MIND THE GAP: BUILDING A BRIDGE FROM INTRUSION TO DETECTION

Bart Hopper
WHAT WILL WE COVER?
(SLIDES AVAILABLE ON REQUEST)

• Defining the Threat
• Yara
• Python
• Python PEFFile module
• Entropy measurements
• ClamAV
WHOAMI
THE MOST INTERESTING TOILET IN THE WORLD

Los Banditos Restaurant
Prague, Czech Republic
BUT I HAVE AN ANTIVIRUS PROGRAM.....
<table>
<thead>
<tr>
<th>FA10</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>FA20</td>
<td>00</td>
</tr>
<tr>
<td>FA30</td>
<td>00</td>
</tr>
<tr>
<td>FA40</td>
<td>00</td>
</tr>
<tr>
<td>FA50</td>
<td>00</td>
</tr>
<tr>
<td>FA60</td>
<td>00</td>
</tr>
<tr>
<td>FA70</td>
<td>00</td>
</tr>
<tr>
<td>FA80</td>
<td>00</td>
</tr>
<tr>
<td>FA90</td>
<td>00</td>
</tr>
<tr>
<td>FAA0</td>
<td>00</td>
</tr>
<tr>
<td>FAB0</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>3F</td>
</tr>
<tr>
<td>-----</td>
<td>----</td>
</tr>
<tr>
<td>18ED0</td>
<td></td>
</tr>
<tr>
<td>18EE0</td>
<td></td>
</tr>
<tr>
<td>18EF0</td>
<td></td>
</tr>
<tr>
<td>18FO0</td>
<td></td>
</tr>
<tr>
<td>18F10</td>
<td></td>
</tr>
<tr>
<td>18F20</td>
<td></td>
</tr>
<tr>
<td>18F30</td>
<td></td>
</tr>
<tr>
<td>18F40</td>
<td></td>
</tr>
<tr>
<td>18F50</td>
<td></td>
</tr>
<tr>
<td>18F60</td>
<td></td>
</tr>
<tr>
<td>18F70</td>
<td></td>
</tr>
<tr>
<td>18F80</td>
<td></td>
</tr>
<tr>
<td>18F90</td>
<td></td>
</tr>
<tr>
<td>18FA0</td>
<td></td>
</tr>
<tr>
<td>18FB0</td>
<td></td>
</tr>
<tr>
<td>18FC0</td>
<td></td>
</tr>
<tr>
<td>18FD0</td>
<td></td>
</tr>
</tbody>
</table>

*RESUME SENT TO HR*
rule This_is_a_rule_name
{
  This is a multi-line comment
  strings:
    $a = "MZ"
    $b = { 62 79 74 65 20 63 6F 64 65 } // "byte code" in hex
    $c = /[Ee][Vv][Ii][Ll] / // Regular expressions
    $d = { 77 ?9 6C 64 ?? 63 61 72 64 } // "wild card" "wild_card" "wild-card"
    $e = { 61 62 [2-6] 65 } // Jumps
    $f = { ( 48 | 68 ) 65 6C 6C 6F 20 77 6F 72 6C 64 } // Alternatives
    $g = "Hello world" nocase // ignore case
    $h = "Hello world" wide // Finds UTF-16 unicode
    $i = "hello world" wide ascii // ASCII or UTF-16 unicode

  condition:
    ($a and $b) or
    #a == 6 or //c counting strings
    $a at 0 or // file or memory offset
    filesize > 10KB // filesizes check
}

YARA
HTTP://PLUSVIC.GITHUB.IO/YARA/
XOR RULES

```
rule XOR_3F
{
  strings:

  $MZ = \{72 65\}
  $This = \{6B 57 56 4C\}
  $program = \{4F 4D 50 58 4D 5E 52\}
  $cannot = \{5C 5E 51 51 50 4B\}
  $be = \{5D 5A\}
  $run_in_DOS_mode = \{4D 4A 51 1F 56 51 1F 7B 70 6C 1F 52 50 5B 5A\}
  $PE = \{6F 7A\}

  condition:
    all of them
}
```
HIDDEN EVIL
rule Embedded_Executable
{
    strings:
    $MZ = "MZ"
    $PE = "PE"

    condition:
    ((#MZ > 1) and (#PE > 1) and
    (for any i in (2..#MZ) : (uint16(uint32(@MZ[i]+0x3c)) == 0x4550)))
}

rule PE_file
{
    condition:
    uint16(uint32(0x3c)) == 0x4550 and uint16(0) == 0x5A4D
}

rule DLL
{
    strings:
    $MZ = "MZ"

    condition:
    $MZ and
    for any i in (1..#MZ): (uint16(uint32(@MZ[i] + 0x3c)) == 0x4550) and ((uint16(uint32(0x3c) + 0x16) & 0x2000) == 0x2000)
}
include "javascript.rule"

rule PDF_AutoAction
{
    strings:
    $aa = "/AA"
    condition:
    $aa
}

rule PDF_AcroForm
{
    strings:
    $acroform = "/AcroForm"
    condition:
    $acroform
}

rule PDF_Colors
{
    strings:
    $colors = "/Colors"
    condition:
    $colors
}

rule PDF_OpenAction
ALTERNATE DATA STREAMS (ADS)

c:\ads>echo yara > hidden.txt:hidden.txt

c:\ads>dir
Volume in drive C is Windows7_OS
Volume Serial Number is CC24-EZD0

Directory of c:\ads

06/17/2014 08:46 PM   <DIR>          .
06/17/2014 08:46 PM   <DIR>          ..
06/17/2014 08:46 PM   0 hidden.txt   0 bytes
               1 File(s)
               2 Dir(s)   6,289,940,480 bytes free

c:\ads>dir /R
Volume in drive C is Windows7_OS
Volume Serial Number is CC24-EZD0

Directory of c:\ads

06/17/2014 08:46 PM   <DIR>          .
06/17/2014 08:46 PM   <DIR>         ..
06/17/2014 08:46 PM   0 hidden.txt   0 bytes
               7 hidden.txt:hidden.txt:$DATA
               1 File(s)
               2 Dir(s)   6,289,940,480 bytes free
SCANNING ALTERNATE DATA STREAMS

```
\% ads\%> cat --version
\% ads\%> cat (GNU textutils) 2.0
Written by Torbjorn Granlund and Richard M. Stallman.

Copyright (C) 1999 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

\% ads\%> cat hidden.txt:hidden.txt
yara

\% ads\%> cat hidden.txt:hidden.txt > scanme.txt

\% ads\%
```
SCANNING THE REGISTRY
SCANNING A PROCESS
Dos MZ Header
DOS Stub
PE File Header
   PE Signature
Image_Optional_Header

Section Table
   Array of Image_Section_Headers
   Data Directories

   Sections
      .idata
      .rsrc
      .data
      .text
      .src

Source: http://resources.infosecinstitute.com/presenting-the-pe-header/
SCANNING A PROCESS (PART 2)
PROBLEMS WITH SCANNING

- Scanning via smb

- GIL (Global Interpreter Lock)
  (https://jessenoller.com/blog/2009/02/01/python-threads-and-the-global-interpreter-lock)

- yara-ctypes
  (https://github.com/mjdorma/yara-ctypes)
GENERATING METERPRETER PAYLOADS

```
root@kali:/work/SANS# cat command.txt
./msfpayload windows/meterpreter/reverse_tcp R | msfencode -c 5 -o /work/SANS/payload1.exe -t exe
root@kali:/work/SANS#  
```
YARAGENERATOR
(HTTPS://YARAGENERATOR.COM/)

```
root@kali:~

root@kali:~/work/SANS# python yaraGenerator.py -r Metasploit_reverse_tcp -a Bart -d "Sample Yara Rules" -v -f /oex /work/SANS/payloads/
```
rule Meterpreter_reverse_tcp
{
  author = "Bart"
date = "2014-05-04"
description = "Sample Yara Rules"
hash0 = "a7a9ab15a334653efb4743d607a1577d9"
hash1 = "05e94038ac48cdd59755107c9e69a296"
hash2 = "077960f52f1223d2220741ce1e62af8a"
hash3 = "ddfc200493cc0eb5fe434cd97e07035a"
hash4 = "66bccecd3e861f09107bbbc5d6421c02a"
sample_filetype = "exe"
yaragenerator = "https://github.com/Xen0ph0n/YaraGenerator"
strings:
  $string0 = "<tr %s><th colspan"
  $string1 = "ProductName" wids
  $string2 = "2 %s>Requests per second:<th><td colspan"
  $string3 = "-sd] median max"
  $string4 = "@@@@-
  $string5 = "Partial results are valid but processing is incomplete"
  $string6 = "%AsAsAs"
  $string7 = "End of file found"
  $string8 = "Your code just forked, and you are currently executing in the child process"
  $string9 = "Cannot open gnuplot output file"
  $string10 = "2 %s>Total POSTsd:<th><td colspan"
  $string11 = "apr_sockaddr_info_get() for %s"
  $string12 = "Network system is unavailable"
  $string13 = "Licensed to The Apache Software Foundation, http://www.apache.org/"
  $string14 = "Socket operation on non-socket"
  $string15 = "Unless required by applicable law or agreed to in writing, software distributed under the License is" wids
  $string16 = "ABCDEFGHIJKLMNOPQRSTUVWXYZKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"
  $string17 = "deviation apart. These results are NOT reliable."
}
condition:
  17 of them
CREATING RULES BASED ON RESOURCES
(HTTPS://WWW.ANGUSJ.COM/RESOURCHEACKER/)
PEVIEW
(HTTP://WWW.MAGMA.CA/~WJR)
EASY STRING TO HEXCODE CONVERSION

```python
>>> def hexit(target):
...     for i in target:
...         print(hex(ord(i))[2:], end=' ')
...     print('n')
...
>>> while 1:
...     a = input('Text to convert: ')
...     hexit(a)
...
Text to convert: Hello World!
48 65 6c 6c 6f 20 67 6f 72 6f 6e 20 61

Text to convert: Hex strings made easy
48 65 74 20 73 74 72 69 6e 67 6f 72 6f 6e 73 20 61 64 65 64 20 65 61 73 74

Text to convert:
```
COMPUTING FILE HASH USING PYTHON

```python
>>> import hashlib
>>> a = open('c://opensource/sqlite3.exe', 'rb')
>>> filecontents = a.read()
>>> hashlib.md5(filecontents).hexdigest()
'86d0b9d71a89d6d20716ad2ca7d79a7f'
>>> hashlib.sha256(filecontents).hexdigest()
'f315ababef49827ad09f3374b49f660312fda23a77ae948d060c7d24ca8748b33'
>>> for i in dir(hashlib):
    ...    print(i)
    ...
    ...
    _all_
    __builtins__
    __cached__
    __doc__
    __file__
    __get_builtin_constructor
    __initializing_
    __loader__
    __name__
    __package__
    __qualname__
    _hashlib
    algorithms_available
    algorithms_guaranteed
    md5
    new
    sha1
    sha224
    sha256
    sha384
    sha512
```
PYTHON MODULE – PEFILE
(HTTPS://CODE.GOOGLE.COM/P/PEFILE/)
INSTALLING ON PYTHON3
GENERATING HASHES WITH PEFILE

```python
C:\Users\Bart>"c:\portable-python\Portable Python 3.2.1.1\App\python.exe"
Python 3.2.1 (default, Jul 10 2011, 21:51:15) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import pefile
>>> a = pefile.PE('c:\opensource\sqlite3.exe')
>>> for section in a.sections:
...     print(section.Name.decode('UTF-8') + ' \t ' + section.get_hash_md5())
... .text  4cb191a96df7164f0ed311d388b087c2
.data   18af4b7d4f332d6e1506e6ad8974d667
.rdata  057c209c229fc6eee9ebf17283036bb5
.bss    d41d8cd98f00b204e9800998ecf8427e
.idata  acdd13a79b8147f14860d8ae5a882ae
.CRT    53395a20920ad888ec1295694001968b
.tls    072e42667dff2dba0986986adb1ac313
>>>```
EXTRACTING FILE SEGMENTS

7-Zip [64] 9.20 Copyright (c) 1999-2010 Igor Pavlov 2010-11-18
p7zip Version 9.20 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,1 CPU)

Processing archive: payload1.exe

Extracting .text
Extracting .rdata
Extracting .data
Extracting .rsrcc/VERSION/1
Extracting .debug0

Everything is Ok

Files: 5
Size: 67506
Compressed: 73802

root@kali:/work/SANS#
HASHES SORTED BY SECTIONS

```
root@kali:~/work/SANS# cat hashes.txt | sort -k 2 1
sort: open failed: 1: No such file or directory
root@kali:~/work/SANS# cat hashes.txt | sort -k 2
```

```
payload1.exe .data 283b5f792323d57b9db4d2bcc46580f8
payload2.exe .data 283b5f792323d57b9db4d2bcc46580f8
payload3.exe .data 283b5f792323d57b9db4d2bcc46580f8
payload4.exe .data 283b5f792323d57b9db4d2bcc46580f8
payload5.exe .data 283b5f792323d57b9db4d2bcc46580f8
payload1.exe .rdata 25d7ceee3aa85bb3e8c5174736f6f830
payload2.exe .rdata 25d7ceee3aa85bb3e8c5174736f6f830
payload3.exe .rdata 25d7ceee3aa85bb3e8c5174736f6f830
payload4.exe .rdata 25d7ceee3aa85bb3e8c5174736f6f830
payload5.exe .rdata 25d7ceee3aa85bb3e8c5174736f6f830
payload1.exe .rsic r13a9413aa7291b6fc85d75bfcede381
payload2.exe .rsic r13a9413aa7291b6fc85d75bfcede381
payload3.exe .rsic r13a9413aa7291b6fc85d75bfcede381
payload4.exe .rsic r13a9413aa7291b6fc85d75bfcede381
payload5.exe .rsic r13a9413aa7291b6fc85d75bfcede381
payload1.exe .text 0b952e2fc924f7fc96d3170e2b57e5
payload5.exe .text 19adb790b4f51a3dc48da8f02386630
payload4.exe .text db73d2f2aa4e4807f96a4c2a5c8056
payload3.exe .text e1bc7b397e90216ce3fd139f1d20459
payload2.exe .text f4e310263fa48b2ee4e7f9b4ffbf1bb51
```
NOTHING IN COMMON?!

```
root@kali:/work/SANS# python yaraGenerator.py -r Meterpreter -a Bart -d "Only the payload, not the template" -f exe -v /work/SANS/payloads2/

[+] Generating Yara Rule Meterpreter from files located in: /work/SANS/payloads2/
[!] No Common Attributes Found For All Samples, Please Be More Selective
root@kali:/work/SANS#  
```

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Average Entropy</th>
<th>99% Confidence Intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain Text</td>
<td>4.347</td>
<td>4.401 - 5.030</td>
</tr>
<tr>
<td>Native Executables</td>
<td>5.099</td>
<td>6.084 - 6.369</td>
</tr>
<tr>
<td>Packed Executables</td>
<td>6.801</td>
<td>7.199 - 7.267</td>
</tr>
<tr>
<td>Encrypted Executables</td>
<td>7.175</td>
<td>7.295 - 7.312</td>
</tr>
</tbody>
</table>
#!/usr/bin/python

import os
import pefile
import re

for target_file in os.listdir('.\n if (re.search('\.exe\.',target_file)): # a = pefile.PE(target_file)
    for i in a.sections:
        print(target_file + ':\t' + i.Name + ':' + str(i.get_entropy()))
ENTROPY > 6.5
IMPHASH (HASH OF IMPORT TABLE)
(HTTPS://WWW.MANDIANT.COM/BLOG/TRACKING-MALWARE-IMPORT-HASHING/)
IMPHASH OF PAYLOADS

```python
#!/usr/bin/python

import os
import pefile
import re

for i in os.listdir('.
    if re.search('.exe', i):
        a = pefile.PE(i)
        print(i + ' ' + a.get_imphash())
```

```
root@kali: ~

root@kali:/work/SANS# python imphash.py
payload2.exe  481f47bb2c9c21e108d65f52b04c448
payload5.exe  481f47bb2c9c21e108d65f52b04c448
payload3.exe  481f47bb2c9c21e108d65f52b04c448
payload4.exe  481f47bb2c9c21e108d65f52b04c448
payload1.exe  481f47bb2c9c21e108d65f52b04c448
```
IDA PRO
(HTTPS://WWW.HEX-RAYS.COM/INDEX.SHTML)
ANTI-FORENSICS CODE

```
loc_4093F1:
push esi
lea eax, [ebp+SystemTimeAsFileTime]
push eax ; lpSystemTimeAsFileTime
call ds:GetSystemTimeAsFileTime
mov esi, [ebp+SystemTimeAsFileTime.dwHighDateTime]
xor esi, [ebp+SystemTimeAsFileTime.dwLowDateTime]
call ds:GetCurrentProcessId
xor esi, eax
call ds:GetCurrentThreadId
xor esi, eax
call ds:GetTickCount
xor esi, eax
lea eax, [ebp+PerformanceCount]
push eax ; lpPerformanceCount
call ds:QueryPerformanceCounter
mov eax, dword ptr [ebp+PerformanceCount+4]
xor eax, dword ptr [ebp+PerformanceCount]
xor esi, eax
cmp esi, edi
jnz short loc_4093E7
```

```
loc_4093E7:
test esi, ebx
jnz short loc_4093F7
```

```
nov esi, 80B40E64Fh
jmp short loc_4093F7
```

```
nov eax, esi
or eax, 4711h
shl eax, 10h
or esi, eax
```
loc_4093A1:
56    push   esi
8D 45 F8    lea   eax, [ebp+SystemTimeAsFileTime]
50    push   eax    ; lpSystemTimeAsFileTime
FF 15 B0 10 40 00    call   ds:GetSystemTimeAsFileTime
8B 75 FC    mov   esi, [ebp+SystemTimeAsFileTime.dwHighDateTime]
33 75 F8    xor   esi, [ebp+SystemTimeAsFileTime.dwLowDateTime]
FF 15 AC 10 40 00    call   ds:GetCurrentProcessId
33 F0    xor   esi, eax
FF 15 98 10 40 00    call   ds:GetCurrentThreadId
33 F0    xor   esi, eax
FF 15 10 10 40 00    call   ds:GetTickCount
33 F0    xor   esi, eax
8D 45 F0    lea   eax, [ebp+PerformanceCount]
50    push   eax    ; lpPerformanceCount
FF 15 A8 10 40 00    call   ds:QueryPerformanceCounter
8B 45 F4    mov   eax, dword ptr [ebp+PerformanceCount+4]
33 45 F0    xor   eax, dword ptr [ebp+PerformanceCount]
33 F0    xor   esi, eax
3B F7    cmp   esi, edi
75 07    jnz    short loc_4093E7

call
loc_4093E7:
85 F3    test   esi, ebx
75 0C    jnz    short loc_4093F7

BE 4F E6 40 BB    mov   esi, 0BB40E64Fh
EB 10    jmp    short loc_4093F7

8B C6    mov   eax, esi
8D 11 47 00 00    or   eax, 4711h
C1 E0 10    shl   eax, 10h
8B F0    or   esi, eax
CLAMAV
HTTP://WWW.CLAMWIN.COM/

- Hash of the entire file (MD5:Size:MalwareName)
  sigtool –md5 evil.exe > my_signatures.hdb
  (Hashstring:Filesize:MalwareName)

- Hash of Section (Use 7zip)
  .mdb (PESectionSize:MD5:MalwareName)

- Byte patterns
  .ndb (MalwareName:TargetType:Offset:HexSignature)
def main():
    # Local variables
    target = bytes(sys.argv[1], 'UTF-8')
    number_of_rules = c_int()
    virname = c_char_p()
    bytes_scanned = c_int()

    # Bind to DLL
    clam = cdll.libclamav

    # Initialize the database
    result = clam.cl_init()
    if DEBUG_LEVEL > 0:
        if result == 0:
            print('\n[*] Init: Success')

    # Create new engine
    engine = clam.cl_engine_new()

    # Load the database engine
    result = clam.cl_load(DATABASE, engine, byref(number_of_rules), 0x2 | 0x8 | 0x20)
    if DEBUG_LEVEL > 0:
SUMMARY

• Traditional Antivirus programs are ineffective
• Your organization is getting hit with undetected threats
• Undetected malware can cause catastrophic damage
• You need to perform threat analysis to know your specific threats and vulnerabilities
• “For every complex problem there is an answer that is clear, simple, and wrong.” - H. L. Mencken
QUESTIONS?

Spondulas:
http://sourceforge.net/projects/spondulas/

Contact:
Twitter: @d4ncingd4n
Email: barthopper@gmail.com
http://www.linkedin.com/in/barthopper