

Malware Dynamic Analysis

Part 1

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:



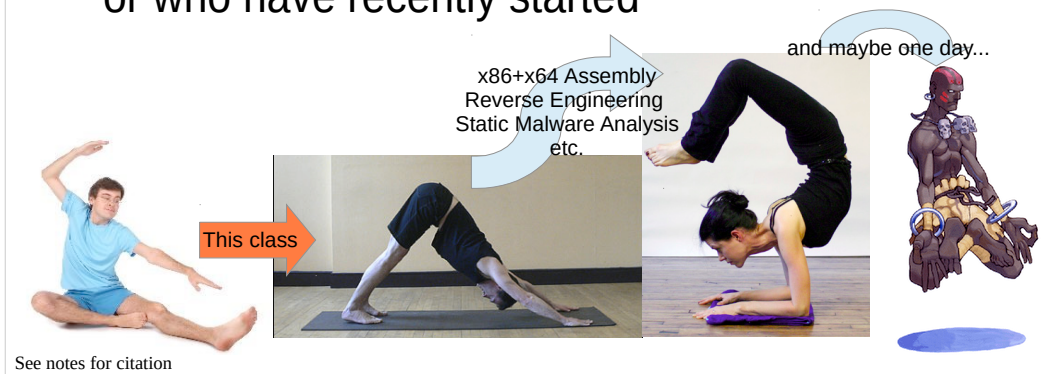
Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).



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.

This class is for people

- Who are interested in computer security
- Who want to understand how malware works
- Who want to start working on malware analysis, or who have recently started



[Image Sources]

- Left, <http://test3-img.ehowcdn.com/article-new/ehow/images/a01/vv/m9/start-yoga-as-male-beginner-800x800.jpg>
- Middle-left, http://www.bbc.co.uk/northyorkshire/content/images/2006/04/05/downward_dog_400x300.jpg
- Middle-right, <http://www.spiritualhealingportal.com/images/photo/yoga9.jpg>
- Right, http://media.giantbomb.com/uploads/0/3/665089-a2dhalsim_thumb.png

 Thanks to 



- Xeno Kovah, Ben Schmoker and Frank Poz for reviewing class materials
- Ezra Moses, MITRE Institute tech support for setting up Ubuntu on the lab machines
- Openmalware.org (offensivecomputing.net) for sharing samples, very good resource

See notes for citation

4

[Image Sources]

- Top, <http://www.thirdcoastrs.com/AP-353%20ASIAN%20GNOME%20-%20BOWING%204web.jpg>
- Middle, http://thumbs.dreamstime.com/thumblarge_510/127589357290776N.jpg

About me and you

- BE in CS and MS in CE (but mostly CS background)
- Security related work experience:
 - Vulnerability research
 - Network IDS/IPS signature development
 - Windows memory integrity measurement/verification
 - Malware analysis and analysis tool development
 - Security product reverse engineering
- Like hands-on work (coding, debugging, and reversing)
- How about you? Any particular topic that you want to learn from this class?

Outline



- Part 1
 - Background concepts & tools
 - Observing an isolated malware analysis lab setup
 - Malware terminology
- Part 2
 - RAT exploration - Poison IVY
 - Persistence techniques
 - Maneuvering techniques
(How malware strategically positions itself)

See notes for citation

6

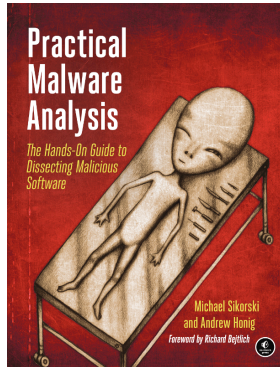
[Image Sources]

- <http://www.sponsor700reps.com/uploads/6/4/3/5/6435698/344557.jpg?620>

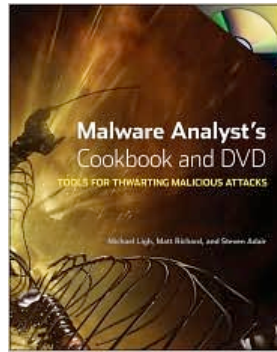
Outline

- 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

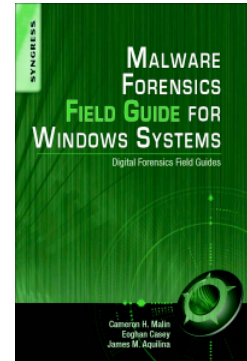
Books



Michael Sikorski
Andrew Honig



Michael Ligh
Steven Adair
Blake Hartstein
Matthew Richard



Cameron H. Malin
Eoghan Casey
James M. Aquilina



See notes for citation

8

[Image Sources]

- Left, http://nostarch.com/sites/default/files/imagecache/product_main_page/practical_malware_analysis_s.png
- Middle, <http://img2.imagesbn.com/images/77180000/77183529.JPG>
- Right, <http://secure-ecsd.elsevier.com/covers/80/Tango2/large/9781597494724.jpg>

Class Conventions

- Slides with  on the left corner means we will perform hands-on lab activities
- Slides with  include answers to lab questions, which often follows lab slides. Please do not read the answers early ;)
- Lines starting with
 - **C:\>** means, you are asked to type in a DOS window on the Windows XP VM but it does not mean the command needs to be executed at the top level
 - **\$** means, you are asked to type in a Linux terminal on the Ubuntu host machine

See notes for citation

9

[Image Sources]

- Top, http://thetalentcode.com/wp-content/uploads/119498535838791757esperimento_chimico_arch_01.svg_med_.png
- Middle, Microsoft clip art

Class Materials

- On the Ubuntu host machine
 - \$ cd ~/MalwareClass && ls
 - \$ cd docs && ls
 - \$ libreoffice MalwareDynamicAnalysis01.odp
 - \$ virtualbox &
- On the *victim* VM
 - On Desktop, open MalwareClass directory
- Please see Notes for citation and check out the original works

Outline

- Part 1
 - **Background concepts & tools**
 - Observing an isolated malware analysis lab setup
 - Malware terminology
- Part 2
 - RAT exploration - Poison IVY
 - Persistence techniques
 - Maneuvering techniques
(How malware strategically positions itself)

PE File

- PE (Portable Executable) is the file format for Windows' executable binaries
 - You can find imported libraries/functions from the PE headers
- 3 conventional ways to use libraries
 - Dynamic link at compile time: .dll files are loaded into the memory space of a process at load time, and the main executable just calls the needed functions in the DLLs
 - LoadLibrary at run time: .dll files are loaded into the memory space of a process on run time
 - Static link at compile time: .lib files are combined into a PE file to make a big fat file that doesn't have external dependencies

Packers

- Originally used to compress executables back when disk space was at a premium
- The executable then decompresses itself in memory and runs as normal
- Nowadays they are mostly used for obfuscating binaries. Specifically since all the data for the original binary is compressed and/or encrypted, it prevents analysts from being able to infer things about the binary based on strings or function imports
- UPX, ASPack, MPRESS, Themida, etc.
- For dynamic analysis, since we will actually execute a sample, this is not a hindrance

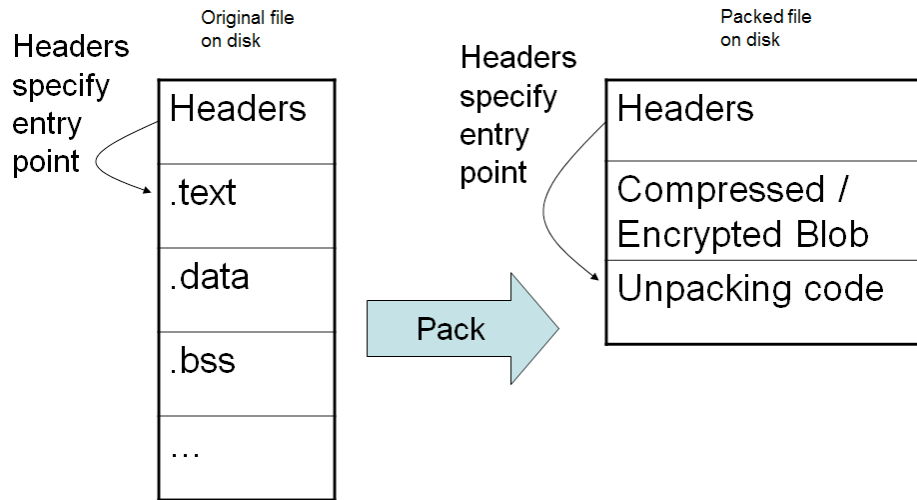
See notes for citation

13

[References]

- UPX, <http://upx.sourceforge.net/>
- ASPack, <http://www.aspack.com/aspack.html>
- MPRESS, <http://www.matcode.com/>
- Themida, <http://www.oreans.com/themida.php>

Packing: File On Disk



From the Life of Binaries class

14

See notes for citation

[References]

- Xeno Kovah, <http://opensecuritytraining.info/LifeOfBinaries.html>



Identifying File Types



Beware impersonation!!

- Identify 5 files' formats in
~/MalwareClass/samples/unknown/
 - By using **file** and **TrID** tools on Ubuntu
 - \$ file ~/MalwareClass/samples/unknown/sample04.exe
 - \$ cd ~/MalwareClass/tools/TrID/
 - \$./trid ~/MalwareClass/samples/unknown/sample04.exe
 - By using **TrIDNet** on the *victim* VM
- File extension
 - Don't rely on the file extension at all!!
 - exe, dll, pdf, doc, docx, xls, xlsx, ppt, pptx, jpg, etc.
- This class focuses on malware in PE files (.exe, .dll, .sys, .scr, .ocx, etc.)

See notes for citation

15

[References]

- Marco Pontello, TrID, <http://mark0.net/soft-trid-e.html>

[Image Sources]

- Right, <http://www.gadgetreview.com/wp-content/uploads/2012/05/Dog-Pirate-Costume-650x472.jpg>

Windows Library Files

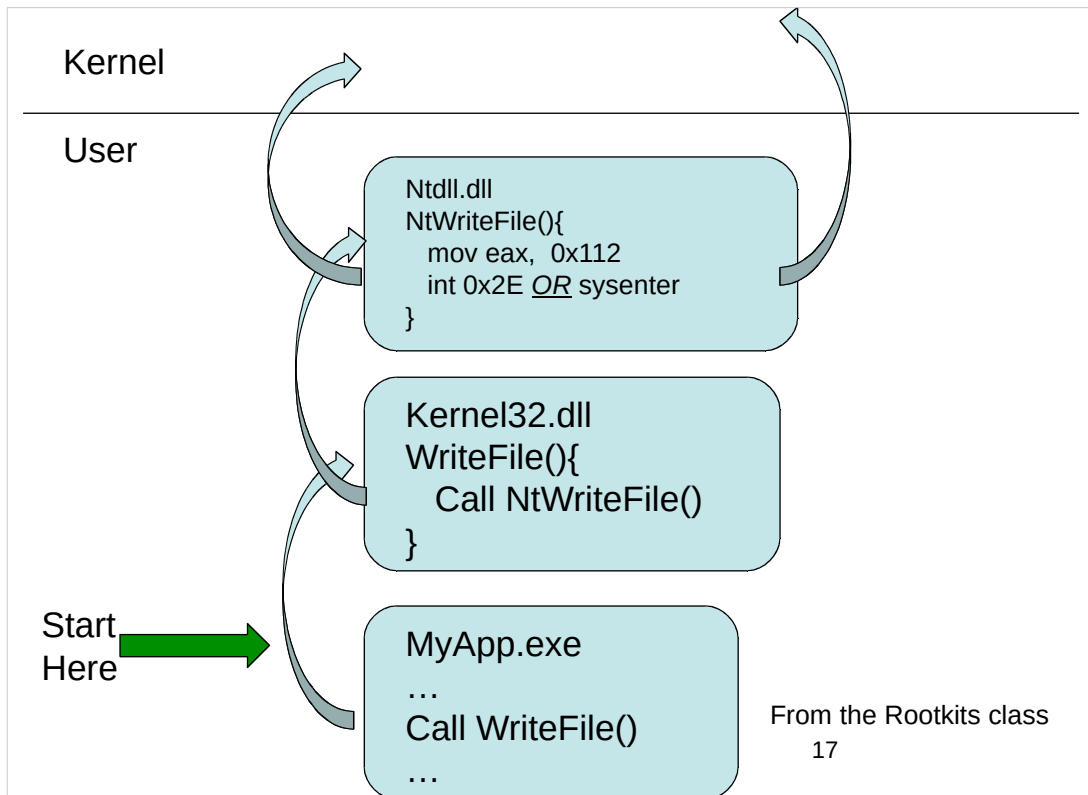
DLL Name	Description
Kernel32.dll	Provides APIs for memory management, file operations, process/thread creation
User32.dll	Implements Windows USER component to provide graphical user interface such as menu bar, scroll bar, button, mouse pointer cursor, etc.
GDI32.dll	Exports Graphics Device Interface functions for drawing, text output, font management, etc.
Ntdll.dll	Interface to kernel for memory management, file operations, process/thread creation. It is not normally used by Windows applications directly
Ws2_32.dll	Exports Windows Sockets APIs
Wininet.dll	Provides high level network API such as HttpOpenRequest and FtpGetFile

See notes for citation

16

[References]

- Michael Sikorski et al., Practical Malware Analysis
- http://en.wikipedia.org/wiki/Microsoft_Windows_library_files
- http://en.wikipedia.org/wiki/Windows_USER



[References]

- Xeno Kovah, <http://opensecuritytraining.info/Rootkits.html>

Process

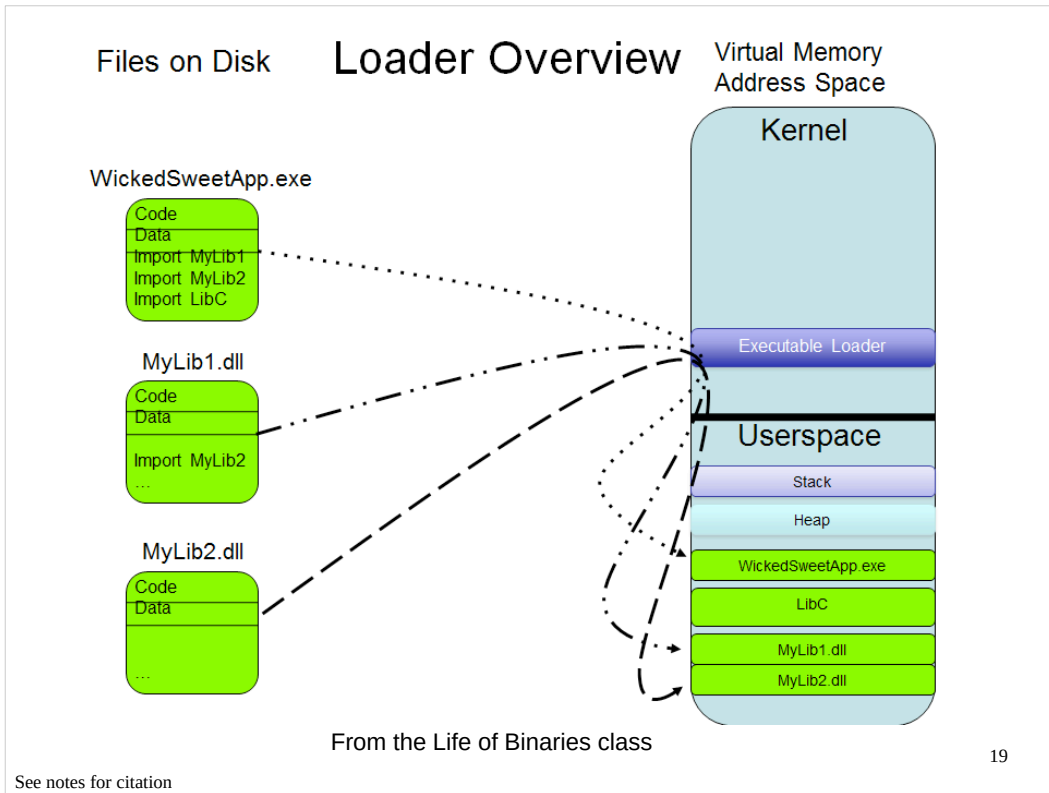
- An instance of program code in execution
 - An executable file itself is not a process
- Each process has own virtual memory address space and executable and library files, stacks, and heap reside on it
- APIs to access to other process's memory
 - ReadProcessMemory, WriteProcessMemory, VirtualAllocEx
- On process context switch, the state of the process and the resources are stored in Process Control Block for resumption later

See notes for citation

18

[References]

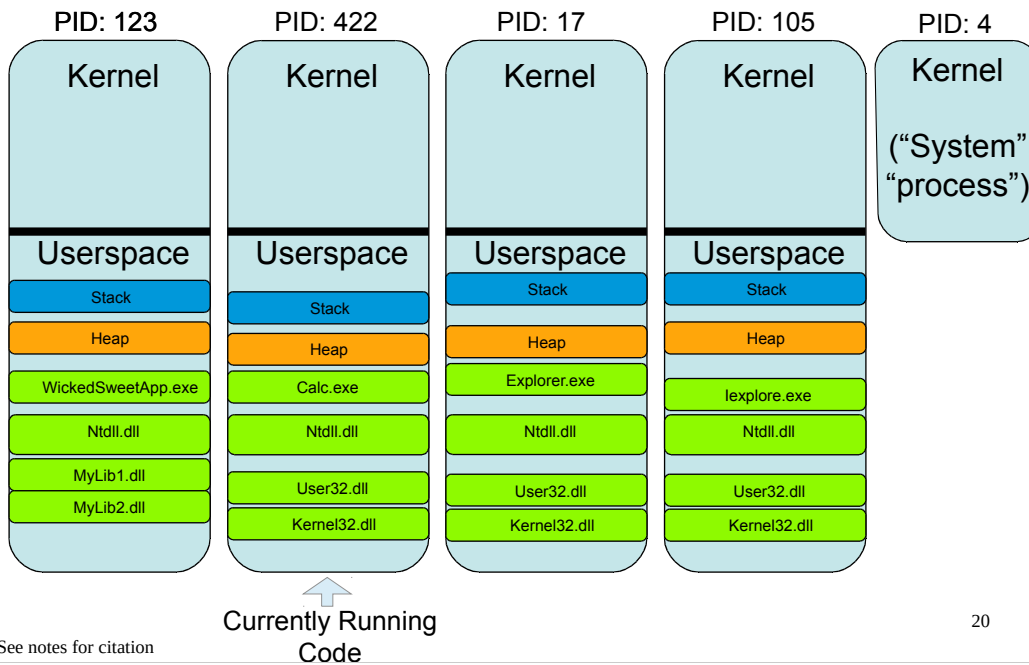
- Silberscharz Galvin, Operating System Concepts 5th Edition



[References]

- Xeno Kovah, <http://opensecuritytraining.info/LifeOfBinaries.html>

Many processes, each with their own view of memory, and the kernel schedules different ones to run at different times





Checking Running Processes

- On the *victim* VM
- Use Task Manager
 - Start → Run... → type “taskmgr”
 - Select “Processes” tab
 - View → Select Columns... → check PID
- Use SysInternals tools (a shortcut key is on the desktop)
 - Process Explorer (procexp.exe)
 - Process Monitor (procmon.exe)
 - Show registry, network, file system activities
- What's the calc.exe's PID and which process is its parent?

See notes for citation

21

[References]

Mark Russinovich, Sysinternals Suite, <http://technet.microsoft.com/en-us/sysinternals/bb842062.aspx>

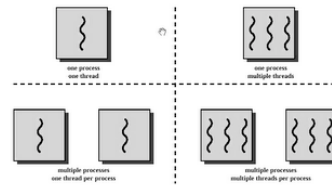


Finding DLL dependencies

- Use CFF Explorer
 - Open c:\Windows\notepad.exe
 - How many DLLs are imported directly by notepad.exe?
- Start notepad.exe
- Use Process Explorer
 - On the menu bar, select
 - View → Show Lower Pane
 - View → Lower Pane View → DLLs
 - How many DLLs are loaded?
- Another good tool: Dependency Walker

Thread

- AKA light weight process who has own program counter (EIP), a register set, and a stack
- Multiple threads can exist in a process and share a process's resources, such as opened file and network connection, concurrently
- Thread context switching is much cheaper than process context switching



See notes for citation

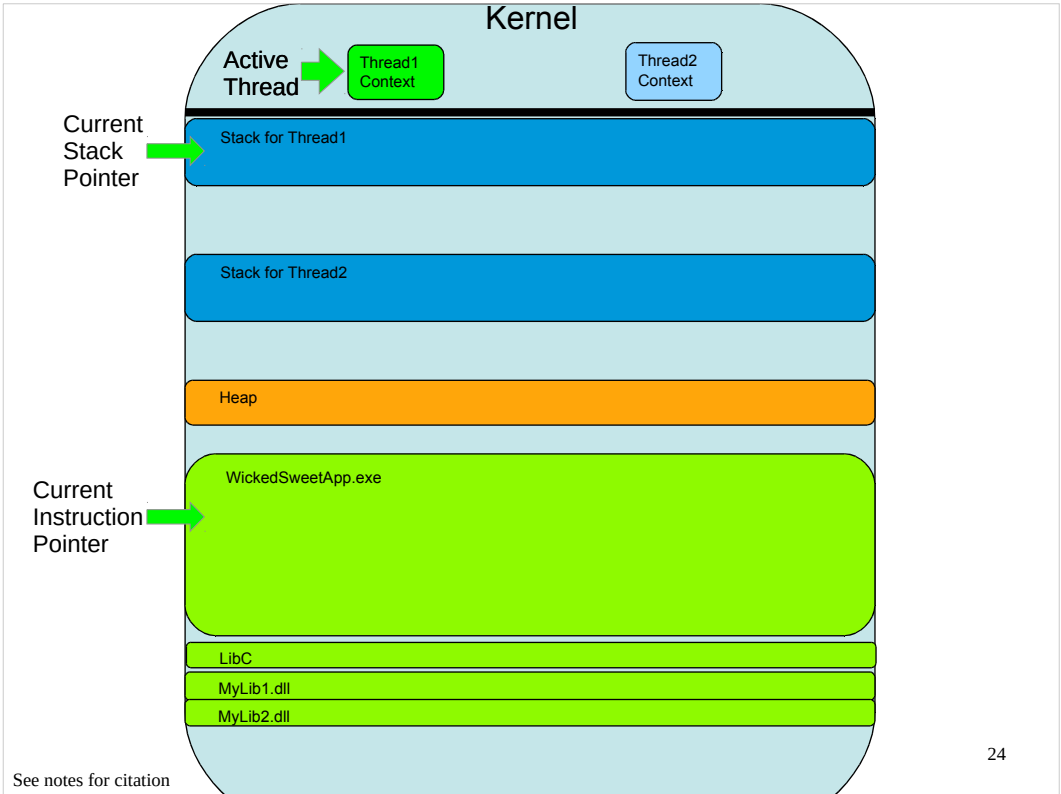
23

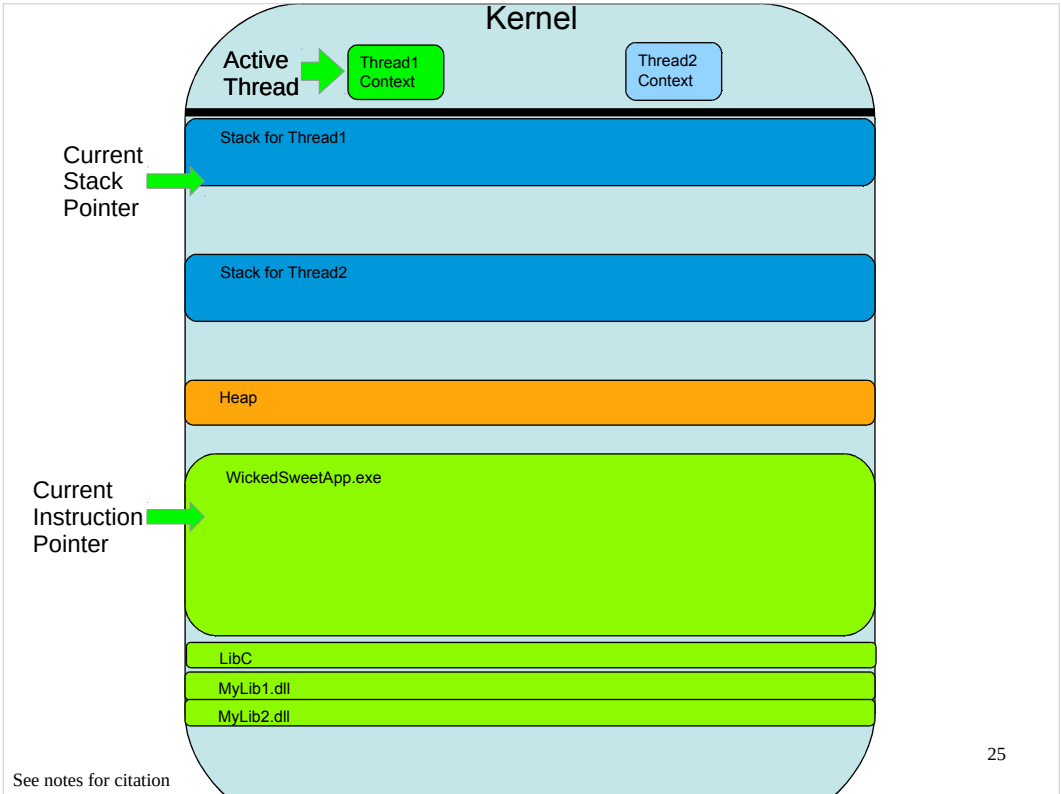
[References]

- Silberscharz Galvin, Operating System Concepts 5th Edition

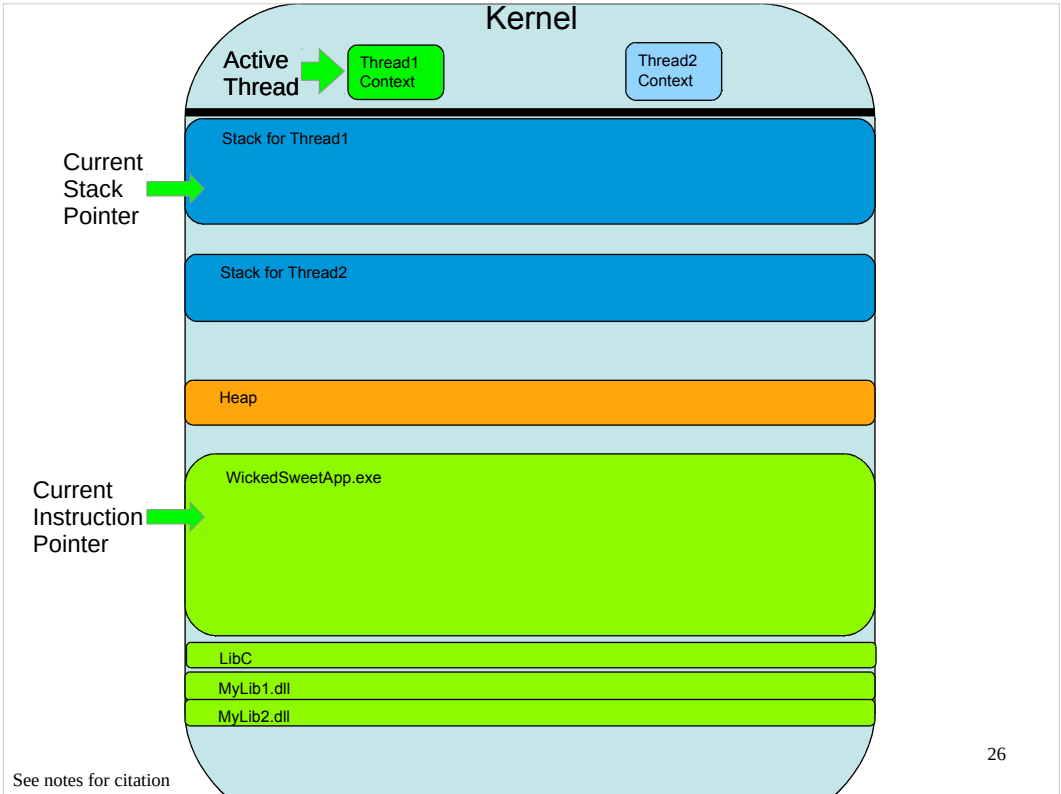
[Image Sources]

- <http://www.cs.cf.ac.uk/Dave/C/mthread.gif>

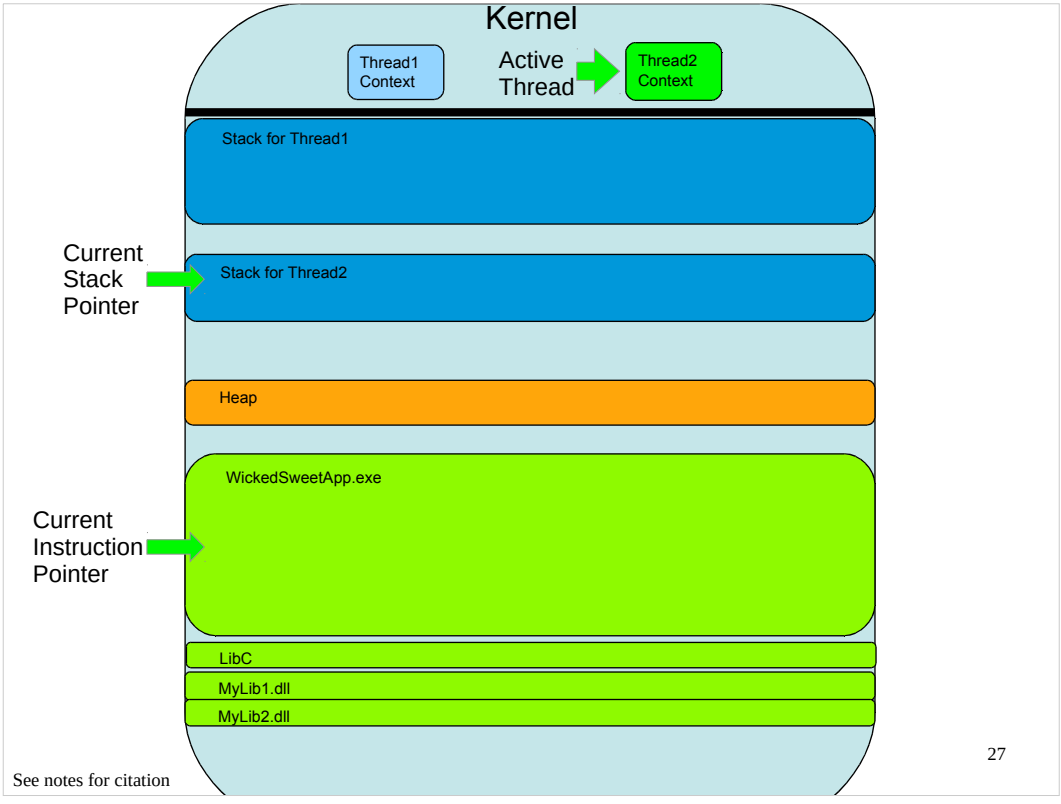




See notes for citation



See notes for citation



See notes for citation

The Registry (1)

- Repository for configuration and control of Windows systems
- Systemwide
 - Which device drivers to load, how to configure memory manager, process manager, etc.
 - Applications read systemwide settings
- Per-user settings
 - Per-user preferences
 - Most-recently accessed documents

See notes for citation

28

[References]

- Mark Russinovich et al., Windows Internals 4th Edition

The Registry (2)

- A key is a container consisting of other keys (subkeys) or values
- Values stores data whose type can be REG_SZ, REG_DWORD, REG_BINARY, etc.

5 Root Key	Stored Information	Link
HKEY_CLASSES_ROOT (HKCR)	File association and Component Object Model (COM) object registration (e.g ProgID and CLSID)	Merged
HKEY_CURRENT_USER (HKCU)	Data associated with the currently logged-on user	Yes
HKEY_LOCAL_MACHINE (HKLM)	Global settings for the machine	No
HKEY_USERS (HKU)	All the accounts on the machine	No
HKEY_CURRENT_CONFIG (HKCC)	Current hardware profile	Yes

See notes for citation

29

[References]

- Mark Russinovich et al., Windows Internals 4th Edition

The Registry (3)

- REG_LINK
 - HKEY_CURRENT_USER is a link to HKEY_USERS\Security ID (SID) of current user
 - HKEY_CURRENT_CONFIG is a link to HKLM\SYSTEM\CurrentControlSet\Hardware Profiles\Current
 - HKLM\SYSTEM\CurrentControlSet is a link to HKLM\SYSTEM\ControlSet00x
- Registry Hive
 - Logical group of keys, subkeys and values in the registry that has a set of supporting files containing backups of its data
 - HKLM\SAM's hive path is c:\windows\system32\config
 - Or constructed dynamically in memory
 - HKLM\HARDWARE is a volatile hive in memory only

See notes for citation

30

[References]

- Mark Russinovich et al., Windows Internals 4th Edition
- Registry Hives, [http://msdn.microsoft.com/en-us/library/windows/desktop/ms724877\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms724877(v=vs.85).aspx)
- Heige Klein, <http://www.sepago.de/d/helge/2008/05/04/free-tool-list-registry-links-reglink>



Checking The Registry

- On the *victim* VM
- Which registry location does HKCU point to?
 - Use Registry Editor (regedit.exe)
 - start → run → regedit
 - Use PsGetSid.exe to get the current user's SID
 - C:\> cd c:\SysinternalSuite
 - C:\> psgetsid.exe student
- Nirsoft's regscanner.exe provides various search options

Microsoft Windows Services

- Long-running executables without user interaction (like a *nix daemon)
- Can be automatically started when the computer boots
- CreateService() Windows API is called to register a service
- Registered services can be found under the registry key
HKLM\System\CurrentControlSet\Services

See notes for citation

32

[References]

- Mark Russinovich et al., Windows Internals 4th Edition

SvcHost

- C:\Windows\System32\svchost.exe is a generic host process for services that run from DLLs
- Multiple instances are often running
 - One instance contains a group of services
- Groups are listed in the registry key
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Svchost
- It is common to have malware name itself svchost.exe but run from somewhere other than C:\Windows\System32, e.g. C:\Windows
- Or alternatively they will just add a new DLL for the real svchost to run as a service

See notes for citation

33

[References]

- Mark Russinovich et al., Windows Internals 4th Edition



Checking Running Services

- On the *victim* VM
- Use Services, a Windows administrative tool
 - On the victim VM, Control Panel → Administrative Tools → Component Services → Services
- Use PsServices.exe, a SysInternals tool
 - C:\> cd c:\SysinternalsSuite
 - C:\> PsService.exe
- Or you can also use a Windows tool, sc.exe
 - C:\> sc query state= all
- Find “Terminal Services” service, what's its status?



Checking SVCHOST Services

- How many svchost.exe instances are running?
 - Use Process Explorer
- List service groups run by svchost.exe by checking the following registry key
 - HKLM\Software\Microsoft\Windows NT\CurrentVersion\Svchost
- Look at the DcomLaunch group – it has two services, “DcomLaunch” and “TermService”
- Check the following registry key to identify services
 - HKLM\System\CurrentControlSet\Services\TermService
- Under the TermService registry key,
 - What is the *ImagePath* value?
 - In the subkey *Parameters*, what's in *ServiceDLL* value?



Checking Normal Services

- Check the following registry key to identify services
 - HKLM\System\CurrentControlSet\Services\CiSvc
- Under the CiSvc registry key.
 - What is the *ImagePath* value?
 - For this service the image path is the executable that's invoked directly
 - The *Start* value determines whether this starts at boot, when the user logs in, or only manually

Network Recap

- Layered architecture



- Common port list
 - HTTP (80), HTTPS (443), DNS (53), SMB (445)
 - <http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml>
- Connection initiator's port is usually randomly picked between 1024 and $2^{16} - 1$
- Common open ports not blocked by firewall
 - DNS (UDP 53): inbound and outbound
 - HTTP (TCP 80, 8080): outbound



Inspecting a Packet Capture

- Wireshark comes with various decoders (e.g. TCP, HTTP and SMB) and presents the network traffic in human readable format for common protocols
- Open misc/sample.pcap using Wireshark
 - `$ wireshark ~/MalwareClass/misc/sample.pcap &`
 - What's the DNS server's IP address?
 - What's the IP, domain name, URL of the website visited?
 - What's the file name a user copied to his/her network share and what kind of file is it?
 - Is there anything suspicious about this file?



Monitoring Network Activity

- Check information about the association between opened ports and processes
- Use TCPView, a SysInternals tool
 - What is listening on port 135?
 - Options → Deselect “Resolve Addresses”
- Use Netstat, a Windows tool
 - C:\>netstat -anob
 - Could you give me more specific answer for the previous question?
- Procmon shows process which is opening a network connection

Outline

- Part 1
 - Background concepts & tools
 - **Observing an isolated malware analysis lab setup**
 - Malware terminology
- Part 2
 - RAT exploration - Poison IVY
 - Persistence techniques
 - Maneuvering techniques
(How malware strategically positions itself)

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.

See notes for citation

41

[References]

- Deep Freeze, <http://www.faronics.com/products/deep-freeze/standard/>
- 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

See notes for citation

42

[References]

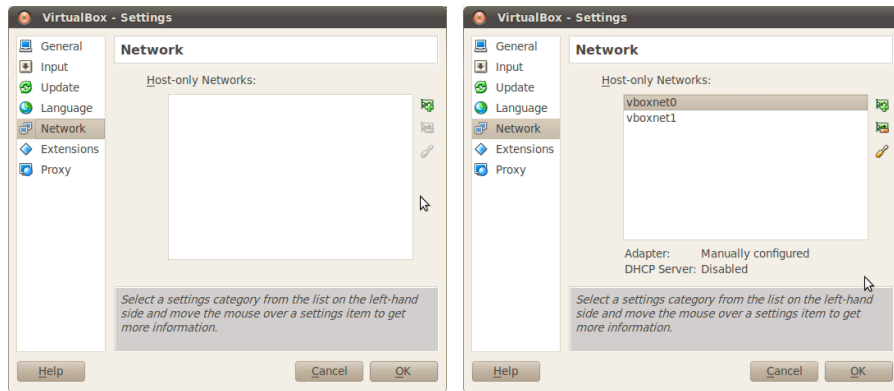
- VirtualBox, <https://www.virtualbox.org/>
- Chapter 6. Virtual networking, <http://www.virtualbox.org/manual/ch06.html>

[Image Sources]

- https://www.virtualbox.org/graphics/vbox_logo2_gradient.png

2 Host-only Networks

- File->Preferences...->Network



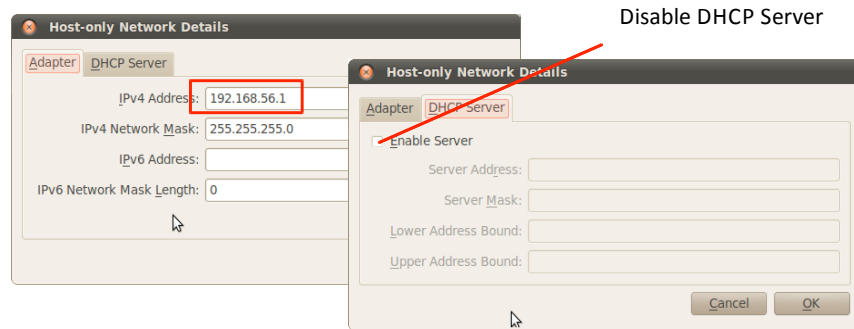
See notes for citation

43

[References]

- Oracle VM VirtualBox User Manual, <http://www.virtualbox.org/manual/UserManual.html>

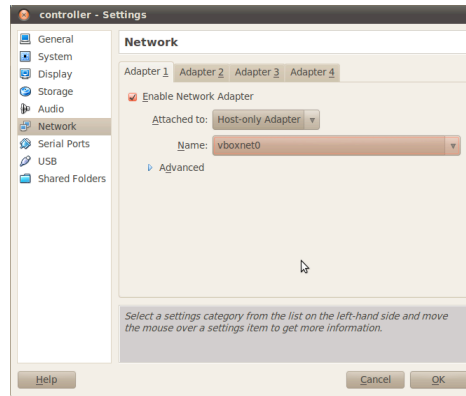
Network Details



- 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



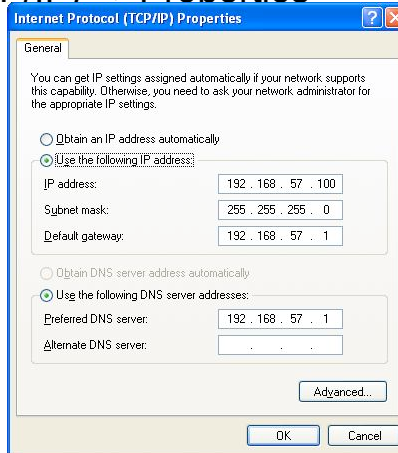
VM's network setting (2)

- Open *victim* VM's Settings → Network → Adapter 1
 - Attached to = 'Host-only Adapter'
 - Name = 'vboxnet1'
- Start VMs and change network setting

VM name	controller	victim
IP address	192.168.56.20	192.168.57.100
Subnet mask	255.255.255.0	255.255.255.0
Default Gateway	192.168.56.1	192.168.57.1
Preferred DNS Server	192.168.56.1	192.168.57.1

Change IP on Windows

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



See notes for citation

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` (installed already)
- Add *wireshark* group
 - `$ sudo groupadd wireshark`
 - `$ sudo usermod -a -G wireshark student`
 - Close all open windows and terminals
 - `$ gnome-session-quit`

See notes for citation

50

[References]

- Jeremy Stretch, <http://packetlife.net/blog/2010/mar/19/sniffing-wireshark-non-root-user/>

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?

INetSim

- 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
 - v.1.2.4 can be installed via a package manager

See notes for citation

54

[References]

- Thomas Hungenberg and Matthias Eckert, <http://www.inetsim.org/>



inetsim (v.1.2.3) 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

# Outline

- Part 1
  - Background concepts & tools
  - Observing an isolated malware analysis lab setup
  - **Malware terminology**
- Part 2
  - RAT exploration - Poison IVY
  - Persistence techniques
  - Maneuvering techniques  
(How malware strategically positions itself)



# 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”

<http://www.microsoft.com/security/portal/Threat/Encyclopedia/Glossary.aspx>

## 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.”

<http://www.microsoft.com/security/portal/Threat/Encyclopedia/Glossary.aspx>

- 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.viruslist.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).”  
<http://www.microsoft.com/security/portal/Threat/Encyclopedia/Glossary.aspx>
- 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](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”  
<http://www.microsoft.com/security/portal/Threat/Encyclopedia/Glossary.aspx>

## 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>

61

# Vendor Naming Convention

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

**Worm:Win32/Taterf.K!dll**



| Vendor    | Name Convection                            | Example                  |
|-----------|--------------------------------------------|--------------------------|
| Symantec  | Prefix.Name.Suffix                         | Infostealer.Banker.C     |
| Avira     | Prefix:Name [Type]                         | Win32:Zbot-BS [Trj]      |
| Kaspersky | [Prefix:]Behaviour.Platform.Name[.Variant] | Trojan.Win32.Genome.taql |

See notes for citation

62

## [References]

- Microsoft Malware Protection Center Naming Standards, <http://www.microsoft.com/security/portal/Shared/MalwareNaming.aspx>
- Virus Naming Conventions, [http://www.symantec.com/security\\_response/virusnaming.jsp](http://www.symantec.com/security_response/virusnaming.jsp)
- Michal Krejdl, What to imagine behind Win32:MalOb [Cryp], <https://blog.avast.com/2009/07/29/what-to-imagine-behind-win32malob-cryp/>
- Rules for naming detected objects, <http://www.securelist.com/en/threats/detect?chapter=136>

# Non-standardized Naming Scheme

| Antivirus     | Result                   | Update   |
|---------------|--------------------------|----------|
| AhnLab-V3     | Win32/Kido.worm.167698   | 20120502 |
| AntiVir       | Worm/Conficker.Z.43      | 20120502 |
| Antiy-AVL     | Worm/Win32.Kido.gen      | 20120503 |
| Avast         | Win32:Rootkit-gen [Rtk]  | 20120502 |
| AVG           | Worm/Downadup            | 20120502 |
| BitDefender   | Worm.Generic.41342       | 20120503 |
| ByteHero      | -                        | 20120502 |
| CAT-QuickHeal | Win32.Worm.Conficker.B.3 | 20120502 |
| ClamAV        | Trojan.Dropper-18535     | 20120503 |
| Commtouch     | W32/Conficker!Generic    | 20120503 |
| Comodo        | NetWorm.Win32.Kido.A     | 20120502 |
| DrWeb         | Win32.HLLW.Shadow.based  | 20120503 |

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