CSRF (Cross-Site Request Forgery) Explained

What is CSRF (Cross-Site Request Forgery)?

CSRF is an attack where a malicious site causes a user's browser to perform an unwanted action on a trusted site where the user is authenticated.

Example:

- You are logged into your bank account.
- Then you visit a malicious website.
- That site sends a request to your bank using your browser session, for example:
  GET https://www.mybank.com/transfer?amount=1000&toAccount=attacker
- The bank believes the request is from you since you're already logged in.

How a CSRF Attack Works:

1. The user is logged into a site (e.g., Bank, Facebook).
2. Session cookie is stored in the browser.
3. User visits a malicious website.
4. That site sends a forged request to the trusted site using the user's session.
5. The server processes the request, assuming it’s a legitimate action.

CSRF Example:

A malicious site might include this HTML code:

html

******

When the user visits that page, their browser automatically makes a GET request to the bank.

Methods to Prevent CSRF:

1. **CSRF Tokens**
    - Send a unique token with each request.
    - Server validates the token.
2. **SameSite Cookies**
    - Use SameSite=Strict to limit cross-site cookie usage.
3. **Referer Header Validation**
    - Check the origin of incoming requests.
4. **User Confirmation**
    - Require password re-entry or action confirmation for sensitive requests.

CSRF vs XSS:

CSRF	**XSS**
Uses victim's session to send forged requests	Injects client-side code (JavaScript) to execute
Victim triggers the action	Victim's browser executes malicious code
Mostly server-side requests	Mostly client-side scripts

Short Summary:

CSRF = Cross-Site Request Forgery
Uses a logged-in user's session to perform unwanted actions without their consent.