Installation und Konfiguration von ClamAV unter CentOS 6.x

ClamAV[1] ist ein unter der GNU GPL[2] stehender Virenscanner. ClamAV beinhaltet eine Bibliothek, zur Einbindung in eigene Programme und Abläufe, dem ClamAV-Daemon und sowie eine Kommandozeilenapplikation.

Hauptsächlich wird ***ClamAV*** im Zusammenhang mit Postfix und AMaViS genutzt. Natürlich kann die Virusscan-Engine auch in Verbindiung mit dem Web-Content-Scanner Dansguardian zum Scannen des HTTP-Traffics benutzt werden.

Die Installation von ClamAV im Mailserverumfeld ist hier sowie auf dem DeskTop hier beschrieben.

Installation

Als erstes installieren wir uns die benötigten Server-Komponenten via **YUM**. Da die benötigten Pakete aus dem rpmforge-Repository kommen, wird die korrekte Einbindung dieses 3rd-party Repositories vorausgesetzt. Wie das geht, steht hier.

```
# yum install clamd clamav clamav-db -y
```

Was uns die einzelnen Pakete mitliefern, offentbart uns jeweils ein **rpm -iql**.

Paket clamd

```
# rpm -qil clamd
```

```
Name        : clamd                         Relocations: (not relocatable)
Version     : 0.97.3                             Vendor: Dag Apt Repository,
http://dag.wieers.com/apt/
Release     : 1.el6.rf                       Build Date: Mon 17 Oct 2011
07:38:53 PM CEST
Install Date: Wed 16 Nov 2011 12:52:14 PM CET      Build Host:
lisse.hasselt.wieers.com
Group       : System Environment/Daemons    Source RPM:
clamav-0.97.3-1.el6.rf.src.rpm
Size        : 602652                           License: GPL
Signature   : DSA/SHA1, Fri 21 Oct 2011 02:49:48 PM CEST, Key ID
a20e52146b8d79e6
Packager    : Dag Wieers <dag@wieers.com>
URL         : http://www.clamav.net/
Summary     : The Clam AntiVirus Daemon
Description :
```

```
The Clam AntiVirus Daemon
/etc/clamd.conf
/etc/logrotate.d/clamav
/etc/rc.d/init.d/clamd
/usr/bin/clamconf
/usr/bin/clamdscan
/usr/bin/clamdtop
/usr/sbin/clamd
/usr/share/doc/clamd-0.97.3
/usr/share/doc/clamd-0.97.3/clamd.conf
/usr/share/man/man1/clambc.1.gz
/usr/share/man/man1/clamconf.1.gz
/usr/share/man/man1/clamdscan.1.gz
/usr/share/man/man1/clamdtop.1.gz
/usr/share/man/man5/clamd.conf.5.gz
/usr/share/man/man8/clamd.8.gz
/var/clamav
/var/log/clamav
/var/log/clamav/clamd.log
/var/run/clamav
```

Paket clamav-db

```
 # rpm -qil clamav-db
```

```
Name        : clamav-db                    Relocations: (not relocatable)
Version     : 0.97.3                            Vendor: Dag Apt Repository,
http://dag.wieers.com/apt/
Release     : 1.el6.rf                      Build Date: Mon 17 Oct 2011
07:38:53 PM CEST
Install Date: Wed 16 Nov 2011 12:52:11 PM CET      Build Host:
lisse.hasselt.wieers.com
Group       : Applications/Databases        Source RPM:
clamav-0.97.3-1.el6.rf.src.rpm
Size        : 31043501                         License: GPL
Signature   : DSA/SHA1, Fri 21 Oct 2011 02:49:46 PM CEST, Key ID
a20e52146b8d79e6
Packager    : Dag Wieers <dag@wieers.com>
URL         : http://www.clamav.net/
Summary     : Virus database for clamav
Description :
The actual virus database for clamav
/etc/cron.daily/freshclam
/etc/logrotate.d/freshclam
/var/clamav
/var/clamav/daily.cvd
/var/clamav/main.cvd
/var/log/clamav
/var/log/clamav/freshclam.log
```

Paket clamav

```
 # rpm -qil clamav
```

```
Name        : clamav                        Relocations: (not relocatable)
Version     : 0.97.3                             Vendor: Dag Apt Repository,
http://dag.wieers.com/apt/
Release     : 1.el6.rf                       Build Date: Mon 17 Oct 2011
07:38:53 PM CEST
Install Date: Wed 16 Nov 2011 12:52:12 PM CET      Build Host:
lisse.hasselt.wieers.com
Group       : Applications/System           Source RPM:
clamav-0.97.3-1.el6.rf.src.rpm
Size        : 6113670                          License: GPL
Signature   : DSA/SHA1, Fri 21 Oct 2011 02:49:47 PM CEST, Key ID
a20e52146b8d79e6
Packager    : Dag Wieers <dag@wieers.com>
URL         : http://www.clamav.net/
Summary     : Anti-virus software
Description :
Clam AntiVirus is a GPL anti-virus toolkit for UNIX. The main purpose of
this software is the 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. Most
importantly, the virus database is kept up to date
/etc/freshclam.conf
/usr/bin/clambc
/usr/bin/clamscan
/usr/bin/freshclam
/usr/bin/sigtool
/usr/lib64/libclamav.so
/usr/lib64/libclamav.so.6
/usr/lib64/libclamav.so.6.1.12
/usr/lib64/libclamunrar.so
/usr/lib64/libclamunrar.so.6
/usr/lib64/libclamunrar.so.6.1.12
/usr/lib64/libclamunrar_iface.so
/usr/lib64/libclamunrar_iface.so.6
/usr/lib64/libclamunrar_iface.so.6.1.12
/usr/share/doc/clamav-0.97.3
/usr/share/doc/clamav-0.97.3/AUTHORS
/usr/share/doc/clamav-0.97.3/BUGS
/usr/share/doc/clamav-0.97.3/COPYING
/usr/share/doc/clamav-0.97.3/ChangeLog
/usr/share/doc/clamav-0.97.3/FAQ
/usr/share/doc/clamav-0.97.3/INSTALL
/usr/share/doc/clamav-0.97.3/NEWS
```

```
/usr/share/doc/clamav-0.97.3/README
/usr/share/doc/clamav-0.97.3/clamav-mirror-howto.pdf
/usr/share/doc/clamav-0.97.3/clamdoc.pdf
/usr/share/doc/clamav-0.97.3/freshclam.conf
/usr/share/doc/clamav-0.97.3/phishsigs_howto.pdf
/usr/share/doc/clamav-0.97.3/signatures.pdf
/usr/share/man/man1/clambc.1.gz
/usr/share/man/man1/clamscan.1.gz
/usr/share/man/man1/freshclam.1.gz
/usr/share/man/man1/sigtool.1.gz
/usr/share/man/man5/freshclam.conf.5.gz
```

clamav Konfiguration

/etc/clamd.conf

Die Konfigurationsdatei des ClamAV-Daemons **/etc/clamd.conf** passen wir unseren Gegebenheiten entsprechend an. Wichtig sind dabei insbesonders die drei Paramter:

- **User clamav**
- **AllowSupplementaryGroups yes**
- **LocalSocket /var/run/clamav/clamd.sock**

Mit unserem Lieblingseditor **vim** bearbeiten wir nun die Konfigurationsdatei ***/etc/clamd.conf***.

vim /etc/clamd.conf

vim /etc/clamd.conf

```
##
## Example config file for the Clam AV daemon
## Please read the clamd.conf(5) manual before editing this file.
##


# Comment or remove the line below.
#Example

# Uncomment this option to enable logging.
# LogFile must be writable for the user running daemon.
# A full path is required.
# Default: disabled
LogFile /var/log/clamav/clamd.log

# By default the log file is locked for writing - the lock protects
against
# running clamd multiple times (if want to run another clamd, please
# copy the configuration file, change the LogFile variable, and run
# the daemon with --config-file option).
```

```
# This option disables log file locking.
# Default: no
#LogFileUnlock yes

# Maximum size of the log file.
# Value of 0 disables the limit.
# You may use 'M' or 'm' for megabytes (1M = 1m = 1048576 bytes)
# and 'K' or 'k' for kilobytes (1K = 1k = 1024 bytes). To specify the
size
# in bytes just don't use modifiers.
# Default: 1M
LogFileMaxSize 0

# Log time with each message.
# Default: no
LogTime yes

# Also log clean files. Useful in debugging but drastically increases
the
# log size.
# Default: no
#LogClean yes

# Use system logger (can work together with LogFile).
# Default: no
LogSyslog yes

# Specify the type of syslog messages - please refer to 'man syslog'
# for facility names.
# Default: LOG_LOCAL6
#LogFacility LOG_MAIL

# Enable verbose logging.
# Default: no
#LogVerbose yes

# Log additional information about the infected file, such as its
# size and hash, together with the virus name.
#ExtendedDetectionInfo yes

# This option allows you to save a process identifier of the listening
# daemon (main thread).
# Default: disabled
PidFile /var/run/clamav/clamd.pid

# Optional path to the global temporary directory.
# Default: system specific (usually /tmp or /var/tmp).
TemporaryDirectory /var/tmp

# Path to the database directory.
# Default: hardcoded (depends on installation options)
```

```
DatabaseDirectory /var/clamav

# Only load the official signatures published by the ClamAV project.
# Default: no
#OfficialDatabaseOnly no

# The daemon can work in local mode, network mode or both.
# Due to security reasons we recommend the local mode.

# Path to a local socket file the daemon will listen on.
# Default: disabled (must be specified by a user)
LocalSocket /var/run/clamav/clamd.sock

# Sets the group ownership on the unix socket.
# Default: disabled (the primary group of the user running clamd)
#LocalSocketGroup virusgroup

# Sets the permissions on the unix socket to the specified mode.
# Default: disabled (socket is world accessible)
#LocalSocketMode 660

# Remove stale socket after unclean shutdown.
# Default: yes
FixStaleSocket yes

# TCP port address.
# Default: no
TCPSocket 3310

# TCP address.
# By default we bind to INADDR_ANY, probably not wise.
# Enable the following to provide some degree of protection
# from the outside world.
# Default: no
TCPAddr 127.0.0.1

# Maximum length the queue of pending connections may grow to.
# Default: 200
MaxConnectionQueueLength 30

# Clamd uses FTP-like protocol to receive data from remote clients.
# If you are using clamav-milter to balance load between remote clamd
daemons
# on firewall servers you may need to tune the options below.

# Close the connection when the data size limit is exceeded.
# The value should match your MTA's limit for a maximum attachment
size.
# Default: 25M
#StreamMaxLength 10M
```

```
# Limit port range.
# Default: 1024
#StreamMinPort 30000
# Default: 2048
#StreamMaxPort 32000

# Maximum number of threads running at the same time.
# Default: 10
MaxThreads 50

# Waiting for data from a client socket will timeout after this time
(seconds).
# Default: 120
ReadTimeout 300

# This option specifies the time (in seconds) after which clamd should
# timeout if a client doesn't provide any initial command after
connecting.
# Default: 5
#CommandReadTimeout 5

# This option specifies how long to wait (in miliseconds) if the send
buffer is full.
# Keep this value low to prevent clamd hanging
#
# Default: 500
#SendBufTimeout 200

# Maximum number of queued items (including those being processed by
MaxThreads threads)
# It is recommended to have this value at least twice MaxThreads if
possible.
# WARNING: you shouldn't increase this too much to avoid running out
of file descriptors,
# the following condition should hold:
# MaxThreads*MaxRecursion + (MaxQueue - MaxThreads) + 6< RLIMIT_NOFILE
(usual max is 1024)
#
# Default: 100
#MaxQueue 200

# Waiting for a new job will timeout after this time (seconds).
# Default: 30
#IdleTimeout 60

# Don't scan files and directories matching regex
# This directive can be used multiple times
# Default: scan all
#ExcludePath ^/proc/
#ExcludePath ^/sys/
```

```
# Maximum depth directories are scanned at.
# Default: 15
#MaxDirectoryRecursion 20

# Follow directory symlinks.
# Default: no
#FollowDirectorySymlinks yes

# Follow regular file symlinks.
# Default: no
#FollowFileSymlinks yes

# Scan files and directories on other filesystems.
# Default: yes
#CrossFilesystems yes

# Perform a database check.
# Default: 600 (10 min)
#SelfCheck 600

# Execute a command when virus is found. In the command string %v will
# be replaced with the virus name.
# Default: no
#VirusEvent /usr/local/bin/send_sms 123456789 "VIRUS ALERT: %v"

# Run as another user (clamd must be started by root for this option to
work)
# Default: don't drop privileges
User clamav

# Initialize supplementary group access (clamd must be started by
root).
# Default: no
AllowSupplementaryGroups yes

# Stop daemon when libclamav reports out of memory condition.
#ExitOnOOM yes

# Don't fork into background.
# Default: no
#Foreground yes

# Enable debug messages in libclamav.
# Default: no
#Debug yes

# Do not remove temporary files (for debug purposes).
# Default: no
#LeaveTemporaryFiles yes

# Detect Possibly Unwanted Applications.
```

```
# Default: no
#DetectPUA yes

# Exclude a specific PUA category. This directive can be used multiple
times.
# See http://www.clamav.net/support/pua for the complete list of PUA
# categories.
# Default: Load all categories (if DetectPUA is activated)
#ExcludePUA NetTool
#ExcludePUA PWTool

# Only include a specific PUA category. This directive can be used
multiple
# times.
# Default: Load all categories (if DetectPUA is activated)
#IncludePUA Spy
#IncludePUA Scanner
#IncludePUA RAT

# In some cases (eg. complex malware, exploits in graphic files, and
others),
# ClamAV uses special algorithms to provide accurate detection. This
option
# controls the algorithmic detection.
# Default: yes
#AlgorithmicDetection yes


##
## Executable files
##

# PE stands for Portable Executable - it's an executable file format
used
# in all 32 and 64-bit versions of Windows operating systems. This
option allows
# ClamAV to perform a deeper analysis of executable files and it's also
# required for decompression of popular executable packers such as UPX,
FSG,
# and Petite. If you turn off this option, the original files will
still be
# scanned, but without additional processing.
# Default: yes
ScanPE yes

# Executable and Linking Format is a standard format for UN*X
executables.
# This option allows you to control the scanning of ELF files.
# If you turn off this option, the original files will still be
scanned, but
# without additional processing.
```

```
# Default: yes
ScanELF yes

# With this option clamav will try to detect broken executables (both
PE and
# ELF) and mark them as Broken.Executable.
# Default: no
DetectBrokenExecutables yes


##
## Documents
##

# This option enables scanning of OLE2 files, such as Microsoft Office
# documents and .msi files.
# If you turn off this option, the original files will still be
scanned, but
# without additional processing.
# Default: yes
ScanOLE2 yes


# With this option enabled OLE2 files with VBA macros, which were not
# detected by signatures will be marked as
"Heuristics.OLE2.ContainsMacros".
# Default: no
#OLE2BlockMacros no

# This option enables scanning within PDF files.
# If you turn off this option, the original files will still be
scanned, but
# without decoding and additional processing.
# Default: yes
#ScanPDF yes


##
## Mail files
##

# Enable internal e-mail scanner.
# If you turn off this option, the original files will still be
scanned, but
# without parsing individual messages/attachments.
# Default: yes
ScanMail yes

# Scan RFC1341 messages split over many emails.
# You will need to periodically clean up $TemporaryDirectory/clamav-
partial directory.
```

```
# WARNING: This option may open your system to a DoS attack.
#          Never use it on loaded servers.
# Default: no
#ScanPartialMessages yes


# With this option enabled ClamAV will try to detect phishing attempts
by using
# signatures.
# Default: yes
#PhishingSignatures yes

# Scan URLs found in mails for phishing attempts using heuristics.
# Default: yes
#PhishingScanURLs yes

# Always block SSL mismatches in URLs, even if the URL isn't in the
database.
# This can lead to false positives.
#
# Default: no
#PhishingAlwaysBlockSSLMismatch no

# Always block cloaked URLs, even if URL isn't in database.
# This can lead to false positives.
#
# Default: no
#PhishingAlwaysBlockCloak no

# Allow heuristic match to take precedence.
# When enabled, if a heuristic scan (such as phishingScan) detects
# a possible virus/phish it will stop scan immediately. Recommended,
saves CPU
# scan-time.
# When disabled, virus/phish detected by heuristic scans will be
reported only at
# the end of a scan. If an archive contains both a heuristically
detected
# virus/phish, and a real malware, the real malware will be reported
#
# Keep this disabled if you intend to handle "*.Heuristics.*" viruses
# differently from "real" malware.
# If a non-heuristically-detected virus (signature-based) is found
first,
# the scan is interrupted immediately, regardless of this config
option.
#
# Default: no
#HeuristicScanPrecedence yes

##
```

```
## Data Loss Prevention (DLP)
##

# Enable the DLP module
# Default: No
#StructuredDataDetection yes

# This option sets the lowest number of Credit Card numbers found in a
file
# to generate a detect.
# Default: 3
#StructuredMinCreditCardCount 5

# This option sets the lowest number of Social Security Numbers found
# in a file to generate a detect.
# Default: 3
#StructuredMinSSNCount 5

# With this option enabled the DLP module will search for valid
# SSNs formatted as xxx-yy-zzzz
# Default: yes
#StructuredSSNFormatNormal yes

# With this option enabled the DLP module will search for valid
# SSNs formatted as xxxyyzzzz
# Default: no
#StructuredSSNFormatStripped yes


##
## HTML
##

# Perform HTML normalisation and decryption of MS Script Encoder code.
# Default: yes
# If you turn off this option, the original files will still be
scanned, but
# without additional processing.
#ScanHTML yes


##
## Archives
##

# ClamAV can scan within archives and compressed files.
# If you turn off this option, the original files will still be
scanned, but
# without unpacking and additional processing.
# Default: yes
ScanArchive yes
```

```
# Mark encrypted archives as viruses (Encrypted.Zip, Encrypted.RAR).
# Default: no
ArchiveBlockEncrypted no


##
## Limits
##

# The options below protect your system against Denial of Service
attacks
# using archive bombs.

# This option sets the maximum amount of data to be scanned for each
input file.
# Archives and other containers are recursively extracted and scanned
up to this
# value.
# Value of 0 disables the limit
# Note: disabling this limit or setting it too high may result in
severe damage
# to the system.
# Default: 100M
#MaxScanSize 150M

# Files larger than this limit won't be scanned. Affects the input file
itself
# as well as files contained inside it (when the input file is an
archive, a
# document or some other kind of container).
# Value of 0 disables the limit.
# Note: disabling this limit or setting it too high may result in
severe damage
# to the system.
# Default: 25M
#MaxFileSize 30M

# Nested archives are scanned recursively, e.g. if a Zip archive
contains a RAR
# file, all files within it will also be scanned. This options
specifies how
# deeply the process should be continued.
# Note: setting this limit too high may result in severe damage to the
system.
# Default: 16
#MaxRecursion 10

# Number of files to be scanned within an archive, a document, or any
other
# container file.
```

```
# Value of 0 disables the limit.
# Note: disabling this limit or setting it too high may result in
severe damage
# to the system.
# Default: 10000
#MaxFiles 15000


##
## Clamuko settings
##

# Enable Clamuko. Dazuko must be configured and running. Clamuko
supports
# both Dazuko (/dev/dazuko) and DazukoFS (/dev/dazukofs.ctrl). DazukoFS
# is the preferred option. For more information please visit
www.dazuko.org
# Default: no
#ClamukoScanOnAccess yes

# The number of scanner threads that will be started (DazukoFS only).
# Having multiple scanner threads allows Clamuko to serve multiple
# processes simultaneously. This is particularly beneficial on SMP
machines.
# Default: 3
#ClamukoScannerCount 3

# Don't scan files larger than ClamukoMaxFileSize
# Value of 0 disables the limit.
# Default: 5M
#ClamukoMaxFileSize 10M

# Set access mask for Clamuko (Dazuko only).
# Default: no
#ClamukoScanOnOpen yes
#ClamukoScanOnClose yes
#ClamukoScanOnExec yes

# Set the include paths (all files inside them will be scanned). You
can have
# multiple ClamukoIncludePath directives but each directory must be
added
# in a seperate line. (Dazuko only)
# Default: disabled
#ClamukoIncludePath /home
#ClamukoIncludePath /students

# Set the exclude paths. All subdirectories are also excluded. (Dazuko
only)
# Default: disabled
#ClamukoExcludePath /home/bofh
```

```
# With this option you can whitelist specific UIDs. Processes with
these UIDs
# will be able to access all files.
# This option can be used multiple times (one per line).
# Default: disabled
#ClamukoExcludeUID 0

# With this option enabled ClamAV will load bytecode from the database.
# It is highly recommended you keep this option on, otherwise you'll
miss detections for many new viruses.
# Default: yes
#Bytecode yes

# Set bytecode security level.
# Possible values:
#       None - no security at all, meant for debugging. DO NOT USE THIS
ON PRODUCTION SYSTEMS
#         This value is only available if clamav was built with --
enable-debug!
#       TrustSigned - trust bytecode loaded from signed .c[lv]d files,
#                insert runtime safety checks for bytecode loaded from
other sources
#       Paranoid - don't trust any bytecode, insert runtime checks for
all
# Recommended: TrustSigned, because bytecode in .cvd files already has
these checks
# Note that by default only signed bytecode is loaded, currently you
can only
# load unsigned bytecode in --enable-debug mode.
#
# Default: TrustSigned
#BytecodeSecurity TrustSigned

# Set bytecode timeout in miliseconds.
#
# Default: 5000
# BytecodeTimeout 1000
```

/etc/group

Die Benutzerkennung bei CentOS 6.x unter welcher der Dämon ClamAV läuft, ist **clamav**. Der Serverdienst DansGuardian hingegen läuft unter dem Nutzerkonto **dansguardian**.

Damit nun der gemeinsame Zugriff auf den Unix-Socket funktionieren kann, muss der Gruppe **dansguardian** der Nutzer **clamav** hinzugefügt werden.

```
# usermod -a -G dansguardian clamav
```

Möchte man explizit nachkucken, ob das Hinzufügen des Nutzers zum Gruppe auch wirklich geklappt hat, benutzt man folgende Abfrage.

```
# cat /etc/group | grep dansguardian
```

erster Programmstart

Nun ist es an der Zeit unseren **ClamAV**-Daemon das erste mal zu starten.

```
# service clamd start
Starting Clam AntiVirus Daemon: LibClamAV Warning:
**********************************************
LibClamAV Warning: ***  The virus database is older than 7 days!  ***
LibClamAV Warning: ***   Please update it as soon as possible.    ***
LibClamAV Warning: **********************************************
                                                     [  OK  ]
```

Wir müssen also unser Virendatenbank erst einmal updaten - Hierzu nutzen wir das Programm **freshclam** aus dem Paket ***clamav***. Wir stoppen nun erst einmal unseren Daemon uns fahren mit der Installation und Konfiguration der weiteren Schritte fort.

```
# service clamd stop
Stopping Clam AntiVirus Daemon:                            [  OK  ]
```

automatisches Starten des Daemon beim Systemstart

Damit nun unser ClamAV-Daemon beim Booten automatisch gestartet wird, nehmen wir noch folgende Konfigurationsschritte vor.

```
# chkconfig clamd on
```

Anschließend überprüfen wir noch unsere Änderung:

```
# chkconfig --list | grep clamd
clamd           0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

freshlam Konfiguration

Damit ClamAV stets mit den aktuellen Vireninformationen versorgen wird, steht und das Programm **freshclam** aus dem Paket **clamav** zu Diensten.

In der Standardkonfiguration sorgt **freshclam** dafür, dass **1x am Tag** ein Update der Virenpattern-Datenbank vorgenommen wird. Bei Bedarf können wir den Updatezyklus unseren Erfordernissen anpassen und so z.B. alle Stunde überprüfen lassen ob neue Patternfiles vorhanden sind und diese dann auf unseren Rechner herunterzuladen und in die lokale Datenbak einfließen zu lassen. Hierbei stehen uns prinzipiell zwei Mechanismen zur Verfügung, die **crontab** und der **Daemon-Modus**. Beide

Varianten könnten im System parallel genutzt werden - nachfolgend werden bei Möglichkeiten kurz beschrieben.

Nutzung crontab

Die erste und einfache Variante besteht darin das Update-Script, welches sich mit dem Namen **freshclam** aktuell und standardmäßig unter ***/etc/cron.daily*** befindet, nach ***/etc/cron.hourly/*** zu verschieben. Das Updatescript beinhaltet folgende Parameter und Aufrufe:

```
#!/bin/sh

### A simple update script for the clamav virus database.
### This could as well be replaced by a SysV script.

### fix log file if needed
LOG_FILE="/var/log/clamav/freshclam.log"
if [ ! -f "$LOG_FILE" ]; then
    touch "$LOG_FILE"
    chmod 644 "$LOG_FILE"
    chown clamav.clamav "$LOG_FILE"
fi

/usr/bin/freshclam \
    --quiet \
    --datadir="/var/clamav" \
    --log="$LOG_FILE" \
    --daemon-notify="/etc/clamd.conf"
```

Wir verschieben also das Script bei Bedarf nach ***/etc/cron.hourly/***.

```
 # mv /etc/cron.daily/freshclam /etc/cron.hourly/
```

Nutzung Daemon-Modus

Die zuvor erwähnte zweite Möglichkeit zum Updaten der Virenpattern-Datenbank ist die Nutzung des **freshclam-Daemons**, der im Hintergrund läuft und regelmäßig zu den Pattenservern eine Abfrage startet.

Startscript

Da bei unserer Installation kein passendes Init-V-Script mitgeliefert wurde legen wir uns ein eigenes Startscript an.

```
# vim /etc/init.d/freshclamd
```

/etc/init.d/freshclamd

```
#!/bin/sh
#
# freshclamd    Init Script to start/stop the freshclamd.
#
# chkconfig: - 62 38
# description: freshclam is an update daemon for Clam AV database.
#
# processname: freshclamd
# config: /etc/freshclam.conf
# pidfile: /var/run/clamav/freshclam.pid

# Source function library
. /etc/init.d/functions

# Get network config
. /etc/sysconfig/network

test -f /etc/freshclam.conf || exit 0

RETVAL=0
DATA_DIR="/var/clamav"
CLAMD_CONF_FILE="/etc/clamd.conf"
LOG_FILE="/var/log/clamav/freshclam.log"

if [ ! -f "$LOG_FILE" ]; then
    touch "$LOG_FILE"
    chmod 644 "$LOG_FILE"
    chown clamav.clamav "$LOG_FILE"
fi

start() {
        echo -n $"Starting freshclam: "
        # Start me up!
        #       --log="$LOG_FILE" \
        #       --log-verbose \
        daemon /usr/bin/freshclam -d -p /var/run/clamav/freshclam.pid \
                -c 48 \
                --quiet \
                --datadir="$DATA_DIR" \
                --daemon-notify="$CLAMD_CONF_FILE"
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/freshclam
        return $RETVAL
}

stop() {
        echo -n $"Stopping freshclam: "
        killproc freshclam
        RETVAL=$?
        echo
```

```
        [ $RETVAL -eq 0 ] && rm -f /var/run/clamav/freshclam.pid
/var/lock/subsys/freshclam
        return $RETVAL
}

restart() {
        stop
        start
}

reload() {
        echo -n $"Reloading DB: "
        killproc freshclam -ALRM
        RETVAL=$?
        echo
        return $RETVAL
}


case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status freshclam
        ;;
  restart)
        restart
        ;;
  condrestart)
        [ -f /var/lock/subsys/freshclam ] && restart || :
        ;;
  reload)
        reload
        ;;
  *)
        echo $"Usage: $0
{start|stop|status|restart|condrestart|reload}"
        exit 1
esac

exit $?
```

Anschließend passen wir noch die Dateirechte an:

```
# chmod +x /etc/init.d/freshclamd
```

Konfiguration

Wir passen nun in der Konfigurationsdatei ***/etc/freshclam.conf*** das Updateintervall unseren Vorstellungen entsprechend an.

```
 # vim /etc/freshclam.conf
```

```
# vim /etc/freshclam.conf

...
# Number of database checks per day.
# Default: 12 (every two hours)
# Django 2011-11-169 für halbstündlichen Virenpatterndatenbankcheck
# default :# Checks 48
Checks 48
...
```

Somit ergibt sich folgende Gesamtkonfigurationsdatei des ***freshclam-daemon***.

/etc/freshclam.conf

```
##
## Example config file for freshclam
## Please read the freshclam.conf(5) manual before editing this file.
##


# Comment or remove the line below.
#Example

# Path to the database directory.
# WARNING: It must match clamd.conf's directive!
# Default: hardcoded (depends on installation options)
DatabaseDirectory /var/clamav

# Path to the log file (make sure it has proper permissions)
# Default: disabled
UpdateLogFile /var/log/clamav/freshclam.log

# Maximum size of the log file.
# Value of 0 disables the limit.
# You may use 'M' or 'm' for megabytes (1M = 1m = 1048576 bytes)
# and 'K' or 'k' for kilobytes (1K = 1k = 1024 bytes).
# in bytes just don't use modifiers.
# Default: 1M
#LogFileMaxSize 2M

# Log time with each message.
# Default: no
#LogTime yes
```

```
# Enable verbose logging.
# Default: no
#LogVerbose yes

# Use system logger (can work together with UpdateLogFile).
# Default: no
LogSyslog yes

# Specify the type of syslog messages - please refer to 'man syslog'
# for facility names.
# Default: LOG_LOCAL6
#LogFacility LOG_MAIL

# This option allows you to save the process identifier of the daemon
# Default: disabled
#PidFile /var/run/freshclam.pid

# By default when started freshclam drops privileges and switches to
the
# "clamav" user. This directive allows you to change the database
owner.
# Default: clamav (may depend on installation options)
DatabaseOwner clamav

# Initialize supplementary group access (freshclam must be started by
root).
# Default: no
#AllowSupplementaryGroups yes

# Use DNS to verify virus database version. Freshclam uses DNS TXT
records
# to verify database and software versions. With this directive you can
change
# the database verification domain.
# WARNING: Do not touch it unless you're configuring freshclam to use
your
# own database verification domain.
# Default: current.cvd.clamav.net
#DNSDatabaseInfo current.cvd.clamav.net

# Uncomment the following line and replace XY with your country
# code. See http://www.iana.org/cctld/cctld-whois.htm for the full
list.
# You can use db.XY.ipv6.clamav.net for IPv6 connections.
#DatabaseMirror db.XY.clamav.net

# database.clamav.net is a round-robin record which points to our most
# reliable mirrors. It's used as a fall back in case db.XY.clamav.net
is
# not working. DO NOT TOUCH the following line unless you know what you
```

```
# are doing.
DatabaseMirror db.de.clamav.net
DatabaseMirror db.local.clamav.net

# How many attempts to make before giving up.
# Default: 3 (per mirror)
#MaxAttempts 5

# With this option you can control scripted updates. It's highly
recommended
# to keep it enabled.
# Default: yes
#ScriptedUpdates yes

# By default freshclam will keep the local databases (.cld)
uncompressed to
# make their handling faster. With this option you can enable the
compression;
# the change will take effect with the next database update.
# Default: no
#CompressLocalDatabase no

# With this option you can provide custom sources (http:// or file://)
for
# database files. This option can be used multiple times.
# Default: no custom URLs
#DatabaseCustomURL http://myserver.com/mysigs.ndb
#DatabaseCustomURL file:///mnt/nfs/local.hdb

# Number of database checks per day.
# Default: 12 (every two hours)
# Django 2011-11-169 für halbstündlichen Virenpatterndatenbankcheck
# default :# Checks 48
Checks 48

# Proxy settings
# Default: disabled
#HTTPProxyServer myproxy.com
#HTTPProxyPort 1234
#HTTPProxyUsername myusername
#HTTPProxyPassword mypass

# If your servers are behind a firewall/proxy which applies User-Agent
# filtering you can use this option to force the use of a different
# User-Agent header.
# Default: clamav/version_number
#HTTPUserAgent SomeUserAgentIdString

# Use aaa.bbb.ccc.ddd as client address for downloading databases.
Useful for
# multi-homed systems.
```

```
# Default: Use OS'es default outgoing IP address.
#LocalIPAddress aaa.bbb.ccc.ddd

# Send the RELOAD command to clamd.
# Default: no
NotifyClamd /etc/clamd.conf

# Run command after successful database update.
# Default: disabled
#OnUpdateExecute command

# Run command when database update process fails.
# Default: disabled
#OnErrorExecute command

# Run command when freshclam reports outdated version.
# In the command string %v will be replaced by the new version number.
# Default: disabled
#OnOutdatedExecute command

# Don't fork into background.
# Default: no
#Foreground yes

# Enable debug messages in libclamav.
# Default: no
#Debug yes

# Timeout in seconds when connecting to database server.
# Default: 30
#ConnectTimeout 60

# Timeout in seconds when reading from database server.
# Default: 30
#ReceiveTimeout 60

# With this option enabled, freshclam will attempt to load new
# databases into memory to make sure they are properly handled
# by libclamav before replacing the old ones.
# Default: yes
#TestDatabases yes

# When enabled freshclam will submit statistics to the ClamAV Project
about
# the latest virus detections in your environment. The ClamAV
maintainers
# will then use this data to determine what types of malware are the
most
# detected in the field and in what geographic area they are.
# Freshclam will connect to clamd in order to get recent statistics.
# Default: no
```

```
#SubmitDetectionStats /path/to/clamd.conf

# Country of origin of malware/detection statistics (for statistical
# purposes only). The statistics collector at ClamAV.net will look up
# your IP address to determine the geographical origin of the malware
# reported by your installation. If this installation is mainly used to
# scan data which comes from a different location, please enable this
# option and enter a two-letter code (see
http://www.iana.org/domains/root/db/)
# of the country of origin.
# Default: disabled
#DetectionStatsCountry country-code

# This option enables support for our "Personal Statistics" service.
# When this option is enabled, the information on malware detected by
# your clamd installation is made available to you through our website.
# To get your HostID, log on http://www.stats.clamav.net and add a new
# host to your host list. Once you have the HostID, uncomment this
option
# and paste the HostID here. As soon as your freshclam starts
submitting
# information to our stats collecting service, you will be able to view
# the statistics of this clamd installation by logging into
# http://www.stats.clamav.net with the same credentials you used to
# generate the HostID. For more information refer to:
# http://www.clamav.net/support/faq/faq-cctts/
# This feature requires SubmitDetectionStats to be enabled.
# Default: disabled
#DetectionStatsHostID unique-id

# This option enables support for Google Safe Browsing. When activated
for
# the first time, freshclam will download a new database file
(safebrowsing.cvd)
# which will be automatically loaded by clamd and clamscan during the
next
# reload, provided that the heuristic phishing detection is turned on.
This
# database includes information about websites that may be phishing
sites or
# possible sources of malware. When using this option, it's mandatory
to run
# freshclam at least every 30 minutes.
# Freshclam uses the ClamAV's mirror infrastructure to distribute the
# database and its updates but all the contents are provided under
Google's
# terms of use. See
http://code.google.com/support/bin/answer.py?answer=70015
# and http://safebrowsing.clamav.net for more information.
# Default: disabled
#SafeBrowsing yes
```

```
# This option enables downloading of bytecode.cvd, which includes
additional
# detection mechanisms and improvements to the ClamAV engine.
# Default: enabled
#Bytecode yes

# Download an additional 3rd party signature database distributed
through
# the ClamAV mirrors. Here you can find a list of available databases:
# http://www.clamav.net/download/cvd/3rdparty
# This option can be used multiple times.
#ExtraDatabase dbname1
#ExtraDatabase dbname2
```

erster Programmstart

Unseren Updatemechanismus **freshclam-daemon** starten wir wie gewohnt mit:

```
# service freshclamd start
Starting freshclam:                                        [  OK  ]
```

Im Logfile ***/var/log/clamav/freshclam.log*** wird der Programmaufruf entsprechend dokumentiert:

```
# tail -f /var/log/clamav/freshclam.log
```

```
-------------------------------------
freshclam daemon 0.97.3 (OS: linux-gnu, ARCH: x86_64, CPU: x86_64)
ClamAV update process started at Wed Nov 16 13:23:31 2011
main.cvd is up to date (version: 54, sigs: 1044387, f-level: 60, builder:
sven)
WARNING: getfile: Unknown response from remote server (IP: 213.174.32.130)
WARNING: getpatch: Can't download daily-13811.cdiff from db.de.clamav.net
WARNING: getfile: daily-13811.cdiff not found on remote server (IP:
188.40.42.237)
WARNING: getpatch: Can't download daily-13811.cdiff from db.de.clamav.net
WARNING: getfile: daily-13811.cdiff not found on remote server (IP:
212.1.60.18)
WARNING: getpatch: Can't download daily-13811.cdiff from db.de.clamav.net
WARNING: Incremental update failed, trying to download daily.cvd
Downloading daily.cvd [100%]
daily.cvd updated (version: 13952, sigs: 30442, f-level: 60, builder:
arnaud)
Downloading bytecode.cvd [100%]
bytecode.cvd updated (version: 152, sigs: 38, f-level: 60, builder: edwin)
Database updated (1074867 signatures) from db.de.clamav.net (IP:
62.27.56.14)
-------------------------------------
```

automatisches Starten des Daemon beim Systemstart

Damit nun unser freshcam-Daemon beim Booten automatisch gestartet wird, nehmen wir noch folgende Konfigurationsschritte vor.

```
# chkconfig freshclamd on
```

Anschließend überprüfen wir noch unsere Änderung:

```
# chkconfig --list | grep freshclamd
freshclamd      0:Aus   1:Aus   2:Ein   3:Ein   4:Ein   5:Ein   6:Aus
```

clamav Start

Da unsere Virendatenbank nun uptodate ist können wir den clamav-Daemon nun ohne Fehlermeldung starten:

```
# service clamd start
Starting Clam AntiVirus Daemon:                            [  OK  ]
```

Im Logfile ***/var/log/clamav/clamd.log*** wir der Programmstart entsprechend dokumentiert:

```
# tail -f /var/log/clamav/clamd.log
```

```
Wed Nov 16 13:16:15 2011 -> +++ Started at Wed Nov 16 13:16:15 2011
Wed Nov 16 13:16:15 2011 -> clamd daemon 0.97.3 (OS: linux-gnu, ARCH:
x86_64, CPU: x86_64)
Wed Nov 16 13:16:15 2011 -> Running as user clamav (UID 497, GID 497)
Wed Nov 16 13:16:15 2011 -> Log file size limited to -1 bytes.
Wed Nov 16 13:16:15 2011 -> Reading databases from /var/clamav
Wed Nov 16 13:16:15 2011 -> Not loading PUA signatures.
Wed Nov 16 13:16:15 2011 -> Bytecode: Security mode set to "TrustSigned".
Wed Nov 16 13:16:18 2011 -> Loaded 1054224 signatures.
Wed Nov 16 13:16:18 2011 -> TCP: Bound to address 127.0.0.1 on port 3310
Wed Nov 16 13:16:18 2011 -> TCP: Setting connection queue length to 30
Wed Nov 16 13:16:18 2011 -> LOCAL: Unix socket file
/var/run/clamav/clamd.sock
Wed Nov 16 13:16:18 2011 -> LOCAL: Setting connection queue length to 30
Wed Nov 16 13:16:18 2011 -> Limits: Global size limit set to 104857600
bytes.
Wed Nov 16 13:16:18 2011 -> Limits: File size limit set to 26214400 bytes.
Wed Nov 16 13:16:18 2011 -> Limits: Recursion level limit set to 16.
Wed Nov 16 13:16:18 2011 -> Limits: Files limit set to 10000.
Wed Nov 16 13:16:18 2011 -> Archive support enabled.
Wed Nov 16 13:16:18 2011 -> Algorithmic detection enabled.
Wed Nov 16 13:16:18 2011 -> Portable Executable support enabled.
Wed Nov 16 13:16:18 2011 -> ELF support enabled.
Wed Nov 16 13:16:18 2011 -> Detection of broken executables enabled.
```

```
Wed Nov 16 13:16:18 2011 -> Mail files support enabled.
Wed Nov 16 13:16:18 2011 -> OLE2 support enabled.
Wed Nov 16 13:16:18 2011 -> PDF support enabled.
Wed Nov 16 13:16:18 2011 -> HTML support enabled.
Wed Nov 16 13:16:18 2011 -> Self checking every 600 seconds.
Wed Nov 16 13:16:24 2011 -> Pid file removed.
Wed Nov 16 13:16:24 2011 -> --- Stopped at Wed Nov 16 13:16:24 2011
Wed Nov 16 13:16:24 2011 -> Socket file removed.
Wed Nov 16 13:25:51 2011 -> +++ Started at Wed Nov 16 13:25:51 2011
Wed Nov 16 13:25:51 2011 -> clamd daemon 0.97.3 (OS: linux-gnu, ARCH:
x86_64, CPU: x86_64)
Wed Nov 16 13:25:51 2011 -> Running as user clamav (UID 497, GID 497)
Wed Nov 16 13:25:51 2011 -> Log file size limited to -1 bytes.
Wed Nov 16 13:25:51 2011 -> Reading databases from /var/clamav
Wed Nov 16 13:25:51 2011 -> Not loading PUA signatures.
Wed Nov 16 13:25:51 2011 -> Bytecode: Security mode set to "TrustSigned".
Wed Nov 16 13:25:54 2011 -> Loaded 1073608 signatures.
Wed Nov 16 13:25:54 2011 -> TCP: Bound to address 127.0.0.1 on port 3310
Wed Nov 16 13:25:54 2011 -> TCP: Setting connection queue length to 30
Wed Nov 16 13:25:54 2011 -> LOCAL: Unix socket file
/var/run/clamav/clamd.sock
Wed Nov 16 13:25:54 2011 -> LOCAL: Setting connection queue length to 30
Wed Nov 16 13:25:54 2011 -> Limits: Global size limit set to 104857600
bytes.
Wed Nov 16 13:25:54 2011 -> Limits: File size limit set to 26214400 bytes.
Wed Nov 16 13:25:54 2011 -> Limits: Recursion level limit set to 16.
Wed Nov 16 13:25:54 2011 -> Limits: Files limit set to 10000.
Wed Nov 16 13:25:54 2011 -> Archive support enabled.
Wed Nov 16 13:25:54 2011 -> Algorithmic detection enabled.
Wed Nov 16 13:25:54 2011 -> Portable Executable support enabled.
Wed Nov 16 13:25:54 2011 -> ELF support enabled.
Wed Nov 16 13:25:54 2011 -> Detection of broken executables enabled.
Wed Nov 16 13:25:54 2011 -> Mail files support enabled.
Wed Nov 16 13:25:54 2011 -> OLE2 support enabled.
Wed Nov 16 13:25:54 2011 -> PDF support enabled.
Wed Nov 16 13:25:54 2011 -> HTML support enabled.
Wed Nov 16 13:25:54 2011 -> Self checking every 600 seconds.
```

clamscan testen

Zum Schluß überprüfen wir noch, ob unser Virenscanner richtig arbeitet. Hierzu besorgen wir uns ein Virenpattern-Testfile.

```
 # wget
http://dansguardian.org/downloads/2/Variants/AVTest/danger/eicar.com.txt -O
/tmp/eicar.com.txt
```

```
--2011-11-16 13:29:49--
http://dansguardian.org/downloads/2/Variants/AVTest/danger/eicar.com.txt
```

```
Resolving dansguardian.org... 89.16.172.190, 2001:41c8:1:5847::2
Connecting to dansguardian.org|89.16.172.190|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 68 [text/plain]
Saving to: “/tmp/eicar.com.txt”

100%[==========================================================================
==========================================>] 68          --.-K/s   in 0s
```

Die erhalten Eicar-Testdatei lassen wir nun von **clamscan** überprüfen.

```
 # clamscan -v /tmp/eicar.com.txt
```

```
Scanning /tmp/eicar.com.txt
/tmp/eicar.com.txt: Eicar-Test-Signature FOUND

----------- SCAN SUMMARY -----------
Known viruses: 1073608
Engine version: 0.97.3
Scanned directories: 0
Scanned files: 1
Infected files: 1
Data scanned: 0.00 MB
Data read: 0.00 MB (ratio 0.00:1)
Time: 3.774 sec (0 m 3 s)
```

Links

- **Zurück zum Kapitel >>Proxyserver Squid mit Web-Contentfilter Dansguardian und Virenfilter ClamAV unter CentOS 6.x<<**
- **Zurück zu >>Projekte und Themenkapitel<<**
- **Zurück zur Startseite**

[1)]
Clam AntiVirus
[2)]
GNU General Public License