Objective:

To understand how HTTP sessions are maintained using cookies, focusing on the exchange and usage of cookies between the client and server.

Instructions:

1. **Start Capturing in Wireshark:** Begin by capturing traffic using Wireshark.

First , the server may send a `Set-Cookie` header in its response. This header instructs the browser to store a cookie with specific attributes.

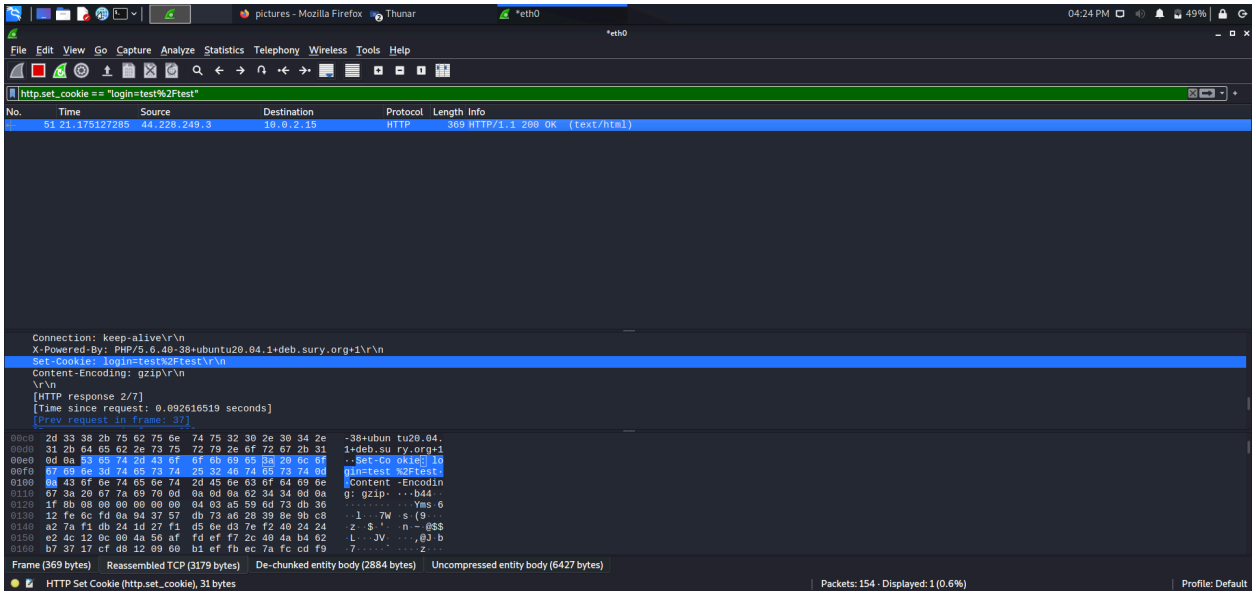
The `Set-Cookie` header typically includes:

1. **Name=Value**: The name-value pair that identifies the cookie.
2. **Expires/Max-Age**: Defines the expiration date or maximum age of the cookie.
3. **Domain**: Specifies the domain for which the cookie is valid.
4. **Path**: Specifies the URL path for which the cookie is valid.
5. **Secure**: Indicates that the cookie should only be sent over HTTPS connections.
6. **HttpOnly**: Restricts access to the cookie from JavaScript, enhancing security.

In the below image we see that there is a set cookie in http protocol

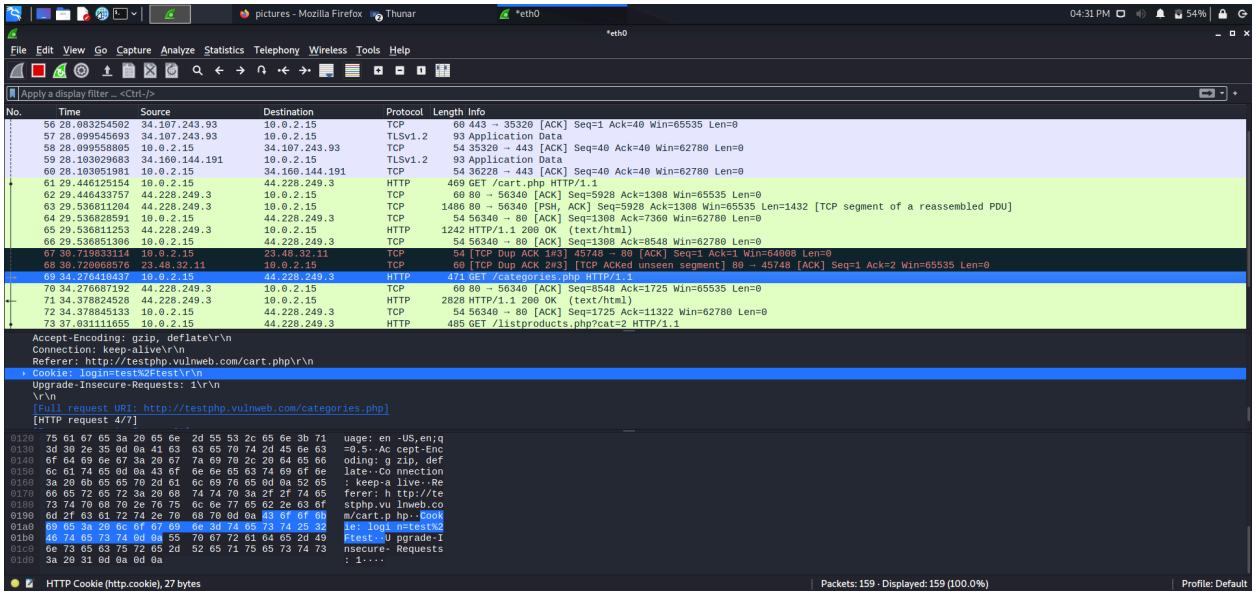
set- cookie: login= test %2Ftest.

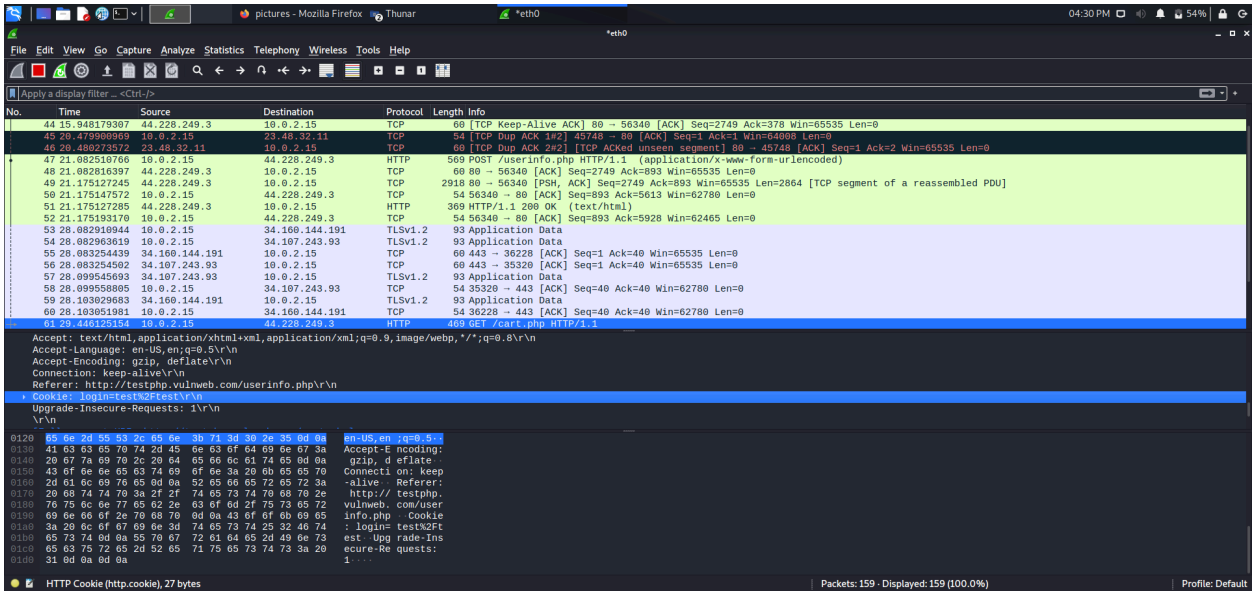
In some web browsers there will be session_id with a value.



1. On subsequent requests while I was navigating the webserver test.vulnweb.com to check in cart and various options in the same domain, the browser included the cookie in the `Cookie` header. This allows the server to recognize the client and maintain session continuity.

2. The `Cookie` header contains all cookies associated with the request's domain and path.
3. In the below images we see that there is Cookie header. This is the session where “categories” option is captured.





In the above image the wireshark analysis is for “cart” in the web browser.

Summary - Upon successful login, the server creates a session ID and sends it to the client in a cookie.
Subsequent Requests: For every request to the server, the browser includes the session ID in the `Cookie` header, allowing the server to recognize the user and maintain their logged-in state.
Session Expiry: If the session ID cookie has an expiration time, the user will be logged out when the cookie expires. The server can also invalidate the session upon user logout or after a period of inactivity.

Why are cookies and session persistence necessary -
By setting cookies with the `Set-Cookie` header, servers can instruct browsers to store session identifiers and other relevant data. Subsequent requests include this data in the `Cookie` header, allowing servers to manage user sessions effectively. Cookies provide a mechanism to bridge this gap by allowing stateful interactions over the stateless HTTP protocol.

Managing Session State

Session Identification: Cookies often store a session identifier (session ID) that is unique to each user session. This ID maps to a session record on the server, allowing the server to keep track of user-specific data such as authentication status, user preferences, and shopping cart contents.

Session Persistence:

1. **Session Lifetime**: Cookies can be configured with an expiration time. Session cookies (without an expiration time) are deleted when the browser is closed, while persistent cookies remain until their expiration time is reached.
2. **Session Cookies**: Useful for temporary data that should not persist beyond the browser session.
3. **Persistent Cookies**: Useful for data that needs to be retained across multiple sessions or visits.

How are cookies used to maintain session state?
Set-Cookie Header: When a user first visits a website, the server sends a `Set-Cookie` header in the HTTP response. This instructs the client (browser) to store a cookie.

set- cookie: login= test %2Ftest. : The name and value of the cookie.
Path=/: The URL path for which the cookie is valid.
HttpOnly: Prevents JavaScript from accessing the cookie, mitigating certain types of attacks.
Secure: Ensures the cookie is only sent over HTTPS connections.
Max-Age=3600: Specifies the cookie's expiration time in seconds.

Client-Side Storage: The browser stores this cookie in a cookie jar associated with the domain that set it. The cookie is stored in memory and on disk based on the attributes defined .