



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 Pane:** 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:

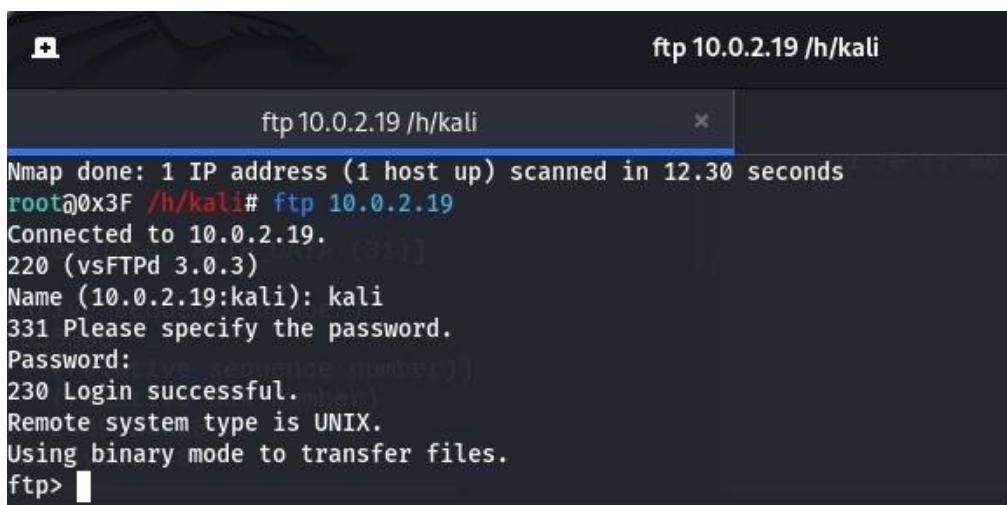
1. Sniffing FTP credentials using wireshark

FTP(File transfer protocol) 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,

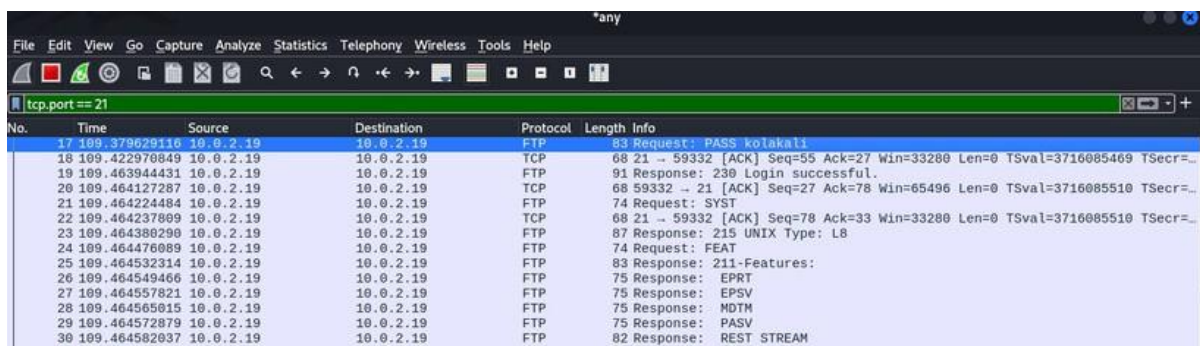
it is vulnerable to Man-in-the-Middle (MITM) attacks, where an attacker can eavesdrop on or manipulate data traffic between the client and FTP server.

Victim accesses FTP and enters credentials.



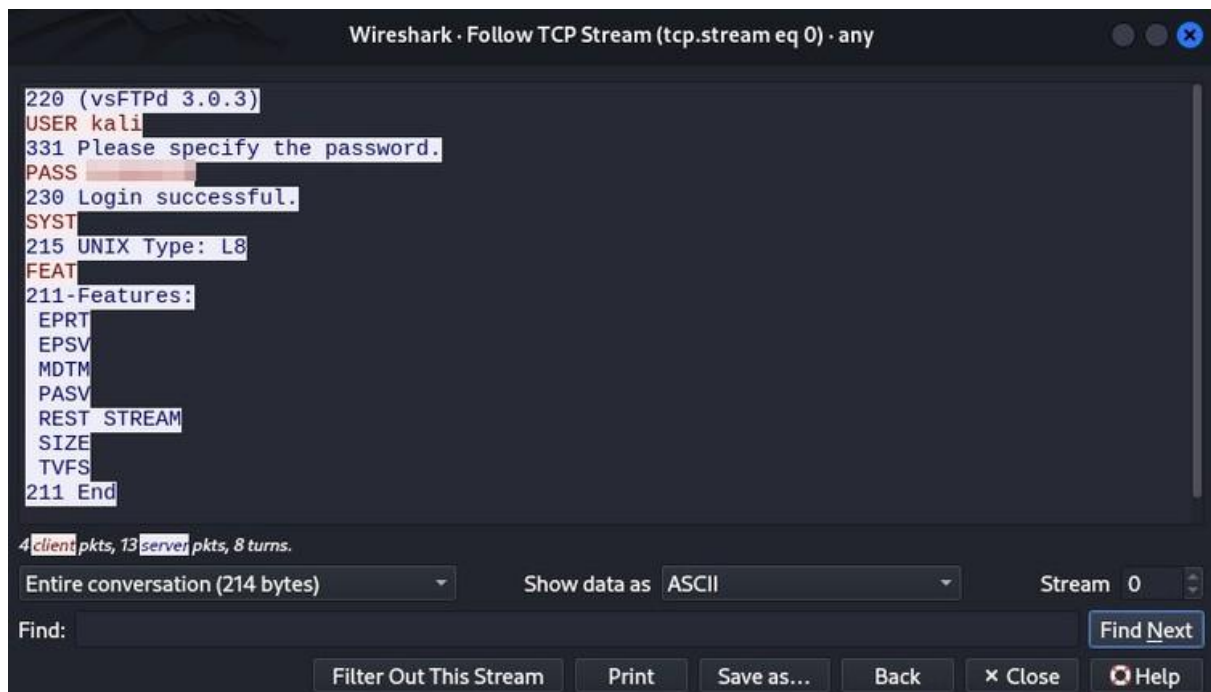
```
ftp 10.0.2.19 /h/kali
ftp 10.0.2.19 /h/kali
Nmap done: 1 IP address (1 host up) scanned in 12.30 seconds
root@0x3F /h/kali# ftp 10.0.2.19
Connected to 10.0.2.19.
220 (vsFTPD 3.0.3)
Name (10.0.2.19:kali): kali
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.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

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



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	109.464386290	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	109.464565815	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.



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.

← → ↻ 🏠 testphp.vulnweb.com/login.php

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

acunetix **acuart**

TEST and Demonstration site for **Acunetix Web Vulnerability Scanner**

[home](#) | [categories](#) | [artists](#) | [disclaimer](#) | [your cart](#) | [guestbook](#) | [AJAX Demo](#) | [Logout test](#)

search art

[Browse categories](#)
[Browse artists](#)
[Your cart](#)
[Signup](#)
[Your profile](#)
[Our guestbook](#)
[AJAX Demo](#)
[Logout](#)

Links
[Security art](#)
[PHP scanner](#)
[PHP vuln help](#)
[Fractal Explorer](#)

If you are already registered please enter your login information below:

Username :
Password :

You can also [signup here](#).
Signup disabled. Please use the username **test** and the password **test**.

[About Us](#) | [Privacy Policy](#) | [Contact Us](#) | ©2019 Acunetix Ltd

Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

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 display 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.

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.



```
Wireshark · Follow HTTP Stream (tcp.stream eq 2) · eth0

<p style="padding-left:5%;padding-right:5%"><b>Warning</b>: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.</p>
</div>
</body>
<!-- InstanceEnd --></html>
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 20
Origin: http://testphp.vulnweb.com
Connection: keep-alive
Referer: http://testphp.vulnweb.com/login.php
Upgrade-Insecure-Requests: 1

uname=test&pass=testHTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Mon, 23 Dec 2024 11:56:40 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Set-Cookie: login=test%2Ftest
Content-Encoding: gzip

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php" codeOutsideHTMIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
<!-- InstanceBeginEditable name="document_title_rgn" -->
<title>user info</title>
```

3. Analyzing Hydra Brute Force using Wireshark

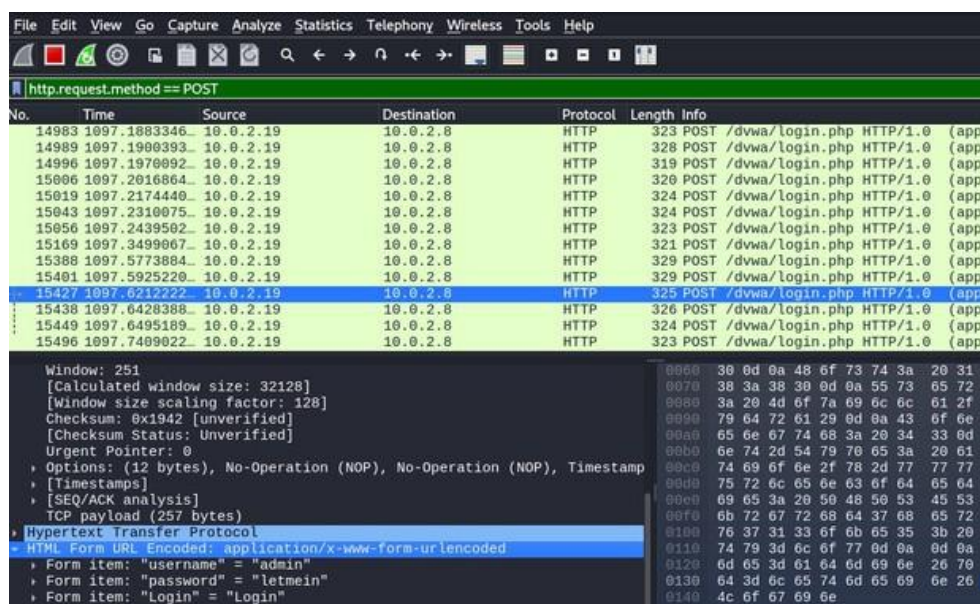
Hydra is one of the most well-known and powerful penetration testing tools used to conduct brute force attacks. A brute force attack is a method where an attacker tries all possible combinations of passwords to gain unauthorized access to a system.

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 or 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.



The image shows a Wireshark packet capture of an HTTP POST request to /dvwa/login.php. The packet list on the left shows multiple POST requests. The selected packet (No. 15427) is expanded in the packet details pane, showing the following structure:

- Window: 251
- [Calculated window size: 32128]
- [Window size scaling factor: 128]
- Checksum: 0x1942 [unverified]
- [Checksum Status: Unverified]
- Urgent Pointer: 0
- Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamp
- [Timestamps]
- [SEQ/ACK analysis]
- TCP payload (257 bytes)
- Hypertext Transfer Protocol
- HTML Form URL Encoded: application/x-www-form-urlencoded
- Form item: "username" = "admin"
- Form item: "password" = "letmein"
- Form item: "Login" = "Login"

The packet bytes pane on the right shows the raw data of the packet, including the HTTP headers and the form data.

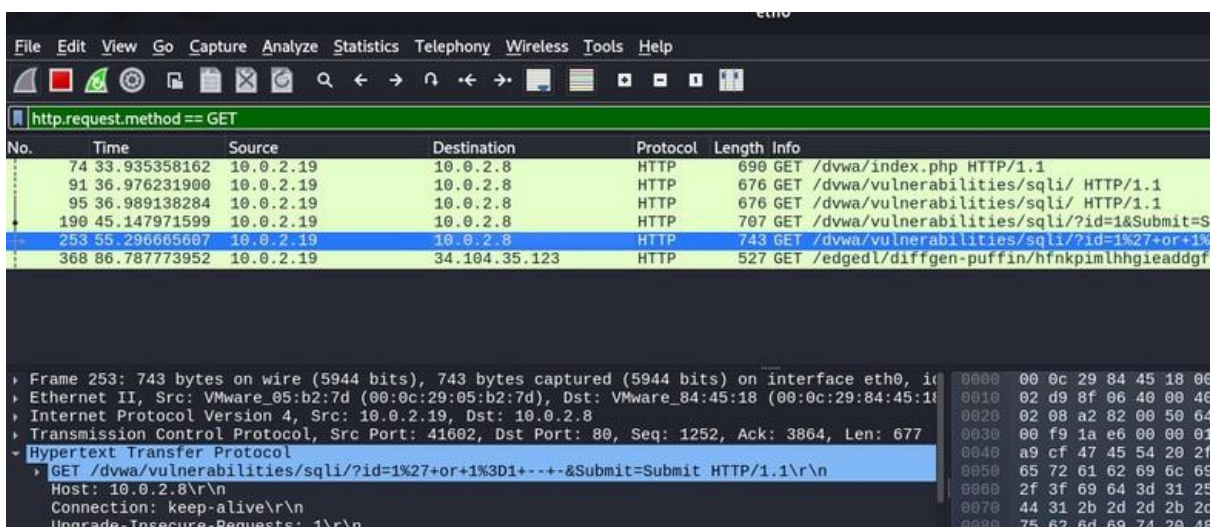
4. Detecting SQL Injection with Wireshark

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.



The screenshot shows the Acunetix AJAX Demo web application. The header includes the Acunetix logo and navigation links: home, categories, artists, disclaimer, your cart, guestbook, and AJAX Demo. A search bar is present with a 'go' button. A sidebar on the left contains links for 'Browse categories', 'Browse artists', 'Your cart', 'Signup', 'Your profile', 'Our guestbook', 'AJAX Demo', and 'Links' (Security art, PHP scanner, PHP vuln help, Fractal Explorer). The main content area has a login form with fields for 'Username' and 'Password', and a 'login' button. The 'Username' field contains the payload '1' or 1==1--'. Below the form, a message states: 'You can also [signup here](#). Signup disabled. Please use the username **test** and the password **test**.' A warning at the bottom states: 'Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.'

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.



The screenshot shows the Wireshark network protocol analyzer. The filter bar at the top is set to 'http.request.method == GET'. The packet list on the left shows several HTTP GET requests. The selected packet (No. 253) is an HTTP GET request to '/dvwa/vulnerabilities/sqli/?id=1%27+or+1%3D1+---&Submit=Submit'. The packet details on the right show the request structure: Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol. The raw data on the far right shows the hexadecimal representation of the packet bytes.

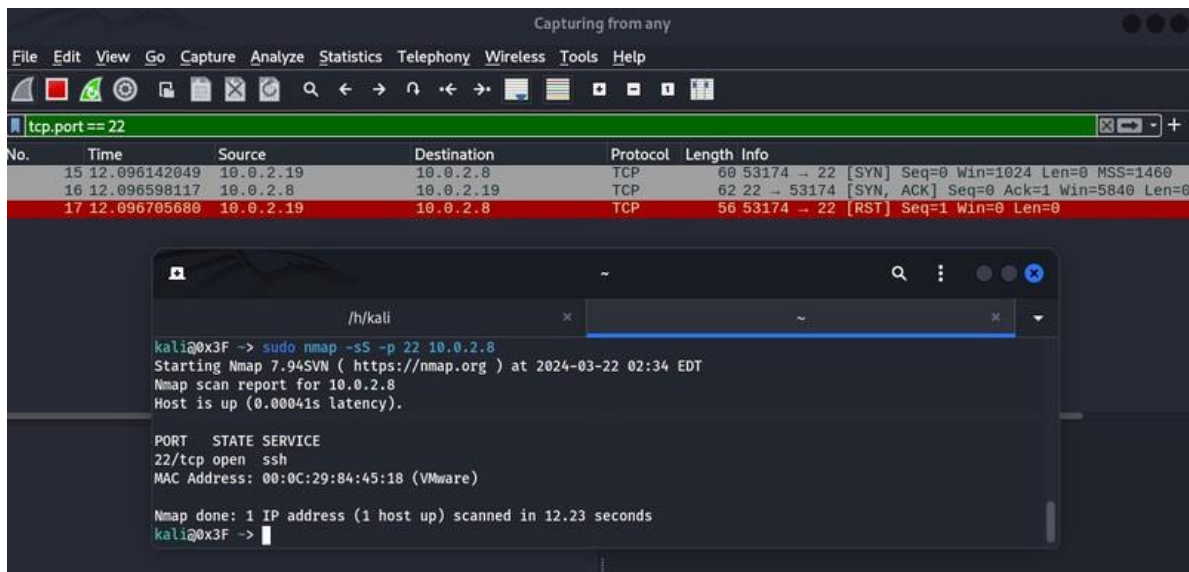
No.	Time	Source	Destination	Protocol	Length	Info
74	33.935358162	10.0.2.19	10.0.2.8	HTTP	690	GET /dvwa/index.php HTTP/1.1
91	36.976231900	10.0.2.19	10.0.2.8	HTTP	676	GET /dvwa/vulnerabilities/sqli/ HTTP/1.1
95	36.989138284	10.0.2.19	10.0.2.8	HTTP	676	GET /dvwa/vulnerabilities/sqli/ HTTP/1.1
190	45.147971599	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%3D1+---&Submit=Submit HTTP/1.1\r\n
368	86.787773952	10.0.2.19	34.104.35.123	HTTP	527	GET /edgedl/diffgen-puffin/hfnkpm1hghieaddgf

5. Analysis of Nmap Scanning using Wireshark

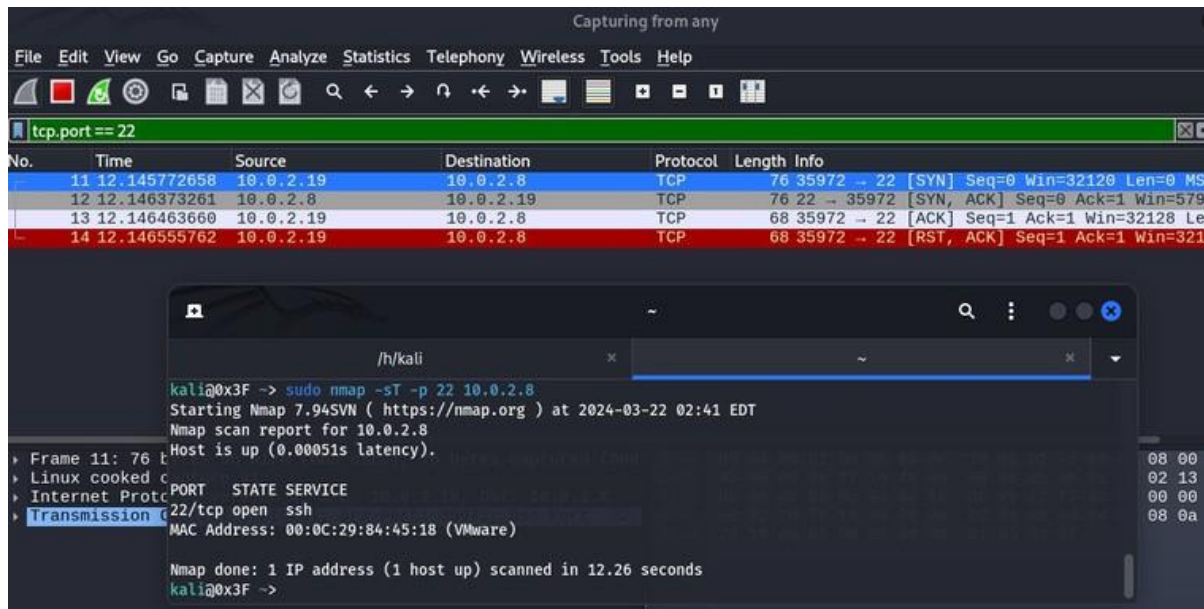
Nmap (Network Mapper) is one of the most popular and powerful network scanning tools used to discover hosts and services within a computer network. It allows users to conduct network scans to determine which hosts are active, what services are running on those hosts, as well as additional information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

Several flags commonly used in conjunction with Nmap are:

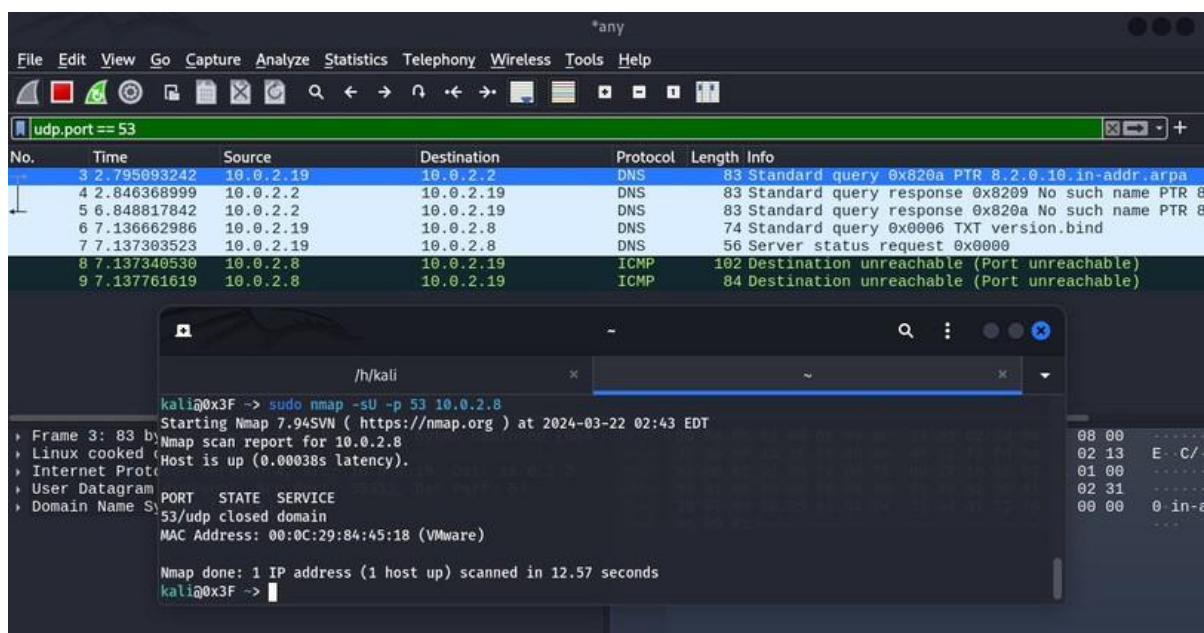
1. -sS (TCP SYN scan): This is the most common type of scan. Nmap sends a SYN packet to the target port and waits for a response. If the response is SYN/ACK, it means the port is open. If the response is RST, it means the port is closed. If there is no response, it may indicate the port is filtered.



- 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.



- 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.



- 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.

The image shows a Wireshark packet capture window with a filter set to `tcp.port == 22`. The packet list shows several TCP and SSH packets between `10.0.2.19` and `10.0.2.8`. Packet 14 is highlighted in red, showing a RST (Reset) packet from `10.0.2.19` to `10.0.2.8` with sequence number 565102852. Below the packet list, a terminal window is open, showing the execution of an Nmap scan with the command `sudo nmap -sS -p 22 -sV 10.0.2.8`. The terminal output shows the scan results for port 22, identifying it as an open SSH service running OpenSSH 5.3p1 Debian 3ubuntu4.

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

```
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/ .
```