7 Antivirus Tumors

by Eric Davisson

McAfee Enterprise VirusScan (not the home version of their AV) has a peculiar way of quarantining malware. If an anti-virus product wants to keep a forensic copy of removed malware, it must either move it to an area of the system that it doesn't scan, or it must somehow transform this malware data so it can no longer be seen by the anti-virus signature. VirusScan is almost able to get away with the second option. Almost.

A VirusScan quarantine file (`.bup`) is an odd form of an archive format called Compound File Binary Format that can usually be read by `7zip`. This file contains two files. One of them is a file that contains metadata on the original malware. The other file is the malware file that was removed. Both of these files have been XOR encoded with a one byte key of `0x6a` (ASCII '`j`'). This `7zip` file is archive mode only, so it has no compression. All of this is extremely useful.

Let's say that hypothetically all '`X`' characters look like malware to our AV. (This is a bit contrived, but we'll get back to a real example soon.) This X is `0x58` or `0b01011000`. To bitwise XOR this char with `0x6A` would give us '`2`' (`0x32` or `0b00110010`). So our PoC would be '`X2`' for a signature that looked for '`X`'. Why? Our tumor has the contents of '`X2`', and since that contains '`X`', it's bad malware and needs to be quarantined. The file gets XORed to become '`2X`' and archived with the metadata. If you did a hexdump on this forensic `.bup` file, the contents of '`2X`' are still visibly malicious and need to be quarantined!

I neither have nor want access to McAfee's signatures, but we all have access to ClamAV's set of signatures. It is possible (and highly verified) that there is some signature overlap, as files can come up dirty on multiple vendors' scans. In this PoC, I will use ClamAV's "Worm.VBS.IRC.Alba (Clam)" signature. Despite the name, I assure you that if you submit the file through McAfee, it scans dirty.

The following script extracts a plaintext Clam signature database, parses out the data of our signature, and writes the original and XOR'd form of this signature to a file called `tumor`. This assumes you're on a Linux system with ClamAV installed with signatures loaded in `/var/lib/clamav/`.

```
dd if=/var/lib/clamav/main.cvd of=hivs.tar \
   bs=512 skip=1 2> /dev/null;
tar -x main.db -f hivs.tar 2> /dev/null;
chmod 666 main.db;
rm hivs.tar;
grep "IRC.Alba" main.db                    \
   | grep -o "[0-9a-f]\+\$"                \
   | xxd -r -p | perl -0777 -e             \
   '$k = <>; print $k;
print ($k ^ ("j" x length($k)));'          \
   > tumor;
rm main.db
```

This tumor is *benign*, as its growth eventually stops after a few rounds, and I've not yet been able

```
0000000: 7269 7074 5d27 2b43 6861 7228 2444 292b  ript]'+Char($D)+
0000010: 4368 6172 2824 4129 2b0d 0a27 6e30 3d6f  Char($A)+..'n0=o
0000020: 6e20 313a 4a4f 494e 3a23 3a20 6966 2028  n 1:JOIN:#: if (
0000030: 2024 6d65 2021 3d20 246e 6963 6b20 2927   $me != $nick )'
0000040: 0d0a 277b 202f 6463 6320 7365 6e64 2024  ..'{ /dcc send $
0000050: 6e69 636b 2063 3a5c 6d69 7263 5c64 6f77  nick c:\mirc\dow
0000060: 6e6c 6f61 645c 616c 6261 2e65 7865 207d  nload\alba.exe }
0000070: 272b 4318 031a 1e37 4d41 2902 0b18 424e  '+C....7MA)...BN
0000080: 2e43 4129 020b 1842 4e2b 4341 6760 4d04  .CA)...BN+CAg`M.
0000090: 5a57 0504 4a5b 5020 2523 2450 4950 4a03  ZW..J[P %#$PIPJ.
00000a0: 0c4a 424a 4e07 0f4a 4b57 4a4e 0403 0901  .JBJN..JKWJN....
00000b0: 4a43 4d67 604d 114a 450e 0909 4a19 0f04  JCMg`M.JE...J...
00000c0: 0e4a 4e04 0309 014a 0950 3607 0318 0936  .JN....J.P6....6
00000d0: 0e05 1d04 0605 0b0e 360b 0608 0b44 0f12  ........6....D..
00000e0: 0f4a 174d 4129                           .J.MA)
```

to compose a proof of concept of a *malignant* tumor, one that eventually fills the hard disk. Through experimentation, I suspect that McAfee signatures are more complex than string matches. For example, when McAfee pulls out of my pool a file that previously had no nulls but now does, it often no longer sees it as malware and rejoices. This is a problem as `7zip` introduces nulls in its metadata. Also some malicious data no longer triggers the antivirus when pushed deeper into the file. These barriers may be bypassed by more intimate knowledge of the McAfee signatures.