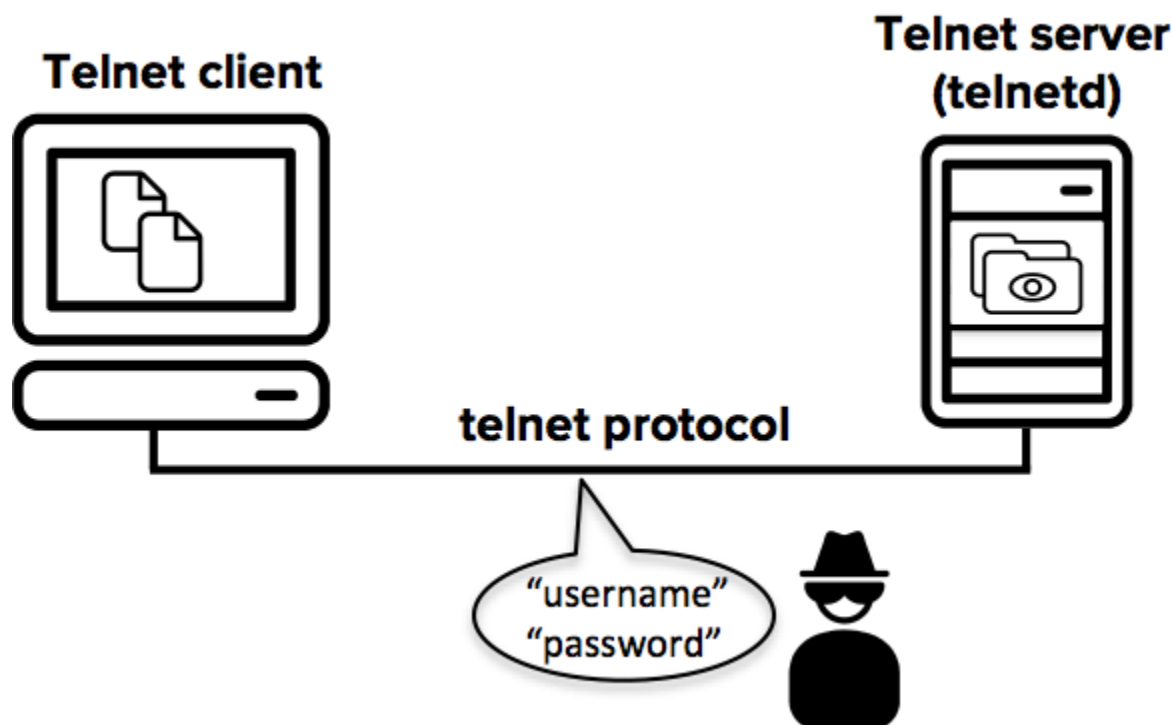


بررسی ارتباط دو سیستم از طریق Telnet و شنود شبکه به وسیله WireShark



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 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 می نماید. 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 network ,terminal network telecommunications 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 ترمینال های

مختلف در دانشگاه‌ها و سازمان‌ها به قسمت‌های 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 می‌کند؟ پورت telnet چیست ؟

Telnet در واقع 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 فرمان‌های 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 آدرس‌های پی‌های مجاز را محدود کنید تا حداقل امنیت شبکه کامپیوتری شما بیشتر شود. اما یکی از بزرگترین ضعف‌هایی که تلنت داشت، امنیت آن بود. زیر اتصالاتی که بین **Client / server** صورت می‌گرفت فاقد هر گونه رمزنگاری درست و مطمئنی بود ، از این رو کم‌کم 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 شده ای محسوب می شود اما با این حال، در دوران خود بسیار نوآورانه بود و کمک کرد تا بسیاری از دیگر پروتکل ها و تکنولوژی ها 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 می کنند. این پروتکل احراز هویت امنی را ارائه داده و اطلاعات رد و بدل شده حتی روی شبکه کامپیوتری نا امن را رمزنگاری می کند.

شروع SSH

جامعه توسعه دهندگان با توجه به ضعف های امنیتی موجود در Telnet به دنبال جایگزینی 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 دهندگان بسیار دوست داشتنی کرد بود Open Source (متن باز) بودن SSH بود به طوری که هر کسی میتواند 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 [-46AaCfGgKkMnqsTtVvXxYy] [-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

وابسته 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 تست

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 سازی 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 IP دریافت می کنه برای خودش

و همچنین باید Firewall جفت سیستم ها رو هم 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

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

```
C:\Users\alisharify>ping 10.10.10.12

Pinging 10.10.10.12 with 32 bytes of data:
Reply from 10.10.10.12: bytes=32 time<1ms TTL=128
Reply from 10.10.10.12: bytes=32 time<1ms TTL=128
Reply from 10.10.10.12: bytes=32 time<1ms TTL=128
Reply from 10.10.10.12: bytes=32 time<1ms TTL=128

Ping statistics for 10.10.10.12:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\alisharify>
```

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

```
C:\Users\alisharify>ping 10.10.10.12

Pinging 10.10.10.12 with 32 bytes of data:
Reply from 10.10.10.12: bytes=32 time<1ms TTL=128
Reply from 10.10.10.12: bytes=32 time<1ms TTL=128
Reply from 10.10.10.12: bytes=32 time<1ms TTL=128
Reply from 10.10.10.12: bytes=32 time<1ms TTL=128

Ping statistics for 10.10.10.12:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\alisharify>telnet 10.10.10.12
```

```
Telnet 10.10.10.12

Welcome to Microsoft Telnet Client
You are about to send your password information to a remote computer in Internet zone. This might not be safe. Do
nt to send anyway (y/n):L+]
```

```
C:\> Telnet 10.10.10.12

Welcome to Microsoft Telnet Service

login: telnet
password:

C:\> Telnet 10.10.10.12

*=====
Microsoft Telnet Server.
*=====
C:\Users\telnet>
```

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 و شبکه رو شنود کنیم

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

زمانی که شنود ما تموم شد میتونیم فایل پکت های شبکه رو به صورت .pcap ذخیره کنیم تا بعدا بتونیم دیتا مورد نظرمون رو ازش استخراج کنیم
روش های زیادی برای آنالیز دیتای شبکه وجود داره که خود WireShark هم این ابزار های آنالیز رو داره

اما شاید روش های خاص منظوره تر 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 آنالیزی انجام داد. برای این شنودی که ما انجام دادیم ابتدا تمام پکت هایی که پروتکلشون **telnet** هست رو جدا می کنیم بعد از منو **Analyz** میتونیم **tcp/STREAM** هارو انتخاب کنیم و تمام!

The image shows a Wireshark packet capture of a Telnet session. The left pane displays a list of packets, and the right pane shows the details of a selected packet (packet 158). The packet details show a Telnet login attempt with a password of 123654. The packet data is displayed in ASCII format.

Packet List (Left Pane):

No.	Time	Source
37	4.552147	10.10.1
38	4.552155	10.10.1
43	4.552474	10.10.1
44	4.552477	10.10.1
45	4.552512	10.10.1
46	4.552515	10.10.1
112	13.329682	10.10.1
113	13.329689	10.10.1
114	13.541634	10.10.1
115	13.541655	10.10.1
116	13.560800	10.10.1
117	13.560805	10.10.1
122	13.760824	10.10.1
123	13.760832	10.10.1
157	16.844132	10.10.1
158	16.844137	10.10.1
159	16.844213	10.10.1
160	16.844218	10.10.1
161	16.844362	10.10.1
162	16.844365	10.10.1
163	16.844381	10.10.1
164	16.844384	10.10.1
165	16.844509	10.10.1
166	16.844512	10.10.1
167	16.887632	10.10.1
168	16.887641	10.10.1
195	19.588853	10.10.1
196	19.588860	10.10.1
197	19.589715	10.10.1
198	19.589720	10.10.1
199	19.589742	10.10.1
200	19.589745	10.10.1
201	19.797608	10.10.1
202	19.797621	10.10.1
203	19.797650	10.10.1
204	19.797654	10.10.1
205	19.797887	10.10.1
206	19.797896	10.10.1
207	19.798462	10.10.1
208	19.798466	10.10.1
209	20.000796	10.10.1
210	20.000821	10.10.1
213	20.816088	10.10.1
214	20.816095	10.10.1
215	20.816338	10.10.1
216	20.816360	10.10.1
217	20.869045	10.10.1
218	20.869052	10.10.1
219	20.905094	10.10.1
220	20.905114	10.10.1

Packet Details (Right Pane):

158 16.844137 10.10.1

login: tteellnneett

password: 123654

The handle is invalid.

Login Failed

login: tteellnneett

password: 123654

.....ANSI...

[1;1H*=====

[2;1HMicrosoft Telnet Server.

[3;1H*=====

[4;1HC:\Users\telnet>

[5;1H.[K.[6;1H.[K.[7;1H.[K.[8;1H.[K.[9;1H.[K.[10;1H.[K.[11;1H.[K.[12;1H.[K.[13;1H.[K.[14;1H.[K.[15;1H.[K.[16;1H.[K.[17;1H.[K.[18;1H.[K.[19;1H.[K.[20;1H.[K.[21;1H.[K.[22;1H.[K.[23;1H.[K.[24;1H.[K.[25;1H.[K.[26;1H.[K.[27;1H.[K.[4;17Hddiirr

[5;2HVolume in drive C has no label..[6;2HVolume Serial Number is E6B3-D821.

[8;2HDirectory of C:\Users\telnet.[10;1H03/21/2023 02:13 PM <DIR> ..

[11;1H03/21/2023 02:13 PM <DIR> ...[12;1H07/14/2009 06:04 AM <DIR>

Desktop.[13;1H03/21/2023 02:13 PM <DIR> Documents.[14;1H07/14/2009 06:04 AM <DIR>

Downloads.[15;1H07/14/2009 06:04 AM <DIR> Favorites.

[16;1H07/14/2009 06:04 AM <DIR> Links.[17;1H07/14/2009 06:04 AM <DIR>

Music.[18;1H07/14/2009 06:04 AM <DIR> Pictures.[19;1H07/14/2009 06:04 AM <DIR>

Saved Games.[20;1H07/14/2009 06:04 AM <DIR> Videos.

[21;16H0 File(s) 0 bytes.[22;15H11 Dir(s) 47,280,959,488 bytes free.

[24;1HC:\Users\telnet>ss..[24;17H .[24;17H.ll..[24;17H .[24;17H.ccdd .[24;20HCC::

[25;1HC:\Users\telnet.[27;1HC:\Users\telnet>

53 client pkts, 37 server pkts, 73 turns.

Entire conversation (2261 bytes) Show data as ASCII Stream 0

Find: Find Next

و به همین راحتی داده رد و بدل شده تحت پروتکل Telnet قابل مشاهده است!

همینقدر ناامن

اما 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 توابع داخلش توی برنامه هایی که می
نویسیم استفاده کنیم / یکی از ویژگی هایی که بای این پروژه از **Scapy** استفاده
کردیم متن باز بود و همچنین بروز بودن اون بود / **Scapy** تقریبا بروز ترین
کتابخونه برای انجام این کاره و همچنین متن بازه XD

کتابخانه **Scapy** به صورت متن باز در لینک زیر موجوده

<https://github.com/secdev/scapy>

```
Scapy 2.5.0
Microsoft Windows [Version 10.0.19044.2720]
(c) Microsoft Corporation. All rights reserved.

C:\Users\alisharify\Music\pcap-app\scapy
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
INFO: Can't import python-cryptography v1.7+. Disabled PKI & TLS crypto-related features.
INFO: Can't import python-cryptography v1.7+. Disabled MEP decryption/encryption. (D0t11)
INFO: Can't import python-cryptography v1.7+. Disabled IPsec encryption/authentication.

      sSP//VNSa
      appyyycv/////////Vca
      sV/////////VSpCs scpCV//Pp
      sYP sYyYyYyYySsCv//Pp      syV//C
      AVASsYyYyYyYyYyYy//Pp      cv//S
      pCCCV//Pp      cSps y//Y      https://github.com/secdev/scapy
      sYpV//Pp      pP//Isc//Y
      A//A      cyP//Jc
      p//Isc      sc//A
      P//Isc      N//A
      scccccb//PpSP//Pp      p//Y
      sV/////////Y cma      S//P
      cVAsYpV//Pp      pY//p
      sVpVpV/////////VCS      hC//Yp
      sc secMv//IscYpVpV//Pp      sPcP/////////VSpCs
      cCABCS

      Welcome to Scapy
      Version 2.5.0
      Have fun!
      What is dead may never die!
      -- Python 2

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:188 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 checksum=0x0 src=10.10.10.13 dst=10.10.10.10 [<UDP sport=63827 dport=domain len=43 checksum=0x2867 |> 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()
>>>
```

```
      cCABCS

      Welcome to Scapy
      Version 2.5.0
      Have fun!
      What is dead may never die!
      -- Python 2

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:188 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 checksum=0x0 src=10.10.10.13 dst=10.10.10.10 [<UDP sport=63827 dport=domain len=43 checksum=0x2867 |> 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 checksum=0x0 src=10.10.10.13 dst=10.10.10.10 [<UDP sport=63827 dport=domain len=43 checksum=0x2867 |> 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 > ...
1  from scapy.all import rdpcap
2  from scapy.layers.inet import TCP
3
4  # read all data from pcap file
5  fp = rdpcap("telnet.pcap")
6
7  server_ip = "10.10.10.12"
8  client_ip = "10.10.10.13"
9
10 server = list()
11 client = list()
12
13 # iterate over all packets
14 for each in fp:
15     try:
16         # get that packets that source ip is equal to client ip
17         if each.haslayer(TCP) and each[1].src == client_ip:
18             client.append(each[2][1].load)
19         # get that packets that source ip is equal to server ip
20         elif each.haslayer(TCP) and each[1].src == server_ip:
21             server.append(each[2][1].load)
22     except (AttributeError, IndexError):
23         pass
24
25 # write server data to file
26 with open("server.txt", "w") as f:
27     for each in server:
28         try:
29             f.write(each.decode())
30         except UnicodeDecodeError:
31             pass
32
33 # write client data to file
34 with open("client.txt", "w") as f:
35     for each in client:
36         try:
37             f.write(each.decode())
38         except UnicodeDecodeError:
39             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


```
server.txt
1 Welcome to Microsoft Telnet Service
2 login: Welcome to Microsoft Telnet Service
3 login: tteellnneett
4 password:
5 password:
6 The handle is invalid.
7 Login Failed
8 login:
9 The handle is invalid.
10 Login Failed
11 login: tteellnneett
12 password:
13 password:
14 esc 1;1H*
15 =====
16 esc 2;
17 1HMicrosoft Telnet Server.
18 esc 3;1H*=====
19 esc 4;1HC:\Users\telnet>
20 esc 5;1Hesc Kesc 6;1Hesc Kesc 7;1Hesc Kesc 8;1Hesc Kesc 9;1Hesc Kesc 10;1Hesc Kesc 11;1Hesc Kesc 12;1Hesc Kesc 13;1Hesc Kesc 14;1Hesc Kesc 15;1Hesc Kesc 16;1Hesc Kesc 17;1Hesc
21 1H*=====
22 esc 2;1HMicrosoft Telnet Server.
23 esc 3;1H*=====
24 esc 4;1HC:\Users\telnet>
25 esc 5;1Hesc Kesc 6;1Hesc Kesc 7;1Hesc Kesc 8;1Hesc Kesc 9;1Hesc Kesc 10;1Hesc Kesc 11;1Hesc Kesc 12;1Hesc Kesc 13;1Hesc Kesc 14;1Hesc Kesc 15;1Hesc Kesc 16;1Hesc Kesc 17;1Hesc
26 2HVolume in drive C has no label.esc 6;
27 2HVolume Serial Number is E6B3-DB21esc 8;
28 2HDirectory of C:\Users\telnet
29 esc 10;1H03/21/2023 02:13 PM <DIR>
30 esc 11;1H03/21/2023 02:13 PM <DIR>
31 .esc 12;1H07/14/2009 06:04 AM <DIR>
32 Desktopesc 13;1H03/21/2023 02:13 PM <DIR>
33 Documentsesc 14;1H07/14/2009 06:04 AM <DIR>
34 Downloadsesc 15;1H07/14/2009 06:04 AM <DIR>
35 Favoritesesc 16;1H07/14/2009 06:04 AM <DIR>
36 Linksesc 17;1H07/14/2009 06:04 AM <DIR>
37 Musicesc 18;1H07/14/2009 06:04 AM <DIR>
38 Picturesesc 19;1H07/14/2009 06:04 AM <DIR>
39 Saved Gamesesc 20;1H07/14/2009 06:04 AM <DIR>
40 Videosesc 21;1H0 File(s)
```

```
client.txt
1 ETXETXETXtteellnneett
2 112323665544
3 tteellnneett
4 112323665544
5 ddiirr
6 ss BS BS BS BS ll BS BS BS BS ccdd CC::
7
8
9
10
```

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

جمع بندی :

پروتکل Telnet همین طور که در ابتدای این مقاله ذکر شد منسوخ شده و قدیمی هست اما همچنان در بعضی از جاها استفاده می شه و یکی از دلایل ضعف اصلی جدی که داشت امنیت پایین پروتکل بود به طوری که اگر شخص سومی ارتباط رو شنود می کرد می تونست تمام داده ها و اطلاعات رد و بدل شده رو بخونه و کاملاً ارتباط توی این پروتکل ناامنه.

برای رفع این ضعف ها SSH اومد که امنیت بالاتری داشت چون تمام بسته ها و داده هایی که ارسال می شد رمزنگاری می شد ، اما باید توجه داشته باشیم زمانی که عملیات رمزنگاری و رمزگشایی انجام میدیم به طور کلی به سرباری داره.

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