

Malware Analysis Lab Configuration

NCI Post Graduate Diploma Cyber Security
Semester 1

Author: Jonathon Taaffe

Title: Malware Analysis Lab Configuration

Author: Jonathon Taaffe

Copyright© 2020 Jonathon Taaffe

All rights reserved. This publication is protected by copyright, and permission must be obtained from the author prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise.

No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this publication, the author assumes no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

Warning and Disclaimer

The information is provided on an “as is” basis. The author shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this publication. The opinions expressed in this publication belong to the author.

Trademark Acknowledgments

All terms mentioned in this publication that are known to be trademarks or service marks have been appropriately capitalised. The author cannot attest to the accuracy of this information. Use of a term in this publication should not be regarded as affecting the validity of any trademark or service mark.

Contents

Malware Analysis Lab Configuration	3
Malware Analysis Utilities Selection	3
Virtual Network Configuration.....	3
VirtualBox NATNetwork	3
VirtualBox Host-Only Adapter	3
VirtualBox Internal Network.....	3
Utilities VM Configuration.....	4
File Transfer: Windows Server 2016 File Server.....	4
Configuring File and Storage Services	4
Malware Analysis Utilities	5
Gateway OS – REMnux VM Configuration.....	5
FakeNet-NG VM Configuration.....	6
FakeNet-NG Installation – Pre-Requisites	6
FakeNet-NG Installation – Installation	6
Client Configuration.....	7
Client VM Configuration	7
Client Network Configuration.....	7
Client OS Configuration	8
Malware Analysis Lab Environment Configuration	9
Phase 1 Installation and Configuration	9
Phase 2 File Transfer.....	10
Phase 3 Client Application File Transfer	11
Phase 4 Client Application Installation	11
Phase 5 Malware Analysis Tools File Transfer to Clients.....	12
Phase 6 Dynamic Malware Analysis Configuration	12
Practical Malware Analysis.....	13
Introduction	13
Practical Malware Analysis Lab	13
VirtualBox NATNetwork	13
Kali Utilities VM Configuration	13
Clients Configuration	13
Labs, Applications and Analysis Tools Installation.....	14
Practical Malware Analysis Lab Environment Configuration.....	15
Practical Malware Analysis Exercises.....	16
Chapter 1: Basic Static Techniques - Lab1-1.....	17
Chapter 1: Basic Static Techniques - Lab1-2.....	22
Chapter 3: Basic Dynamic Analysis - Lab3-1	25
Chapter 5: Anti Reverse Engineering - Lab5-1.....	31
Chapter 9: OllyDBG - Lab9-1.....	35
Chapter 11: Malware Behaviour - Lab11-1	44
Chapter 12: Covert Malware Launching - Lab12-1.....	51
Chapter 13: Data Encoding - Lab13-1.....	56
References.....	61

Malware Analysis Lab Configuration

Malware Analysis Utilities Selection

As detailed in Lecture ‘Malware Types & Lab Setups File’¹, Gateway OS <https://REMnux.org>² was documented as a Linux Toolkit for Reverse-Engineering and Analyzing Malware.

Virtual Network Configuration

Oracle VirtualBox³ was chosen as the virtual platform for the Malware Analysis Lab.

VirtualBox NATNetwork

NATNetwork configured with DHCP range of 10.0.2.0/24 for initial VM install, update and configuration.

VirtualBox Host-Only Adapter

Host-Only connection configured from File Server to local host to allow for initial file transfer from local host to clients on the isolated Internal Network.

VirtualBox Internal Network

Isolated Internal Network called ‘malware-analysis-network01’ was configured which VM’s would connect to once fully installed and configured. As DHCP services would not be available on the Internal Network, and to easily identify VM’s connected to the Internal Network by IP, static IP addresses were assigned as follows:

Category	Type	Interface	Adapter	IP Range
Utilities	File Server	LAN	Host-Only Adapter	192.168.202.10
Utilities	File Server	LAN	Internal Network	10.0.0.10-19/24
Utilities	Malware Analysis	LAN	Internal Network	10.0.0.20-29/24
Client	XP	LAN	Internal Network	10.0.0.60-69/24
Client	Windows 7	LAN	Internal Network	10.0.0.70-79/24
Client	Windows 8	LAN	Internal Network	10.0.0.80-89/24
Client	Windows 10	LAN	Internal Network	10.0.0.100-109/24

Table 1. malware-analysis-network01 Static Assignments⁴

¹ Malware Types & Lab Setups File (2019) *Gateway OS Slide 24* <https://moodle.ncirl.ie/mod/resource/view.php?id=59717> [Accessed 1st July 2019].

² REMnux.org (2019) *REMnux: A Linux Toolkit for Reverse-Engineering and Analyzing Malware* <https://remnux.org/> [Accessed 1st July 2019].

³ VirtualBox.org (2019) *VirtualBox* <https://www.virtualbox.org/> [Accessed 1st July 2019].

⁴ Jonathon Taaffe (2019) *Table 1. malware-analysis-network01 Static Assignments* [Created 1st July 2019].

Utilities VM Configuration

File Transfer: Windows Server 2016⁵ File Server

With Automatic Windows Updates, Virus Protection and Windows Firewall disabled on the clients on which Malware was going to run, Internet connectivity from these clients was not an option. To facilitate the transfer of required Application Source files as documented in Lecture ‘Malware Types & Lab Setups File’⁶ ,a Windows Server 2016 File Server was installed and configured.

This File Server had 2 Network Adapters configured, a Host-Only Adapter with a static IP assigned and an Internal Network static IP assigned. This facilitated downloading the required Application Source files from the Internet to local host, transferring the files to the File Server over the Host-Only adapter and allowed the clients to connect to the File Server from the Internal Network and download the required files.

Once all Application Source files had been copied to the clients, the File Server will be powered off to ensure the Internal Network is fully isolated. Summary of the File Server VM configuration is as follows:

Name	WS2016-File-Server01		
Type	Microsoft		
Version	Windows 2016 (64-bit)		
RAM	2048MB		
Disk	VDI Dynamic		
Disk Size	C:50GB D:50GB		
Network Details	Initial VM Install	Upload from Host	Transfer to Clients
Adapter Number	01	01	02
Adapter	NATNetwork	Host-Only Adapter	Internal Network
Network	NATNetwork	Host-Only Adapter	malware-analysis-network01
Type	Intel PRO/1000MT	Intel PRO/1000MT	Intel PRO/1000MT
MAC	08:00:27:63:60:FB	08:00:27:18:93:FF	08:00:27:13:9B:15
IP	DHCP	192.168.202.10	10.0.0.20
Share Name	n/a	mw_client_files	mw_client_files
Share Location	n/a	D:\mw_client_files	D:\mw_client_files
Share Account	n/a	Administrator	MWUser01
Share Permissions	n/a	Full	Read

Table 2. Windows Server 2016 File Server VM Configuration⁷

Configuring File and Storage Services⁸

Added the File and Storage Services\ File Server Role and created a new SMB Share Quick as follows:

Share Name: MW_Client_Files

Share Settings: Enable Access-Based Enumeration

Specify Permissions: WS2016_File_Server\mwclient

⁵ Portal.Azure.com (2019) Windows Server 2016 Standard

http://dl.msdn.com/pr/en_windows_server_2016_updated_feb_2018_x64_dvd_11636692.iso [Accessed 1st July 2019].

⁶ Malware Types & Lab Setups File (2019) Tools Slide 28 <https://moodle.ncirl.ie/mod/resource/view.php?id=59717> [Accessed 1st July 2019].

⁷ Jonathon Taaffe (2019) Table 2. Windows Server 2016 File Server VM Configuration [Created 1st July 2019].

⁸ Tectig (2019) How to Share Files and Folders in Windows Server 2016 <https://www.tectig.com/share-files-folders-windows-server-2016/> [Accessed 1st July 2019].

Malware Analysis Utilities

As detailed in Lecture ‘Malware Types & Lab Setups File’ Gateway OS – REMnux⁹ and FakeNet-NG¹⁰ were recommended for Malware Analysis. Both Malware Analysis Utilities were installed and configured in the Malware Lab allowing for analysis from 2 different platforms. To use these utilities, the TCP/IP Gateway IP of the victim clients needs to be set to the static IP of the analysis utility as follows

Utility	Static IP
Gateway OS – REMnux	10.0.0.20
FakeNet-NG	10.0.0.21

Table 3. Malware Analysis Utilities Static IP Assignment¹¹

Gateway OS – REMnux VM Configuration

Downloaded REMnux Virtual Appliance OVA¹² and connected it to the NATNetwork for initial appliance install, update and configuration. Successfully updated using the 'update-remnux full' command.

Once the VM was updated and operational, the VM’s network connection was changed to the isolated Internal Network ‘malware-analysis-network01’ and set a static IP address of 10.0.0.20/24 with no Gateway specified. Summary of the Gateway OS – REMnux VM configuration is as follows:

Name	Remnux Gateway OS	
Type	Linux	
Version	Ubuntu 14.04 (64-bit)	
RAM	1024MB	
Disk	VDI Dynamic	
Disk Size	25GB	
Network Details	Initial Install and Configuration	Isolated Network
Adapter 1	NATNetwork	Internal Network
Adapter 1 Network	NATNetwork	malware-analysis-network01
Adapter 1 Type	Intel PRO/1000MT Server (82545EM)	Intel PRO/1000MT Server (82545EM)
Adapter 1 MAC	08:00:27:93:01:19	08:00:27:93:01:19
Adapter 1 IP	DHCP	10.0.0.20

Table 4. Gateway OS – REMnux VM Configuration¹³

⁹ Malware Types & Lab Setups File (2019) *Gateway OS Slide 24* <https://moodle.ncirl.ie/mod/resource/view.php?id=59717> [Accessed 1st July 2019].

¹⁰ Malware Types & Lab Setups File (2019) *FakeNet-NG Slide 25* <https://moodle.ncirl.ie/mod/resource/view.php?id=59717> [Accessed 1st July 2019].

¹¹ Jonathon Taaffe (2019) *Table 3. Malware Analysis Utilities Static IP Assignment* [Created 1st July 2019].

¹² REMnux 6.0 OVA Public (2019) *remnux-6.0-ova-public.ova* (2.0G)

https://docs.google.com/uc?id=0B6fULLT_NpxMampUWIBCQXVJZzA&export=download [Accessed 01/07/2019].

¹³ Jonathon Taaffe (2019) *Table 4. Gateway OS – REMnux VM Configuration* [Created 1st July 2019].

FakeNet-NG VM Configuration¹⁴

Downloaded Ubuntu 18.04 ISO and installed a new VM connected to the NATNetwork for initial appliance install, update and configuration as per the following VM configuration:

Name	FakeNet-NG	
Type	Linux Ubuntu (64-bit)	
RAM	1024MB	
Disk	VDI Dynamic	
Disk Size	50GB	
Network Details	Initial Install and Configuration	Isolated Network
Adapter 1	NATNetwork	Internal Network
Adapter 1 Network	NATNetwork	malware-analysis-network01
Adapter 1 Type	Intel PRO/1000MT Server (82545EM)	Intel PRO/1000MT Server (82545EM)
Adapter 1 MAC	08:00:27:65:65:4F	08:00:27:65:65:4F
Adapter 1 IP	DHCP	10.0.0.21

Table 5. FakeNet-NG VM Configuration¹⁵

FakeNet-NG Installation – Pre-Requisites¹⁶

Completed an apt-get update and apt-get upgrade and installed all FakeNet-NG prerequisites including:

Package	Command
Git package manager	apt install git
Python 2.7 pip package manager	apt install python-pip
Python file transfer package	pip install pysendfile
Python FPTS package	pip install pyopenssl
Python 2.7 development files	apt install python-dev
OpenSSL development files	apt install libssl-dev
libffi development files	apt install libffi-dev
libnetfilterqueue development files	apt install libnetfilter-queue-dev

Table 6. FakeNet-NG Pre-Requisites¹⁷

FakeNet-NG Installation – Installation¹⁸

Once all pre-requisites were installed, fakenet-ng was cloned into the /opt directory and python installer run as follows:

```
fakenet@FakeNet-NG:/$ cd /opt
fakenet@FakeNet-NG:/opt$ git clone https://github.com/fireeye/flare-fakenet-ng/
fakenet@FakeNet-NG:/opt$ cd flare-fakenet-ng
fakenet@FakeNet-NG:/opt/flare-fakenet-ng$ python setup.py install
```

Network Adapter changed to Internet Network ‘malware-analysis-network01’ and configured a static IP address of 10.0.0.21/24 with no Gateway specified.

¹⁴ FireEye/Flare-FakeNet-NG (2019) *FakeNet-NG - Next Generation Dynamic Network Analysis Tool* <https://github.com/fireeye/flare-fakenet-ng> [Accessed 1st July 2019].

¹⁵ Jonathon Taaffe (2019) *Table 5. Gateway OS – REMnux VM Configuration* [Created 1st July 2019].

¹⁶ FireEye/Flare-FakeNet-NG (2019) *FakeNet-NG - Next Generation Dynamic Network Analysis Tool* <https://github.com/fireeye/flare-fakenet-ng> [Accessed 1st July 2019].

¹⁷ Jonathon Taaffe (2019) *Table 6. FakeNet-NG Pre-Requisites* [Created 1st July 2019].

¹⁸ FireEye/Flare-FakeNet-NG (2019) *FakeNet-NG - Next Generation Dynamic Network Analysis Tool* <https://github.com/fireeye/flare-fakenet-ng> [Accessed 1st July 2019].

Client Configuration

Client VM Configuration

As the vast majority of today's malware is written for Microsoft Windows Operating Systems and to allow Malware testing on a variety of clients, 32bit versions of Windows XP, Windows 7, Windows 8.1 and Windows 10¹⁹ were chosen as the victim clients.

As these client VM's were being installed with Automatic Windows Updates, Virus Protection and Windows Firewall disabled, they were installed and configured connected to the isolated Internal Network 'malware-analysis-network01'. VM configurations as follows:

Client	WindowsXP-01	Windows7-01	Windows8-01	Windows10-01
Type	Microsoft Windows			
OS Version	Windows XP 32bit	Windows 7 Pro 32bit	Windows 8 32bit	Windows 10 32bit
OS Build	5.1.2600 SP3	6.1.7601 SP1	6.3.9600	10.0.10240
RAM	1024MB			
Disk	VDI Dynamic			
Disk Size	50GB			
Adapter 1	Internal Network			
Network	malware-analysis-network01			
MAC	08:00:27:D2:93:84	08:00:27:1E:0F:51	08:00:27:1A:2E:5A	08:00:27:A7:A7:32
IP	10.0.0.60	10.0.0.70	10.0.0.80	10.0.0.100
Subnet	255.255.255.0			
Gateway	10.0.0.20 (Remnux Gateway OS)			

Table 7. Windows XP, 7, 8.1 and 10 VM Client Configuration²⁰

Client Network Configuration

To ensure the Windows Clients were optimally configured for TCP/IPv4 on the malware-analysis-network01, the following network settings were configured:

Client	WindowsXP-01	Windows7-01	Windows8.1-01	Windows10-01
Client for MS Networks	Enabled			
QoS Packet Scheduler	Enabled			
TCP/IPv4	Enabled			
Register connection in DNS	Disabled			
Enable LMHOSTS Lookup	Disabled			
Disable NetBIOS over TCPIP	Selected			
Microsoft ISATAP Adapter	Disabled			
ISATAP Adapter	Uninstalled from Device Manager			
ISATAP Adapter Disabled	netsh interface isatap set state disabled			

Table 8. Windows XP, 7, 8.1 and 10 Network Configuration²¹

¹⁹ Microsoft.com (2019) Download Virtual Machines <https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/> [Accessed 2nd July 2019].

²⁰ Jonathon Taaffe (2019) Table 7. Windows XP, 7, 8.1 and 10 VM Client Configuration [Created 2nd July 2019].

²¹ Jonathon Taaffe (2019) Table 8. Windows XP, 7, 8.1 and 10 Network Configuration [Created 2nd July 2019].

Client OS Configuration

To ensure the clients were optimally configured as per Lecture ‘Malware Types & Lab Setups File’²² and are ready for malware analysis, the following OS settings were configured:

Client	WindowsXP-01	Windows7-01	Windows8-01	Windows10-01
Automatic Updates	Off			
Virus Protection	n/a	Off	Off	Off
Windows Firewall	Off			
User Account Control	n/a	Off	Off	Off
Show File Extensions	Enabled			
Show Hidden Files	Enabled			
Disable Zone Checking*	Configured			
CMD Shortcut on Desktop	Configured			
c:\tools\bin Created	Configured			
c:\tools\bin added to %PATH%	Configured			

Table 9. Windows XP, 7, 8.1 and 10 OS Configuration²³

***Note on Disabling Internet Explorer Zone Checking²⁴:** To disable Zone Checking in Internet Explorer add the following Windows registry keys:

```
reg add "HKCU\Environment" /V SEE_MASK_NOZONECHECKS /T REG_SZ /D 1 /F

reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\1" /v "Flags"
/t REG_DWORD /d 219 /f

reg add "HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment" /V
SEE_MASK_NOZONECHECKS /T REG_SZ /D 1 /F
```

²² Malware Types & Lab Setups File (2019) *FakeNet-NG Slide 25* <https://moodle.ncirl.ie/mod/resource/view.php?id=59717> [Accessed 2nd July 2019].

²³ Jonathon Taaffe (2019) *Table 9. Windows XP, 7, 8.1 and 10 OS Configuration* [Created 2nd July 2019].

²⁴ Microsoft.com (2019) *Internet Explorer security zones registry entries for advanced users* <https://support.microsoft.com/en-us/help/182569/internet-explorer-security-zones-registry-entries-for-advanced-users> [Accessed 2nd July 2019].

Malware Analysis Lab Environment Configuration

Phase 1 Installation and Configuration

The following diagram details Phase 1 configuration of the Malware Analysis Lab which includes the Windows Server 2016²⁵ File Server, Gateway OS – REMnux²⁶ and FakeNet-NG²⁷ Malware Analysis Utilities and Windows XP, 7, 8.1 and 10²⁸ Clients network Configuration.

This configuration allows for:

1. OS install, update and configuration of the File Server and Malware Analysis Utility VM's
2. OS install and configuration of the Windows clients

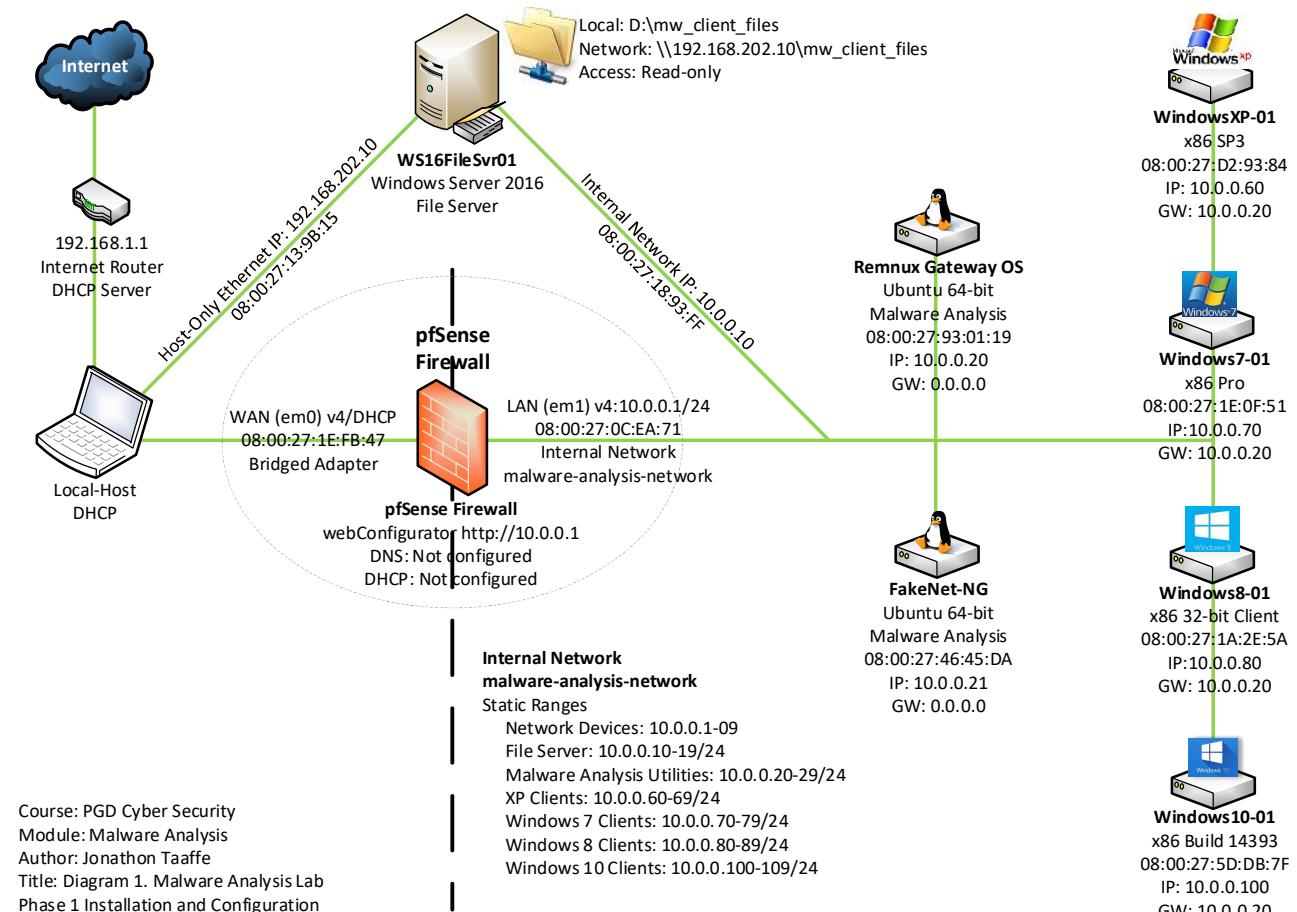


Diagram 1. Malware Analysis Lab Phase 1 Installation and Configuration ²⁹

²⁵ Portal.Azure.com (2019) *Windows Server 2016 Standard*

http://dl.msdn.com/pr/en/windows_server_2016_updated_feb_2018_x64_dvd_11636692.iso [Accessed 2nd July 2019].

²⁶ REMnux 6.0 OVA Public (2019) *remnux-6.0-ova-public.ova* (2.0G)

https://docs.google.com/uc?id=0B6fULL_NpxMampUWIBCQXVJzA&export=download [Accessed 2nd July 2019].

²⁷ FireEye/Flare-FakeNet-NG (2019) *FakeNet-NG - Next Generation Dynamic Network Analysis Tool* <https://github.com/fireeye/flare-fakenet-ng> [Accessed 2nd July 2019].

²⁸ Microsoft.com (2019) *Download Virtual Machines* <https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/> [Accessed 2nd July 2019].

²⁹ Jonathon Taaffe (2019) *Diagram 1. Malware Analysis Lab Phase 1 Installation and Configuration* [Created 2nd July 2019].

Phase 2 File Transfer

The following diagram details Phase 2 configuration of Lab 1 which includes the Windows Server 2016³⁰ File Server, Network File Share for File Transfer and the Windows XP, 7, 8.1 and 10³¹ Clients network configuration.

This configuration allows for

1. Download of all required application files from the Internet to the local host
2. Transfer of files from the local host to the Windows Server 2016 File Server share mw_client_files
3. Download of files from the network share mw_client_files to each of the Windows clients

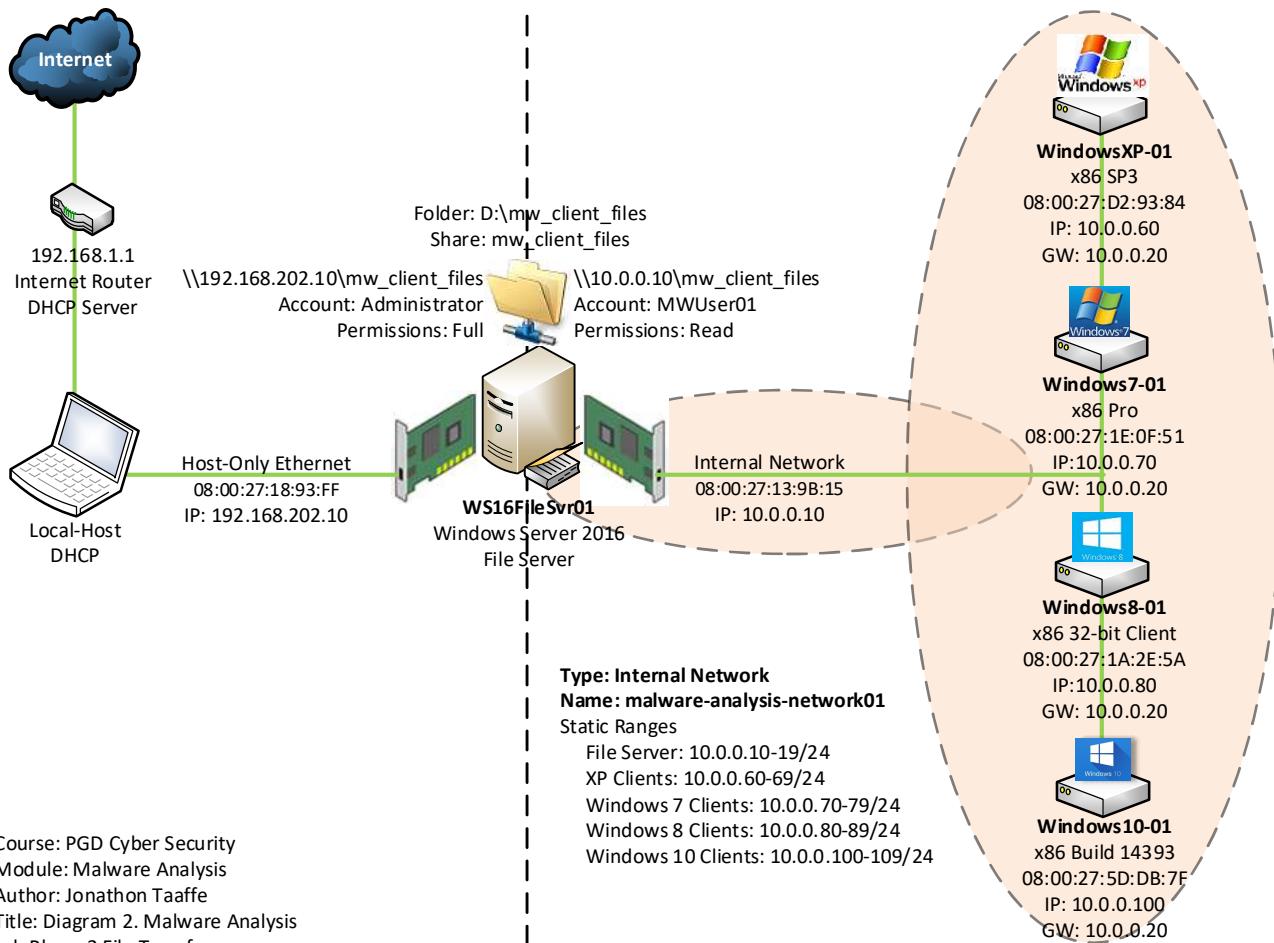


Diagram 2. Malware Analysis Lab Phase 2 File Transfer³²

³⁰ Portal.Azure.com (2019) *Windows Server 2016 Standard*

http://dl.msdn.com/pr/en/windows_server_2016_updated_feb_2018_x64_dvd_11636692.iso [Accessed 2nd July 2019].

³¹ Microsoft.com (2019) *Download Virtual Machines* <https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/> [Accessed 2nd July 2019].

³² Jonathon Taaffe (2019) *Diagram 2. Malware Analysis Lab Phase 2 File Transfer* [Created 2nd July 2019].

Phase 3 Client Application File Transfer

With the File Server configured allowing for File Transfer from local host to the Windows Clients on the Internal ‘malware-analysis-network01’ Network, the following Application Files were transferred as per Lecture ‘Malware Types & Lab Setups File’³³, to C:\Temp on each Windows Client:

Application	File	OS
Microsoft .NET Framework 3.5 SP1	Microsoft_.NET_Framework_3.5_SP1_WinXP.exe	XP
Microsoft .NET Framework 4	Microsoft_.NET_Framework_4_x86_x64.exe	7,8,10
Adobe Reader 11.0.08	Adobe_Reader_11.0.08_WinXP_SP3.exe	XP
Adobe Reader 2019.012.20034	Adobe_Reader_2019.012.20034.exe	7,8,10
Chrome 1.3.34.7	ChromeStandaloneSetup_x86_WinXP.exe	XP
Chrome 49.0.2623.75 32bit	Chrome_32bit_49.0.2623.75.exe	XP
Chrome 1.3.34.7	ChromeStandaloneSetup_x86.exe	7,8,10
Firefox Setup 52.0esr.exe	Firefox_Setup_52.0esr.exe	XP
Firefox Setup 60.0b13.exe	Firefox_Setup_60.0b13.exe	XP
Firefox Setup 67.0.1 x86	Firefox_Setup_67.0.1_x86.exe	7,8,10
Java JRE 6 Windows i586	jre-6-windows-i586.exe	XP
Java JRE 8u211 Windows i586	Java_JRE_8u211_Windows_i586.exe	7,8,10
Microsoft Office 365 Home x86 x64	Office\Setup32.exe	All
Microsoft PowerShell 1.0 KB926139-v2	Microsoft_PowerShell_1.0_(KB926139-v2)_x86_WinXP	XP
Python 2.7.9 and 3.4.3 x86	Python_2.7.9_x86.msi and Python_3.4.3_x86.msi	XP
Python 3.7.3 x86	Python_3.7.3_x86.exe	7,8,10
WinRAR 561 and 571 x86	WinRAR561.exe and WinRAR571.exe	All

Table 10. Client Application File Transfer³⁴

Phase 4 Client Application Installation

VirtualBox snapshots were used before and after each application install as follows

Action	Details
Snapshot 01	VM Snapshot
Application Install	Active Perl, Adobe Reader, Chrome
Snapshot 02	Post Application Install Snapshot
Application Install	Firefox, Java, Microsoft .NET
Snapshot 03	Post Application Install Snapshot
Application Install	Microsoft Office, Microsoft Powershell
Snapshot 04	Post Application Install Snapshot
Application Install	Microsoft Visual C++ 2008, C++ 2010
Snapshot 06	Post Application Install Snapshot
Application Install	Python, WinRAR
VM Clone	OS, Files and Installed Applications

Table 11. Client Application Installation³⁵

³³ Malware Types & Lab Setups File (2019) Tools Slide 28 <https://moodle.ncirl.ie/mod/resource/view.php?id=59717> [Accessed 2nd July 2019].

³⁴ Jonathon Taaffe (2019) Table 10. Client Application File Transfer [Created 2nd July 2019].

³⁵ Jonathon Taaffe (2019) Table 11. Client Application Installation [Created 2nd July 2019].

Phase 5 Malware Analysis Tools File Transfer to Clients

On reviewing Lecture 'Malware Types & Lab Setups File'³⁶, the current version of each of the malware analysis tools listed were downloaded. All tools were transferred to each Windows client to C:\Temp\Tools.

Phase 6 Dynamic Malware Analysis Configuration

With the 2 Malware Analysis Utility VM's and all client VM's configured with required applications installed and all analysis tools copied, the Internal 'malware-analysis-network01' Network was completely isolated by powering off the Windows Server 2016³⁷ File Server.

Note: Malware Analysis Utility Selection

Setting Windows Client Gateway IP to 10.0.0.20, clients will communicate with REMnux (red line)

Setting Windows Client Gateway IP to 10.0.0.21, clients will communicate with FakeNet-NG (blue line)

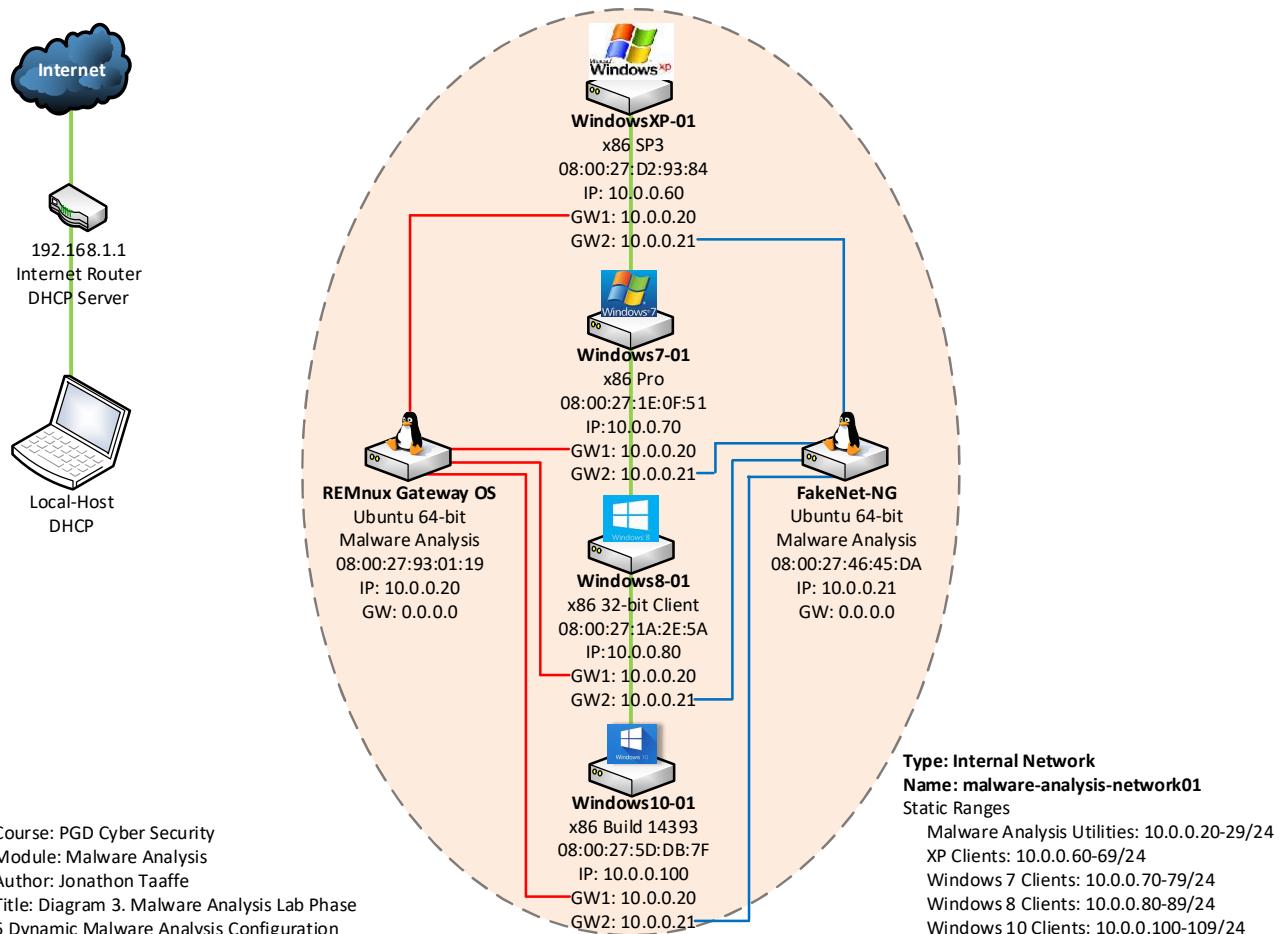


Diagram 3. Malware Analysis Lab Phase 6 Dynamic Malware Analysis Configuration³⁸

³⁶ Malware Types & Lab Setups File (2019) Tools Slides 28, 29, 30, 31, 32 <https://moodle.ncirl.ie/mod/resource/view.php?id=59717> [Accessed 2nd July 2019].

³⁷ Portal.Azure.com (2019) Windows Server 2016 Standard

http://dl.msdn.com/pr/en_windows_server_2016_updated_feb_2018_x64_dvd_11636692.iso [Accessed 2nd July 2019].

³⁸ Jonathon Taaffe (2019) Diagram 3. Malware Analysis Lab Phase 6 Dynamic Malware Analysis Configuration [Created 2nd July 2019].

Practical Malware Analysis

Introduction

Researching Malware variants, the Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software³⁹ guide was reviewed. Further research resulted in locating Sam Bowne's Practical Malware Analysis website⁴⁰ which aligned to the malware analysis labs detailed in Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software. Utilising these 2 resources, a Practical Malware Analysis lab was installed and configured.

Practical Malware Analysis Lab

Reviewing the labs content, the files used for analysis were not malicious. A NATNetwork was configured for the Practical Malware Analysis lab.

VirtualBox NATNetwork

NATNetwork was configured with DHCP range of 10.0.2.0/24 for initial VM install, update and configuration.

Kali Utilities VM Configuration

A Kali Linux distro was used in the labs to utilise iNetSim⁴¹, configured as follows

Name	Kali KDE 2019.1
Type	Linux Ubuntu (64-bit)
RAM	2048MB
Disk	VDI Dynamic (50GB)
Adapter 1 Network	NATNetwork
Adapter 1 Type	Intel PRO/1000MT Server (82545EM)
Adapter 1 MAC	08:00:27:39:E6:5A
Adapter 1 NAT DHCP IP	10.0.2.3

Table 12. Kali KDE 2019.1 VM Configuration⁴²

Clients Configuration

Windows XP x86 and Windows Server 2008 x86 were referenced in the labs to install and configure analysis tools on. VM's were configured as follows for analysis

VM Name	WindowsXP-01	WS2K8-x86-01
Operating System	Windows XP (32-bit)	Windows 2008 (32-bit)
RAM	2048MB	2048MB
Disk	VDI Dynamic (50GB)	VDI Dynamic (50GB)
Adapter 1 Network	NATNetwork	NATNetwork
Adapter 1 MAC	08:00:27:DB:DD:01	08:00:27:5B:95:06
Adapter 1 NAT DHCP IP	10.0.2.5	10.0.2.6
Adapter 1 Primary DNS	10.0.2.3 (Kali IP)	10.0.2.3 (Kali IP)

Table 13. Windows XP and Windows Server 2008 VM Configuration⁴³

³⁹ Sikorski, Michael; Honig, Andrew (2012) *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software* <https://library.ncirl.ie/items/32216> [Accessed 20th May 2019].

⁴⁰ Sam Bowne (2016) CNIT 126: Practical Malware Analysis https://samsclass.info/126/126_S16.shtml#lecture [Accessed 20th May 2019].

⁴¹ iNetSim.org (2018) *INetSim: Internet Services Simulation Suite* <https://www.inetsim.org/> [Accessed 20th May 2019].

⁴² Jonathon Taaffe (2019) Table 12. Kali KDE 2019.1 VM Configuration [Created 20th May 2019].

⁴³ Jonathon Taaffe (2019) Table 13. Windows XP and Windows Server 2008 VM Configuration [Created 20th May 2019].

Labs, Applications and Analysis Tools Installation

The following labs, applications and analysis tools were then downloaded and installed

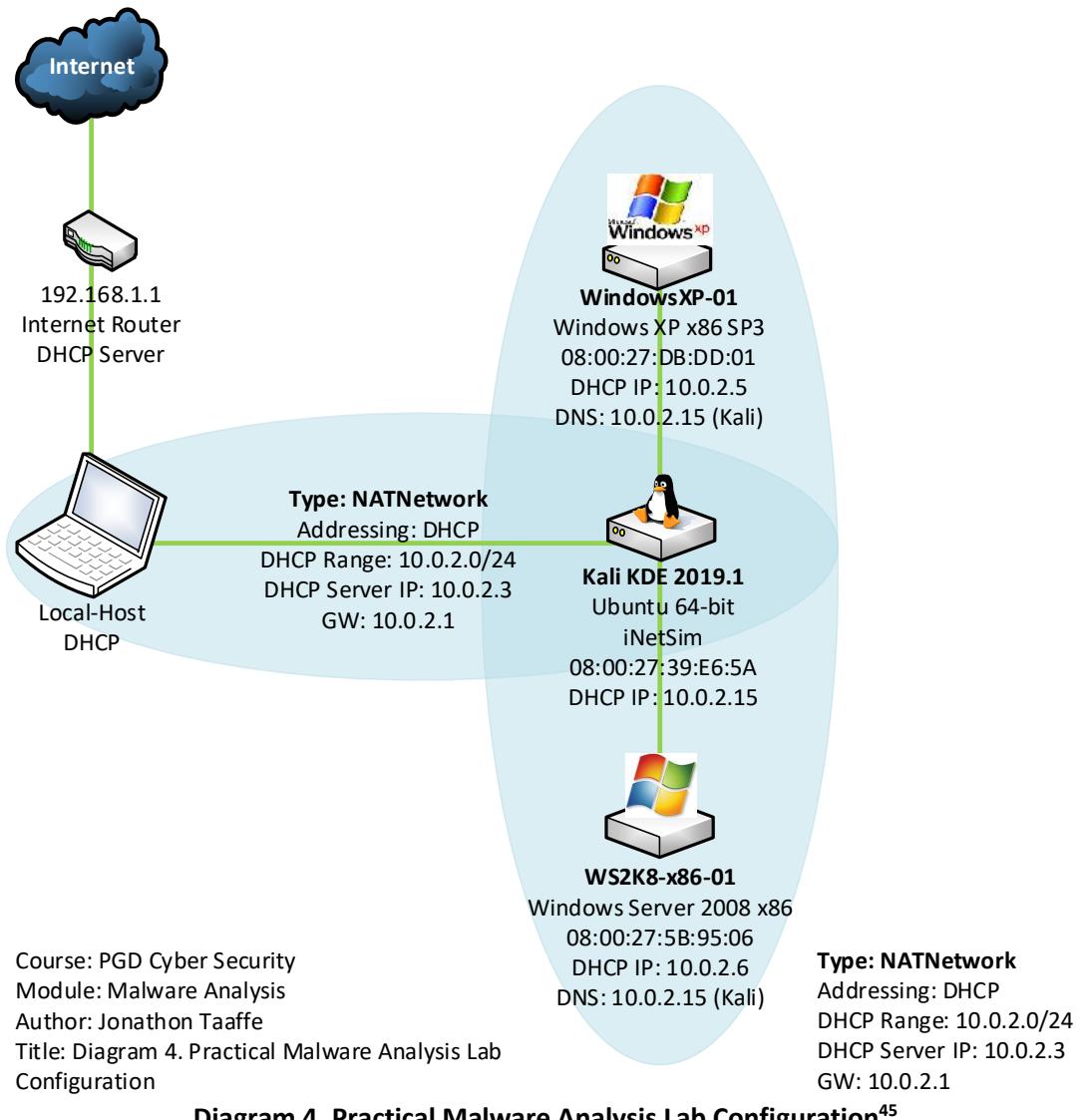
Type	Name	URL
Labs	Practical Malware Analysis	https://practicalmalwareanalysis.com/labs/
Labs	Practical Malware Analysis	https://github.com/mikesiko/PracticalMalwareAnalysis-Labs
App	7-zip	https://www.7-zip.org/
App	Java JRE 6 Windowsi586	https://www.oracle.com
App	.NET Framework 4	https://www.microsoft.com/en-us/download/details.aspx?id=17718
App	Microsoft Windows SDK	https://www.microsoft.com/en-us/download/details.aspx?id=8422
Browser	Firefox	https://www.mozilla.org/en-US/firefox/all
Site	Virus Total	https://www.virustotal.com
Analysis	PEView	http://wjrdburn.com/software/
Analysis	PEiD 0.95	http://www.softpedia.com/
Analysis	BinText3.03	https://www.softpedia.com/get/System/File-Management/BinText.shtml
Analysis	DependencyWalker 2.2	http://www.dependencywalker.com/
Analysis	UPX 3.95	https://upx.github.io/
Analysis	Strings	https://docs.microsoft.com/en-gb/sysinternals/downloads/strings
Analysis	Nmap	http://nmap.org
Analysis	SysInternals	http://docs.microsoft.com/en-gb/sysinternals/downloads
Analysis	SysInternals Strings v2.52	https://docs.microsoft.com/en-gb/sysinternals/downloads/strings
Analysis	SysInternals Process Monitor v3.50	https://docs.microsoft.com/en-gb/sysinternals/downloads/procmon
Analysis	SysInternals Process Explorer v 16.21	https://docs.microsoft.com/en-gb/sysinternals/downloads/process-explorer
Analysis	Resource Hacker 5.1.7	http://www.angusj.com/resourcehacker/
Analysis	Wireshark (Windows Server 2008) v2.0.0	https://1.eu.dl.wireshark.org/win32/all-versions/Wireshark-win32-2.0.0.exe
Analysis	IDA Pro v5.0	https://samsclass.info/126/proj/idafree50.exe
Analysis	IDA Pro v7.0	https://www.hex-rays.com/products/ida/support/download_freeware.shtml
Analysis	OllyDbg 1.10	http://www.ollydbg.de/download.htm
Analysis	LiveKd v5.62	https://docs.microsoft.com/en-gb/sysinternals/downloads/livekd
Analysis	HashCalc 2.02	http://www.slavasoft.com/hashcalc/
Analysis	WinHex	http://winhex.com/winhex/

Table 14. Labs, Applications and Analysis Tools Installation⁴⁴

⁴⁴ Jonathon Taaffe (2019) Table 14. Labs, Applications and Analysis Tools Installation [Created 20th May 2019].

Practical Malware Analysis Lab Environment Configuration

With the NATNetwork configured and the VM installed, the Practical Malware Analysis Lab was configured as follows:



⁴⁵ Jonathon Taaffe (2019) *Diagram 4. Practical Malware Analysis Lab Configuration* [Created 20th May 2019].

Practical Malware Analysis Exercises

Referencing Practical Malware Analysis, the following lab exercises were utilised

Practical Malware Analysis Reference			
Report Section	Chapter	Lab	Page
2.3a	Chapter 1: Basic Static Techniques	Lab1-1	27
2.3b	Chapter 1: Basic Static Techniques	Lab1-2	27
2.3c	Chapter 3: Basic Dynamic Analysis	Lab3-1	61
2.3d	Chapter 5: Anti Reverse Engineering	Lab5-1	107
2.3e	Chapter 9: OllyDBG	Lab9-1	202
2.3f	Chapter 11: Malware Behaviour	Lab11-1	251
2.3g	Chapter 12: Covert Malware Launching	Lab12-1	266
2.3h	Chapter 13: Data Encoding	Lab13-1	295

Table 15. Practical Malware Analysis Exercises⁴⁶

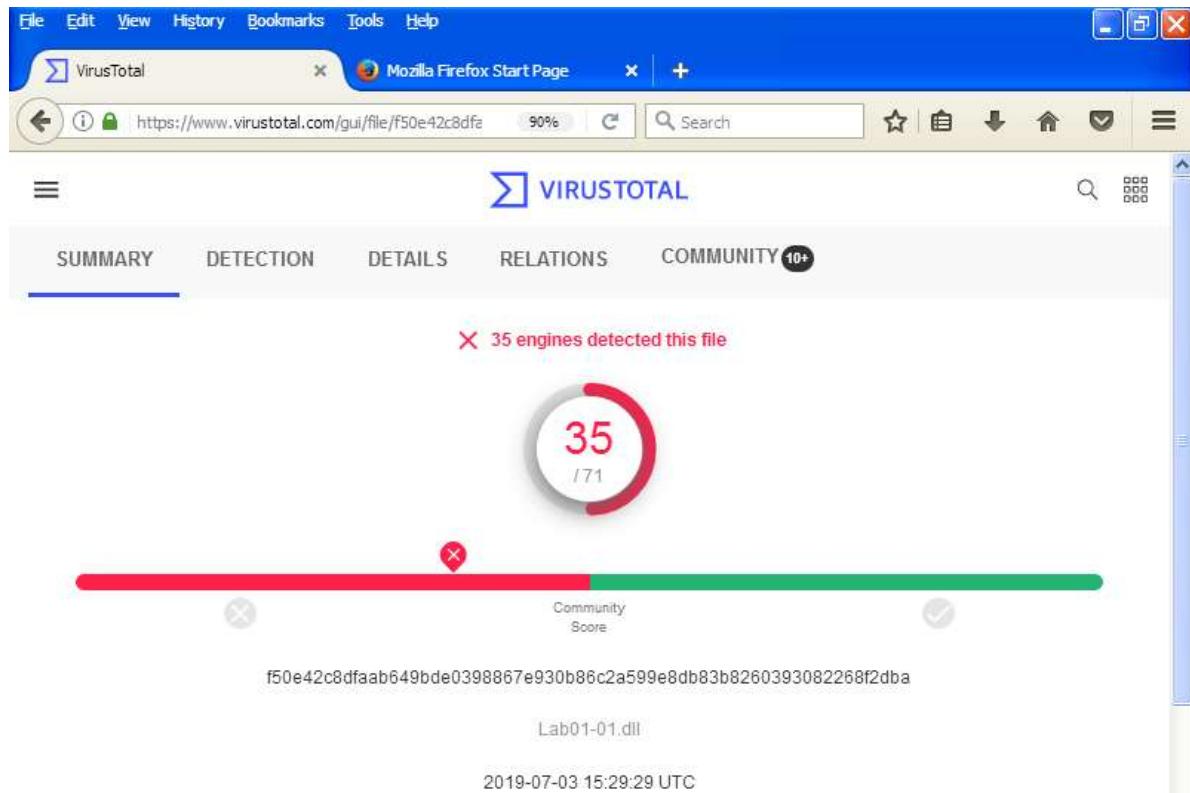
⁴⁶ Jonathon Taaffe (2019) Table 15. Practical Malware Analysis Exercises [Created 20th May 2019].

Chapter 1: Basic Static Techniques - Lab1-1

Source Files URL: <https://github.com/mikesiko/PracticalMalwareAnalysis-Labs>⁴⁷

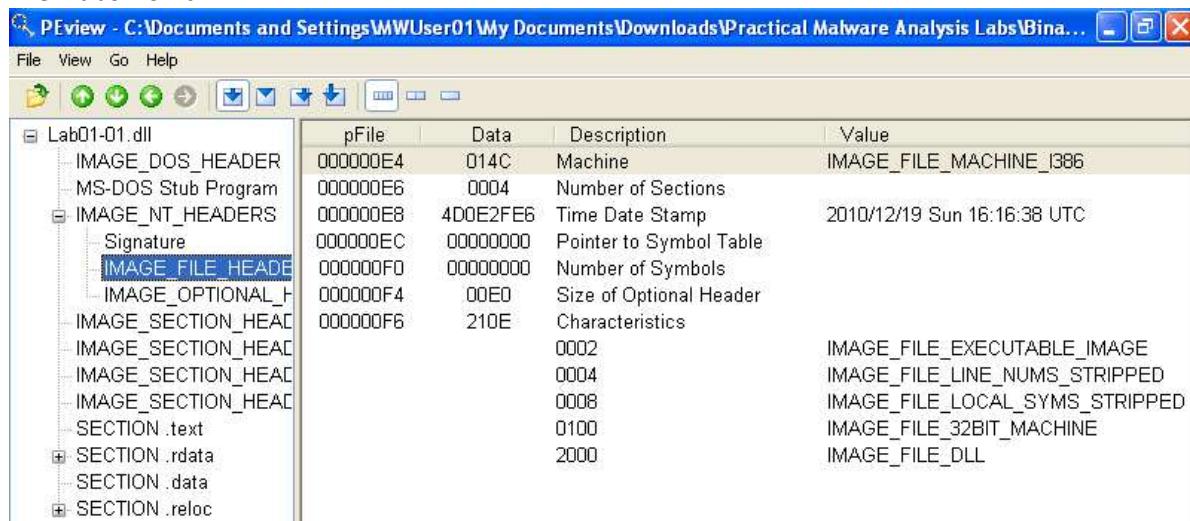
Virus Total⁴⁸ Analysis

File: Lab01-01.dll



PEView⁴⁹ Analysis

File: Lab01-01.dll



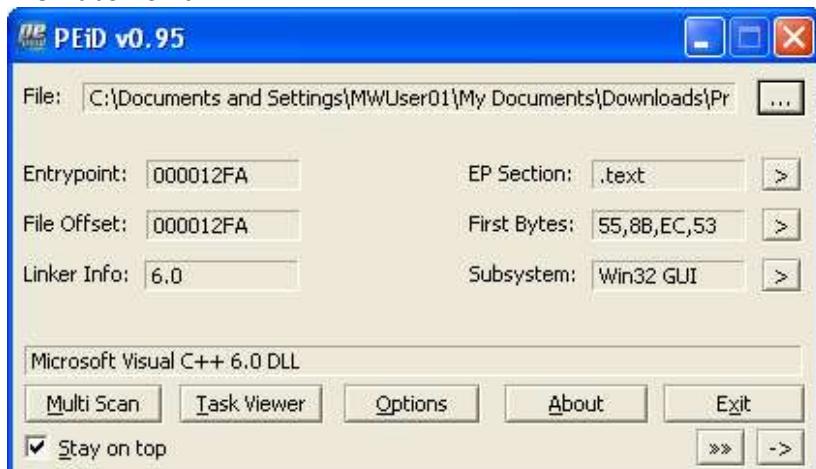
⁴⁷ PracticalMalwareAnalysis-Labs (2017) *Binaries for the book Practical Malware Analysis* <https://github.com/mikesiko/PracticalMalwareAnalysis-Labs> [Accessed 22nd May 2019].

⁴⁸ Virus Total (2019) *Virus Total* <https://www.virustotal.com> [Accessed 22nd May 2019].

⁴⁹ Radburn, Wayne J. (2018) *PEview version 0.9.9 (.zip 31KB)* <http://wjradbun.com/software/> [Accessed 22nd May 2019].

PEiD⁵⁰ Analysis

File: Lab01-01.dll



BinText3.03⁵¹ Analysis

File: Lab01-01.dll

The screenshot shows the BinText 3.0.3 interface. The main window displays a table of memory dump information. The columns are File pos, Mem pos, ID, and Text. The table contains numerous entries, many of which are highlighted in green. The last few entries are: .0000000000200 0000100000200 0 .rdata and .0000000000227 0000100000227 0 @.data. The bottom status bar shows Ready, AN: 36, UN: 0, RS: 0, Find, and Save buttons.

File pos	Mem pos	ID	Text
A 00000000002118	000010002118	0	Sleep
A 00000000002120	000010002120	0	CreateProcessA
A 00000000002132	000010002132	0	CreateMutexA
A 00000000002142	000010002142	0	OpenMutexA
A 0000000000214E	00001000214E	0	KERNEL32.dll
A 0000000000215C	00001000215C	0	WS2_32.dll
A 0000000000216A	00001000216A	0	strcmp
A 00000000002172	000010002172	0	MSVCRT.dll
A 00000000002180	000010002180	0	free
A 00000000002188	000010002188	0	_initterm
A 00000000002194	000010002194	0	malloc
A 0000000000219E	00001000219E	0	_adjust_fdiv
A 000000000026010	000010026010	0	exec
A 000000000026018	000010026018	0	sleep
A 000000000026020	000010026020	0	hello
A 000000000026028	000010026028	0	127.26.152.13
A 000000000026038	000010026038	0	SADFHUHF
A 000000000027008	000010027008	0	/010[0h0p0
A 000000000027029	000010027029	0	141G1[11]
A 000000000027039	000010027039	0	1Y2a2g2r2
A 000000000027058	000010027058	0	3 3 3
A 000000000004D	00001000004D	0	!This program cannot be run in DOS mode.
A 00000000000C0	0000100000C0	0	Rich
A 00000000001D8	0000100001D8	0	.text
A 0000000000200	000010000200	0	.rdata
A 0000000000227	000010000227	0	@.data

IP Address: 127.26.152.13

⁵⁰ Softpedia.com (2019) PEiD 0.95 <https://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/PEiD-updated.shtml> [Accessed 22nd May 2019].

⁵¹ Softpedia.com (2019) BinText 3.0.3 <https://www.softpedia.com/get/System/File-Management/BinText.shtml> [Accessed 22nd May 2019].

Dependency Walker 2.2⁵² Analysis

File: Lab01-01.dll

The screenshot shows the Dependency Walker 2.2 interface. The left pane displays a tree view of the DLL's dependencies, starting with LAB01-01.DLL, which depends on KERNEL32.DLL, WS2_32.DLL, ADVAPI32.DLL, SECUR32.DLL, and NETAPI32.DLL. ADVAPI32.DLL has its own dependencies: KERNEL32.DLL, NTDLL.DLL, RPCRT4.DLL, WINTRUST.DLL, and NETAPI32.DLL. The right pane contains two tables: one for exports (PI) and one for imports (E). The imports table highlights several functions: accept, bind, closesocket, connect, getpeername, and getsockname. Below these tables is a table of modules, showing details like file and link times, sizes, and checksums. At the bottom, there are two warning messages: "Warning: At least one delay-load dependency module was not found." and "Warning: At least one module has an unresolved import due to a missing export function in a delay-load dependent module."

Module	File Time Stamp	Link Time Stamp	File Size	Attr.	Link Checksum	Real Checksum	CPU	Subsystem	Symb...
MSJAVA.DLL	Error opening file. The system cannot find the file specified (2).								
MPR.DLL	14/04/2008 05:41	14/04/2008 01:10	59,904	A	0x00013C87	0x00013C87	x86	Console	CV
ADVAPI32.DLL	14/04/2008 05:41	14/04/2008 01:09	617,472	A	0x0009B625	0x0009B625	x86	Console	CV
KERNEL32.DLL	14/04/2008 05:41	14/04/2008 01:11	989,696	A	0x000F44A2	0x000F44A2	x86	Console	CV
LAB01-01.DLL	19/12/2010 11:16	19/12/2010 17:16	163,840	A	0x00000000	0x000327BE	x86	GUI	None

PEView⁵³ Analysis

Lab01-01.exe

The screenshot shows the PEView 0.9.9 interface. The left pane displays the file structure of Lab01-01.exe, including sections like IMAGE_DOS_HEADER, IMAGE_NT_HEADERS, and various sections (.text, .rdata, .data). The right pane shows a detailed table of the IMAGE_NT_HEADERS structure, listing fields such as Machine, Number of Sections, Time Date Stamp, Pointer to Symbol Table, Number of Symbols, Size of Optional Header, and Characteristics, along with their corresponding values.

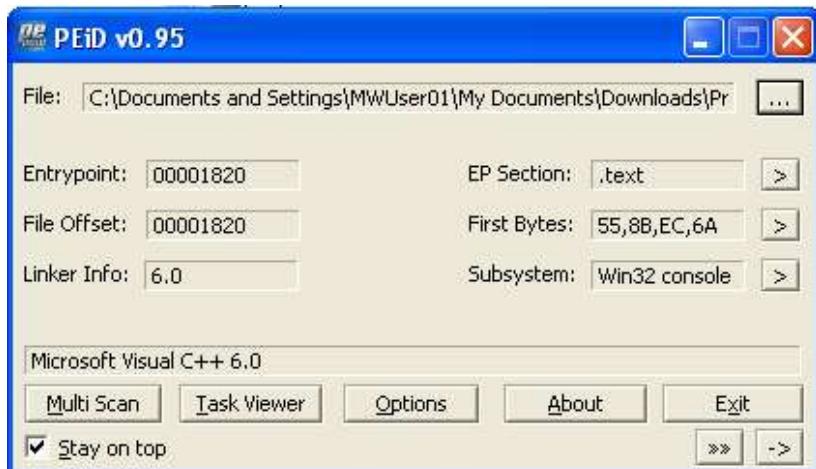
pFile	Data	Description	Value
000000EC	014C	Machine	IMAGE_FILE_MACHINE_I386
000000EE	0003	Number of Sections	
000000F0	4D0E2FD3	Time Date Stamp	2010/12/19 Sun 16:16:19 UTC
000000F4	00000000	Pointer to Symbol Table	
000000F8	00000000	Number of Symbols	
000000FC	00E0	Size of Optional Header	
000000FE	010F	Characteristics	
	0001		IMAGE_FILE_RELOCS_STRIPPED
	0002		IMAGE_FILE_EXECUTABLE_IMAGE
	0004		IMAGE_FILE_LINE_NUMS_STRIPPED
	0008		IMAGE_FILE_LOCAL_SYMS_STRIPPED
	0100		IMAGE_FILE_32BIT_MACHINE

⁵² Dependency Walker 2.2 (2019) *Dependency Walker 2.2* <http://www.dependencywalker.com/> [Accessed 22nd May 2019].

⁵³ Radburn, Wayne J. (2018) *PEview version 0.9.9 (.zip 31KB)* <http://wjradbun.com/software/> [Accessed 22nd May 2019].

PEiD⁵⁴ Analysis

File: Lab01-01.exe



BinText3.03⁵⁵ Analysis

File: Lab01-01.exe

The screenshot shows the BinText 3.0.3 application window. The interface includes a search bar at the top with 'Search', 'Filter', and 'Help' buttons. Below the search bar is a file selection field 'File to scan' set to 'C:\Documents and Settings\MWUser01\My Documents\Downloads\Practical Malware Analysis Labs\B...' with a 'Browse' button and a 'Go' button. A status message indicates 'Time taken : 0.000 secs Text size: 615 bytes (0.60K)'. The main area is a table titled 'Advanced view' with columns 'File pos', 'Mem pos', 'ID', and 'Text'. The table lists numerous memory locations and their corresponding assembly or text strings, such as 'FindNextFileA', 'FindFirstFileA', 'CopyFileA', 'KERNEL32.dll', 'malloc', 'exit', 'MSVCRT.dll', '_exit', '_XcptFilter', '_p__initenv', '_getmainargs', '_initterm', '_setusermatherr', '_adjust_fdiv', '_p__commode', '_p__fmode', '_set_app_type', '_except_handler3', '_controlfp', '_strcmp', 'kernel32.dll', 'kernel32.dll', '.exe', 'C:*', 'C:\windows\system32\kernel32.dll', and 'Kernel32.'. At the bottom of the table are buttons for 'Ready', 'AN: 63', 'UN: 0', 'RS: 0', 'Find', and 'Save'.

File pos	Mem pos	ID	Text
A 000000002196	000000402196	0	FindNextFileA
A 0000000021A6	0000004021A6	0	FindFirstFileA
A 0000000021B8	0000004021B8	0	CopyFileA
A 0000000021C2	0000004021C2	0	KERNEL32.dll
A 0000000021D2	0000004021D2	0	malloc
A 0000000021DC	0000004021DC	0	exit
A 0000000021E2	0000004021E2	0	MSVCRT.dll
A 0000000021F0	0000004021F0	0	_exit
A 0000000021F8	0000004021F8	0	_XcptFilter
A 000000002206	000000402206	0	_p__initenv
A 000000002216	000000402216	0	_getmainargs
A 000000002226	000000402226	0	_initterm
A 000000002232	000000402232	0	_setusermatherr
A 000000002246	000000402246	0	_adjust_fdiv
A 000000002256	000000402256	0	_p__commode
A 000000002266	000000402266	0	_p__fmode
A 000000002274	000000402274	0	_set_app_type
A 000000002286	000000402286	0	_except_handler3
A 000000002294	000000402294	0	_controlfp
A 0000000022A8	0000004022A8	0	_strcmp
A 000000003010	000000403010	0	kernel32.dll
A 000000003020	000000403020	0	kernel32.dll
A 000000003030	000000403030	0	.exe
A 000000003044	000000403044	0	C:*
A 00000000304C	00000040304C	0	C:\windows\system32\kernel32.dll
A 000000003070	000000403070	0	Kernel32.

⁵⁴ Softpedia.com (2019) PEiD 0.95 <https://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/PEiD-updated.shtml> [Accessed 22nd May 2019].

⁵⁵ Softpedia.com (2019) BinText 3.03 <https://www.softpedia.com/get/System/File-Management/BinText.shtml> [Accessed 22nd May 2019].

Dependency Walker 2.2⁵⁶ Analysis

File: Lab01-01.exe

The screenshot shows the Dependency Walker 2.2 interface. The main window title is "Dependency Walker - [Lab01-01.exe]". The menu bar includes File, Edit, View, Options, Profile, Window, Help. The toolbar has icons for opening files, saving, printing, and other common operations. The left pane displays a tree view of the executable's dependencies:

- LAB01-01.EXE
 - KERNEL32.DLL
 - MSVCRT.DLL

Below the tree view are two tables of exported functions:

PI	Ordinal ^	Hint	Function	Entry Point
[green]	N/A	53 (0x0035)	CreateFileMappingA	Not Bound
[green]	N/A	144 (0x0090)	FindClose	Not Bound
[green]	N/A	148 (0x0094)	FindFirstFileA	Not Bound
[green]	N/A	157 (0x009D)	FindNextFileA	Not Bound
[green]	N/A	437 (0x01B5)	IsBadReadPtr	Not Bound
[green]	N/A	470 (0x01D6)	MapViewOfFile	Not Bound

E	Ordinal ^	Hint	Function	Entry Point
[blue]	1 (0x0001)	0 (0x0000)	ActivateActCtx	0x0000AED4
[blue]	2 (0x0002)	1 (0x0001)	AddAtomA	0x00035505
[blue]	3 (0x0003)	2 (0x0002)	AddAtomW	0x000326D9
[blue]	4 (0x0004)	3 (0x0003)	AddConsoleAliasA	0x00071CDF
[blue]	5 (0x0005)	4 (0x0004)	AddConsoleAliasW	0x00071CA1

The bottom section shows a table of loaded modules:

Module	File Time Stamp	Link Time Stamp	File Size	Attr.	Link Checksum	Real Checksum	CPU	Subsystem	Symbols
KERNEL32.DLL	14/04/2008 05:41	14/04/2008 01:11	989,696	A	0x000F44A2	0x000F44A2	x86	Console	CV
LAB01-01.EXE	08/01/2012 02:19	19/12/2010 17:16	16,384	A	0x00000000	0x0007428	x86	Console	None
MSVCRT.DLL	14/04/2008 05:42	14/04/2008 01:12	343,040	A	0x00057341	0x00057341	x86	GUI	CV
NTDLL.DLL	14/04/2008 05:41	14/04/2008 01:11	706,048	A	0x000B62BC	0x000B62BC	x86	Console	CV

At the bottom of the interface, there is a status bar with the text "For Help, press F1".

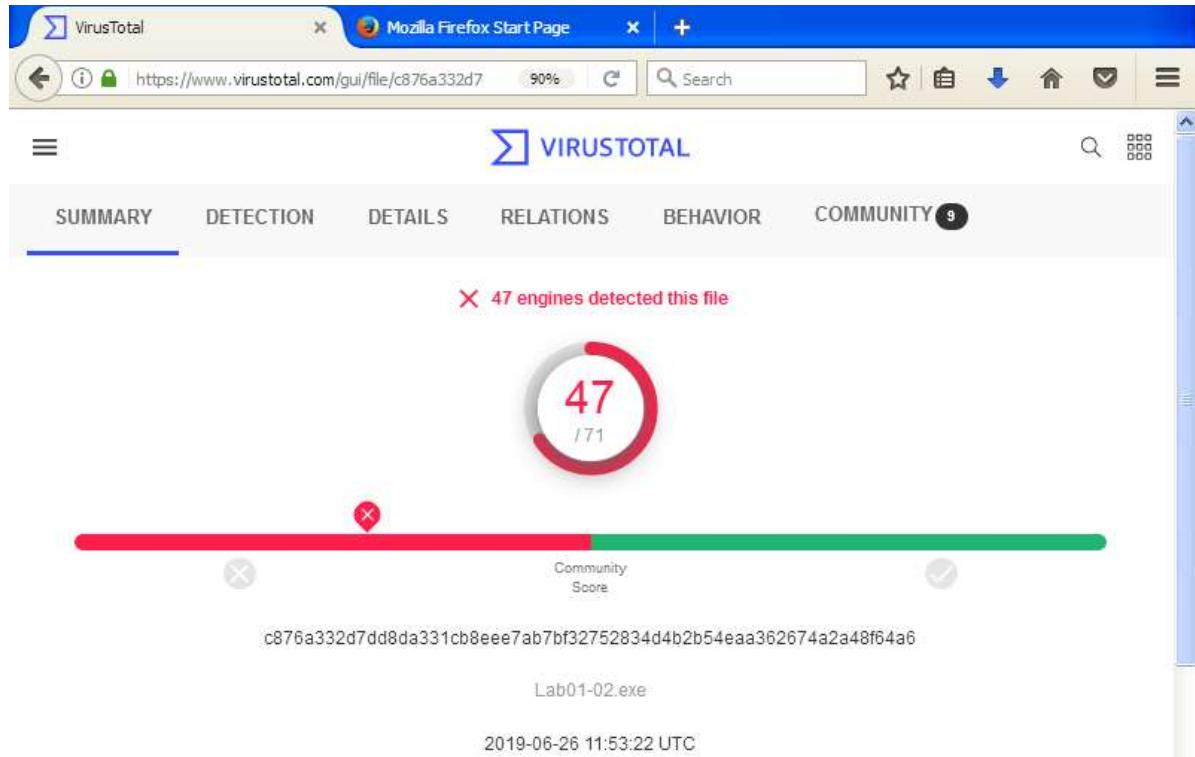
⁵⁶ Dependency Walker 2.2 (2019) *Dependency Walker 2.2* <http://www.dependencywalker.com/> [Accessed 22nd May 2019].

Chapter 1: Basic Static Techniques - Lab1-2

Source Files URL: <https://github.com/mikesiko/PracticalMalwareAnalysis-Labs>⁵⁷

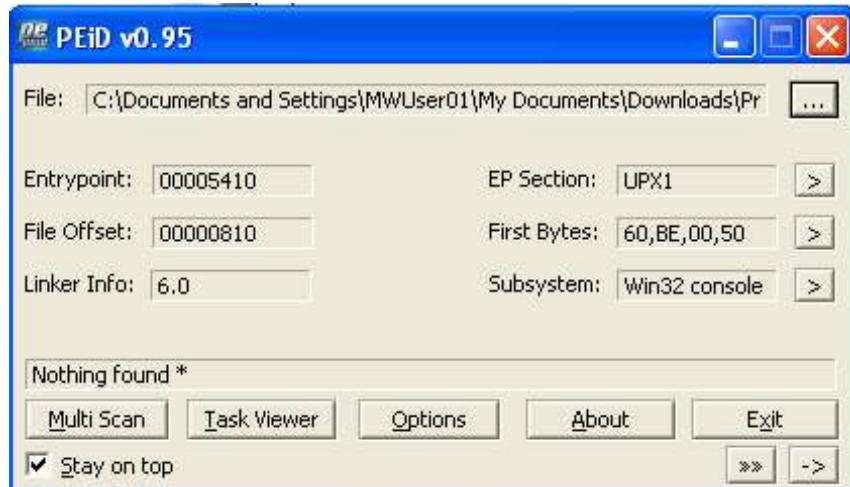
VirusTotal⁵⁸ Analysis

File: Lab01-02.exe



PEiD⁵⁹ Analysis

File: Lab01-02.exe



Nothing found *

⁵⁷ PracticalMalwareAnalysis-Labs (2017) *Binaries for the book Practical Malware Analysis* <https://github.com/mikesiko/PracticalMalwareAnalysis-Labs> [Accessed 22nd May 2019].

⁵⁸ Virus Total (2019) *Virus Total* <https://www.virustotal.com> [Accessed 22nd May 2019].

⁵⁹ Softpedia.com (2019) *PEiD 0.95* <https://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/PEiD-updated.shtml> [Accessed 22nd May 2019].

UPX 3.95⁶⁰ Analysis

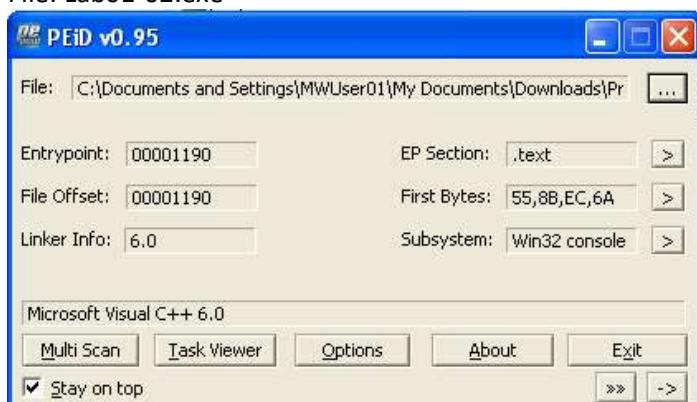
File: Lab01-02.exe

```
C:\Documents and Settings\MWUser01\My Documents\Downloads\Practical Malware Analysis Labs\BinaryCollection\Chapter_1L>upx.exe -d -o Lab01-02-unpacked.exe Lab01-02.exe
          Ultimate Packer for eXecutables
          Copyright (C) 1996 - 2018
UPX 3.95w      Markus Oberhumer, Laszlo Molnar & John Reiser   Aug 26th 2018
----- File size ----- Ratio ----- Format ----- Name -----
  16384 <-    3072    18.75%    win32/pe    Lab01-02-unpacked.exe
-----
```

Unpacked 1 file.

PEiD⁶¹ Analysis

File: Lab01-02.exe



Microsoft Visual C++ 6.0

Dependency Walker⁶² Analysis

File: Lab01-02.exe

The screenshot shows the Dependency Walker interface. The left pane displays the module hierarchy: LAB01-02-UNPACKED.EXE depends on KERNEL32.DLL, ADVAPI32.DLL, MSVCRT.DLL, and WININET.DLL. The right pane shows the imported functions for each module. For example, the InternetOpenA function is imported from KERNEL32.DLL at ordinal 0x0000. The bottom pane shows a table of loaded modules with their file and link times, sizes, checksums, and CPU subsystems.

Module	File Time Stamp	Link Time Stamp	File Size	Attr.	Link Checksum	Real Checksum	CPU	Subsystem
MSJAVA.DLL	Error opening file. The system cannot find the file specified (2).							
MPR.DLL	14/04/2008 05:41	14/04/2008 01:10	59,904	A	0x00013C87	0x00013C87	x86	Console
ADVAPI32.DLL	14/04/2008 05:41	14/04/2008 01:09	617,472	A	0x0009B625	0x0009B625	x86	Console
CRYPT32.DLL	14/04/2008 05:41	14/04/2008 01:09	599,040	A	0x0009C530	0x0009C530	x86	GUI
GDI32.DLL	14/04/2008 05:41	14/04/2008 01:09	285,184	A	0x000472FF	0x000472FF	x86	Console

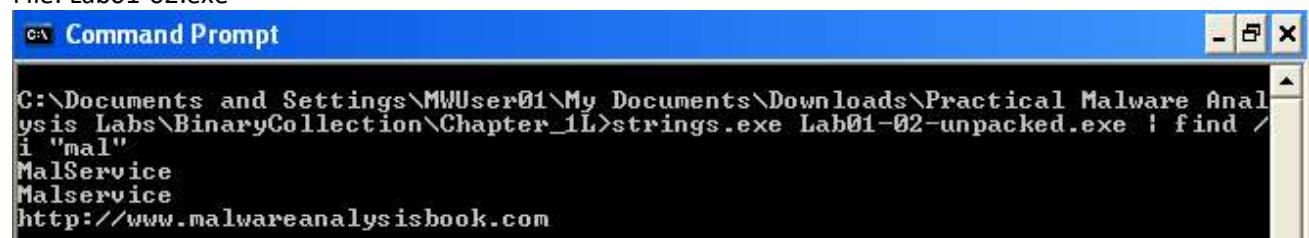
⁶⁰ UPX Ultimate Packer for eXecutables (2019) UPX 3.95 <https://github.com/upx/upx/releases/tag/v3.95> [Accessed 22nd May 2019].

⁶¹ Softpedia.com (2019) PEiD 0.95 <https://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/PEiD-updated.shtml> [Accessed 22nd May 2019].

⁶² Dependency Walker 2.2 (2019) Dependency Walker 2.2 <http://www.dependencywalker.com/> [Accessed 22nd May 2019].

Strings⁶³ Analysis

File: Lab01-02.exe



```
C:\Documents and Settings\MWUser01\My Documents\Downloads\Practical Malware Analysis Labs\BinaryCollection\Chapter_1L>strings.exe Lab01-02-unpacked.exe | find /i "mal"
MalService
Malservice
http://www.malwareanalysisbook.com
```

⁶³ Microsoft.com (2019) *Strings* v2.53 <https://docs.microsoft.com/en-gb/sysinternals/downloads/strings> [Accessed 22nd May 2019].

Chapter 3: Basic Dynamic Analysis - Lab3-1

File: Lab03-01.exe

PEview⁶⁴ Analysis

File: Lab03-01.exe

The screenshot shows the PEview interface with the file 'Lab03-01.exe' loaded. The left pane displays the file's structure, including the IMAGE_DOS_HEADER, MS-DOS Stub Program, IMAGE_NT_HEADERS (with its sub-sections: Signature, IMAGE_FILE_HEADER, IMAGE_OPTIONAL_HEADER, IMAGE_SECTION_HEADER, IMAGE_SECTION_HEADER), SECTION .text (containing the IMPORT Address Table, IMPORT Directory Table, IMPORT Name Table, and IMPORT Hints/Names & I), and SECTION .data. The right pane shows a table for the IMPORT Address Table:

pFile	Data	Description	Value
00000200	0000024C	Hint/Name RVA	0080 ExitProcess
00000204	00000000	End of Imports	kernel32.dll

Strings⁶⁵ Analysis

The screenshot shows a Windows Command Prompt window titled 'Administrator: C:\Windows\system32\cmd.exe'. The command 'strings Lab03-01.exe' has been run, displaying various strings found in the binary. The output includes:

```
C:\Users\Administrator\Downloads\Practical Malware Analysis Labs\BinaryCollection\Chapter_3L>strings Lab03-01.exe

Strings v2.53 - Search for ANSI and Unicode strings in binary images.
Copyright (C) 1999-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

!This program cannot be run in DOS mode.

Rich
.text
`data
ExitProcess
kernel32.dll
ws2_32
A>!
~
"p7
cks=u
ttt=
cks=
CONNECT zzz HTTP/1.0
```

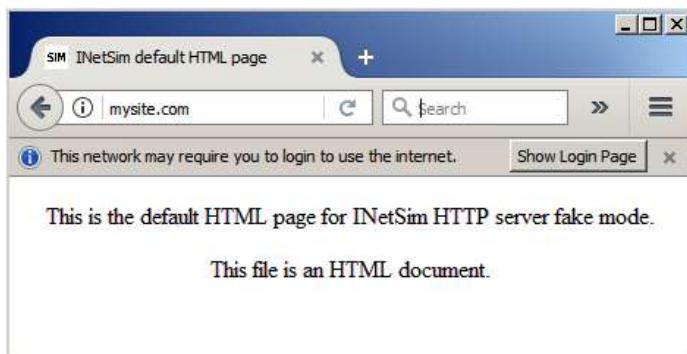
⁶⁴ Radburn, Wayne J. (2018) PEview version 0.9.9 (.zip 31KB) <http://wiradburn.com/software/> [Accessed 22nd May 2019].

⁶⁵ Microsoft.com (2019) Strings v2.53 <https://docs.microsoft.com/en-gb/sysinternals/downloads/strings> [Accessed 22nd May 2019].

```
Administrator: C:\Windows\system32\cmd.exe
StubPath
SOFTWARE\Classes\http\shell\open\command\0
Software\Microsoft\Active Setup\Installed Components\
test
www.practicalmalwareanalysis.com
admin
VideoDriver
WinUMX32-
vmx32to64.exe
SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Ph?
U5h
U>U
SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
PWj
AppData
j@h
UQj
UiW
UzX_
```

Dynamic Analysis

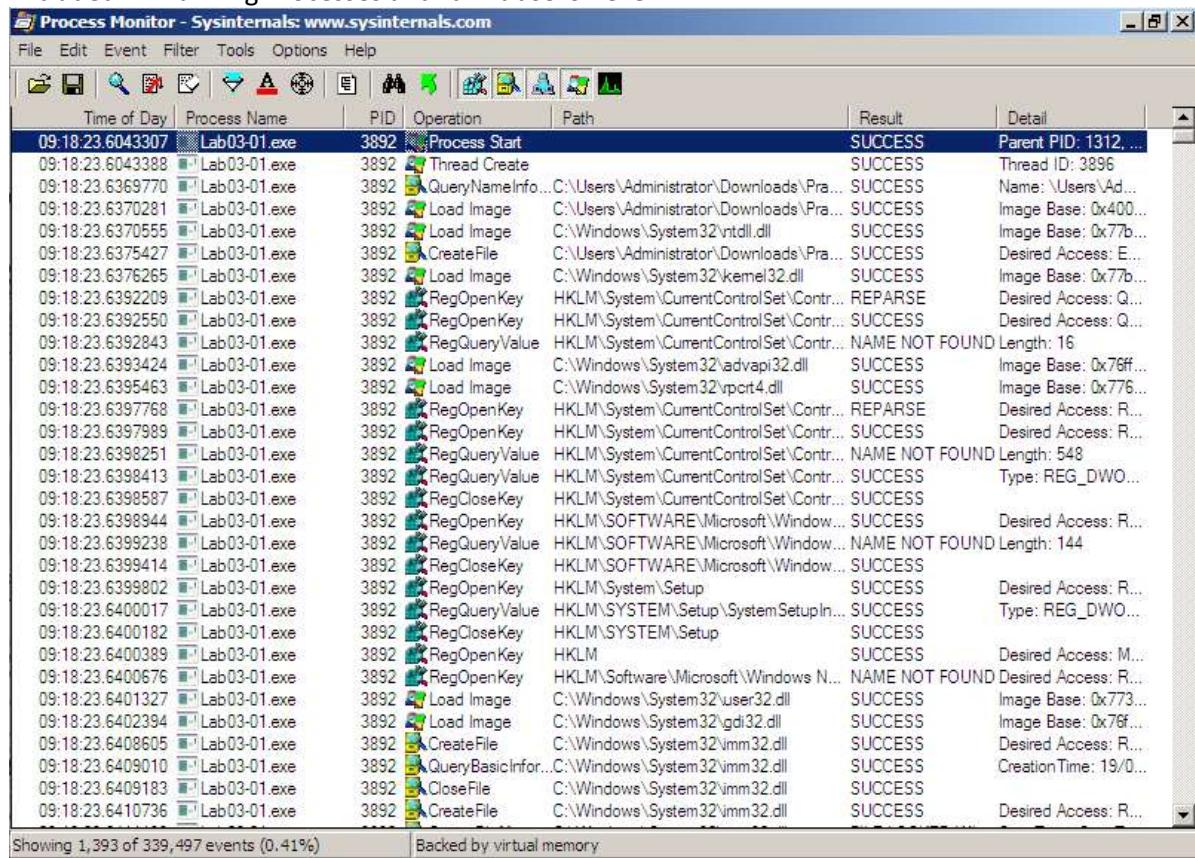
1. Started iNetSim⁶⁶ on Kali KDE 2019
2. On Windows Server 2008 set Primary DNS to 10.0.2.15 (Kali IP)
3. From Windows Server 2008 browsed through Firefox to <http://mysite.com>
4. Received the iNetSim HTTP server fake mode page.



⁶⁶ iNetSim.org (2018) INetSim: Internet Services Simulation Suite <https://www.inetsim.org/> [Accessed 20th May 2019].

Process Explorer⁶⁷ Analysis

Excluded All Running Processes and ran Lab03-01.exe



The screenshot shows the Process Monitor application interface. The title bar reads "Process Monitor - Sysinternals: www.sysinternals.com". The menu bar includes File, Edit, Event, Filter, Tools, Options, and Help. Below the menu is a toolbar with various icons. The main window is a grid table with columns: Time of Day, Process Name, PID, Operation, Path, Result, and Detail. The table lists many events for the process "Lab03-01.exe" with PID 3892. The operations include Process Start, Thread Create, QueryNameInfo, Load Image, CreateFile, RegOpenKey, RegQueryValue, RegCloseKey, and RegOpenKey. Paths mostly involve system and user DLLs like ntdll.dll, kernel32.dll, advapi32.dll, and user32.dll. Results are mostly SUCCESS, with some NAME NOT FOUND errors. Details column shows parent PID (1312), thread ID, file names, and desired access levels (e.g., Read, Write, Execute). At the bottom, status bars show "Showing 1,393 of 339,497 events (0.41%)" and "Backed by virtual memory".

Time of Day	Process Name	PID	Operation	Path	Result	Detail
09:18:23.6043307	Lab03-01.exe	3892	Process Start		SUCCESS	Parent PID: 1312, ...
09:18:23.6043388	Lab03-01.exe	3892	Thread Create		SUCCESS	Thread ID: 3896
09:18:23.6369770	Lab03-01.exe	3892	QueryNameInfo	C:\Users\Administrator\Downloads\Pr... SUCCESS		Name: \Users\Ad...
09:18:23.6370281	Lab03-01.exe	3892	Load Image	C:\Users\Administrator\Downloads\Pr... SUCCESS		Image Base: 0x400...
09:18:23.6370555	Lab03-01.exe	3892	Load Image	C:\Windows\System32\ntdll.dll	SUCCESS	Image Base: 0x77b...
09:18:23.6375427	Lab03-01.exe	3892	CreateFile	C:\Users\Administrator\Downloads\Pr... SUCCESS		Desired Access: E...
09:18:23.6376265	Lab03-01.exe	3892	Load Image	C:\Windows\System32\kernel32.dll	SUCCESS	Image Base: 0x77b...
09:18:23.6392209	Lab03-01.exe	3892	RegOpenKey	HKLM\System\CurrentControlSet\Contr... REPARSE		Desired Access: Q...
09:18:23.6392550	Lab03-01.exe	3892	RegOpenKey	HKLM\System\CurrentControlSet\Contr... SUCCESS		Desired Access: Q...
09:18:23.6392843	Lab03-01.exe	3892	RegQueryValue	HKLM\System\CurrentControlSet\Contr... NAME NOT FOUND Length: 16		
09:18:23.6393424	Lab03-01.exe	3892	Load Image	C:\Windows\System32\advapi32.dll	SUCCESS	Image Base: 0x76f...
09:18:23.6395463	Lab03-01.exe	3892	Load Image	C:\Windows\System32\pcrit4.dll	SUCCESS	Image Base: 0x776...
09:18:23.6397768	Lab03-01.exe	3892	RegOpenKey	HKLM\System\CurrentControlSet\Contr... REPARSE		Desired Access: R...
09:18:23.6397989	Lab03-01.exe	3892	RegOpenKey	HKLM\System\CurrentControlSet\Contr... SUCCESS		Desired Access: R...
09:18:23.6398251	Lab03-01.exe	3892	RegQueryValue	HKLM\System\CurrentControlSet\Contr... NAME NOT FOUND Length: 548		
09:18:23.6398413	Lab03-01.exe	3892	RegQueryValue	HKLM\System\CurrentControlSet\Contr... SUCCESS		Type: REG_DWO...
09:18:23.6398587	Lab03-01.exe	3892	RegCloseKey	HKLM\System\CurrentControlSet\Contr... SUCCESS		
09:18:23.6398944	Lab03-01.exe	3892	RegOpenKey	HKLM\Software\Microsoft\Window... SUCCESS		Desired Access: R...
09:18:23.6399238	Lab03-01.exe	3892	RegQueryValue	HKLM\Software\Microsoft\Window... NAME NOT FOUND Length: 144		
09:18:23.6399414	Lab03-01.exe	3892	RegCloseKey	HKLM\Software\Microsoft\Window... SUCCESS		
09:18:23.6399802	Lab03-01.exe	3892	RegOpenKey	HKLM\System\Setup	SUCCESS	Desired Access: R...
09:18:23.6400017	Lab03-01.exe	3892	RegQueryValue	HKLM\SYSTEM\Setup\SystemSetupIn... SUCCESS		Type: REG_DWO...
09:18:23.6400182	Lab03-01.exe	3892	RegCloseKey	HKLM\SYSTEM\Setup	SUCCESS	
09:18:23.6400389	Lab03-01.exe	3892	RegOpenKey	HKLM	SUCCESS	Desired Access: M...
09:18:23.6400676	Lab03-01.exe	3892	RegOpenKey	HKLM\Software\Microsoft\Windows N... NAME NOT FOUND Desired Access: R...		
09:18:23.6401327	Lab03-01.exe	3892	Load Image	C:\Windows\System32\user32.dll	SUCCESS	Image Base: 0x773...
09:18:23.6402394	Lab03-01.exe	3892	Load Image	C:\Windows\System32\gdi32.dll	SUCCESS	Image Base: 0x76f...
09:18:23.6408605	Lab03-01.exe	3892	CreateFile	C:\Windows\System32\imm32.dll	SUCCESS	Desired Access: R...
09:18:23.6409010	Lab03-01.exe	3892	QueryBasicInfor...	C:\Windows\System32\imm32.dll	SUCCESS	CreationTime: 19/0...
09:18:23.6409183	Lab03-01.exe	3892	CloseFile	C:\Windows\System32\imm32.dll	SUCCESS	
09:18:23.6410736	Lab03-01.exe	3892	CreateFile	C:\Windows\System32\imm32.dll	SUCCESS	Desired Access: R...

⁶⁷ Microsoft.com (2019) *Process Explorer v16.26* <https://docs.microsoft.com/en-gb/sysinternals/downloads/process-explorer> [Accessed 20th May 2019].

Process Explorer > Lower Pane > Handles

Process Explorer - Sysinternals: www.sysinternals.com [WIN-U6WJ2MZR960\Administrator] (Administrator)

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
explorer.exe	1.56	26,924 K	38,416 K	1312	Windows Explorer	Microsoft Corporation
firefox.exe	< 0.01	92,944 K	117,064 K	1056	Firefox	Mozilla Corporation
firefox.exe		27,156 K	47,932 K	2084	Firefox	Mozilla Corporation
Interrupts	< 0.01	0 K	0 K	n/a	Hardware Interrupts and DPCs	
Lab03-01.exe		1,176 K	3,584 K	3892		
lsass.exe		3,184 K	8,200 K	736	Local Security Authority Proc...	Microsoft Corporation
lsm.exe		1,480 K	3,664 K	744	Local Session Manager Serv...	Microsoft Corporation
msdtc.exe		2,784 K	6,708 K	392	MS DTCConsole program	Microsoft Corporation
procesexp.exe	< 0.01	19,936 K	27,696 K	2688	Sysinternals Process Explorer	Sysinternals - www.sysinter...
Promon.exe	< 0.01	15,460 K	22,092 K	3424	Process Monitor	Sysinternals - www.sysinter...
services.exe		1,988 K	5,964 K	724	Services and Controller app	Microsoft Corporation
SLsvc.exe		4,196 K	8,872 K	1144	Microsoft Software Licensing...	Microsoft Corporation
smss.exe		252 K	688 K	528	Windows Session Manager	Microsoft Corporation

Type	Name
Desktop	\Default
Directory	\KnownDlIs
Directory	\Sessions\1\BaseNamedObjects
File	C:\Users\Administrator\Downloads\Practical Malware Analysis Labs\BinaryCollection\Chapt...
File	C:\Windows\win32e\x86_microsoft.windows.common-controls_6595b6414ccf1df_6.0.600...
File	\Device\Nsi
Key	HKLM\SYSTEM\ControlSet001\Control\Session Manager
Key	HKLM
Key	HKLM\SYSTEM\ControlSet001\Services\WinSock2\Parameters\Protocol_Catalog9
Key	HKLM\SYSTEM\ControlSet001\Services\WinSock2\Parameters\NameSpace_Catalog5
Mutant	\Sessions\1\BaseNamedObjects\WinVMX32
Thread	Lab03-01.exe(3892): 3896
Thread	Lab03-01.exe(3892): 3896
WindowStation	\Sessions\1\Windows\WindowStations\WinSta0
WindowStation	\Sessions\1\Windows\WindowStations\WinSta0

CPU Usage: 3.13% Commit Charge: 13.91% Processes: 40 Physical Usage: 32.15% [InetSim default HTML page - Mozilla Firefox]

Process Explorer > Lower Pane > DLL's

Process Explorer - Sysinternals: www.sysinternals.com [WIN-U6WJ2MZR960\Administrator] (Administrator)

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
explorer.exe	1.52	26,544 K	37,992 K	1312	Windows Explorer	Microsoft Corporation
firefox.exe		96,048 K	120,524 K	1056	Firefox	Mozilla Corporation
firefox.exe		27,156 K	47,932 K	2084	Firefox	Mozilla Corporation
Interrupts	< 0.01	0 K	0 K	n/a	Hardware Interrupts and DPCs	
Lab03-01.exe		1,176 K	3,584 K	3892		
lsass.exe		3,184 K	8,180 K	736	Local Security Authority Proc...	Microsoft Corporation
lsm.exe		1,480 K	3,664 K	744	Local Session Manager Serv...	Microsoft Corporation
msdtc.exe		2,784 K	6,708 K	392	MS DTCConsole program	Microsoft Corporation
procesexp.exe	3.03	20,016 K	28,296 K	2688	Sysinternals Process Explorer	Sysinternals - www.sysinter...
Promon.exe	< 0.01	18,132 K	25,620 K	3424	Process Monitor	Sysinternals - www.sysinter...
services.exe		1,928 K	5,936 K	724	Services and Controller app	Microsoft Corporation
SLsvc.exe		4,160 K	8,860 K	1144	Microsoft Software Licensing...	Microsoft Corporation
smss.exe		252 K	688 K	528	Windows Session Manager	Microsoft Corporation

Name	Description	Company Name	Path
oleaut32.dll		Microsoft Corporation	C:\Windows\System32\oleaut32.dll
psapi.dll	Process Status Helper	Microsoft Corporation	C:\Windows\System32\psapi.dll
rasadhlp.dll	Remote Access AutoDial Helper	Microsoft Corporation	C:\Windows\System32\rasadhlp.dll
rpcrt4.dll	Remote Procedure Call Runtime	Microsoft Corporation	C:\Windows\System32\rpcrt4.dll
secur32.dll	Security Support Provider Interface	Microsoft Corporation	C:\Windows\System32\secur32.dll
setupapi.dll	Windows Setup API	Microsoft Corporation	C:\Windows\System32\setupapi.dll
shlwapi.dll	Shell Light-weight Utility Library	Microsoft Corporation	C:\Windows\System32\shlwapi.dll
user32.dll	Multi-User Windows USER API Cli...	Microsoft Corporation	C:\Windows\System32\user32.dll
usp10.dll	Uniscribe Unicode script processor	Microsoft Corporation	C:\Windows\System32\usp10.dll
version.dll	Version Checking and File Installati...	Microsoft Corporation	C:\Windows\System32\version.dll
winnsi.dll	Network Store Information RPC int...	Microsoft Corporation	C:\Windows\System32\winnsi.dll
winmr.dll	LDAP RnR Provider DLL	Microsoft Corporation	C:\Windows\System32\winmr.dll
Wldap32.dll	Win32 LDAP API DLL	Microsoft Corporation	C:\Windows\System32\Wldap32.dll
ws2_32.dll	Windows Socket 2.0 32-Bit DLL	Microsoft Corporation	C:\Windows\System32\ws2_32.dll
WSHTCPIP.DLL	Winsock2 Helper DLL (TL/IPv4)	Microsoft Corporation	C:\Windows\System32\WSHTCPIP.DLL

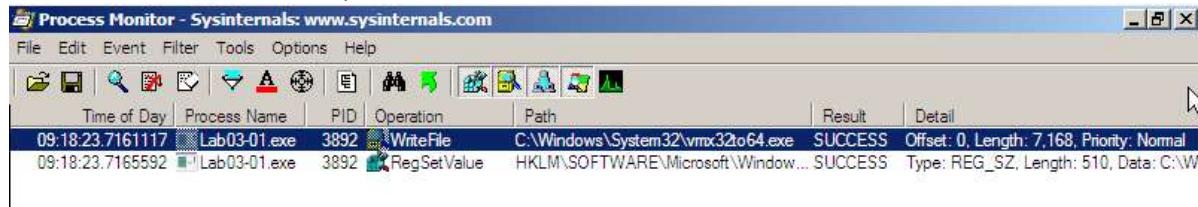
CPU Usage: 6.06% Commit Charge: 15.16% Processes: 39 Physical Usage: 32.15%

Process Monitor⁶⁸: View Malicious Process's Events

Process Monitor > Filter > Process Name > is > Lab03-01.exe > Add

Process Monitor > Filter > Operation > is > RegSetValue > Add

Process Monitor > Filter > Operation > is > WriteFile > Add



The screenshot shows the Process Monitor interface with the title bar "Process Monitor - Sysinternals: www.sysinternals.com". The menu bar includes File, Edit, Event, Filter, Tools, Options, and Help. Below the menu is a toolbar with various icons. The main window displays a table of events. The columns are: Time of Day, Process Name, PID, Operation, Path, Result, and Detail. There are two rows of data:

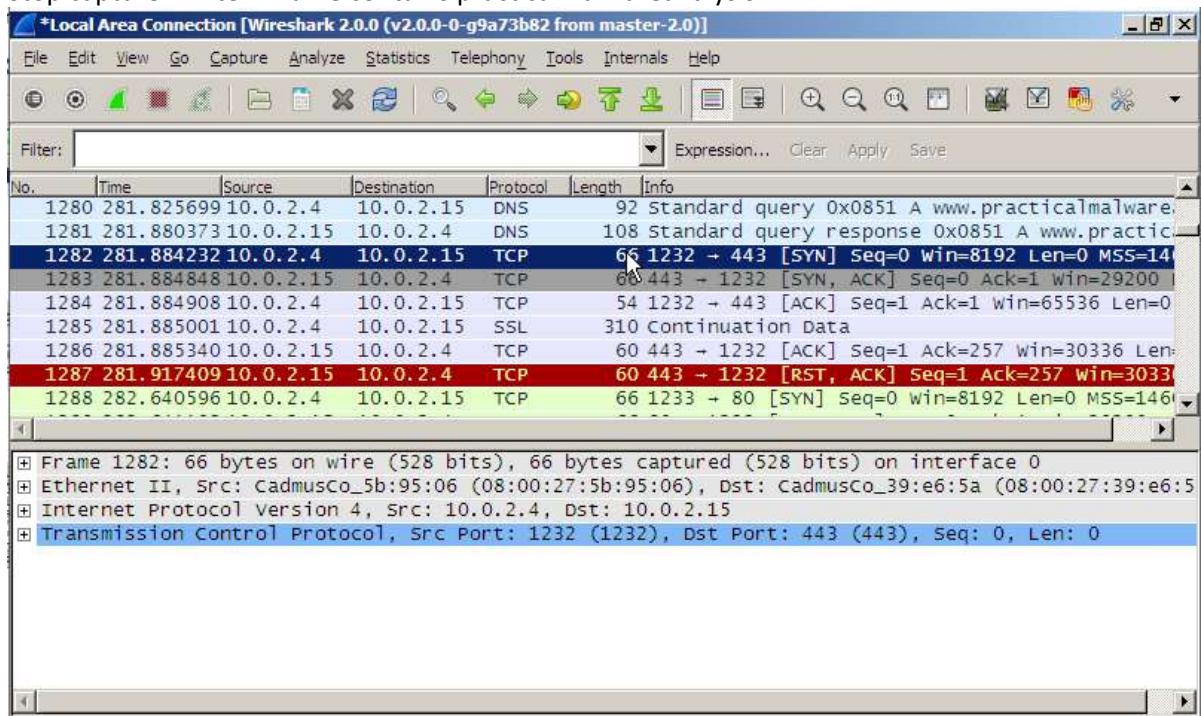
Time of Day	Process Name	PID	Operation	Path	Result	Detail
09:18:23.7161117	Lab03-01.exe	3892	WriteFile	C:\Windows\System32\vmx32to64.exe	SUCCESS	Offset: 0, Length: 7,168, Priority: Normal
09:18:23.7165592	Lab03-01.exe	3892	RegSetValue	HKLM\SOFTWARE\Microsoft\Window...	SUCCESS	Type: REG_SZ, Length: 510, Data: C:\W

1. Double-click event with Path ending vmx32to64.exe
2. Properties sheet: this event created a file named vmx32to64.exe
3. This event copied the malware itself to a file named vmx32to64.exe, so that filename is a useful indicator of infection.
4. Double-click event with Path ending VideoDriver
5. Created new Run key in registry called "VideoDriver" with a value of "C:\WINDOWS\system32\vmx32to64.exe"
6. Persistence mechanism to re-launch the malware when the machine restarts

⁶⁸ Microsoft.com (2019) *Process Monitor v3.52* <https://docs.microsoft.com/en-gb/sysinternals/downloads/procm> [Accessed 20th May 2019].

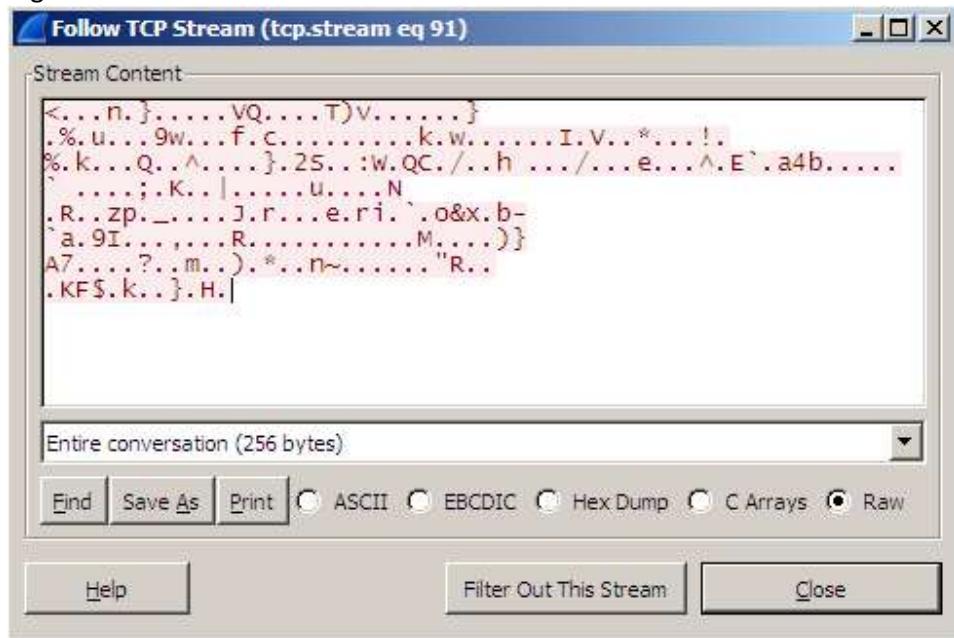
Wireshark⁶⁹ Analysis

Stop capture > Filter: 'frame contains practicalmalwareanalysis'



Packet 1282: SYN packet sent from port 1232 > https port 443

Right-click > Follow TCP Stream



Stream Content: Entire conversation (256 bytes) of random packets

These are beacons and are used by malware to notify the Command and Control server that the machine is infected and ready to use.

⁶⁹ Wireshark.org (2019) Wireshark-win32-2.0.0.exe <https://1.eu.dl.wireshark.org/win32/all-versions/> [Accessed 20th May 2019].

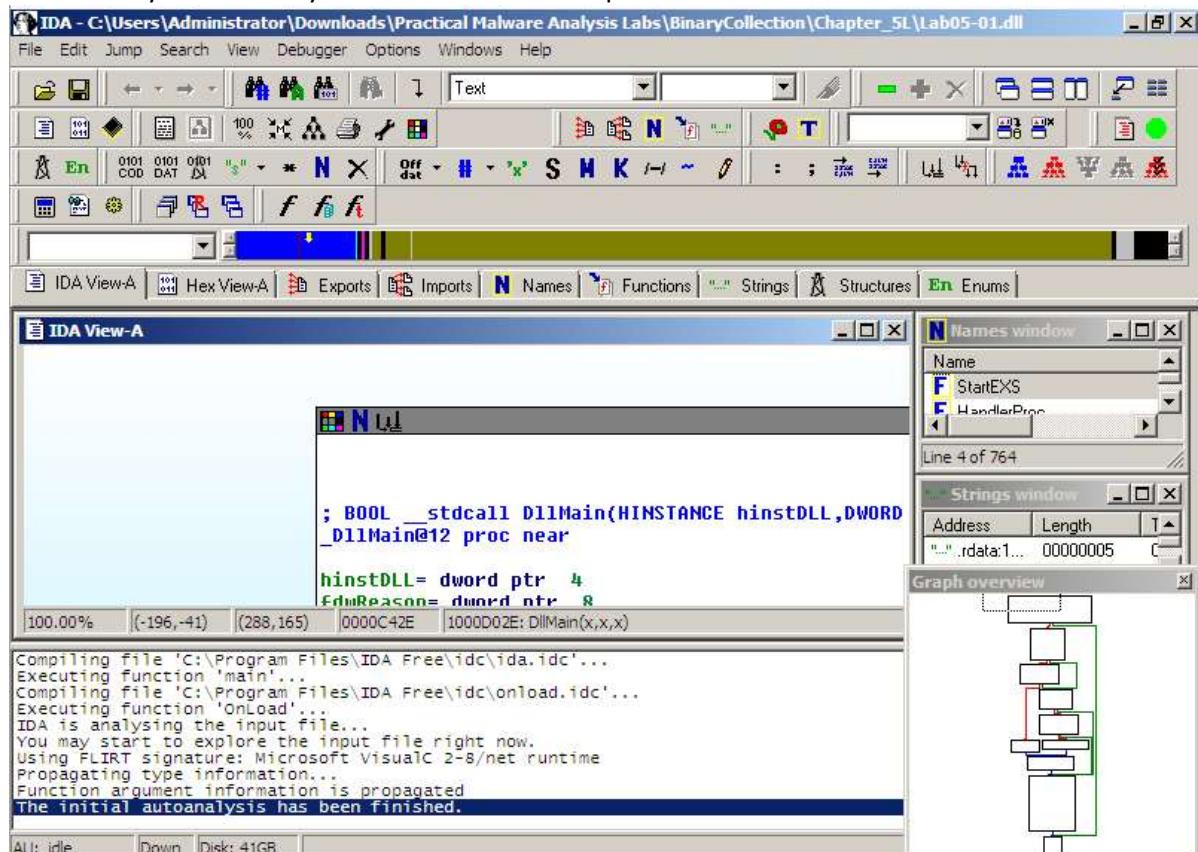
Chapter 5: Anti Reverse Engineering - Lab5-1

Source Files URL: <https://github.com/mikesiko/PracticalMalwareAnalysis-Labs>⁷⁰

File: Lab05-01.dll

IDA Pro⁷¹ Analysis

New > PE Dynamic Library > OK > Lab05-01.dll > Open



Find DLLMain Address

IDA Pro > Windows > Functions window > click "Function name" header to sort by name > locate DLLMain

Function name	Segment	Start	Length	R	F	L	S	B	T	=
BlockInput	.text	100111E2	00000006	R						
CreateToolhelp32Snapshot	.text	100111C4	00000006	R					T	
DllEntryPoint	.text	1001516D	0000009D	R		L		B	T	
DllMain(x,x,x)	.text	1000D02E	000000DF	R					T	
EnumProcessModules	.text	100111AC	00000006	R						
GetAdaptersInfo	.text	100111B2	00000006	R						
GetModuleFileNameExA	.text	100111A6	00000006	R						
HandlerProc	.text	1000C9DF	00000077	R					T	
ICClose	.text	100113D6	00000006	R					T	

⁷⁰ PracticalMalwareAnalysis-Labs (2017) *Binaries for the book Practical Malware Analysis*

<https://github.com/mikesiko/PracticalMalwareAnalysis-Labs> [Accessed 22nd May 2019].

⁷¹ Hex-Rays (2019) *IDA Freeware for Windows* (48 MB) https://www.hex-rays.com/products/ida/support/download_freeware.shtml [Accessed 22nd May 2019].

Find import for gethostbyname

IDA Pro > Windows > Imports > click Name header to sort by name > locate gethostbyname

Address	Ordinal	Name	Library
10016234		fread	MSVCRT
100162DC		free	MSVCRT
100162D8		fseek	MSVCRT
10016278		ftell	MSVCRT
100162A0		fwrite	MSVCRT
100163CC	52	gethostbyname	WS2_32
100163E4	9	htons	WS2_32
100163C8	11	inet_addr	WS2_32
100163D0	12	inet_ntoa	WS2_32

Local Variables Count for Subroutine at 0x10001656

IDA Pro > Windows > IDA View-A > press SPACEBAR for text view > press g for Go > enter address 0x10001656 > OK

IDA View-A Hex View-A Exports Imports Names Functions Strings Structures Enums	
<pre>.text:10001656 ; DWORD _stdcall sub_10001656(LPUOID) .text:10001656 sub_10001656 proc near ; DATA XREF: DllMain(x,x,x)+C8↓o .text:10001656 .text:10001656 var_675 = byte ptr -675h .text:10001656 var_674 = dword ptr -674h .text:10001656 hModule = dword ptr -670h .text:10001656 timeout = timeval ptr -66Ch .text:10001656 name = sockaddr ptr -664h .text:10001656 var_654 = word ptr -654h .text:10001656 in = in_addr ptr -650h .text:10001656 Parameter = byte ptr -644h .text:10001656 CommandLine = byte ptr -63Fh .text:10001656 Data = byte ptr -638h .text:10001656 var_544 = dword ptr -544h .text:10001656 var_50C = dword ptr -500h .text:10001656 var_500 = dword ptr -500h .text:10001656 var_4FC = dword ptr -4FCh .text:10001656 readFds = fd_set ptr -4BCh .text:10001656 phkResult = HKEY__ ptr -3B8h .text:10001656 var_380 = dword ptr -3B0h</pre>	

Purpose of \cmd.exe /c Code Reference

IDA Pro > Windows > Strings > sort by String > find String "\cmd.exe /c"

Address	Length	Type	String
"...." xdoors_d:100939A0	0000000F	C	\Device\Video0
"...." xdoors_d:100954B0	0000000C	C	\Parameters
"...." xdoors_d:10095B34	0000000D	C	\cmd.exe /c
"...." xdoors_d:10095B20	00000011	C	\Command.exe /c
"...." xdoors_d:10093844	0000000B	C	\n\n[%% %%]
"...." xdoors_d:100943C4	0000000F	C	\n%-16d%-20s%d
"...." xdoors_d:10093D50	00000023	C	\n(1) Enter Current Directory %s'
"...." xdoors_d:10093A98	00000034	C	\n(1) Enter Current Directory Error,Update Failed\n
"...." xdoors_d:10093D34	0000001C	C	\n(2) Get DLL FileName %s'
"...." xdoors_d:10093ACC	0000002D	C	\n(2) Get DLL FileName Error,Update Failed\n
"...." xdoors_d:10093AFC	00000055	C	\n(3) Move %s! To %s! Failed,Perhaps Other Process Updating!Updated ...
"...." xdoors_d:10093D04	00000025	C	\n(3) Move %s! To %s! Successfully
"...." xdoors_d:10093CE8	0000001C	C	\n(4) Get New FileName %s'
"...." xdoors_d:10093BC0	0000002D	C	\n(4) Get New FileName Error,Update Failed\n
"...." xdoors_d:10093B54	00000031	C	\n(4) Resume %s! To %s! Failed,Update Failed\n
"...." xdoors_d:10093B88	00000037	C	\n(4) Resume %s! To %s! Successfully,Update Failed\n
"...." xdoors_d:10093CC0	00000025	C	\n(5) Copy %s! To %s! Successfully
"...." xdoors_d:10093C90	0000002D	C	\n(5) Move %s! To %s! Failed,Update Failed

Double-click "\cmd.exe /c"

File View A Hex View A Exports Imports Names Functions Strings Structures Enums

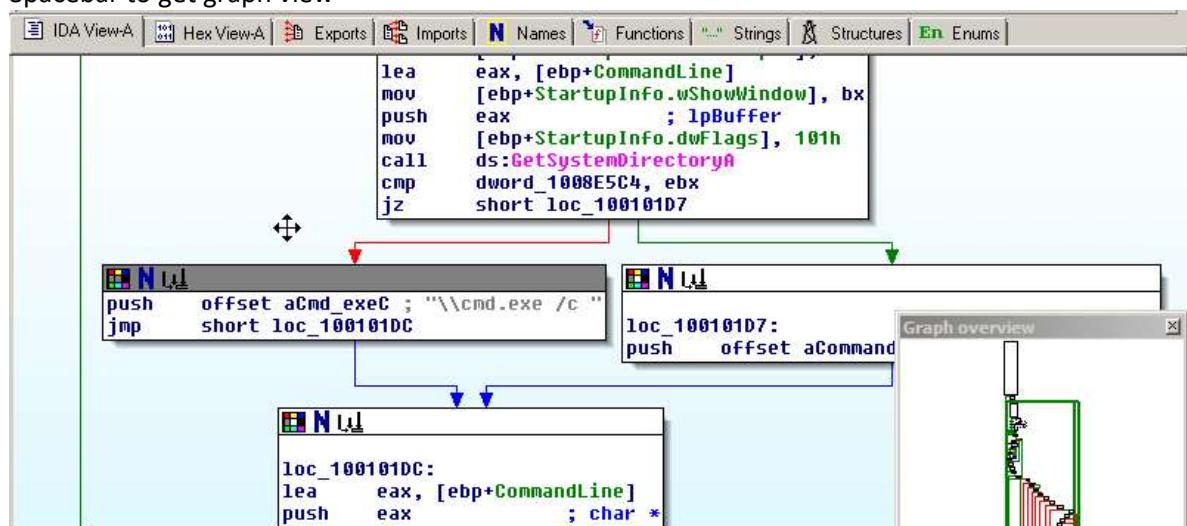
```
oors_d:10095B18 aQuit db 'quit',0 ; DATA XREF: sub_1000FF58+36F to
• oors_d:10095B1D align 10h
• oors_d:10095B20 ; char aCommand_execC
oors_d:10095B20 aCommand_execC db '\command.exe /c ',0 ; DATA XREF: sub_1000FF58:loc_10
• oors_d:10095B31 align 4
• oors_d:10095B34 aCmd_execC db '\cmd.exe /c ',0 ; DATA XREF: sub_1000FF58+278 to
• oors_d:10095B41 align 4
• oors_d:10095B44 ; char aHiMasterDDDDDD[]
oors_d:10095B44 aHiMasterDDDDDD db 'Hi,Master [%d/%d/%d %d:%d:%d]',0Dh,0Ah
oors_d:10095B44 align 4 ; DATA XREF: sub_1000FF58+145 to
oors_d:10095B44 db 'Welcome Back...Are You Enjoying Today?',0Dh,0Ah
oors_d:10095B44 db 0Dh,0Ah
oors_d:10095B44 db 'Machine UpTime [%-.2d Days %-.2d Hours %-.2d Minute
oors_d:10095B44 db 'ds],0Dh,0Ah
oors_d:10095B44 db 'Machine IdleTime [%-.2d Days %-.2d Hours %-.2d Minut
oors_d:10095B44 db 'nds]',0Dh,0Ah
oors_d:10095B44 db 0Dh,0Ah
oors_d:10095B44 db 'Encrypt Magic Number For This Remote Shell Session ['
oors_d:10095B44 db 0Dh,0Ah,0
oors_d:10095C5C ; char asc_10095C5C[]
```

Double-click address to right of XREF: sub_1000FF58+278^0

IDA View-A | Hex View-A | Exports | Imports | Names | Functions | Strings | Structures | Enums

• 100101BA	push	eax	; lpBuffer
• 100101BB	mov	[ebp+StartupInfo.dwFlags], 101h	
• 100101C2	call	ds:GetSystemDirectoryA	
• 100101C8	cmp	dword_1008E5C4, ebx	
• 100101CE	jz	short loc_100101D7	
• 100101D0	push	offset aCmd_exeC ; "\\cmd.exe /c "	
• 100101D5	jmp	short loc_100101DC	
100101D7 ; -----			
100101D7			
100101D7 loc_100101D7:			; CODE XREF: sub_1000FF58+276↑j
• 100101D7	push	offset aCommand_exeC ; "\\command.exe /c "	
100101DC			
100101DC loc_100101DC:			; CODE XREF: sub_1000FF58+27D↑j
• 100101DC	lea	eax, [ebp+CommandLine]	
• 100101E2	push	eax	; char *
• 100101E3	call	strcat	
• 100101E8	pop	ecx	
• 100101E9	lea	eax, [ebp+var_5C0]	
• 100101EF	pop	ecx	
• 100101F0	push	0FFh	; size_t

Spacebar to get graph view



Drag graph view down to view subroutines > locate text "Hi, Master"

The screenshot shows the IDA Pro interface. The assembly window displays the following code:

```
push  eax
lea   eax, [ebp+var_E0]
push  offset aHiMasterDDDDD ; "Hi,Master [%d/%d/%d %d:%d:%d]\r\nWelCome ..."
push  eax
call  ds:sprintf
add   esp, 44h
xor   ebx, ebx
lea   eax, [ebp+var_E0]
push  ebx
push  eax      ; char *
call  strlen
pop   ecx
push  eax      ; int
lea   eax, [ebp+var_E0]
push  eax      ; int
push  [ebp+s]   ; s
call  sub_100038EE
add   esp, 10h
cmp   eax, 0FFFFFFFh
jz    loc_10010714
```

The graph overview window on the right shows a complex call graph with many nodes and edges.

Double-click aHiMasterDDDDD to view complete message

The screenshot shows the IDA Pro interface with the string dump window open. The dump shows the following data:

```
:10095B20 ; char aCommand_exeC[]
:10095B20 aCommand_exeC db '\command.exe /c ',0 ; DATA XREF: sub_1000FF58:loc_100101D7↑o
*:10095B31           align 4
*:10095B34 aCmd_exec db '\cmd.exe /c ',0      ; DATA XREF: sub_1000FF58+278↑o
*:10095B41           align 4
*:10095B44 ; char aHiMasterDDDDD[]
:10095B44 aHiMasterDDDDD db 'Hi,Master [%d/%d/%d %d:%d:%d]',0Dh,0Ah
:10095B44           ; DATA XREF: sub_1000FF58+145↑o
:10095B44           db 'WelCome Back...Are You Enjoying Today?',0Dh,0Ah
:10095B44           db 0Dh,0Ah
:10095B44           db 'Machine UpTime [-.2d Days %.2d Hours %.2d Minutes %.2d Seco
:10095B44           db 'ds]',0Dh,0Ah
:10095B44           db 'Machine IdleTime [-.2d Days %.2d Hours %.2d Minutes %.2d Sec
:10095B44           db 'nds]',0Dh,0Ah
:10095B44           db 0Dh,0Ah
:10095B44           db 'Encrypt Magic Number For This Remote Shell Session [0x%02x]',0Dh
:10095B44           db 0Dh,0Ah,0
:10095C5C ; char asc_10095C5C[]
:10095C5C asc_10095C5C:                      ; DATA XREF: sub_1000FF58+4B↑o
:10095C5C           ; sub_1000FF58+3E1↑o
```

Malware purpose: Encrypt Magic Number for This Remote Shell Session [0x%02x]

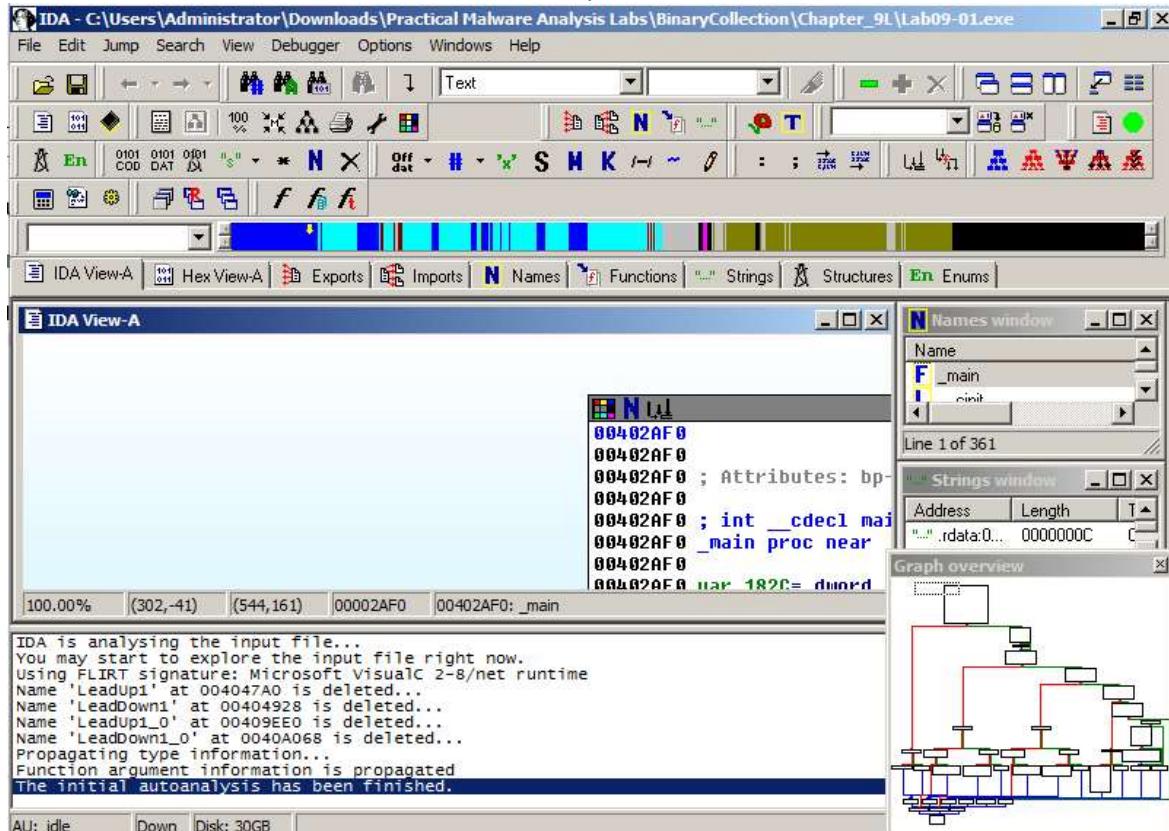
Chapter 9: OllyDBG - Lab9-1

File: Lab09-01.exe

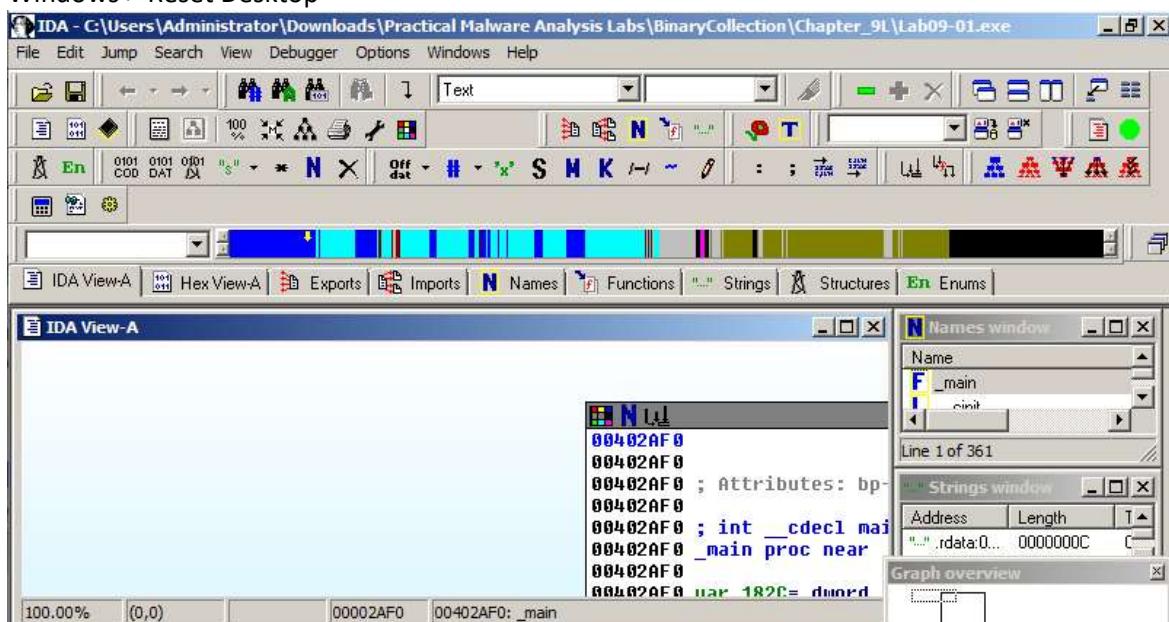
OllyDbg 1.10⁷² Analysis

Find Main Entry Point

IDA Pro Free > PE Executable > Lab09-01.exe > Options > General > check Line Prefixes > OK

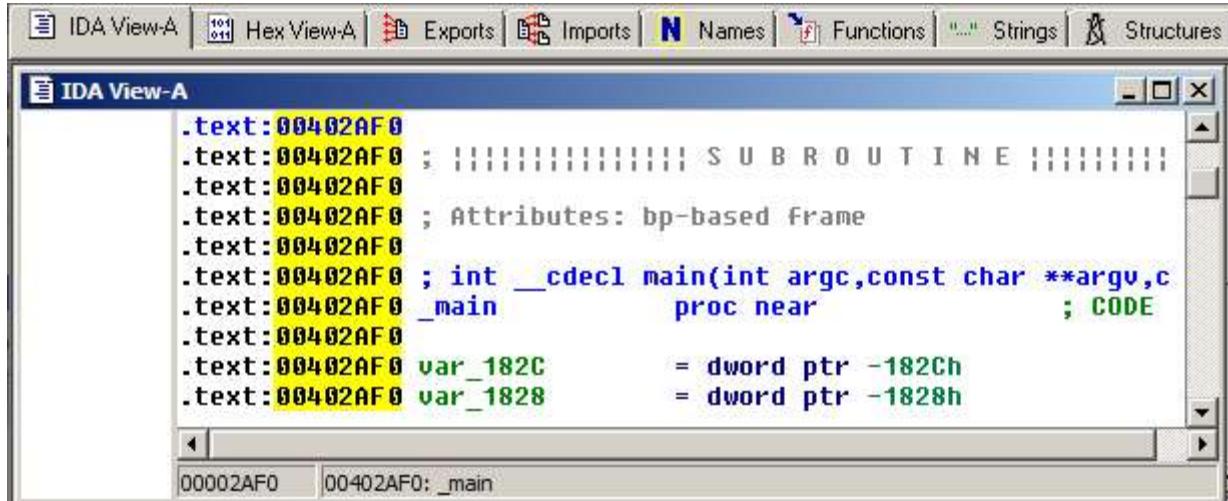


Windows > Reset Desktop



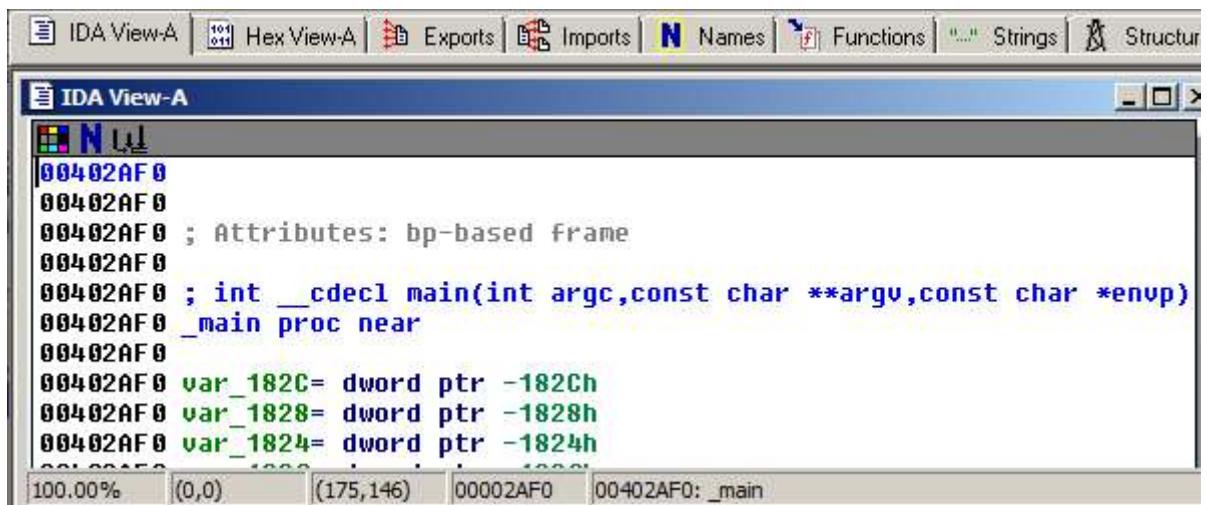
⁷² OllyDBG (2019) Download OllyDbg 1.10 (final version) <http://www.ollydbg.de/download.htm> [Accessed 22nd May 2019].

Main starts at 0x402AF0



IDA View-A window showing assembly code for the main function. The code is annotated with comments and attributes:

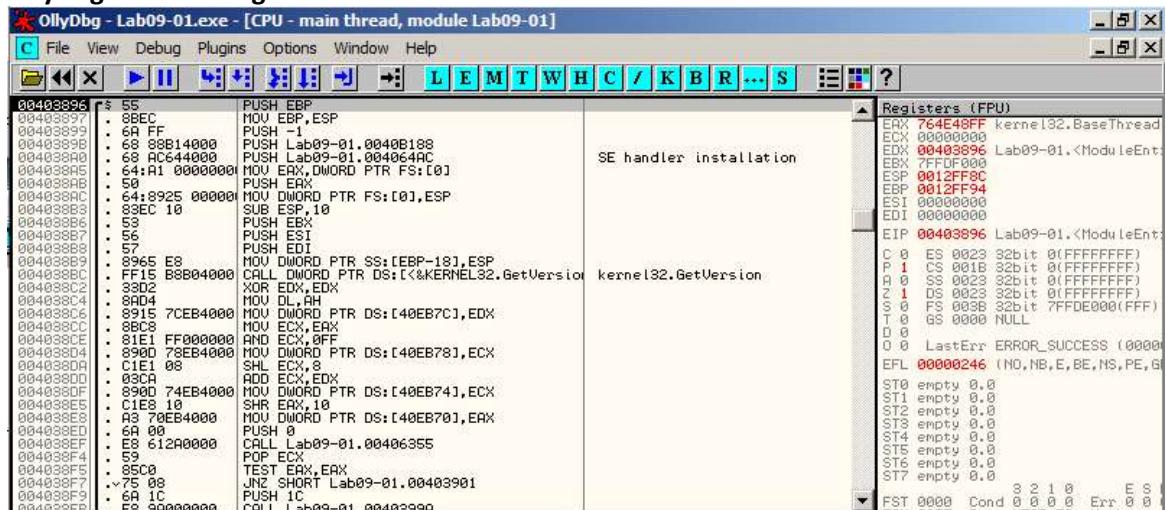
```
.text:00402AF0 ; SUBROUTINE
.text:00402AF0 ; Attributes: bp-based Frame
.text:00402AF0 ; int __cdecl main(int argc,const char **argv,const char *envp)
.text:00402AF0 _main proc near ; CODE
.text:00402AF0 var_182C = dword ptr -182Ch
.text:00402AF0 var_1828 = dword ptr -1828h
```



IDA View-A window showing assembly code for the main function with names:

```
00402AF0
00402AF0 ; Attributes: bp-based Frame
00402AF0 ; int __cdecl main(int argc,const char **argv,const char *envp)
00402AF0 _main proc near
00402AF0 var_182C= dword ptr -182Ch
00402AF0 var_1828= dword ptr -1828h
00402AF0 var_1824= dword ptr -1824h
```

OllyDbg Walk Through



Press F8 > highlight address 0x403933 > scroll to display Arg3, Arg2, Arg1

The screenshot shows the OllyDbg debugger interface. The assembly window displays the following code:

```

00403901 > 8365 FC 00 AND DWORD PTR SS:[EBP-4],0
00403909 . E8 47170000 CALL Lab09-01.00403905
00403909P . FF15 B4B40000 CALL DWORD PTR DS:[<&a href="#">GetCommandLineA]
00403910A A3 A4014100 MOV DWORD PTR DS:[4101A4],EAX
00403915 E8 94270000 CALL Lab09-01.004060AE
00403918 A3 D4E84000 MOV DWORD PTR DS:[40EBD0],EAX
0040391F E8 3D250000 CALL Lab09-01.00405E61
00403924 E8 7F240000 CALL Lab09-01.00405DAB
00403929 E8 4EF4FFF CALL Lab09-01.00402D7C
0040392E A1 8CE84000 MOV ECX,DWORD PTR DS:[40EBB8C]
00403932E A3 90E84000 MOV DWORD PTR DS:[40EB90],ERX

```

The Registers window shows:

- EIP: 00403933 Lab09-01.00403933
- Arg3 => 01AB0B18
- Arg2 = 01AB0A80
- Arg1 = 00000001
- Lab09-01.00402AF0

The status bar at the bottom right shows: FST 0000 Cond 3 2 1 0 Err E S

Press F7 > highlight 0x402AF0

The screenshot shows the OllyDbg debugger interface. The assembly window displays the following code:

```

00402AF0 $ E5 PUSH EBP
00402AF1 . 8BEC MOU EBP,ESP
00402AF2 . B8 21C10000 MOU EAX,182C
00402AF5 . E8 B3030000 CALL Lab09-01.00402EB0
00402AF8 . 837D 08 01 CMP DWORD PTR SS:[EBP+8],1
00402B01 .> 75 1A JNZ SHORT Lab09-01.00402B10
00402B03 . E8 F8E4FFFF CALL Lab09-01.00401000
00402B05 . 85C0 TEST EAX,EAX
00402B07 .> 74 07 JE SHORT Lab09-01.00402B13
00402B09 . E8 4FF8FFFF CALL Lab09-01.00402360
00402B11 .> EB 05 JMP SHORT Lab09-01.00402B18
00402B13 .> E8 F8F8FFFF CALL Lab09-01.00402410
00402B18 .> E9 59020000 JMP Lab09-01.00402D76
00402B1D .> 8B45 08 MOV EAX,DWORD PTR SS:[EBP+8]
00402B20 .> 8B40 0C MOV ECX,DWORD PTR SS:[EBP+C]
00402B23 .> BB5481 FC MOV EDX,DWORD PTR DS:[ECX+EXX*4-4]
00402B27 .> 8955 FC MOV DWOR PTR SS:[EBP-4],EDX
00402B29 .> 8B45 FC MOV EAX,DWORD PTR SS:[EBP-4]
00402B2D .> 50 PUSH EAX
00402B2E .> E8 DDF9FFFF CALL Lab09-01.00402510
00402B30 .> 83C4 04 ADD ESP,4
00402B32 .> 85C0 TEST EAX,EAX
00402B33 .> 75 05 JNZ SHORT Lab09-01.00402B3F
00402B34 .> E8 D1F8FFFF CALL Lab09-01.00402410
00402B37 .> 8B4D 0C MOV ECX,DWORD PTR SS:[EBP+C]
00402B42 .> BB51 04 MOV EDX,DWORD PTR DS:[ECX+4]
00402B45 .> 8995 E0E7FFFF MOV DWOR PTR SS:[EBP-1820],EDX
00402B49 .> 68 70C14000 PUSH Lab09-01.0040C170
00402B50 .> 8B85 E0E7FFFF MOV EAX,DWORD PTR SS:[EBP-1820]
00402B54 .> 50 PUSH EAX
00402B55 .> E8 B30C0000 CALL Lab09-01.0040380F
00402B57 .> 83C4 08 ADD ESP,8
00402B5F .> 85C0 TEST EAX,EAX
00402B61 .> 75 64 JNZ SHORT Lab09-01.00402BC7
00402B63 .> 837D 08 03 CMP DWORD PTR SS:[EBP+8],3
00402B67 .> 75 31 JNZ SHORT Lab09-01.00402B90

```

The Registers window shows:

- EIP: 00402AF0 Lab09-01.00402AF0
- Arg1 => Lab09-01.00402510
- ASCII "in"

The status bar at the bottom right shows: FST 0000 Cond 3 2 1 0 Err E S

Press F7 > highlight 0x402AFD

The screenshot shows the OllyDbg debugger interface. The assembly window displays the following code:

```

00402AFD . 837D 08 01 CMP DWORD PTR SS:[EBP+8],1
00402B01 .> 75 1A JNZ SHORT Lab09-01.00402B10
00402B03 . E8 F8E4FFFF CALL Lab09-01.00401000
00402B05 . 85C0 TEST EAX,EAX
00402B07 .> 74 07 JE SHORT Lab09-01.00402B13
00402B09 . E8 4FF8FFFF CALL Lab09-01.00402360
00402B11 .> EB 05 JMP SHORT Lab09-01.00402B18
00402B13 .> E8 F8F8FFFF CALL Lab09-01.00402410
00402B18 .> E9 59020000 JNP Lab09-01.00402D76
00402B1D .> 8B45 08 MOV EAX,DWORD PTR SS:[EBP+8]
00402B20 .> 8B40 0C MOV ECX,DWORD PTR SS:[EBP+C]
00402B23 .> BB5481 FC MOV EDX,DWORD PTR DS:[ECX+EXX*4-4]
00402B27 .> 8955 FC MOV DWOR PTR SS:[EBP-4],EDX
00402B29 .> 8B45 FC MOV EAX,DWORD PTR SS:[EBP-4]
00402B2D .> 50 PUSH EAX
00402B2E .> E8 DDF9FFFF CALL Lab09-01.00402510
00402B30 .> 83C4 04 ADD ESP,4
00402B32 .> 85C0 TEST EAX,EAX
00402B33 .> 75 05 JNZ SHORT Lab09-01.00402B3F
00402B34 .> E8 D1F8FFFF CALL Lab09-01.00402410
00402B37 .> 8B4D 0C MOV ECX,DWORD PTR SS:[EBP+C]
00402B42 .> BB51 04 MOV EDX,DWORD PTR DS:[ECX+4]
00402B45 .> 8995 E0E7FFFF MOV DWOR PTR SS:[EBP-1820],EDX
00402B49 .> 68 70C14000 PUSH Lab09-01.0040C170
00402B50 .> 8B85 E0E7FFFF MOV EAX,DWORD PTR SS:[EBP-1820]
00402B54 .> 50 PUSH EAX
00402B55 .> E8 B30C0000 CALL Lab09-01.0040380F
00402B57 .> 83C4 08 ADD ESP,8
00402B5F .> 85C0 TEST EAX,EAX
00402B61 .> 75 64 JNZ SHORT Lab09-01.00402BC7
00402B63 .> 837D 08 03 CMP DWORD PTR SS:[EBP+8],3
00402B67 .> 75 31 JNZ SHORT Lab09-01.00402B90

```

The Registers window shows:

- EIP: 00402AFD Lab09-01.00402AFD
- Arg1 => Lab09-01.00402510
- ASCII "in"

The status bar at the bottom right shows: FST 0000 Cond 3 2 1 0 Err E S

CMP operation testing to see number of command-line arguments is 1

```

00402AF0    . 837D 08 01    CMP DWORD PTR SS:[EBP+8],1
00402B01    .> 75 1A        JNZ SHORT Lab09-01.00402B10
00402B03    . E8 F8E4FFFF    CALL Lab09-01.00401000
00402B08    . 85C0          TEST EAX,EAX
00402B09    .> 74 07        JE SHORT Lab09-01.00402B13

```

F7 x3 to pass test and go to 0x00401000

Routine 0x401000 calls RegOpenKeyExA at 0x40101B

```

00401000    $ 55            PUSH EBX
00401001    . 8BEC          MOV EBX,ESP
00401003    . 83EC 08        SUB ESP,8
00401006    . 8D45 F8        LEA EAX,DWORD PTR SS:[EBP-8]
00401009    . 50            PUSH EAX
0040100A    . 68 3F0000F000  PUSH 0F003F
0040100F    . 6A 00          PUSH 0
00401011    . 68 40C04000    PUSH Lab09-01.0040C040
00401016    . 68 02000002  PUSH 00000002
0040101B    . FF15 20B040000 CALL DWORD PTR DS:[&&ADVAPI32.RegOpenKeyExA]
00401021    . 85C0          TEST EAX,EAX
00401023    .> 74 04        JE SHORT Lab09-01.00401029
00401025    . 33C0          XOR EAX,EAX
00401027    .> EB 3D        JMP SHORT Lab09-01.00401066
00401029    .> 6A 00        PUSH 0
0040102B    . 6A 00        PUSH 0
0040102D    . 6A 00        PUSH 0
00401031    .> 6A 00        PUSH 0
00401033    . 68 30C040000 PUSH Lab09-01.0040C030
00401036    . 8B4D F8        MOV ECX,DWORD PTR SS:[EBP-8]
00401039    . 51            PUSH ECX
0040103B    . FF15 24B040000 CALL DWORD PTR DS:[&&ADVAPI32.RegQueryValueExA]
00401040    . 8945 FC        MOV DWORD PTR SS:[EBP-41],EAX
00401043    . 837D FC 00        CMP DWORD PTR SS:[EBP-41],0
00401047    .> 74 0E        JE SHORT Lab09-01.00401057
00401049    . 8B55 F8        MOV EDX,DWORD PTR SS:[EBP-8]
0040104C    . 52            PUSH EDX
0040104D    . FF15 64B040000 CALL DWORD PTR DS:[&&KERNEL32.CloseHandle]
00401053    . 33C0          XOR EHX,EHX
00401055    .> EB 0F        JMP SHORT Lab09-01.00401066
00401057    .> 8B45 F8        MOV EAX,DWORD PTR SS:[EBP-8]
0040105D    . 50            PUSH EBX

```

Registers (FPU)

ERX	00402AF0	Lab09-01.00402AF0
ECX	0040C110	Lab09-01.0040C210
EDX	00000000	
EBX	7FFDD0000	
ESP	0012E718	
EIP	00401000	Lab09-01.00401000
C 0	ES 0023	32bit 0(FFFFFFFF)
P 1	CS 001B	32bit 0(FFFFFF)
A 0	SS 0023	32bit 0(FFFFFF)
Z 1	DS 0023	32bit 0(FFFFFF)
S 0	FS 003B	32bit 7FFDF000(FFF)
T 0	GS 0000	NULL
D 0		LastErr ERROR_SUCCESS (0000)
EFL	00000246	(NO,NB,E,BE,NS,PE,GI)
ST0	empty	0.0
ST1	empty	0.0
ST2	empty	0.0
ST3	empty	0.0
ST4	empty	0.0
ST5	empty	0.0
ST6	empty	0.0
ST7	empty	0.0
FST	0000	Cond 0 0 0 0 Err 0 0
CPU	00000000	MIPS

Click line starting 0x401021 > press F2 to insert breakpoint (address turns red)

Click line starting 0x401000 > press F9 to run breakpoint

```

00401000    $ 55            PUSH EBX
00401001    . 8BEC          MOV EBX,ESP
00401003    . 83EC 08        SUB ESP,8
00401006    . 8D45 F8        LEA EAX,DWORD PTR SS:[EBP-8]
00401009    . 50            PUSH EAX
0040100A    . 68 3F0000F000  PUSH 0F003F
0040100F    . 6A 00          PUSH 0
00401011    . 68 40C04000    PUSH Lab09-01.0040C040
00401016    . 68 02000002  PUSH 00000002
0040101B    . FF15 20B040000 CALL DWORD PTR DS:[&&ADVAPI32.RegOpenKeyExA]
00401021    .> 74 04        JE SHORT Lab09-01.00401029
00401023    . 33C0          XOR EAX,EAX
00401025    .> EB 3D        JMP SHORT Lab09-01.00401066
00401027    .> 6A 00        PUSH 0
00401029    .> 6A 00        PUSH 0
0040102B    . 6A 00        PUSH 0
0040102D    . 6A 00        PUSH 0
00401031    .> 6A 00        PUSH 0
00401033    . 68 30C040000 PUSH Lab09-01.0040C030
00401036    . 8B4D F8        MOV ECX,DWORD PTR SS:[EBP-8]
00401039    . 51            PUSH ECX
0040103B    . FF15 24B040000 CALL DWORD PTR DS:[&&ADVAPI32.RegQueryValueExA]
00401040    . 8945 FC        MOV DWORD PTR SS:[EBP-41],EAX
00401043    . 837D FC 00        CMP DWORD PTR SS:[EBP-41],0
00401047    .> 74 0E        JE SHORT Lab09-01.00401057
00401049    . 8B55 F8        MOV EDX,DWORD PTR SS:[EBP-8]
0040104C    . 52            PUSH EDX
0040104D    . FF15 64B040000 CALL DWORD PTR DS:[&&KERNEL32.CloseHandle]
00401053    . 33C0          XOR EHX,EHX
00401055    .> EB 0F        JMP SHORT Lab09-01.00401066
00401057    .> 8B45 F8        MOV EAX,DWORD PTR SS:[EBP-8]
0040105D    . 50            PUSH EBX

```

Registers (FPU)

ERX	00000002	
ECX	77000611	ADVAPI32.77000611
EDX	0010B16C	
EBX	7FFDD0000	
ESP	0012E70C	
EIP	0012E714	
EST	00000000	
EDI	00000000	
EIP	00401021	Lab09-01.00401021
C 0	ES 0023	32bit 0(FFFFFFFF)
P 1	CS 001B	32bit 0(FFFFFF)
A 0	SS 0023	32bit 0(FFFFFF)
Z 1	DS 0023	32bit 0(FFFFFF)
S 0	FS 003B	32bit 7FFDF000(FFF)
T 0	GS 0000	NULL
D 0	0 0	LastErr ERROR_SUCCESS (0000)
EFL	00000246	(NO,NB,E,BE,NS,PE,GI)
ST0	empty	0.0
ST1	empty	0.0
ST2	empty	0.0
ST3	empty	0.0
ST4	empty	0.0
ST5	empty	0.0
ST6	empty	0.0
ST7	empty	0.0
FST	0000	Cond 0 0 0 0 Err 0 0
CPU	00000000	MIPS

Registers (FPU) EAX 0000002

```
Registers (FPU)
EAX 00000002
ECX 77000611 ADVAPI32.77000611
EDX 0010016C
EBX 7FFDD000
ESP 0012E70C
EBP 0012E714
ESI 00000000
EDI 00000000
EIP 00401021 Lab09-01.00401021
C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 001B 32bit 0(FFFFFFFF)
A 0 SS 0023 32bit 0(FFFFFFFF)
Z 1 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 003B 32bit 7FFDF000(FFF)
T 0 GS 0000 NULL
D 0
O 0 LastErr ERROR_SUCCESS (0000)
EFL 00000246 (NO,NB,E,BE,NS,PE,G)
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
          3 2 1 0      E S
FST 0000 Cond 0 0 0 0 Err 0 0
FCW 027F Prec NEAR,53 Mask
```

Non-zero error code which means the registry key was not found

RegOpenKeyExA function⁷³

Opens the specified registry key. Note that key names are not case sensitive.

To perform transacted registry operations on a key, call the RegOpenKeyTransacted function.

C++ Syntax

```
LSTATUS RegOpenKeyExA(
    HKEY hKey,
    LPCSTR lpSubKey,
    DWORD ulOptions,
    REGSAM samDesired,
    PHKEY phkResult
);
```

hKey: A handle to an open registry key. Predefined keys:

```
HKEY_CLASSES_ROOT HKEY_CURRENT_CONFIG HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE HKEY_USERS
```

lpSubKey: The name of the registry subkey to be opened.

ulOptions: Specifies the option to apply when opening the key

samDesired: A mask that specifies the desired access rights to the key to be opened

phkResult: A pointer to a variable that receives a handle to the opened key

⁷³ Microsoft.com (2019) *RegOpenKeyExA function* [http://msdn.microsoft.com/en-us/library/windows/desktop/ms724897\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms724897(v=vs.85).aspx) [Accessed 22nd May 2019].

F7 x3 to go to 0x401027

The screenshot shows the OllyDbg debugger interface. The assembly window displays the following code:

```

00401000  $ 55      PUSH EBP
00401001  . 8BEC    MOV EBP,ESP
00401003  . 83C4 08  SUB ESP,8
00401006  . 8D45 F8  LEA EAX, DWORD PTR SS:[EBP-8]
00401009  . 58      PUSH EAX
0040100A  . 68 3F000F00 PUSH 0F000F
0040100B  . 6A 00  PUSH 0
00401011  . 68 40C04000 PUSH Lab09-01.0040C040
00401018  . 68 02000000 PUSH 00000000
00401019  . FF15 2B004000 CALL DWORD PTR DS:[&ADVAPI32.RegOpenKeyExA]
00401021  . 85C0  TEST EAX,EAX
00401022  . 74 04  JE SHORT Lab09-01.00401029
00401023  . 33C0  XOR EAX,EAX
00401027  . EB 3D  JMP SHORT Lab09-01.00401066
00401029  > 6A 00  PUSH 0
0040102B  . 6A 00  PUSH 0
0040102D  . 6A 00  PUSH 0
0040102F  . 6A 00  PUSH 0
00401031  . 68 30C04000 MOV ECX,DWORD PTR SS:[EBP-8]
00401036  . 8B4D F8  PUSH ECX
00401039  . FF15 24B04000 CALL DWORD PTR DS:[&ADVAPI32.RegQueryValueExA]
00401040  . 8945 FC  MOV DWORD PTR SS:[EBP-4],EAX
00401043  . 837D FC 00 CMP DWORD PTR SS:[EBP-4],0
00401047  . 74 0E  JE SHORT Lab09-01.00401057
00401049  . 8B55 F8  MOV EDX,DWORD PTR SS:[EBP-8]
0040104C  . 52      PUSH EDX
00401051  . FF15 64B04000 CALL DWORD PTR DS:[&KERNEL32.CloseHandle]
00401053  . 33C0  XOR EAX,EAX
00401055  . EB 0F  JMP SHORT Lab09-01.00401066
00401059  > 8B45 F8  MOV EAX,DWORD PTR SS:[EBP-8]
0040105A  . 50      PUSH EAX

```

The registers window shows the following state:

Register	Value
ECX	00000000
EDX	0010016C
EBX	7FFDD0000
ESP	0012E70C
EBP	0012E714
ESI	00000000
EDI	00000000
EIP	00401027 Lab09-01.00401027
C	0 ES 0023 32bit 0(FFFFFF)
P	1 CS 001B 32bit 0(FFFFFF)
A	0 SS 0023 32bit 0(FFFFFF)
Z	1 DS 0023 32bit 0(FFFFFF)
S	0 FS 003B 32bit 7FFDF000(F)
T	0 GS 0000 NULL
D	0 0 LastErr ERROR_SUCCESS (0000)
EFL	00000246 (NO,NB,E,BE,NS,PE,G)
ST0	empty 0.0
ST1	empty 0.0
ST2	empty 0.0
ST3	empty 0.0
ST4	empty 0.0
ST5	empty 0.0
ST6	empty 0.0
ST7	empty 0.0
FST	0000 Cond 0 0 0 0 Err 0 0
FNM	007E Pmode NFOR FZ Mask

F7 to execute JMP

The screenshot shows the OllyDbg debugger interface. The assembly window displays the following code:

```

00401066  > B8E5  MOV ESP,EBP
00401067  . 5D  POP EBP
00401068  . C3  RETN
00401069  . CC  INT3
0040106A  . CC  INT3
0040106B  . CC  INT3
0040106C  . CC  INT3
0040106D  . CC  INT3
0040106E  . CC  INT3
0040106F  . CC  INT3
00401070  $ 55      PUSH EBP
00401071  . 8BEC  MOV EBP,ESP
00401073  . B8 0C100000 MOV EAX,100C
00401078  . E8 331E0000 CALL Lab09-01.00402EB0
0040107D  . 56      PUSH ESI
0040107E  . S7      PUSH EDI
0040107F  . B9 00040000 MOV ECX,400
00401084  . 33C0  XOR EAX,EAX
00401086  . 8DDB F8FFFFF LEA EDI,DWORD PTR SS:[EBP-1008]
0040108C  . REP STOS DWORD PTR ES:[EDI]
0040108E  . AA      STOS BTNE PTR ES:[EDI]
0040108F  . 8D85 F8EFFFFF LEA EAX,DWORD PTR SS:[EBP-1008]
00401095  . 8945 FC  MOV DWORD PTR SS:[EBP-4],EAX
00401098  . 8B7D 08  MOV EDI,DWORD PTR SS:[EBP+8]
0040109B  . 8655 FC  MOV EDX,DWORD PTR SS:[EBP-4]
0040109E  . 83C9 FF  OR ECX,FFFFFF
004010A1  . 33C0  XOR EAX,EAX
004010A3  . F2:AE  REPNE SCAS BYTE PTR ES:[EDI]
004010A5  . F7D1  NOT ECX
004010A7  . 2BF9  SUB EDI,ECX
004010A9  . 8BF7  MOV ESI,EDI
004010AB  . 8BC1  MOV EAX,ECX
004010AD  . 8BFA  MOV EDI,EDX

```

The registers window shows the following state:

Register	Value
ECX	00000000
EDX	0010016C
EBX	7FFDD0000
ESP	0012E70C
EBP	0012E714
ESI	00000000
EDI	00000000
EIP	00401066 Lab09-01.00401066
C	0 ES 0023 32bit 0(FFFFFF)
P	1 CS 001B 32bit 0(FFFFFF)
A	0 SS 0023 32bit 0(FFFFFF)
Z	1 DS 0023 32bit 0(FFFFFF)
S	0 FS 003B 32bit 7FFDF000(F)
T	0 GS 0000 NULL
D	0 0 LastErr ERROR_SUCCESS (0000)
EFL	00000246 (NO,NB,E,BE,NS,PE,G)
ST0	empty 0.0
ST1	empty 0.0
ST2	empty 0.0
ST3	empty 0.0
ST4	empty 0.0
ST5	empty 0.0
ST6	empty 0.0
ST7	empty 0.0
FST	0000 Cond 3 2 1 0 Err 0 0
FNM	007E Pmode NFOR FZ Mask

F7 x3 to step through subroutine and get to 0x402B08

The screenshot shows the OllyDbg debugger interface. The assembly window displays the following code:

```

00402B08  . 85C0  TEST EAX,EAX
00402B09  . 74 07  JE SHORT Lab09-01.00402B13
00402B0C  . E8 4FF8FFFF CALL Lab09-01.00402B36
00402B11  . E8 05  JMP SHORT Lab09-01.00402B18
00402B13  . E8 F8FFFF JMP Lab09-01.00402B10
00402B18  . E8 59E20000 CALL Lab09-01.00402B76
00402B1D  . 8845 08  MOU ECX,DWORD PTR SS:[EBP+8]
00402B20  . 884D 0C  MOU ECX,DWORD PTR SS:[EBP+C]
00402B23  . 865481 FC MOU EDX,DWORD PTR DS:[ECX+EAX*4-4]
00402B27  . 8955 FC  MOU DWORD PTR SS:[EBP-4],EDX
00402B2D  . 8845 FC  MOU ECX,DWORD PTR SS:[EBP-4]
00402B30  . 50      PUSH EAX
00402B32  . E8 DDF9FFFF CALL Lab09-01.00402510
00402B33  . ADD ESP,4
00402B34  . 85C0  TEST EAX,EAX
00402B38  . 75 05  JNZ SHORT Lab09-01.00402B3F
00402B3B  . E8 D1F8FFFF CALL Lab09-01.00402410
00402B3C  . 884D 0C  MOU ECX,DWORD PTR SS:[EBP+C]
00402B42  . 8851 04  MOU EDX,DWORD PTR DS:[ECX+4]
00402B45  . 8995 E0E7FFFF MOU DWORD PTR SS:[EBP-1820],EDX
00402B48  . 68 70C14000 PUSH Lab09-01.0040C170
00402B50  . 8885 E0E7FFFF MOU EAX,DWORD PTR SS:[EBP-1820]
00402B56  . 8845 08  ADD ESP,8
00402B57  . 83C4 08  TEST EAX,EAX
00402B5C  . 75 64  JNZ SHORT Lab09-01.00402B77
00402B63  . 837D 08 03  CMP DWORD PTR SS:[EBP+8],3
00402B67  . 75 31  JNZ SHORT Lab09-01.00402B9A
00402B69  . 68 00040000 PUSH 400
00402B6E  . 80D0 FCFBFFFF LEA ECX,DWORD PTR SS:[EBP-404]
00402B74  . 51      PUSH ECX

```

The registers window shows the following state:

Register	Value
ECX	00000000
EDX	77DD00611 ADVAPI32.77DD00611
EBX	0010016C
ESP	7FFDD0000
EBP	0012E71C
ESI	00000000
EDI	00000000
EIP	00402B08 Lab09-01.00402B08
C	0 ES 0023 32bit 0(FFFFFF)
P	1 CS 001B 32bit 0(FFFFFF)
A	0 SS 0023 32bit 0(FFFFFF)
Z	1 DS 0023 32bit 0(FFFFFF)
S	0 FS 003B 32bit 7FFDF000(F)
T	0 GS 0000 NULL
D	0 0 LastErr ERROR_SUCCESS (0000)
EFL	00000246 (NO,NB,E,BE,NS,PE,G)
ST0	empty 0.0
ST1	empty 0.0
ST2	empty 0.0
ST3	empty 0.0
ST4	empty 0.0
ST5	empty 0.0
ST6	empty 0.0
ST7	empty 0.0
FST	0000 Cond 3 2 1 0 Err 0 0
FNM	007E Pmode NFOR FZ Mask

F7 x3 to go to 0x402410

```

00402410 $ 55 PUSH EBP
00402411 . 8BEC MOV EBP,ESP
00402413 . 81EC 08020000 SUB ESP,208
00402419 . 53 PUSH EBX
0040241A . 56 PUSH ESI
0040241B . 57 PUSH EDI
0040241C . 68 04010000 PUSH 104
00402421 . 8D85 F8FDFFFF LEA EAX,DWORD PTR SS:[EBP-208]
00402427 . 6A 00 PUSH 0
0040242A . FF15 38B04000 CALL DWORD PTR DS:[&KERNEL32.GetModule
00402430 . 68 04010000 PUSH 104
00402435 . 8D80 F8FDFFFF LEA ECX,DWORD PTR SS:[EBP-208]
0040243B . 51 PUSH ECX
0040243C . 8D95 F8FDFFFF LEA EDX,DWORD PTR SS:[EBP-208]
00402442 . 52 PUSH EDX
00402443 . FF15 3CB04000 CALL DWORD PTR DS:[&KERNEL32.GetShortP
00402449 . BF DCC04000 MOV EDI,Lab09-01.0040C0DC
00402454 . 83C9 FF OR ECX,FFFFFFF
00402457 . 33C0 XOR EAX,EAX
0040245B . F2:AЕ REPNE SCAS BYTE PTR ES:[EDI]
0040245D . F7D1 NOT ECX
0040245F . 2BF9 SUB EDI,ECX
00402461 . 8BF7 MOV ESI,EDI
00402463 . 8BC1 MOV EAX,ECX
00402465 . 8BF8 MOV EDI,EDX
00402468 . C1E9 02 SHR ECX,2
0040246B . F3:A5 REP MOVS DWORD PTR ES:[EDI],DWORD PTR D
0040246C . 8BC8 MOV ECX,EAX
0040246E . 83E1 03 AND ECX,3
0040246F . F3:A4 REP MOVS BVTF PTR FS:[EDI1].BVTF PTR DS:

```

Registers (FPU)

- ECX 770DD611 ADUAPI32.770DD611
- EDX 001D016C
- EBX 7FFD0000
- ESP 0012FF48
- ESI 00000000
- EDI 00000000
- EIP 00402410 Lab09-01.00402410
- C 0 ES 0023 32bit 0(FFFFFFFF)
- P 1 CS 001B 32bit 0(FFFFFFFF)
- A 0 SS 0023 32bit 0(FFFFFFFF)
- Z 1 DS 0023 32bit 0(FFFFFFFF)
- S 0 FS 003B 32bit 7FFDF000(FFF)
- T 0 GS 0000 NULL
- D 0
- O 0 LastErr ERROR_SUCCESS (0000)
- EFL 00000246 (NO,NB,E,BE,NS,PE,G)
- ST0 empty 0.0
- ST1 empty 0.0
- ST2 empty 0.0
- ST3 empty 0.0
- ST4 empty 0.0
- ST5 empty 0.0
- ST6 empty 0.0
- ST7 empty 0.0

FST 0000 Cond 0 0 0 0 Err E S

EFN 027E Pmn NFOR F2 Mack

GetModuleFilename function to get path to current exe and builds ASCII string

/c del path-to-executable >> NUL

```

00402410 $ 55 PUSH EBP
00402411 . 8BEC MOV EBP,ESP
00402413 . 81EC 08020000 SUB ESP,208
00402419 . 53 PUSH EBX
0040241A . 56 PUSH ESI
0040241B . 57 PUSH EDI
0040241C . 68 04010000 PUSH 104
00402421 . 8D85 F8FDFFFF LEA EAX,DWORD PTR SS:[EBP-208]
00402427 . 50 PUSH EAX
0040242A . 6A 00 PUSH 0
0040242D . FF15 38B04000 CALL DWORD PTR DS:[&KERNEL32.GetModu
00402430 . 68 04010000 PUSH 104
00402435 . 8D80 F8FDFFFF LEA ECX,DWORD PTR SS:[EBP-208]
0040243B . 51 PUSH ECX
0040243C . 8D95 F8FDFFFF LEA EDX,DWORD PTR SS:[EBP-208]
00402442 . 52 PUSH EDX
00402443 . FF15 3CB04000 CALL DWORD PTR DS:[&KERNEL32.GetShortP
00402449 . BF DCC04000 MOV EDI,Lab09-01.0040C0DC
00402454 . 83C9 FF OR ECX,FFFFFFF
00402457 . 33C0 XOR EAX,EAX
0040245B . F2:AЕ REPNE SCAS BYTE PTR ES:[EDI]
0040245D . F7D1 NOT ECX
0040245F . 2BF9 SUB EDI,ECX
00402461 . 8BF7 MOV ESI,EDI
00402463 . 8BC1 MOV EAX,ECX
00402465 . 8BF8 MOV EDI,EDX
00402468 . C1E9 02 SHR ECX,2
0040246B . F3:A5 REP MOVS DWORD PTR ES:[EDI],DWORD PTR D
0040246C . 8BC8 MOV ECX,EAX
0040246E . 83E1 03 AND ECX,3
0040246F . F3:A4 REP MOVS BVTF PTR FS:[EDI1].BVTF PTR DS:

```

Registers (FPU)

- ECX 770DD611 ADUAPI32.770DD611
- EDX 001D016C
- EBX 7FFD0000
- ESP 0012FF48
- ESI 00000000
- EDI 00000000
- EIP 00402410 Lab09-01.00402410
- C 0 ES 0023 32bit 0(FFFFFFFF)
- P 1 CS 001B 32bit 0(FFFFFFFF)
- A 0 SS 0023 32bit 0(FFFFFFFF)
- Z 1 DS 0023 32bit 0(FFFFFFFF)
- S 0 FS 003B 32bit 7FFDF000(FFF)
- T 0 GS 0000 NULL
- D 0
- O 0 LastErr ERROR_SUCCESS (0000)
- EFL 00000246 (NO,NB,E,BE,NS,PE,G)
- ST0 empty 0.0
- ST1 empty 0.0
- ST2 empty 0.0
- ST3 empty 0.0
- ST4 empty 0.0
- ST5 empty 0.0
- ST6 empty 0.0
- ST7 empty 0.0

FST 0000 Cond 0 0 0 0 Err E S

EFN 027E Pmn NFOR F2 Mack

EBP=0012FF48
Local calls from 00402B13, 00402B3A, 00402BBD, 00402C45, 00402CCF, 00402D6A, 00402D71

Click 00402449 > F2 to insert breakpoint

OllyDbg - Lab09-01.exe - [CPU - main thread, module Lab09-01]

C File View Debug Plugins Options Window Help

LEMTW H C / K B R ... S

```
00402410 $ 55 PUSH EBP
00402411 . 8BEC MOV EBP,ESP
00402413 . 81EC 08020000 SUB ESP,208
00402419 . 53 PUSH EBX
0040241A . 56 PUSH ESI
0040241B . 57 PUSH EDI
0040241C . 68 04010000 PUSH 104
00402421 . 8D85 F8FDFFFF LEA EAX,DWORD PTR SS:[EBP-208]
00402427 . 50 PUSH EAX
00402428 . 6A 00 PUSH 0
0040242A . FF15 38B04000 CALL DWORD PTR DS:[&KERNEL32.GetModule
00402430 . 68 04010000 PUSH 104
00402435 . 8D8D F8FDFFFF LEA ECX,DWORD PTR SS:[EBP-208]
0040243B . 51 PUSH ECX
0040243C . 8D95 F8FDFFFF LEA EDX,DWORD PTR SS:[EBP-208]
00402442 . 52 PUSH EDX
00402443 . FF15 3CB04000 CALL DWORD PTR DS:[&KERNEL32.GetShortP
0040244E . BF DCC04000 MOV EDI,Lab09-01.0040C0DC
00402454 . 8D95 FCFFEFFF LEA EDX,DWORD PTR SS:[EBP-104]
00402457 . 83C9 FF OR ECX,FFFFFFF
00402459 . 33C0 XOR EAX,EAX
0040245A . F2:AE REPNE SCAS BYTE PTR ES:[EDI]
0040245B . F701 NOT ECX
0040245D . 2BF9 SUB EDI,ECX
0040245F . 8BF7 MOV ESI,EDI
00402461 . 8BC1 MOV EAX,ECX
00402463 . 8BFA MOV EDI,EDX
00402465 . C1E9 02 SHR ECX,2
00402468 . F3:A5 REP MOVS DWORD PTR ES:[EDI],DWORD PTR DS:
0040246A . 8BC8 MOV ECX,EAX
0040246C . 83E1 03 AND ECX,3
0040246F . F3:A4 REP MOVS BYTF PTR FS:[EDI1].BYTF PTR DS:
0040C0DC=Lab09-01.0040C0DC (ASCII "/c del ")
```

Click line 0x402410 > F9 to run breakpoint

Runs to line 00402449 which ends with ASCII "/c del "

OllyDbg - Lab09-01.exe - [CPU - main thread, module Lab09-01]

C File View Debug Plugins Options Window Help

LEMTW H C / K B R ... S

```
00402410 $ 55 PUSH EBP
00402411 . 8BEC MOV EBP,ESP
00402413 . 81EC 08020000 SUB ESP,208
00402419 . 53 PUSH EBX
0040241A . 56 PUSH ESI
0040241B . 57 PUSH EDI
0040241C . 68 04010000 PUSH 104
00402421 . 8D85 F8FDFFFF LEA EAX,DWORD PTR SS:[EBP-208]
00402427 . 50 PUSH EAX
00402428 . 6A 00 PUSH 0
0040242A . FF15 38B04000 CALL DWORD PTR DS:[&KERNEL32.GetModule
00402430 . 68 04010000 PUSH 104
00402435 . 8D8D F8FDFFFF LEA ECX,DWORD PTR SS:[EBP-208]
0040243B . 51 PUSH ECX
0040243C . 8D95 F8FDFFFF LEA EDX,DWORD PTR SS:[EBP-208]
00402442 . 52 PUSH EDX
00402443 . FF15 3CB04000 CALL DWORD PTR DS:[&KERNEL32.GetShortP
00402449 . BF DCC04000 MOV EDI,Lab09-01.0040C0DC
0040244E . 8D95 FCFFEFFF LEA EDX,DWORD PTR SS:[EBP-104]
00402454 . 83C9 FF OR ECX,FFFFFFF
00402457 . 33C0 XOR EAX,EAX
00402459 . F2:AE REPNE SCAS BYTE PTR ES:[EDI]
0040245B . F701 NOT ECX
0040245D . 2BF9 SUB EDI,ECX
0040245F . 8BF7 MOV ESI,EDI
00402461 . 8BC1 MOV EAX,ECX
00402463 . 8BFA MOV EDI,EDX
00402465 . C1E9 02 SHR ECX,2
00402468 . F3:A5 REP MOVS DWORD PTR ES:[EDI],DWORD PTR DS:
0040246A . 8BC8 MOV ECX,EAX
0040246C . 83E1 03 AND ECX,3
0040246F . F3:A4 REP MOVS BYTF PTR FS:[EDI1].BYTF PTR DS:
0040C0DC=Lab09-01.0040C0DC (ASCII "/c del ")
EDI=00000000
```

Press F7 to 'play' the code forward

Stop at EDX 0012E610 ASCII "/c del " in Registers (FPU)

The screenshot shows the OllyDbg debugger interface. The assembly window displays the following code:

```
00402419: 53          PUSH EBX
0040241A: 56          PUSH ESI
0040241B: 57          PUSH EDI
0040241C: 68 04010000  PUSH 104
00402421: 8D85 F8FDFFFF LEA EDX,DWORD PTR SS:[EBP-208]
00402422: 50          PUSH EAX
00402423: 6A 00        PUSH 0
00402424: FF15 38B04000 CALL DWORD PTR DS:[&KERNEL32.GetModule
00402429: 68 04010000  PUSH 104
00402435: 8D8D F8FDFFFF LEA ECX,DWORD PTR SS:[EBP-208]
00402436: 51          PUSH ECX
00402437: 8D95 F8FDFFFF LEA EDX,DWORD PTR SS:[EBP-208]
00402438: 52          PUSH EDX
00402439: 8D95 3CB04000 CALL DWORD PTR DS:[&KERNEL32.GetShortP
0040243A: BF DCC04000 MOV EDI,Lab09-01.0040C0DC
0040243B: 8D95 FCFFEFEE LEA EDX,DWORD PTR SS:[EBP-104]
0040243C: 83C9 FF      OR ECX,FFFFFFFF
0040243D: 33C0          XOR ERX,ERX
0040243E: F2:RE        REPNE SCAS BYTE PTR ES:[EDI]
0040243F: F7D1          NOT ECX
00402440: 2BF9          SUB EDI,ECX
00402441: 8BF7          MOV ES1,EDI
00402442: 8BC1          MOV ERX,ECX
00402443: 8BF9          MOV EDI,EDX
00402444: C1E9 02      SHR ECX,2
00402445: F3:A5        REP MOVS DWORD PTR ES:[EDI],DWORD PTR DS:
00402446: 8BC8          MOV ECX,ERX
00402447: 83E1 03      AND ECX,3
00402448: F3:A4        REP MOVS BYTE PTR ES:[EDI],BYTE PTR DS:
00402449: 8D8D F8FDFFFF LEA EDI,DWORD PTR SS:[EBP-208]
0040244A: 8D95 FCFFEFEE LEA EDX,DWORD PTR SS:[EBP-104]
0040244B: 83C9 FF      OR ECX,FFFFFFFF
```

The registers window shows the following values:

Register	Value
ECX	00000000 (decimal 0)
DS:[ESI]=[0040C0E4]=75 ('u')	
ES:[EDI]=stack [0012E618]=39 ('9')	
EDX	0012E610 ASCII "/c del "
EBX	7FFDD0000
ESP	0012E5000
EIP	0040246F Lab09-01.0040246F
C	0 ES 0023 32bit 0(FFFFFFFF)
P	1 CS 0018 32bit 0(FFFFFFFF)
A	0 SS 0023 32bit 0(FFFFFFFF)
Z	1 DS 0023 32bit 0(FFFFFFFF)
S	0 FS 003B 32bit 7FFDF000(FFF)
T	0 GS 0000 NULL
D	0 0 LastErr: ERROR_SUCCESS (00000000)
EFL	00000246 (NO,NB,E,BE,NS,PE,G)
ST0	empty 0,0
ST1	empty 0,0
ST2	empty 0,0
ST3	empty 0,0
ST4	empty 0,0
ST5	empty 0,0
ST6	empty 0,0
ST7	empty 0,0
FST	0000 Cond 0 0 0 0 Err 0 0
FCW	027F Prec NEAR,53 Mask

Chapter 11: Malware Behaviour - Lab11-1

Source Files URL: <https://github.com/mikesiko/PracticalMalwareAnalysis-Labs>⁷⁴

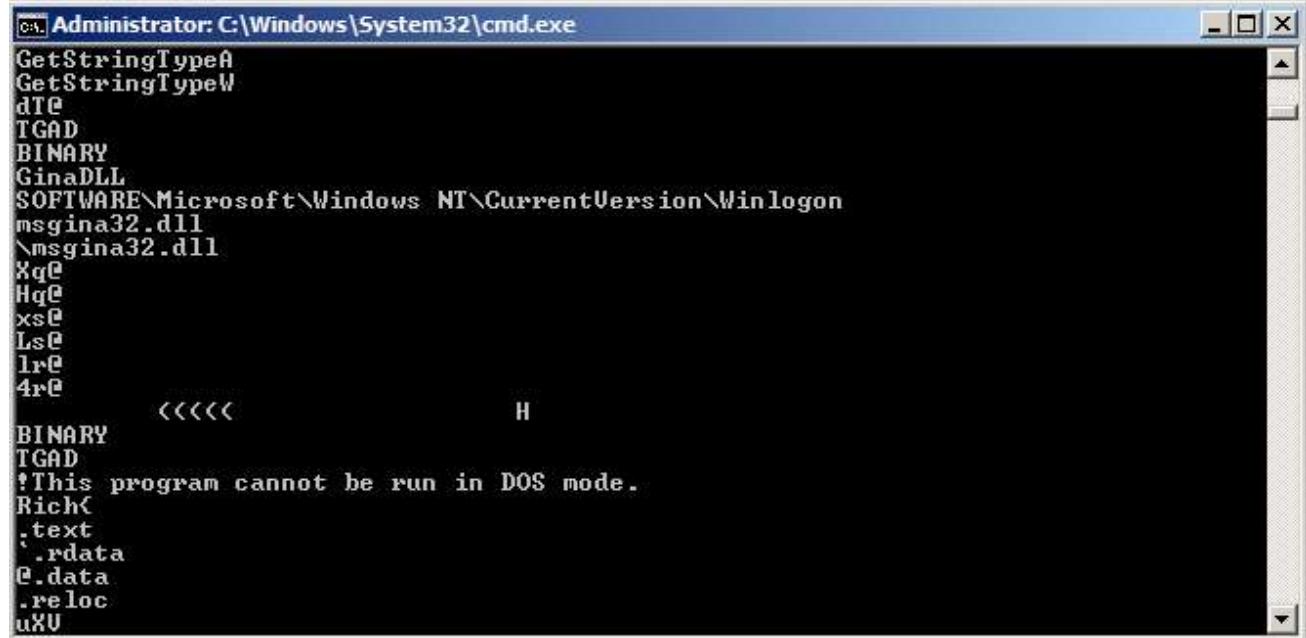
File: Chapter_11L\Lab-01.exe

Strings⁷⁵ Analysis

(Admin)C:\>strings C:\Users\Administrator\Downloads\Practical Malware

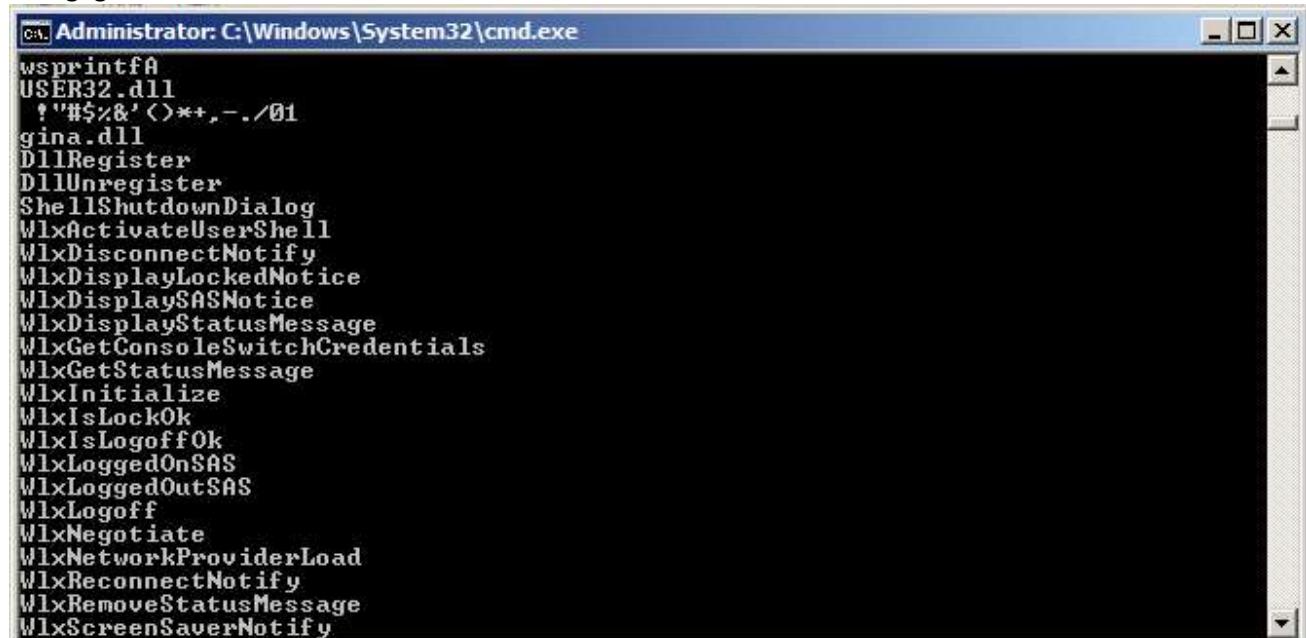
Analysis\BinaryCollection\Chapter_11L\Lab-01.exe

String: SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon



```
Administrator: C:\Windows\System32\cmd.exe
GetStringTypeA
GetStringTypeW
dT@
TGAD
BINARY
GinaDLL
SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
msgina32.dll
\msgina32.dll
Xq@
Hq@
xs@
Ls@
lr@
4r@      <<<<          H
BINAY
TGAD
!This program cannot be run in DOS mode.
RichC
.text
.rdata
@.data
.reloc
uXU
```

String: gina.DLL



```
Administrator: C:\Windows\System32\cmd.exe
wsprintfA
USER32.dll
!"#%&'<>*+,-./01
gina.dll
DllRegister
DllUnregister
ShellShutdownDialog
WlxActivateUserShell
WlxDisconnectNotify
WlxDisplayLockedNotice
WlxDisplaySASNotice
WlxDisplayStatusMessage
WlxGetConsoleSwitchCredentials
WlxGetStatusMessage
WlxInitialize
WlxIsLockOk
WlxIsLogoffOk
WlxLoggedOnSAS
WlxLoggedOutSAS
WlxLogoff
WlxNegotiate
WlxNetworkProviderLoad
WlxReconnectNotify
WlxRemoveStatusMessage
WlxScreenSaverNotify
```

⁷⁴ PracticalMalwareAnalysis-Labs (2017) *Binaries for the book Practical Malware Analysis*

<https://github.com/mikesiko/PracticalMalwareAnalysis-Labs> [Accessed 22nd May 2019].

⁷⁵ Microsoft.com (2019) *Strings* v2.53 <https://docs.microsoft.com/en-gb/sysinternals/downloads/strings> [Accessed 22nd May 2019].

BinText 3.03⁷⁶ Analysis

C:\Users\Administrator\Downloads\Practical Malware Analysis\BinaryCollection\Chapter_11L\Lab-01.exe

String: SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon

The screenshot shows the BinText 3.03 interface with the search term "SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon" entered. The results table lists several memory locations, their file positions, memory positions, IDs, and text content. The text column includes entries like "GetStringTypeW", "BINAR", "GinaDLL", and "SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon". A note in the text column states, "This program cannot be run in DOS mode." The search bar at the bottom contains "gina.dll".

File pos	Mem pos	ID	Text
A 000000000799E	00000040799E	0	GetStringTypeW
A 0000000008040	000000408040	0	BINAR
A 000000000804C	00000040804C	0	GinaDLL
A 0000000008054	000000408054	0	SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
A 0000000008090	000000408090	0	msgina32.dll
A 00000000080A4	0000004080A4	0	\msgina32.dll
A 000000000B0BD	00000040C0BD	0	This program cannot be run in DOS mode.
A 000000000B130	00000040C130	0	Rich{
A 000000000B240	00000040C240	0	.text
A 000000000B268	00000040C268	0	.rdata
A 000000000B28F	00000040C28F	0	@.data
A 000000000B2B8	00000040C2B8	0	.reloc
A 000000000B94A	00000040C94A	0	D\$ QRP
A 000000000B969	00000040C969	0	RQPHX2

String: gina.DLL

The screenshot shows the BinText 3.03 interface with the search term "gina.dll" entered. The results table lists several memory locations, their file positions, memory positions, IDs, and text content. The text column includes entries like "wsprintfA", "USER32.dll", "gina.dll", and "DllRegister". The search bar at the bottom contains "gina.dll".

File pos	Mem pos	ID	Text
A 000000000BF72	00000040CF72	0	wsprintfA
A 000000000BF7C	00000040CF7C	0	USER32.dll
A 000000000C118	00000040D118	0	gina.dll
A 000000000C121	00000040D121	0	DllRegister
A 000000000C12D	00000040D12D	0	DllUnregister
A 000000000C13B	00000040D13B	0	ShellShutdownDialog
A 000000000C14F	00000040D14F	0	WlxActivateUserShell
A 000000000C164	00000040D164	0	WlxDisconnectNotify
A 000000000C178	00000040D178	0	WlxDisplayLockedNotice
A 000000000C18F	00000040D18F	0	WlxDisplaySASNotice
A 000000000C1A3	00000040D1A3	0	WlxDisplayStatusMessage
A 000000000C1BB	00000040D1BB	0	WlxGetConsoleSwitchCredentials
A 000000000C1DA	00000040D1DA	0	WlxGetStatusMessage
A 000000000C1EE	00000040D1EE	0	WlxInitialize

⁷⁶ Softpedia.com (2019) *BinText 3.03* <https://www.softpedia.com/get/System/File-Management/BinText.shtml> [Accessed 22nd May 2019].

GINA Interception Malware

GINA-Graphical Identification and Authentication ⁷⁷

The Graphical Identification and Authentication (GINA) is a component of Windows 2000, Windows XP and Windows Server 2003 that provides secure authentication and interactive logon services
GINA is discontinued in Windows Vista

On Windows XP, GINA interception is a technique that malware uses to steal user credentials.
GINA is implemented in a DLL, msgina.dll, and is loaded by the Winlogon executable during the login process.

The GINA interception consists of injecting a malicious DLL between winlogon.exe and msgina.dll to intercept credentials

PEview⁷⁸ Static Analysis

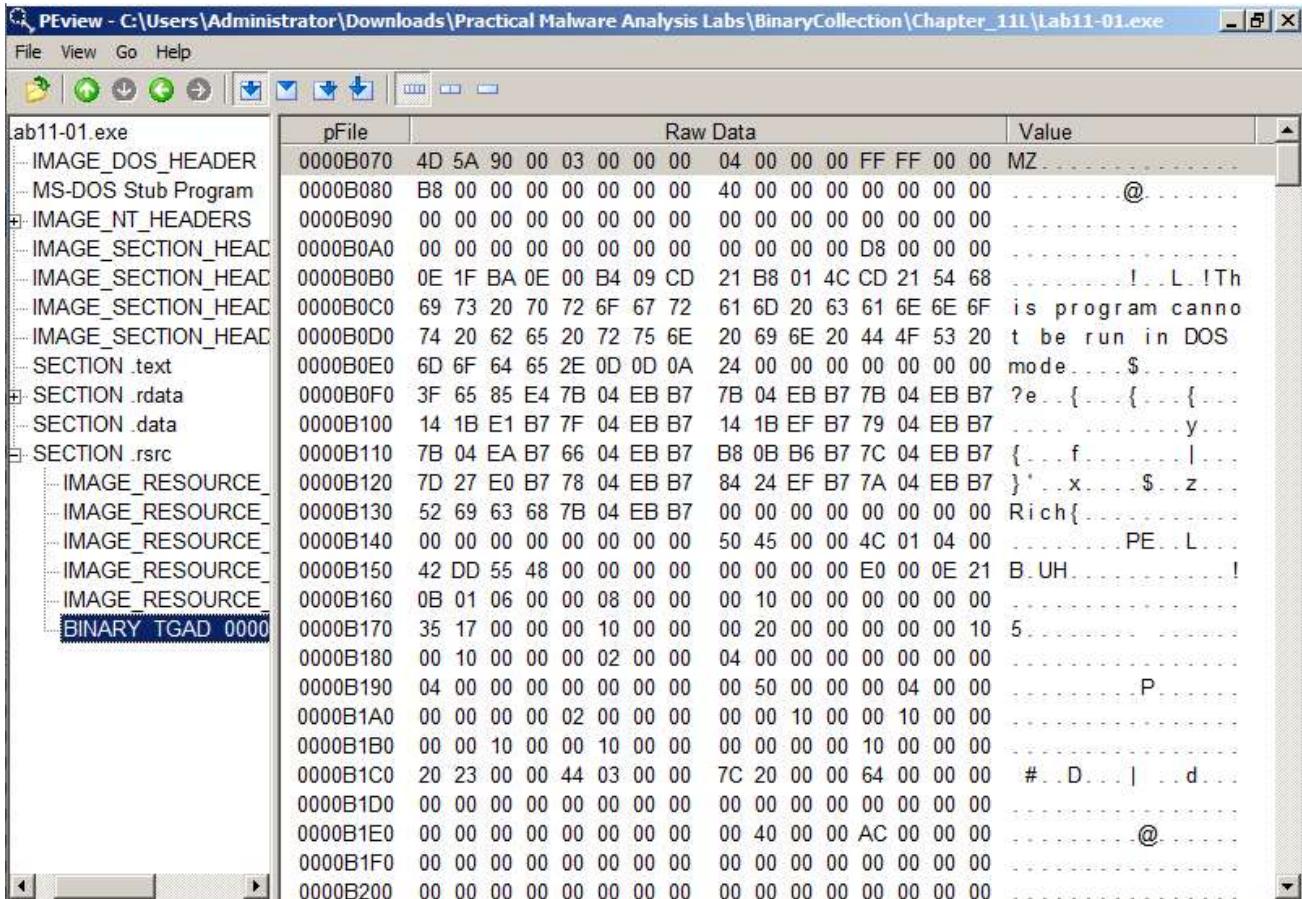
C:\Users\Administrator\Downloads\Practical Malware Analysis\BinaryCollection\Chapter_11L\Lab-01.exe

pFile	Data	Description	Value
00007000	000076AC	Hint/Name RVA	0186 RegSetValueExA
00007004	000076BE	Hint/Name RVA	015F RegCreateKeyExA
00007008	00000000	End of Imports	ADVAPI32.dll
0000700C	00007632	Hint/Name RVA	0295 SizeofResource
00007010	00007644	Hint/Name RVA	01D5 LockResource
00007014	00007654	Hint/Name RVA	01C7 LoadResource
00007018	00007622	Hint/Name RVA	02BB VirtualAlloc
0000701C	00007674	Hint/Name RVA	0124 GetModuleFileNameA
00007020	0000768A	Hint/Name RVA	0126 GetModuleHandleA
00007024	00007612	Hint/Name RVA	00B6 FreeResource
00007028	00007664	Hint/Name RVA	00A3 FindResourceA
0000702C	00007604	Hint/Name RVA	001B CloseHandle
00007030	000076DE	Hint/Name RVA	00CA GetCommandLineA
00007034	000076F0	Hint/Name RVA	0174 GetVersion
00007038	000076FE	Hint/Name RVA	007D ExitProcess
0000703C	0000770C	Hint/Name RVA	019F HeapFree
00007040	00007718	Hint/Name RVA	011A GetLastError
00007044	00007728	Hint/Name RVA	02DF WriteFile
00007048	00007734	Hint/Name RVA	029E TerminateProcess
0000704C	00007748	Hint/Name RVA	00F7 GetCurrentProcess
00007050	0000775C	Hint/Name RVA	02AD UnhandledExceptionFilter
00007054	00007778	Hint/Name RVA	00B2 FreeEnvironmentStringsA
00007058	00007792	Hint/Name RVA	00B3 FreeEnvironmentStringsW
0000705C	000077AC	Hint/Name RVA	02D2 WideCharToMultiByte
00007060	000077C2	Hint/Name RVA	0106 GetEnvironmentStrings

⁷⁷ Aldeid (2019) *GINA-Graphical Identification and Authentication* https://www.aldeid.com/wiki/GINA-Graphical_Identification_and_Authentication [Accessed 22nd May 2019].

⁷⁸ Radburn, Wayne J. (2018) PEview version 0.9.9 (.zip 31KB) <http://wjrdburn.com/software/> [Accessed 22nd May 2019].

BINARY TGAD 0000



Procmon⁷⁹ Dynamic Analysis

Run C:\Users\Administrator\Downloads\Practical Malware Analysis\BinaryCollection\Chapter_11\Lab-01.exe

Procmon > Filter > Reset Filter

Filter > Filter > Filter for: Process Name Lab11-01.exe > Add

Time of Day	Process Name	PID	Operation	Path	Result	Detail
14:56:48.3964339	Lab11-01.exe	3016	Process Start		SUCCESS	Parent PID: 1372, Command line: "C:\
14:56:48.3964445	Lab11-01.exe	3016	Thread Create		SUCCESS	Thread ID: 2868
14:56:48.4402257	Lab11-01.exe	3016	QueryNameInfo...C:\Users\Administrator\Downloads\Pr...	C:\Users\Administrator\Downloads\Pr...	SUCCESS	Name: \Users\Administrator\Download
14:56:48.4402749	Lab11-01.exe	3016	Load Image C:\Users\Administrator\Downloads\Pr...	C:\Users\Administrator\Downloads\Pr...	SUCCESS	Image Base: 0x400000, Image Size: 0
14:56:48.4403065	Lab11-01.exe	3016	Load Image C:\Windows\System32\ntdll.dll	C:\Windows\System32\ntdll.dll	SUCCESS	Image Base: 0x77b70000, Image Size:
14:56:48.4407392	Lab11-01.exe	3016	CreateFile C:\Users\Administrator\Downloads\Pr...	C:\Users\Administrator\Downloads\Pr...	SUCCESS	Desired Access: Execute/Traverse, S
14:56:48.4410236	Lab11-01.exe	3016	Load Image C:\Windows\System32\kernel32.dll	C:\Windows\System32\kernel32.dll	SUCCESS	Image Base: 0x774b0000, Image Size:
14:56:48.5091200	Lab11-01.exe	3016	Load Image C:\Windows\System32\advapi32.dll	C:\Windows\System32\advapi32.dll	SUCCESS	Image Base: 0x77770000, Image Size:
14:56:48.5093289	Lab11-01.exe	3016	Load Image C:\Windows\System32\vpct4.dll	C:\Windows\System32\vpct4.dll	SUCCESS	Image Base: 0x77840000, Image Size:
14:56:48.5104402	Lab11-01.exe	3016	RegOpenKey HKLM\System\CurrentControlSet\Contr...	HKLM\System\CurrentControlSet\Contr...	REPARSE	Desired Access: Read
14:56:48.5104738	Lab11-01.exe	3016	RegOpenKey HKLM\System\CurrentControlSet\Contr...	HKLM\System\CurrentControlSet\Contr...	SUCCESS	Desired Access: Read
14:56:48.5105053	Lab11-01.exe	3016	RegQueryValue HKLM\System\CurrentControlSet\Contr...	HKLM\System\CurrentControlSet\Contr...	NAME NO...	Length: 548
14:56:48.5105257	Lab11-01.exe	3016	RegQueryValue HKLM\System\CurrentControlSet\Contr...	HKLM\System\CurrentControlSet\Contr...	SUCCESS	Type: REG_DWORD, Length: 4, Data
14:56:48.5105450	Lab11-01.exe	3016	RegCloseKey HKLM\System\CurrentControlSet\Contr...	HKLM\System\CurrentControlSet\Contr...	SUCCESS	
14:56:48.5105897	Lab11-01.exe	3016	RegOpenKey HKLM\Software\Microsoft\Window...	HKLM\Software\Microsoft\Window...	SUCCESS	Desired Access: Read
14:56:48.5106249	Lab11-01.exe	3016	RegQueryValue HKLM\Software\Microsoft\Window...	HKLM\Software\Microsoft\Window...	NAME NO...	Length: 144
14:56:48.5106450	Lab11-01.exe	3016	RegCloseKey HKLM\Software\Microsoft\Window...	HKLM\Software\Microsoft\Window...	SUCCESS	
14:56:48.5106928	Lab11-01.exe	3016	RegOpenKey HKLM\System\Setup	HKLM\System\Setup	SUCCESS	Desired Access: Read
14:56:48.5107193	Lab11-01.exe	3016	RegQueryValue HKLM\SYSTEM\Setup\SystemSetupIn...	HKLM\SYSTEM\Setup\SystemSetupIn...	SUCCESS	Type: REG_DWORD, Length: 4, Data

⁷⁹ Microsoft.com (2019) *Process Monitor v3.52* <https://docs.microsoft.com/en-gb/sysinternals/downloads/procmon> [Accessed 20th May 2019].

Key Procmon Entries

CreateFile ... msgina32.dll or IRP_MU_CREATE ... msgina.dll
RegCreateKey HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
RegSetValue HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL

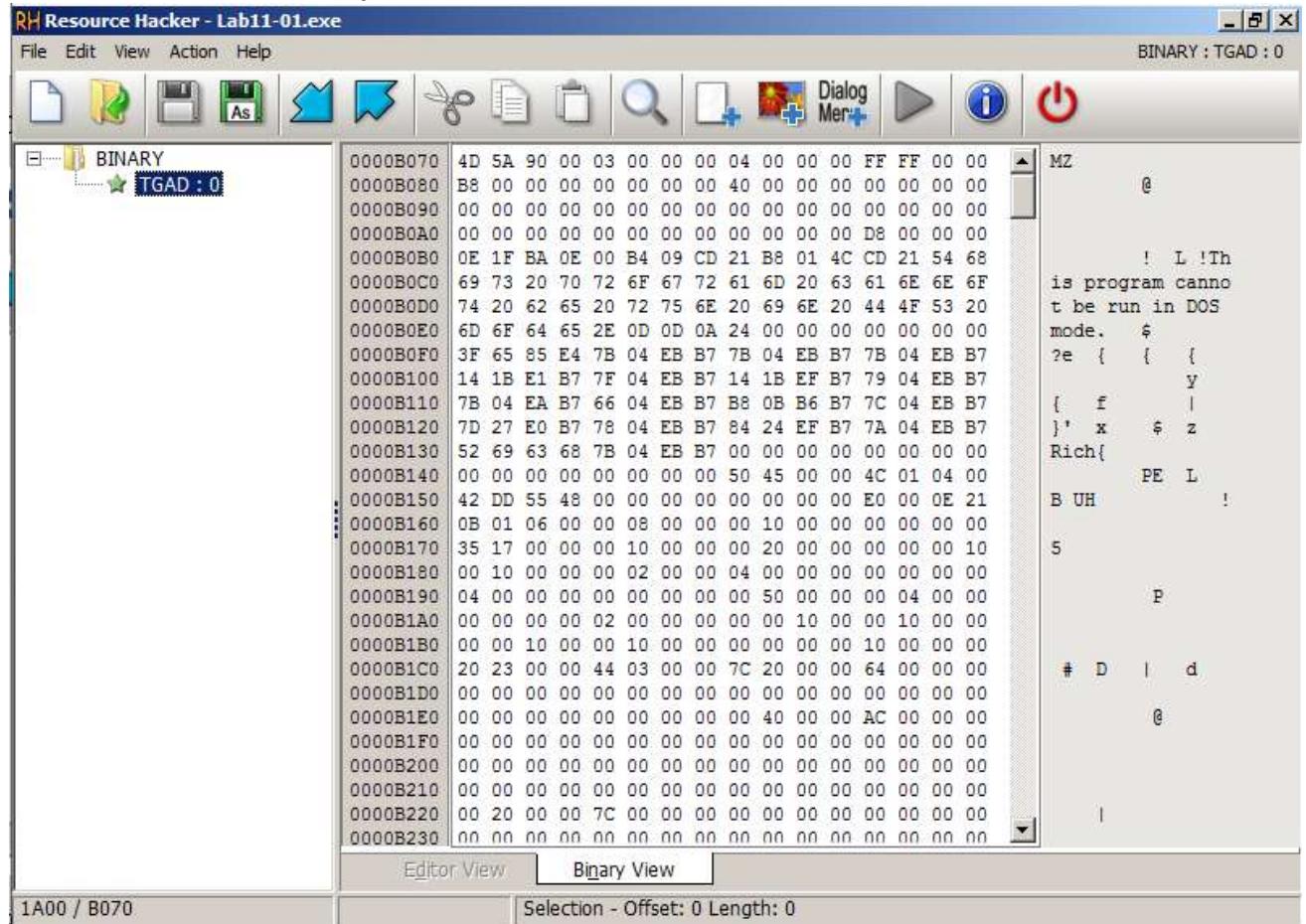
File created called msgina.dll

Registry key created

Registry path added to key

DLL will launch on next boot

Resource Hacker 5.1.7⁸⁰ Analysis

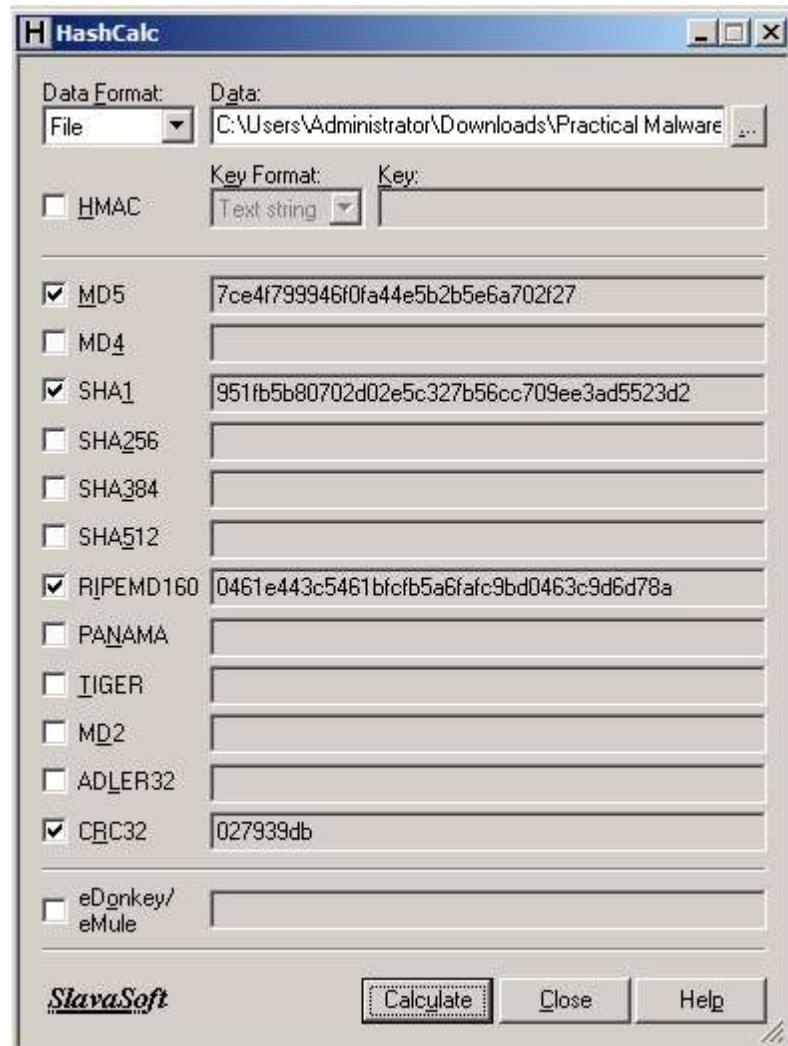


Resource Hacker > Action > Save *.bin resource... > Lab11-01-TAGD.exe

⁸⁰ Resource Hacker (2019) *Resource Hacker 5.1.7* <http://www.angusj.com/resourcehacker/> [Accessed 20th May 2019].

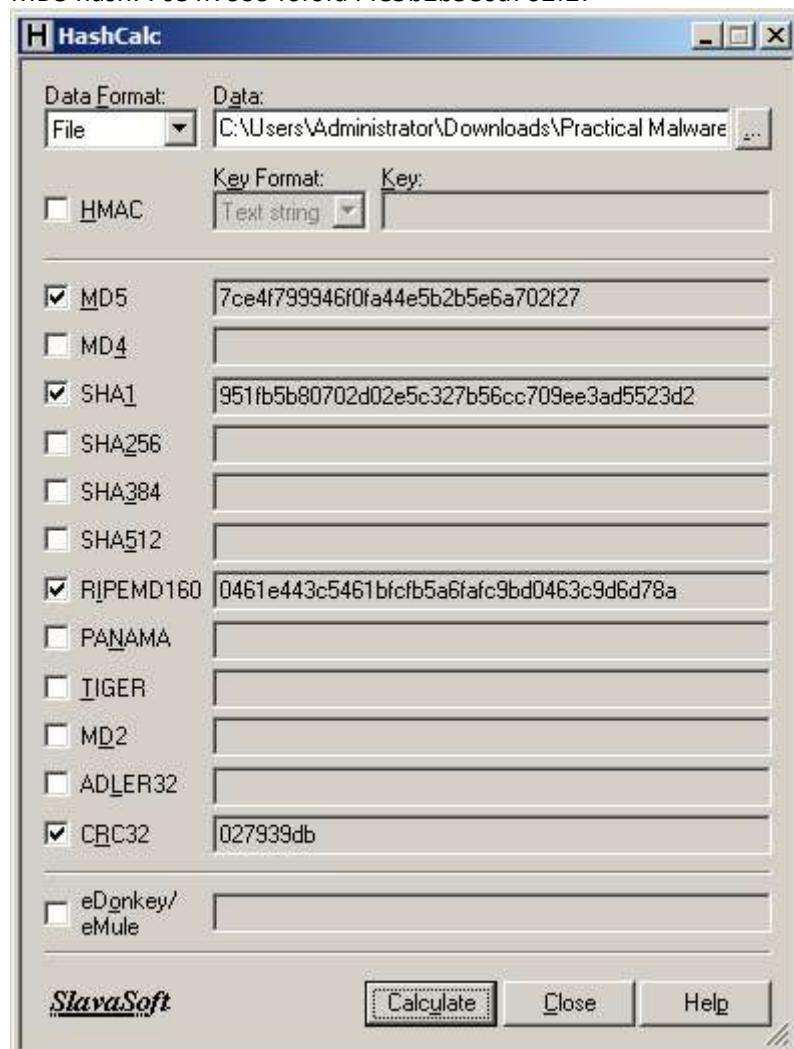
HashCalc 2.02⁸¹ Analysis

MD5 Hash of C:\Users\Administrator\Downloads\Practical Malware Analysis\BinaryCollection\Chapter_11\msgina32.dll
MD5 hash: 7ce4f799946f0fa44e5b2b5e6a702f27



⁸¹ SlavaSoft (2019) HashCalc 2.02 <http://www.slavasoft.com/hashcalc/> [Accessed 20th May 2019].

MD5 Hash of C:\Users\Administrator\Downloads\Practical Malware Analysis\BinaryCollection\Chapter_11L\Lab11-01-TAGD.exe
MD5 hash: 7ce4f799946f0fa44e5b2b5e6a702f27



Chapter 12: Covert Malware Launching - Lab12-1

Source Files URL: <https://github.com/mikesiko/PracticalMalwareAnalysis-Labs>⁸²

File: Lab12-01.exe

PEView⁸³ Analysis

C:\Users\Administrator\Downloads\Practical Malware Analysis\BinaryCollection\Chapter_11L\Lab12-01.exe

pFile	Data	Description	Value
00005000	0000552C	Hint/Name RVA	001B CloseHandle
00005004	0000553A	Hint/Name RVA	01EF OpenProcess
00005008	00005548	Hint/Name RVA	0046 CreateRemoteThread
0000500C	0000555E	Hint/Name RVA	0126 GetModuleHandleA
00005010	00005572	Hint/Name RVA	02E9 WriteProcessMemory
00005014	00005588	Hint/Name RVA	02BC VirtualAllocEx
00005018	0000559A	Hint/Name RVA	02F9 IstrcatA
0000501C	000055A6	Hint/Name RVA	00F5 GetCurrentDirectoryA
00005020	000055BE	Hint/Name RVA	013E GetProcAddress
00005024	000055D0	Hint/Name RVA	01C2 LoadLibraryA
00005028	000055EE	Hint/Name RVA	00CA GetCommandLineA
0000502C	00005600	Hint/Name RVA	0174 GetVersion
00005030	0000560E	Hint/Name RVA	007D ExitProcess
00005034	0000561C	Hint/Name RVA	029E TerminateProcess
00005038	00005630	Hint/Name RVA	00F7 GetCurrentProcess
0000503C	00005644	Hint/Name RVA	02AD UnhandledExceptionFilter
00005040	00005660	Hint/Name RVA	0124 GetModuleFileNameA
00005044	00005676	Hint/Name RVA	00B2 FreeEnvironmentStringsA
00005048	00005690	Hint/Name RVA	00B3 FreeEnvironmentStringsW
0000504C	000056AA	Hint/Name RVA	02D2 WideCharToMultiByte
00005050	000056C0	Hint/Name RVA	0106 GetEnvironmentStrings
00005054	000056D8	Hint/Name RVA	0108 GetEnvironmentStringsW
00005058	000056F2	Hint/Name RVA	026D SetHandleCount
0000505C	00005704	Hint/Name RVA	0152 GetStdHandle
00005060	00005714	Hint/Name RVA	0115 GetFileType

⁸² PracticalMalwareAnalysis-Labs (2017) *Binaries for the book Practical Malware Analysis*

<https://github.com/mikesiko/PracticalMalwareAnalysis-Labs> [Accessed 22nd May 2019].

⁸³ Radburn, Wayne J. (2018) PEview version 0.9.9 (.zip 31KB) <http://wjrdburn.com/software/> [Accessed 22nd May 2019].

Strings⁸⁴ Analysis

C:\Users\Administrator\Downloads\Practical Malware Analysis\BinaryCollection\Chapter_11L\Lab12-01.exe

```
Administrator: C:\Windows\System32\cmd.exe
C:\Users\Administrator\Downloads\Practical Malware Analysis\BinaryCollection\Chapter_12L>strings Lab12-01.exe

Strings v2.53 - Search for ANSI and Unicode strings in binary images.
Copyright (C) 1999-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

!This program cannot be run in DOS mode.

RichLu
;text
;.rdata
@.data
@e
D'e
H'e
h0`e
$Pe
`Pe
$Pe
`Pe
$Pe
`Pe
$Pe
`Pe
h1`e
h1`e
h\`e
hL`e
`Pe
$UW
`Pe
<Pe
0Pe
```

```
Administrator: C:\Windows\System32\cmd.exe
SetHandleCount
GetStdHandle
GetFileType
GetStartupInfoA
GetEnvironmentVariableA
GetVersionExA
HeapDestroy
HeapCreate
VirtualFree
HeapFree
RtlUnwind
WriteFile
HeapAlloc
GetCPInfo
GetACP
GetOEMCP
VirtualAlloc
HeapReAlloc
MultiByteToWideChar
LCMapStringA
LCMapStringW
GetStringTypeA
GetStringTypeW
explorer.exe
<unknown>
LoadLibraryA
kernel32.dll
Lab12-01.dll
EnumProcesses
GetModuleBaseNameA
psapi.dll
EnumProcessModules
WS2_32.dll
```

⁸⁴ Microsoft.com (2019) *Strings* v2.53 <https://docs.microsoft.com/en-gb/sysinternals/downloads/strings> [Accessed 22nd May 2019].

IDA Pro⁸⁵ Analysis

C:\Users\Administrator\Downloads\Practical Malware Analysis\BinaryCollection\Chapter_11L\Lab12-01.exe
Options > General > Line Prefixes > Number of opcode bytes: 6

```
0040111F push    offset ProcName ; "EnumProcessModules"
00401124 push    offset LibFileName ; "psapi.dll"
00401129 call    ds:LoadLibraryA
0040112F push    eax      ; hModule
00401130 call    ds:GetProcAddress
00401136 mov     dword_408714, eax
0040113B push    offset aGetmodulebasen ; "GetModuleBaseNameA"
00401140 push    offset LibFileName ; "psapi.dll"
00401145 call    ds:LoadLibraryA
0040114B push    eax      ; hModule
0040114C call    ds:GetProcAddress
00401152 mov     dword_40870C, eax
00401157 push    offset aEnumprocesses ; "EnumProcesses"
0040115C push    offset LibFileName ; "psapi.dll"
```

```
00401157 push    offset aEnumprocesses ; "EnumProcesses"
0040115C push    offset LibFileName ; "psapi.dll"
00401161 call    ds:LoadLibraryA
00401167 push    eax      ; hModule
00401168 call    ds:GetProcAddress
0040116E mov     dword_408710, eax
00401173 lea     ecx, [ebp+Buffer]
00401179 push    ecx      ; lpBuffer
0040117A push    104h    ; nBufferLength
0040117F call    ds:GetCurrentDirectoryA
00401185 push    offset String2 ; "\\"
0040118A lea     edx, [ebp+Buffer]
00401190 push    edx      ; lpString1
00401191 call    ds:lstrcmpA
```

The code above references psapi three times

00401124 push offset LibFileName ; "psapi.dll"

00401140 push offset LibFileName ; "psapi.dll"

0040115C push offset LibFileName ; "psapi.dll"

Psapi is attempting to locate a Windows API function and store its address in a numerical address
This obfuscates the code, so later calls to these functions will be difficult to recognize

⁸⁵ Hex-Rays (2019) *IDA Freeware for Windows (48 MB)* https://www.hex-rays.com/products/ida/support/download_freeware.shtml
[Accessed 22nd May 2019].

Assign Memory Addresses Labels

Address: 00401136

Data: dword_408714

Renamed to: myEnumProcessModules

Address: 00401152

Data: dword_40870C

Renamed to: myGetModuleBaseNameA

Address: 0040116E

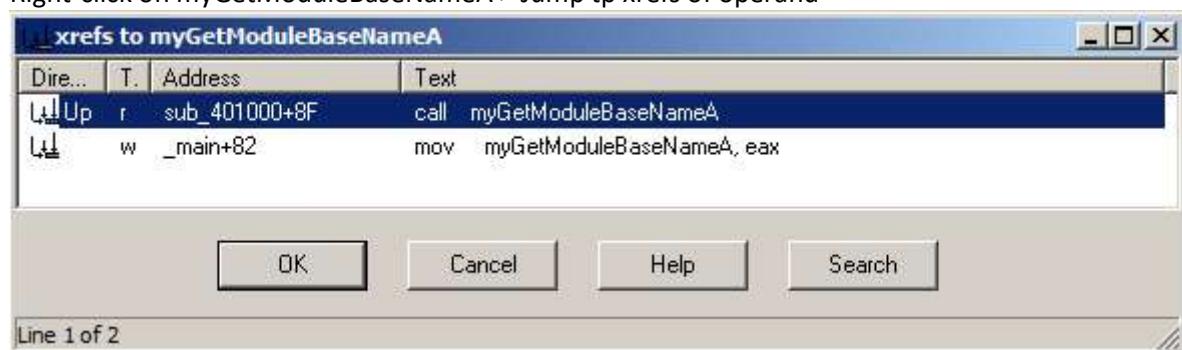
Data: dword_408710

Renamed to: myEnumProcesses

The screenshot shows the assembly dump in IDA Pro. The code is as follows:

```
00401136 myEnumProcessModules:
00401136    mov     dword_408714, eax
0040113B    push    offset aGetmodulebasen ; "GetModuleBaseNameA"
00401140    push    offset LibFileName ; "psapi.dll"
00401145    call    ds:LoadLibraryA
00401148    push    eax             ; hModule
0040114C    call    ds:GetProcAddress
00401152 myGetModuleBaseNameA:
00401152    mov     dword_40870C, eax
00401157    push    offset aEnumprocesses ; "EnumProcesses"
0040115C    push    offset LibFileName ; "psapi.dll"
00401161    call    ds:LoadLibraryA
00401167    push    eax             ; hModule
00401168    call    ds:GetProcAddress
0040116E myEnumProcesses:
```

Right-click on myGetModuleBaseNameA > Jump tp xrefs of operand



Routine enumerates modules and compares each module name to "explorer.exe", to find the module into which to inject code

```

00401078 68 04 01 00 00    push    104h
0040107D 8D 8D F8 FE FF FF lea     ecx, [ebp+var_108]
00401083 51                push    ecx
00401084 8B 95 F4 FE FF FF mov     edx, [ebp+var_10C]
0040108A 52                push    edx
0040108B 8B 45 FC          mov     eax, [ebp+hObject]
0040108E 50                push    eax
0040108F FF 15 0C 87 40 00 call   myGetModuleBaseNameA

```

```

00401095
00401095      loc_401095:           ; size_t
00401095 6A 0C                push    0Ch
00401097 68 30 60 40 00        push    offset aExplorer_exe ; "explorer.exe"
0040109C 8D 8D F8 FE FF FF lea     ecx, [ebp+var_108]
004010A2 51                push    ecx           ; char *
004010A3 E8 B8 37 00 00        call   _strnicmp

```

Process Explorer⁸⁶ Analysis

Select explorer.exe

View > Show Lower Pane

View > Lower Pane View > DLLs

Locate Lab12-01.dll which has been injected into explorer.exe as follows

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
explorer.exe	< 0.01	24,268 K	36,140 K	1372	Windows Explorer	Microsoft Corpora
jusched.exe		1,284 K	4,424 K	404	Java(TM) Platform SE binary	Sun Microsystems
jucheck.exe		3,236 K	6,708 K	3028	Java(TM) Update Checker	Sun Microsystems
VBoxTray.exe	< 0.01	2,392 K	4,788 K	2056	VirtualBox Guest Additions Tr...	Oracle Corporatio
proexp.exe	4.48	14,864 K	22,232 K	2548	Sysinternals Process Explorer	Sysinternals - ww
conime.exe		764 K	3,128 K	3520	Console IME	Microsoft Corpora

Name	Description	Company Name	Path
kernel32.dll	Windows NT BASE API Client DLL	Microsoft Corporation	C:\Windows\System32\kernel32.dll
Lab12-01.dll			C:\Users\Administrator\Downloads\Practical Malwa
loc2008.nls			C:\Windows\System32\loc2008.nls
loc2008.nls			C:\Windows\System32\loc2008.nls
lpk.dll	Language Pack	Microsoft Corporation	C:\Windows\System32\lpk.dll
mlang.dll	Multi Language Support DLL	Microsoft Corporation	C:\Windows\System32\mlang.dll
MMDevAPI.dll	MMDevice API	Microsoft Corporation	C:\Windows\System32\MMDevAPI.dll
mpr.dll	Multiple Provider Router DLL	Microsoft Corporation	C:\Windows\System32\mpr.dll

CPU Usage: 4.48% Commit Charge: 7.02% Processes: 39 Physical Usage: 17.39%

⁸⁶ Microsoft.com (2019) *Process Explorer v16.26* <https://docs.microsoft.com/en-gb/sysinternals/downloads/process-explorer> [Accessed 20th May 2019].

Chapter 13: Data Encoding - Lab13-1

Source Files URL: <https://github.com/mikesiko/PracticalMalwareAnalysis-Labs>⁸⁷

File: Lab13-01.exe

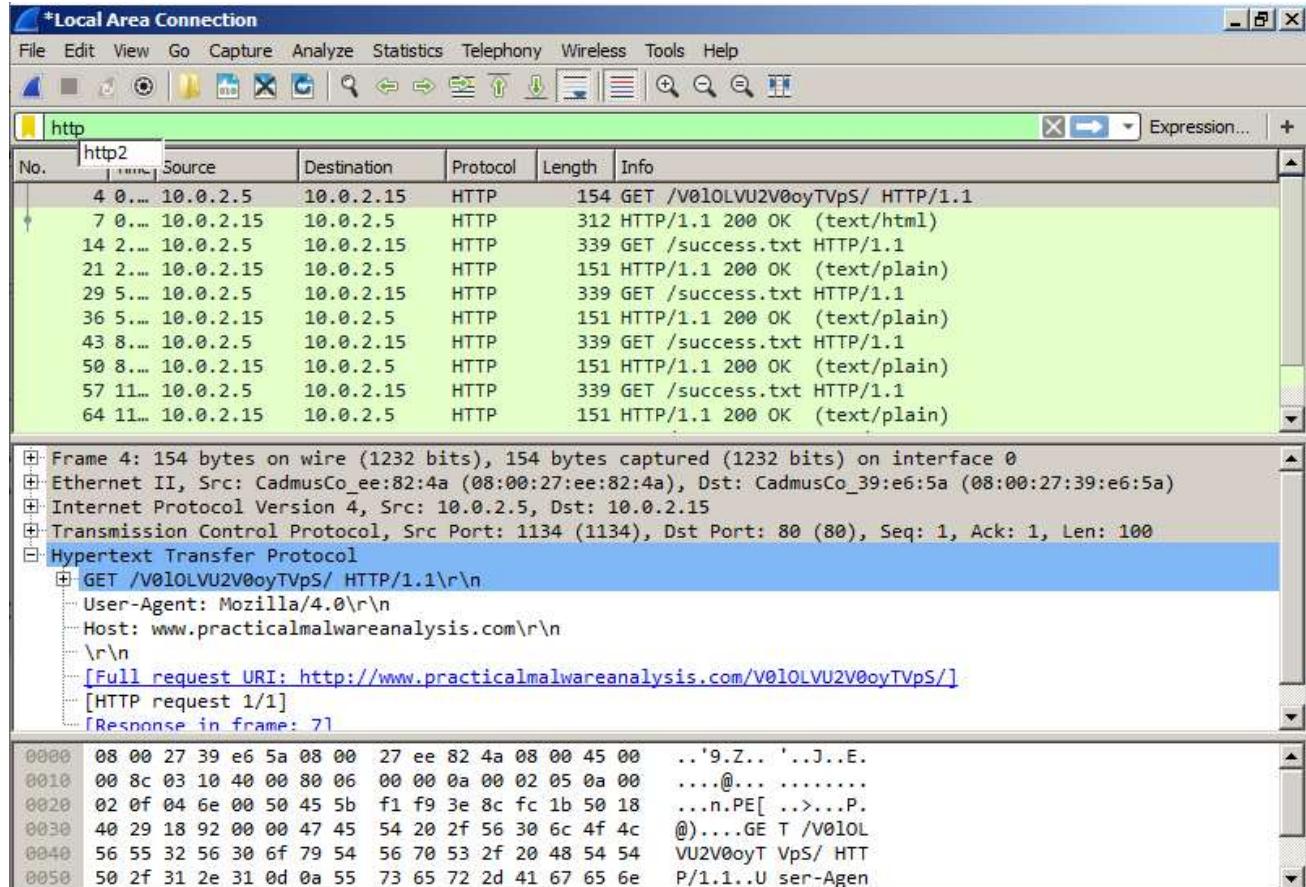
Kali KDE: start iNetSim

Windows Server 2008: set Primary DNS to Kali IP

Windows Server 2008: Firefox > <http://mysite.com> > INetSim default HTML page : OK

Wireshark⁸⁸ Capture

Start Wireshark capture > stop capture > filter for http

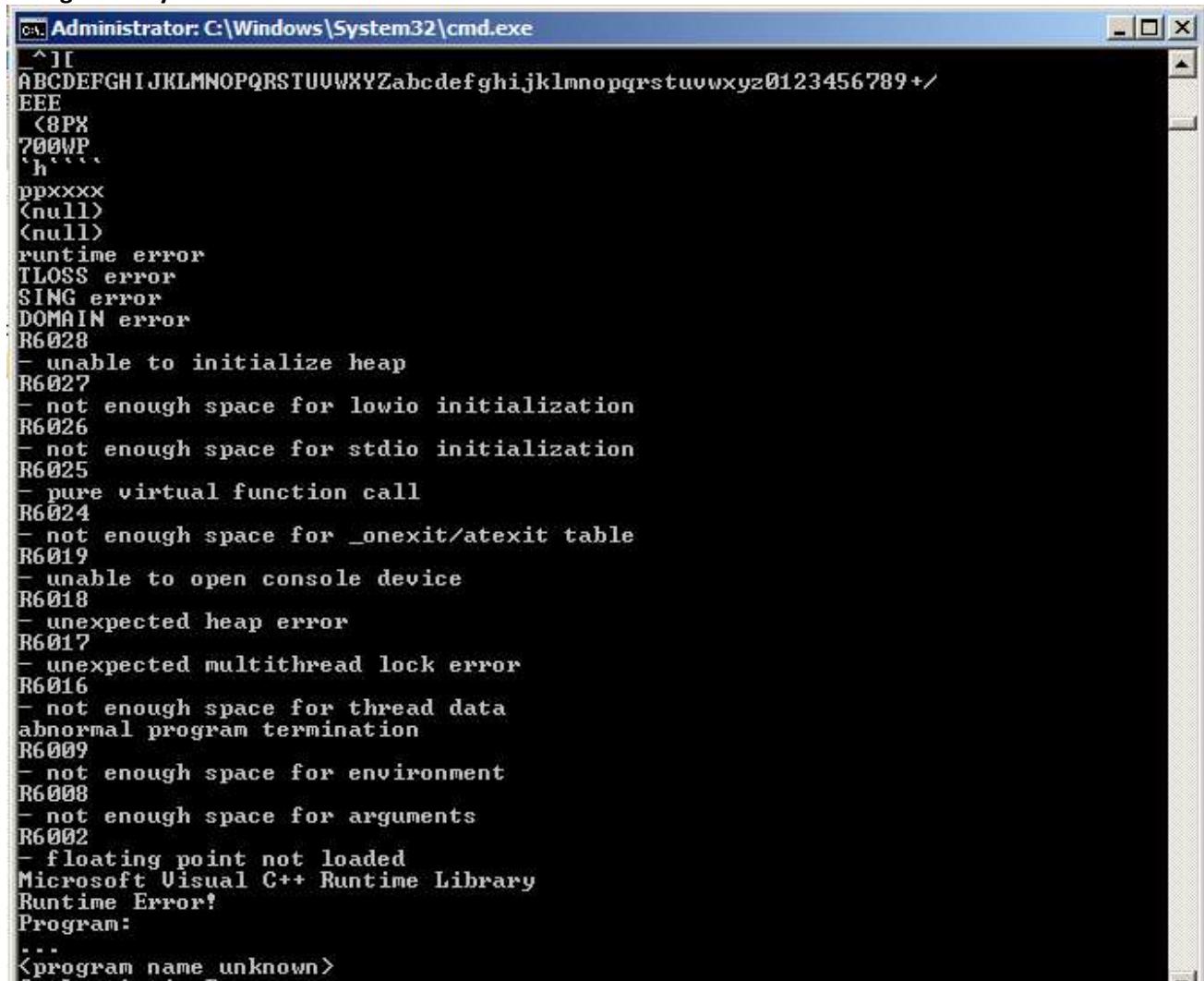


⁸⁷ PracticalMalwareAnalysis-Labs (2017) *Binaries for the book Practical Malware Analysis*

<https://github.com/mikesiko/PracticalMalwareAnalysis-Labs> [Accessed 22nd May 2019].

⁸⁸ Wireshark.org (2019) *Wireshark-win32-2.0.0.exe* <https://1.eu.dl.wireshark.org/win32/all-versions/> [Accessed 20th May 2019].

Strings⁸⁹ Analysis



The screenshot shows a Windows Command Prompt window titled "Administrator: C:\Windows\System32\cmd.exe". The window displays the output of the "strings" command, which extracts text strings from memory dump files. The output includes various error codes and messages such as R6028, R6027, R6026, R6025, R6024, R6019, R6018, R6017, R6016, R6009, R6008, R6002, and R6000. It also shows the Microsoft Visual C++ Runtime Library and a Program section with a placeholder <program name unknown>. The strings are color-coded in blue, red, green, and yellow.

```
Administrator: C:\Windows\System32\cmd.exe
^[[A
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
EEE
<8PX
?00WP
`h
ppxxxx
<null>
<null>
runtime error
TLOSS error
SING error
DOMAIN error
R6028
- unable to initialize heap
R6027
- not enough space for lowio initialization
R6026
- not enough space for stdio initialization
R6025
- pure virtual function call
R6024
- not enough space for _onexit/atexit table
R6019
- unable to open console device
R6018
- unexpected heap error
R6017
- unexpected multithread lock error
R6016
- not enough space for thread data
abnormal program termination
R6009
- not enough space for environment
R6008
- not enough space for arguments
R6002
- floating point not loaded
Microsoft Visual C++ Runtime Library
Runtime Error!
Program:
...
<program name unknown>
```

⁸⁹ Microsoft.com (2019) *Strings* v2.53 <https://docs.microsoft.com/en-gb/sysinternals/downloads/strings> [Accessed 22nd May 2019].

IDA Pro⁹⁰ Analysis

Options > General > Line Prefixes > OK

Click IDA View-A window > Search > Text > xor > Find all occurrences

Address	Instruction
.text:00401007	xor ecx,ecx
.text:0040101C	xor edx,edx
.text:00401029	xor ecx,ecx
.text:0040104E	xor eax,eax
.text:0040105C	xor edx,edx
.text:0040108D	xor ecx,ecx
.text:004011B4	xor eax,eax
.text:004011B8	xor eax,3Bh
.text:004011D6	xor eax,eax
.text:004012A2	xor al,al
.text:004012E6	xor al,al
.text:004012FA	xor al,al
.text:00401332	xor eax,eax
.text:00401350	xor eax,eax

Double-click xor eax, 3Bh instruction. This function performs xor encoding

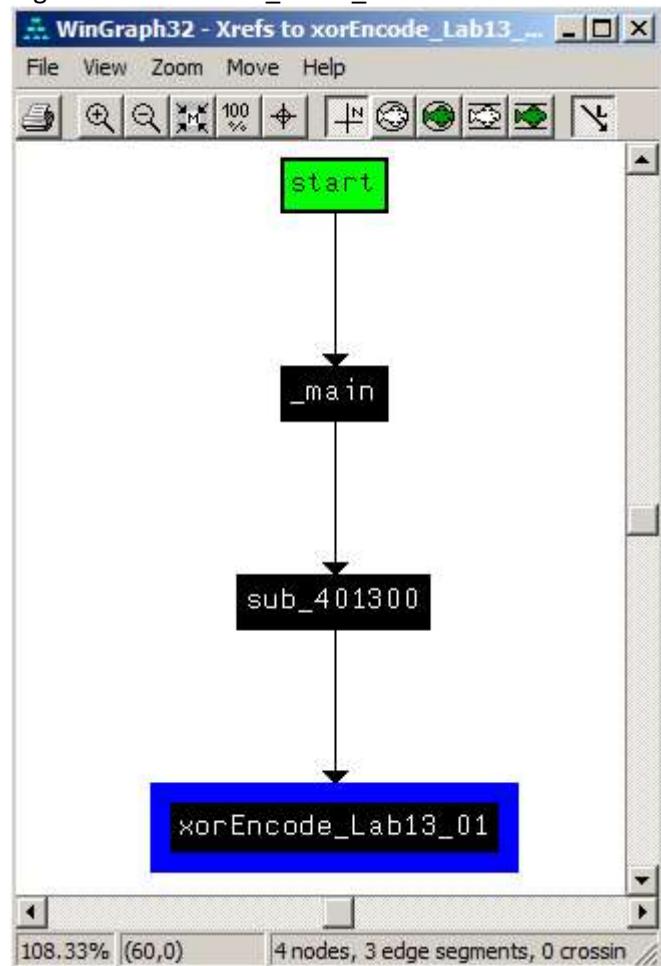
Rename sub_401190 to xorEncode_Lab13_01

```
00401190
00401190
00401190 ; Attributes: bp-based frame
00401190
00401190 xorEncode_Lab13_01 proc near
00401190
00401190 var_4= dword ptr -4
00401190 arg_0= dword ptr 8
00401190 arg_4= dword ptr 0Ch
00401190
00401190 push    ebp
00401191 mov     ebp, esp
00401193 push    ecx
00401194 mov     [ebp+var_4], 0
0040119B jmp     short loc_4011A6

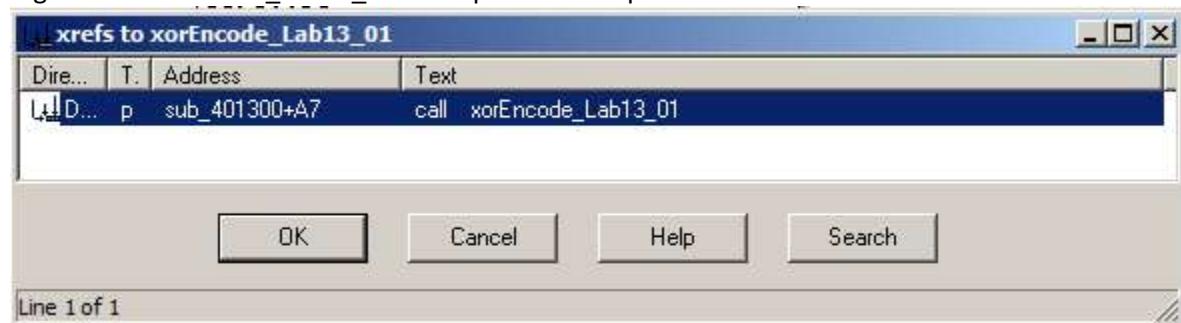
004011A6 loc_4011A6:
```

⁹⁰ Hex-Rays (2019) *IDA Freeware for Windows (48 MB)* https://www.hex-rays.com/products/ida/support/download_freeware.shtml [Accessed 22nd May 2019].

Right-click xorEncode_Lab13_01 > Chart of xrefs to



Right-click xorEncode_Lab13_01 > Jump to xref to operand



PEview⁹¹ Analysis

PEView > SECTION .rsrc > RCDATA 0065 0409

Locate starting address 00007060

The screenshot shows the PEview interface. On the left is a tree view of the file structure:

- Lab13-01.exe
 - IMAGE_DOS_HEADER
 - MS-DOS Stub Program
 - + IMAGE_NT_HEADERS
 - IMAGE_SECTION_HEADER
 - IMAGE_SECTION_HEADER
 - IMAGE_SECTION_HEADER
 - IMAGE_SECTION_HEADER
 - IMAGE_SECTION_HEADER
 - SECTION .text
 - + SECTION .rdata
 - SECTION .data
 - SECTION .rsrc
 - IMAGE_RESOURCE_I
 - IMAGE_RESOURCE_I
 - IMAGE_RESOURCE_I
 - IMAGE_RESOURCE_I
 - RCDATA 0065 0409

On the right, there are three tabs: pFile, Raw Data, and Value. The Raw Data tab is selected, showing the memory dump. The Value tab displays the ASCII representation of the data.

pFile	Raw Data	Value
00007060	4C 4C 4C 15 4B 49 5A 58 4F 52 58 5A 57 56 5A 57	LLL KIZXORXZWZW
00007070	4C 5A 49 5E 5A 55 5A 57 42 48 52 48 15 58 54 56	LZI^ZUZWBHRH XTV

WinHex⁹² Analysis

Setup.exe > File > Open > Lab13-01.exe

Highlight bytes 0028768 through 0028784

The screenshot shows the WinHex interface. The menu bar includes File, Edit, Search, View, Tools, Specialist, Options, Window, and Help. The version is 19.8 SR-4.

The main window displays a table of memory data:

Offset	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	ANSI ASCII
00028768	4C 4C 4C 15 4B 49 5A 58 4F 52 58 5A 57 56 5A 57	LLL KIZXORXZWZW
00028784	4C 5A 49 5E 5A 55 5A 57 42 48 52 48 15 58 54 56	LZI^ZUZWBHRH XTV

Edit > Modify Data > Modify Block Data > check XOR > enter 3B

The screenshot shows the WinHex interface after modifying the data. The menu bar includes File, Edit, Search, View, Tools, Specialist, Options, Window, and Help. The version is 19.8 SR-4.

The main window displays a table of memory data:

Offset	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	ANSI ASCII
00028768	77 77 77 2E 70 72 61 63 74 69 63 61 6C 6D 61 6C	www.practicalmal
00028784	77 61 72 65 61 6E 61 6C 79 73 69 73 2E 63 6F 6D	wareanalysis.com

⁹¹ Radburn, Wayne J. (2018) PEview version 0.9.9 (.zip 31KB) <http://wjjradburn.com/software/> [Accessed 22nd May 2019].

⁹² WinHex.com (2019) WinHex: Computer Forensics & Data Recovery Software, Hex Editor & Disk Editor <http://winhex.com/winhex/> [Accessed 22nd May 2019].

References

1	Malware Types & Lab Setups File	(2019)	Gateway OS Slide 24	https://moodle.ncirl.ie/mod/resource/view.php?id=59717	[Accessed 1st July 2019].
2	REMnux.org	(2019)	REMnux: A Linux Toolkit for Reverse-Engineering and Analyzing Malware	https://remnux.org/	[Accessed 1st July 2019].
3	VirtualBox.org	(2019)	VirtualBox	https://www.virtualbox.org/	[Accessed 1st July 2019].
4	Jonathon Taaffe	(2019)	Table 1. malware-analysis-network01 Static Assignments		[Created 1st July 2019].
5	Portal.Azure.com	(2019)	Windows Server 2016 Standard	http://dl.msdn.com/pr/en_windows_server_2016_updated_feb_2018_x64_dvd_11636692.iso	[Accessed 1st July 2019].
6	Malware Types & Lab Setups File	(2019)	Tools Slide 28	https://moodle.ncirl.ie/mod/resource/view.php?id=59717	[Accessed 1st July 2019].
7	Jonathon Taaffe	(2019)	Table 2. Windows Server 2016 File Server VM Configuration		[Created 1st July 2019].
8	Tecting	(2019)	How to Share Files and Folders in Windows Server 2016	https://www.tactig.com/share-files-folders-windows-server-2016/	[Accessed 1st July 2019].
9	Malware Types & Lab Setups File	(2019)	Gateway OS Slide 24	https://moodle.ncirl.ie/mod/resource/view.php?id=59717	[Accessed 1st July 2019].
10	Malware Types & Lab Setups File	(2019)	FakeNet-NG Slide 25	https://moodle.ncirl.ie/mod/resource/view.php?id=59717	[Accessed 1st July 2019].
11	Jonathon Taaffe	(2019)	Table 3. Malware Analysis Utilities Static IP Assignment		[Created 1st July 2019].
12	REMnux 6.0 OVA Public	(2019)	remnux-6.0-ova-public.ova (2.0G)	https://docs.google.com/uc?id=0B6fULLT_NpxMa mpUWIBCQXVJZzA&export=download	[Accessed 01/07/2019].
13	Jonathon Taaffe	(2019)	Table 4. Gateway OS – REMnux VM Configuration		[Created 1st July 2019].
14	FireEye/Flare-FakeNet-NG	(2019)	FakeNet-NG - Next Generation Dynamic Network Analysis Tool	https://github.com/fireeye/flare-fakenet-ng	[Accessed 1st July 2019].
15	Jonathon Taaffe	(2019)	Table 5. Gateway OS – REMnux VM Configuration		[Created 1st July 2019].
16	FireEye/Flare-FakeNet-NG	(2019)	FakeNet-NG - Next Generation Dynamic Network Analysis Tool	https://github.com/fireeye/flare-fakenet-ng	[Accessed 1st July 2019].
17	Jonathon Taaffe	(2019)	Table 6. FakeNet-NG Pre-Requisites		[Created 1st July 2019].
18	FireEye/Flare-FakeNet-NG	(2019)	FakeNet-NG - Next Generation Dynamic Network Analysis Tool	https://github.com/fireeye/flare-fakenet-ng	[Accessed 1st July 2019].
19	Microsoft.com	(2019)	Download Virtual Machines	https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/	[Accessed 2nd July 2019].
20	Jonathon Taaffe	(2019)	Table 7. Windows XP, 7, 8.1 and 10 VM Client Configuration		[Created 2nd July 2019].
21	Jonathon Taaffe	(2019)	Table 8. Windows XP, 7, 8.1 and 10 Network Configuration		[Created 2nd July 2019].
22	Malware Types & Lab Setups File	(2019)	FakeNet-NG Slide 25	https://moodle.ncirl.ie/mod/resource/view.php?id=59717	[Accessed 2nd July 2019].
23	Jonathon Taaffe	(2019)	Table 9. Windows XP, 7, 8.1 and 10 OS Configuration		[Created 2nd July 2019].
24	Microsoft.com	(2019)	Internet Explorer security zones registry entries for advanced users	https://support.microsoft.com/en-us/help/182569/internet-explorer-security-zones-registry-entries-for-advanced-users	[Accessed 2nd July 2019].
25	Portal.Azure.com	(2019)	Windows Server 2016 Standard	http://dl.msdn.com/pr/en_windows_server_2016_updated_feb_2018_x64_dvd_11636692.iso	[Accessed 2nd July 2019].
26	REMnux 6.0 OVA Public	(2019)	remnux-6.0-ova-public.ova (2.0G)	https://docs.google.com/uc?id=0B6fULLT_NpxMa mpUWIBCQXVJZzA&export=download	[Accessed 2nd July 2019].
27	FireEye/Flare-FakeNet-NG	(2019)	FakeNet-NG - Next Generation Dynamic Network Analysis Tool	https://github.com/fireeye/flare-fakenet-ng	[Accessed 2nd July 2019].
28	Microsoft.com	(2019)	Download Virtual Machines	https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/	[Accessed 2nd July 2019].
29	Jonathon Taaffe	(2019)	Diagram 1. Lab Phase 1 Installation and Configuration		[Created 2nd July 2019].
30	Portal.Azure.com	(2019)	Windows Server 2016 Standard	http://dl.msdn.com/pr/en_windows_server_2016_updated_feb_2018_x64_dvd_11636692.iso	[Accessed 2nd July 2019].
31	Microsoft.com	(2019)	Download Virtual Machines	https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/	[Accessed 2nd July 2019].
32	Jonathon Taaffe	(2019)	Diagram 2. Lab Phase 2 File Transfer		[Created 2nd July 2019].
33	Malware Types & Lab Setups File	(2019)	Tools Slide 28	https://moodle.ncirl.ie/mod/resource/view.php?id=59717	[Accessed 2nd July 2019].
34	Jonathon Taaffe	(2019)	Table 10. Client Application File Transfer		[Created 2nd July 2019].
35	Jonathon Taaffe	(2019)	Table 11. Client Application Installation		[Created 2nd July 2019].

36	Malware Types & Lab Setups File	(2019)	Tools Slides 28, 29, 30, 31, 32	https://moodle.ncirl.ie/mod/resource/view.php?id=59717	[Accessed 2nd July 2019].
37	Portal.Azure.com	(2019)	Windows Server 2016 Standard	http://dl.msdn.com/pr/en_windows_server_2016_updated_feb_2018_x64_dvd_11636692.iso	[Accessed 2nd July 2019].
38	Jonathon Taaffe	(2019)	Diagram 3. Lab Phase 6 Dynamic Malware Analysis Configuration		[Created 2nd July 2019].
39	Sikorski, Michael; Honig, Andrew	(2012)	Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software	https://library.ncirl.ie/items/32216	[Accessed 20th May 2019].
40	Sam Bowne	(2016)	CNIT 126: Practical Malware Analysis	https://samsclass.info/126/126_S16.shtml#lecture	[Accessed 20th May 2019].
41	iNetSim.org	(2018)	INetSim: Internet Services Simulation Suite	https://www.inetsim.org/	[Accessed 20th May 2019].
42	Jonathon Taaffe	(2019)	Table 12. Kali KDE 2019.1 VM Configuration		[Created 20th May 2019].
43	Jonathon Taaffe	(2019)	Table 13. Windows XP and Windows Server 2008 VM Configuration		[Created 20th May 2019].
44	Jonathon Taaffe	(2019)	Table 14. Labs, Applications and Analysis Tools Installation		[Created 20th May 2019].
45	Jonathon Taaffe	(2019)	Diagram 4. Practical Malware Analysis Lab Configuration		[Created 20th May 2019].
46	Jonathon Taaffe	(2019)	Table 15. Practical Malware Analysis Exercises		[Created 20th May 2019].
47	PracticalMalware Analysis-Labs	(2017)	Binaries for the book Practical Malware Analysis	https://github.com/mikesiko/PracticalMalwareAnalysis-Labs	[Accessed 22nd May 2019].
48	Virus Total	(2019)	Virus Total	https://www.virustotal.com	[Accessed 22nd May 2019].
49	Radburn, Wayne J.	(2018)	PEview version 0.9.9 (.zip 31KB)	http://wjrdburn.com/software/	[Accessed 22nd May 2019].
50	Softpedia.com	(2019)	PEiD 0.95	https://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/PEiD-updated.shtml	[Accessed 22nd May 2019].
51	Softpedia.com	(2019)	BinText 3.03	https://www.softpedia.com/get/System/File-Management/BinText.shtml	[Accessed 22nd May 2019].
52	Dependency Walker 2.2	(2019)	Dependency Walker 2.2	http://www.dependencywalker.com/	[Accessed 22nd May 2019].
53	Radburn, Wayne J.	(2018)	PEview version 0.9.9 (.zip 31KB)	http://wjrdburn.com/software/	[Accessed 22nd May 2019].
54	Softpedia.com	(2019)	PEiD 0.95	https://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/PEiD-updated.shtml	[Accessed 22nd May 2019].
55	Softpedia.com	(2019)	BinText 3.03	https://www.softpedia.com/get/System/File-Management/BinText.shtml	[Accessed 22nd May 2019].
56	Dependency Walker 2.2	(2019)	Dependency Walker 2.2	http://www.dependencywalker.com/	[Accessed 22nd May 2019].
57	PracticalMalware Analysis-Labs	(2017)	Binaries for the book Practical Malware Analysis	https://github.com/mikesiko/PracticalMalwareAnalysis-Labs	[Accessed 22nd May 2019].
58	Virus Total	(2019)	Virus Total	https://www.virustotal.com	[Accessed 22nd May 2019].
59	Softpedia.com	(2019)	PEiD 0.95	https://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/PEiD-updated.shtml	[Accessed 22nd May 2019].
60	UPX Ultimate Packer for eXecutables	(2019)	UPX 3.95	https://github.com/upx/upx/releases/tag/v3.95	[Accessed 22nd May 2019].
61	Softpedia.com	(2019)	PEiD 0.95	https://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/PEiD-updated.shtml	[Accessed 22nd May 2019].
62	Dependency Walker 2.2	(2019)	Dependency Walker 2.2	http://www.dependencywalker.com/	[Accessed 22nd May 2019].
63	Microsoft.com	(2019)	Strings v2.53	https://docs.microsoft.com/en-gb/sysinternals/downloads/strings	[Accessed 22nd May 2019].
64	Radburn, Wayne J.	(2018)	PEview version 0.9.9 (.zip 31KB)	http://wjrdburn.com/software/	[Accessed 22nd May 2019].
65	Microsoft.com	(2019)	Strings v2.53	https://docs.microsoft.com/en-gb/sysinternals/downloads/strings	[Accessed 22nd May 2019].
66	INetSim.org	(2018)	INetSim: Internet Services Simulation Suite	https://www.inetsim.org/	[Accessed 20th May 2019].
67	Microsoft.com	(2019)	Process Explorer v16.26	https://docs.microsoft.com/en-gb/sysinternals/downloads/process-explorer	[Accessed 20th May 2019].
68	Microsoft.com	(2019)	Process Monitor v3.52	https://docs.microsoft.com/en-gb/sysinternals/downloads/procmon	[Accessed 20th May 2019].
69	Wireshark.org	(2019)	Wireshark-win32-2.0.0.exe	https://1.eu.dl.wireshark.org/win32/all-versions/	[Accessed 20th May 2019].
70	PracticalMalware Analysis-Labs	(2017)	Binaries for the book Practical Malware Analysis	https://github.com/mikesiko/PracticalMalwareAnalysis-Labs	[Accessed 22nd May 2019].
71	Hex-Rays	(2019)	IDA Freeware for Windows (48 MB)	https://www.hex-rays.com/products/ida/support/download_freeware.shtml	[Accessed 22nd May 2019].

72	OllyDBG	(2019)	Download OllyDbg 1.10 (final version)	http://www.ollydbg.de/download.htm	[Accessed 22nd May 2019].
73	Microsoft.com	(2019)	RegOpenKeyExA function	http://msdn.microsoft.com/en-us/library/windows/desktop/ms724897(v=vs.85).aspx	[Accessed 22nd May 2019].
74	PracticalMalwareAnalysis-Labs	(2017)	Binaries for the book Practical Malware Analysis	https://github.com/mikesiko/PracticalMalwareAnalysis-Labs	[Accessed 22nd May 2019].
75	Microsoft.com	(2019)	Strings v2.53	https://docs.microsoft.com/en-gb/sysinternals/downloads/strings	[Accessed 22nd May 2019].
76	Softpedia.com	(2019)	BinText 3.03	https://www.softpedia.com/get/System/File-Management/BinText.shtml	[Accessed 22nd May 2019].
77	Aldeid	(2019)	GINA-Graphical Identification and Authentication	https://www.aldeid.com/wiki/GINA-Graphical_Identification_and_Authentication	[Accessed 22nd May 2019].
78	Radburn, Wayne J.	(2018)	PEview version 0.9.9 (.zip 31KB)	http://wjradburn.com/software/	[Accessed 22nd May 2019].
79	Microsoft.com	(2019)	Process Monitor v3.52	https://docs.microsoft.com/en-gb/sysinternals/downloads/procmon	[Accessed 20th May 2019].
80	Resource Hacker	(2019)	Resource Hacker 5.1.7	http://www.angusj.com/resourcehacker/	[Accessed 20th May 2019].
81	SlavaSoft	(2019)	HashCalc 2.02	http://www.slavasoft.com/hashcalc/	[Accessed 20th May 2019].
82	PracticalMalwareAnalysis-Labs	(2017)	Binaries for the book Practical Malware Analysis	https://github.com/mikesiko/PracticalMalwareAnalysis-Labs	[Accessed 22nd May 2019].
83	Radburn, Wayne J.	(2018)	PEview version 0.9.9 (.zip 31KB)	http://wjradburn.com/software/	[Accessed 22nd May 2019].
84	Microsoft.com	(2019)	Strings v2.53	https://docs.microsoft.com/en-gb/sysinternals/downloads/strings	[Accessed 22nd May 2019].
85	Hex-Rays	(2019)	IDA Freeware for Windows (48 MB)	https://www.hex-rays.com/products/ida/support/download_freeware.shtml	[Accessed 22nd May 2019].
86	Microsoft.com	(2019)	Process Explorer v16.26	https://docs.microsoft.com/en-gb/sysinternals/downloads/process-explorer	[Accessed 20th May 2019].
87	PracticalMalwareAnalysis-Labs	(2017)	Binaries for the book Practical Malware Analysis	https://github.com/mikesiko/PracticalMalwareAnalysis-Labs	[Accessed 22nd May 2019].
88	Wireshark.org	(2019)	Wireshark-win32-2.0.0.exe	https://1.eu.dl.wireshark.org/win32/all-versions/	[Accessed 20th May 2019].
89	Microsoft.com	(2019)	Strings v2.53	https://docs.microsoft.com/en-gb/sysinternals/downloads/strings	[Accessed 22nd May 2019].
90	Hex-Rays	(2019)	IDA Freeware for Windows (48 MB)	https://www.hex-rays.com/products/ida/support/download_freeware.shtml	[Accessed 22nd May 2019].
91	Radburn, Wayne J.	(2018)	PEview version 0.9.9 (.zip 31KB)	http://wjradburn.com/software/	[Accessed 22nd May 2019].
92	WinHex.com	(2019)	WinHex: Computer Forensics & Data Recovery Software, Hex Editor & Disk Editor	http://winhex.com/winhex/	[Accessed 22nd May 2019].