01 AIM: Analyzing data packet send on network using Wireshark Packet Analyzer tool.

02 SOCOPE:

- Capturing and examining live data packets to understand network traffic flow, with a focus on identifying various protocols such as IP, TCP, UDP,http etc.
- Utilize Wireshark to investigate common web vulnerabilities as interacted with on testphp.vulnweb.com

03 FACILITIES

SOFTWARE Wireshark

04 THEORY

Network Protocol for Wireshark

Network protocols are the rules and conventions that dictate how data is exchanged over a network. These protocols are essential for facilitating communication between different devices and systems. When using Wireshark for network analysis, understanding these protocols is crucial for interpreting the data captured during packet analysis.

1**. Introduction to Network Protocols:**

Network protocols define a set of rules for data communication within and between networks.

They ensure data is transmitted in a structured and reliable manner, specifying how data is segmented, transmitted, and reassembled.

2. **The OSI Model:**

The Open Systems Interconnection (OSI) model is a conceptual framework used to understand network interactions in seven layers: Physical, Data Link, Network, Transport, Session, Presentation, and Application.Each layer has specific protocols and functions that contribute to the overall process of data communication.

3. **Key Network Protocols:**

Ethernet: A fundamental protocol used for data transmission over LANs, operating at the Data Link layer of the OSI model.

Internet Protocol (IP): Operates at the Network layer, facilitating routing and addressing of data packets across networks.

Transmission Control Protocol (TCP) and User Datagram Protocol (UDP): Operate at the Transport layer, managing the transmission of data. TCP ensures reliable delivery, establishing connections and ensuring data integrity. UDP, on the other hand, provides faster transmissions with no guarantee of delivery.

Hypertext Transfer Protocol (HTTP) and HTTP Secure (HTTPS): Application layer protocols used for web communication. HTTPS provides encryption for secure data transmission.

4. **Protocol Headers and Payloads:**

Each data packet encapsulated by a protocol includes a header and payload. The header contains metadata about the packet (such as source and destination addresses, protocol version, and checksums), while the payload carries the actual data.

Wireshark displays these components, allowing analysts to inspect the details of each protocol used in a packet's journey.

5. **Packet Analysis with Wireshark:**

Wireshark captures network packets in real-time, displaying the data at various protocol layers.

Analysts can use Wireshark's filtering and analysis tools to inspect protocol interactions, understand network behavior, diagnose issues, and identify security threats.

6. **Interpreting Protocol Information:**

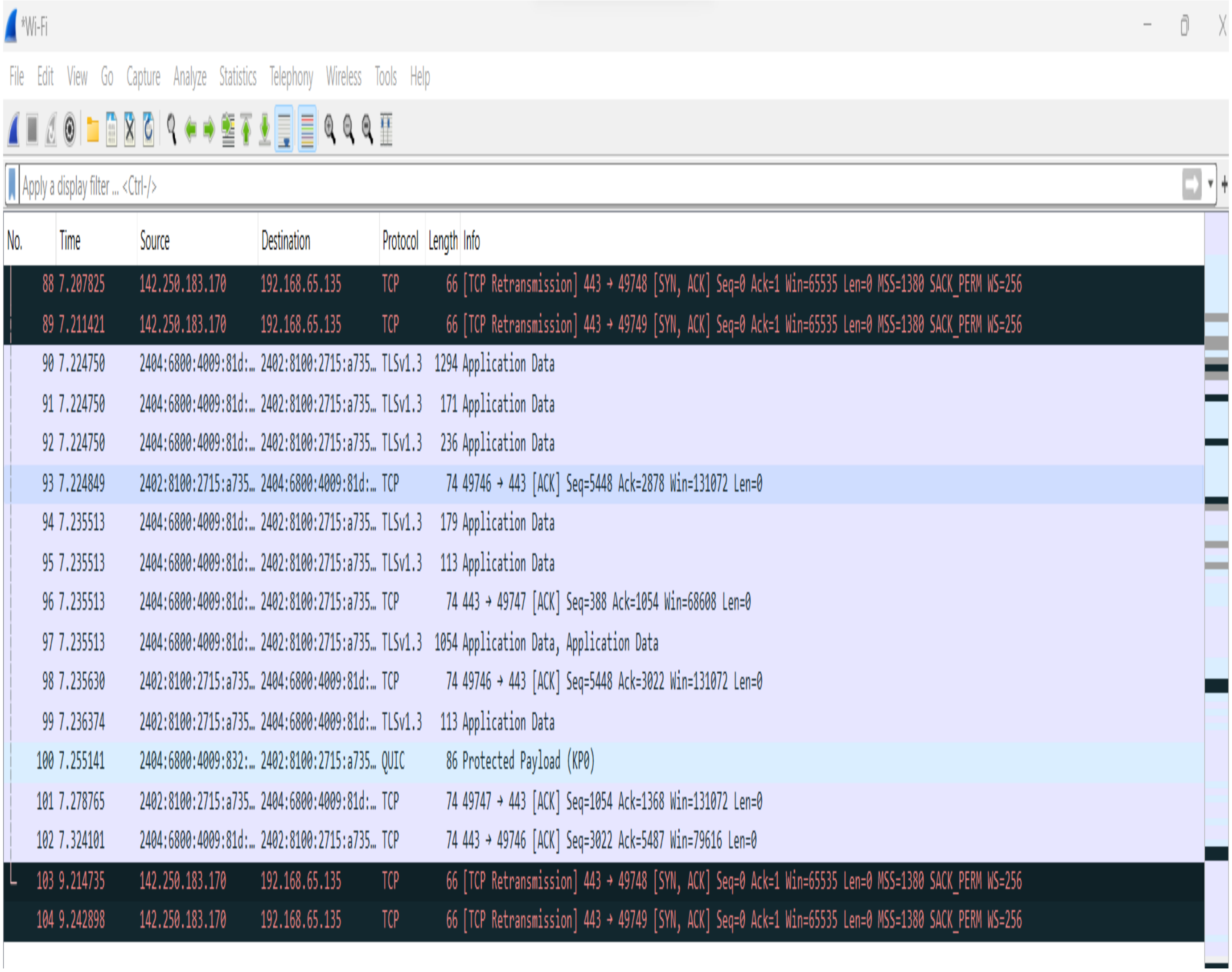
Understanding each protocol's role and structure enables analysts to interpret the information within packets accurately.

This includes analyzing protocol-specific fields, identifying errors or anomalies, and understanding the sequence of events in a communication session.

7. **Practical Application:**

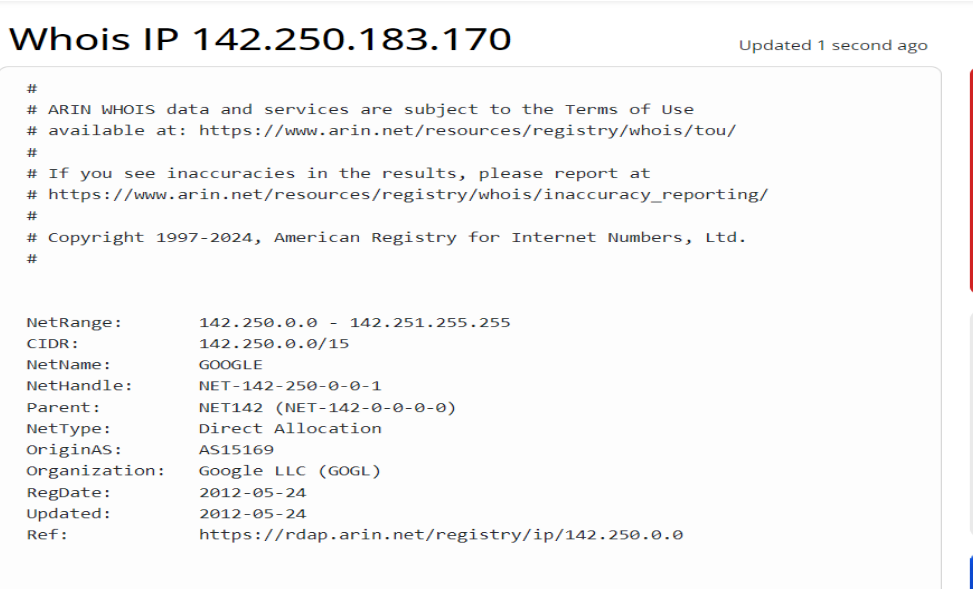
By applying theoretical knowledge of network protocols in Wireshark, practitioners can efficiently troubleshoot network problems, optimize network performance, and enhance security through detailed analysis of network traffic.

Analyzing source & destination IP Address



No.	Time	Source	Destination	Protocol	Length	Info
88	7.207825	142.250.183.170	192.168.65.135	TCP	66	[TCP Retransmission] 443 → 49748 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1380 SACK_PERM WS=256
89	7.211421	142.250.183.170	192.168.65.135	TCP	66	[TCP Retransmission] 443 → 49749 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1380 SACK_PERM WS=256
90	7.224750	2404:6800:4009:81d:…	2402:8100:2715:a735…	TLSv1.3	1294	Application Data
91	7.224750	2404:6800:4009:81d:…	2402:8100:2715:a735…	TLSv1.3	171	Application Data
92	7.224750	2404:6800:4009:81d:…	2402:8100:2715:a735…	TLSv1.3	236	Application Data
93	7.224849	2402:8100:2715:a735…	2404:6800:4009:81d:…	TCP	74	49746 → 443 [ACK] Seq=5448 Ack=2878 Win=131072 Len=0
94	7.235513	2404:6800:4009:81d:…	2402:8100:2715:a735…	TLSv1.3	179	Application Data
95	7.235513	2404:6800:4009:81d:…	2402:8100:2715:a735…	TLSv1.3	113	Application Data
96	7.235513	2404:6800:4009:81d:…	2402:8100:2715:a735…	TCP	74	443 → 49747 [ACK] Seq=388 Ack=1054 Win=68608 Len=0
97	7.235513	2404:6800:4009:81d:…	2402:8100:2715:a735…	TLSv1.3	1054	Application Data, Application Data
98	7.235630	2402:8100:2715:a735…	2404:6800:4009:81d:…	TCP	74	49746 → 443 [ACK] Seq=5448 Ack=3022 Win=131072 Len=0
99	7.236374	2402:8100:2715:a735…	2404:6800:4009:81d:…	TLSv1.3	113	Application Data
100	7.255141	2404:6800:4009:832:…	2402:8100:2715:a735…	QUIC	86	Protected Payload (KP0)
101	7.278765	2402:8100:2715:a735…	2404:6800:4009:81d:…	TCP	74	49747 → 443 [ACK] Seq=1054 Ack=1368 Win=131072 Len=0
102	7.324101	2404:6800:4009:81d:…	2402:8100:2715:a735…	TCP	74	443 → 49746 [ACK] Seq=3022 Ack=5487 Win=79616 Len=0
103	9.214735	142.250.183.170	192.168.65.135	TCP	66	[TCP Retransmission] 443 → 49748 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1380 SACK_PERM WS=256
104	9.242898	142.250.183.170	192.168.65.135	TCP	66	[TCP Retransmission] 443 → 49749 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1380 SACK_PERM WS=256

With the help of www.whois.com ,identify the IP address



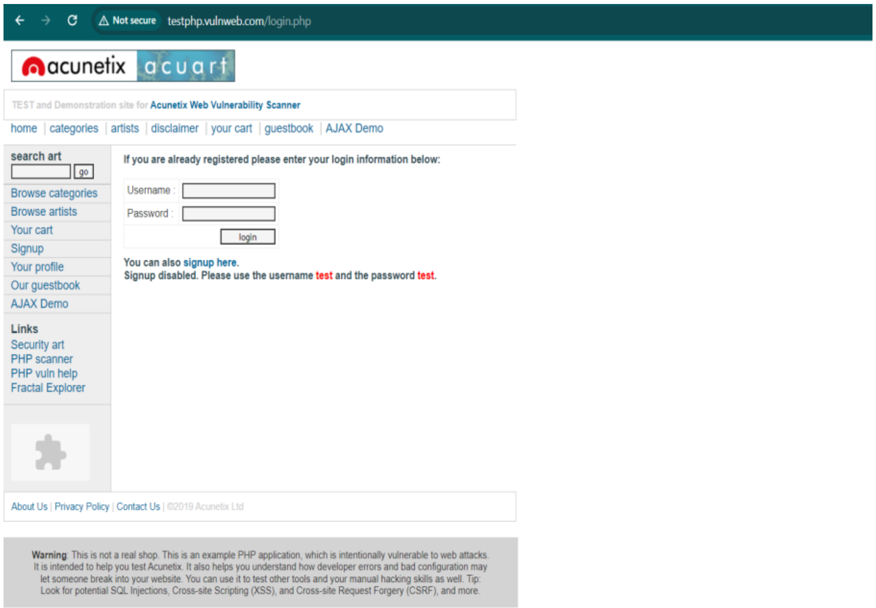
Whois IP 142.250.183.170

Updated 1 second ago

```
#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
#
# Copyright 1997-2024, American Registry for Internet Numbers, Ltd.
#


NetRange:       142.250.0.0 - 142.251.255.255
CIDR:           142.250.0.0/15
NetName:        GOOGLE
NetHandle:      NET-142-250-0-0-1
Parent:         NET142 (NET-142-0-0-0-0)
NetType:        Direct Allocation
OriginAS:       AS15169
Organization:   Google LLC (GOGL)
RegDate:        2012-05-24
Updated:        2012-05-24
Ref:            https://rdap.arin.net/registry/ip/142.250.0.0
```

Case Study of testphp.vulweb.com



Not secure
testphp.vulnweb.com/login.php
acunetix acuart
TEST and Demonstration site for Acunetix Web Vulnerability Scanner
home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo
search art
go
Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo
Links
Security art
PHP scanner
PHP vuln help
Fractal Explorer
login
About Us | Privacy Policy | Contact Us | ©2019 Acunetix Ltd

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

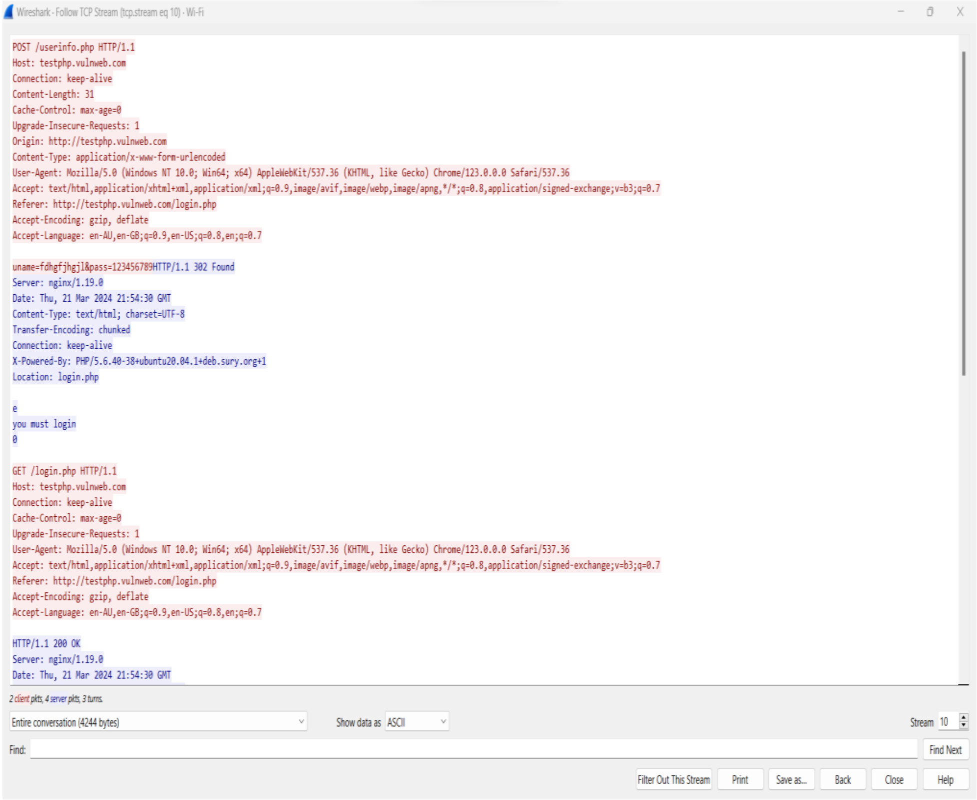
Username :

Password :

You can also signup here.
Signup disabled. Please use the username test and the password test.

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.

Trace the TCP



Wireshark - Follow TCP Stream (tcp.stream eq 10) - Wi-Fi

```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Connection: keep-alive
Content-Length: 31
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://testphp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like 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://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7

uname=fdhgfjhgjl&pass=123456789HTTP/1.1 302 Found
Server: nginx/1.19.0
Date: Thu, 21 Mar 2024 21:54:30 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
Location: login.php

e
you must login
0

GET /login.php HTTP/1.1
Host: testphp.vulnweb.com
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like 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://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7

HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Thu, 21 Mar 2024 21:54:30 GMT
```

2 client pkts, 4 server pkts, 3 turns.

Entire conversation (4244 bytes)   Show data as ASCII   Stream 10

Find:

05 CONCULSION

The conclusion of this practical is to demonstrates the vital role of Wireshark in analyzing web traffic and identifying security vulnerabilities, using the testphp.vulnweb.com website as a case study.