

به نام یکتا سازنده هستی



پروژه دوم شبکه های کامپیوتری

شبکه های کامپیوتری

دانشگاه صنعتی اصفهان

---

خرداد ماه ۱۴۰۱

استاد درس:

دکتر محمدرضا حیدرپور

انجام دهندگان پروژه:

مصطفی دریس پور - احمد مردانی

۹۹۳۰۵۷۳ - ۹۹۰۰۲۸۳



## ۲- ارسال هر بسته دلخواه

(۱) ۱۴ بایت زیرا حداقل طول هدر ethernet این مقدار است.

(۲) فرمت بسته باید به صورت روبرو باشد: dest mac, src mac, length/type

(۳)

```
mostafa@mostafa-IdeaPad-L3-15IML05:~/src/Term 4/Computer Network/Project2/code$ sudo python3 pkt_sender.py
What is your packet content? 123456789012342eb7bce8320800
Which interface do you want to use? wlp0s20f3
Sent 14-byte on wlp0s20f3
```

eth.addr == 12:34:56:78:90:12						
No.	Time	Source	Destination	Protocol	Length	Info
11...	521.4777...	IntelCor_bc:e8...	12:34:56:78:90...	IPv4	14	[Malformed Packet]

(۴) بسته پایین کپی شد

eth.addr == 12:34:56:78:90:12						
No.	Time	Source	Destination	Protocol	Length	Info
11...	521.4777...	IntelCor_bc:e8...	12:34:56:78:90...	IPv4	14	[Malformed Packet]

بسته پایین مجددا ارسال شد

```
428 17.44507... fe80::ca3a:35f... fe80::f61a:f05... ICMP... 142 Destination Unreachable (no route to destinat...
23... 192.4613... fe80::f61a:f05... 2600:1901:0:38... TCP 94 [TCP Retransmission] 58880 -> 80 [SYN] Seq=0 W...
```

خروجی اجرای دستور:

```
mostafa@mostafa-IdeaPad-L3-15IML05:~/src/Term 4/Computer Network/Project2/code$ sudo python3 pkt_sender.py
What is your packet content? c83a3520b678342eb7bce83286dd6005da0300280640fe80000000000000f61a0f05f3d4ff0326001901000038d700000000000000e60000
5085943d9300000000a002fd206e800000020405a00402080a8e9275b80000000001030307
Which interface do you want to use? wlp0s20f3
Sent 94-byte on wlp0s20f3
```

(۵) حمله replay attack زمانی اتفاق می افتد که یک مجرم سایبری یک ارتباط شبکه امن را شنود

می کند، آن را رهگیری می کند، و سپس با فریبکاری آن را به تأخیر می اندازد یا دوباره ارسال

می کند تا گیرنده را به اشتباه هدایت کند تا آنچه را که هکر می خواهد انجام دهد.

معرفی ضمن یک مثال:

یکی از کارکنان یک شرکت با ارسال یک پیام رمزگذاری شده به مدیر مالی شرکت درخواست انتقال مالی می کند. یک مهاجم این پیام را شنود می کند و آن را ذخیره می کند و اکنون در موقعیتی است که می تواند آن را دوباره ارسال کند. از آنجا که این یک پیام معتبر است که به سادگی دوباره ارسال شده است، پیام قبلاً به درستی رمزگذاری شده است و برای مدیر مالی قانونی به نظر می رسد.



می توان از برنامه packet sender برای ارسال پکت های که توسط مهاجم ارسال می شوند و قانونی به نظر می رسند استفاده کرد زیرا این برنامه می تواند هر پکت دلخواهی را بفرستد.

### ۳- ارسال بسته های TCP SYN

(۱)

dest\_ip می تواند مقدار دلخواهی باشد و بستگی دارد که به کجا بخواهیم بسته بفرستیم.

dest\_port می تواند مقدار عددی پورتنی باشد که در سرور باز است مثلاً اگر وب باشد ۸۰.

Flag نیز می تواند بسته به اینکه چه نوع بسته tcp می خواهیم بفرستیم مقادیر متفاوتی داشته باشد.

ttl نیز می تواند مقدار متفاوتی داشته باشد.

seq\_num نیز می تواند مقدار دلخواهی داشته باشد.

(۲)

No.	Time	Source	Destination	Protocol	Length	Info
530	43.78576...	172.27.101.240	93.184.216.34	TCP	54	3000 → 80 [SYN] Seq=0 Win=29200 Len=0
534	44.03252...	93.184.216.34	172.27.101.240	TCP	60	80 → 3000 [SYN, ACK] Seq=0 Ack=1 Win=65535 Le...
535	44.03258...	172.27.101.240	93.184.216.34	TCP	54	3000 → 80 [RST] Seq=1 Win=0 Len=0

(۳) بسته دوم بسته ack ای هست که در پاسخ به بسته syn که توسط برنامه tcp\_sender.py ارسال

شده از سرور دریافت می شود.

پس از دریافت بسته دوم سیستم عامل که هیچ ایده ای ندارد که چرا این بسته دریافت شده است.

و خود او این handshake را شروع نکرده است اتصال را ریست می کند.

منبع پاسخ اول در این فروم:

<https://stackoverflow.com/questions/59612288/client-sends-rst-after-receive-syn-ack>

راه حل در فروم زیر:

<https://stackoverflow.com/questions/9058052/unwanted-rst-tcp-packet-with-scapy>

17...	117.7148...	172.27.101.240	93.184.216.34	TCP	54	3000 → 80 [SYN] Seq=0 Win=29200 Len=0
17...	118.0011...	93.184.216.34	172.27.101.240	TCP	60	80 → 3000 [SYN, ACK] Seq=0 Ack=1 Win=65535 Le...
17...	118.0011...	172.27.101.240	93.184.216.34	TCP	54	3000 → 80 [RST] Seq=1 Win=0 Len=0



۴) فایل ifinfo.sh به همراه سایر فایل ها در پروژه ضمیمه شده است.

#### ۴ - مینی وایر شارک

(۱) ۱۴ بایت

(۲) مینیمم ۲۰ بایت و ماکسیمم ۶۰ بایت

(۳) با چک کردن فلگ های SYN و ACK

(۴)

```
ahmad@ahmad-UX360CAK: ~/Public
ahmad@ahmad-UX360CAK:~/Public$ sudo python3 tcp_syn_sender.py
Sent 54-byte TCP SYN packet on wlp2s0
ahmad@ahmad-UX360CAK:~/Public$

ahmad@ahmad-UX360CAK:~/Public$ sudo python3 test.py
port 80 is open on 93.184.216.34
```

#### ۵ - مینی-انمپ

(۱)

```
ahmad@ahmad-UX360CAK: ~/Public
Send TCP SYN packet to port 1978
Send TCP SYN packet to port 1979
Send TCP SYN packet to port 1980
Send TCP SYN packet to port 1981
Send TCP SYN packet to port 1982
Send TCP SYN packet to port 1983
Send TCP SYN packet to port 1984
Send TCP SYN packet to port 1985
Send TCP SYN packet to port 1986
Send TCP SYN packet to port 1987
Send TCP SYN packet to port 1988
Send TCP SYN packet to port 1989
Send TCP SYN packet to port 1990
Send TCP SYN packet to port 1991
Send TCP SYN packet to port 1992
Send TCP SYN packet to port 1993
Send TCP SYN packet to port 1994
Send TCP SYN packet to port 1995
Send TCP SYN packet to port 1996
Send TCP SYN packet to port 1997
Send TCP SYN packet to port 1998
Send TCP SYN packet to port 1999
Send TCP SYN packet to port 2000
ahmad@ahmad-UX360CAK:~/Public$

ahmad@ahmad-UX360CAK:~/Public$ sudo python3 test.py
port 25 is open on 176.101.52.70
port 80 is open on 176.101.52.70
port 110 is open on 176.101.52.70
port 143 is open on 176.101.52.70
port 443 is open on 176.101.52.70
port 465 is open on 176.101.52.70
port 587 is open on 176.101.52.70
port 993 is open on 176.101.52.70
port 995 is open on 176.101.52.70
```

(۲) پورت ۲۵ برای SMTP relaying

پورت ۸۰ برای http



پورت ۱۱۰ برای unencrypted access to electronic mail

پورت ۱۴۳ برای Internet Message Access Protocol

پورت ۴۴۳ برای (either https or http) to divert network traffic

پورت ۴۶۵ برای implicit TLS

پورت ۵۸۷ برای encrypt SMTP messages using STARTTLS

پورت ۹۹۳ برای IMAP over SSL/TLS (IMAPS)

پورت ۹۹۵ برای SSL-encrypted POP3 service for encrypted mail transfer

(۳) به علت اینکه با سوکت tcp این کار را انجام میدهیم سرعت به شدت پایین تر میباشد.