Malware DB : Pre-processing Malicious Samples for Preliminary Analysis

Raphaela Mettig

Advisor: Dr. Golden G. Richard III

Project Overview

- Database
  - Data bank containing information about our population of malware samples
- Why do we want it?
  - Speed up the preliminary analysis of malware samples
  - Query specific malware from our lab’s repository

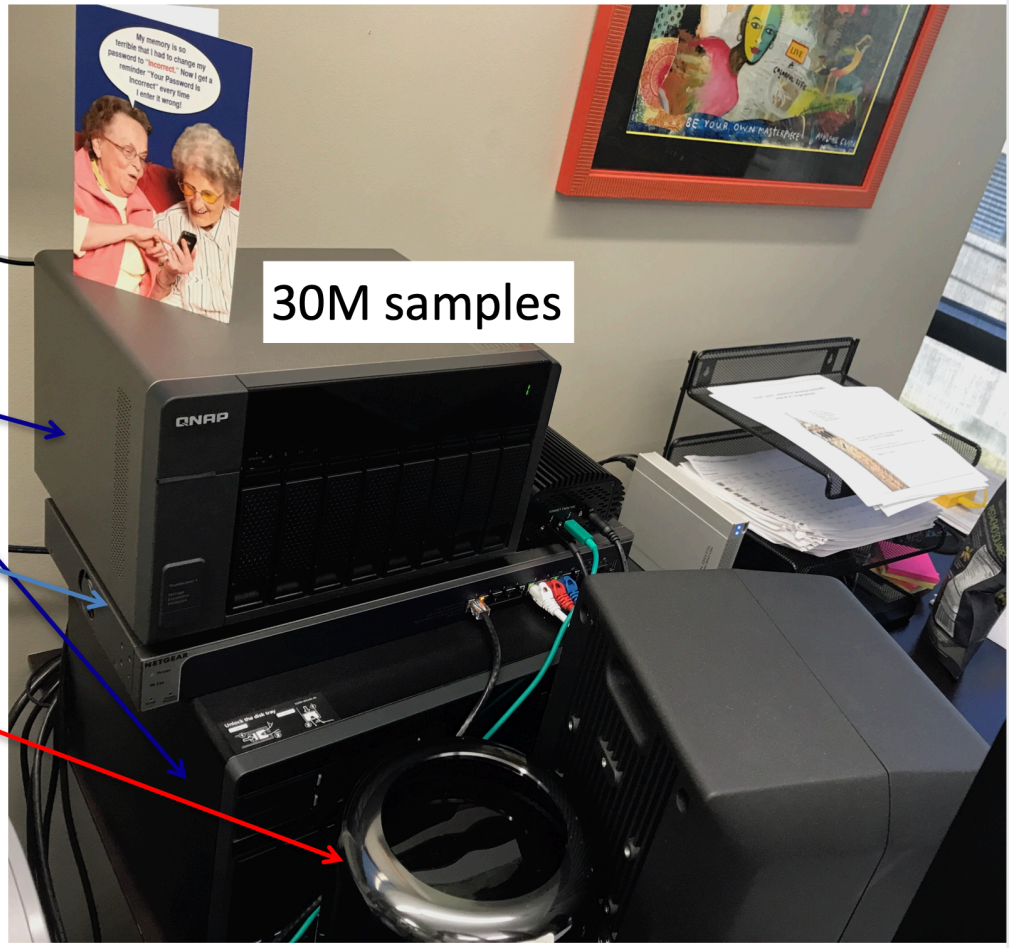
Malware: There's Lots of It

cybersecurity lab

128TB of storage

10Gb networking

12 core Mac Pro w/ 128GB RAM



30M samples

How Can We Use This Information?

- Memory Forensics research
  - Ongoing research projects
    - Android
    - .NET
    - Swift
    - Hooktracer
- Reverse engineering
- Development of malware analysis tools

Malware Analysis Overview

- What is Malware?
  - Malicious computer software
  - Steal credentials, spy on people, encrypt/delete files, cause physical damage to devices and critical infrastructure (yes, really), business disruption, revenue loss, and much more.
- Why is it relevant for cybersecurity research?
  - One of the largest global cybersecurity threats
  - Quickly evolving threat

Malware Analysis Process

```
00000000: 4d5a 9000 0300 0000 0400 0000 ffff 0000  MZ..............
00000010: b800 0000 0000 0000 4000 0000 0000 0000  ........@.......
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000030: 0000 0000 0000 0000 0000 0000 d800 0000  ................
00000040: 0e1f ba0e 00b4 09cd 21b8 014c cd21 5468  ........!..L.!Th
00000050: 6973 2070 726f 6772 616d 2063 616e 6e6f  is program canno
00000060: 7420 6265 2072 756e 2069 6e20 444f 5320  t be run in DOS
00000070: 6d6f 6465 2e0d 0d0a 2400 0000 0000 0000  mode....$.......
00000080: 3843 2a1d 7c22 444e 7c22 444e 7c22 444e  8C*.|"DN|"DN|"DN
00000090: ff3e 4a4e 7222 444e 4a04 4e4e 4422 444e  .>JNr"DNJ.NND"DN
000000a0: 1e3d 574e 7e22 444e bf2d 194e 7f22 444e  .=WN~"DN.-.N."DN
000000b0: 7c22 454e 4e22 444e 4a04 4f4e 7b22 444e  |"ENN"DNJ.ON{"DN
000000c0: bb24 424e 7d22 444e 5269 6368 7c22 444e  .$BN}"DNRich|"DN
000000d0: 0000 0000 0000 0000 5045 0000 4c01 0400  ........PE..L...
000000e0: 73c0 f34a 0000 0000 0000 0000 e000 0f01  s..J............
000000f0: 0b01 0600 00a0 0000 00c0 0000 0000 0000  ................
00000100: 6764 0000 0010 0000 00b0 0000 0000 4000  gd............@.
00000110: 0010 0000 0010 0000 0400 0000 0000 0000  ................
00000120: 0400 0000 0000 0000 0070 0100 0010 0000  .........p......
00000130: 0000 0000 0200 0000 0000 1000 0010 0000  ................
00000140: 0000 1000 0010 0000 0000 0000 1000 0000  ................
00000150: 0000 0000 0000 0000 cce9 0000 3c00 0000  ............<...
00000160: 0040 0100 6026 0000 0000 0000 0000 0000  .@..`&..........
00000170: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000180: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000190: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001a0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001b0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001c0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001d0: 2e74 6578 7400 0000 409c 0000 0010 0000  .text...@.......
000001e0: 00a0 0000 0010 0000 0000 0000 0000 0000  ................
000001f0: 0000 0000 2000 0060 2e72 6461 7461 0000  .... ..`.rdata..
00000200: 383e 0000 00b0 0000 0040 0000 00b0 0000  8>.......@......
00000210: 0000 0000 0000 0000 0000 0000 4000 0040  ............@..@
00000220: 2e64 6174 6100 0000 3c41 0000 00f0 0000  .data...<A......
00000230: 0030 0000 00f0 0000 0000 0000 0000 0000  .0..............
00000240: 0000 0000 4000 00c0 2e72 7372 6300 0000  ....@....rsrc...
00000250: 6026 0000 0040 0100 0030 0000 0020 0100  `&...@...0... ..
00000260: 0000 0000 0000 0000 0000 0000 4000 0040  ............@..@
00000270: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

Fig.1: terminal output after dumping malware binary with xxd

Malware Analysis Process

- Static Analysis
  - Preliminary information collection – file information
  - Reverse Engineering – code analysis (disassembly)
- Dynamic Analysis – examine malware's behavior

```
seg000:000000A6                 lea     eax, [ebp-34h]
seg000:000000A9                 push    eax
seg000:000000AA                 mov     eax, [ebp-40h]
seg000:000000AD                 push    eax
seg000:000000AE                 call    dword ptr [esi]
seg000:000000B0                 push    11h
seg000:000000B2                 push    2
seg000:000000B4                 push    2
seg000:000000B6                 call    eax
seg000:000000B8                 push    eax
seg000:000000B9                 lea     eax, [ebp-3Ch]
seg000:000000BC                 push    eax
seg000:000000BD                 mov     eax, [ebp-40h]
seg000:000000C0                 push    eax
seg000:000000C1                 call    dword ptr [esi]
seg000:000000C3                 mov     esi, eax
seg000:000000C5                 or      ebx, ebx
seg000:000000C7                 xor     ebx, 0FFD9613Ch
seg000:000000CD
```

Fig.2: Screen shot of SQL Slammer disassembly code

Malware Analysis Process

- Static Analysis
  - **Preliminary information collection – file information**
  - Reverse Engineering – code analysis (disassembly)
- Dynamic Analysis – examine malware's behavior

The Database

- Multiple sources
  - Virus Share (>30 million samples)
  - VirusTotal
  - Personal collections
  - Public (open source) repositories
- Information collected
  - General file information: path/name, size, magic number (extension/architecture)
  - File hashes: MD5, SHA1, SHA256
  - Portable Executable (PE) information: DLLs, URLs, IP addresses, strings

```
Raphaelas-MacBook-Pro:vxsamples rmettig$ peframe 0a6b9f7f4c6fdd019a4a188de381befab6fbff5a7566ab4aa64cf2c7cf9609f7


-------------------------------------------------------------------------------
File Information (time: 0:00:11.543547)
-------------------------------------------------------------------------------
filename          0a6b9f7f4c6fdd019a4a188de381befab6fbff5a7566ab4aa64cf2c7cf9609f7
filetype          PE32 executable (GUI) Intel 80386, for MS Windows
filesize          4947976
hash sha256       0a6b9f7f4c6fdd019a4a188de381befab6fbff5a7566ab4aa64cf2c7cf9609f7
virustotal        /
imagebase         0x400000
entrypoint        0x6467
imphash           81791b04c48af73263cb1c9edfc6e78a
datetime          2009-11-06 06:21:39
dll               False
directories       import, tls, resources, relocations
sections          .data, .rsrc, .text *, .rdata *
features          antidbg, packer, crypto


-------------------------------------------------------------------------------
Yara Plugins
-------------------------------------------------------------------------------
CRC32 poly Constant
CRC32 table
IsPE32
IsWindowsGUI
IsPacked
HasOverlay
HasRichSignature


-------------------------------------------------------------------------------
Behavior
-------------------------------------------------------------------------------
Xor
win files operation


-------------------------------------------------------------------------------
Crypto
-------------------------------------------------------------------------------
CRC32 poly Constant
CRC32 table
```

Fig.3: Terminal output after running PEFrame tool on malware sample

Stages of Data Collection

STAGE 1

General File Information

- All samples.

STAGE 2

Format Specific Information

- On a file format basis.
- Currently only Windows PE format files are supported (.exe, .dll, .sys, etc.)

STAGE 3

VirusTotal Scan

- All samples.
- AV detection, nicknames, any extra relevant info.
- Not yet implemented.

Project Status

- Current focus:
  - Optimization for speed
  - PEFiles
- In progress:
  - Implementing PEFrame
  - Testing new architecture
- Challenges:
  - What is relevant?
  - Duplicates

Moving Forward

- Analysis:
  - VirusTotal scans
  - Support for other file formats: ELF (Linux), Android malware, Mach-O, etc.
- Collect statistical information on the samples
  - Most common DLLs, architectures, file types, etc.
  - Can we infer behavior from a set of characteristics?

Thank you! Questions?

- Contact information:
  - Twitter: rmettig_
  - Github: raphasmrocha

- Acknowledgements:
  - Dr. Golden Richard
  - Andrew Case
  - CCT Applied Cybersecurity Lab