Session Replay Attack

https://campus.barracuda.com/doc/49058327/

Description

In a Session Replay Attack, an attacker steals a valid session ID of a user, and reuses it to impersonate an authorized user to perform fraudulent transactions/activities. Users can become victims of Session Replay Attacks when session IDs do not have a set session expiration time or the session data is stored in unencrypted form. Web applications that allow reusing old session IDs or session credentials for authorization are also vulnerable to Session Replay Attacks.

A session ID is a unique number assigned to identify a user accessing a web application. The session ID can be in the form of cookies or IDs in the parameter values. When the user is authorized to access a web application, a session ID is created for that user. It is important to maintain the confidentiality of a session ID so other users or attackers do not use it to access the same account. Some web applications allow replaying (reusing) the old session ID to access the resources, without re-authenticating the user. If the session ID is stolen, the attacker can use it to masquerade as an authorized user.

Effects

If the session ID is stolen, the attacker can:

- Use the session ID to keep track of a user.
- View the user’s account details.
- View the user’s account and perform fraudulent transactions or activities.

Method

The following parameters should be sanitized properly before processing the request:

- Cookies
- URLs
- Form Parameters (GET and POST method)

Example

Web applications maintaining sessions in a request parameter:

A web application might maintain a user's session based on the value of a parameter in the request, for example: `http://example.com/home/show.php?SESSIONID=MYSESSION`, where `MYSESSION` is the session ID. Unprotected, this method is vulnerable to a specific type of Session Replay attack, called Session Fixation attack, as in this example:

1. The attacker creates their own session ID, e.g., `ATTACKER-SESSION`.
2. The attacker sends a URL with their session ID to a valid user of the application, e.g., `http://example.com/home/show.php?SESSIONID=ATTACKER-SESSION`.
3. When the valid user clicks the link, a session is started with the session ID, `ATTACKER_SESSION`.
4. The valid user logs into the application with their credentials.
5. The attacker can now impersonate the valid user by going to `http://example.com/home/show.php?SESSIONID=ATTACKER-SESSION`.

How to Prevent

To mitigate session replay attacks:

- Set the web application to invalidate a session after it exceeds the predefined idle timeout, and after the user logs out.
- Set the lifespan for the session to be as short as possible.
- Encrypt the session data.

You can prevent the session replay attacks by configuring Session Timeout for web applications and cookie security on the Barracuda Web Application Firewall.