

پروژه شبکه

مصطفی دریس پور

بخش کد:

پروژه شبکه سؤال اول:

به دلیل اینکه فایروال موجود در کلاینت این اجازه را به سرور نمی‌دهد تا پورتهای بر روی کلاینت باز کند. پس کلاینت نمی‌تواند پورتهای بفرستد تا سرور به آن وصل شود فایروال جلوی آن را میگیرد. بحث اضافه دیگر این است که شرکت های isp از یک ای پی ثابت برای چندین کاربر استفاده میکنند به همین دلیل سرور نمی‌تواند پورتهای را بر روی کلاینت باز کند.

سؤال دوم:

به این حمله *path traversal attack* می گویند.

سؤال اول:

7	2.821759...	127.0.0.1	127.0.0.1	TCP	66 12345 → 46134 [ACK] Seq=293 Ack=5 Win=65536 L...
8	2.821976...	127.0.0.1	127.0.0.1	TCP	358 12345 → 46134 [PSH, ACK] Seq=293 Ack=5 Win=65...
9	2.822012...	127.0.0.1	127.0.0.1	TCP	66 46134 → 12345 [ACK] Seq=5 Ack=585 Win=65280 L...

در تصویر بالا Three Way Handshake کپچر شده است.

سؤال دوم:

(الف)

tcp محدودیت ارسال 64kbyte دارد. و بالای آن را به قسمت‌های کوچکتر تقسیم می‌کند و می‌فرستد.
(ب)

6	0.001247...	127.0.0.1	127.0.0.1	TCP	74 58768 → 4984 [SYN] Seq=0 Win=65495 Len=0 MSS=...
7	0.001301...	127.0.0.1	127.0.0.1	TCP	74 4984 → 58768 [SYN, ACK] Seq=0 Ack=1 Win=65483...
8	0.001351...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=1 Win=65536 Len=...
9	0.001904...	127.0.0.1	127.0.0.1	TCP	20... 4984 → 58768 [PSH, ACK] Seq=1 Ack=1 Win=65536...
10	0.001945...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=2025 Win=64000 L...
11	0.002008...	127.0.0.1	127.0.0.1	TCP	20... 4984 → 58768 [PSH, ACK] Seq=2025 Ack=1 Win=65...
12	0.002022...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=4049 Win=62976 L...
13	0.002070...	127.0.0.1	127.0.0.1	TCP	20... 4984 → 58768 [PSH, ACK] Seq=4049 Ack=1 Win=65...
14	0.002083...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=6073 Win=61952 L...
15	0.002122...	127.0.0.1	127.0.0.1	TCP	20... 4984 → 58768 [PSH, ACK] Seq=6073 Ack=1 Win=65...
16	0.002138...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=8097 Win=60928 L...
17	0.002193...	127.0.0.1	127.0.0.1	TCP	20... 4984 → 58768 [PSH, ACK] Seq=8097 Ack=1 Win=65...
18	0.002210...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=10121 Win=59904 ...
19	0.002247...	127.0.0.1	127.0.0.1	TCP	20... 4984 → 58768 [PSH, ACK] Seq=10121 Ack=1 Win=6...
20	0.002264...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=12145 Win=58880 ...
21	0.002322...	127.0.0.1	127.0.0.1	TCP	20... 4984 → 58768 [PSH, ACK] Seq=12145 Ack=1 Win=6...
22	0.002341...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=14169 Win=57856 ...
23	0.002379...	127.0.0.1	127.0.0.1	TCP	20... 4984 → 58768 [PSH, ACK] Seq=14169 Ack=1 Win=6...
24	0.002394...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=16193 Win=56832 ...
25	0.002446...	127.0.0.1	127.0.0.1	TCP	20... 4984 → 58768 [PSH, ACK] Seq=16193 Ack=1 Win=6...
26	0.002462...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=18217 Win=55808 ...
27	0.002500...	127.0.0.1	127.0.0.1	TCP	20... 4984 → 58768 [PSH, ACK] Seq=18217 Ack=1 Win=6...
28	0.002516...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=20241 Win=54784 ...
29	0.002571...	127.0.0.1	127.0.0.1	TCP	20... 4984 → 58768 [PSH, ACK] Seq=20241 Ack=1 Win=6...
30	0.002590...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=22265 Win=53888 ...
31	0.002632...	127.0.0.1	127.0.0.1	TCP	20... 4984 → 58768 [PSH, ACK] Seq=22265 Ack=1 Win=6...
32	0.002649...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=24289 Win=52864 ...
33	0.002704...	127.0.0.1	127.0.0.1	TCP	20... 4984 → 58768 [PSH, ACK] Seq=24289 Ack=1 Win=6...
34	0.002723...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=26313 Win=51840 ...
35	0.002765...	127.0.0.1	127.0.0.1	TCP	20... 4984 → 58768 [PSH, ACK] Seq=26313 Ack=1 Win=6...
36	0.002781...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=28337 Win=50816 ...
37	0.002834...	127.0.0.1	127.0.0.1	TCP	20... 4984 → 58768 [PSH, ACK] Seq=28337 Ack=1 Win=6...
38	0.002852...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=30361 Win=49792 ...
39	0.002893...	127.0.0.1	127.0.0.1	TCP	20... 4984 → 58768 [PSH, ACK] Seq=30361 Ack=1 Win=6...
40	0.002908...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=32385 Win=48768 ...
41	0.002985...	127.0.0.1	127.0.0.1	TCP	20... 4984 → 58768 [PSH, ACK] Seq=32385 Ack=1 Win=6...
42	0.003257...	127.0.0.1	127.0.0.1	TCP	32... 4984 → 58768 [ACK] Seq=34409 Ack=1 Win=65536 ...
43	0.003361...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=67177 Win=48512 ...
44	0.003414...	127.0.0.1	127.0.0.1	TCP	77... 4984 → 58768 [PSH, ACK] Seq=67177 Ack=1 Win=6...
45	0.003607...	127.0.0.1	127.0.0.1	TCP	32... 4984 → 58768 [ACK] Seq=74889 Ack=1 Win=65536 ...
46	0.003678...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [ACK] Seq=1 Ack=107657 Win=8064 ...

شروع فرایند دانلود از پورت رندم به صورت بالا شروع می‌شود
18 بسته ارسال شده است.

120	0.024510...	127.0.0.1	127.0.0.1	TCP	30... 4984 → 58768 [FIN, PSH, ACK] Seq=1254537 Ack=...
121	0.026293...	127.0.0.1	127.0.0.1	TCP	66 58768 → 4984 [FIN, ACK] Seq=1 Ack=1285091 Win=...
122	0.026312...	127.0.0.1	127.0.0.1	TCP	66 4984 → 58768 [ACK] Seq=1285091 Ack=2 Win=6553...

و به این صورت ارتباط آن‌ها قطع می‌شود. (چون none-persistent است).

بخش :ngrok

[illegible]

در تصویر بالا در wireshark قسمت loopback در سرور این پکت ها را دریافت می کنیم و پکت ها نشان داده برای مثال در تصویر بالا پیام دستور PWD از سمت کلاینت به سرور فرستاده شده است.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port == 14386

No.	Time	Source	Destination	Protocol	Length	Info
96	3.488123...	18.192.93.86	172.27.118.106	TCP	66	14386 → 45960 [ACK] Seq=1 Ack=1 Win=210 Len=0
97	3.488163...	172.27.118.106	18.192.93.86	TCP	66	[TCP ACKed unseen segment] 45960 → 14386 [ACK]
669	18.63919...	18.192.93.86	172.27.118.106	TCP	66	[TCP Dup ACK 96#1] 14386 → 45960 [ACK] Seq=1
672	18.63926...	172.27.118.106	18.192.93.86	TCP	66	[TCP Dup ACK 97#1] [TCP ACKed unseen segment]
11...	33.79413...	18.192.93.86	172.27.118.106	TCP	66	[TCP Dup ACK 96#2] 14386 → 45960 [ACK] Seq=1
11...	33.79419...	172.27.118.106	18.192.93.86	TCP	66	[TCP Dup ACK 97#2] [TCP ACKed unseen segment]
11...	36.36695...	172.27.118.106	18.192.93.86	TCP	69	[TCP ACKed unseen segment] 45960 → 14386 [PSH]
11...	36.46870...	18.192.93.86	172.27.118.106	TCP	66	[TCP Previous segment not captured] 14386 → 4
11...	36.57289...	18.192.93.86	172.27.118.106	TCP	67	14386 → 45960 [PSH, ACK] Seq=2 Ack=4 Win=210
11...	36.57294...	172.27.118.106	18.192.93.86	TCP	66	[TCP ACKed unseen segment] 45960 → 14386 [ACK]
16...	51.24031...	172.27.118.106	18.192.93.86	TCP	70	45960 → 14386 [PSH, ACK] Seq=4 Ack=3 Win=501
16...	51.34163...	18.192.93.86	172.27.118.106	TCP	66	14386 → 45960 [ACK] Seq=3 Ack=8 Win=210 Len=0
16...	51.87350...	18.192.93.86	172.27.118.106	TCP	358	14386 → 45960 [PSH, ACK] Seq=3 Ack=8 Win=210
16...	51.87354...	172.27.118.106	18.192.93.86	TCP	66	45960 → 14386 [ACK] Seq=8 Ack=295 Win=499 Len=

Frame 1682: 358 bytes on wire (2864 bits), 358 bytes captured (2864 bits) on interface wlp0s20f3, id 0

Ethernet II, Src: Cisco.5b:96:60 (00:26:99:5b:96:60), Dst: IntelCor_bc:e8:32 (34:2e:b7:bc:e8:32)

Internet Protocol Version 4, Src: 18.192.93.86, Dst: 172.27.118.106

Transmission Control Protocol, Src Port: 14386, Dst Port: 45960, Seq: 3, Ack: 8, Len: 292

```

0000 34 2e b7 bc e8 32 00 26 99 5b 96 60 08 00 45 00 4...2 & .[...E-
0010 01 58 29 87 40 00 e8 06 d5 7c 12 c0 5d 56 ac 1b .X).@...[...V...
0020 76 6a 38 32 b3 88 9f 15 b8 dc 89 23 8c 28 80 18 vj82...#(...
0030 00 d2 dc b7 00 00 01 01 08 0a 17 e1 69 06 1b 69 .....i.i
0040 3a 81 0a 43 61 6c 6c 20 6f 6e 65 20 6f 66 20 74 :..Call one of t
0050 68 65 20 66 6f 6c 6c 6f 77 69 6e 67 20 66 75 6e he follo wing fun
0060 63 74 69 6f 6e 73 3a 0a 48 45 4c 50 20 2d 20 74 ctions:- HELP - t
0070 6f 20 67 65 74 20 74 68 69 73 20 6d 65 73 73 61 o get th is messa
0080 67 65 0a 4c 49 53 54 20 2d 20 74 6f 20 6c 69 73 ge LIST - to lis
0090 74 20 74 68 65 20 66 69 6c 65 73 20 69 6e 20 74 t the fi les in t
00a0 68 65 20 63 75 72 72 65 6e 74 20 64 69 72 65 63 he curre nt direc
00b0 74 6f 72 79 0a 50 57 44 20 2d 20 74 6f 20 67 65 tory.PWD - to ge
00c0 74 20 74 68 65 20 63 75 72 72 65 6e 74 20 77 6f t the cu rrent wo
00d0 72 6b 69 6e 67 20 64 69 72 65 63 74 6f 72 79 0a rking di rectory.
00e0 43 44 20 3c 64 69 72 70 61 74 68 3e 20 2d 20 74 CD <dirp ath> - t
00f0 6f 20 63 68 61 6e 67 65 20 74 68 65 20 63 75 72 o change the cur

```

Help پیام

در تصویر بالا در wireshark بخش wifi می باشد که برای بررسی پکت های است که از سرور به کلاینت فرستاده می شود برای مثال در پیام بالا پیام help فرستاده شده است. تفاوتی که با حالت قبل دارد این است که در حالت قبل تمام پیام ها در بخش loopback رد و بدل می شود اما با استفاده از ngrok این پیام ها برای کلاینت در بخش wifi (بستگی به نوع اتصال) و در اینترنت رد و بدل می شد که همانطور که در تصویر بالا نشان داده شده است. برای بخش سرور هم کماکان پیام ها در بخش loopback ارسال می شد.

در ابتدا بدون هیچ تغییری این امر ممکن نیست زیرا نیاز است پورت جدیدی باز شود و ngrok تنها یک پورت در اختیار ما می گذارد.

شیوه پیاده سازی ngrok برای download کردن:
ابتدا در سرور یک دستور ngrok tcp random_port (با کتابخانه sys) میزنیم و هاست و پورت را برای کلاینت می فرستیم
و سپس کلاینت به آن port و host که فرستاده شده وصل می شود و فایل را دریافت می کند سپس آن اتصال بسته می شود.

نکته: اجرای دستور ngrok tcp random_port در سرور انجام می شود. و خروجی آنکه نام هاست و پورت است برای کلاینت فرستاده می شود تا به آن متصل شود.