Building a Home Lab for Malware Analysis

SARAH KERN skern@mitre.org

SUSIE HEILMAN sheilman@mitre.org

Approved for Public Release; Distribution Unlimited. Case Number 18-0738

whoami

Sarah Kern

Susie Heilman

Introduction

Malware lab environments can range from complex to very simple

This workshop will walk through how to set up a home malware analysis lab environment for analyzing malware targeted at Microsoft Windows OS's

The entire lab will be built using free and/or open source software

Purpose

Learn how to build a home analysis lab for fun and educational purposes

- Great way to learn about malware by playing with actual samples in a controlled environment

Provide you with a basic setup where you can gain experience with malware analysis tools at home

- Even analysis of known-good software can assist in gaining the skills and understanding of the tools
- More advanced analysis will require additional resources and precautions

What This Workshop IS NOT

Focus is not on malware detection or hunting

Not an extensive review of malware and everything it can do

Does not cover tactics and techniques for reverse engineering malware and code analysis

We will not distribute malware nor do we condone public spreading or distribution of malware

OK, Why Though?

Can’t we just use automated services???

Important to learn how to configure and to maintain your own environment

- Become versed in the various tools available

Automated services must be verified

- They can be easily bypassed
- They are only as smart as the person implementing them

Malware sample may be sensitive and you cannot share it with automated services

- In this case, you would need to know how to run the tools without the help of online services

Before We Get Started…

Download the following:

- Windows VM
- Ubuntu VM
- pestudio
- Process Monitor
- Process Explorer
- Regshot

Terminology

What is Malware Analysis?

What is Malware Analysis?

Malware analysis is the art of dissecting malware to understand how it works, how to identify it, and how to defeat or eliminate it.

– Practical Malware Analysis book

Goals Of Malware Analysis

Determine what the malware can do

Develop signatures for detecting malware on your system/network

Find out if it has spread

Contain the damage

Types of Malware

Types of Malware

- Backdoor
- Botnet
- Downloader
- Information stealing malware (keyloggers, password grabbers)
- Launcher
- Rootkit
- Scareware
- Spam-sending malware
- Virus or worm…to name a few

Malware Analysis Techniques

Fully-automated Analysis

- “Quick and dirty” approach
- Example: Cuckoo Sandbox

Static Analysis

- Gathering static properties for basic indicators of compromise

Dynamic Analysis

- Interactive behavioral analysis, actually executing the malware & observing it

Reverse Engineering Code

- Manually reversing the code
- Time consuming and intricate

Malware Analysis Techniques Covered Today

Static Analysis

Dynamic Analysis

Groundwork

Tips For Malware Analysis Beginners

Don’t get stuck in the details

There is no single approach that fits all cases

- If you get stuck, switch to a different tool or approach it from a different angle

Setting Up Shop

Setting up a safe environment is important so that your host computer is not compromised

Just because you are analyzing malware in a sandboxed environment, does not mean you are completely secure

Define Goals, Define Environment

What exactly are you are looking for?

Why are you doing this?

Once you have analyzed the malware specimen, what do you plan to do with the information extracted?

What is your end goal?

Define Goals, Define Environment

What exactly are you are looking for?

- lateral movement, data exfil, etc.

Why are you doing this?

- learning opportunity, protect a company, etc.

Once you have analyzed the malware specimen, what do you plan to do with the information extracted?

- educational report, blog post, create network signatures

What is your end goal?

- Example: contain it from spreading

Analyzing Malware Option 1: Physical

Malware can be safely analyzed on physical, air-gapped machines

- Disadvantages:
  1. No Internet connection (many pieces of malware depend on Internet connection for updates, C&C, etc.)
  2. Malware can be difficult to remove

Main advantage

- Malware sometimes executes differently in VMs

Analyzing Malware Option 2: Virtual

Advantages:

1. Easy to revert machines after running malware – clean snapshots
2. Offers rapid OS deployment
3. Advanced networking options – easy to isolate infected hosts
4. Standardized hardware

Downfalls/Challenges:

- Some malware can detect that it is being run in a VM and behave differently

Virtual machines are most commonly used for dynamic analysis because of the disadvantages and risks of using physical machines

Workshop Environment

Virtual target:

- Windows-based VM
- Location malware will be executed (may be useful to have multiple versions of windows)
- Can be reverted back to pre-infection state

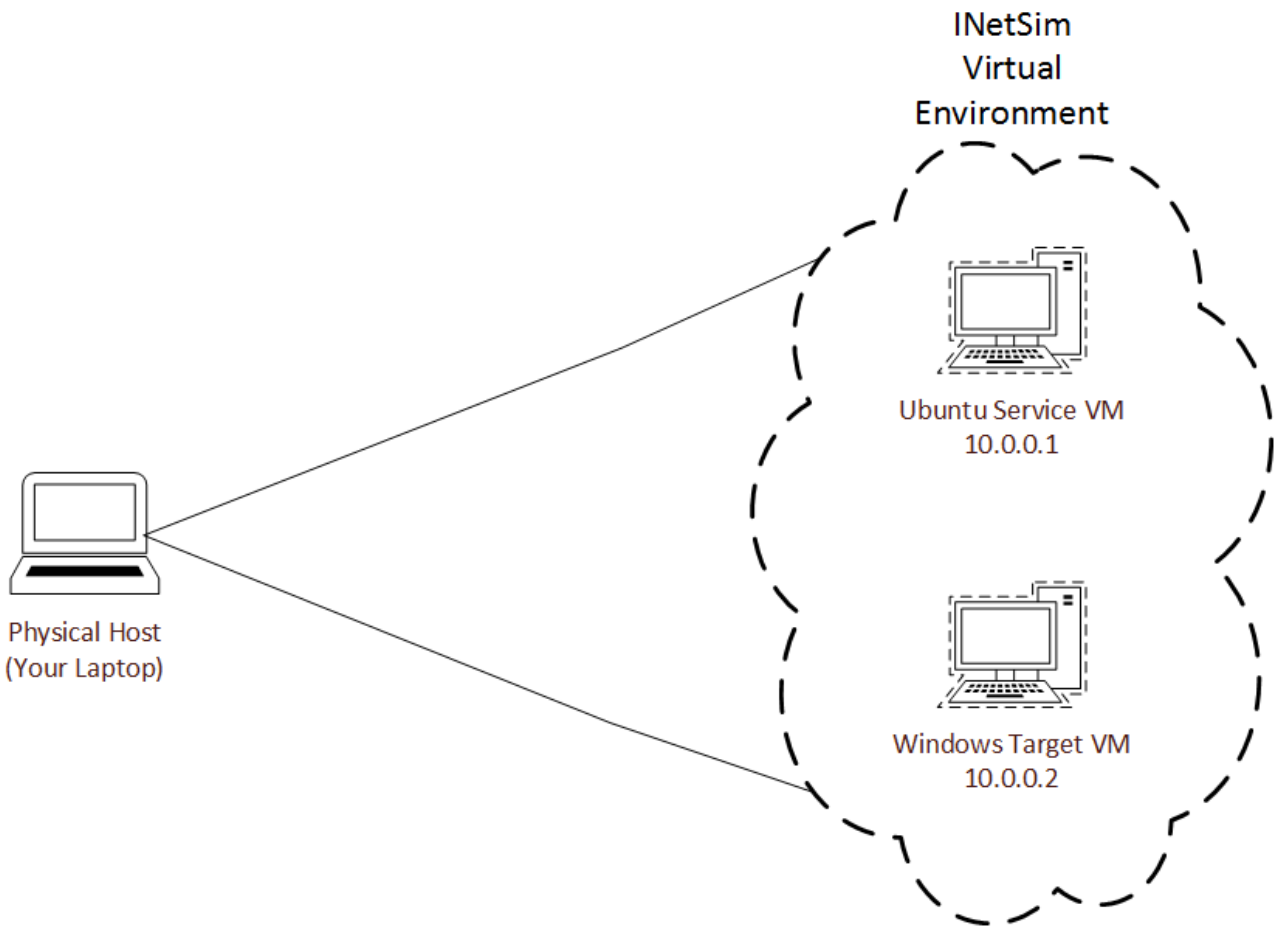
Virtual service machine:

- Ubuntu VM
- Houses the services and virtual network that malware connects to

Host/Controller:

- Physical computer that runs virtualization software to control the virtual targets
- It is recommended to use a Linux based system as the controller's OS, but not required
- This is your physical computer

Workshop Environment Diagram



Let’s Get Started!

Time to set up your own home malware analysis lab!

Slight Disclaimer

A lot of it will come down to personal preference, there are many ways to go about setting up a personal lab

We will cover one possible base setup

List of Steps

1. Install Hypervisor
2. Install VMs & Guest Additions
3. Download/Install Analysis Tools
4. Configure Networking
5. Analysis Precautions
6. Create Shared Folder
7. Create Snapshots
8. Transfer Files

Step 1 – Install Hypervisor

Acquire VMs:

- Windows VM: https://goo.gl/X7n6XK
- Ubuntu machine

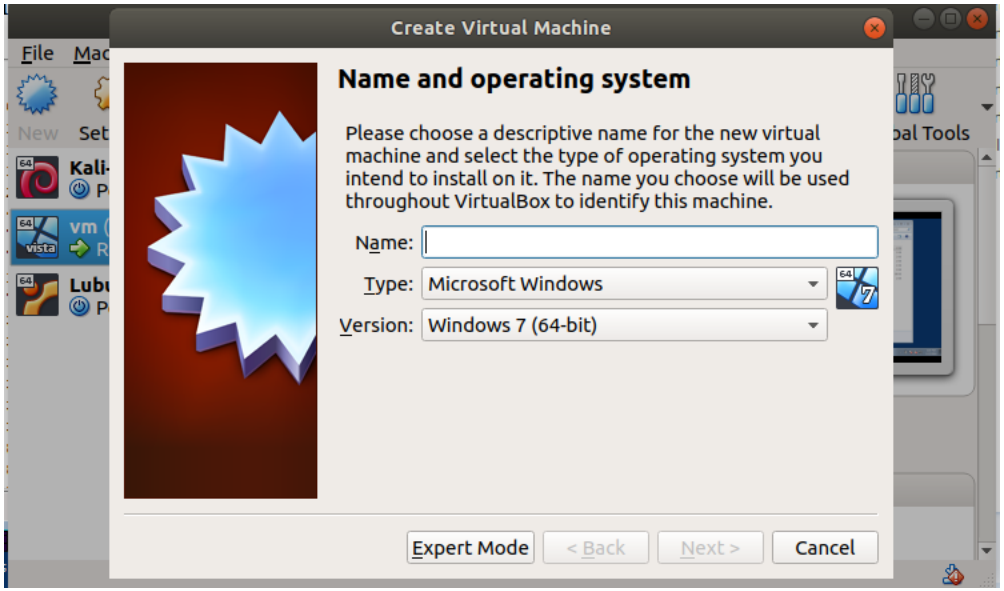
Install hypervisor:

- Recommend VirtualBox or VMWare Player as free virtualization products
- VirtualBox has snapshot feature that is especially useful
- There are many paid for versions that work well too

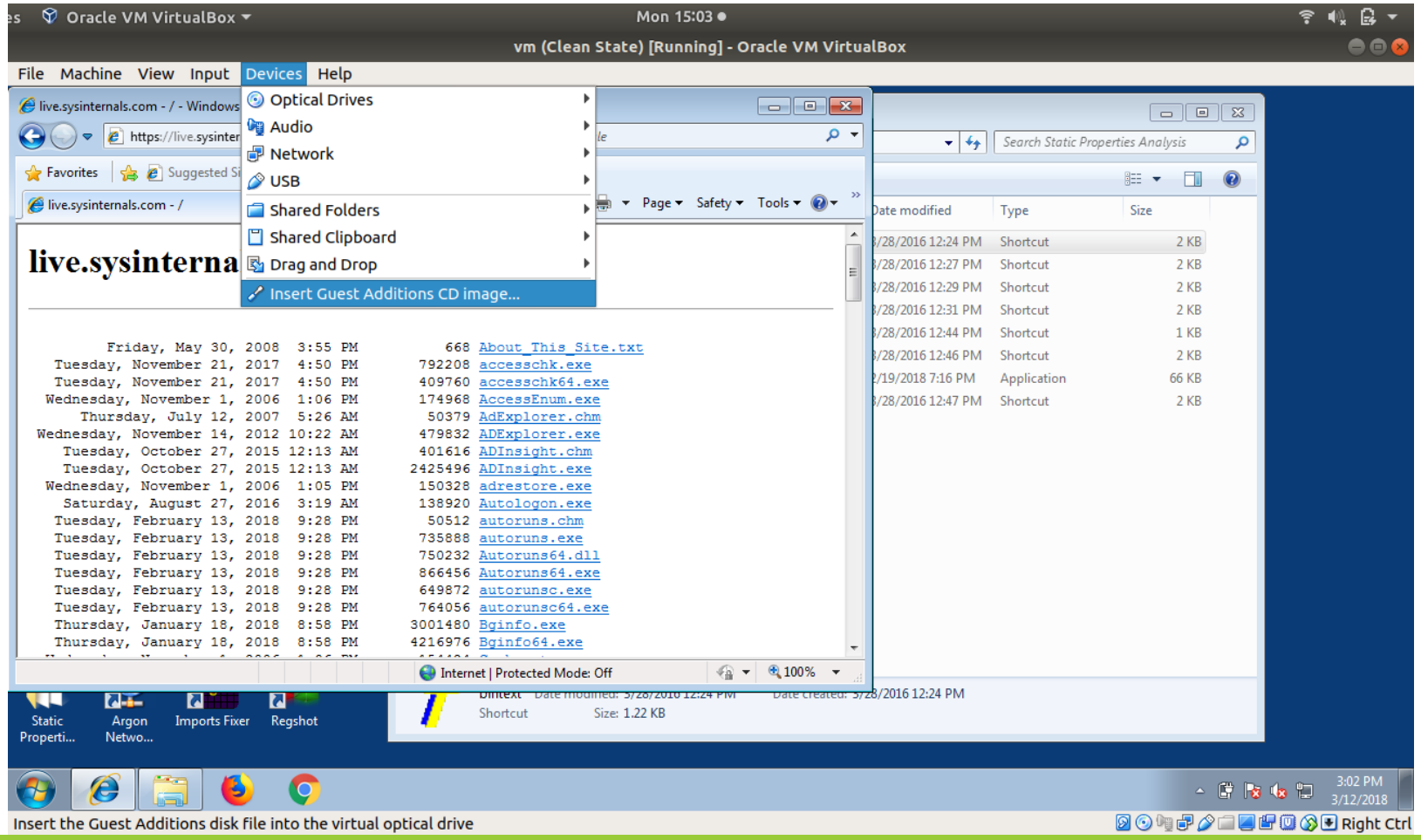
Step 2 – Install VMs & VM Tools

Install your VMs and install Guest Additions on each VM

- Let's walk through this together



Guest Additions – Windows Target VM



Guest Additions – Ubuntu Service VM

Install necessary packages

```
$ sudo apt-get install make gcc linux-headers-$(uname -r) virtualbox-guest-dkms linux-headers-virtual
```

Insert and mount Guest Additions CD, then run it

```
$ sudo mount /dev/cdrom /media/cdrom
$ sudo /media/cdrom/VBoxLinuxAdditions.run (may need to reboot first)
$ sudo usermod -a -G vboxsf 'username'
$ sudo reboot
```

Step 3 – Download & Install Analysis Tools

Download analysis tools for Windows VM

- PEStudio
- Regshot
- Process Explorer
- Process Monitor

Download service tools for Ubuntu VM

- INetSim
- Burp Suite

Optional: download Windows applications that are popular targets (Firefox, Chrome, Adobe Reader, Adobe Flash, Skype, etc.)

Recommended Base Toolset for Windows

- PEStudio or PEView
- Regshot
- Process Explorer
- Process Monitor
- Wireshark
- CFF Explorer
- Fakenet/ApateDNS
- Any hex tool (HxD, Frhed, Cygnus, etc.)
- Resource Hacker
- 7zip
- Create bookmark to https://live.sysinternals.com

Even More Tools

More extensive lists of well-known malware analysis tools can be found at:

https://github.com/rshipp/awesome-malware-analysis

https://github.com/fireeye/flare-vm

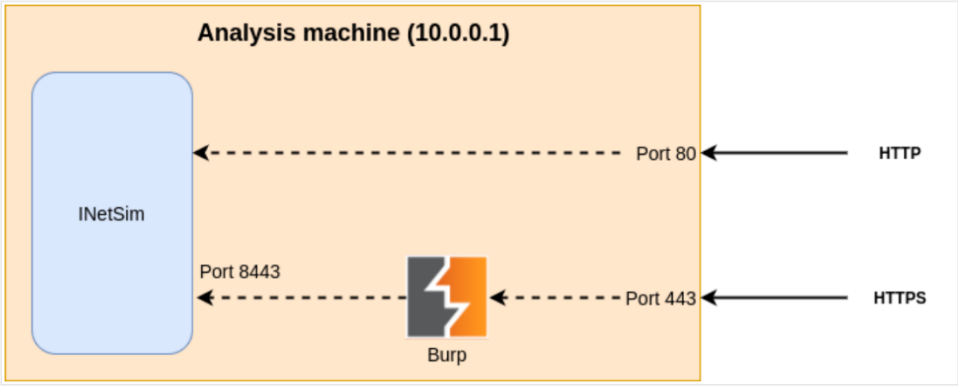
Recommended Tools for Ubuntu

INetSim

- Simulates standard Internet services such as DNS, HTTP and SMTP

Burp Suite

- Provides support for SSL communication
- Only really need it if you want to intercept SSL comms



Taken from blog: https://christophetd.fr/

INetSim Setup

Install INetSim as root

```
$ sudo su
$ echo "deb http://www.inetsim.org/debian/ binary/" > /etc/apt/sources.list.d/inetsim.list
$ wget -O - http://www.inetsim.org/inetsim-archive-signing-key.asc | apt-key add -
$ apt update
$ apt install inetsim
```

Continue setup as regular user

```
$ mkdir analysis/test_nw
$ cp /etc/inetsim/inetsim.conf analysis/test_nw
$ sudo cp -r /var/lib/inetsim analysis/test_nw/data
$ cd analysis/test_nw
$ sudo chmod -R 777 data
```

INetSim Setup (continued)

Edit the file analysis/test_nw/inetsim.conf by changing the following lines:

```
service_bind_address    0.0.0.0
dns_default_ip    10.0.0.1
https_bind_port 8443
```

Stop the default Ubuntu DNS Server

```
$ sudo systemctl disable systemd-resolved.service
$ sudo service systemd-resolved stop
```

Burp Suite Setup (Optional)

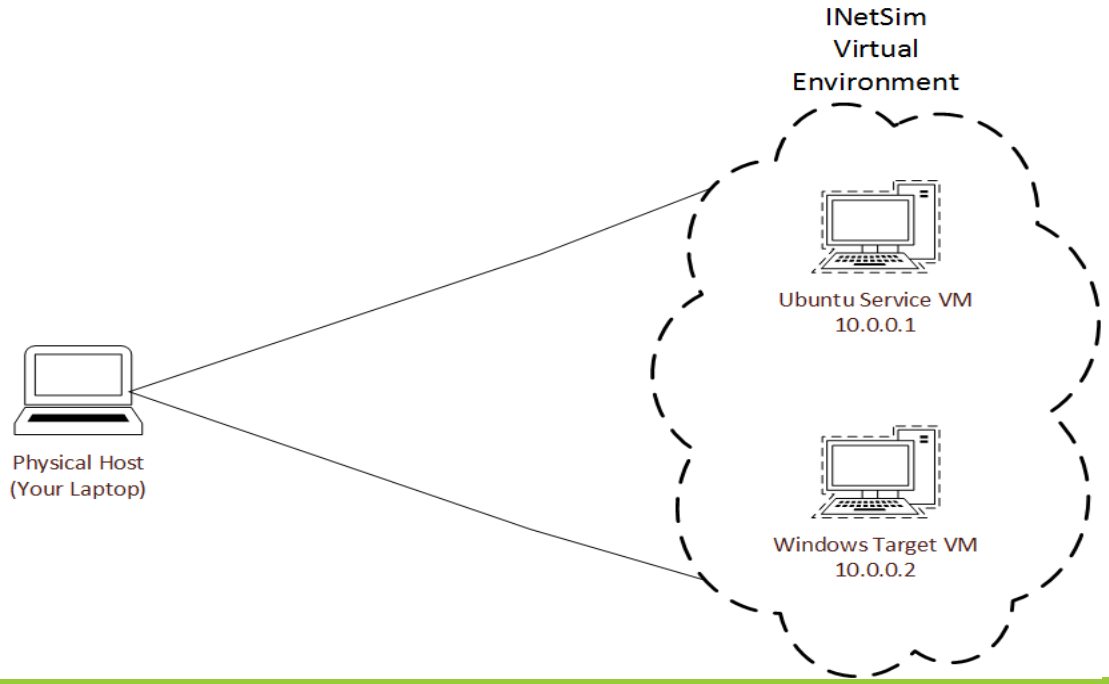
Download Burp from:
https://portswigger.net/burp/communitydownload

Install

```
jane@lubuntu:~$ bash ~/Downloads/burpsuite_community_linux_v1_7_32.sh
Unpacking JRE ...
Starting Installer ...
jane@lubuntu:~$
```

Step 4 – Configure Networking

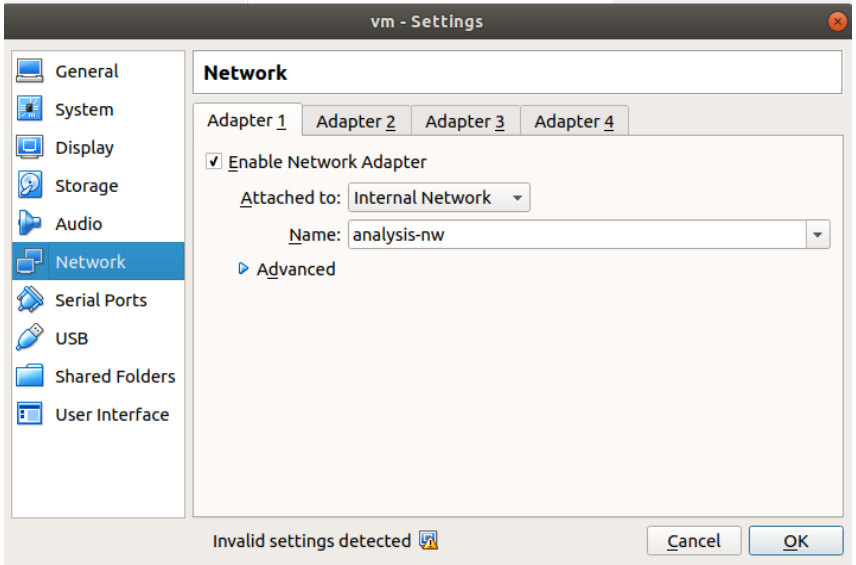
We will now create a private, segmented network so that our VMs can communicate with each other, while preventing them from reaching out to the external Internet



Virtual Box Network Configuration

For each VM:

1. Open Settings
2. Go to Network tab
3. Create Internal Network
4. Disable all other Adapters
   - Avoids cross-contamination



Ubuntu Network Configuration

Run ifconfig

- Do you see enp0s3 or similar?

If not, open /etc/network/interfaces

- Append to end of file

```
auto enp0s3
iface enp0s3 inet static
  address 10.0.0.1
  netmask 255.255.255.0
```

Bring the network up

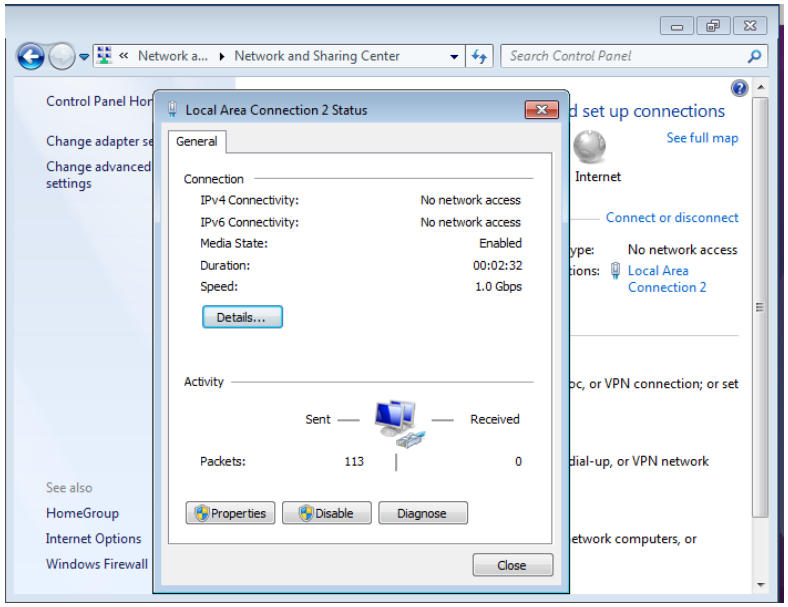
```
jane@lubuntu:~$ sudo su
root@lubuntu:/home/jane# vim /etc/network/interfaces
root@lubuntu:/home/jane# ifup enp0s3
root@lubuntu:/home/jane#
```

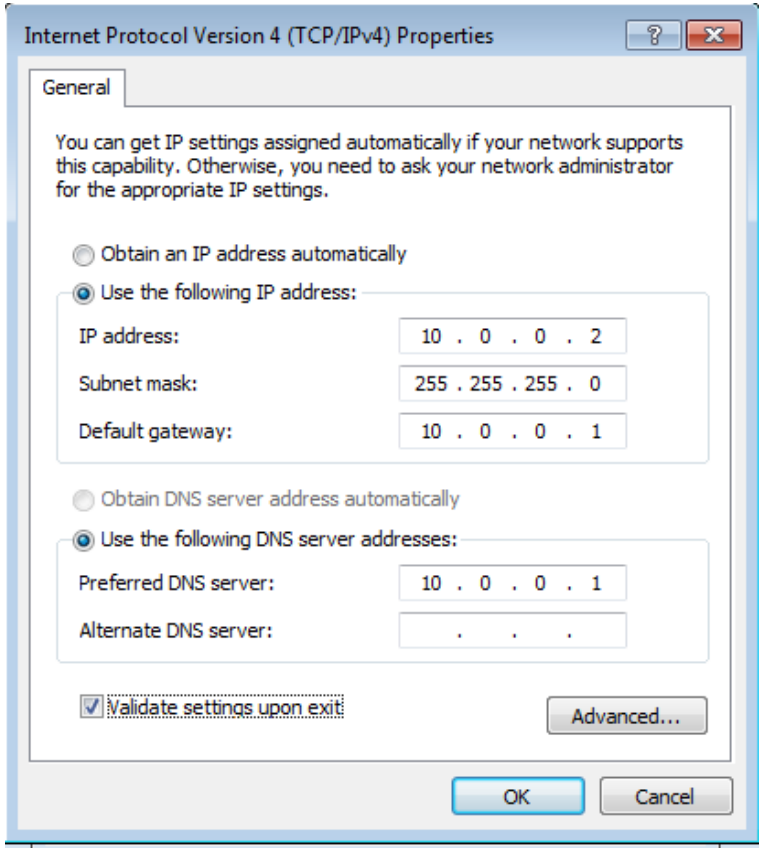
Should Look Like This

```
jane@lubuntu:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.1  netmask 255.255.255.0  broadcast 10.0.0.255
        inet6 fe80::a00:27ff:fe5e:a775  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:5e:a7:75  txqueuelen 1000  (Ethernet)
        RX packets 2007  bytes 158032 (158.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1871  bytes 185623 (185.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Windows Network Configuration

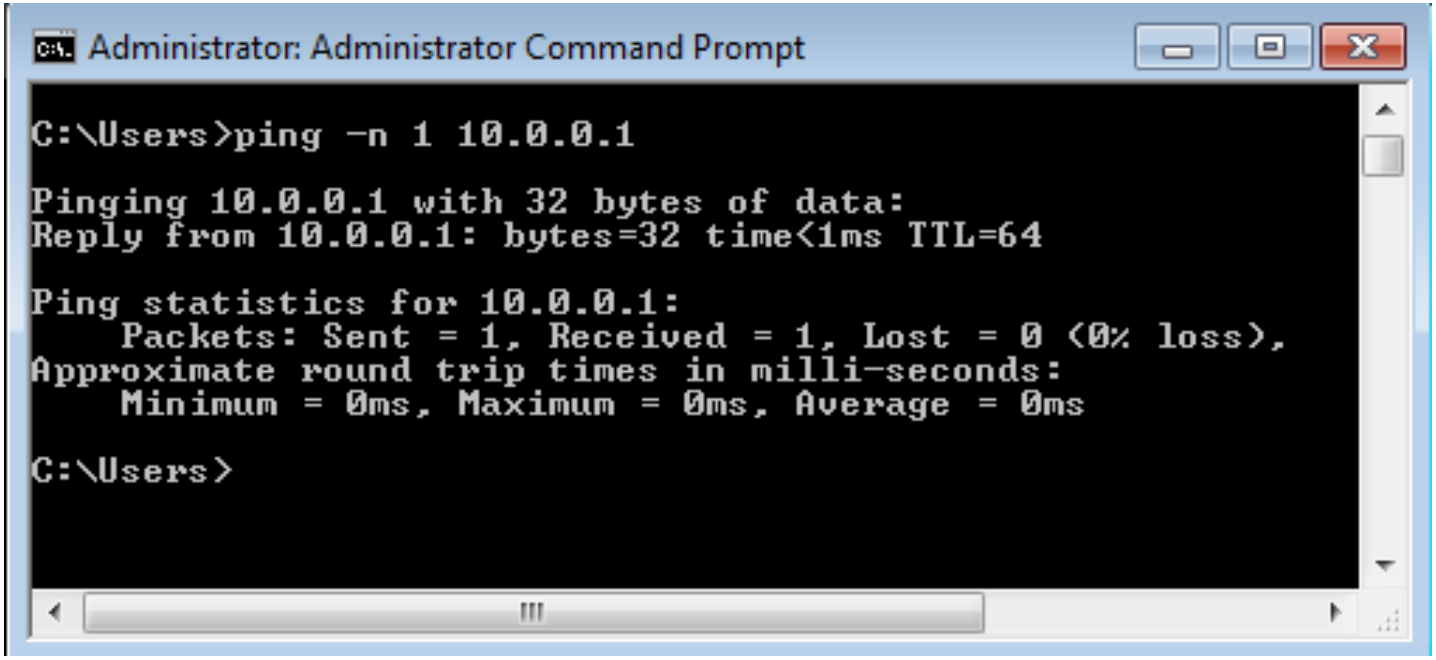
- Control Panel > Network and Internet > Network and Sharing Center
- Local Area Connection 2 > Properties
- Select IPV4 > Properties
- Fill in static IP





Test Network Connections

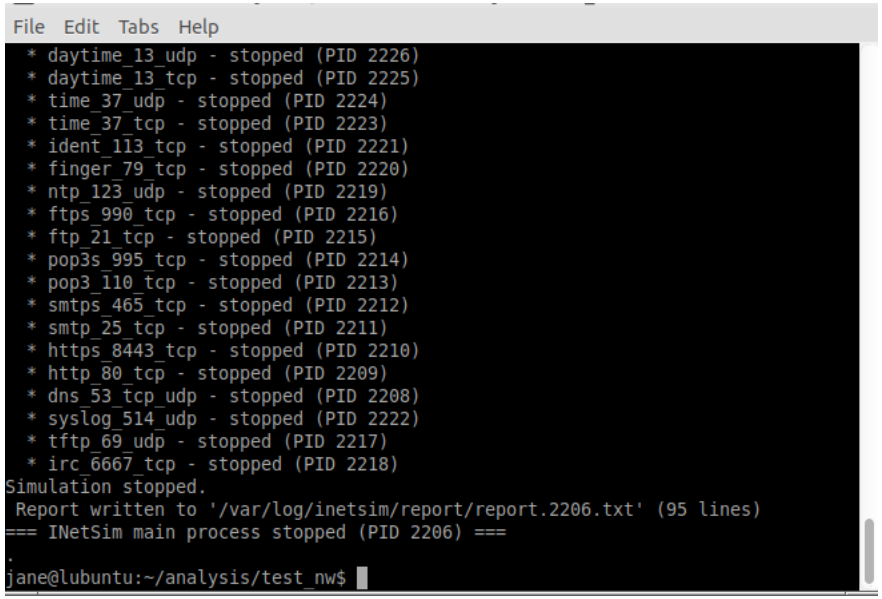
Ping 10.0.0.1 from Windows cmd



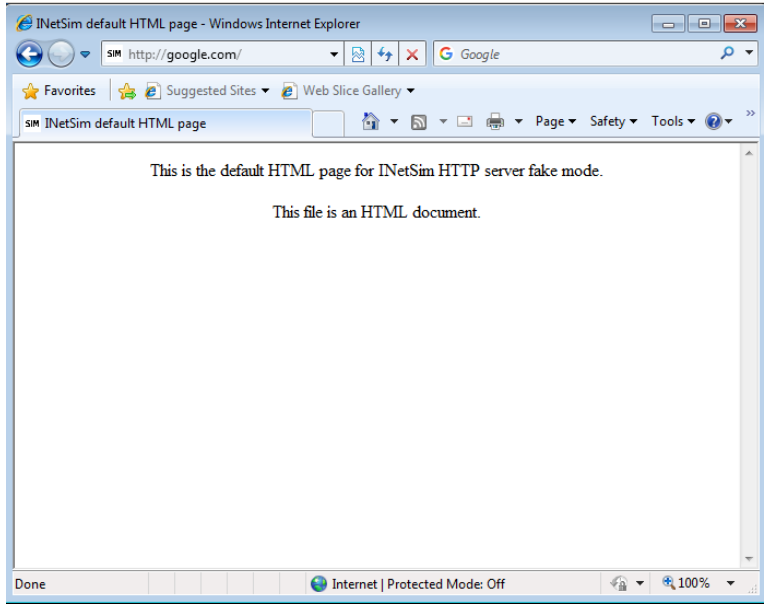
Test Network Connections Part 2

1. Run INetSim on Ubuntu from analysis/test_nw directory

sudo inetsim --data data --conf inetsim.conf



2. Navigate to website on Windows



Step 5 – Precautions

Disconnect Windows VM peripherals so that malware cannot escape the VM

1. Make sure VM is powered off
2. Open Settings in VirtualBox
3. Disable USB Controller
4. Unmount any Shared Folders
5. Double check Network Adapters

Disconnect Windows VM peripherals so that malware cannot escape the VM

Step 6 – Create Shared Folder on Ubuntu

Power off Ubuntu VM

Create Shared Folder

Step 7 - Snapshots

Create snapshots for both VMs

Step 8 – Transfer Files to Victim VM

Edit inetsim.conf file again

Browse to website on Windows Victim VM

Caveats

Desktop virtualization products have vulnerabilities too and must be kept up-to-date

Analysis tools will have to be updated over time (new versions, improved tools, etc.)

- Update tools and rebase your snapshot periodically

Some risk is always present

- Do not perform analysis on critical or sensitive computers
- Do not perform analysis on shared computers or networks (e.g. university computers and networks)

Walkthrough/Demo

Static Analysis

Analysis that's done *without* executing the binary

- Dynamic analysis executes binary

Techniques

- Run sample through an Antivirus tool
- Use hash to identify malware
- Assemble information from the sample's headers, functions, and strings

Let's walk through an example

Download Samples At Your Own Risk

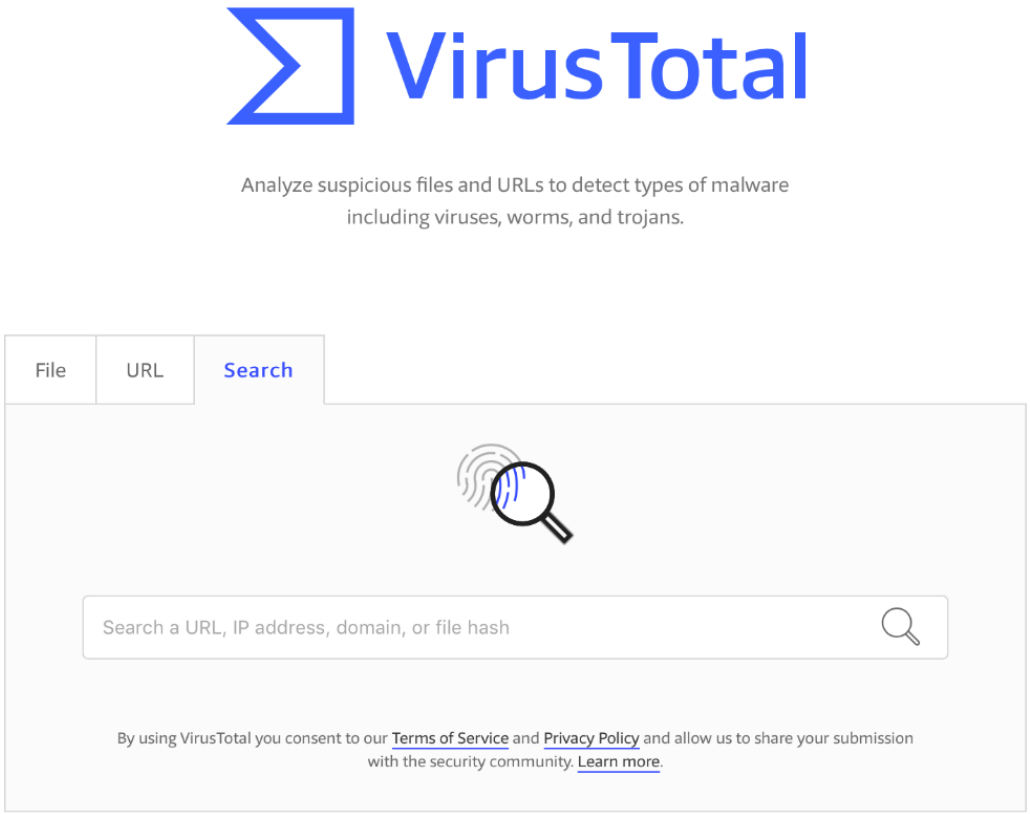
Download Practical Malware Analysis labs

- https://github.com/mikesiko/PracticalMalwareAnalysis-Labs

Alternatively, analyze a program you already have

- C:\Program Files (x86)\Internet Explorer
  - Open any DLL in pestudio
- Analyzing *known good* software is also beneficial

Anti-Virus Scan and VirusTotal Reports



VirusTotal (continued)

Search or scan a URL, IP address, domain, or file hash

Sign in

40 / 67

40 engines detected this file

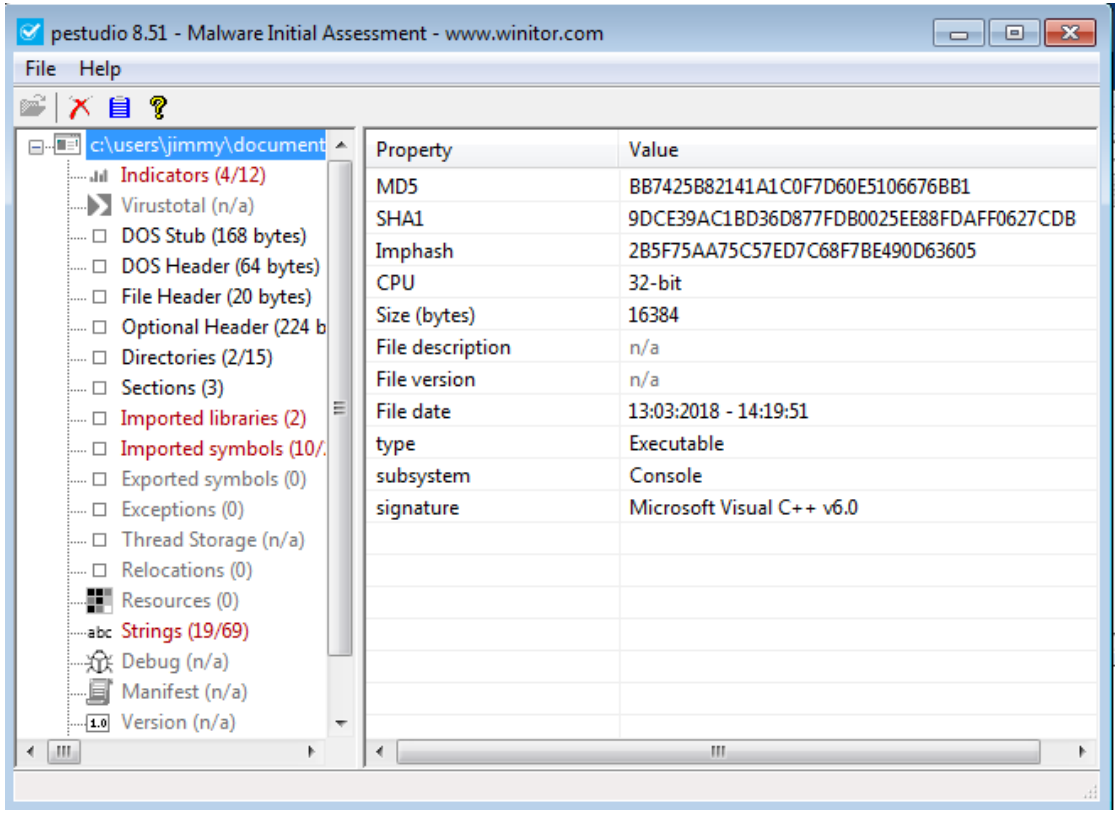
SHA-256	58898bd42c5bd3bf9b1389f0eee5b39cd59180e8370eb9ea838a0b327bd6fe47
File name	Lab01-01.exe
File size	16 KB
Last analysis	2018-03-12 11:51:05 UTC
Community score	-9

Detection | Details | Relations | Behavior | Community 9

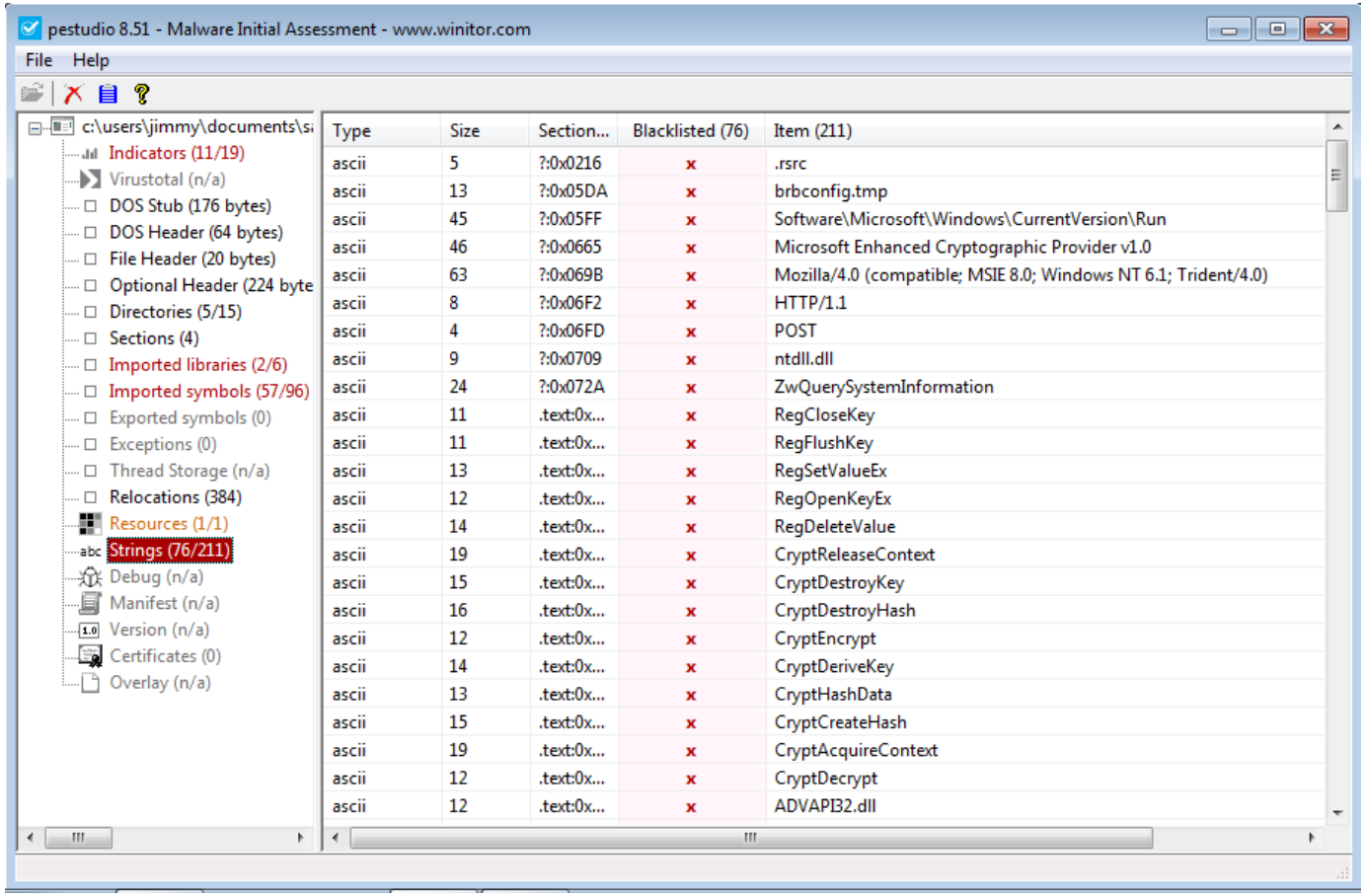
AegisLab	Troj.Rogue.Gen!c	AhnLab-V3	Trojan/Win32.Agent.C957604
ALYac	Trojan.Agent.16384SS	Antiy-AVL	Trojan/Win32.TSGeneric
Avast	Win32:Malware-gen	AVG	Win32:Malware-gen
Avira	TR/Rogue.11196274	AVware	Trojan.Win32.Generic!BT
Baidu	Win32.Trojan.WisdomEyes.16070401....	CAT-QuickHeal	Trojan.IGENERIC
ClamAV	Win.Malware.Agent-6342616-0	Comodo	.UnclassifiedMalware
CrowdStrike Falcon	malicious_confidence_60% (W)	Cylance	Unsafe
Cyren	W32/Trojan.CZAN-7287	eGambit	Unsafe.AI_Score_96%
Endgame	malicious (high confidence)	ESET-NOD32	a variant of Win32/Agent.WOM
Fortinet	W32/Agent.WOM!tr	GData	Win32.Trojan.Agent.RE19WZ

Get the File's Hash

Open the file in pestudio



Let’s Get Some Strings



Type	Size	Section...	Blacklisted (76)	Item (211)
ascii	5	?:0x0216	x	.rsrc
ascii	13	?:0x05DA	x	brbconfig.tmp
ascii	45	?:0x05FF	x	Software\Microsoft\Windows\CurrentVersion\Run
ascii	46	?:0x0665	x	Microsoft Enhanced Cryptographic Provider v1.0
ascii	63	?:0x069B	x	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0)
ascii	8	?:0x06F2	x	HTTP/1.1
ascii	4	?:0x06FD	x	POST
ascii	9	?:0x0709	x	ntdll.dll
ascii	24	?:0x072A	x	ZwQuerySystemInformation
ascii	11	.text:0x...	x	RegCloseKey
ascii	11	.text:0x...	x	RegFlushKey
ascii	13	.text:0x...	x	RegSetValueEx
ascii	12	.text:0x...	x	RegOpenKeyEx
ascii	14	.text:0x...	x	RegDeleteValue
ascii	19	.text:0x...	x	CryptReleaseContext
ascii	15	.text:0x...	x	CryptDestroyKey
ascii	16	.text:0x...	x	CryptDestroyHash
ascii	12	.text:0x...	x	CryptEncrypt
ascii	14	.text:0x...	x	CryptDeriveKey
ascii	13	.text:0x...	x	CryptHashData
ascii	15	.text:0x...	x	CryptCreateHash
ascii	19	.text:0x...	x	CryptAcquireContext
ascii	12	.text:0x...	x	CryptDecrypt
ascii	12	.text:0x...	x	ADVAPI32.dll

Something to Keep in Mind

Packed or obfuscated code

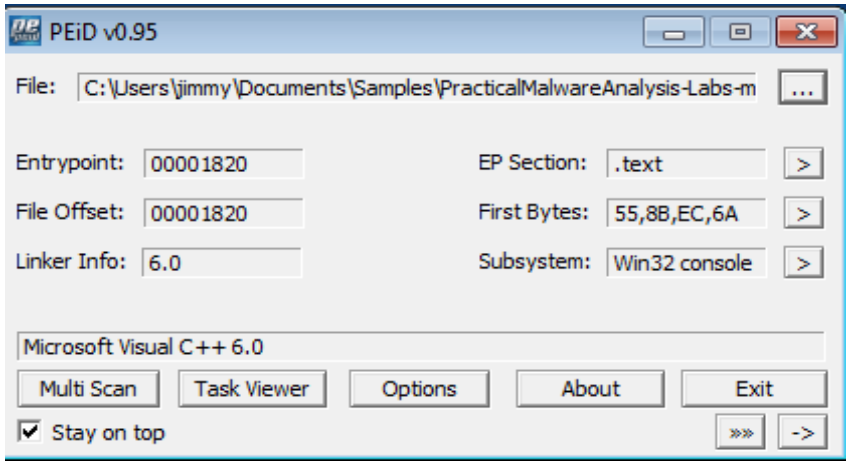
- *Obfuscated*: Author has intentionally tried to hide the code functionality, or
- *Packed*: The program has been compressed in some way that hinders you from analyzing it

Signs

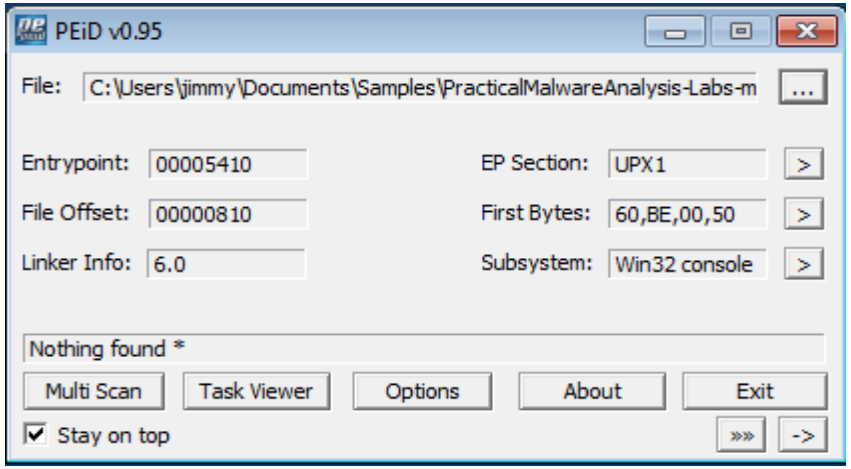
- Little to no strings
- Open it in PEiD
- Will see the functions *LoadLibrary* and *GetProcAddress*

Comparison in PEiD

Not packed



Packed



Additional Artifacts

- Filename
- File creation date
- Compile date
- File size (useful for comparison to other samples)
- File appearance (see anything suspicious?)
- File Type
  - Portable executable
  - Powershell
  - Etc.
- Imported Libraries
- Imported Symbols
- Resources

Static Analysis ✓

This should be the first step in the inspection of any file

It can provide insight into the authors of the malware, the timeline, etc.

Next step is dynamic analysis

Dynamic Analysis

Running malware and observing its behavior

- aka Behavioral Analysis

Techniques

- Running malware
- Monitoring processes (Process Monitor)
- Viewing processes (Process Explorer)
- Observing registry changes (Regshot)
- Faking a network (INetSim)
- Capturing & analyzing network traffic

Demo time!

Dynamic Analysis ✓

A way to confirm static analysis results

Acquire a better understanding of the functionality and purpose of the malware

Moving Forward

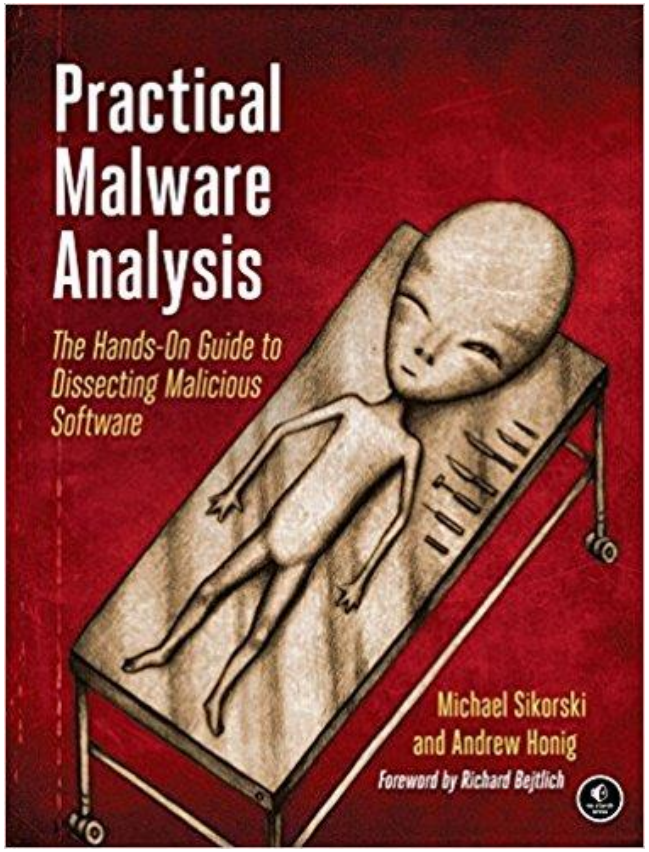
Expand your lab:

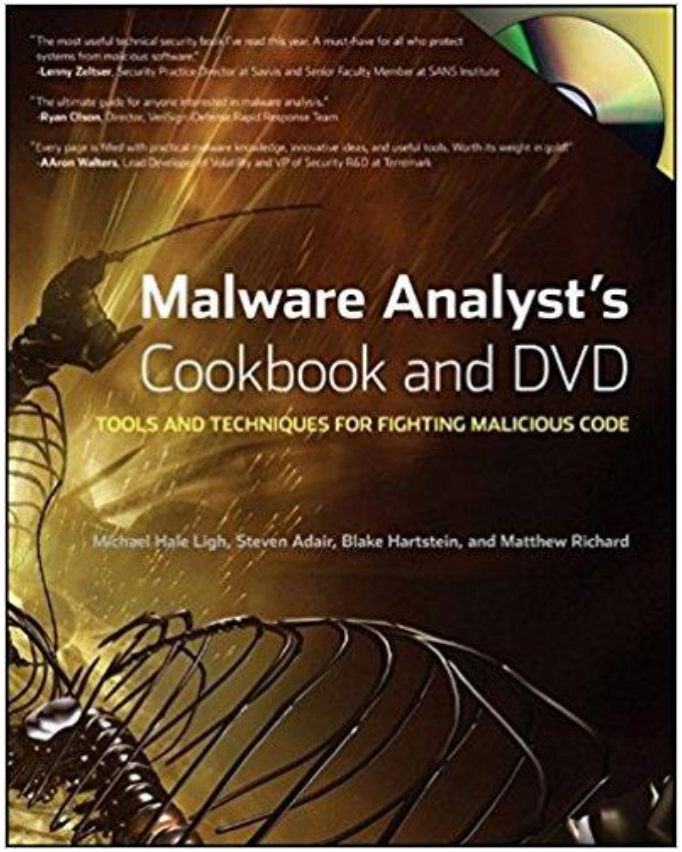
- Add pfsense firewall
- Add IDS box – i.e. snort, bro
- Set up email spam traps for collecting macro malware

Try out different tools

Get creative!

Utilize Library Resources at Your School





Questions

MITRE is Hiring!

Email us your resume directly or apply online

skern@mitre.org // sheilman@mitre.org

https://www.mitre.org/careers/working-at-mitre

Cyber New Professional - 00047874

McLean, Virginia

6 additional locations

Cyber Security | 00047874 | Jan 29, 2018

Apply for Job | Share this Job | Sign Up for Job Alerts

Please return all flash drives!!!

Resources

https://www.alienvault.com/blogs/security-essentials/building-a-home-lab-to-become-a-malware-hunter-a-beginners-guide

https://www.sans.org/reading-room/whitepapers/tools/building-automated-behavioral-malware-analysis-environment-open-source-software-33129

http://opensecuritytraining.info/MalwareDynamicAnalysis.html

https://blog.christophetd.fr/malware-analysis-lab-with-virtualbox-inetsim-and-burp/

https://zeltser.com/mastering-4-stages-of-malware-analysis/

Sikorski, Michael and Honig, Andrew. *Practical Malware Analysis*. San Francisco, CA: No Starch, 2012. Print.

Michael Hale Ligh. Steven Adair. Blake Hartstein. Matthew Richard. *Malware Analyst's Cookbook: Tools and Techniques for Fighting Malicious Code*. N.p.: John Wiley & Sons, 2011. Print.