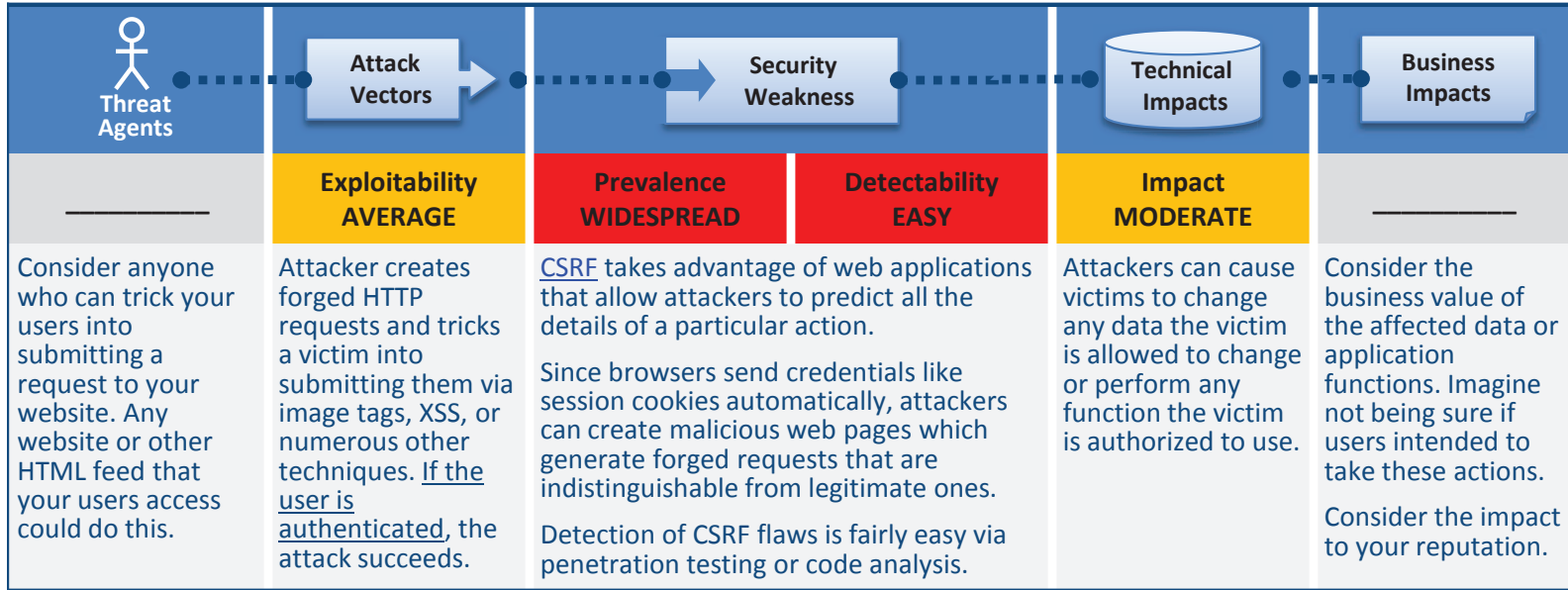


# Cross-Site Request Forgery (CSRF)



## Am I Vulnerable to CSRF?

The easiest way to check whether an application is vulnerable is to see if each link and form contains an unpredictable token for each user. Without such an unpredictable token, attackers can forge malicious requests. Focus on the links and forms that invoke state-changing functions, since those are the most important CSRF targets.

You should check multistep transactions, as they are not inherently immune. Attackers can easily forge a series of requests by using multiple tags or possibly JavaScript.

Note that session cookies, source IP addresses, and other information that is automatically sent by the browser doesn't count since this information is also included in forged requests.

OWASP's [CSRF Tester](#) tool can help generate test cases to demonstrate the dangers of CSRF flaws.

## How Do I Prevent CSRF?

Preventing CSRF requires the inclusion of an unpredictable token in the body or URL of each HTTP request. Such tokens should at a minimum be unique per user session, but can also be unique per request.

1. The preferred option is to include the unique token in a hidden field. This causes the value to be sent in the body of the HTTP request, avoiding its inclusion in the URL, which is subject to exposure.
2. The unique token can also be included in the URL itself, or a URL parameter. However, such placement runs the risk that the URL will be exposed to an attacker, thus compromising the secret token.

OWASP's [CSRF Guard](#) can be used to automatically include such tokens in your Java EE, .NET, or PHP application. OWASP's [ESAPI](#) includes token generators and validators that developers can use to protect their transactions.

## Example Attack Scenario

The application allows a user to submit a state changing request that does not include anything secret. Like so:

```
http://example.com/app/transferFunds?amount=1500
&destinationAccount=4673243243
```

So, the attacker constructs a request that will transfer money from the victim's account to their account, and then embeds this attack in an image request or iframe stored on various sites under the attacker's control.

```

```

If the victim visits any of these sites while already authenticated to example.com, any forged requests will include the user's session info, inadvertently authorizing the request.

## References

### OWASP

- [OWASP CSRF Article](#)
- [OWASP CSRF Prevention Cheat Sheet](#)
- [OWASP CSRFGuard - CSRF Defense Tool](#)
- [ESAPI Project Home Page](#)
- [ESAPI HTTPUtilities Class with AntiCSRF Tokens](#)
- [OWASP Testing Guide: Chapter on CSRF Testing](#)
- [OWASP CSRFTester - CSRF Testing Tool](#)

### External

- [CWE Entry 352 on CSRF](#)