Cookie Replay Attack

https://campus.barracuda.com/doc/42049419/

Description

A cookie replay attack occurs when an attacker steals a valid cookie of a user, and reuses it to impersonate that user to perform fraudulent or unauthorized transactions/activities.

Effects

After stealing a cookie, an attacker can effectively impersonate the user as long as the cookie remains valid. Attackers can do anything that users can within their accounts.

Methods

Cookies can be stolen:

- **In transit** – When SSL is not used and a man-in-the-middle is able to snoop on the traffic.
- **Directly from client machines** – Using a secondary vector like XSS or malware.

Examples

There are many ways session hijacking can be performed.

Example 1: Using session cookies issued to the user by the server.

In this scenario, there is a social networking site with a valid user logged in, and the server issued a session cookie, SESSION-ID, to the user. If the SESSION-ID is the cookie that identifies the session of the user, the attacker can use the SESSION-ID cookie value to log in as the valid user.

The attacker can perform cross-site scripting or other technique to steal the cookie from the victim's browser. If the attacker steals the cookie SESSION-ID=User-abc-logged-in-2341785645, they can use the cookie with the following request to post a status `I am hacked!!!!!!` in the victim's home page:

POST /home/post_status.php HTTP/1.1

Host: www.social-site.com

Cookie: SESSION-ID=User-abc-logged-in-2341785645

Content-Length:38

Content-Type:application/x-www-form-urlencoded

Status=I am hacked!!!!!!&Submit=submit

The attacker uses the cookie issued to the authorized user, and gains control on the user's session.

In this example, the server failed to check the client IP address and browser information in the request, which led to cookie and session hijacking.

Example 2: Guessing the cookie values of users if a complicated algorithm is not used for the cookie generation

In this scenario, a social networking website (`www.socialnetworking-site.com`) uses an algorithm to generate cookies for the users. If the user name is `John`, then the cookie generated for the user could be `LOGINID=1322020-iknpgimo`. In this case, the algorithm used to generate the cookie can be as follows: first part of the cookie is the date (i.e., 13/2/2020), and second part is the combination of the previous and following alphabet letter for each letter of the username `John` (i.e., the previous letter for `J` is `I` and the following letter is `k`). After cracking the algorithm, the hacker can guess the cookies of any number of users and hack their sessions.

To take over the session of a user named Albert, the hacker can create a cookie as `LOGINID =1322015-zbkmacdfqssu`, log into Albert's session, and post a status on Albert's account.

POST /Status/post.asp HTTP/1.1

Host: www.another-social-site.com

Cookie:LOGINID =1322015-zbkmacdfqssu

Content-Length:45

Content-Type:application/x-www-form-urlencoded

Todays_status=I am hacked!!!!!!&Submit=submit

Prevention

To mitigate cookie replay attacks, a web application should:

- 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.
- Have a mechanism to detect when a cookie is seen from multiple clients

You can prevent cookie replay attacks by configuring **Cookie Protection** policies on the Barracuda Web Application Firewall. Among other things, this can also detect when a cookie is seen from multiple IP addresses and allows mitigating controls when this happens. See How to Secure HTTP Cookies for more information.