Malware Dynamic Analysis
Part 1

Veronica Kovah
vkovah.ost at gmail

http://opensecuritytraining.info/MalwareDynamicAnalysis.html
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.
Where are we at?

• Part 1: Introduction
  – Observing an isolated malware analysis lab setup
  – Malware terminology
  – RAT exploration - Poison IVY
  – Behavioral analysis

• Part 2: Persistence techniques
  – Using registry keys
  – Using file systems
  – Using Windows services
Isolated Lab Settings

• It is very important to have an isolated lab machine ready to avoid accidental malware escape
• It should be easy to restore the old state, which is not infected by malware
• Lab with physical machines
  — Use Deep Freeze (restore), FOG (clone/restore), etc.
• Lab with virtual machines
  — Use virtualization solution such as VMware, VirtualBox, KVM, Xen, etc.

[References]
• FOG, http://sourceforge.net/projects/freeghost/
• Vmware, http://www.vmware.com/
• VirtualBox, https://www.virtualbox.org/
• KVM, http://www.linux-kvm.org/page/Main_Page
• Xen, http://www.xen.org/
VirtualBox

• Oracle VM VirtualBox is freely available open source software
• 6 network modes are available
  – Not attached, NAT, Bridged Adapter, Internal Network, Host-only Adapter, Generic Driver
• Can use VMware or Microsoft Virtual PC generated formats

[References]
• VirtualBox, https://www.virtualbox.org/

[Image Sources]
• https://www.virtualbox.org/graphics/vbox_logo2_gradient.png
2 Host-only Networks

• File->Preferences...->Network

[References]
Network Details

- Same for vboxnet1 except IPv4 address, 192.168.57.1
- On host machine, check if you see new network interfaces
  - $ ifconfig
VM’s network setting (1)

- Start *controller* VM first
  - C:\> ipconfig
- Open *controller* VM’s Settings, change
  Network->Adapter 1
VM’s network setting (2)

• Open *victim* VM’s Settings → Network → Adapter 1
  – Attached to = ‘Host-only Adapter’
  – Name = ‘vboxnet1’

• Start VMs and change network setting

<table>
<thead>
<tr>
<th>VM name</th>
<th>controller</th>
<th>victim</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP address</td>
<td>192.168.56.20</td>
<td>192.168.57.100</td>
</tr>
<tr>
<td>Subnet mask</td>
<td>255.255.255.0</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>Default Gateway</td>
<td>192.168.56.1</td>
<td>192.168.57.1</td>
</tr>
<tr>
<td>Preferred DNS Server</td>
<td>192.168.56.1</td>
<td>192.168.57.1</td>
</tr>
</tbody>
</table>
Change IP on Windows

- Start → Control Panel → Network Connections → Local Area Connection → Properties → Internet Protocols (TCP/IP) → Properties
IP Forwarding

• IP Forwarding is disabled by default on Ubuntu
  – $ sudo su
  – # echo 1 > /proc/sys/net/ipv4/ip_forward
• Enable packet forwarding in firewall
  – # iptables -P FORWARD ACCEPT
• Can you ping from victim to controller VM?
Network Sniffing

- Try to capture network traffic
  - `$ wireshark&`
  - From the menu bar, Capture->Options...
  - Do you see any interface?
- Running Wireshark as root is not safe
  - Malformed network traffic can exploit a Wireshark vulnerability
- But you cannot access to any interface without root privilege
- Choice 1: use a simple dumper (wireshark uses this) and then open up the file as a non-root user
  - `$ sudo dumpcap -i vboxnet0 -w /tmp/pi.pcap`
  - Shortcoming: you cannot see network traffic in real time
Wireshark with User Privilege (1)

• This lab will setup sniffing with Wireshark as a non-root user
• Install setcap
  – $ sudo apt-get install libcap2-bin
• Add wireshark group
  – $ sudo groupadd wireshark
  – $ sudo usermod -a -G wireshark student
  – Close all open windows and terminals
  – $ gnome-session-quit

[References]
Wireshark with User Privilege (2)

- $ sudo chgrp wireshark /usr/bin/dumpcap
- $ sudo chmod 750 /usr/bin/dumpcap

• Grant capabilities
  - $ sudo setcap cap_net_raw,cap_net_admin=eip /usr/bin/dumpcap

• Verify
  - $ getcap /usr/bin/dumpcap
Network Connectivity Test

• Can you ping from the victim VM to the host machine?
  – C:\> ping 192.168.57.1

• Start the controller VM

• Can you ping from the victim VM to the controller VM?
  – On the victim VM
    • C:\> ping 192.168.56.20
Capturing Network Packets

- Ping from victim VM to controller VM.
- Capture the traffic using Wireshark with non-root privilege.
- Which network interface did you choose?
IINetSim

• Software suite for simulating common internet services in a lab environment
  – HTTP/HTTPS, DNS, SMTP, etc. servers
  – IRC channel with basic command sets
• Open source, written in Perl
  – v1.2.4 can be installed on Ubuntu 12.04 via the package manager

[References]
• Thomas Hungenberg and Matthias Eckert, INetSim, http://www.inetsim.org/
inetsim setup

• On the host machine
  – $ gedit /etc/inetsim/inetsim.conf
    
    service_bind_address 192.168.57.1
dns_default_ips 192.168.57.1
  – $ sudo inetsim
  – $ wireshark &
    • listen on vboxnet1
• On the victim VM
  – c:> ping www.google.com
Where are we at?

• Part 1: Introduction
  – Observing an isolated malware analysis lab setup
  – Malware terminology
  – RAT exploration - Poison IVY
  – Behavioral analysis

• Part 2: Persistence techniques
  – Using registry keys
  – Using file systems
  – Using Windows services
Malware Terminology (1)
(Just so we can be on the same page throughout the class)

• Virus: “Malware that replicates, commonly by infecting other files in the computer, thus allowing the execution of the malware code and its propagation when those files are activated. Other forms of viruses include boot sector viruses and replicating worms.”

• Worm: “A worm is a self-propagating program that can automatically distribute itself from one computer to another. Worms may propagate themselves using one or more of the following methods”

Malware Terminology (2)

• Trojan: “A malicious application that is unable to spread of its own accord. Historically, the term has been used to refer to applications that appear legitimate and useful, but perform malicious and illicit activity on an affected computer.”

• Backdoor: “A backdoor is a piece of software which, once running on a system, opens a communication vector to the outside so that the computer can be accessed remotely by an attacker.”
  http://www.virusbtn.com/resources/glossary/backdoor.xml
Malware Terminology (3)

- Bot: “A malicious program installed on a computer that is part of a bot network (botnet). Bots are generally backdoor trojans that allow unauthorized access and control of an affected computer. They are often controlled via IRC from a centralized location (although other models of command and control exist).”

- Remote Administration Tool (RAT): “A piece of software that allows a remote "operator" to control a system as if he has physical access to that system.”
  http://en.wikipedia.org/wiki/Remote_administration_software
Malware Terminology (4)

• Downloader: “A type of trojan that downloads other files, which are usually detected as other malware, onto the computer. The Downloader needs to connect to a remote host to download files”

• Dropper: “A type of trojan that drops other files, which are usually detected as other malware, onto the computer. The file to be dropped is included as part of the dropper package”

Malware Terminology (5)

- Spyware: “The term 'spyware' essentially covers any software that gathers information and passes it to a third party without adequate permission from the owner of the data.”
- Adware: “Adware is essentially any software that is funded by advertising.”
- Ransomware: “A type of malware that encrypts files on a victim's system, demanding payment of a ransom in return for the access codes required to unlock the files.”

http://www.virusbtn.com/resources/glossary/index
Vendor Naming Convention

• Conventions derived from Computer Antivirus Research Organization (CARO) Malware Naming Scheme
  — Microsoft, F-Secure

![Worm:Win32/Taterf.K!dll](image)

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Name Convention</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avira</td>
<td>Prefix:Name [Type]</td>
<td>Win32:Zbot-BS [Trj]</td>
</tr>
</tbody>
</table>

[References]
• Virus Naming Conventions, http://www.symantec.com/security_response/virusnaming.jsp
Non-standardized Naming Scheme

<table>
<thead>
<tr>
<th>Antivirus</th>
<th>Result</th>
<th>Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avira</td>
<td>Win32/Kids.worm.167008</td>
<td>20100502</td>
</tr>
<tr>
<td>AvNwr</td>
<td>Worm/Conficker.Z.43</td>
<td>20100502</td>
</tr>
<tr>
<td>Avira+AVL</td>
<td>Worm/W32:Kid sophomore.29Q</td>
<td>20100503</td>
</tr>
<tr>
<td>Avast</td>
<td>Win32/Rockit.gen [RH]</td>
<td>20100502</td>
</tr>
<tr>
<td>AVG</td>
<td>Worm/Downadup</td>
<td>20100502</td>
</tr>
<tr>
<td>BitDefender</td>
<td>Worm.Generic.41542</td>
<td>20100503</td>
</tr>
<tr>
<td>ByteHero</td>
<td></td>
<td>20100502</td>
</tr>
<tr>
<td>CAT-Quick Heal</td>
<td>Win32.Worm.Conficker 3.0</td>
<td>20100502</td>
</tr>
<tr>
<td>ClamAV</td>
<td>Trojan.Dropper-18535</td>
<td>20100503</td>
</tr>
<tr>
<td>Commtouch</td>
<td>W32/Conficker.Generic</td>
<td>20100503</td>
</tr>
<tr>
<td>DrWeb</td>
<td>Win32.HL.UW.Shadow.based</td>
<td>20100503</td>
</tr>
</tbody>
</table>

A result of a Conficker sample at https://www.virustotal.com

See notes for citation
Where are we at?

• Part 1: Introduction
  – Observing an isolated malware analysis lab setup
  – Malware terminology
  – RAT exploration - Poison IVY
  – Behavioral analysis

• Part 2: Persistence techniques
  – Using registry keys
  – Using file systems
  – Using Windows services
Poison Ivy

- Freely available RAT, the latest version is v2.3.2
- **Implant** (Server)
  - Customizable features: Encrypted communications, registry and file manager, screen capture, key logger, NTLM hash captures, etc.
  - No need to update for new features
  - Support 3rd party plugins
    - E.g. port scanner, wifi enumerator (“stumbler”), etc
- **Controller** (Client)
  - Once an implant is deployed, the implant connects to a controller, whose information is built into the implant.

[References]

[Image Sources]
- http://25.media.tumblr.com/tumblr_m83rfveJWO1r6dcg4o1_500.jpg
Simple PI Server Creation

- On the *controller* VM
- Start Poison Ivy
  - MalwareClass/samples/PoisonIvy/Poison Ivy 2.3.2.exe
- File→New Server
- Create Profile with name “pi_agent”
- Connection: set DNS/Port to the controller VM’s IP and set port to 3460
  - 192.168.56.20:3460:0,
Connection
Install

[Image of an installation window with options for profiles, connection, install, and advanced settings.]

See notes for citation 31
Creating pitest.exe

- Advanced: Leave as it is
- Build:
  - Click ‘Generate’ and save as “pitest.exe”
  - Then click ‘OK =>’
- We need to copy pitest.exe to the *victim* VM
  but will skip the step to save time
Client Creation

- On the *controller* VM
- File → New Client
- Verify ‘Listen on Port’ is set to 3460
- Click ‘Start’ button
Executing Poison Ivy Implant

- On the *victim* VM
  - Execute the already prepared PI server
    (MalwareClass/samples/PoisonIvy/pi_agent.exe)
- Once a server connects to the client, you will see
  the following entry on the *controller* VM
Think Evil!

- On the *controller* VM, double click on the ‘pi_agent’ line
  
  **Q1.** Select ‘Remote Shell’ on the left panel, then on the right panel, click the right mouse button and select ‘Activate’. Can you start a calculator to surprise the victim? Hint: “cmd.exe /c ...”
  
  **Q2.** Can you kill the calculator on the *victim* VM?
  
  **Q3.** What’s in the registry value ‘secret_agent’ under HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run? Anything special about it?
Answers for PI Lab (1)

A1. C:\> cmd.exe /c c:\Windows\system32\calc.exe

A2. You can kill the calculator process using Managers→Processes left-side bar
**Answers for PL Lab (2)**

**A3.** Alternate Data Stream (ADS) is attached to C:\WINDOWS\System32

- If you go to C:\WINDOWS\System32, you won't see anything named “pidriver.exe”. Let’s find it with gmer
- Malware occasionally stores data in Alternate Data Stream (ADS). ADS is a mechanism for attaching metadata to files.
- If you use a colon in a filename, the part after the colon will be the metadata name/file, and the part before the colon will be the file it's being attached to
- Explorer doesn't show ADS files, but functions like CreateFile() can access them just fine, so the file still runs.

[References]
- GMER, http://www.gmer.net/
- AlternateStreamView, http://www.nirsoft.net/utils/alternate_data_streams.html
Where are we at?

• Part 1: Introduction
  – Observing an isolated malware analysis lab setup
  – Malware terminology
  – RAT exploration - Poison IVY
  – Behavioral analysis

• Part 2: Persistence techniques
  – Using registry keys
  – Using file systems
  – Using Windows services
Diffing

- Take a snapshot of a clean system state and a snapshot of a compromised system state
- Compare before and after
- Pros: Artifacts can be observed easily
- Cons: Can miss evidence that is created during malware activities and erased purposely by malware
- Tools: regshot, autoruns

[References]

[Image Sources]
- http://familyfun.go.com/assets/cms/printables/0707c_findthedifference.jpg
System Monitoring

- From a clean system state, record every individual change on system and network traffic that appear after execution of made by the suspicious file
- Pro: Can collect all manifested changes
- Cons: Often too much information and need to weed out irrelevant data
- Tools: procmon, Wireshark

[Image Sources]
API Tracing

• Hook and record important API calls made by the suspicious process
• Pro: Can provide visibility into activity beyond the typical file/process/registry/network shown by other tools. Gets you a little closer to the type of interpretation that is required when doing static analysis.
• Cons: Often too much of information and need to weed out irrelevant data. API-specific interpretation can take a lot of time (but still less than static analysis ;))
• Tools: WinApiOverride, Rohitab API Monitor

[References]
• WinAPIOverride, http://jacquelin.potier.free.fr/winapioverride32/
• API Monitor, http://www.rohitab.com/apimonitor

[Image Sources]
• Left, http://fc03.deviantart.net/fs39/f/2008/332/c/d/HAND_TURKEY_by_Bilious.jpg
Debugging

- Set breakpoints inside the suspicious file to stop its execution at a given location and inspect its state. Can break when it calls to important APIs.
- Pro: Provides a superset of the functionality of an API monitor
- Cons: Typically must be be done in conjunction with some basic static analysis and assembly reading. Malware will often change its behavior or refuse to run when being debugged, which requires a work-around.
- Tools: IDA Pro Debugger, OllyDbg, Immunity Debugger, WinDbg
- We will NOT cover this in this class, because x86 assembly is not a prerequisite. See the Intro x86 and Intro Reverse Engineering classes to start working with debuggers.

[Image Sources]
• Top left, http://www.wpclipart.com/computer/humour/debugging.png
• Top right, http://www.phdcomics.com/comics/archive/phd011406s.gif
Behavioral Analysis Techniques

“Always use the easiest tool for the job” :)

- Diffing
- File, Registry, Process, Network Monitoring
- API Tracing
- Debugging

Ease-of-use/Abstraction level

AWESOMA POWA!