

Experiment No. 9

Title: Network Sniffing - Wireshark

**Batch: A3 Roll No.:16010421073 Experiment No.:9**

**Aim:** To perform network sniffing using wire shark tool

**Resources needed:** Wire shark tool

# Theory

Wireshark is a network packet analyzer. Any network packet analyzer will try to capture network packets and will try to display that packet data as detailed as possible in human readable format. Wireshark is an open source software project, and is released under the GNU General Public License (GPL). We can freely use Wireshark on any number of computers, without worrying about license keys. In addition, all source code is freely available under the GPL. Because of that, it is very easy for people to add new protocols to Wireshark, either as plug-in, or built into the source code. In the past, such tools were either very expensive, proprietary. However, with the advent of Wire-shark, all that has changed. Wireshark is perhaps one of the best open source packet analyzers available today.

**What Wireshark is not......**

Here are some things Wireshark does not provide:

1. Wireshark isn’t an intrusion detection system. It will not warn us when someone does strange things on our network that he/she isn’t allowed to do. However, if strange things happen, Wireshark might help you figure out what is really going on.

2. Wireshark will not manipulate things on the network, it will only "measure" things from it. Wireshark doesn’t send packets on the network or do other active things.

**Applications of Wireshark**:

Here are some applications. Many people use Wireshark for doing following things,

* Network administrators use it to troubleshoot network problems.
* Network security engineers use it to examine security problems (Network Forensics.)
* Developers use it to debug protocol implementations.
* People use it to learn network protocol internals.

Beside these examples Wireshark can be helpful in many other situations too.

**Features of Wireshark:**

The following are some of the many features Wireshark has:

* Available for UNIX and Windows operating systems.
* Capture live packet data from a chosen network interface.
* Open files containing packet data captured with tcpdump/WinDump and a number of other packet capture programs.
* Import packets from text files containing hex dumps of packet data.
* Display packets with very detailed protocol information.
* Save packet data captured.
* Export some or all packets in a number of capture file formats.
* Filter packets on many criteria.
* Search for packets on many criteria.
* Colorize packet display based on filters.
* Create various statistics.

……and a lot more!

Most important menus are : 1) Capture 2) Analyze 3) Statistics

Students are expected to explore all these menus and sub-menus in details.

Wireshark can capture traffic from many different network media types including wireless LAN as well. Which media types are supported, depends on many things like the operating system we are using and the hardware support.

**Physical interfaces supported:**

* ATM - capture ATM traffic
* Bluetooth- capture Bluetooth traffic .
* Cisco HDLC links - capture on synchronous links using Cisco HDLC encapsulation.
* Ethernet- capture on different topologies, including switched networks.
* Framerelay – captures framerelay traffic.
* IrDA capture IrDA traffic - currently limited to Linux.
* PPP links - capture on dial-up lines, ISDN connections and PPP-over-Ethernet (PPPoe, e.g. ADSL)
* Tokenring - capture on Tokenring adapters, promiscuous mode and switched networks
* USB- capture of raw USB traffic
* WLAN- capture on 802.11 (WLAN, Wi-Fi) interfaces, including "monitor mode" , raw 802.11 headers and radio information

**Virtual interfaces :**

* Loopbak - capture traffic from a machine to itself, including the IP address 127.0.0.1
* Pipes - use UNIX pipes to capture from other applications (even remote!)
* VLAN – capture VLAN traffic, including VLAN tags.

**In addition to this, Wireshark can do following things.**

* Import files from many other capture programs.
* Wireshark can open packets captured from a large number of other capture programs.
* Export files for many other capture programs.
* Wireshark can save packets captured in a large number of formats of other capture programs.
* Can be used as a protocol decoder.

**Procedure / Approach /Algorithm / Activity Diagram:**

1. Go to the official website of Wire shark ( www.wireshark.org) and download the stable version of Wire shark for 64 bit windows operating system.
2. After successful installation you will get the blue icon of Wire shark on the desktop.
3. Click on the icon and start the software.
4. Choose an interface and start capturing the packets.
5. Study the packet details of all the protocols.
6. Understand colour code in details.
7. Perform the statistics for a particular protocol. (Every student should perform for different protocol).

# Implementation:

Task1: Design your own registration and login pages (along with user database of registered users)

Task2: Run wire shark and capture the login page request data using wire shark and locate the captured password.

**Questions:**

1. **What is the difference between Burp suite and Wire shark tools?**

**Ans:**

Burp Suite and Wireshark are both popular tools used in the field of cybersecurity and network analysis, but they serve different purposes and have distinct functionalities. Here are the key differences between Burp Suite and Wireshark:

**1. Purpose:**

**- Burp Suite**: Burp Suite is a cybersecurity tool primarily used for web application security testing, including web vulnerability scanning, penetration testing, and security assessment of web applications.

**- Wireshark:** Wireshark, on the other hand, is a network protocol analyzer. It's used for monitoring and analyzing network traffic at the packet level. Wireshark is not specifically designed for web application security testing but for broader network analysis.

**2. Scope:**

**- Burp Suite:** Burp Suite is typically used for assessing the security of web applications, focusing on issues like cross-site scripting (XSS), SQL injection, security misconfigurations, and other web-related vulnerabilities.

**- Wireshark:** Wireshark is used to capture and analyze network traffic, which can be applied to a wide range of network-related tasks, including troubleshooting, debugging, and understanding network protocols.

**3. User Interface:**

**- Burp Suite:** Burp Suite has a user-friendly graphical interface tailored for web application testing. It provides various tools and features for intercept.

1. **Suggest the methods and/or security mechanisms to protect the password being leaked using tools like wireshark.**

**Ans:** Protecting passwords from being leaked when using network analysis tools like Wireshark is crucial for maintaining the security of your systems and data. Here are some methods and security mechanisms to help prevent password leaks in network traffic:

**1. Use HTTPS:** Whenever possible, use HTTPS for your web applications and services. HTTPS encrypts the data transferred between the client and server, including passwords. It ensures that the data, including login credentials, is secure in transit.

**2. Implement Strong Authentication:** Use strong authentication methods, such as multi-factor authentication (MFA). Even if a password is intercepted, an attacker would still need additional authentication factors to gain access.

**3. Hashed Passwords:** Store passwords securely on the server using strong, one-way cryptographic hash functions. Don't store plain text passwords. When a user logs in, hash their input and compare it to the stored hash.

**4. Salting Passwords:** Add a unique salt value to each password before hashing it. Salting ensures that even if two users have the same password, their hashed values will differ, making it harder for attackers to use precomputed tables (rainbow tables) to crack passwords.

**5. Credential Policies:** Enforce strong password policies that require users to choose complex, unique passwords. Educate users about password best practices.

By implementing these measures, you can significantly reduce the risk of passwords being leaked through tools like Wireshark and enhance the overall security of your network and systems.

**Result:** Task 1 implementation code and Task 2 screenshots.

**Code:**

**Index.php**

<!DOCTYPE html>

<html>

<head>

<title>Login Form</title>

<style>

body {

font-family: Arial, sans-serif; background-color: #f2f2f2;

text-align: center;

}

.container {

width: 300px; margin: 0 auto; padding: 20px;

background-color: #fff; border-radius: 5px;

box-shadow: 0 0 5px rgba(0, 0, 0, 0.2);

}

h2 {

color: #333;

}

.input-container { margin: 10px 0;

}

input {

width: 100%; padding: 10px;

border: 1px solid #ccc; border-radius: 5px;

}

button {

width: 100%; padding: 10px;

background-color: #007BFF; color: #fff;

border: none; border-radius: 5px; cursor: pointer;

}

#responseMessage { color: #007BFF;

}

</style>

</head>

<body>

<div class="container">

<h2>Login Form</h2>

<div id="loginForm">

<form action="login.php" method="post">

<div class="input-container">

<input type="text" id="loginUsername" name="loginUsername" placeholder="Username" required>

</div>

<div class="input-container">

<input type="password" id="loginPassword" name="loginPassword" placeholder="Password" required>

</div>

<button type="submit" id="loginButton">Login</button>

</form>

</div>

<p id="responseMessage"></p>

</div>

<script>

document.getElementById('loginButton').addEventListener('click', function () {

var loginUsername = document.getElementById('loginUsername').value;

var loginPassword = document.getElementById('loginPassword').value;

document.getElementById('responseMessage').textContent = "Welcome, " + loginUsername + "!";

});

</script>

</body>

</html>

**Login.php**

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$loginUsername = $\_POST['loginUsername'];

$loginPassword = $\_POST['loginPassword'];

if ($loginUsername === "keyur" && $loginPassword === "password") { echo "Welcome, " . htmlspecialchars($loginUsername) . "!";

} else {

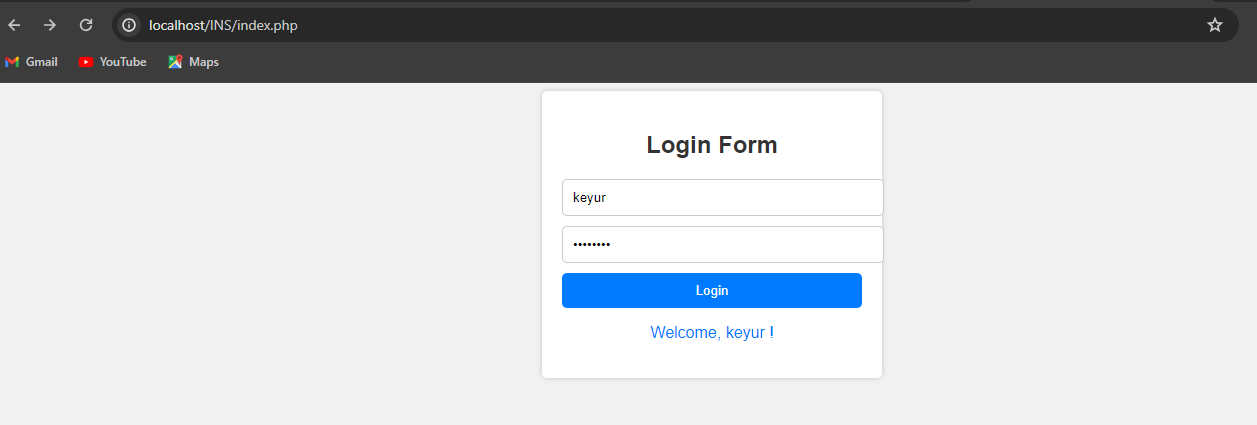
echo "Login failed. Invalid credentials.";

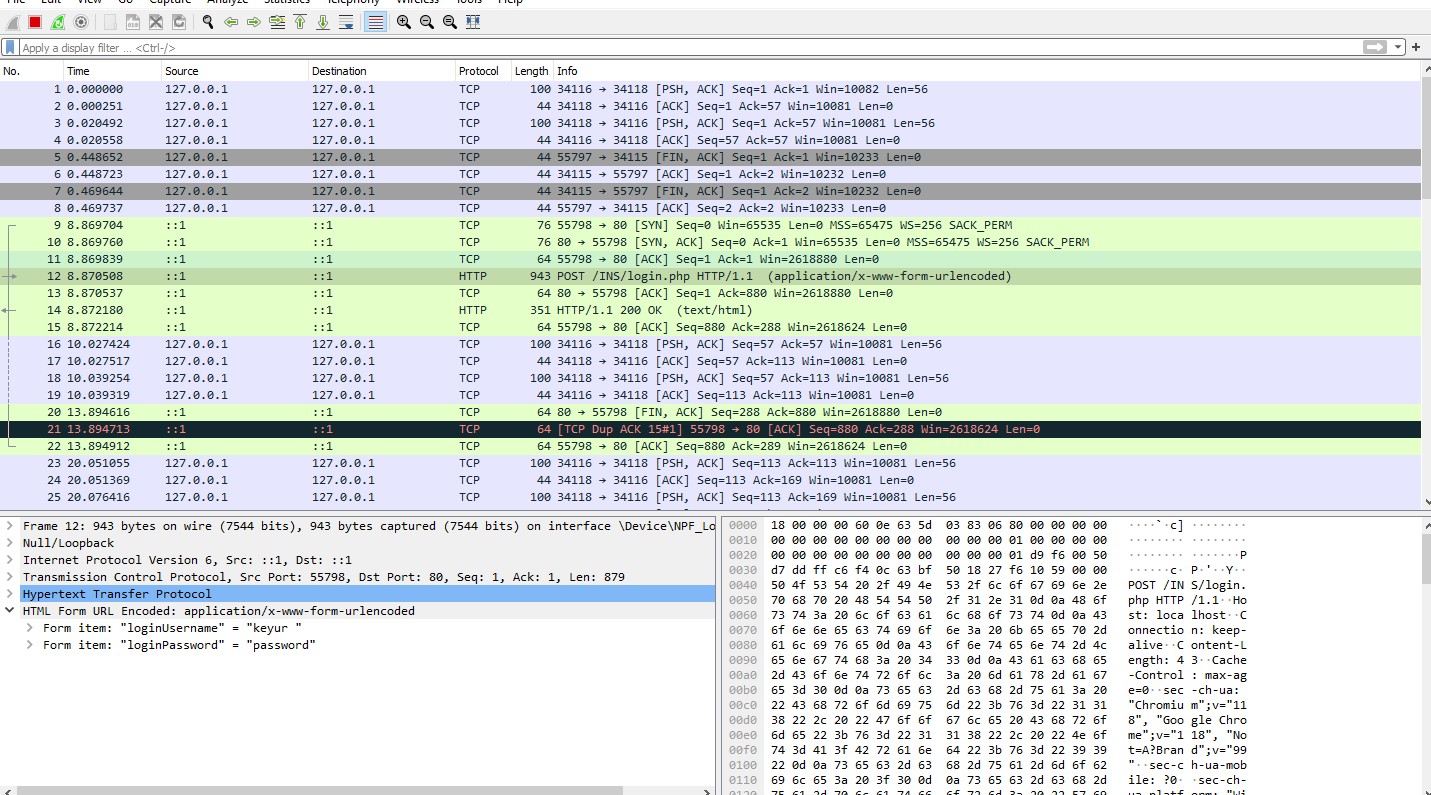
}

}

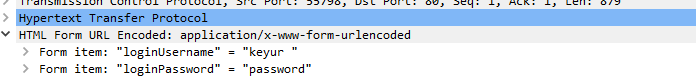
?>

**Output:**









**Outcomes:**

**CO4 :** Understand Security issues related to Software, Web and Networks.

**Conclusion: (**Conclusion to be based on the objectives and outcomes achieved**)**

Thus we performed network sniffing using wireshark tool and captured to login page request data and also password.

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of faculty in-charge with date**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**References:**

**Books/ Journals/ Websites:**

1. [**https://www.wireshark.org/ \_(software)**](https://en.wikipedia.org/wiki/Cain_and_Abel_(software))
2. [**https://en.wikipedia.org/wiki/Wireshark**](https://en.wikipedia.org/wiki/Wireshark)
3. <https://www.wireshark.org/docs/>
4. https://www.youtube.com/watch?v=UBfSgjUCEi0