De l’analyse morphologique à Cyber-Detect : valorisons les virus !

Jean-Yves Marion

Le 6 Novembre 2017 – Jussieu
About LORIA@Nancy

- Network security
- Cryptography
- Protocol Security and electronic vote
- Malware
- Scada
- Drones
Research on malware detection

LORIA’s High Security Lab
10 millions of malware

- New approaches to detect malware by morphological analysis
- Detection of hidden functionalities
- Computer forensics
In a nutshell: **Multi-dimensional Signatures**

Each signature denotes a piece of a functionality

A database of signatures is a collection of functionalities
- Obfuscated codes
- Packers
- On the fly code generation
- Anti-debug protections
Morphological analysis in a nutshell

Sample name: Email-Korm.Win32.Lentin.h
Number of nodes: 1665

```
push ebp
mov ebp, esp
push 0xff
push dword 0x4091b0
push dword 0x406710
mov eax, [fs:0x0]
push eax
mov [fs:0x0], esp
sub esp, 0x58
push ebx
push esi
push edi
mov [ebp-0x18], esp
```
Functionality identification

A scenario

Is the functionality AES Encrypt implemented inside a binary code?

An untrusted binary code

An implementation of a functionality

www.cyber-detect.com
Functionality identification

The compilation environment
OpenSSL version 1.1.0f
Visual Studio 2013 – 64 bit
Compilation option 01

6 functionalities from demo programs of the OpenSSL distribution

aesni_set_encrypt_key  x86_64_AES_set_encrypt_key_key
b64_read              SEH_begin_AES_cbc_encrypt
MD5_Update             RC5_32_ecb_encrypt

Goal: Find these 6 functionalities inside Openssl.exe
# Functionality identification

<table>
<thead>
<tr>
<th>Version 1.1.0f</th>
<th>aesccm.exe (version 1.1.0f)</th>
<th>OpenSSL.exe</th>
</tr>
</thead>
<tbody>
<tr>
<td>VS 2013 64 bit O1</td>
<td>aesni_set_encrypt_key</td>
<td>b64_read</td>
</tr>
<tr>
<td>0x14000cc10 : 0x14000fb10</td>
<td>0x14010c27c : 0x1400feb0c</td>
<td>0x140af384 : 0x140156e6c</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Version 1.1.0f</th>
<th>aesgcm.exe (version 1.1.0f)</th>
<th>OpenSSL.exe</th>
</tr>
</thead>
<tbody>
<tr>
<td>VS 2013 64 bit O1</td>
<td>AES_set_encrypt_key</td>
<td>SEH_begin_AES_CBC_encrypt</td>
</tr>
<tr>
<td>0x140002040 : 0x14000251d</td>
<td>0x14000251d</td>
<td>0x140af384 : 0x140156e6c</td>
</tr>
</tbody>
</table>
Conclusion

Multi-purpose set of tools to identify similarities between binary codes

- Works on highly obfuscated X86 (arm) codes
- Fast file investigation
- Speed-up reverse engineering
- Simple Integration (IDA, RPC, Scriptable)

- Accurate with Code synchronization
- Home-made signature data-bases of functionalities

www.cyber-detect.com
Questions ?
Thank you!