|  |  |  |
| --- | --- | --- |
| Week | Reverse Engineering Malware | Duration |
| 13 | IDA Pro and OllYDBG Practical | 240 mins |

Marks allocation: 10/100 for CA practical submission

**Lesson Objectives**

The objective of this practical is to familiarise the students of how to use IDA Pro and OllyDBG.

# Lab 01

Analyse the malware found in the file Lab09-01.exe using OllyDbg and IDA Pro to answer the following questions.

Questions

1. How can you get this malware to install itself?

To install the malware, need 2 argument, first is “-in” and second is the password

2. What are the command-line options for this program? What is the password requirement?

The command-line option is “-in” to install, “-re” to uninstall, “-cc” to show current configuration and “-c” to modify configuration

3. How can you use OllyDbg to permanently patch this malware, so that it doesn’t require the special command-line password?

To patch command-line password, we need to patch 0x402510 which is where password is check. We need 0x402510 to return true every time no matter the password is correct or not. To do it, we edit the binary on 0x402510 to B8 01 00 00 00 C3.

4. What are the host-based indicators of this malware?

* HKLM\SOFTWARE\Microsoft \XPS\Configuration
* Lab09-01\_patched Manager Service
* C:\WINDOWS\system32\system32\Lab09-01\_patched.exe

5. What are the different actions this malware can be instructed to take via the network?

* Sleep (do nothing a period of time)
* Upload (read a file from the network than save to the system)
* Download (send a file from the system to the network)
* Execute command
* Do nothing

6. Are there any useful network-based signatures for this malware?

There is a network-based signatures which is practicalmalwareanalysis.com for this malware

# Lab 02

Analyse the malware found in the file Lab09-02.exe using OllyDbg to answer the following questions.

Questions

1. What strings do you see statically in the binary?

String cmd

2. What happens when you run this binary?

It close without doing anything

3. How can you get this sample to run its malicious payload?

Rename the malware file to ocl.exe

4. What is happening at 0x00401133?

A string is spilt into each character

5. What arguments are being passed to subroutine 0x00401089?

String “1qaz2wsx3edc” and a stack buffer pointer is passed

6. What domain name does this malware use?

http://www.practicalmalwareanalysis.com

7. What encoding routine is being used to obfuscate the domain name?

XOR is used to obfuscate the domain name

8. What is the significance of the CreateProcessA call at 0x0040106E?

At 0x40106E, StartupInfo’s hStdInput, hStdOutput, hStdError is being configured before CreateProcessA is called using cmd. It look like a reverse shell using StartupInfo structure which allow attacker to send command to victim system