav* or another user *freshclam* will be running as (*--user*):

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

Now you *should* edit the configuration file (usually *freshclam.conf*) and configure the *UpdateLogFile* directive to point the created log file (it's highly recommended). Optionally you may force the log file path with *-l*. Finally, to run *freshclam* in the daemon mode execute:

```
# freshclam -d
```

The other method is to use the *cron* daemon. You have to add the following line to the crontab of the **root** or **clamav** users:

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

to check for a new database every hour. **N should be a number between 1 and 59 of your choice. Please don't choose any multiple of 10, because there are already too many servers using those time slots.** Proxy settings are only configurable via the configuration file (so you can setup proper permissions to protect your proxy password):

```
HTTPProxyServer myproxyserver.com
HTTPProxyPort 1234
HTTPProxyUsername myusername
HTTPProxyPassword mypass
```

2.10 Database mirrors

freshclam downloads the database from <http://database.clamav.net>. This is a round robin record that tries to equally balance the traffic between all the database mirrors:

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.nettron.co.za	160.124.112.17	South Africa	Ryan Zwankhuizen <info*nettron.co.za>
clamav.nchost.net	203.208.228.153	Singapore	Nicholas Chua <nicholas*ncmbox.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>
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>

Mirror	IP	Location	Administrator
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.iml.net	65.77.42.207	Florida, US	Dmitri Pavlenkov <dmitri*iml.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>

There is a *DatabaseMirror* directive in the config file, which specifies the main database server freshclam will attempt to connect to (up to *MaxAttempts* times) in order to check or update the database. If there are more *DatabaseMirror* lines specified it will switch automatically to the next one if a connection with the previous mirror failed for some reason.

3 Usage

3.1 Clam daemon

clamd is a multi-threaded daemon and uses *libclamav* to scan files against viruses. It may work in one of the two following network modes, listening on a:

- Unix (local) socket
- TCP socket

The daemon is fully configurable via the *clamav.conf* file. You will find a description for every directive in the *clamav.conf(5)* manual. *clamd* recognizes the following commands:

- **PING**
Check a daemon state (should reply with "PONG").
- **VERSION**
Print version information.
- **RELOAD**
Reload databases.
- **SHUTDOWN**
Perform a clean exit.
- **SCAN file/directory** Scan a file or directory (recursively) with archive support enabled. A full path is required.
- **RAWSCAN file/directory** Scan a file or directory (recursively) with archive support disabled. A full path is required.
- **CONTSCAN file/directory** Scan a file or directory (recursively) with archive support enabled and don't stop even if virus is found.
- **STREAM** Scan stream - clamd will return a new port number you should connect to and send a data to scan. *The protocol is obsolete and there will be a new version soon (however this one will still be supported).*
- **SESSION, END** Start/end a clamd session - you can do multiple commands per TCP session (WARNING: due to the clamd implementation the **RELOAD** command will break the session).

Clamd reacts on the three special signals:

- **SIGTERM** - perform a clean exit
- **SIGHUP** - reopen the log file
- **SIGUSR2** - reload the database

3.2 Clamscan

clamscan is a simple clamd client. In many cases you can use it as a clamscan replacement but you must remember that:

- it only depends on clamd

- although accepts the same command line options as clamscan most of them are ignored, for example: it accepts `--mbox` but the option has no effect - you must enable the `ScanMail` option in `clamd` to be able to scan mail files
- scanned files must be accessible for `clamd`
- it can't use external unpackers

3.3 Clamuko

Clamuko is a special thread in *clamd* that performs on-access scanning under Linux and FreeBSD. It is implemented as a thread in `clamd` and cannot work as a `clamd` client because of the Dazuko implementation. There are however some benefits of the current implementation - clamuko is sharing the internal virus database with `clamd` and it's updated with the `RELOAD` command. **You must follow some important rules when using clamuko:**

- Always stop the daemon cleanly - using the `QUIT` command or the `SIGTERM` signal. In other case you can lose your access to protected files until the system is restarted.
- Never protect a directory your mail-scanner software uses for attachment unpacking. Access to all infected files will be automatically blocked and the scanner (even `clamd`) won't be able to detect a virus. **All infected mails will be delivered.**

You need to enable clamuko in *clamav.conf*. To protect the `/home` directory enable:

```
ClamukoIncludePath /home
```

To protect the whole system:

```
ClamukoIncludePath /  
ClamukoExcludePath /proc  
ClamukoExcludePath /temporary/dir/of/your/mail/scanning/software
```

You can use clamuko to protect files on Samba/Netatalk (but far more better and safe idea is to use the **samba-vscan** module 5.17. NFS is not supported (Dazuko doesn't intercept NFS access calls). Yet another idea - you may build a database that contains signatures for popular exploits and setup `clamd` to protect your server from script-kiddies.

3.4 Archives and compressed files

All ClamAV scanners depend on LibClamAV. It has a built-in support for the following formats:

- Zip
- Gzip
- Bzip2
- RAR (2.0 only)

Archive types are determined by magic number tests.⁵ You need the zlib library for the Zip/Gzip support. Zip archives are accessed with the zziplib library by Guido Draheim and Tomi Ollila. RAR support is based on the Unique RAR File Library by Christian Scheurer and Johannes Winkelmann. Both of them are included and slightly modified in the clamav sources. Unrarlib supports RAR 2.0 archives only and according to Christian the new format (introduced in WinRAR 3.0) will never be supported (however clamscan can scan WinRAR 3.0 archives, see below). Due to security reasons clamd only scans archives supported by libclamav and can't use external programs. Clamscan is more clever and can switch to the external unpacker when the built-in decompressor fails:

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

```
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: You can force clamscan to list all infected files in archive using `--disable-archive` (it disables the built-in transparent decompressors) and `--unzip --unrar...`

If the scanner runs on a superuser level unpackers are executed with clamav privileges what makes the process far more secure. It also assures the clamav user has read access to all files. **You must enable recursive scanning with the -r option (--recursive)**

⁵It works similarly to the well known file(1) command.

in order to scan a whole content of an archive (including subdirectories), this option is also (usually) required to scan nested archive. External unpackers supported:

-unzip: Usually you don't need this option because Zip format is supported by libclamav. However it may be useful if libclamav fails to unzip some file. clamscan was tested with *UnZip 5.41 of 16 April 2000, by Info-ZIP*.

-unrar: Tested with *UNRAR 3.00 freeware*.

-unace: It uses an options supported by *UNACE v1.2 public version*, not tested, but should work.

-arj: Tested with *arj 3.10b*.

-zoo: Tested with *zoo 2.1*.

-lha: Tested with *LHa for Unix V 1.14e*.

-jar: clamscan uses *unzip* for .jar files. Tested with *UnZip 5.41 of 16 April 2000, by Info-ZIP*.

-tar: This option enables support for non-compressed archives. Tested with *GNU tar 1.13.17*.

-deb: This option enables support for debian binary packages. Tested with *GNU ar 2.12.90.0.14*. Implies `-tgz`, but doesn't conflict with `-tgz=FULLPATH`

-tgz: This option supports .tar.gz and .tgz files. You need *GNU tar*, on non-Linux system you probably have it installed as *gtar* and if it can be found in *\$PATH* please use `-tgz=gtar` to tell clamscan to use *gtar* instead of *tar*. Otherwise please supply a full path with `-tgz`

3.5 Mail files

Support for mail files is disabled by default. To enable it use the `--mbox` option in clamscan and uncomment the `ScanMail` directive in `clamav.conf` (for `clamd`). All popular mail formats (Mbox, Maildir, ...) are supported. Mail support is still under development and may cause stability problems - if you encounter them please send the problematic samples directly to Nigel Horne <njh@clamav.net>. You don't need this option if you are already using some wrapper such as AMaViS because it takes MIME decoding on its shoulders.

3.6 Output format

clamd uses a clamscan compatible output format:

```
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.
```

In the **SCAN** mode it closes the connection when first virus is found. In the case of archives the output is exactly the same as with normal files because archive support is transparent:

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

CONTSCAN displays all infected files found. Error messages are printed in the following format:

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

and they can be easily parsed.

clamscan writes all messages to **stderr** (only help is written to **stdout** by default). You may want to redirect it to **stdout** - this is handled with `--stdout`. An example of the clamscan output is:

```
/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
```

When a virus is found its name is printed between the `filename:` and `FOUND` strings. If a virus is found in an archive that has been extracted with an external unpacker it's noticed with `Infected Archive`. "Infected Archives" are not counted as infected files - only files within them are. Notice the difference with built-in unarchiver - extraction process is realized transparently by `libclamav` and `clamscan` doesn't know which concrete file is infected - just marks whole archives as infected.

3.7 Signature Tool

sigtool automates signature creation. If you have got an infected file not recognized by ClamAV and there is another anti-virus scanner working in a console that detects the virus you can try to generate the signature automatically. *Sigtool is only partially useful because it only detects the last part of a real signature. It will fail for multipart signatures (and especially for polymorphic viruses).* Example usage: create a random file (with any content) and insert the `test/test1` file content into it. We will use *clamscan* to generate the signature. Remember this is only an example - in a real life you don't need such tricks - just an infected file. Scan it with `clamscan --stdout testfile` - the output should be:

```
testfile: ClamAV-Test-Signature FOUND
```

```
----- SCAN SUMMARY -----  
Known viruses: 20673  
Scanned directories: 0  
Scanned files: 1  
Data scanned: 0.95 MB  
Infected files: 1  
I/O buffer size: 131072 bytes  
Time: 0.245 sec (0 m 0 s)
```

The unique string in this output is "ClamAV-Test-Signature" so run *sigtool* with the following arguments:

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

The program will concatenate arguments for `-c` (`--command`) and `-f` (`--file`) that's why the scanner's options must be given in the proper order. At the end it will generate a file called *testfile.sig*, which should be 100 bytes in size (in our example). It contains the proper signature.

```
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.
```

To make the generated signature complete you only to add the `VirusName=` string at the beginning of the hexadecimal signature in `testfile.sig`.

TIP: ClamAV scanners read all .db files in the database directory. You can create your own database files (e.g. local.db) and they won't be modified by freshclam !

4 Problem solving

4.1 Return codes

Return codes are very useful in system scripts. You can check the return code from `clamscan` by running:

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

Here is the full list of return codes for `clamscan`:

- 0:** No virus was found.
- 1:** Virus(es) detected.
- 40:** Unknown option passed to `clamscan`. Please check `clamscan --help` or manual page for available options.
- 50:** Virus database initialization error. Probably it doesn't exist at the default location or it's malformed (e.g. broken digital signature)
- 52:** Not supported file type - `clamscan` only supports regular files, directories and symlinks.
- 53:** Can't open directory.
- 54:** Can't open file.⁶
- 55:** I/O error during read. ⁶
- 56:** Can't stat input file or directory. File (or directory) you want to scan doesn't exist.
- 57:** Can't get absolute pathname of current working directory. Your current pathname is longer than 200 characters. This is bad and you may need to recompile ClamAV to fix it.
- 58:** I/O error. Please check the filesystem.
- 59:** Can't get information about current user (running `clamscan`).
- 60:** Can't get information about user `clamav`. User `clamav` (default unprivileged user)

⁶Only in a one-file mode (in recursive mode those errors are ignored).

doesn't exist in /etc/passwd.

61: Can't fork. Can't create new process, please check your system limits.

63: Can't create temporary file or directory. Please check /tmp permissions or use `-tempdir`

64: Can't write to temporary directory. Please specify another one.

70: Can't allocate and clear memory. This is a critical error, please check your system.

71: Can't allocate memory. Look above.

5 ClamAV-certified software

There are many projects that support ClamAV. Here is a list of software that was tested and is known to work well.

5.1 clamav-milter

Homepage: part of the ClamAV package

Supports: clamd

Nigel Horne's clamav-milter is a very fast email scanner designed for sendmail. It's written entirely in C and uses ClamAV's internal mail scanner (also written by Nigel).

Installation:

You need libmilter development files. Configure ClamAV with

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

and recompile. The program will be installed in /usr/local/sbin/clamav-milter. The following instructions were adopted from Nigel's INSTALL file: add to /etc/mail/sendmail.mc:

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

Check entries in clamav.conf of the form:

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

Start clamav-milter:

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

and restart sendmail.

5.2 IVS Milter

Homepage: <http://ivs-milter.lbsd.net>

Supports: clamd

IVS Milter is a virus and spam scanning milter. The name stands for Industrial Virus + Spam milter. It's designed to be used by anything from home users to large ISP's.

5.3 smtp-vilter

Homepage: <http://www.etc.msys.ch/software/smtp-vilter>

Supports: clamd

smtp-vilter is a high performance content filter for sendmail using the milter API. The software scans e-mail messages for viruses and drops or marks infected messages. ClamAV is the default scanner backend.

5.4 mod_clamav

Homepage: http://software.othello.ch/mod_clamav

Supports: libclamav, clamd

mod_clamav is an Apache virus scanning filter. It was written and is currently maintained by Andreas Muller. The project is very well documented and the installation is quite easy.

5.5 TrashScan

Homepage: clamav-sources/support/trashscan

Supports: clamscan

This is a procmail based scanner from Trashware and it's extremely easy to setup, however this is for single users only and not as efficient as MTA based scanners.

5.6 AMaViS - "Next Generation"

Homepage: <http://sourceforge.net/projects/amavis>

Supports: clamscan

AMaViS-ng is a rewritten, more modular version of amavis-perl/amavisd, developed by Hilko Bengen. Home site:

Installation:

Please download the newest version (at least 0.1.4). After installation (which is quite easy), please uncomment the following line in amavis.conf:

```
virus-scanner = CLAM
```

and eventually change the path to clamscan in the [CLAM] section:

```
[CLAM]
```

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

5.7 amavisd-new

Homepage: <http://www.ijs.si/software/amavisd>

Supports: clamd, clamscan

amavisd-new is a rewritten version of amavis maintained by Mark Martinec.

Installation:

clamscan is enabled automatically if clamscan binary is found at amavisd-new startup time. clamd is activated by uncommenting its entry in the @av_scanners list, file /etc/amavisd.conf.

5.8 Qmail-Scanner

Homepage: <http://qmail-scanner.sf.net>

Supports: clamscan

Please increase the softlimit value if you are going to use it with clamscan.

5.9 Sagator

Homepage: <http://www.salstar.sk/sagator>

Supports: clamscan, clamd, libclamav

This program is an email antivirus/antispam gateway. It is an interface to the postfix

(or any other smtpd), which runs antivirus and/or spamchecker. Its modular architecture can use any combination of antivirus/spamchecker according to configuration.

5.10 ClamdMail

Homepage: <http://clamdmal.sf.net>

Supports: clamd

A mail processing client for ClamAV. Small, fast and easy to install.

5.11 BlackHole

Homepage: <http://iland.net/~ckennedy/blackhole.shtml>

Supports: clamscan, clamd

BlackHole is an advanced spam / virus filter for Qmail, Postfix, Sendmail, Exim and Courier written by Chris Kennedy. This tool is for advanced administrators (installation is hard).

5.12 MailScanner

Homepage: <http://www.mailscanner.info>

Supports: clamscan

MailScanner scans all e-mail for viruses, spam and attacks against security vulnerabilities. It is not tied to any particular virus scanner, but can be used with any combination of 14 different virus scanners, allowing sites to choose the "best of breed" virus scanner.

5.13 MIMEDefang

Homepage: <http://www.roaringpenguin.com/mimedefang>

Supports: clamscan, clamd

This is an efficient mail scanner for Sendmail/milter.

5.14 exiscan

Homepage: <http://duncanthrax.net/exiscan>

Supports: clamscan, clamd

exiscan is a patch against exim version 4, providing support for content scanning in email messages received by exim. Four different scanning facilities are supported: antivirus, antispam, regular expressions, and file extensions.

5.15 scanexi

Homepage: <http://wl.231.telia.com/~u23107873/scanexi.html>

Supports: clamscan, clamd

scanexi is a plugin for exim version 4.14 with dlopen patch, providing support for content scanning in email messages received by exim.

5.16 Mail::ClamAV

Homepage: <http://cpan.gossamer-threads.com/modules/by-authors/id/S/SA/SABECK/>

Supports: libclamav

Perl extension for the libclamav library.

5.17 OpenAntiVirus samba-vscan

Homepage: <http://www.openantivirus.org/projects.php#samba-vscan>

Supports: 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.

5.18 Sylpheed Claws

Homepage: <http://claws.sylpheed.org>

Supports: libclamav

Sylpheed Claws is a bleeding edge branch of Sylpheed, a light weight mail user agent for UNIX. It can scan attachments in mail received from POP, IMAP or a local account and optionally delete the mail or save it to a designated folder.

5.19 nclamd

Homepage: <http://www.kyzo.com/nclamd/>

Supports: libclamav

nclamd, nclamav-milter and nclamdscan are rewritten versions of the original tools and use processes instead of threads and ripMIME instead of the clamav built-in MIME decoder.

5.20 cgpav

Homepage: <http://program.farit.ru>

Supports: clamd

This is a fast (written in C) CommuniGate Pro anti-virus plugin with support for clamd.

5.21 j-chkmail

Homepage: <http://j-chkmail.ensmp.fr>

Supports: libclamav, clamd

j-chkmail is a fast (written in C) filter for sendmail. It does spam and dangerous content (virus) filtering with help of ClamAV. The program supports many modes of monitoring and run time controlling and was designed to work on highly loaded servers. It's an open source software available for free to registered users (for non-commercial usage).

5.22 qscanq

Homepage: <http://budney.homeunix.net:8080/users/budney/software/qscanq/index.html>

Supports: clamscan

qscanq replaces qmail-queue. It initiates a scan (using clamscan or clamdscan) on an incoming email, and returns the exit status of the scanner or of qmail-queue to the caller.

5.23 clamavr

Homepage: <http://raa.ruby-lang.org/list.rhtml?name=clamavr>

Supports: libclamav

Ruby binding for ClamAV.

5.24 DansGuardian Anti-Virus Plugin

Homepage: <http://www.pcxperience.org/dgvirus/>

Supports: clamscan

DG AVP is a GPL addon that takes the Virus Scanning capabilities of The MailScanner and integrates them into the content filtering web proxy DansGuardian.

5.25 ClamAssassin

Homepage: <http://drivel.com/clamassassin/>

Supports: clamscan

clamassassin is a simple script for virus scanning with clamscan which works similarly to spamassassin. It's designed for integration with procmail.

5.26 Gadoyanvirus

Homepage: <http://oss.mdamt.net/gadoyanvirus/>

Supports: libclamav

gadoyanvirus is a (yet another) virus stopper for qmail. It replaces the original qmail-queue program. It scans incoming messages using the ClamAV anti-virus library. Suspected message will be quarantined and (optionally) a notification message will be sent to the recipients. By default, gadoyanvirus needs QMAILQUEUE patched qmail installation.

5.27 OpenProtect

Homepage: <http://opencompt.com/>

Supports: ClamAV via MailScanner

OpenProtect is a server side e-mail protection solution consisting of MailScanner, Spamassassin, ClamAV with support for Sendmail, Postfix, Exim and qmail. It also consists of a fully automatic installer and uninstaller, which configures everything automatically including setting up perl modules and virus scanner settings.

5.28 POP3 Virus Scanner Daemon

Homepage: <http://p3scan.sourceforge.net/>

Supports: clamscan

This is a full-transparent proxy-server for POP3-Clients. It runs on a Linux box with iptables (for port re-direction). It can be used to provide POP3 email scanning from the internet, to any internal network and is ideal for helping to protect your Other OS LAN from harm, especially when used in conjunction with a firewall and other Internet Proxy servers.

5.29 mailman-clamav

Homepage: <http://www.tummy.com/Software/mailman-clamav>

Supports: clamd

This module includes a Mailman handler for scanning incoming messages through ClamAV. The handler allows Mailman to be configured to hold or discard messages which contain viruses. Particularly useful is the discard option, which prevents list administrators from having to manually deal with viruses.

5.30 wbmclamav

Homepage: <http://wbmclamav.labs.libre-entreprise.org/>

Supports: ClamAV

wbmclamav is a webmin module to manage Clam AntiVirus, written by Emmanuel Saracco.

5.31 Scan Log Analyzer

Homepage: <http://pandaemail.sourceforge.net/av-tools/>

Supports: ClamAV

Scan analyzer allows you to plot and view graphical representation of log data from virus logs of RAV, ClamAV and Vexira.

5.32 mailgraph

Homepage: <http://people.ee.ethz.ch/~dws/software/mailgraph/>

Supports: clamd

mailgraph is a very simple mail statistics RRDtool frontend for Postfix that produces daily, weekly, monthly and yearly graphs of received/sent and bounced/rejected mail (SMTP traffic).

5.33 INSERT

Homepage: http://www.inside-security.de/INSERT_en.html

Supports: ClamAV

INSERT (the Inside Security Rescue Toolkit) aims to be a multi-functional, multi-purpose disaster recovery and network analysis system. It boots from a credit card-sized CD-ROM and is basically a stripped-down version of Knoppix. It features good hardware detection, fluxbox, emelfm, links-hacked, ssh, tcpdump, nmap, chntpwd, and much more. It provides full read-write support for NTFS partitions (using captive), and the ClamAV virus scanner (including the signature database).

5.34 Local Area Security

Homepage: <http://www.localareasecurity.com/>

Supports: ClamAV

Local Area Security Linux is a Live CD distribution with a strong emphasis on security tools and small footprint. It can be used to run ClamAV from a CDROM.

6 LibClamAV

libclamav may be used to add a virus protection to your software. The library is thread-safe and can transparently recognize and scan archives, mail files and MS Office document files.

6.1 General API

Every program based on libclamav must include the `clamav.h` header file:

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

A first step is to initialize the scanning engine. There are three functions available:

```
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()` loads a particular database, `cl_loaddbdir()` loads all *.cvd* (and older *.db*, *.db2*) databases from a directory `dirname`. `cl_retdbdir()` returns a hardcoded database directory path. Initial internal database (Aho-Corasick tree, trie; see 6.3) will be saved under `root` and a number of signatures loaded will be **added**⁷ to `virnum`. Pointer to the trie must initially point to `NULL`. If you don't care about number of signatures pass `NULL` as a third argument. `cl_loaddb` functions return 0 on success and an other value on failure.

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

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

There's an elegant way to print libclamav's error codes:

⁷Remember to initialize the virus counter variable with 0.

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

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

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

When database is loaded you must build the final trie with:

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

In our example:

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

Now you can scan a buffer, a descriptor or a file with:

```
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 the functions save a virus name address under `virname` pointer. It points to a name in the trie structure thus it can't be released directly. `cl_scandesc()` and `cl_scanfile()` can increase the scanned value in `CL_COUNT_PRECISION` units, they also support archive limits:

```
struct cl_limits {
    int maxrecllevel; /* 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
    */
};

```

The last argument in the `cl_scan` family configures the scan engine. It supports the following flags:

- **CL_RAW**
It does nothing. Please use it (alone) if you don't want to scan any special files.
- **CL_ARCHIVE**
This flag enables the transparent archive scanning.
- **CL_DISABLERAR**
Disables the built-in RAR unpacker which is known to cause memory leaks.
- **CL_ENCRYPTED**
Marks encrypted archives as viruses (Encrypted.Zip, Encrypted.RAR).
- **CL_MAIL**
Required to scan various types of mail files.
- **CL_OLE2**
Enables support for Microsoft Office document files.

All functions return 0 (`CL_CLEAN`) if the file is clean, `CL_VIRUS` when virus is detected and an other value on failure.

```

    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.maxrecllevel = 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));
}
```

Release the trie if you no longer need it:

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

You will find an example scanner in clamav sources (/example). All programs based on libclamav must be linked against it:

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

Enjoy !

6.2 Database reloading

The most important thing is to keep the internal instance of the database up to date. You can watch database changes with the `cl_stat` functions family:

```
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);
```

Initialization:

```
    struct cl_stat dbstat;

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

To check for a change you only need to call `cl_statchkdir`:

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

Remember to reinitialize the structure after a reload.

6.3 Scan engine

New versions of Clam AntiVirus use a mutation of the Aho-Corasick pattern matching algorithm. The algorithm is based a finite state pattern matching automaton [1] and it's a generalization of the famous Knuth-Morris-Pratt algorithm. Please take a look at the `matcher.h` for data type definitions. The automaton is represented by a trie. It is a rooted tree with some specific properties [2]. Every node of the trie represents some state of the automaton. In our implementation, the node is defined as follows:

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

[To be continued...]

6.4 CVD format

CVD (ClamAV Virus Database) is a digitally signed tarball file that contains one or more databases. You can find some useful information in the ASCII header of the file. It's a 512 bytes long string with the following colon separated fields:

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

and can be easily parsed by scripts or with `sigtool --info`. There are two CVD databases in ClamAV: *main.cvd* and *daily.cvd* for daily updates. You can use `sigtool` to unpack a CVD file (`--unpack`) and to list virus names (`--list-sigs`).

7 Credits

7.1 Contributors

The following people contributed to our project in some way (providing patches, bug reports, technical support, documentation, good ideas...):

- Kamil Andrusz <wizz*mniam.net>
- Jean-Edouard Babin <Jeb*jeb.com.fr>
- Marc Baudoin <babafou*babafou.eu.org>
- 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>
- Nicholas Chua <nicholas*ncmbox.net>
- 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>
- Alejandro Dubrovsky <s328940*student.uq.edu.au>

- Magnus Ekdahl <magnus*debian.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>
- 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 Koupprie <robbert*exx.nl>
- Martin Kraft <martin.kraft*fal.de>
- Petr Kristof <Kristof.P*fce.vutbr.cz>
- Henk Kuipers <henk*opensourceolutions.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>
- 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>
- 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>
- Wojciech Noworyta <wnow*konarski.edu.pl>
- 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>
- Rudolph Pereira <r.pereira*isu.usyd.edu.au>
- Ed Phillips <ed*UDe1.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>
- 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>
- Scsi <scsi*softland.ru>
- Hector M. Rulot Segovia <Hector.Rulot*uv.es>
- Ole Stanstrup <ole*stanstrup.dk>
- Adam Stein <adam*scan.mc.xerox.com>
- Richard Stevenson <richard*endace.com>
- Matt Sullivan <matt*sullivan.gen.nz>

- Dr Zbigniew Szewczak <zssz*mat.uni.torun.pl>
- Joe Talbott <joseph*t*ystone.net>
- Gernot Tenchio <g.tenchio*telco-tech.de>
- Masahiro Teramoto <markun*onohara.to>
- Michael L. Torrie <torriem*chem.byu.edu>
- Trashware <trashware*gmx.net>
- Daniel Mario Vega <dv5a*dc.uba.ar>
- Laurent Wacrenier <lwa*teaser.fr>
- Nicklaus Wicker <n.wicker*cnk-networks.de>
- David Woakes <david*mitredata.co.uk>
- Troy Wollenslegel <troy*intranet.org>
- Leonid Zeitlin <lz*europa.com>
- Andoni Zubimendi <andoni*lpsat.net>

7.2 Donors

We received financial support from:

- 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
- Dynamic Network Services, Inc (<http://www.dyndns.org>)
- Electric Embers
- Epublica
- David Eriksson (<http://www.2good.nu>)

- Explido Software USA Inc. (<http://www.explido.us>)
- David Farrick
- Andries Filmer (<http://www.netexpo.nl>)
- Jack Fung
- Jeremy Garcia (<http://www.linuxquestions.org>)
- Bill Gradwohl (<http://www.ycc.com>)
- Juergen Hahn
- Invisik Corporation (<http://www.invisik.com>)
- Keith (<http://www.textpad.com>)
- Brad Koehn
- Logic Partners Inc. (<http://www.logicpartners.com>)
- Michael Nolan
- Oneworkspace.com (<http://www.oneworkspace.com>)
- Origin Solutions (<http://www.originsolutions.com.au>)
- Seattle Server (<http://www.seattleserver.com>)
- Solutions In A Box (<http://www.siab.com.au>)
- Stephane Rault
- Jeremy Vanderburg (<http://www.jeremytech.com>)
- Nicklaus Wicker

8 Authors

8.1 Virus Database Maintainers

Virus database is a heart of every anti-virus software. The following people care ClamAV's heart to be in a fine condition:

- 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>
- Denis De Messemaeker <ddm*clamav.net>
- Tomasz Papszun <tomek*clamav.net>

Our database includes the virus database (about 5000 signatures) from `OpenAntiVirus.org`.

8.2 Network management

Thanks to Luca 'NERvOus' Gibelli <nervous*clamav.net> you can download our database from all the mirrors listed in 2.10. Luca is also responsible for our main site `www.clamav.net`, mailing lists, and the virus submission mechanism.

8.3 Graphics

The authors of the nice ClamAV logo (look at the title page) are Mia Kalenius and Sergei Pronin <sp*finndesign.fi>.

8.4 Core developers

Nigel Horne <njh*clamav.net> is a very active ClamAV developer responsible for the mbox code and clamav-milter. Trog <trog*clamav.net> develops the OLE2 code and the new thread manager in clamd. Thomas Lamy is a great memory leak killer and code stabilizer. Tomasz Kojm <tkojm*clamav.net> navigates the project and keeps an eye on everything 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>: Aho-Corasick algorithm description