به‌نام یکتا سازنده هستی

هر چه می خواهد دل تنگت ~~بگو~~ بگیر یا بفرست!

پروژه دوم درس شبکه‌های کامپیوتری

دانشکده‌ی برق و کامپیوتر

دانشگاه صنعتی اصفهان

۱ معرفی

در پروژه اول درس با نحوه ارسال و دریافت بسته توسط ایجاد ارتباطات TCP آشنا شدیم. در این پروژه می خواهیم کل پشته پروتکلی TCP/IP در داخل سیستم عامل را دور بزنیم و مستقیما «هر چیزی» (هر دنباله ای از ۰ و ۱ ها) را توسط یکی از interface های ماشین میزبان خود ارسال کنیم و یا «هر چیزی» که توسط آن دریافت می شود را تحویل بگیریم. سپس از این مهارت استفاده می کنیم تا یک مینی‌ـ‌وایرشاک و یک مینی‌ـ‌اِنمَپ بسازیم.

۲ ارسال هر بسته دلخواه

برنامه بنویسید که از کاربر نام یکی از interface های موجود (مثلا به نام interface0) و همچنین یک رشته هگزا دسیمال را دریافت کند و یک بسته اترنت که محتویات آن همان رشته ای است که از کاربر دریافت کرده است را توسط interface0 ارسال کند و گزارش موفقیت ارسال بسته را به صورت مشخص شده در راهنمایی زیر در ترمینال چاپ کند (برنامه خود را pkt_sender.py نامگذاری کنید).

راهنمایی ها:

۱. از دستور

```
$ ip link
```

در محیط ترمینال برای مشاهده interface های موجود دستگاه خود می توانید استفاده کنید.

۲. برای ارسال یک بسته دلخواه به نام pkt (بدون استفاده از پروتکل TCP/IP) توسط interface0 می توانید از دستورات زیر در کد پایتون استفاده کنید.

```
from socket import *
s = socket(AF_PACKET, SOCK_RAW)
s.bind((interface0, 0))
s.send(pkt)
```

همچنین توجه کنید که رشته هگزا دسیمال دریافتی از کاربر که به فرمت string است باید قبل از ارسال به فرمت باینری تبدیل شود. برنامه نمونه زیر برای تبدیل یک رشته هگزا دسیمالی به معادل باینری می تواند به شما برای انجام این کار کمک کند.

```
from binascii import unhexlify
message = "ac7ba14f4c0fea18"
pkt = " ".join(message[i:i+2] for i in range(0, len(message), 2))
```

۳. ورودی و خروجی برنامه شما می بایست به فرمتی مشابه تصویر زیر باشد.

```
→  p2codes sudo python pkt_sender.py
What is your packet content? acb2eff3476ac06178834eff800009adfe
Which interface do you want to use? wlan0
Sent 17-byte packet on wlan0
→  p2codes
```

سوالات

۱) حداقل طول بسته باید چند بایت باشد تا برنامه شما بتواند آن را ارسال کند؟ چرا؟
۲) برای ارسال یک بسته اترنت که توسط وایرشاک شناسایی شود، بسته شما باید چه فرمتی داشته باشد؟
۳) یک بسته اترنت معتبر خالی توسط برنامه خود ارسال کنید که در آن آدرس فیزیکی (MAC address) فرستنده به صورت 12:34:56:78:90:12 باشد. سپس وایرشاک خود را نصب باز کنید و در قسمت فیلتر مقدار زیر را قرار دهید:

```
eth.addr == 12:34:56:78:90:12
```

برنامه خود را اجرا کنید و به نحوی به آن مقدار دهید که بسته مزبور را ارسال کند و در وایرشاک آن را دریافت کنید. تصویر ترمینال که در برگیرنده ورودی و خروجی برنامه شما هست به همراه تصویر وایرشاک که بسته مزبور را دریافت کرده است را به عنوان جواب به این سوال در گزارش بیاورید.
۴) وایرشاک خود را باز کنید و مقدار قیلتر آن را برابر با `tcp` قرار دهید تا بسته های TCP که توسط دستگاه شما ارسال و یا دریافت می شوند را مشاهده کنید. یکی از این بسته ها را به دلخواه انتخاب کنید. سپس بر روی آن کلیک راست کرده و از منوی ظاهر شده گزینه Copy: ...as a HEX Stream را انتخاب نموده و رشته ای را که در کلیپ بورد شما ذخیره شده است به عنوان ورودی به برنامه pkt_sender.py بدهید تا همان بسته را مجدد ارسال کنید. اکنون در وایرشارک مقدار فیلتر را برابر با `tcp.port=<port number>` قرار دهید که در آن `<port number>` برابر با شماره پورت کلاینت شما در بسته TCP که انتخاب کردید می باشد. تصویر برنامه وایرشارک را به عنوان پاسخ به این سوال در گزارش خود بیاورید و مشخص کنید از بین بسته هایی که وایرشاک نشان میدهد کدام بسته، بسته تکراری است که برنامه شما ارسال کرده است.
۵) در مورد حمله replay attack تحقیق کنید و خلاصه ای از نحوه انجام این حمله را در پاسخ به این قسمت بیان کنید. آیا می توان از برنامه pkt_sender.py برای این حمله استفاده کرد؟

۳ ارسال بسته های TCP SYN

همانطور که در درس شبکه آموختید، ارتباطات TCP ارتباطات اتصال گرا هستند که قبل از شروع به رد و بدل داده، بسته های کنترلی بین کلاینت و سرور رد وبدل می شوند. اولیت بسته ای که توسط یک کلاینت ارسال می شود، TCP SYN نام دارد. در این قسمت از پروژه شما مبتنی بر تجربه ای که در قسمت قبل پیدا کرده اید که می توانید «هر چیزی» را توسط یکی از interface های دستگاه خود ارسال کنید، برنامه ای به نام tcp_syn_sender.py خواهید نوشت که می تواند بسته های TCP SYN ارسال کند (در این برنامه می بایست برای ارسال بسته ها نهایتا از همان برنامه pkt_sender.py استفاده کنید). به عنوان ورودی برنامه شما اطلاعات زیر را باید از یک فایل به نام info.txt دریافت کند (ترتیب اطلاعات نیز مهم است).

۱. آدرس IP یک سرور (مانند example.com)
۲. شماره پورت سرور
۳. آدرس IP متعلق به interface0 (interface ای از دستگاه خود که می خواهید برای ارسال بسته از آن استفاده کنید)
۴. شماره پورتی که می خواهید به عنوان پورت فرستنده در بسته ارسالی شما لحاظ شود.

۵. اسم interface0

۶. آدرس فیزیکی interface0

۷. آدرس فیزیکی روتر گذرگاه (gateway) زیرشبکه که در آن قرار دارید.

نمونه ای از فایل info.txt در شکل زیر نشان داده شده است.

info.txt
```
93.184.216.34
80
192.168.48.114
3000
wlan0
ac 7b a1 4f 4c 0f
ea 1e 5f 10 04 01
```

پس از دریافت این اطلاعات برنامه شما می بایست بسته TCP SYN را به مقصد سرور با آدرس IP و شماره پورت داده شده ارسال نماید و گزارش موفقیت ارسال بسته را به صورت مشخص شده در راهنمایی زیر در ترمینال چاپ کند.

راهنمایی ها

۱. برای پیدا کردن آدرس یک سرور می توانید از دستور dig استفاده کنید.

۲. برای پیدا کردن آدرس MAC روتر گذرگاه زیرشبکه خود می توانید از ترکیب اطلاعات دستورات ip rout list و arp استفاده نمایید. دستور مرکب زیر این ترکیب اطلاعات را می تواند برای شما انجام دهد و به عنوان خروجی آدرس MAC روتر گذرگاه را چاپ نماید.

```
$ arp|grep `ip route list|grep "default"|awk 'print $3'`|awk 'print $3'
```

(با جستجو در مورد قسمت های مختلف دستور مرکب فوق به نحوه کارکرد آن پی ببرید.)

۳. برای پیدا کردن آدرس MAC متعلق به interface دستگاه خود که می خواهید برای ارسال بسته ها از آن استفاده کنید می توانید از دستور ip link استفاده کنید.

۴. برای تبدیل آدرس ip_addr از فرمت دیسمال نقطه ای به فرمت هگز می توانید از دستور زیر استفاده کنید

```
inet_aton(ip_addr).encode("hex")
```

۵. محتوای بسته TCP SYN خود را به صورت زیر مقداردهی اولیه کنید.

```
fd= open('info.txt', 'r')
Lines = fd.readlines()

dest_mac = Lines[6][:17] #destination mac
src_mac = Lines[5][:17] #source mac
proto3 = "08 00" #layer 3 protocol number
ver="45" #version, header length
diff = "00" #diffserv
t_len = "00 28" #total length ("00 28" for 40 bytes, "00 3c" for 60 bytes)
id = "07 c3" #id
flags = "40 00" #flags
ttl = "40" #ttl
proto4 = "06" #layer 4 protocol number
cs3 ="00 00" #ip check sum
src_ip =inet_aton(Lines[2]).encode("hex") #source ip
dest_ip =inet_aton(Lines[0]).encode("hex") #dest ip
src_port = "%04x" %int(Lines[3]) #src port
dest_port ="%04x" %int(Lines[1]) #dest port
seq_num ="17 49 30 d1" #seq number
ack ="00 00 00 00" #ack number
h_len = "50 02" #tcp header length and flags ("a0 02" for 40 bytes, "50 02" for 20 bytes)
w_size = "72 10" #window size
cs4 = "00 00" #tcp check sum
up = "00 00" #urgent pointer

interface0 = Lines[4].strip()
```

۶. به منظور ارسال یک بسته معتبر که توسط روترها و یا مقصد دور ریخته نشود می بایست مقدار فیلدهای checksum در هدر IP و هدر TCP مقادیری باشند که بر اساس سایر اطلاعات موجود در بسته به درستی محاسبه شده باشند.

۵-۱) مقدار checksum در هدر IP بر اساس مقادیر تمام فیلدها در هدر IP (با مقدار اولیه صفر برای فیلد checksum) محاسبه می شود.

۵-۲) مقدار checksum در هدر TCP بر اساس مقادیر تمام قیلدها در هدر TCP (با مقدار اولیه صفر برای قیلد checksum)، تمام محتوای لایه کاربرد و همچنین بعضی از فیلدهای هدر IP محاسبه می شود. از آنجایی که بسته TCP SYN یک بسته کنترلی است، داده ای از لایه کاربرد نداریم. از طرف دیگر به فیلدهایی از هدر IP که در محاسبه checksum هدر TCP استفاده می شوند، شبه هدر (tcp pseudo header) می گویند. با جستجوی اینترنتی می توانید این فیلدها را شناسایی کنید و در محاسبه checksum هدر TCP آنها را لحاظ نمایید.

۵-۳) برای محاسبه checksum از فایل همراه پروژه به نام checksum.py استفاده کنید.

سوالات

۱) از بین مقادیری که در شکل بالا برای فیلدهای مختلف بسته مقداردهی شده اند، کدامیک می توانند مقدار دلخواهی داشته باشند؟

۲) برنامه خود را برای وبسایت www.example.com و پورت ۸۰ اجرا کنید. تصاویر وایرشارک با فیلتر `ip.addr == <ip_addr>` که در آن به جای `<ip_addr>` آدرس IP سایت www.example.com میابست قرار بگیرد را به عنوان پاسخ به این قسمت قرار دهید.

۳) در سوال قبل اصولا می بایست ۳ بسته به ازای یک ارسال بسته از دستگاه شما توسط وایرشارک نشان شده باشد. بسته اول همان بسته TCP SYN هست که دستگاه شما ارسال کرده است. دو بسته دیگر را تحلیل کنید (چه بسته هایی هستند و توسط چه برنامه ای ارسال شده اند؟)

۴) (تشویقی) یک اسکریپت (shell script) بنویسید که به عنوان ورودی مقادیر آدرس IP و شماره پورت مقصد، نام interface دستگاه خود و شماره پورت مبدا در بسته TCP SYN را دریافت کند و فایل info.txt را بسازد.

نمونه فراخوانی چنین اسکریپتی به نام ifinfo.sh به صورت زیر است:

```
$ bash ifinfo.sh 93.184.216.34 80 wlan0 3000
```

۴ مینی_وایرشارک

در این قسمت از شما خواسته می شود که یک وایرشارک خاص منظوره به نام miniwireshark.py بنویسید که تمام بسته های TCP SYN-ACK که برای دستگاه میزان و به مقصد یک پورت به خصوص در داخل دستگاه میزبان دریافت می کند را بتواند متوجه شود و نشان دهد که فرستنده این بسته چه آدرس IP و شماره پورتی دارد.

راهنمایی ها

۱. قسمت هایی از برنامه miniwireshark.py به صورت زیر داده شده است.

```
def ether(data):
    dest_mac, src_mac,proto = unpack('!6s 6s H',data[:14])
    dest_mac = ':'.join(re.findall('..', dest_mac.encode('hex')))
    src_mac = ':'.join(re.findall('..', src_mac.encode('hex')))

    return[dest_mac,src_mac,hex(proto),data[14:]]

def ip(data):
    maindata = data
    data=unpack('! B s H 2s 2s B B 2s 4s 4s',data[:20])
    return[data[0]>>4,  #version
    (data[0]&(0x0F))*4, #header length
    "0x"+data[1].encode('hex'), #Diffserv
    data[2], #total length
    "0x"+data[3].encode('hex'), #ID
    "0x"+data[4].encode('hex'), #flags
    data[5], #ttl
    data[6], #protocol
    "0x"+data[7].encode('hex'), #check sum
    socket.inet_ntoa(data[8]), #source ip
    socket.inet_ntoa(data[9]), #destination ip
    maindata[(data[0]&(0x0F))*4:] #ip payload
    ]
```

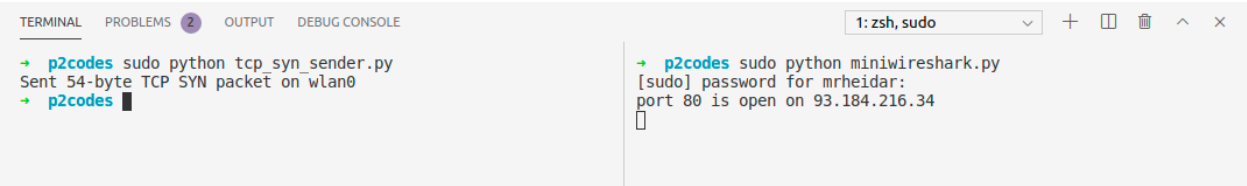
```
conn = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.ntohs(0x0003))
while True:
    raw_dat, add = conn.recvfrom(65535)
    ether_shark=ether(raw_dat)
    if(ether_shark[2]=="0x800"):
        ip_shark=ip(ether_shark[3])
```

شما در برنامه خود می بایست سایر قسمتهای لازم (عمدتا بخش مربوط به فهم هدر TCP) را بنویسید. توجه کنید که در قطعه کد پیشنهادی بار اصلی فهم لایه های پروتکلی را تابع unpack به دوش می کشد. با مراجعه به این

لینک با نحوه کار این تابع آشنا شوید و از همین تابع همانگونه که برای فهم لایه های Ethernet و IP استفاده شده است، برای فهم TCP استفاده نمایید.

سوالات

۱) با توجه به کدهای بالا، طول هدر اترنت چند بایت است؟
۲) با توجه به کدهای بالا، طول هدر اترنت چند بایت است؟
۳) در برنامه miniwireshark چگونه باید مطمئن شد که بسته های دریافتی SYN-ACK هستند؟
۴) برنامه خود را با کار کردن در مجاورت برنامه tcp_syn_sender.py را آزمایش کنید. نمونه ای از این همکاری بین این دو برنامه در شکل زیر نشان داده شده است (برنامه شما نیز باید به فرمت مشخص شده در شکل باشد)



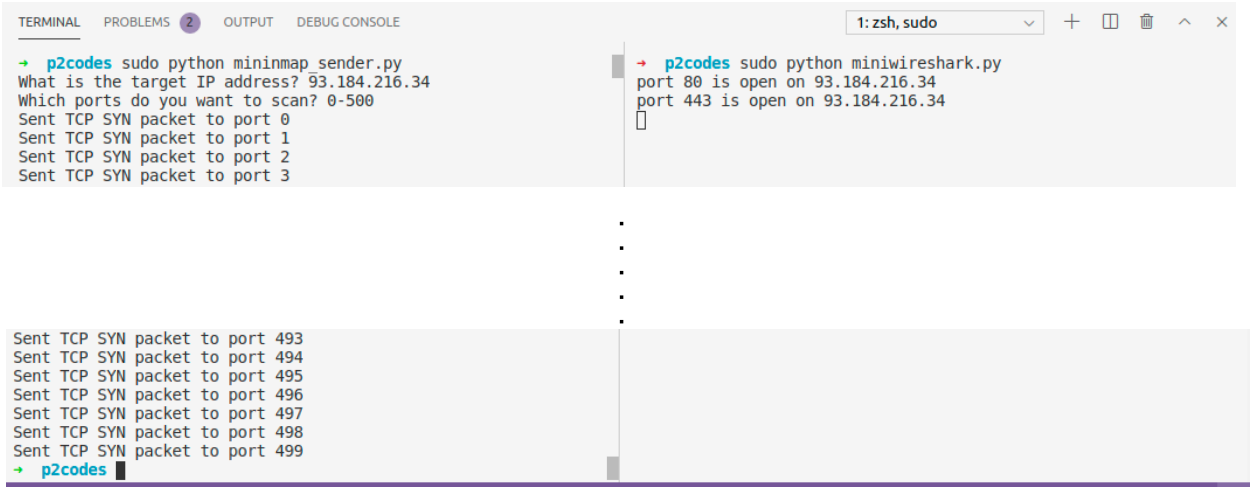
در گزارش خود تصویر متناظر از همکاری برنامه های خود را به عنوان پاسخ به این سوال قرار دهید.

۵ مینی_اِنمَپ

برنامه nmap یک برنامه متن باز برای کشف شبکه ها و بررسی جنبه های امنیتی آنها است. یکی از کارکردهای این برنامه این است یک رنج از پورت ها و یک آدرس IP می گیرد و توسط ارسال بسته های TCP SYN تشخیص می دهد کدامیک از پورتها باز هستند (سروری در حال گوش دادن بر روی آن و پذیرش ارتباط TCP است).
در این قسمت از شما خواسته می شود که با استفاده از برنامه هایی که در قسمت قبل نوشته اید (احیانا با تغییرات جزئی) برنامه ای به نام mininmap_sender.py بنویسید که به عنوان ورودی از کاربر آدرس IP هدف و رنج پورتهایی که قرار هست باز و یا بسته بودن آنها بررسی شود را بگیرد و برای همه آنها بسته TCP SYN ارسال کند. برنامه mininmap_sender.py در کنار برنامه miniwireshark.py می تواند کارکرد فوق از برنامه nmap را برای شما انجام دهد.

راهنمایی ها

۱. شکل زیر نمایی از کار کردن همزمان برنامه های miniwireshark و mininmap_sender را نشان می دهد. برنامه شما نیز باید با فرمت مشابهی از کاربر ورودی بگیرد و خروجی های مشابهی را به عنوان log چاپ کند.



سوالات

۱) با کدهایی که نوشته اید آزمایش کنید چه پورتهایی از آدرس 176.101.52.70 متعلق به دانشگاه صنعتی اصفهان در بازه ۰ تا ۲۰۰۰ باز هستند؟ شکلی مانند شکل فوق از کارکرد برنامه های خود در پاسخ به این سوال قرار دهید.

۲) با توجه به قسمت قبل چه سرویس های شناخته شده ای بر روی این آدرس دانشگاه ارائه می شود؟

۳) (تشویقی) به جای برنامه mininamap_sender برنامه ای به نام mininamp_sender_tcpsocket بنویسید که همان کار را به سریعترین وجه ممکن ولی با استفاده از ایجاد ساکتهای TCP انجام می دهد. در پاسخ به این قسمت عملکرد این برنامه را با برنامه mininamap_sender مقایسه کنید.

شیوه تحویل

برای این پروژه می بایست یک پوشه به نام studentid_proj2 بسازید (به جای studentid باید شماره دانشجویی خود را قرار دهید) که شامل دو فایل زیر باشد:

۱. یک فایل pdf: شامل پاسخ به سوالات و تصاویر خواسته شده

۲. یک پوشه با نام Source که در آن کد های پایتون

pkt_sender, tcp_syn_ender, miniwireshark, mininmap_sender

هستند (به صورت اختیاری فایلهای mininmap_sender_tcpsocket, ifinfo.sh نیز در داخل زیرپوشه مجزایی به نام Bonus ذیل پوشه Source قرار داده شوند).

در نهایت این فایل ها را فشرده کنید و یک فایل با نام studentid_prog1 و فرمت zip یا rar در سامانه یکتا ارسال کنید.

موفق باشید