Ain Shams University
Faculty of Computer & Information Sciences
Computer Systems Department.

Practical Ethical Hacking and Web Security

By:

Seif Eldeen Sameh Mahrous [Csys]
Shreif Mokhtar Eid [Csys]
Kareem Mohamed Anwer [Csys]
Fady Samir Thabet [Csys]
Amr Fathy Moussa [Csys]

Under Supervision of:

[Mervat Al-Qutt]
[Assistant Professor],
Computer Systems Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

[Omar Al Nakib]
[Teaching Assistant],
Computer Systems Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

Acknowledgement

We would like to thank Mervat El Qott for pushing us to be best selves, helping us in completing this project and supporting us throughout the whole year.

We would like to express our gratitude for everyone who helped us during the whole years of study starting with endless thanks for our families in encouraging us to do a great job, our friends who are always support us , our DR’s, and TA's for teaching us across 4 years of education in our college (Faculty of Computer and Information Science).

Abstract

C2 Framework refers to command and control framework , c2 framework helps pentesters and red teamers in their engagement on the targeted system to make the compromised system more secure by finding vulnerabilities in the system and report it to them .

Recent years , the cyber security fields have witnessed the evolution of antivirus and EDRs that made attacking on the system more difficult but with evolution of the antivirus and EDRs, the attacks also evolved and that include c2 frameworks , c2 framework provide everything the attacker need . The problem is , it's too expensive to buy for the small or medium companies , even the cheap ones or the free ones they have problems in making the connection stable between the target and the attacker ,so as long the connection isn't stable the persistent on the targeted machine will be merely impossible, also it gets detected by antiviruses and EDRs, so it doesn't provide the stealthiness that the attacker need .

This control and command framework overcome on all the previous problems that we mentioned earlier as it provide stability in communications and provide to pentesters and red teamers all the capabilities they need in order to perform their engagements.

Command and control framework is the set of tools and techniques that attackers use to maintain communication with compromised devices following initial exploitation. The specific mechanisms vary greatly between attacks, but C2 generally consists of one or more covert communication channels between devices in a victim organization and a platform that the attacker controls .These communication channels are used to issue instructions to the compromised devices, download additional malicious payloads, and pipe stolen data back to the adversary.

Contents

1. **Introduction**

1.1 Motivation

Cyber-attacks against businesses are often deliberate and motivated by financial gain. However, other motivations may include making a social or political point - e.g., through hacktivism. espionage - e.g., spying on competitors for unfair advantage.

Nowadays all our lives depend on technology starting from gaming and contacting others to houses and cars. This dependency force us to implement better applications to secure and that by two methods

The first method is by creating detecting and preventing tools and that is found in antivirus , firewalls , IDS , IPS …etc. and these tools help us in detecting who enter the system and information about him and send them to the system administrator and the defensive team which is called the blue team e.g. soc tier , soc analysis . and they can prevent the attacker from doing it in the system and even block his Ip address . the problem is that some of these tools can be bypassed, and the attacker will gain access to the information and that is where the second method take place .

The second method is by creating an attacking tools and assigning ethical hackers called penetration testers to try to attack the system and by passing the blue team and the detection tools and by that they are acting as hackers with even the same tools but the difference is that these penetration testers report how they entered the system end escalated their privileges so the that it can be solved and preventing the real hackers from getting access to the system and thus corrupt it or even worse getting the users sensitive data and reusing them.

So, by these two methods we can try to prevent some attacks ,but because every day new attacking methods and vectors arise as the technology increase , we need to improve our solutions and update our tools to cope with the new attacks .

1.2 Problem Definition

The main problem that we found that affect the attacking in the system is what to do after the exploitation . the stage that can change the level of the attack from low to severe impact on the system. Is post exploitation stage by using command and control (C2)framework start its job by running different commands on the victim machine to increase the privilege and to be able to download or upload data from/to the system and do other post exploitation command to give us the full control of the system.

when we searched online for other command and control frameworks on different aspects either it is free or commercial and their availability and what they do, and languages used there was some problems with them.

The first problem is that almost all of them are commercial and not even cheap they are about 3.5k $ per year and that for small or medium business is very expensive.

Second problem is that the free versions are very weak that can be detected by antiviruses and cannot bypass most of the detection tools.

Third problem is that some of the free versions and cheap versions have bad connection between the attacker and the agent that was implanted on the system which led to losing the connection easily from the system.

And the last problem we will introduce is that the persistency is almost not found in most of the cheap command and control frameworks .

So, after seeing all these problems assigned to the found command and control framework, we decided to implement our command-and-control framework that can overcome all those problems.

1.3 Objective

This documentation reveals an approach which is implemented as a framework that can help red teamers and pentesters in their attacking scenario on the targeted system as it provides all the capabilities they need to perform a successful attack on the targeted system.

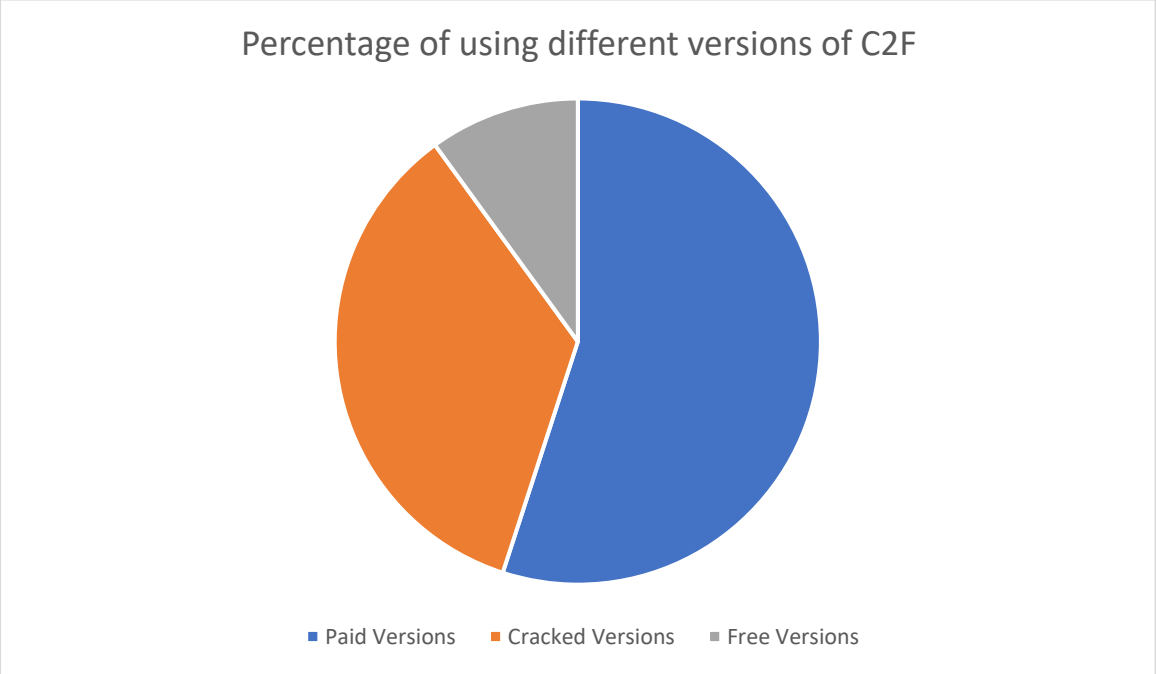
This control and command framework overcome on all the previous problems that we mentioned in the previous section as it provides stability in communication and persistency on the targeted system.

2. Survey

After taking the survey of 200 engineers in the cybersecurity field in different department e.g. Red Team, Penetration Testing , Soc analyst , blue teams we found that 55% of them use paid command and control framework even if they cannot afford it.

Although it is illegal, we found that 35% use cracked versions of the paid command and control framework to be able to get if for free.

Almost 10% use the free versions while knowing they are not the best solution



3-Background

This chapter will introduce the following:

- A detailed description about the field of this project
  - Cyber security in general
  - Network Security
  - Application Security
  - Information Security
  - Operational Security

- Types of teams in cyber security
  - Red Team
  - Blue Team
  - Purple Team

- Penetesting Stages
  - Planning and reconnaissance
  - Scanning
  - Gaining access
  - Maintaining access
  - Analysis and Reporting

- Post-Exploitation Stages
  - Acquiring situation awareness
  - Privilege escalation
  - Maintaining access
  - Cracking the hashes
  - Disable the firewall / Bypassing Antivirus

- Similar C2 Frameworks
  - Cobalt Strike / Covenant

3.1 A description about the field of this project

- **Cyber security**:
  is the practice of defending computers, servers, mobile devices, electronic systems, networks, and data from malicious attacks. It's also known as information technology security or electronic information security. The term applies in a variety of contexts, from business to mobile computing, and can be divided into a few common categories.

- **Network security** :
  is the practice of securing a computer network from intruders, whether targeted attackers or opportunistic malware and in our project we mainly focus on network security.

- **Application security**:
  focuses on keeping software and devices free of threats. A compromised application could provide access to the data its designed to protect. Successful security begins in the design stage, well before a program or device is deployed.

- **Information security**:
  protects the integrity and privacy of data, both in storage and in transit.

- **Operational security:**
  includes the processes and decisions for handling and protecting data assets. The permissions users have when accessing a network and the procedures that determine how and where data may be stored or shared all fall under this umbrella.

3.2 Teams in Cybersecurity

❖ Red Team:

The red team attacks and attempts to break the blue team's defenses. Ideally, these ethical hackers are unaware of an enterprise's defense mechanisms, so their services are often outsourced to a third party.

Red teams use real-world cyber-attack techniques to exploit weaknesses in a company's people, processes, and technologies. They circumvent defense mechanisms, aiming to infiltrate corporate networks and simulate data exfiltration -- all without being noticed by the blue team.

Common red team techniques include:

- watering hole attacks and drive-by downloads that target specific users and their PC using an internet browser
- Phishing, social engineering and other forms of credential theft mechanisms
- Vulnerability scanning.

In addition to these common hacker techniques, red team members use custom-made tools to get into networks, and then often escalate privileges to successfully breach the company.

Because exercises are performed to improve security, red team members write up reports post-attack, including details about techniques used, vectors targeted, and successful and unsuccessful attempts. The reports should also include recommendations about how to strengthen the organization's security posture, ensure defenses are up to par and bolster systems from future threats. The reports help blue teams understand where gaps in coverage exist, how defenses failed and where security needs to be tightened.

❖ Blue Team:

The blue team is responsible for regularly analyzing enterprise systems to properly protect them, identify vulnerabilities, and evaluate the effectiveness of security tools and policies.

Blue team tasks include:

- Monitoring corporate networks, systems, and devices
- Detecting, mitigating, containing and eradicating threats and attacks
- Collecting network traffic and forensic data
- Performing data analysis
- Conducting internal and/or external vulnerability scans, DNS audits and risk assessments.

Blue teams analyze information from these tasks and then update security software, hardware and policies to better protect against potential future attacks.

Blue team members also:

- Create, configure and enforce firewall rules
- Set and implement device and user access controls, often using the principle of least privilege
- Keep enterprise software -- production and security -- patched and up to date
- Deploy IDS/IPS and/or endpoint detection and response systems

Blue teams will notify senior management when risks are found to assess if a risk should be accepted or if a new policy/control needs to be adopted to mitigate it.

Like red teams, after an exercise is completed, blue teams gather evidence, logs and data to write reports about their experiences and findings, as well as develop a list of actions to be taken.

❖ Purple Team:

Calling the purple team a "team" is a bit misleading. The purple team is, in fact, not a standalone team but a mix of blue and red team members.

While red and blue teams have the same goal of improving the security of an organization, too often both are unwilling to share their "secrets." Red teams sometimes will not disclose methods used to infiltrate systems, while blue teams won't say how red team attacks were detected and defended against.

However, sharing these secrets is critical to strengthening the company's security posture. The value of red and blue teams is nil if they don't share their research and reporting data. This is where the purple team steps in. Purple team members get their red and blue teammates to work together and share insights about their resources, reporting and knowledge. To do so, purple teams should focus on fostering communication and collaboration between the red and blue teams.

3.3 Penetration Testing Stages

❖ Planning and Reconnaissance

- Defining the scope and goals of a test, including the systems to be addressed and the testing methods to be used.
- Gathering intelligence (e.g., network and domain names, mail server) to better understand how a target works and its potential vulnerabilities.

❖ Scanning

- **Static analysis** : Inspecting an application's code to estimate the way it behaves while running. These tools can scan the entirety of the code in a single pass.
- **Dynamic analysis** : Inspecting an application's code in a running state. This is a more practical way of scanning, as it provides a real-time view into an application's performance.

❖ Gaining Access

This stage testers try to find vulnerabilities in the target then try to exploit these vulnerabilities, typically by escalating privileges, stealing data, intercepting traffic, etc., to understand the damage they can cause.

❖ Maintaining Access

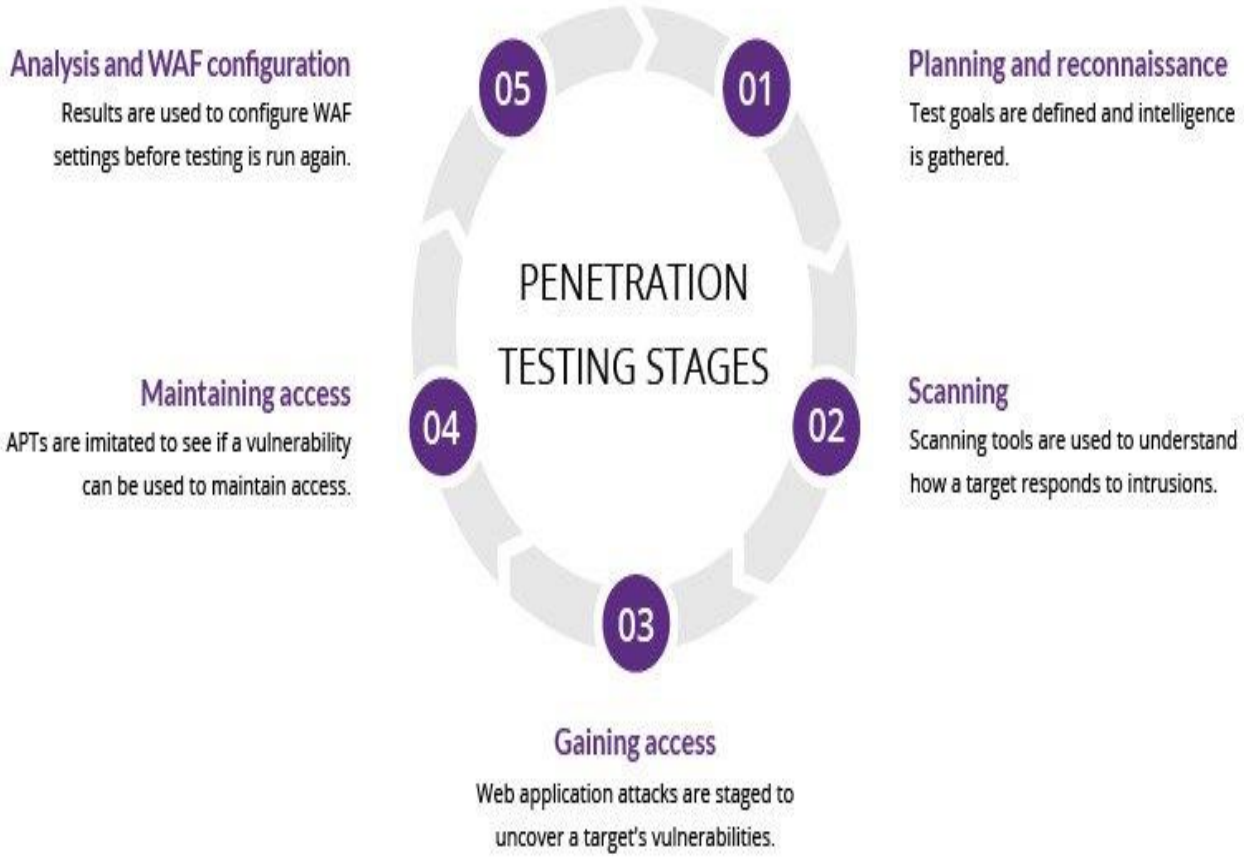
The goal of this stage is to see if the vulnerability can be used to achieve a persistent presence in the exploited system— long enough for a bad actor to gain in-depth access. The idea is to imitate advanced persistent threats, which often remain in a system for months in order to steal an organization's most sensitive data.

❖ Analysis and Reporting

The results of the penetration test are then compiled into a report detailing:

- Specific vulnerabilities that were exploited
- Sensitive data that was accessed
- The amount of time the pen tester was able to remain in the system undetected

This information is analyzed by security personnel to help configure an enterprise's WAF settings and other application security solutions to patch vulnerabilities and protect against future attacks.



3.4 Post-Exploitation Stages

❖ Acquiring Situation Awareness:-

Immediately when compromising the host system, you wish to achieve data regarding the host which is located on the inner network, which might contain a hostname, interfaces, routes, and services of our host. If you are familiar with the host operating system you can take advantage of this and enumerate more information about the host system and network. Windows would be one of our common targets since it's the foremost used OS within the corporate surroundings. Since most of your acquainted with Windows, it might be simple to enumerate it. Our main goals would be to enumerate the network, determine whether other hosts are reachable or not from our compromised host, the interfaces, and also the services. Because this is also important to check the security of the network along with the host if the network is not secure then it may give unauthorized access to the attacker within our network.

❖ Escalating Privileges:

In this stage, we should attempt to escalate the privileges to gain full access to the host machine. Now we are in the security process. there are many ways to escalate privileges.

❖ Maintaining Access:

Currently, we've managed to step up our privileges to either the administrator level or SYSTEM level. So far, we've managed to keep up stability, however, we have a tendency to haven't managed to ascertain tenacity. Whenever the target machine reboots, the method on that we've connected our agent is closed, and that we would lose access. therefore one may raise, why not access the system by mistreatment the vulnerability we have to antecedently exploit.

❖ Cracking the Hashes to Access to Alternative Services:

The second approach we'd remark is getting the hashes then cracking them to get access to alternative services like remote desktop, VNC, or telnet. This approach isn't really sneaky because the administrator might notice the changes you create. Considering that several users area unit allowed access thereto explicit service.

❖ Disable Firewall :

It is very important to disable firewalls protections, to perform the various tasks on the host OS. The reason we wish to disable the firewall is that we don't want it to interrupt the connection. whereas we have to perform our post-exploitation method. From our agent, we'd issue the "shell" command to launch the Windows electronic communication. From the Windows electronic communication, we have to issue the subsequent command to show off the firewall.

❖ Bypass the Antivirus :

The reason to disable the antivirus is that we have to don't want it to identify/delete our backdoor. we would like to stay unseen whereas conducting our penetration take a look at. we will check for the put-in antivirus by typewriting the "net start" command and "tasklist/svc" from the electronic communication to ascertain the method the antivirus is running.

Now is possible to use the "taskkill" command to kill a selected method or let the agent modify it for us. In our agent, we will realize a script named "killav" which will mechanically kill all the processes related to Associate in Nursing antivirus. Let's read the contents of the script by mistreatment the "cat" command followed by the trail of the script.

3.5 Similar C2 Frameworks

❖ Cobalt Strike :

gives you a post-exploitation agent and covert channels to emulate a quiet long-term embedded actor in your customer's network. Malleable C2 lets you change your network indicators to look like different malware each time. These tools complement Cobalt Strike's solid social engineering process, its robust collaboration capability, and unique reports designed to aid blue team training.

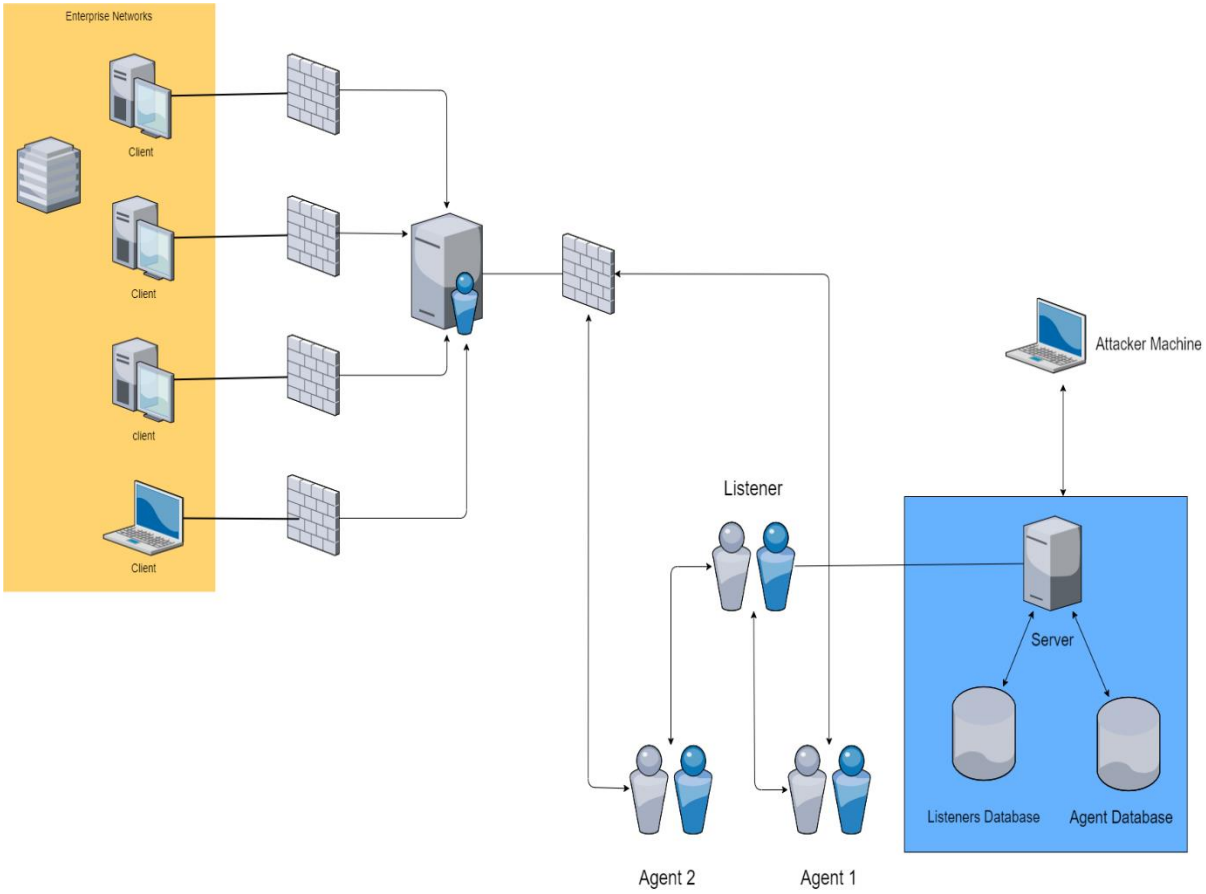
❖ Covenant:

is a .NET command and control framework that aims to highlight the attack surface of .NET, make the use of offensive .NET tradecraft easier, and serve as a collaborative command and control platform for red-teamers. What sets this apart from other C2 Post-Exploitation Frameworks is that it supports .NET Core – which is multi-platform. Hence, Covenant can run natively on Linux, MacOS, and Windows platforms! Additionally, Covenant has docker support, allowing it to run within a container on any system that has docker installed. It consists of three components – Covenant (server-side component), Elite (client-side component) and Grunt (implant).

4- Analysis and Design

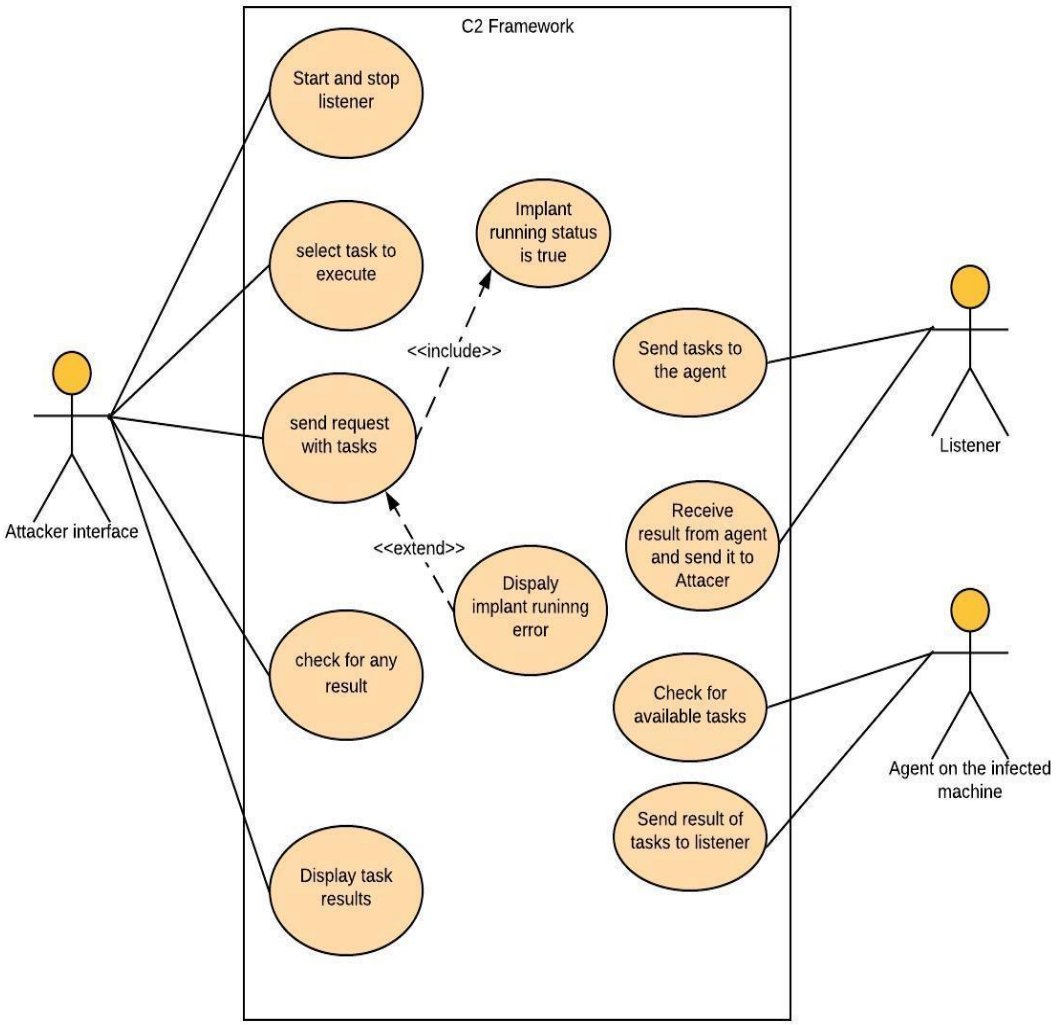
4.1 System Overview

4.1.1 System Architecture



4.2 System Analysis & Design

4.2.1 Use Case Diagram

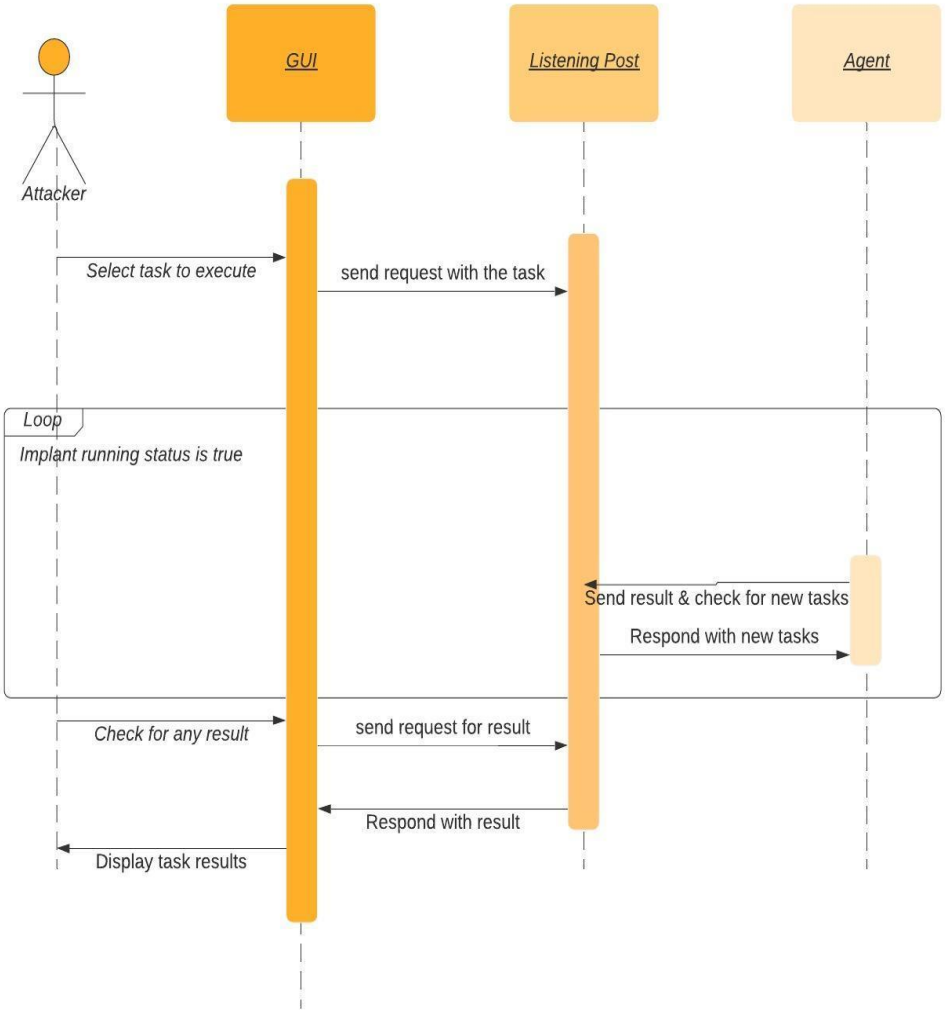


Attacker interface	
Start and Stop listener	Responsible for start & stop the listener between the user and agent.
Select tasks to execute	Attacker selects tasks that want to execute on the targeted machine by sending it to the listener and the listener will send it to agent to execute the tasks.
Send request with tasks	The attacker will send the requested task to the listener and the listener will pass it to the agent to execute it.
Check for any result	Check for any results for the available tasks that the agent is currently executing.
Display task results	Display the tasks result for the attacker.
Implant running status is true	This function in responsible for checking the status of the agent if it was running or not
Display implant running status error	If there is an error This function is responsible for displaying it.

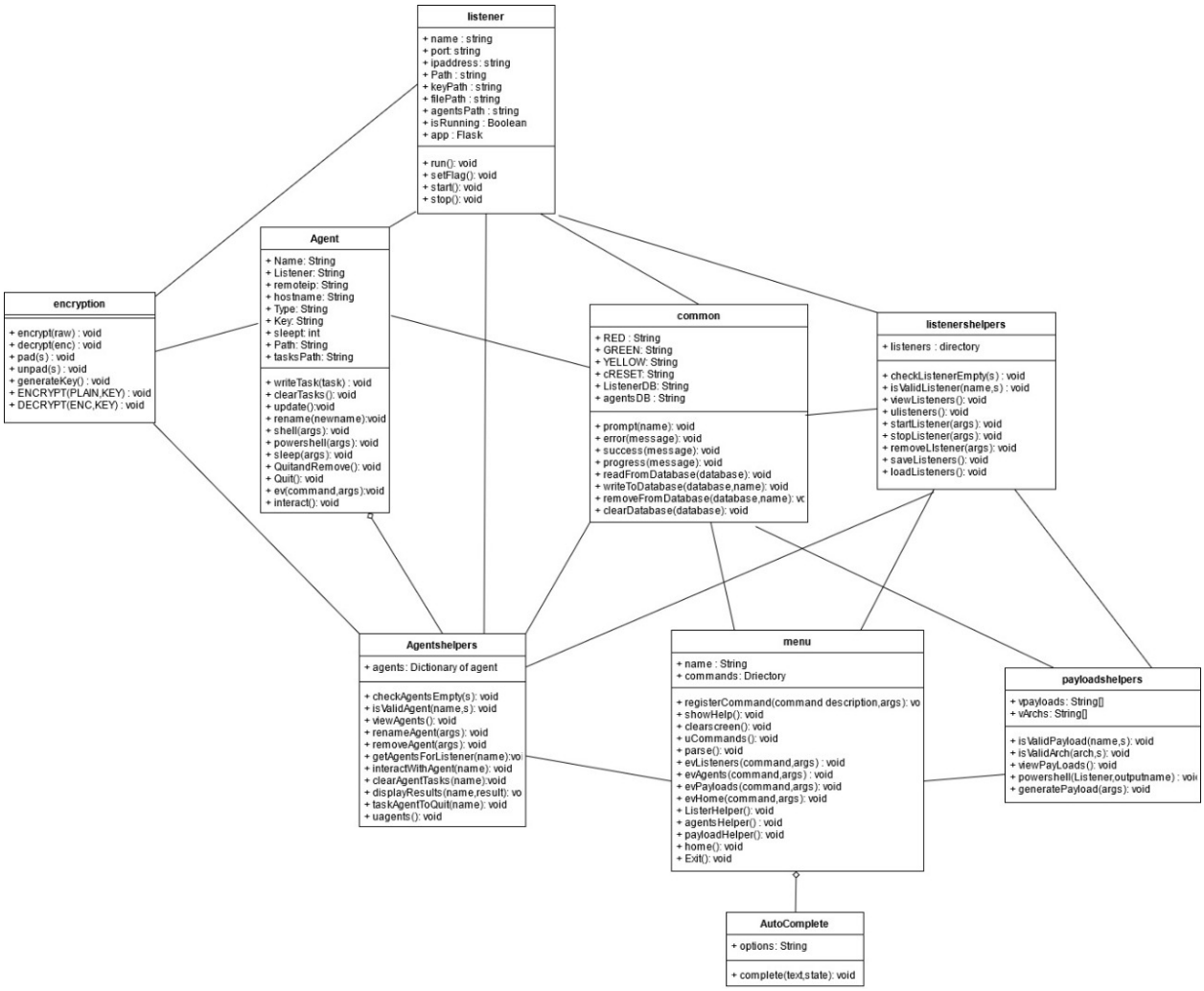
Listener	
Send tasks to the agent	This is responsible for receiving the tasks from the attacker and send it to agent.
Receive tasks result from agent & send it to the attacker	This is responsible for receiving the tasks results from the agent and send it back to the attacker.

Agent	
Check for available tasks	This function is responsible for keep checking if there's available incoming tasks.
Send result of tasks to listener	This function sends the result of the tasks to the listener

4.2.2 Sequence Diagram



4.2.3 Class Diagram



5-Implementation and Testing

This chapter will present :

- Used Software and Hardware
  - Visual Studio Code
  - Linux / Windows Virtual Machines (VMware)
  - Kaspersky Total Security
  - TrendMicro Antivirus

- Programming Languages and frameworks
  - Flask
  - PowerShell
  - Python
  - Go Language

- A detailed description of all techniques and algorithm implemented :
  - Flask Routing
  - Advanced Encryption Standard (AES)
  - PowerShell Invoke-WebRequest

- A detailed description of all the system functions:
  - Server Functions
  - Listeners Functions
  - Agent Functions

- Testing Command and Control Framework functions

5.1 Software and Hardware Tools

5.1.1 Software Tools:

- ### Visual Studio

  Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and MacOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

- ### Linux / Windows Virtual Machines

  In computing, a virtual machine is the virtualization / emulation of a computer system. Virtual machines are based on computer architectures and provide functionality of a physical computer. Their implementations may involve specialized hardware, software, or a combination.

  Used for:

  - Run an operating system in an app window on your desktop that behaves like a full, separate computer.
  - Run software your main operating system can't, and try out apps in a safe, sandboxed environment.

- ### Kaspersky Total Security / TrendMicro Antivirus

  Software used to prevent, scan, detect and delete malicious code from our computer. Once installed most anti-virus run automatically in the background to provide real-time protection against malicious code attacks.

5.1.2 Hardware Tools:

- No hardware tools were used in this project (Command and Control Framework).

5.2 Programming Languages and Frameworks

- **Flask**

  Micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

- **PowerShell**

  Task automation and configuration management framework from Microsoft, Consisting of a command-line shell and the associated scripting language.

- **Python**

  An interpreted high-level general-purpose programming language python's design philosophy emphasizes code readability with its not able to use significant identification.

5.3 Techniques and Algorithms used:

Flask Routing

- Routing is the process of mapping specific URL with the associated function that is intended to perform some tasks. It is usually used to access some particular page in web application.

- We use routing in flask as decorator to bind a function to a URL, Also we can make parts of the URL dynamic and attach multiple rules to a function.

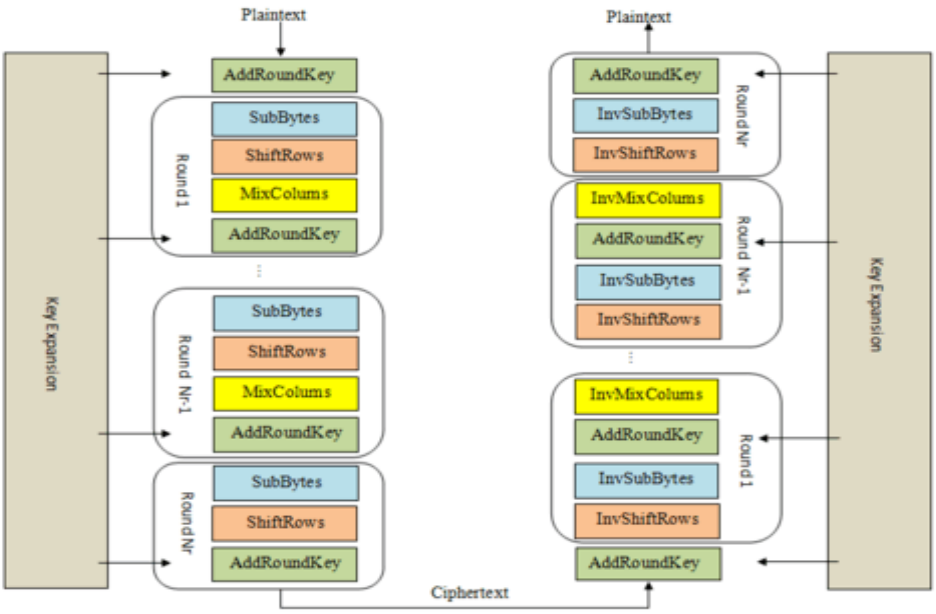
- By using routing in flask, you can use web request methods. You can use only GET method only, but we can use the other methods by using method argument `method=[method_type]` and also by using get it include HEAD method by default

- We used variable sections to help us to organize the code by dividing it sections and make it independent.

  - You can add variable sections to a URI by marking sections with `<variable_name>`. Your function then receives the `<variable_name>` as a keyword argument.

- In Variable section we can use a converter to specify the type of the argument like `<converter: variable_name>`.

Advanced Encryption Standard (AES) :



❖ **Encryption Process:**

- **Initial Round**: XORs the round key to the plaintext
- **9-Rounds:**

  SubBytes: Replace each byte with another byte according to an S-box.

  Shiftrows: Shifts the *i*th row of *i* positions, for *i* ranging from 0 to 3 to the left

  MixColumns: Applies linear transformation to each of the four columns of the state.

  AddRoundKey: XORs the round key to the output matrix of the previous operation (MixCloumns).

- **Final Round:**
  - In the final round we will use the same operations in the same order without using MixColumns function.

❖ Decryption Process:

- **Initial Round:** XORs the round key to the ciphertext

- **For 9-Rounds:**

  InvShift Rows: it's the inverse procedure of **(ShiftRows)**, As it shifts the *i*th row of *i* positions, for *i* ranging from 0 to 3 to the right.

  InvSubBytes: it's the inverse procedure of **(SubBytes)**, Replace each byte with another byte according to an S-box that different from the S-box used in the encryption process.

  AddRoundKey: XORs the round key to the output matrix of the previous operation (InvSubBytes).

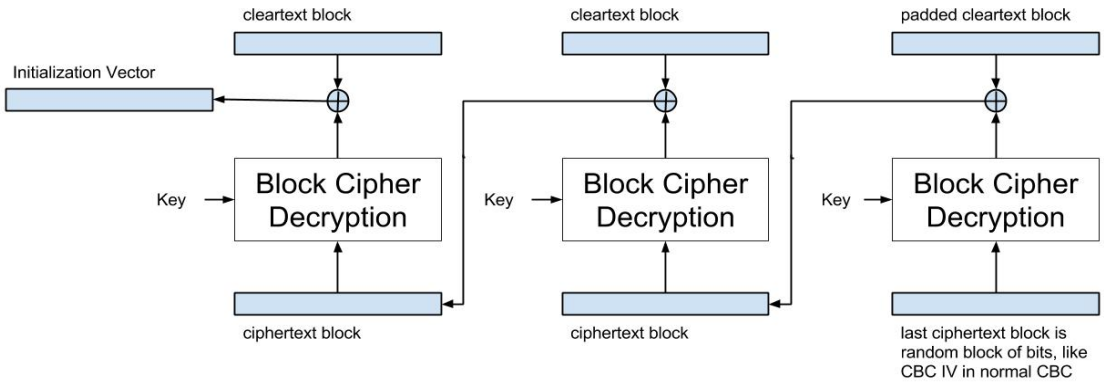
  InvMixColumns: it's the inverse procedure of **(MixColumns)**, Applies linear transformation to each of the four columns of the state.

- **Final Round:**

  In the final round we will use the same operations in the same order without using InvMixColumns function.

❖ Cipher Block chaining Mode (CBC)

- CBC is an advanced form of block cipher encryption. With CBC mode encryption, each ciphertext block is dependent on all plaintext blocks processed up to that point.



CBC **en**cryption using block cipher **de**cryption

PowerShell Invoke-WebRequest

- The `Invoke-WebRequest` cmdlet sends HTTP and HTTPS requests to a web page or web service. It parses the response and returns collections of links, images, and other significant HTML elements.

- Used for:

  - Send a web request.
  - Use a stateful web service.
  - Write the response content to a file using the encoding defined in the requested page.
  - Submit a multipart/form data file.
  - Simplified multipart/form data submission.
  - Catch nonsuccess messages from Invoke-WebRequest

- Notes:

  Because of changes in .NET Core 3.1, PowerShell 7.0 and higher use the HTTPClient.DefaultProxy Property to determine the proxy configuration.

  The value of this property is determined by your platform:

  - **For Windows**: Reads proxy configuration from environment variables. If those variables are not defined the property is derived from the user's proxy settings.

  - **For macOS**: Reads proxy configuration from environment variables. If those variables are not defined the property is derived from the system's proxy settings.

  - **For Linux**: Reads proxy configuration from environment variables. If those variables are not defined the property initializes a non-configured instance that bypasses all addresses.

5.4 System Functions :

- ❖ Server Functions :
  - Start Listeners :

    The function check if the listener is running or not and check for the interface address and port number , if all these conditions are true then the listener will be started else there will be an exception to show the cause of not starting the listener.

  - Stop Listeners :

    The function check if the listeners is already running or not if it's running it will stop the listeners else it will show a message that the listener is already stopped.

  - Payload Generator :
    - PowerShell : The `PowerShell` function takes a listener name, and an output name. It grabs the needed options from the listener then it replaces the needed strings in the `PowerShell` template and saves the new file in two places, `/tmp/` and the files path for the listener. After doing that it generates a download cradle that requests `/sc/`

  - Handle agents and task them through using listeners

❖ Listener Functions :

- Agent Handler:

  An Agent object is instantiated with a name, a Listener name , a remote address , a hostname , a type and encryption key , then it defines the sleep time which is 3 seconds by default , it need to keep track of the sleep time to be able to determine if the agent is dead or not when removing an agent otherwise it will keep for the agent to call forever.

- Host Files:

  After initiating the Agent, it creates the needed directories and files , Agent host these directories and files locally on the attacker machine

❖ Agent Functions :

- Download and execute its tasks:

  When an agent is executed on a system, first thing it does is get the hostname of the system then send the registration request to the server /reg . After receiving the response which contains its name it starts an infinite loop in which it keeps checking if there are any new tasks , if there are new tasks it executes them and sends the results back to the server. After each loop it sleeps for specified amount of time.

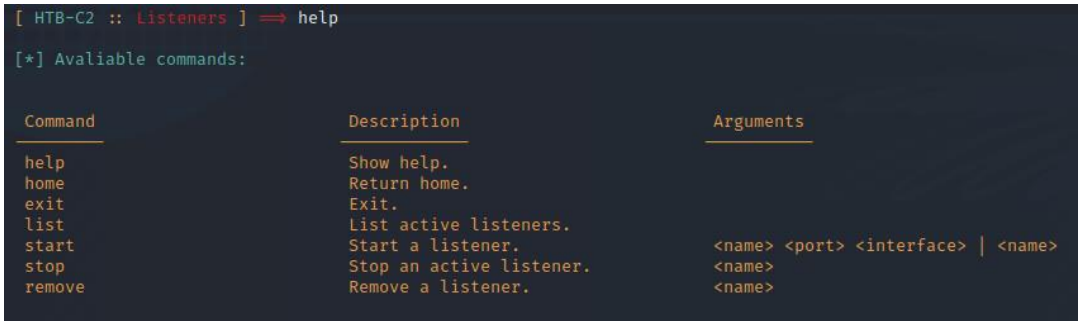
- Agent Persist :

  We used pickle to serialize the agents and save them into the database , when the attacker exits the server it saves all the agents objects , then when you start it again it loads all those objects so do not lose your agents

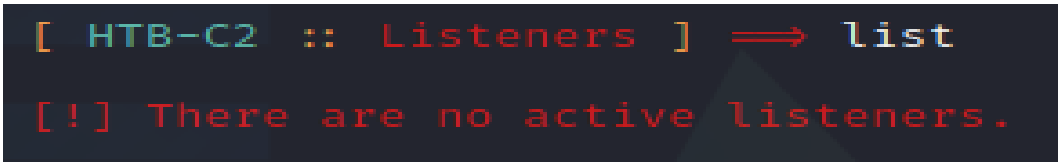
5.5 Testing :

❖ Listener Functions Testing :

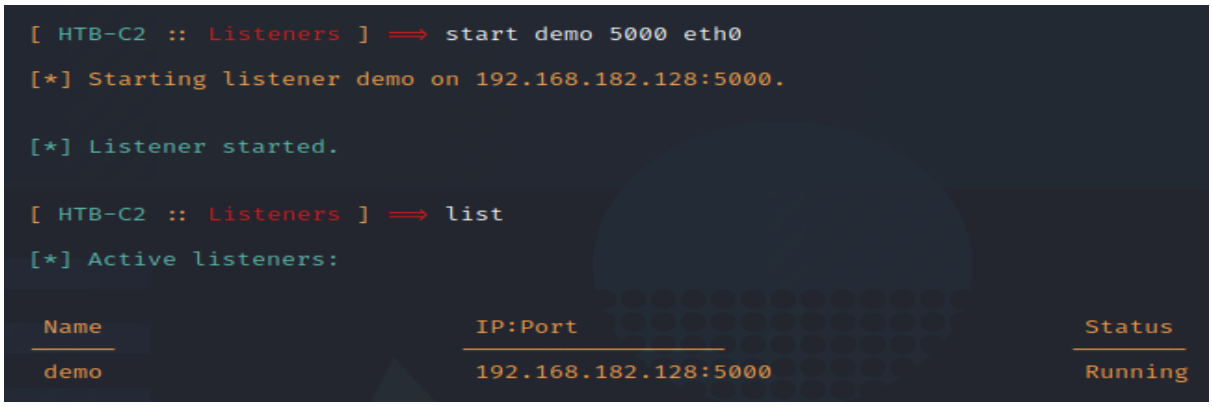
- **Help command** : will show all the available command that the attacker can use.



- **List command :** will list all the available listeners if exist else it will show message that's no active listener exist



- **Start command :** will start the listener at specific port and Ip address

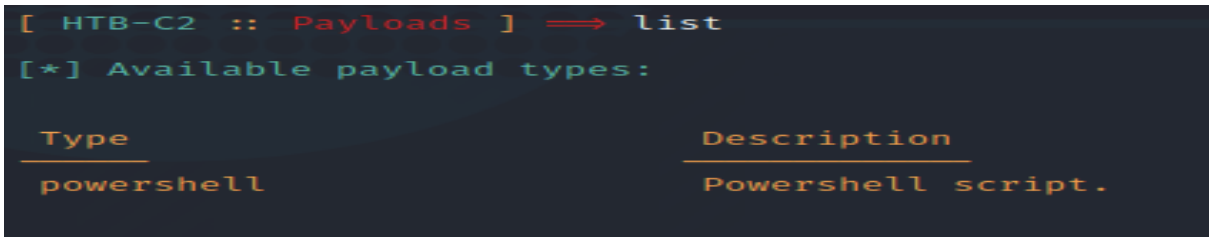


- **Stop command** : will stop the active listener with the given name

- **Remove command :** will remove the listener from the database even if it's running or not .

❖ Payload Generator Functions Testing :

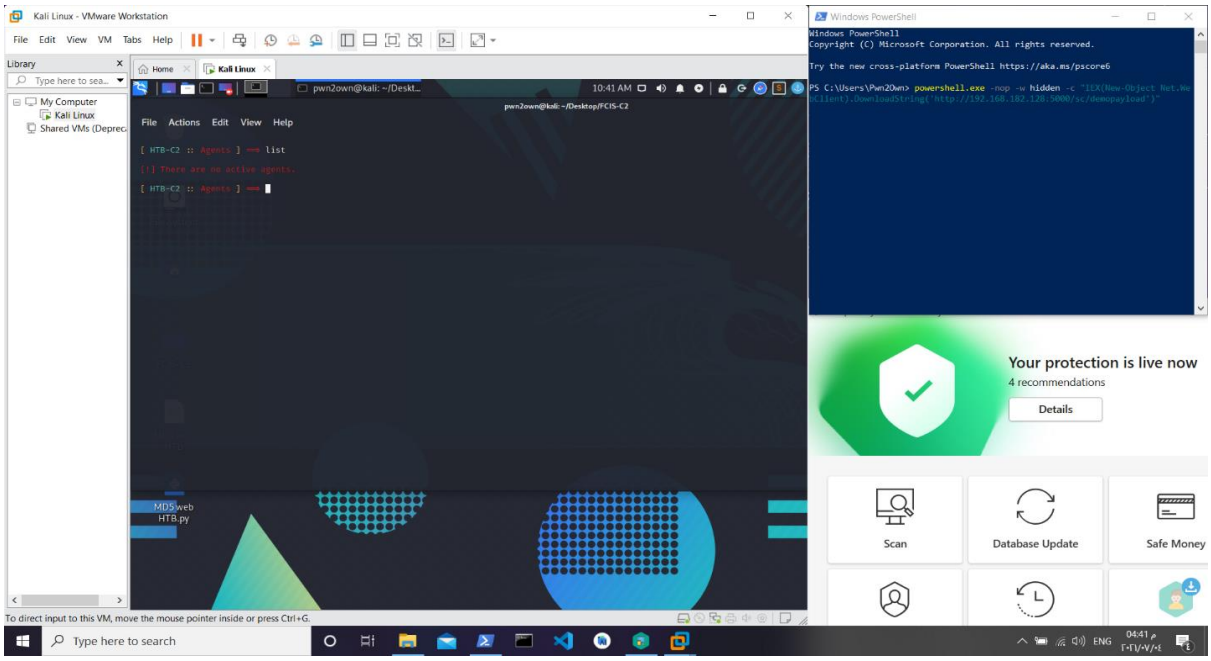
- **List command :** list all the available types of payloads .



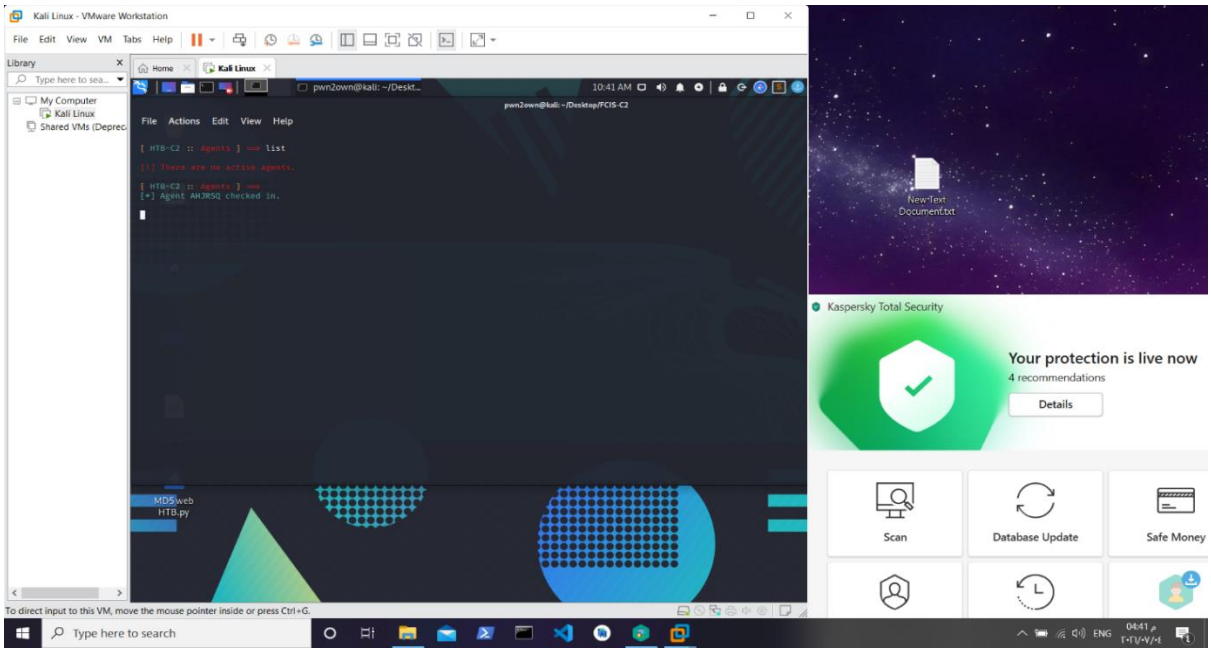
- **Generate command :** will generate the payload automatically based on the attacker input

❖ Agents Functions Testing :

- Before executing the generated payload



- After executing the generated payload



- **List command :** list all the available agents

```
[*] Active Agents:

 Name    Listener    External IP     Hostname
 ─────   ─────────   ─────────────   ──────────────
 AHJRSQ  demo        192.168.182.1   DESKTOP-IBRO2RO
```

- **Rename command :** rename the agent

```
[ HTB-C2 :: Agents ] ⟹ list

[*] Active Agents:

 Name    Listener    External IP     Hostname
 ─────   ─────────   ─────────────   ──────────────
 HODMUJ  demo        192.168.182.1   DESKTOP-IBRO2RO

[ HTB-C2 :: Agents ] ⟹ rename HODMUJ demoagent

[*] Waiting for agent.

[ HTB-C2 :: Agents ] ⟹
[*] Agent demoagent completed task.

[ HTB-C2 :: Agents ] ⟹ list

[*] Active Agents:

 Name      Listener    External IP     Hostname
 ─────     ─────────   ─────────────   ──────────────
 demoagent demo        192.168.182.1   DESKTOP-IBRO2RO
```

- **Interact command :** interact with specific agent
  - **Before interacting:**

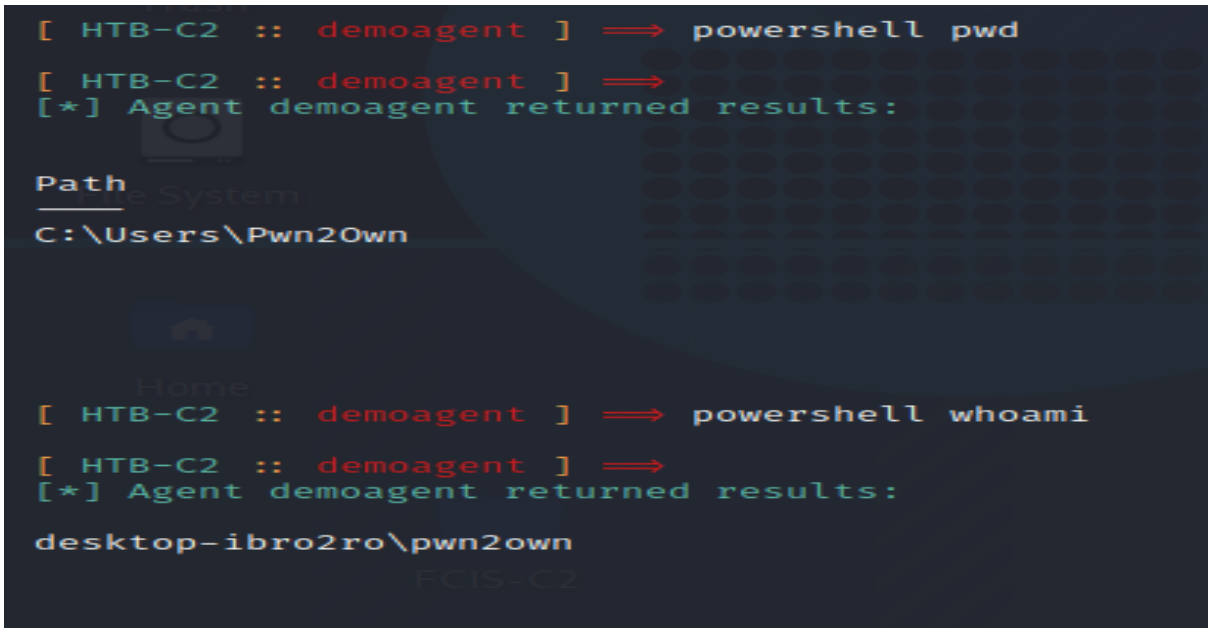
```
[ HTB-C2 :: Agents ] ⟹ interact demoagent
```

  - **After interact command execution:**

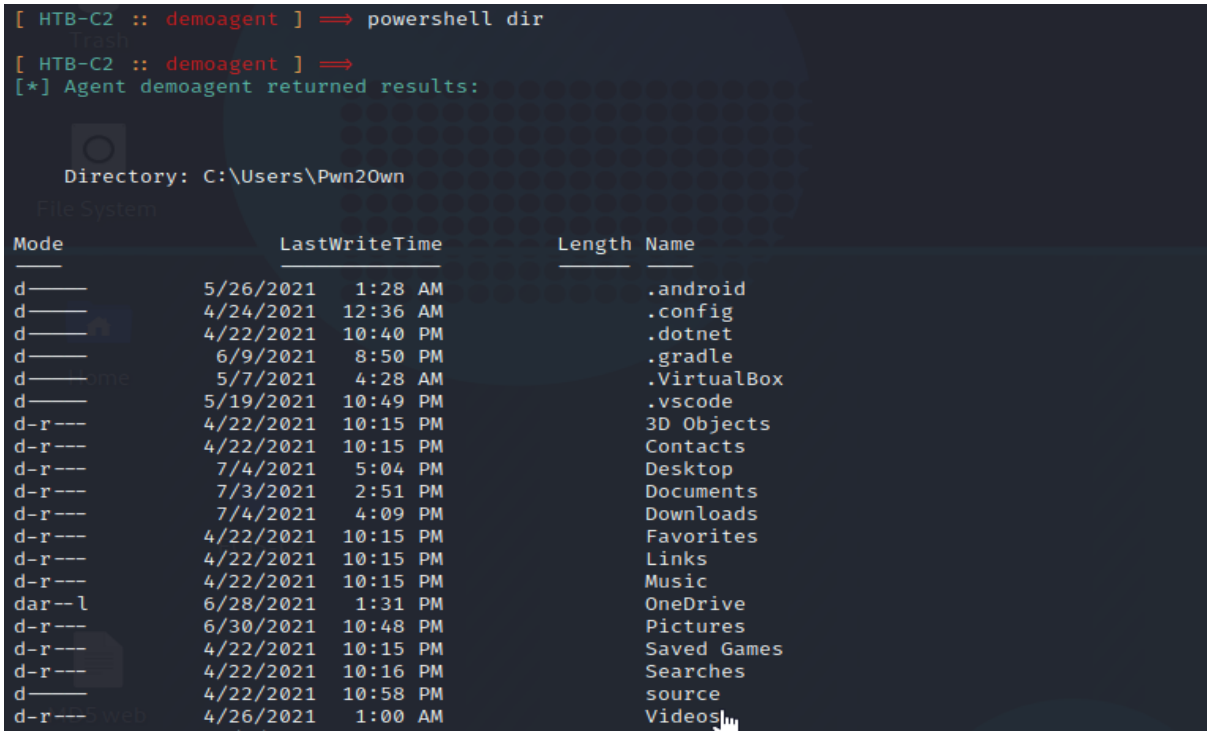
```
[ HTB-C2 :: demoagent ] ⟹ help

[*] Avaliable commands:

 Command     Description                      Arguments
 ─────────   ──────────────────────────────   ──────────────
 help        Show help.
 home        Return home.
 exit        Exit.
 shell       Execute a shell command.         <command>
 powershell  Execute a powershell command.    <command>
 sleep       Change agent's sleep time.       <time (s)>
 clear       Clear tasks.
 quit        Task agent to quit.
```

❖ **Execute commands on the target by using the agent**

- **Pwd command :** Present working directory
- **Whoami command :** it displays the current username of the target machine



Dir command : list all the files in specific directory



6-Conclusion and Future Work

6.1 Conclusion:

A complete summary of the whole project along with the results obtained.

Command and Control Infrastructure, also known as C2 or C&C, is the set of tools and techniques that attackers use to maintain communication with compromised devices following initial exploitation. The specific mechanisms vary greatly between attacks, but C2 generally consists of one or more covert communication channels between devices in a victim organization and a platform that the attacker controls. These communication channels are used to issue instructions to the compromised devices, download additional malicious payloads, and pipe stolen data back to the adversary.

A common strategy is to blend in with other types of legitimate traffic that may be in use at the target organization, such as HTTP/HTTPS or DNS. Attackers may take other actions to disguise their C2 callbacks, such as using encryption or unusual types of data encoding and in our C2 framework we use HTTP and encryption to encrypt all the traffic so that it be harder to get spotted by the antivirus.

Our C2 consists of 3 main components

- Server
- Listeners
- Agents

Let's summary the project by discussing the design for a basic C2 setup. First, we will want to have a server that will publish our tasks and receive the results of those tasks , Next, we want to have a program that will run on a target computer and make contact with our server periodically to find out what tasks to perform, execute those tasks, then respond back with the results (also known as a "Listener"). Lastly, we will want to have a client where an operator can easily create, manage, and submit tasks. Tasks could include things like returning information about the computer/network the implant is running on, executing OS commands, enumerating processes/threads, injecting into another process, establishing persistence, or stealing credentials for lateral movement.

6.2 Future Work:

- Use Key-Exchange in the encryption process of communication .
- Make the communication based HTTP2/DNS bacons
- Transfer data in more secure way like transferring it by embedding the data inside images (Steganography)
- Customized C# payloads that encrypt strings to bypass static detection
- Enhance Wizard module and add more SIEM , AV and sandbox detection along with more important enumeration data.
- Add more customizations and ideas for phishing using C# payloads and macros
- Integration with new exchange RCE
- Create a wipe command to securely remove files on hard disk without being detected and analyzed by blue team.
- integration with new exchange RCE
- Add obfuscation for the agents
- Integrate with cobalt strike payloads

References

- Automation of Cyber Penetration Testing
- Command & Control (Understanding, Denying and Detecting):
- PowerShell Documentation
- Python Documentation
- Serious Cryptography Book
- Commercial C2 Framework (Cobalt Strike)
- Imperva Penetration Testing Stages