# 1 – Introduction

## 1.1 Terms & conditions

This agreement applies to the malware creation course provided. In our tutorials, we only EVER hack our safe systems as a proof of concept and never engage in any illegal activity. The course was created for research purposes, and you should never apply the learning to harm any person or any system. The material is exclusively developed to improve the theoretical knowledge.

## 1.2 About

In malicious software teaching, the theoretic focus is usually presented, however, a practical guidance rarely occurs. Practice allows a better understanding of the attacker's mind, as well as a different point of view from the academic world. To do so, this laboratory produces an interactive platform, where a student generates his own malware in a safe environment. This includes propagation and concealment types of malware, different techniques of obfuscation, amongst others, all of them essential in malware studies.

Currently, the few online labs available start with an installation guide of a whole environment and the development of basic exercises, which sometimes entail problems in configuration, installation, etc. In addition, all students are limited to the same teaching pace and progress, and this fact generates boredom in advance students. The purpose of this research is avoiding all these problems by focusing on exercises to complement theory taught by the professor. Besides, students will be able to create their own malware and watch its performance, instead of merely picturing it.

## 1.3 Objectives

**General:**

To provide an overall understanding from both theoretical and practical perspectives for malicious software.

**Specific:**

- To offer the basic malware knowledge background for infoSec purposes.
- To analyse and use Metasploit framework.
- To create basic and advanced malware types.
- To test evasion and obfuscation techniques.
- To develop one's own payload for malware.

## 1.4 Architecture

This laboratory was created in Linux, explicitly in UBUNTU LTS which is a virtual machine containing the necessary tools for developing and testing malware in a safe environment. Besides, it is easy to distribute, manage and avoid misconfigurations or extra issues.

To create malware, the Metasploit framework is installed providing different modules for different needs. These last ones will be further studied in the upcoming units. In addition, the tutorial presents two vulnerable systems that can be exploited by the student with the malware created by himself. Furthermore, extra tools allow a student to gain supplementary skills for better malware development, including auxiliary methods for obfuscation and concealment.

Finally, the lab provides web tools. The first tool acts as a guide for the tutorial, and the second provides a visual interface where the new created malware is analysed by the most popular antiviruses. The result shows whether the malware will be detected or not.

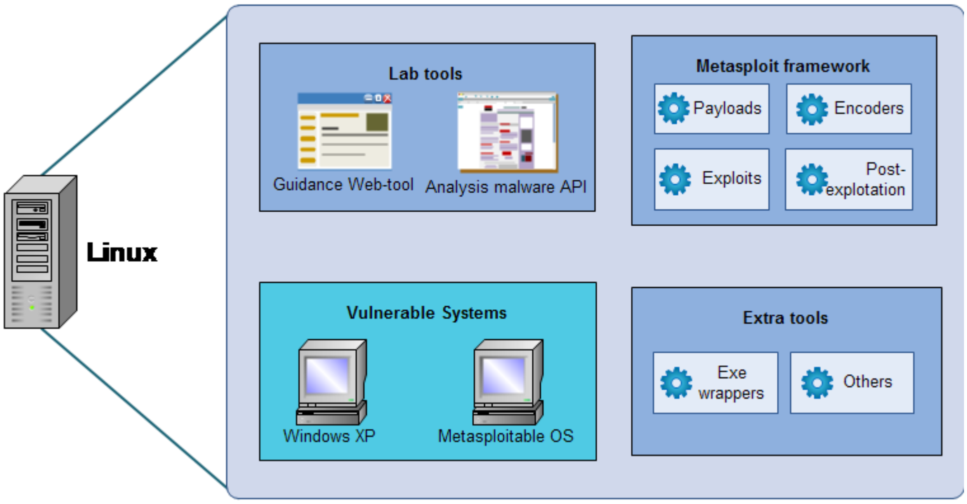

*Figure 1: Environment for the malware laboratory.*

## 1.5 Tools

In order to provide a wide scope, several tools are used in this laboratory. A brief explanation of them is presented below:

### Virtualization Management.

Virtualization manager is an environment that allows to create/handle/monitor/use different systems and services on only a physical machine. This is also known as 'hypervisor', nowadays available in several web providers such as VMware and VirtualBox. The laboratory VM (Virtual Machine) works in any hypervisor, while the vulnerable operating systems are installed in VirtualBox.

### Metasploit Framework.

This is the most widespread open-source platform for the penetration testing community. Metasploit is extremely useful for our purposes since it allows creating, exploiting, and encoding different types of vulnerabilities used for malware development. The forthcoming unit will deeply explain it.

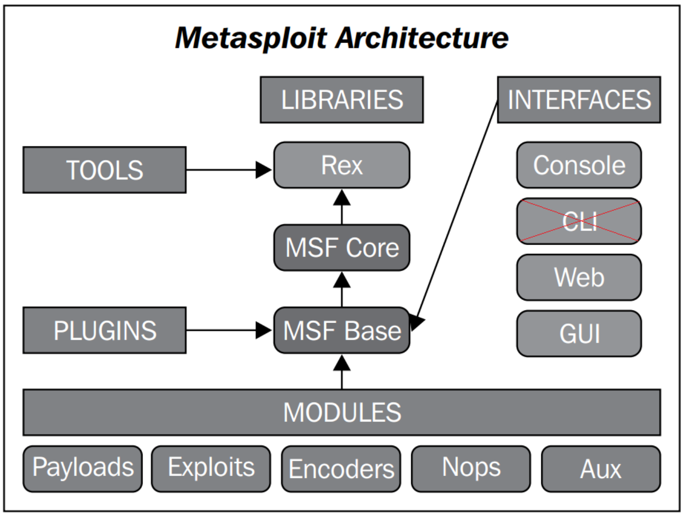

*Figure 2: Architecture for the Metasploit platform.*

**Windows XP (VM).**

Everyone recognises Windows as the most common operating system around the world. Hence, it is highly exposed to attacks by cyber criminals. The Windows security gap occurs for a concrete reason: Microsoft's software is designed for maximum ease-of-use, which means security risks are more likelihood (Scambray, 2007). Microsoft should spend more time for security targets, but this is not always possible. Although there has been a huge improvement on recent operating system versions, it does not imply that old systems are no longer used.

Therefore, Windows XP (service pack 1) is suitable for use in this tutorial, since it is easy to exploit several well-known vulnerabilities, besides, Microsoft support has ended years ago. A virtual machine is installed on VirtualBox, it contains a user called "UserXP" and no password. This will allow us to interact with some malware in future exercises.

**Metasploitable OS (VM).**

This operating system is a modified version of Ubuntu intentionally vulnerable, containing open services, backdoors, weak passwords, etc. These characteristics help us to exploit a system, in a similar way as a cyber-attack would do (Rapid 7 Community, 2012).

**Veil framework ordnance-payload.**

This is a special tool for evading antiviruses, working as a Metasploit compliment and using python. Veil framework allows creating payloads which will not be detected by an antivirus, not even the most popular ones such as ESET, Kaspersky, etc.

**VirusTotal.**

VirusTotal is a web service which scans files searching for malicious content. It provides an extensive report showing the antiviruses that detected or not the file. Besides, it is capable of showing details of the malware, in case it is detected.

## 1.6 Basic background knowledge

The following topics are suggested as basic background:

**Linux.** Students should have basic knowledge and skills on Linux.

Linux cheat sheet

An introduction to Linux and basics

Bind and reverse shell

The command “man”, the Linux manual.

**VirtualBox.** Students should know how to manage VMs, Startup, etc.

Manual for VirtualBox

How to Use VirtualBox: User’s Guide

# References

Christopher Truncer in Informational, T. V. (2017, March 21). *Veil 3.0 Command Line Usage*. Retrieved from Veil – Framework: https://www.veil-framework.com/veil-command-line-usage/

Davis, M. a. (2009). *Hacking Exposed Malware and Rootkits.* McGraw-Hill, Inc.

Fosnock, C. (2005). Computer worms: past, present, and future. *East Carolina University*, 8.

Goswami, D. (2017, 05 14). *Wanna Cry ransomware cyber attack: 104 countries hit, India among worst affected, US NSA attracts criticism.* Retrieved from India Today in.: http://indiatoday.intoday.in/story/wanna-cry-ransomware-attack-104-countries-hit-nsa-criticised/1/953338.html

Lee, J. (2017, May 23). *Metasploit-framework:How a payload works*. Retrieved from Rapid7 Community: https://github.com/rapid7/metasploit-framework/wiki/How-payloads-work

Maynor, D. (2011). *Metasploit toolkit for penetration testing, exploit development, and vulnerability research.* Elsevier.

Microsoft Security TechCenter. (2008, October 23). *Microsoft Security Bulletin MS08-067 - Critical*. Retrieved from https://technet.microsoft.com/en-us/library/security/ms08-067.aspx

Porras, P. A. (2009). A Foray into Conficker's Logic and Rendezvous Points. *LEET*.

Rapid 7 Community. (2012, 06 01). *Metasploitable 2 Exploitability Guide.* Retrieved from Metasploit Community: https://community.rapid7.com/docs/DOC-1875

Rapid 7 Community. (2013, 07 05). *How To Set Up A Penetration Testing Lab.* Retrieved from Rapid 7 Community: https://community.rapid7.com/docs/DOC-2196

Rapid 7 Community. (2016, September 14). *Metasploit-framework: msfvenom*. Retrieved from Metasploit-framework: https://github.com/rapid7/metasploit-framework/wiki/How-to-use-msfvenom

Rapid 7 Community. (2017, 06 06). *Matasploit User Guide.* Retrieved from Penetration testing software for offensive security teams.: https://community.rapid7.com/docs/DOC-1563

Scambray, J. a. (2007). *Hacking Exposed Windows.* Tata McGraw-Hill Education.

Singh, A. (2012). *Metasploit Penetration Testing Cookbook.* Packt Publishing Ltd.

Spafford, E. H. (1989). The Internet worm program: An analysis. *ACM SIGCOMM Computer Communication Review, 19*, 17--57.

The network support company. (2016, 10 06). *What Is Malware? [Infographic].* Retrieved from network-support: https://www.network-support.com/wp-content/uploads/2016/10/What-Is-Malware-Infographic.jpg