

Best Practices and Common Pitfalls in ClamAV deployment

Török Edwin
3rd December 2008



Avoiding Common Pitfalls

General

- Build/install issues
- File/directory permissions and ownership
- Resource limits
- Signature count
- DNS

Upgrading

- Reading about configuration changes
- Reading release notes
- Restarting daemons
- Testing new version
- Follow recommended upgrade procedure

- ❏ Engine configuration overview
 - Logging
 - Setting scan limits
 - Optimizing for SMP (multi-CPU, multi-core)
 - Using PUA
- ❏ Freshclam
 - Distributing updates on local network
 - Configuring SubmitDetectionStats
- ❏ Configuring milter
- ❏ Load balancing

Avoiding Common Pitfalls



KNOW MORE NETWORK RISKS
NO MORE GUESSING



- ❏ Use distribution provided package
 - But don't use obsolete versions (0.90)
 - They can't cope with the current amount of signatures in DB
 - Debian stable users: use volatile!
- ❏ Use certified binaries from Sourcefire
- ❏ Build it yourself
 - Have needed libraries installed (with devel package):
 - Zlib 1.2.2+, GMP 3+, optionally: Bzip2 1.0.5+, check 0.9.5
 - Create *clamav* user and group
 - `./configure && make; make install`
 - If you have lib in non-standard location tell configure, for example: `--with-libgmp-prefix=/usr/local/`



Permissions and ownership

- ❏ Clamd socket
 - Clamd needs write permission for the parent dir
 - Clients need write access to it
 - User *clamav* in *clamd.conf*
- ❏ Database directory
 - Freshclam needs write access, read access for everybody else
 - Owned by *clamav* user is sufficient (same user as in *freshclam.conf*)
- ❏ Set up a special user for freshclam, and allow only that user to have write access to the DB
- ❏ Log file: write access for parent directory for *clamav* user

❏ Clamav-milter

- MTA <-> milter socket
 - Run milter as clamav user, and add it to the MTA's group (AllowSupplementaryGroups yes); needs group write permission on socket
 - Run milter as MTA user, and clamd as clamav (planned for a future release)
- Milter <-> clamd socket
 - If milter user == clamd user -> ok
 - If milter user != clamd user, add to clamav group

❏ Max File size

- Must be equal or larger than the MaxFileSize in clamd.conf
- Also need to be large enough for freshclam to create the databases

❏ Maximum memory

- Must be enough to load the DB + scan + threads
- Don't set it unless absolutely needed
- If you still do, don't set lower than 1 GB for safety
- Hard to estimate memory needed with multiple threads and multiple archive formats

❏ Be aware that init scripts may set these limits (**ulimit**)

Why is resource usage high?

❏ Memory

- Fragmentation (free memory that can't be given back to system), due to the behaviour of **system's libc**
 - Will be reduced in a future release by using custom allocator
 - Can lead to double memory usage when reloading the DB
- Certain archive formats can use a lot of memory
 - Beyond our control, that is how those archive formats works
- Memory mapping (virtual mem usage increased)
 - We don't necessarily read all the file in memory, the OS manages how much it reads, but if we map entire file, the virtual memory usage can rise
 - Glibc's caching of arenas uses virtual memory that don't use swap/physical memory, but increase virtual size
 - Don't put limits on virtual memory usage! (**ulimit -v unlimited**)

- ❏ The number of signatures suddenly decreases
 - This is normal after a main.cvd update
 - Some signatures that were producing false positives have been dropped
 - Some signatures in main.cvd get replaced with better ones from daily.cvd (so total count drops)
- ❏ Number of signatures loaded doesn't match what freshclam/sigtool reports
 - PUA signatures are not loaded by default
- ❏ Signature count is double/half of freshclam's
 - Check for duplicate databases

- ❏ Freshclam uses DNS TXT queries to check for updated database
 - If it cannot make a DNS query to `current.cvd.clamav.net` (for example due to restrictions in an enterprise environment), it will display a message:
 - **Invalid DNS reply. Falling back to HTTP mode**
 - You don't need to worry, freshclam is still functioning correctly, it just needs to fetch a few hundred bytes over HTTP from time to time to check for a new database

Best Practices for Upgrading



KNOW MORE NETWORK RISKS
NO MORE GUESSING



Before you upgrade

- ❏ Read the release notes!
- ❏ Read the release notes!
 - It has important information that you should be aware of when upgrading
- ❏ Read the “What's New” document
 - Explains some of the major new features/changes
- ❏ Review configuration changes
 - New features may be introduced
 - Some options can become default
- ❏ Backup the old configuration file
 - Also make sure you can restore the old version if something goes wrong during the upgrade

- ❏ Do the upgrade on a test server first
 - Remove old version (if built from source)
 - Install new version
 - from package / build from source
 - Edit configuration files and review new flags
 - Restart daemons
 - Check that you are running the correct version
 - `freshclam --version`
 - `clamdscan --version`
 - run *ldconfig* if not
 - Test by using **test/clam.exe** and/or **EICAR**
 - Optionally test by running '**make check**'
 - Valgrind may report leaks/races from libc, you can ignore these

Upgrade (cont'd)

- ❏ Run freshclam to fetch new DB
- ❏ Upgrade production server

Best Practices for Configuring



KNOW MORE NETWORK RISKS
NO MORE GUESSING



Daemon configuration overview

- ❏ /etc/clamd.conf
- ❏ Logging
 - LogFile, LogTime
 - LogSyslog
 - UpdateLogFile (freshclam)
 - Monitor both logfiles
- ❏ Sockets
 - LocalSocket (UNIX socket)
 - TCPSocket, TCPAddr for remote scanning
 - You can use both
 - except if you're using clamav-milter (planned fix for a future release)

Daemon configuration overview (cont'd)

- ❏ User
 - Default is to not drop privileges
 - Do ***not run as root*** without setting User!
- ❏ AllowSupplementaryGroups
 - If the **User** configured above is part of multiple groups, it enables access to all of the files accessible by these groups
 - /etc/group example: clamav:x:61:clamav, amavis
- ❏ Foreground, Debug
 - for debugging
- ❏ ExcludePath (^/proc/, ^/sys/, ^/dev/, ..)
 - Default is scan everything

Engine configuration: limits

StreamMaxLength

- for STREAM command
 - Remote scanning: clamdscan - <file
- Will truncate file if exceeded

MaxFileSize

- Maximum size of original file, considered CLEAN if larger

MaxScanSize

- Maximum amount of data to scan (including files inside archives)

MaxRecursion

MaxFiles

Engine configuration (default off)

❏ ArchiveBlockEncrypted

- Can't be scanned, your decision if you want to block

❏ Potentially Unwanted Applications

- **DetectPUA Yes**

- Categories

- example: **IncludePUA Spy**

- <http://www.clamav.net/support/pua/>

- Good for corporate environments, if you don't want sniffers, password crackers etc. on your network
- Usually not good for home users

Engine configuration (default off) (cont'd)



❏ Data Loss Prevention

- **StructuredDataDetection yes**
- Detects credit card and social security numbers (US only)
- Recommended to do heavy testing, before deploying in a corporate environment

❏ See etc/clamd.conf in the source tarball for other flags

- ❏ Use a fast temporary directory
 - If you have plenty of RAM storage
 - **TemporaryDirectory /dev/shm**
 - If /tmp is using tmpfs, then **TemporaryDirectory /tmp**
 - Or just install plenty of RAM, so the kernel will cache /tmp operations
 - Or use fast disks
 - No RAID (or at most RAID0) for /tmp
 - RAID1 / RAID10 for files you want to scan often
 - A filesystem optimized for read throughput (e.g. XFS)
 - Lower capacity but higher RPM disks

Optimization overview (cont'd)

- ❏ On SMP systems
 - Set **MaxThreads** ~ $2 * \text{Number_of_Cores} + 1$
 - When scanning large directories use **clamscan -m**
 - This will give you maximum scan speed, but you may not be able to do much else on the system
- ❏ Avoid streaming the file if clamd is on same host
 - Use **clamscan file** or **clamscan --fdpass - <file**
 - Instead of *clamscan - <file*

Configuring freshclam

- ❏ DatabaseMirror db.XY.clamav.net (XY = country code)
 - <http://www.iana.org/cctld/cctld-whois.htm>
- ❏ NotifyClamd
 - Tells clamd to immediately reload its databases, instead of waiting for the *SelfCheck* interval
- ❏ Run as a daemon (**-d**), or launch from a cronjob
- ❏ Distribute updates on local network
 - Use a proxy (incremental updates possible)
 - Master freshclam (no incremental updates)
 - Set *ScriptedUpdates Off* on all machines
 - Set master to download to your webserver's *DocumentRoot*
 - Point DatabaseMirror on the slaves to your webserver

Submitting Detection Statistics

SubmitDetectionStats

- Default is off
- Requires LogTime, LogFile in clamd.conf
- When enabled, freshclam will submit last 10 (up to 50) viruses you detected each time it checks for a DB update
 - Data sent: filename, timestamp, virusname
 - 0.94.2 supports sending detection statistics through a proxy
- If your installation is mainly used to scan data which comes from a different location, override with *DetectionStatsCountry*
 - default is geolocation based on IP address

• --submit-stats will only submit, and not update the DB!

SubmitDetectionStats

- ❏ The statistics will be publicly viewable
- ❏ We currently use the data to observe
 - virus outbreaks
 - trends
 - which signatures are better

Configuring milters

Exim

- Native support
- **av_scanner = clamd:/var/run/clamav/clamd**

Postfix

- milter interface: clamav-milter
 - **smtpd_milters = unix:/clamav/clamav-milter**

qmail

- qmail-scanner

Sendmail

- **INPUT_MAIL_FILTER(`clamav',
`S=local:/var/run/clamav/clmilter.sock, F=,
T=S:4m;R:4m;C:30s;E:10m')dnl**
- **define(`confINPUT_MAIL_FILTERS', `clamav')**

Configuring milters (cont'd)

❏ Amavisd-new:

- If your MTA doesn't support native / milter interface
- If you want to interface to spamd also
- Recommended to use a milter/native interface if available instead of amavis
 - Because Amavis does redundant operations in Perl such as splitting email attachments (which clamd does anyway)
 - Not needed as a mail format parser (ClamAV can parse)
- Be careful how it calls clamd
 - Use clamdscan for speed
 - Pass it the raw email, not just attachments
 - `@av_scanners = (['Clam Antivirus-clamd',
 \&ask_daemon, ["CONTSCAN }\n",
 '/var/run/clamav/clamd'], qr/\bOK$/, qr/\bFOUND$/,
 qr/^\.*?: (?!Infected Archive)(.*) FOUND$/],);`

❏ Clamscan Pitfalls

- **Don't call clamscan repeatedly for scanning data!**
- It needs to read the DB each time
- Several seconds per each file, instead of several files per second with clamdscan
- Make sure that **clamdscan** is called, and not clamscan
- You can use clamscan as a fallback if clamd dies

Load balancing

- ❖ First try tweaking MaxThreads and see if you really need to load-balance
- ❖ Clamav-milter does round-robin load balancing if you specify multiple remote clamd (--server, or ClamdSocket in a future release)
- ❖ When load balancing clamd, take into consideration that using the CONTSCAN and FDPASS commands is faster than STREAM

Where to go for help

- ❏ <http://www.clamav.net/support/faq>
- ❏ <http://www.clamav.net/doc/latest/clamdoc.pdf>
- ❏ <http://wiki.clamav.net/Main/WebHome>
- ❏ Clamav-users mailing list
 - Search archives:
<http://lurker.clamav.net/list/clamav-users.html>
 - Subscribe and ask questions if you don't find the answer:
<http://lists.clamav.net/mailman/listinfo/clamav-users>

Questions?



KNOW MORE NETWORK RISKS
NO MORE GUESSING

