What is a Trojan?

A Trojan horse, or Trojan, is a type of malicious code or software that looks legitimate but can take control of user computer. A Trojan is designed to damage, disrupt, steal, or in general inflict some other harmful action on user data or network.

A Trojan acts like a bona fide application or file to trick user.

How do Trojans work?

Here’s a Trojan malware example to show how it works.

User might think he received an email from someone he know and click on what looks like a legitimate attachment. But user has been fooled. The email is from a cybercriminal, and the file user clicked on — and downloaded and opened — has gone on to install malware on your device.

When user execute the program, the malware can spread to other files and damage your computer.

How? It varies. Trojans are designed to do different things. But you’ll probably wish they weren’t doing any of them on your device.

Common types of Trojan malware

Here’s a look at some of the most common types of Trojan malware, including their names and what they do on your computer:

**Backdoor Trojan**

This Trojan can create a “backdoor” on your computer. It lets an attacker access your computer and control it. Your data can be downloaded by a third party and stolen. Or more malware can be uploaded to your device.

**Distributed Denial of Service (DDoS) attack Trojan**

This Trojan performs DDoS attacks. The idea is to take down a network by flooding it with traffic. That traffic comes from your infected computer and others.

**Downloader Trojan**

This Trojan targets your already-infected computer. It downloads and installs new versions of malicious programs. These can include Trojans and adware.

**Fake AV Trojan**

This Trojan behaves like antivirus software, but demands money from you to detect and remove threats, whether they’re real or fake.

**Game-thief Trojan**

The losers here may be online gamers. This Trojan seeks to steal their account information.

**Mailfinder Trojan**

This Trojan seeks to steal the email addresses you’ve accumulated on your device.

**Ransom Trojan**

This Trojan seeks a ransom to undo damage it has done to your computer. This can include blocking your data or impairing your computer’s performance.

**Remote Access Trojan**

This Trojan can give an attacker full control over your computer via a remote network connection. Its uses include stealing your information or spying on you.

**Rootkit Trojan**

A rootkit aims to hide or obscure an object on your infected computer. The idea? To extend the time a malicious program runs on your device.

**SMS Trojan**

This type of Trojan infects your mobile device and can send and intercept text messages. Texts to premium-rate numbers can drive up your phone costs.

**Trojan banker**

This Trojan takes aim at your financial accounts. It’s designed to steal your account information for all the things you do online. That includes banking, credit card, and bill pay data.

That’s just a sample. There are a lot more.

Client-side attacks

It is better to gain access to a target computer using the server-side attacks, like trying to find exploits in the installed applications, or in the operating system. If we are not able to find the exploit, or if our target is hidden behind an IP or using the hidden network, in this case, we will use client-side attacks. Client-side attacks require the user to do something, like download an image, open a link, install an update that will then run the code in their machine. The client-side attacks require user interaction that?s why information gathering is very important. It gathers the information about an individual?s applications and who they are as a person. To do client-side attack successfully, we need to know the friends of that person, what network and website they use, and what website they trust. In client-side attack, when we gather information, our focus is the person, rather than their applications or operating system.

The target machine will be a Window machine, and the attacking machine will be Kali machine. To ensure they are on the same network, both the machine will use NAT networks. In our example, we will be using reserve connections, so separate IP address are not essential in this case.

In this section, we are going to learn how a tool called Veil can be used to generate an undetectable backdoor. After this, we will also discuss payloads. Once we have a brief idea about the payloads, we will generate a backdoor through which we will implement client-side attacks on our system, and enabling us to listen to the connections. Finally, we will learn at how to implement backdoor in real time, as well as techniques we can use to protect our system fromm such attacks.

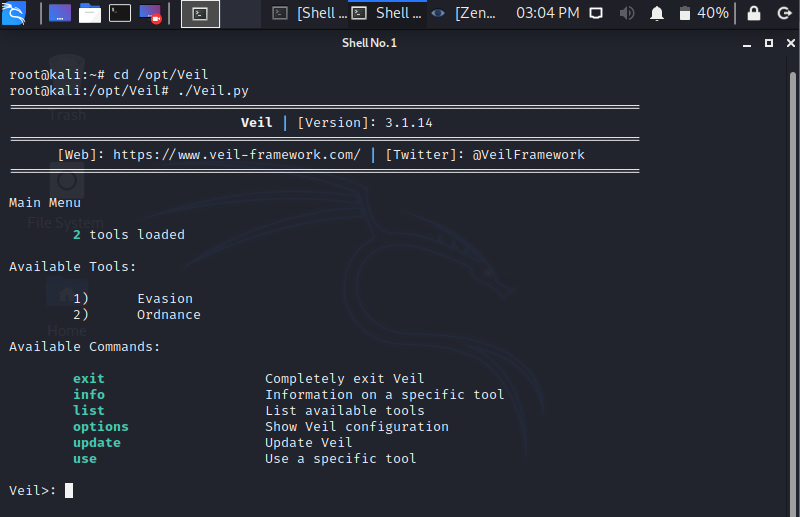
Overview of Payloads

Once Veil is installed, we are going to look at its commands.

Run Veil Using commands

# cd /opt/Veil

# ./Veil.py

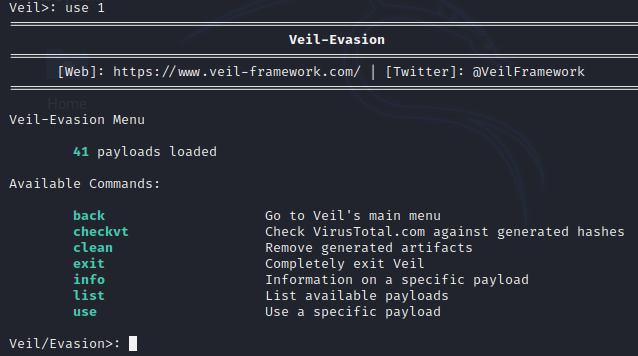


The **exit** allow us to exit the program, **info** is used to provide us the information about a specific tool, **list** is used to list the available tools, **update** is used to update Veil, **use** is used to enable the use of any tool.

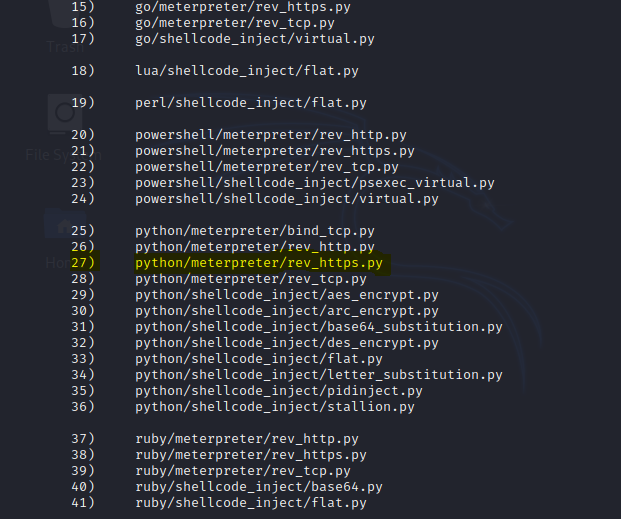
There are two types of tools that are used in the Veil

1. Evasion : This tool is used to generate an undetectable backdoor.
2. Ordnance : This tool is used to generate the payloads used by Evasion. This is more of a secondary tool.

The payload is a part of the code, that does what we want it to. In this case, it gives us a reverse connection, downloads and executes something on a target computer. Now we are using the **use** command to enable the use of any tool. We want to run Evasion, so we are going to run **use 1** command. When Veil-Evasion has loaded, we should see something similar to the following command:



Veil gives a **list** of commands that can run on this tool. In the following screenshot, There are each payload is divided into three parts, and highlighted payload we will be using which are **27) python/meterpreter/rev\_https.py**:



The first part of the payload's name is the programming language in which the payload will be wrapped. In the preceding screenshot, we can see the language used include CS, Python, GO, C, PowerShell, and Ruby. In this example, we are going to use the **go** language.

The second part of the payload is the type of payload. In other words, the type of code that is going to be executed on the target network. In this example, we are going to use **Meterpreter**, which is a payload designed by Metasploit. Metasploit is a huge framework, and sometimes it is used for hacking. Meterpreter runs in memory, so it is difficult to detect, and it does not leave a large footprint. With Meterpreter, we can gain full access over a target computer. It allows us to navigate through the filesystem, install or download files, and much more.

The third part of the payload's name is the method that's going to be used to establish its connection. In our example, that is **rev\_https**. Where **rev** stands for reverse, and **https** is the protocol that will be used to establish the connection. In the preceding screenshot, there are a few examples of **rev\_tcp**, which creates a reverse TCP connection.

A reverse connection is where the target machine connects to the attacker's machine via a backdoor. This method bypass antivirus programs, because the connection is not directed at the target computer, but rather at the attacker instead. In our example, we are going to use a port 80 or 8080, that many websites use, so that the connection will appear as a harmless website connection.

Now set LHOST, LPORT and generate payload:

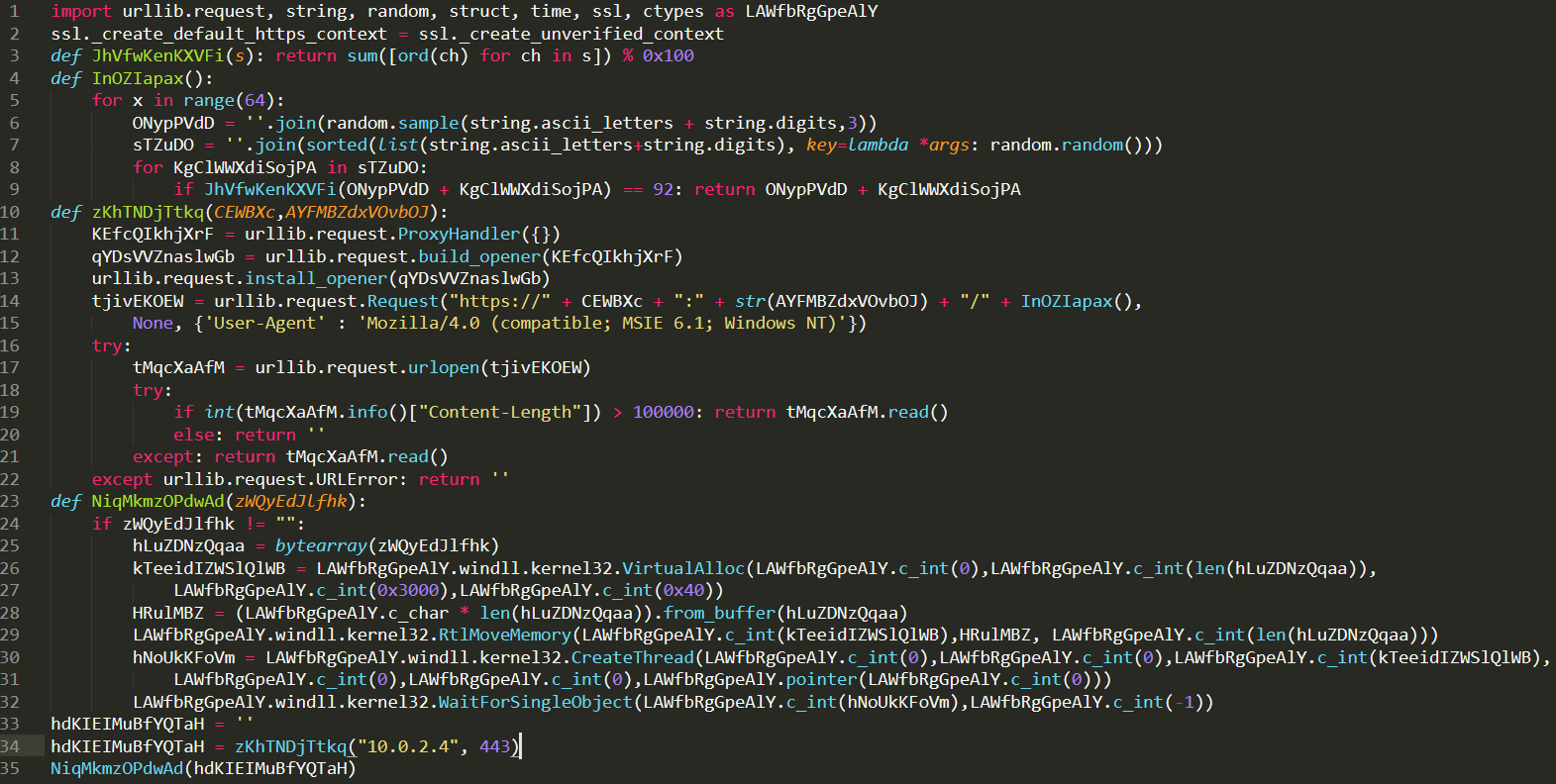


Set file name:



Generated Executable written to: /var/lib/veil/output/compiled/trojanfile.exe

Source Code of executable file:



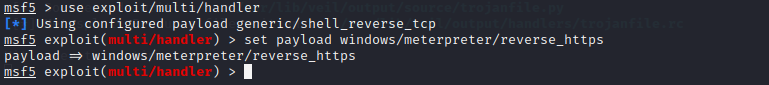
**Listening for connection:**

The backdoor which created uses a reverse payload. To work the reverse payload, a port need to open in our Kali machine so that the target machine can connect to it. When created the backdoor, port set to 447, so port 447 needed to open on Kali machine. In this example, the name of chosen payload is **windows/meterpreter/rev\_https**.

To listen for an incoming connection Metasploit Framework used. To run msfconsole use command:

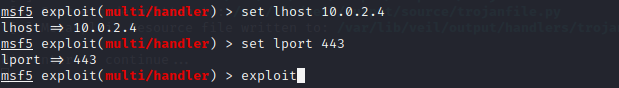


To listen incoming connections a module in Metasploit Framework used. Use following command to launch that module and set PAYLOAD:



Now set the LHOST and LPORT to the IP address of our Kali machine using the following command:

And then execute exploit command:



Server started for listening connections. When user execute .exe file in target Windows Machine, Kali Machine will show connection form target computer.

**Application/Utility:**

Malware Detection, as the name suggests, helps to protect a computer system from malicious and other harmful programs. A computer trojan is a computer program that can cause damage to a computer's software, hardware or data. It is referred to as a trojan because it has the capability to create backdoor in user computer.

**Future Scope :**

There are many ideas and methods that are yet to be implemented for the future research.

Malware analysis is like a cat & mouse game, as new malware analysis techniques are developed, malware authors respond with new techniques to thwart analysis. Antivirus use source code to detect malware i.e. too time consuming, our model is statistical & thus more successful.

In future, a tool or software developed can include the significant finding obtained in our Random Forest Model to classify the malware.

Moreover the tools may be attached with the knowledge base, so that less skilled user can also use the toolkit for forensic analysis.

Last but not least the task of first detecting, analyzing & generating cures for unknown & malicious files is itself an individual research topic.