

# Creating signatures for ClamAV

## 1 Introduction

CVD (ClamAV Virus Database) is a digitally signed tarball file that contains one or more databases. The header is a 512 bytes long string with colon separated fields:

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

`sigtool --info` displays detailed information on CVD files:

```
zolw@localhost:/usr/local/share/clamav$ sigtool -i daily.cvd
Build time: 11 Sep 2004 21-07 +0200
Version: 487
# of signatures: 1189
Functionality level: 2
Builder: ccordes
MD5: a3f4f98694229e461f17d2aa254e9a43
Digital signature: uwJS6d+y/9g5SXGE0Hh1rXyjZW/PgK/zqVtWWVL3/tfHEnA17z6VB2IBR2
Verification OK.
```

There are two CVD databases in ClamAV: *main.cvd* and *daily.cvd* for daily updates.

## 2 MD5 signatures

There's an easy way to create signatures for static malware using MD5 checksums. To create a signature for `test.exe` use the `--md5` option of `sigtool`:

```
zolw@localhost:/tmp/test$ sigtool --md5 test.exe > test.hdb
zolw@localhost:/tmp/test$ cat test.hdb
48c4533230e1ae1c118c741c0db19dfb:17387:test.exe
```

That's it! The signature is ready to use:

```
zolw@localhost:/tmp/test$ clamscan -d test.hdb test.exe
test.exe: test.exe FOUND
```

```
----- SCAN SUMMARY -----
Known viruses: 1
Scanned directories: 0
Scanned files: 1
Infected files: 1
Data scanned: 0.02 MB
I/O buffer size: 131072 bytes
Time: 0.024 sec (0 m 0 s)
```

You can edit it to change the name (by default sigtool uses the file name). Remember that all MD5 signatures must be placed in \*.hdb files, you can include any number of sigs in a file. To get them automatically loaded every time clamscan/clamd starts just copy them to the local virus database directory.

### 3 Hexadecimal signatures

ClamAV keeps viral fragments in hexadecimal format. If you don't know how to get a proper signature please try the MD5 method or submit your sample on <http://www.clamav.net/sendvirus.html>.

#### 3.1 Hexadecimal format

You can use sigtool --hex-dump to convert arbitrary data into hexadecimal format:

```
zolw@localhost:/tmp/test$ sigtool --hex-dump
How do I look in hex?
486f7720646f2049206c6f6f6b20696e206865783f0a
```

#### 3.2 Wildcards

ClamAV supports the following extensions in hex signatures:

- ??  
Match any byte.

- \*  
Match any number of bytes.
- {n}  
Match n bytes.
- {-n}  
Match n or less bytes.
- {n-}  
Match n or more bytes.
- (a|b)  
Match a and b (you can use more alternate characters).

### 3.3 Basic signature format

The simplest signatures are of the format:

```
MalwareName=HexSignature
```

ClamAV will analyse a whole content of a file trying to match it. All signatures of this type must be placed in \*.db files.

### 3.4 Extended signature format

Extended signature format allows target type and offset specification. The format is:

```
MalwareName:TargetType:Offset:HexSignature[:MinEngineFunctionalityLevel]
```

where TargetType is a decimal number describing a target type:

- 0 = any file
- 1 = Portable Executable
- 2 = OLE2 component (e.g. VBA script)
- 3 = HTML (normalised)
- 4 = Mail file
- 5 = Graphics (to help catching exploits in JPEG files)

And `Offset` is an asterisk or a decimal number `n` possibly combined with a special string:

- `*` = any
- `n` = absolute offset
- `EOF-n` = end of file minus `n` bytes

Signatures for Portable Executables files (target = 1) also support:

- `EP+n` = entry point plus `n` bytes (`EP+0` if you want to anchor to `EP`)
- `Sx+n` = start of section's `x` (counted from 0) data plus `n` bytes

All signatures in the extended format must be placed in `*.ndb` files.

### 3.5 Signature names

ClamAV uses the following prefixes for particular malware:

- *Worm* for Internet worms
- *Trojan* for backdoor programs
- *JS* for Java Script malware
- *VBS* for VBS malware
- *W97M*, *W2000M* for Word macro viruses
- *X97M*, *X2000M* for Excel macro viruses
- *O97M*, *O2000M* for general Office macro viruses
- *DoS* for Denial of Service attack software
- *Exploit* for popular exploits
- *VirTool* for virus construction kits
- *Dialer* for dialers
- *Joke* for hoaxes

## 4 Special files

### 4.1 HTML

ClamAV contains a special HTML normalisation code required to detect HTML exploits. Running `sigtool --html-normalise` on a HTML file will produce the following files:

- `comment.html` - the whole file normalised
- `nocomment.html` - the file normalised, with all comments removed
- `script.html` - the parts of the file in `<script>` tags (lowercased)

The code automatically decodes `JScript.encode` parts and char ref's (e.g. `&#102;`). You need to create a signature against one of the created files. To eliminate potential false positive alerts you should use extended signature format with target type of 3.

### 4.2 Compressed Portable Executable files

If the file is compressed with UPX, FSG, or Petite run `clamscan` with `--debug --leave-temps`. Example output on FSG compressed file:

```
LibClamAV debug: UPX/FSG: empty section found - assuming compression
LibClamAV debug: FSG: found old EP @1554
LibClamAV debug: FSG: Successfully decompressed
LibClamAV debug: UPX/FSG: Decompressed data saved in /tmp/clamav-4eba73ff4050
```

And create a signature for `/tmp/clamav-4eba73ff4050a26`

## 5 Building CVD files - ClamAV maintainers only

Run `freshclam` to check you're using the latest databases. Next go to some **empty** temporary directory and execute the following command:

```
sigtool --unpack-current daily.cvd
```

This will unpack all databases from the current *daily.cvd* database. Add signatures to appropriate files and build the final CVD:

```
sigtool --build daily.cvd --server SIGNING_SERVER
```

where `SIGNING_SERVER` is one of the ClamAV Signing Servers you have access to. This command will automatically generate binary database with a digital signature.

```
LibClamAV debug: Loading databases from .
LibClamAV debug: Loading ./daily.db
LibClamAV debug: Loading ./daily.hdb
LibClamAV debug: Initializing trie.
Database properly parsed.
Signatures: 183
COPYING
tar: main.db: Cannot stat: No such file or directory
tar: main.hdb: Cannot stat: No such file or directory
daily.db
daily.hdb
tar: Notes: Cannot stat: No such file or directory
tar: Error exit delayed from previous errors
Builder id: tkojm
Password:
Signature received (length = 171).
Database daily.cvd created.
```

Don't worry about "No such file or directory" *tar* errors. Finally, you can verify the new database with:

```
zolw@localhost:/usr/local/share/clamav$ sigtool -i daily.cvd
Build time: 26 Aug 2004 22:41 +0200
Version: 473
# of signatures: 183
Functionality level: 2
Builder: tkojm
MD5: 0e89235392c1a1142dda0d022f218903
Digital signature: bWBCx3K07rkd0Qo+zTIZXKhGNvmEz5n/fTUsCEVrdFwhWr2gf5MjSm07nF
Verification OK.
```

Now you must update the main rsync server:

```
rsync -tcz --stats --progress -e ssh daily.cvd clamupload@rsync1.clamav.net:public_html/
ssh rsync1.clamav.net -i ~/.ssh/id_rsa -l clamavdb sleep 1
```

Please consult [1] for more information. After an update please send a summary to clamav-virusdb@lists.sf.net. Thanks !

## References

- [1] Luca Gibelli: *Mirroring the Virus Database*  
<http://www.clamav.net/doc/mirrors>