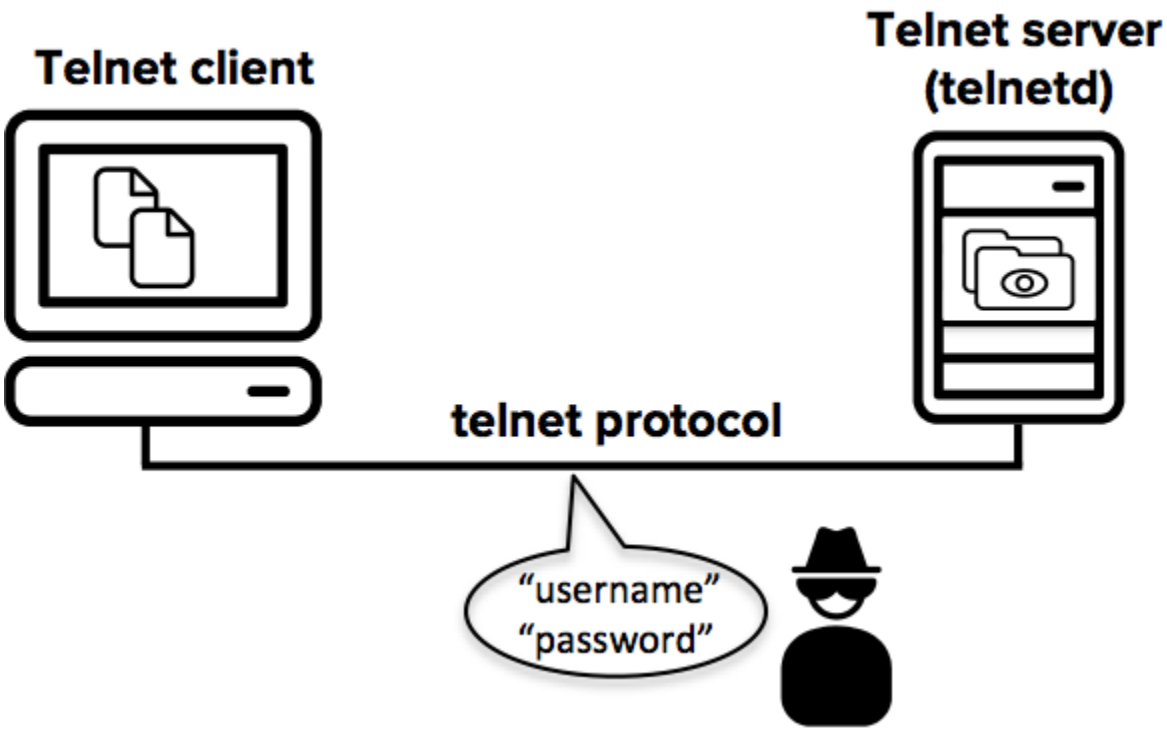
بررسی ارتباط دو سیستم از طریق Telnet و شنود شبکه به وسیله WireShark



Telnet چیست؟

Telnet یک پروتکل مبتنی بر روی TCP/IP است که برای اتصال به دستگاه ها و سرورها از راه دور استفاده می شود. این پروتکل در دهه ۱۹۶۰ توسط شرکت AT&T ایجاد شد و در ابتدا به عنوان یکی از ویژگی های شبکه های ARPANET (پدر بزرگ شبکه اینترنت xD) معرفی شد.

واژه Telnet نخستین بار توسط دو مخترع آمریکایی به نام های Albert Vezza و Thomas O'Sullivan اختراع شد.

با استفاده از Telnet، کاربران می توانند به سرورها و دستگاه های مختلفی که از پروتکل Telnet پشتیبانی می کنند، وصل شوند و دستورات خود را از طریق خط فرمان (command line) برای کنترل دستگاه وارد کنند.

در ابتدا نام کاربری و رمز عبور (در صورت احراز هویت) وارد می شود. سپس، کاربر به قسمتی از سیستم عامل و یا دستگاه که به آن متصل شده، وارد می شود.

تنها برخی از دستورات رایج را می توان در یک اتصال Telnet اجرا کرد. علاوه بر این، به دلیل عدم رمز نگاری، پروتکل Telnet به راحتی می تواند توسط افراد غیرمجاز و هکر هایی که به شبکه دسترسی دارند، مورد هدف برنامه های خود قرار گیرد. به همین دلیل، پروتکل Telnet به عنوان یکی از پروتکل های غیر امن شناخته می شود.

باید بگوییم که تلنت یک پروتکل کامپیوتری است که مسیر ارتباطی دوطرفه برای کامپیوترهای شبکه داخلی و خارجی ایجاد می‌نماید. پروتکل Telnet ، یک رابط کاربری دستوری دارد و عمده شهرت آن به این دلیل است که نخستین پروتکلی بود که اینترنت از طریق آن در سال 1969 استفاده شد.

telnet مخفف چیست ؟

تلنت مخفف network virtual terminal protocol است. این کلمات از teletype telecommunications network ,terminal network network برگرفته شده و به عنوان یک نسخه از (Remote Desktop Protocol) rdp ایجاد شده بود که بتواند کامپیوترها را از راه دور کنترل کند. Telnet به افراد مختلفی اجازه داد که از ترمینال‌های مختلف در دانشگاه‌ها و سازمان‌ها به قسمت های مختلف دیگر آن

ساختمان و سازمان دسترسی داشته باشند. این موضوع سبب شد تا به میزان قابل توجهی در وقت افراد صرفه جویی شود.

پروتکل telnet چگونه کار می کند؟ پورت telnet چیست ؟

Telnet درواقع نوعی پروتکل کلاینت- سرور است که می‌تواند برای باز کردن خط فرمان بر روی رایانه جهت اتصال از راه دور، به طور معمول یک سرور ، استفاده شود. کاربران می‌توانند از این ابزار برای پینگ پورت استفاده کنند و بدانند که آیا یک پورت باز است یا خیر. همچنین ممکن است پروتکل FTP به همراه Telnet برای کاربرانی که در ارسال و دریافت فایل‌ها کار می‌کنند، مورد استفاده قرار بگیرد. پس اگر برای شما سوال پیش آمد که کاربرد Telnet چیست و عملکرد فرمان های برنامه telnet به چه صورت است، می‌توان به پینگ گرفتن از یک پورت نیز اشاره کرد.

کاربران Remote به یک دستگاه با استفاده از Telnet متصل می‌شوند که به آن نیز Telnetting به یک سیستم گفته می‌شود. از آنها خواسته می‌شود نام کاربری و رمز عبور خود را برای دسترسی به رایانه از راه دور وارد کنند، که امکان اجرای خط فرمان به ورود شخص به رایانه را فراهم می‌کند. علیرغم موقعیت فیزیکی کاربران، آدرس IP آنها با رایانه وارد شده به جای شماره فیزیکی که برای اتصال استفاده می‌شود مطابقت دارد و حتی شما می‌توانید لیست آدرس آی‌پی‌های مجاز را محدود کنید تا حداقل امنیت شبکه کامپیوتری شما بیشتر شود. اما یکی از بزرگترین ضعف هایی که تلنت داشت، امنیت آن بود. زیر اتصالی که بین Client / server صورت می گرفت فاقد هر گونه رمزنگاری درست و مطمئنی بود ، از این رو کم کم این به مشکلی بزرگ در این پروتکل تبدیل شد

دستورات Telnet

Name	Byte code	Explanation	Notes
SE	240		
NOP	241		
Data Mark	242		
Break	243		
Interrupt Process	244		
Abort output	245		
Are you there?	246		
Erase character	247		
Erase Line	248		
Go ahead	249		
SB	250		
WILL	251		
WON'T	252		
DO	253		
DON'T	254		
Source: J. Postel and Reynolds (1983)[7]			

دستورات تلنت به صورت کلی حداقل باید شامل 2 Byte می بودند

که بایت اول IAC escape هست و بایت دوم که کد دستور است

شروع SSH(Secure Shell) و افول Telnet

در دنیای امروزی ، تلنت تکنولوژی منسوخ شده‌ای محسوب می‌شود اما با این حال، در دوران خود بسیار نوآورانه بود و کمک کرد تا بسیاری از دیگر پروتکل‌ها و تکنولوژی‌ها شکل بگیرند. با گذشت زمان، پروتکل telnet که بسیار ناامن بود، کم کم به پروتکل ssh تبدیل شد. این پروتکل که امروزه پر استفاده ترین پروتکل استفاده شده توسط مدیران سیستم عامل های لینوکسی است که با استفاده از آن سیستم عامل های لینوکسی و یونیکس را مدیریت می‌کنند. این پروتکل احراز هویت امنی را ارائه داده و اطلاعات رد و بدل شده حتی روی شبکه کامپیوتری نا امن را رمزنگاری می‌کند.

SSH رقیبی بی همتا برای telnet

جامعه توسعه دهنده گان زمانی که دریافتند telnet دارای ضعف های بسیاری است به دنبال جایگزینی امن برای آن بودند.

SSH یا Secure Shell به طور رسمی در سال 1995 توسط یک مهندس دانشگاه فناوری هایلبرن در آلمان به نام Tatu Ylönen ابداع شده است. در آن زمان، او به دنبال راهی برای رمزگذاری تراکنش های شبکه و جلوگیری از دسترسی غیرمجاز به اطلاعات بود.

تاتو یلونن (Tatu Ylönen) برای دسترسی به دستگاه هایی که به اینترنت وصل شده بودند، از پروتکل Telnet استفاده می کرد. اما او در آن زمان در مورد رمزنگاری Telnet اطمینان نداشت و به همین دلیل به دنبال یک راه پیدا کرد که از مزایای Telnet به همراه رمزنگاری قابل قبولی استفاده کند.

این آغاز پروژه، SSH1 بود که در نسخه 1.0 توسط Tatu Ylönen و تیمش در سال 1995 عرضه شد. تا زمانی که SSH1 به بازار عرضه شد، Telnet به عنوان ابزار پیش فرض برای مدیریت از راه دور سرویس دهی و نظارت بر سیستم های کامپیوتری استفاده می شد.

نسخه 2.0 از SSH یا SSH2 به تازگی در سال 1997 عرضه شد و تمرکز بیشتری بر روی امنیت داشت. SSH2 سریعاً شناخته شد و استقبال بسیاری را از سمت ارائه دهندگان خدمات شبکه و نرم افزارهای مختلف، به خود جلب کرد.

اکنون SSH یکی از پروتکل های امنیتی محبوب ترین برای مدیریت سرورها و دسترسی به دستگاه ها و سرورهای دیگر از راه دور است. از آنجا که SSH شامل رمزنگاری است، غیرممکن است برای حملات کرک شده و به راحتی ردیابی شود.

و یکی از نکات دیگری که SSH رو در میان توسعه دهنگان بسیار دوست داشتنی کرد بود Open Source(متن باز) بودن SSH بود به طوری که هر کسی میتوانست به Source Code اصلی دسترسی داشته باشد و در صورت داشتن دانش فنی آن را مطالعه کند!

https://github.com/openssh/openssh-portable

SSH امروزه به طور پیش فرض بر روی اکثر توزیع های لینوکسی نصب شده است

```
alisharify@DESKTOP-A6VOL6B:~$ ssh
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface]
           [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
           [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
           [-i identity_file] [-J [user@]host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun]] destination [command [argument ...]]
alisharify@DESKTOP-A6VOL6B:~$
```

اما برای نصب در توزیع های مختلف میتواند از دستورات زیر استفاده کنید

وابسطه به توزیع و package manger هر توزیع

بر پایه Debian:

```
sudo apt-get update
sudo apt-get install openssh-server
```

بر پایه Red hat:

```
sudo yum install openssh-server
```

بر پایه Fedora :

```
sudo dnf install openssh-server
```

```
sudo systemctl start ssh
sudo systemctl enable ssh
```

sudo systemctl status ssh

حالا که با telnet و ssh آشنا شدیم بیایید یه آزمایش کوچیک عملی با هم انجام بدیم !

در این آزمایش قراره یه telnet به صورت عملی بین 2 کامپیوتر انجام بدیم و شبکه رو شنود کنیم و این بین ببینیم می تونیم چه اطلاعاتی به دست بیاریم

خب برای این کار در داخل یه مجازی ساز یه ویندوز 7 آوردم بالا به عنوان سرور (چون تلنت سرور فقط روی سرور ها و ویندوز 7 به قبل موجوده)

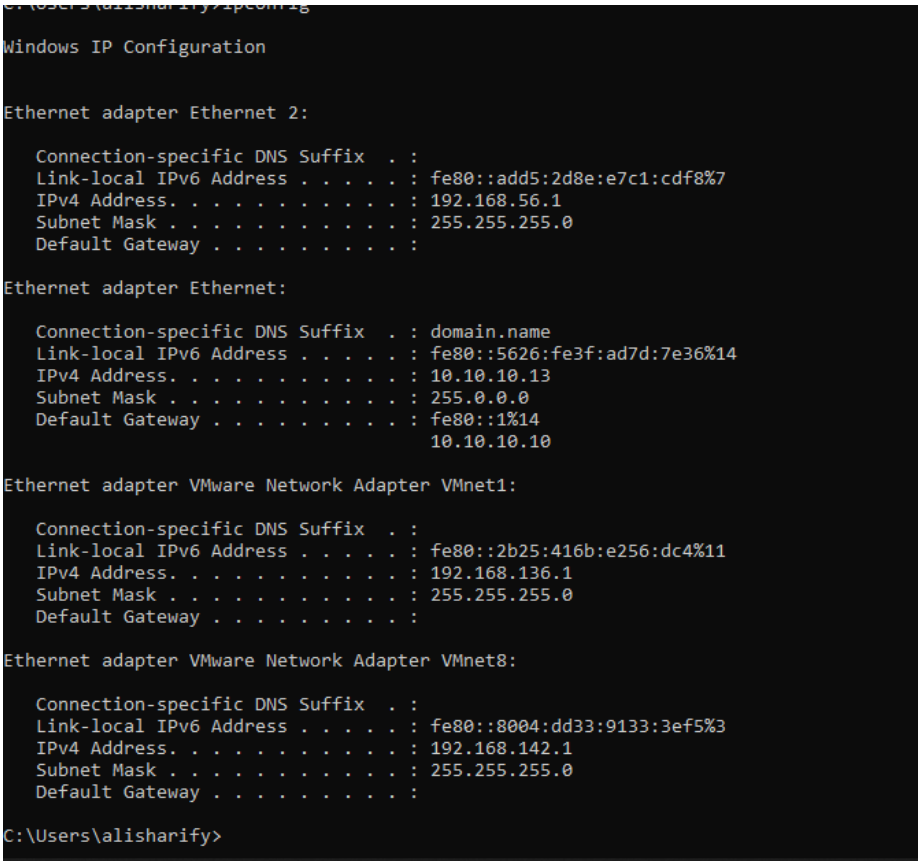
و برای کلاینت هم یه ویندوز 10 هست که قراره با تلنت به ویندوز 7 ریموت بزنه / برای اتصال شبکه هم حالت شبکه ماشین مجازی روی حالت Bridg هست که در این حالت مستقیم به کارت شبکه فیزیکی host متصل میشه و از DHCP یه IPدریافت می کنه برای خودش

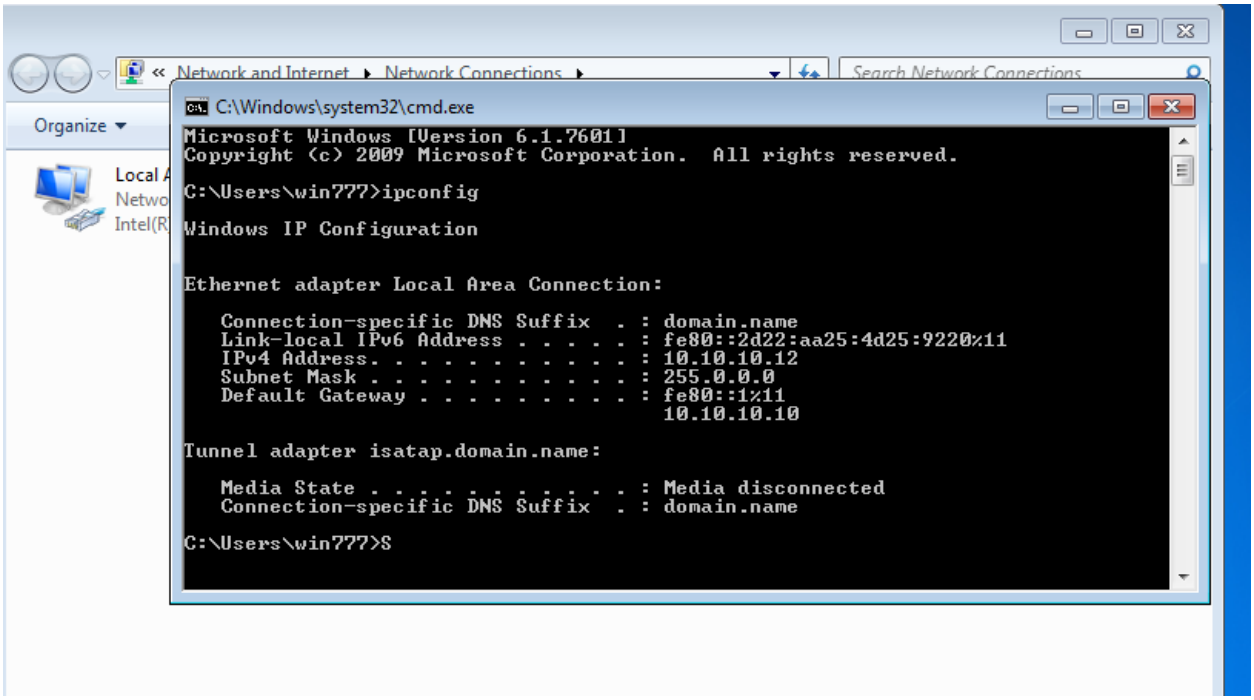
و همچنین باید توی Firewall جفت سیستم ها تنظیم کنیم که همدیگه رو بلاک نکنن و توی شبکه محلی به هم دسترسی داشته باشند

DHCP Server ip: 10.10.10.10

Telnet Server ip: 10.10.10.13

Telnet Client ip: 10.10.10.12





DHCP Mode

This page can be used to config the DHCP mode:None,DHCP Relay or DHCP Server.
(1)Enable the DHCP Server if you are using this device as a DHCP server. This page lists the IP address pools available your LAN. The device distributes numbers in the pool to host on your network as they request Internet access.
(2)Enable the DHCP Relay if you are using the other DHCP server to assign IP address to your host on the LAN. You can DHCP server IP address.
(3)If you choose "None", then the modem will do nothing when the host request a IP address.

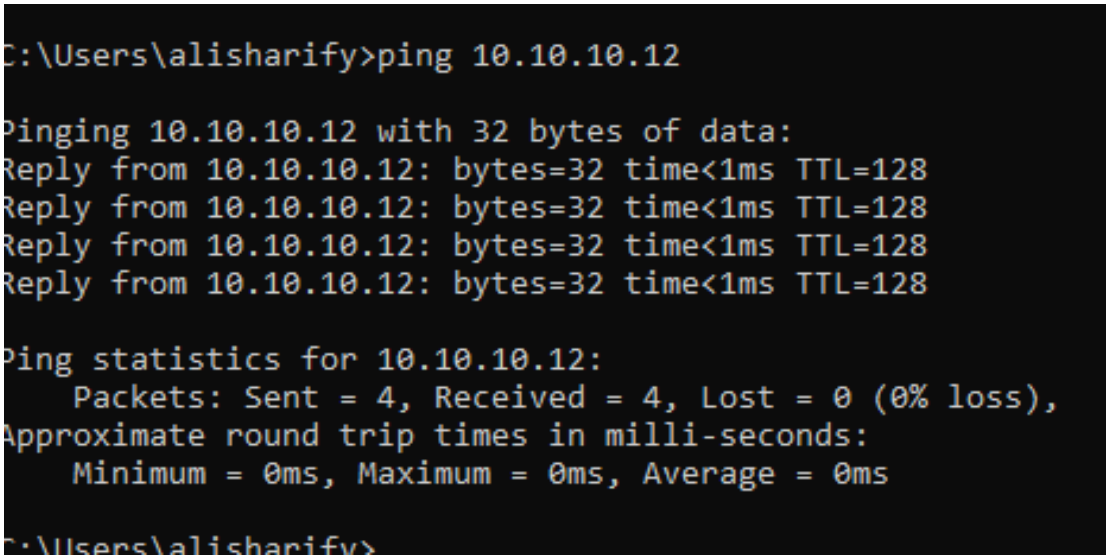
LAN IP Address: 10.10.10.10	**Subnet Mask: 255.0.0.0**
DHCP Mode:	DHCP Server

Interface:	☑ LAN1 ☑ LAN2 ☑ LAN3 ☑ LAN4 ☑ WLAN ☑ VAP0 ☑ VAP1 VAP2 ☑ VAP3
IP Pool Range:	10.10.10. 11 - 10.10.10. 250 Show Client
Subnet Mask:	255.0.0.0
Default Gateway:	10.10.10.10
Max Lease Time:	1440 minutes
Domain Name:	domain.name
DNS Servers:	10.10.10.10

Apply Changes Undo

Set VendorClass IP Range

خب برای شروع اول یه **ping** می کنیم سرور رو ببینیم میتونیم توی شبکه محلی بهش دسترسی داشته باشیم یا نه؟

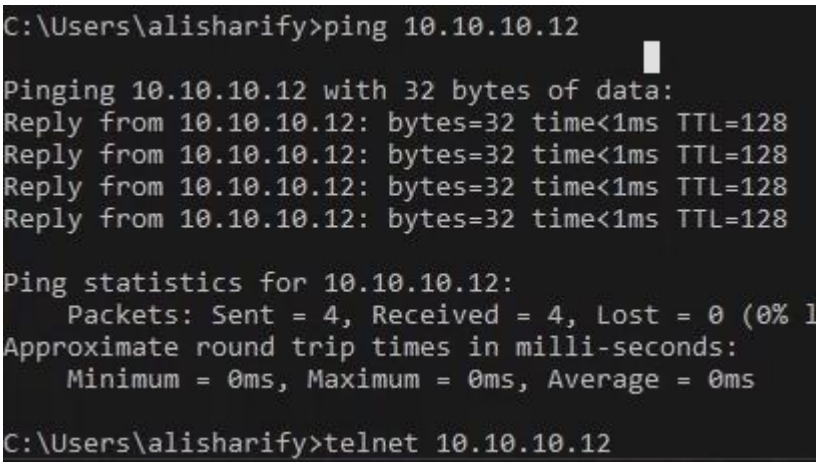


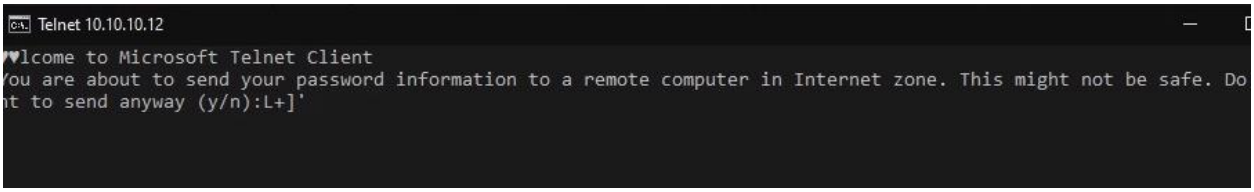
خب زمانی که از دسترسی مطمین شدیم میتونیم از طریق **telnet** به ماشین مجازی متصل ریموت بزنیم

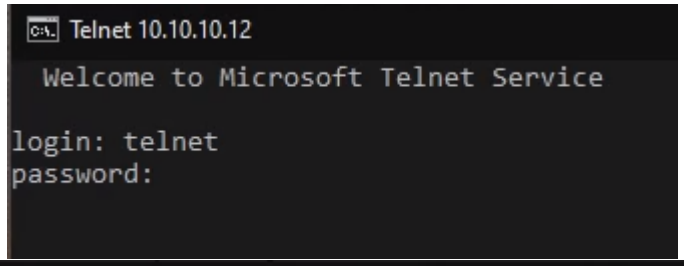
با دستور

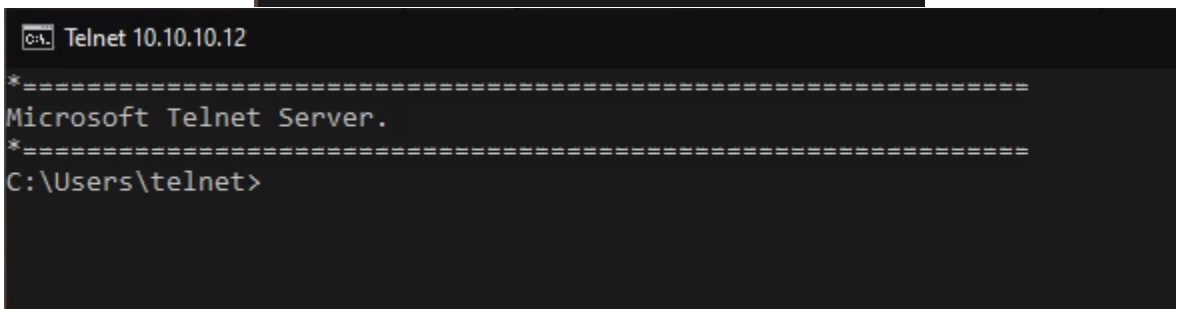
telnet [server ip]

میتونیم به ماشین مورد نظر با استفاده از **telnet** متصل بشیم اما باید روی ماشین مورد نظر **telnet server** نصب شده باشه و همچنین اجرا شده باشه و روی پورت **23** آماده به کار باشه









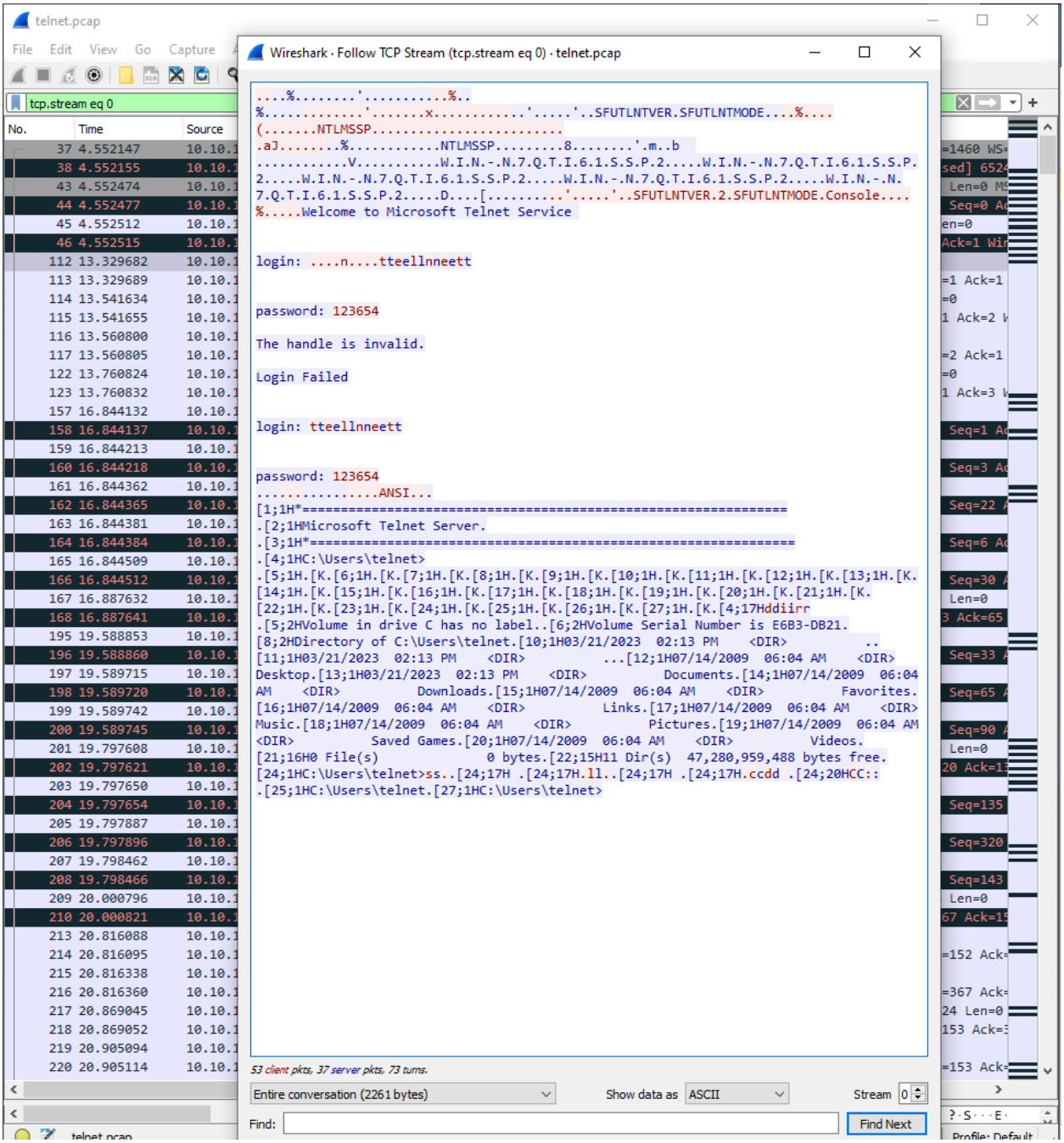
برای شنود شبکه هم قبل از ریموت زدن میتونیم با استفاده از Wireshark که یه برنامه برای شنود شبکه (ذخیره پکت های ارسالی و دریافتی) شبکه هست استفاده کنیم

برنامه های مشابه زیادی برای شنود شبکه وجود داره مثل DSNIFF, driftnet, TCPDUMP و ... که توی این آزمایش ما از wireshark استفاده می کنیم

زمانی که شنود ما تموم شد میتونیم فایل اون پکت های شبکه رو به صورت pcap. ذخیره کنیم تا بعدا بتونیم دیتا مورد نظرمون رو ازش استخراج کنیم

روش معمول برای این کار نوشتن برنامه هایی هست که با استفاده از اون روی فایل شنود شبکه استخراج دیتا انجام بدیم و دیتای مورد نظرمون رو بدست بیاریم چون به خودی خود فایل های شبکه حجم زیادی دیتا دارن به طور معمول بالای 100 هزار پکت و جستجو داخل این پکت ها به صورت دستی تقریبا غیر ممکنه!

اما باز به کمک ابزار هایی مثل wireshark میشه یه سری اعمال انالیزی انجام داد برای این کار ما اول میاییم تمام پکت هایی که پروتکلشون telnet هست رو جدا می کنیم بعد از منو Analyz میتونیمtcp/STREAM هارو انتخاب کنیم و تمام !



و اینجا داده های رد و بدل شده تحت شبکه رو میبینیم! و نام وکاربری و پسورد رو میتونیم خیلی عادی ببینیم!

آما برای اینکه همین دیتارو دستی استخراج کنیم باید براش یه برنامه بنویسیم برای این کار ما از پکیج **scapy Python** استفاده می کنیم که برای همین کار توسعه داده شده

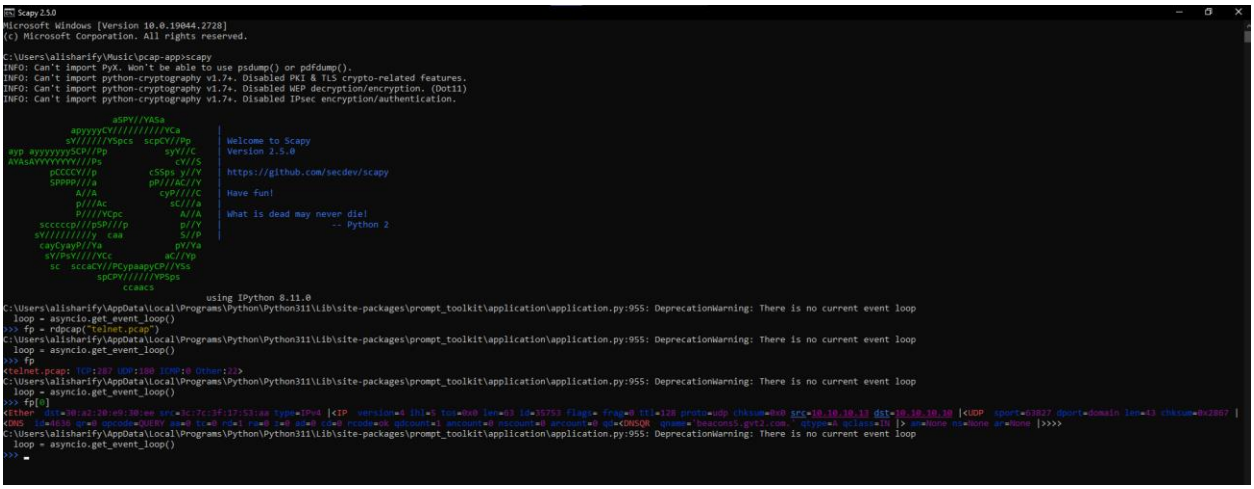
با استفاده از دستور

Pip install scapy

Pip3 install scapy

میتونیم این پکیج رو دانلود و نصب کنیم

Scapy به صورت کلی یه رابط خط فرمان هم داره که میتونیم استفاده کنیم یا به صورت معمول از طریق متد ها و کتابخونه های داخلیش توی برنامه هایی که می نویسیم استفاده کنیم



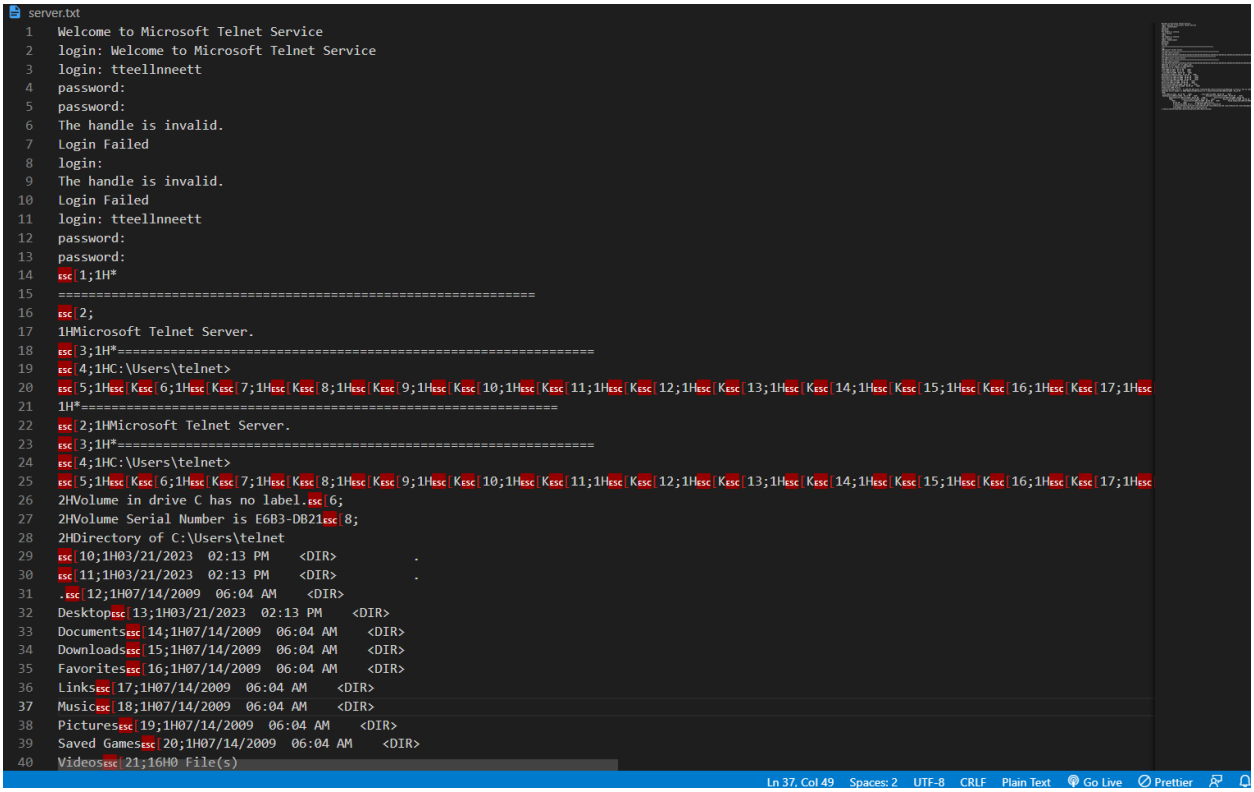
```
                                using IPython 8.11.0
C:\Users\alisharify\AppData\Local\Programs\Python\Python311\Lib\site-packages\prompt_toolkit\application\application.py:955: DeprecationWarning: There is no current event loop
  loop = asyncio.get_event_loop()
>>> fp = rdpcap("telnet.pcap")
C:\Users\alisharify\AppData\Local\Programs\Python\Python311\Lib\site-packages\prompt_toolkit\application\application.py:955: DeprecationWarning: There is no current event loop
  loop = asyncio.get_event_loop()
>>> fp
<telnet.pcap: TCP:287 UDP:180 ICMP:0 Other:22>
C:\Users\alisharify\AppData\Local\Programs\Python\Python311\Lib\site-packages\prompt_toolkit\application\application.py:955: DeprecationWarning: There is no current event loop
  loop = asyncio.get_event_loop()
>>> fp[0]
<Ether  dst=30:a2:20:e9:30:ee src=3c:7c:3f:17:53:aa type=IPv4 |<IP  version=4 ihl=5 tos=0x0 len=63 id=35753 flags= frag=0 ttl=128 proto=udp chksum=0x0 src=10.10.10.13 dst=10.10.10.10 |<UDP  sport=63827 dpor
<DNS  id=4636 qr=0 opcode=QUERY aa=0 tc=0 rd=1 ra=0 z=0 ad=0 cd=0 rcode=ok qdcount=1 ancount=0 nscount=0 arcount=0 qd=<DNSQR  qname='beacons5.gvt2.com.' qtype=A qclass=IN |> an=None ns=None ar=None |>>>>
C:\Users\alisharify\AppData\Local\Programs\Python\Python311\Lib\site-packages\prompt_toolkit\application\application.py:955: DeprecationWarning: There is no current event loop
  loop = asyncio.get_event_loop()
>>> fp[0][1].proto
17
C:\Users\alisharify\AppData\Local\Programs\Python\Python311\Lib\site-packages\prompt_toolkit\application\application.py:955: DeprecationWarning: There is no current event loop
  loop = asyncio.get_event_loop()
>>> fp[0][1]
<IP  version=4 ihl=5 tos=0x0 len=63 id=35753 flags= frag=0 ttl=128 proto=udp chksum=0x0 src=10.10.10.13 dst=10.10.10.10 |<UDP  sport=63827 dport=domain len=43 chksum=0x2867 |<DNS  id=4636 qr=0 opcode=QUERY
=0 rcode=ok qdcount=1 ancount=0 nscount=0 arcount=0 qd=<DNSQR  qname='beacons5.gvt2.com.' qtype=A qclass=IN |> an=None ns=None ar=None |>>>
C:\Users\alisharify\AppData\Local\Programs\Python\Python311\Lib\site-packages\prompt_toolkit\application\application.py:955: DeprecationWarning: There is no current event loop
  loop = asyncio.get_event_loop()
>>> str(fp[0][1].proto)
'17'
C:\Users\alisharify\AppData\Local\Programs\Python\Python311\Lib\site-packages\prompt_toolkit\application\application.py:955: DeprecationWarning: There is no current event loop
  loop = asyncio.get_event_loop()
>>> fp[0][1].proto
17
C:\Users\alisharify\AppData\Local\Programs\Python\Python311\Lib\site-packages\prompt_toolkit\application\application.py:955: DeprecationWarning: There is no current event loop
  loop = asyncio.get_event_loop()
>>> fp[0][1].ttl
128
C:\Users\alisharify\AppData\Local\Programs\Python\Python311\Lib\site-packages\prompt_toolkit\application\application.py:955: DeprecationWarning: There is no current event loop
  loop = asyncio.get_event_loop()
>>>
```

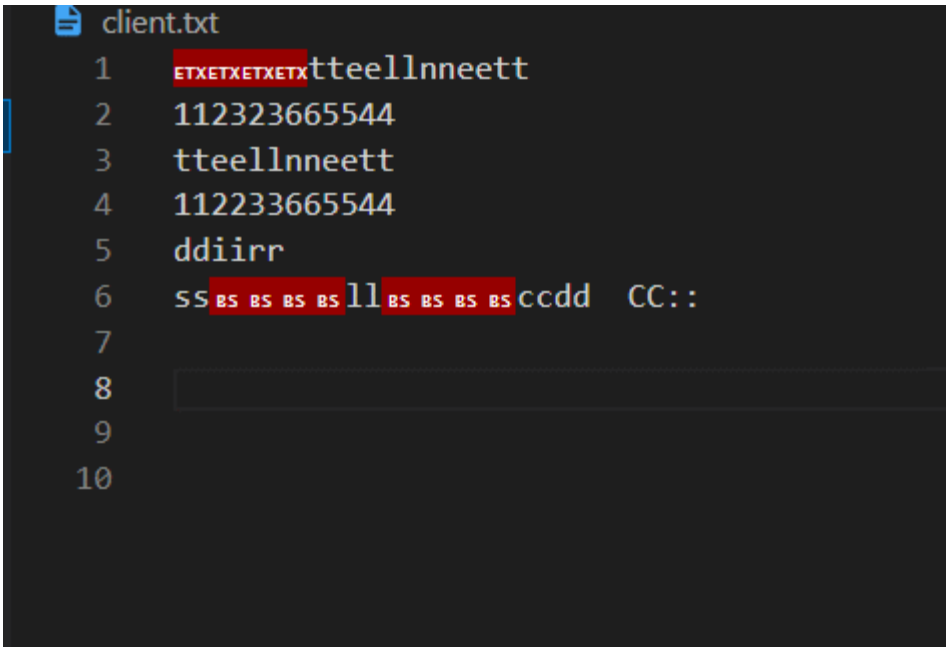
با استفاده از قطعه کد زیر میتونیم دیتا رو استخراج کنیم

app.py > ...

```python
from scapy.all import rdpcap
from scapy.layers.inet import TCP

# read all data from pcap file
fp = rdpcap("telnet.pcap")

server_ip = "10.10.10.12"
client_ip = "10.10.10.13"

server = list()
client = list()

# iterate over all packets
for each in fp:
    try:
        # get that packets that source ip is equal to client ip
        if each.haslayer(TCP) and each[1].src == client_ip:
            client.append(each[2][1].load)
        # get that packets that source ip is equal to server ip
        elif each.haslayer(TCP) and each[1].src == server_ip:
            server.append(each[2][1].load)
    except(AttributeError, IndexError):
        pass

# write server data to file
with open("server.txt", "w") as f:
    for each in server:
        try:
            f.write(each.decode())
        except UnicodeDecodeError:
            pass

# write client data to file
with open("client.txt", "w") as f:
    for each in client:
        try:
            f.write(each.decode())
        except UnicodeDecodeError:
            pass
```

این برنامه خیلی ساده میاد فقط پکت هارو جمع می کنه و توی دو تا فایل txt جدا می نویسه پکت های server رو توی یه فایل به اسم server.txt می نویسه و برای client هم همینطور





و توی فایل های txt جدیدی که ایجاد شده میتونیم دیتای رد و بدل شده رو ببینیم

پایان !

ممنون که تا اینجا همراه من بودید

منابع:

https://arazcloud.com/blog/what-is-telnet/

https://en.wikipedia.org/wiki/Telnet

https://en.wikipedia.org/wiki/Secure_Shell

http://blog.johnmuellerbooks.com/2011/06/07/sniffing-telnet-using-wireshark/

این تحقیق به صورت متن باز در لینک زیر موجوده :

https://github.com/alisharify7/Univercity-Articles

در صورت هر گونه مشکل فنی و ... میتونید Pull Request بزنید

2023/3/22 – 1402/1/2