**LAB 8: Configuring a Malware Lab**

# Purpose

* Basic Static Techniques
* Basic Dynamic Techniques

# What you need:

A Windows 2008 Server virtual machine with a Kali vitual machine running INetSim, which you prepred in the previous project.

# Basic Static Techniques

* 1. **Lab01-01.exe** [**https://www.hybrid-analysis.com/**](https://www.hybrid-analysis.com/)**:**

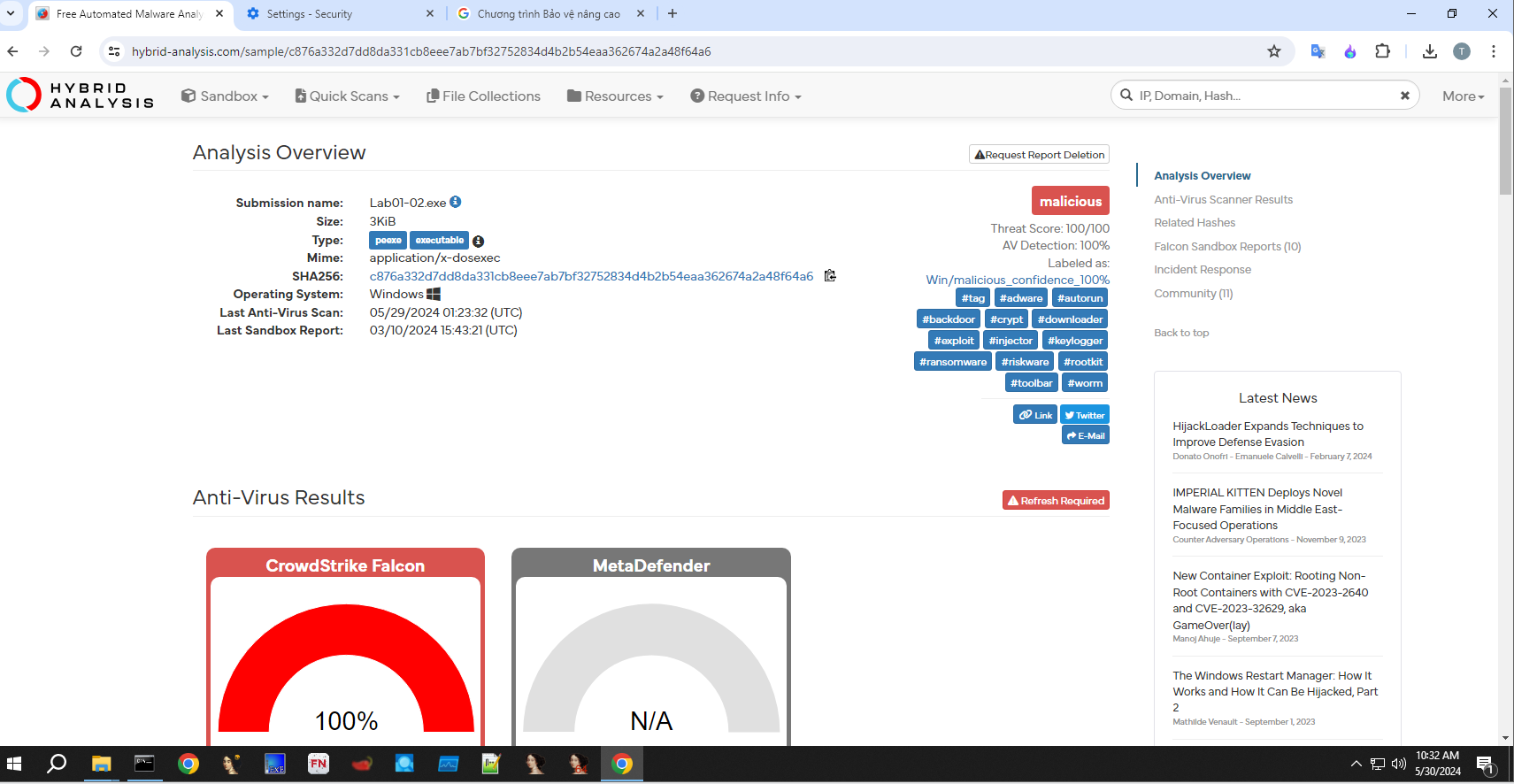
**Upload the Lab01-01.exe and Lab01-01.dll files to https://www.hybrid- analysis.com**

**A screenshot of a computer

Description automatically generatedTurn in the image showing your analysis of Lab01-01.dll as shown below.**

**A screenshot of a computer

Description automatically generated**



**PEview**

You can download PEview from here: <http://wjradburn.com/software/>

Open the files in PEview. For each file, find the "Time Date Stamp" as shown below. The files were both compiled on the same date within a minute of each other, indicating that they are part of the same package.

Turn in the image showing your analysis of Lab01-01.exe as shown below.

A screenshot of a computer

Description automatically generated

# PEiD

You can download PEiD here: <http://www.softpedia.com/progDownload/PEiD-updated-Download-4102.html>

Open the files in PEiD. They are identified as "Microsoft Visual C++" files, which shows that they are unpacked.

Turn in the image showing your analysis of Lab01-01.dll as shown below. We will grade it based on the "First Bytes".

A screenshot of a computer

Description automatically generated

# Strings

A screenshot of a computer

Description automatically generated

**Notice these items, as shown below:**

* "FindNextFileA" and "FindFirstFileA" -- Windows functions to find files
* ".exe" -- suggesting that it will search for EXE files
* "C:\windows\system32\kerne132.dll" -- fake DLL with "kerne132" instead of "kernel32"
* "C:\Windows\System32\Kernel32.dll" -- the real Windows kernel

A screenshot of a computer

Description automatically generated

Look at the strings for **Lab01-01.dll**. Notice these items, as shown below:

* + "exec" and "sleep" -- commands that can be sent over the network to control this backdoor malware
  + ".CreateProcessA" -- used to launch a program in response to the "exec" command
  + "Sleep" -- used to put the backdoor to sleep in response to the "sleep" command

Turn in the image showing your analysis of Lab01-01.dll as shown below. Below "sleep" and "hello" there is an IP address, starting with 127.

We will grade it by checking the last digits of the IP address.

A screen shot of a computer

Description automatically generated

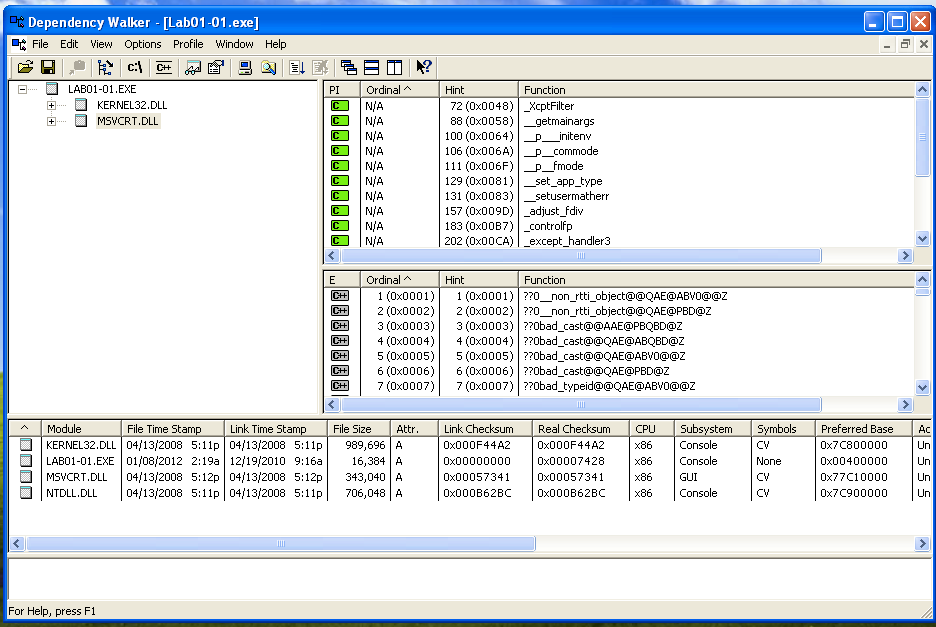
# Dependency Walker

# A screenshot of a computer Description automatically generated

Open Lab01-01.exe in Dependency Walker.

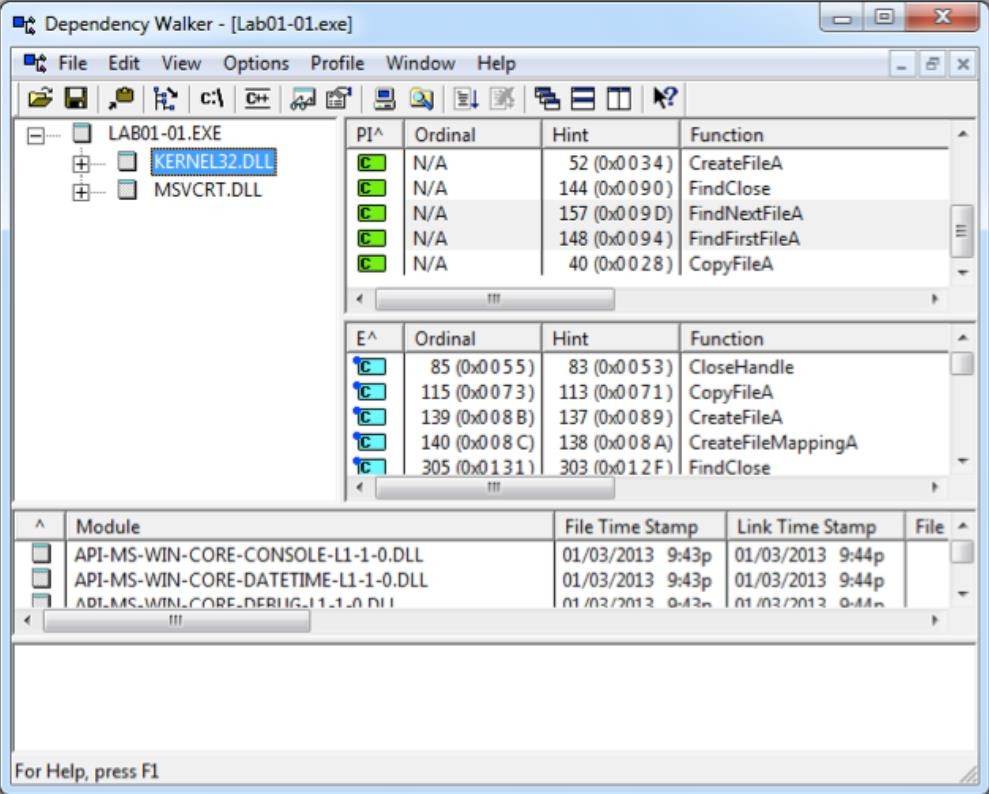
In the left pane, click MSVCRT.DLL as shown below.

There are several imports in the upper right pane, and exports in the middle right pane. Scan through them--these are normal for any EXE



In the left pane, click KERNEL32.DLL.

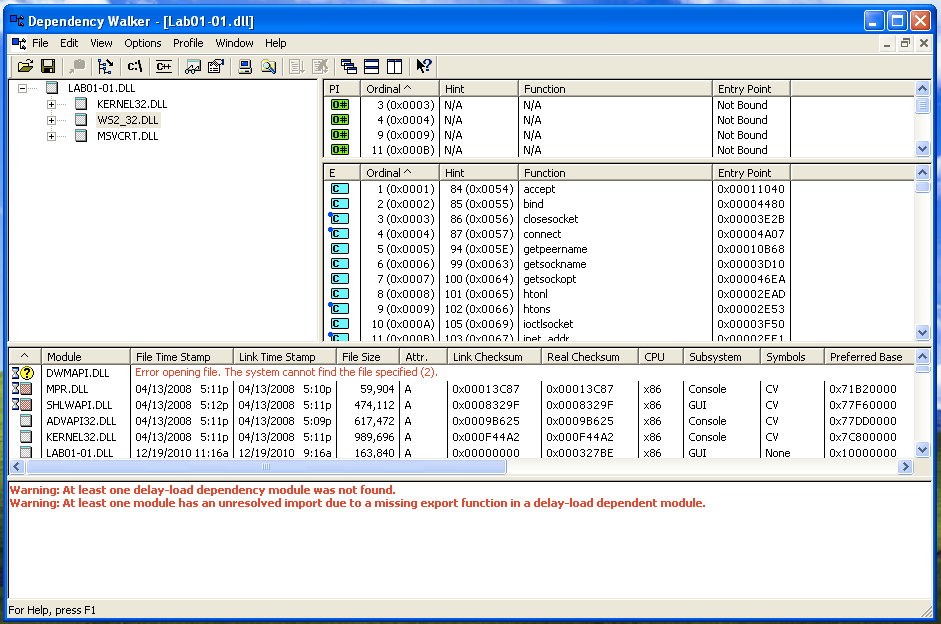
Turn in the image showing your analysis of Lab01-01.exe as shown below. In the "PI^" section (Parent Import), you should see FindNextFileA and FindFirstFileA as shown below.



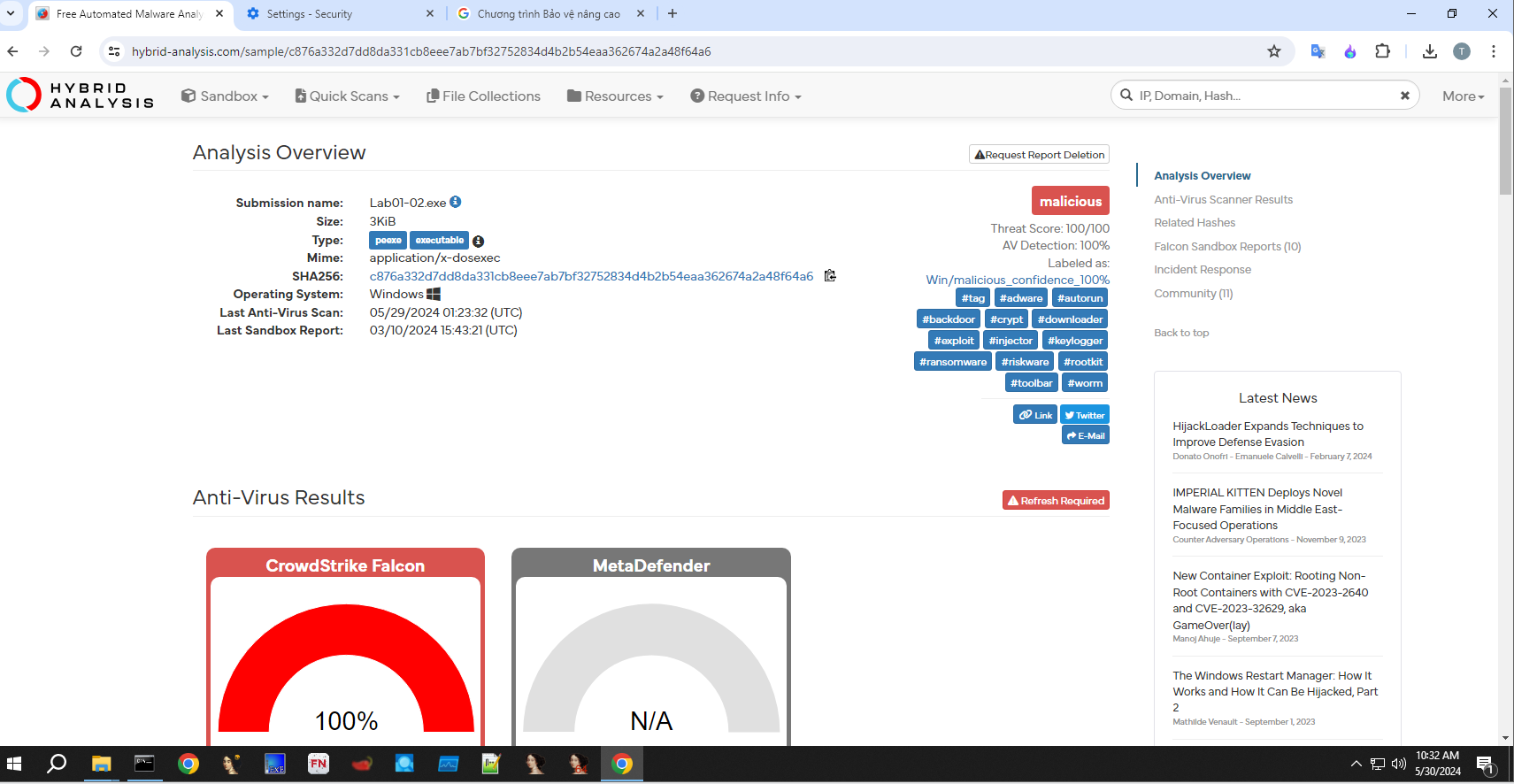
Open **Lab01-01.dll** in Dependency Walker. Notice that it imports functions from "WS2\_32.DLL".

WS2\_32.DLL has networking functions. The right center pane shows function names that perform networking tasks, such as "bind", "closesocket", and "connect", as

shown below.

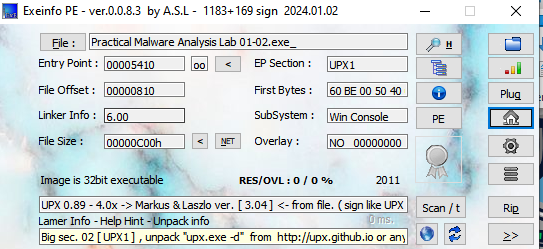


# Lab01-02.exe



**Unpacking the File**

Run PEiD on the file. It shows that the file is packed with UPX, as shown in the "EP Section" below.



Open a Command Prompt window and execute this command:

# UPX

You see a UPX help message, as shown below:

Use the CD command to move to the directory containing your malware samples. On my machine, I used this command:

A computer screen with a black screen

Description automatically generated

Analyze the unpacked file with PEiD. It now is regognized as a "Microsoft Visual C++ 6.0" file, as shown below.

Turn in the image showing your analysis of **Lab01-02-unpacked.exe** as shown below. We will grade it based on the "First Bytes".

A screenshot of a computer

Description automatically generated

Find the unpacked file's imports with Dependency Walker.

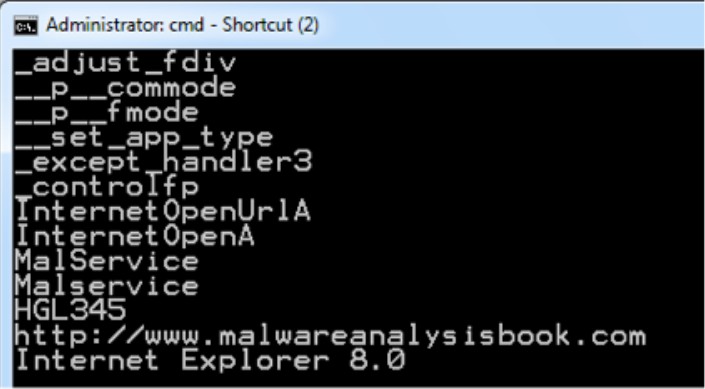
Turn in the image showing the two functions **InternetOpenUrlA** and **InternetOpenA** as shown in the upper right pane of the image below

# Strings

Find the strings in the unpacked file.

You should see **MalService** and [**http://www.malwareanalysisbook.com**](http://www.malwareanalysisbook.com/)as shown below.

These suggest that infected machines will connect to [**http://www.malwareanalysisbook.com**](http://www.malwareanalysisbook.com/)and will show a running service named **MalService**.



# Basic Dynamic Techniques

**What you need**:A Windows 2008 Server virtual machine with a Kali vitual machine running INetSim, which you prepred in the previous project.

# Purpose

You will practice the techniques in chapter 3.

This project follows **Lab 3-1** in the textbook. There are more detailed solutions in the back of the book.

# Downloading Software

At the end of the previous project, you ended up with your Windows 2008 Server machine's DNS address set to your Kali machine's IP address, which means it cannot reach the Internet.

In order to download software, you need to configure a real DNS server, such as 8.8.8.8.

# Setting the DNS Server to 8.8.8.8

On your Windows VM, in Control Panel, open "Network Connections". Right-click "**Local Area Connection**" and click **Properties**.

Double-click "**Internet Protocol (TCP/IP)**". Set your DNS server to 8.8.8.8

# Required Downloads

Make sure you have these items:

**Lab Files** from <http://practicalmalwareanalysis.com/labs/> -- download and unzip them.

**PEview** from <http://wjradburn.com/software/> -- download and install

**Strings** from <http://technet.microsoft.com/en-us/sysinternals/bb897439> -- Click "**Download Strings**" to get **Strings.zip**; unzip it, and copy **strings.exe** to the **C:\Windows\System32** folder.

**Process Monitor** from <http://technet.microsoft.com/en-us/sysinternals/bb896645> -- download and unzip

**Process Explorer** from <http://technet.microsoft.com/en-us/sysinternals/bb896653.aspx> -- download and unzip

**Wireshark** from <http://www.wireshark.org/> -- download and install

# Using PEview

Open **Lab03-01.exe** in PEview. As shown below, the only DLL imported is kernel32.dll, and the only function imported is ExitProcess. That doesn't tell us much--perhaps

this malware is packed and the real imports will come at runtime.

Turn in the image showing the imports of **Lab03-01.exe** as shown below. We will grade it by checking the Data value.

A screenshot of a computer

Description automatically generated

# Using Strings

Examine the strings in **Lab03-01.exe** and find these items, as shown below.

SOFTWARE\Classes\http\shell\open\commandV -- A registry location [www.practicalmalwareanalysis.com](http://www.practicalmalwareanalysis.com/) -- a URL

VideoDriver

These readable strings are surprising--if the malware were packed, the strings would not be readable.

A screenshot of a computer

Description automatically generated

# Preparing for Dynamic Analysis

Dynamic analysis will help us to understand this malware better. Here is the process detailed below:

1. Set up INetSim to simulate the Internet
2. Setting the DNS Server
3. Run Process Explorer
4. Run Wireshark
5. Run Process Monitor

# Start INetSim

Start both the Windows and Linux VMs.

In Linux, start inetsim, as you did in the previous project.

Set the Windows DNS server to the Linux machine's IP address, as you did in the previous project.

Test it by opening a Web browser to this URL: **YOURNAME.com**

You should see the "INetSIM HTTP server" page, as shown below:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# Setting the DNS Server

On your Windows VM, in Control Panel, open "Network Connections". Right-click "**Local Area Connection**" and click **Properties**.

A screenshot of a computer

Description automatically generatedDouble-click "**Internet Protocol (TCP/IP)**".

Set your DNS server to the Kali Linux machine's IP address, as show below:

# Run Process Explorer

Open Process Explorer, as shown below:

A screenshot of a computer

Description automatically generated

# Run Wireshark

A screenshot of a computer

Description automatically generated

Start Wireshark and begin capturing packets from the interface that goes to the Linux machine, which is normally "Local Area Connection".

# Start Process Monitor

It's best to start Process Monitor last, so you can exclude all the harmless processes the other tools are using.

In the folder you unzipped Process Monitor into, double-click **Procmon.exe**. If a Security Warning box pops up, allow the software to run.

Agree to the license.

You should see Process Monitor, with a lot of processes visible, as shown below:

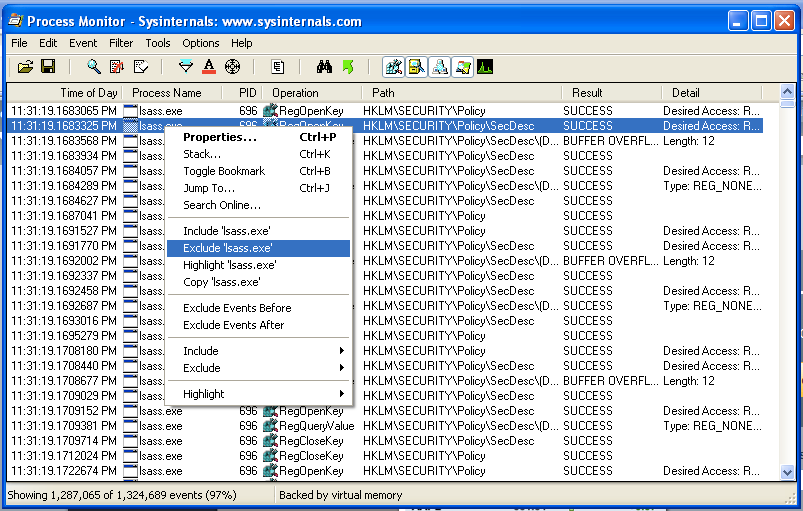
A screenshot of a computer

Description automatically generated

# Excluding Harmless Processes

To make the analysis easier, we will ignore all the processes that are already running before the malware starts.

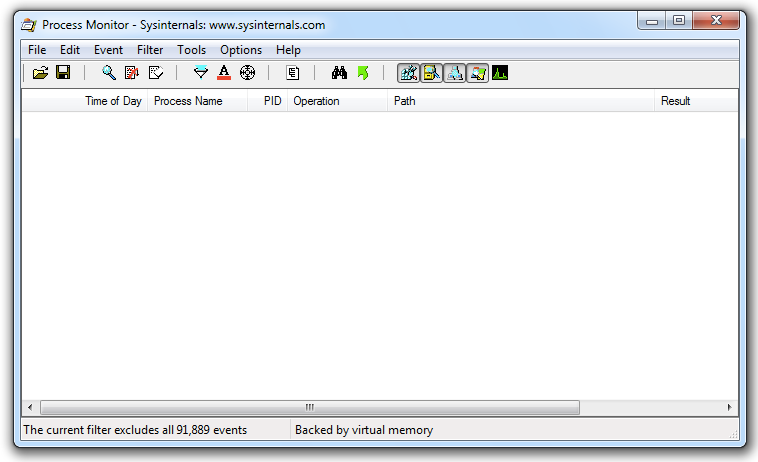
In Process Monitor, right-click the name of one of the visible processes, such as **lsass**, and click "**exclude 'lsass.exe'**", as shown below:



Wait while the event filter is applied.

Right-click a remaining process, such as "svchost.exe" and exclude it too.

Repeat the process until all current processes are hidden, as shown below. When I did it, the remaining processes to exclude were csrss.exe, explorer.exe, services.exe,vmtoolsd.exe, iexplore.exe, VMwareTray.exe, verclsid.exe, winlogon.exe, wmiprvse.exe, wuauclt.exe, regshot.exe, spoolsv.exe, alg.exe, rundll.exe, WMIADAP.EXE, GoogleUpdate.exe, GoogleCrashHandler.exe, chromeinstaller.exe, and setup.exe.



# Run the Lab03-01.exe File

Now double-click the Lab03-01.exe File.

# Viewing the Running Malware in Process Explorer

In Process Explorer, in the top pane, find **Lab03-01.exe** and click it.

# Troubleshooting

If the Lab03-01.exe process does not appear in Process Explorer, that probably means that the malware has already been run on this VM.

To make the malware run properly again, restart the VM, press F8, enter Safe Mode, and delete this file:

C:\Windows\System32\vmx32to64.exe Then restart the VM in normal mode.

In Process Explorer, click **View**, "**Lower Pane View**", **Handles**.

You see the **WinVMX32** mutant, as highlighted below. A mutant, also called a mutex, is used for interprocess connunication. A wonderful explantion of mutexes in terms

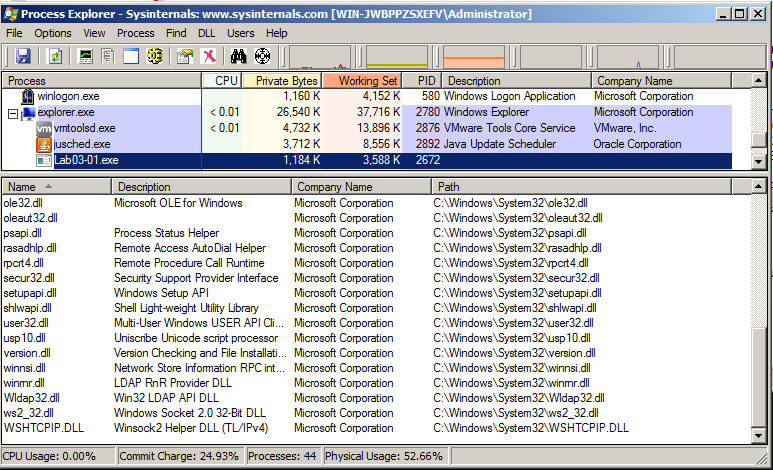
of rubber chickens is [here](https://stackoverflow.com/questions/34524/what-is-a-mutex).

A screenshot of a computer screen

Description automatically generated

In Process Explorer, click **View**, "**Lower Pane View**", **DLLs**.

Scroll to the bottom to find **ws2\_32.dll** and **WSHTCPIP.DLL**, as shown below. This shows that the malware has networking functionality.

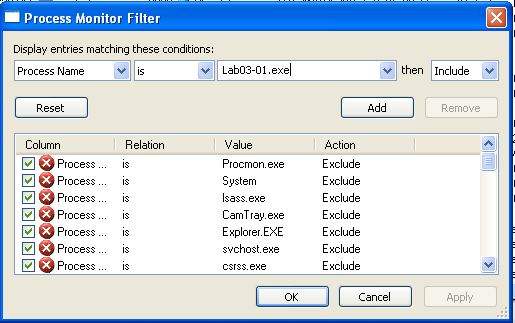


# Viewing the Malicious Process's Events in Process Monitor

In Process Monitor, click the magnifying glass icon on the toolbar to stop capturing events.

In Process Monitor, click **Filter**, **Filter**. Enter a Filter for "**Process Name**" is **Lab03- 01.exe**, **Include**, as shown below.

Click **Add** to add the filter.

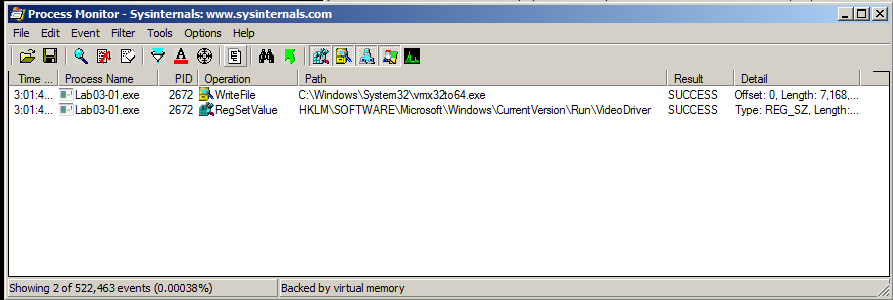


Add two more filters:

# Operation of RegSetValue Operation of WriteFile

Click **OK**.

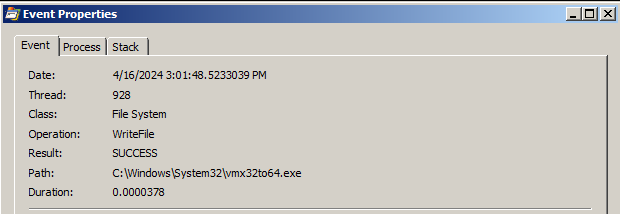
You end up the two events shown below. (Windows XP has an additional 8 events with Paths ending in "Cryptography\RNG\Seed" -- if you see those events, just ignore them.)



Only the second and third events are interesting.

Double-click the event with a Path ending in **vmx32to64.exe**. The Properties sheet shows that this event creates a file named vmx32to64.exe, as shown below.

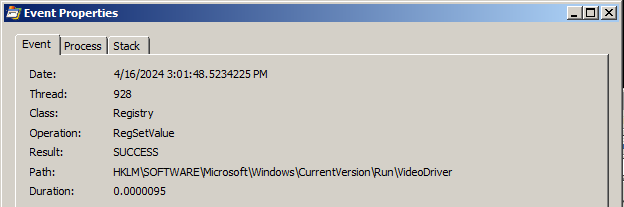
As explained in more detail in the book, this event has copied the malware itself to a file named vmx32to64.exe, so that filename is a useful indicator of infection.



Double-click the with a Path ending in **VideoDriver**.

This creates a new a Run key in the registry named "VideoDriver" with a value of "C:\WINDOWS\system32\vmx32to64.exe" -- this is a persistence mechanism, to relaunch

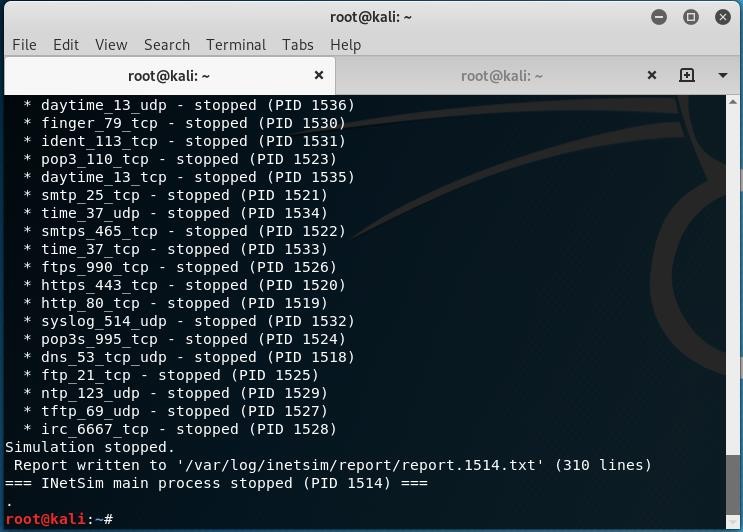
the malware when the machine restarts.



# Viewing INetSim Logs

On the Kali Linux machine, click in the window running inetsim.

Press **Ctrl+C**. A message appears telling you where the Report file is, as shown below:

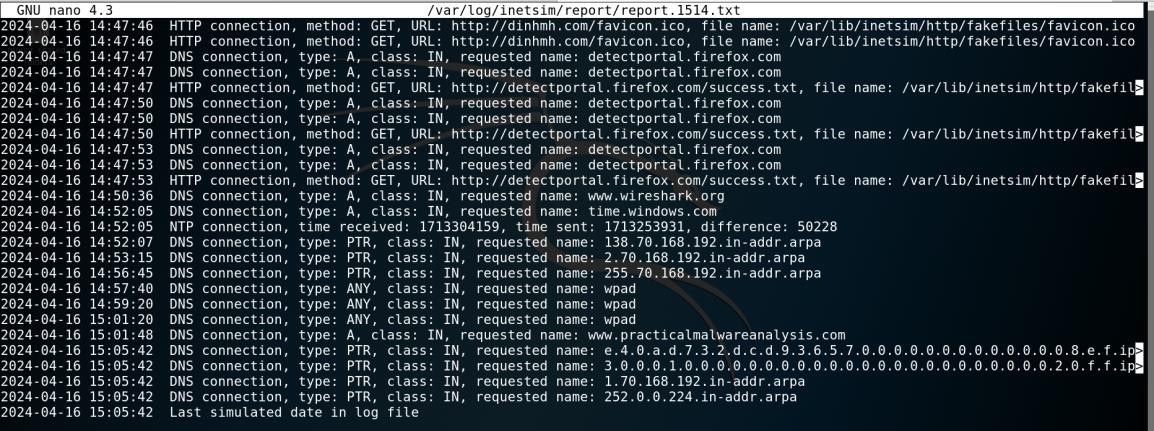


In the Linux machine, execute this command, replacing "report.3384.txt" with the correct name of your report file.

# nano /var/log/inetsim/report/report.3384.txt

Scroll to the bottom and you should see DNS connections to

[**www.practicalmalwareanalysis.com**,](http://www.practicalmalwareanalysis.com/) as shown below:



**CrackMe8**

Kiểm tra file bằng tool ExeInfo để kiểm tra thông tin của file.

A screenshot of a computer

Description automatically generated

Ở đây em vẫn sẽ chỉ cần đổi cờ từ jz sang jnz là em có thể crack được  
A screenshot of a computer

Description automatically generated

Và đây là hiển thị của nó

