Cheatography

CSRF and Logic Flaws Cheat Sheet

by binca via cheatography.com/44948/cs/13462/

Summary

Similar to XSS because the victim is submitting an attacker crafted request.

Difference is CSRF uses **static content**.

The victim must be logged into the vulnerable application.

The victim's authorized browser sends the transaction.

The **same-origin policy** matches the original login and subsequent cookies, the same protocol, name and port.

The vulnerable app doesn't know the link originated from another source.

OWASP suggests **(CSRF) synchronizer tokens** which prevent CSRF by requiring a secure random token for any state change operation.
Token characteristics:
Unique per user session
Larger random value
Generated by a cryptographically secure random number generator

CSRF Example

1. Attacker researches target application to find CSRF flaw, often through a transaction that uses weak anti-CSRF protection (such as checking the referer) or does not require a dynamic element.

2. Attacker sends crafted link to the victim.

3. Victim logs into the vulnerable app and then in another tab opens crafted link.

4. The malicious link enables the attacker to attack the vulnerable app; transferring funds, changing passwords, or other actions.

ZAP Anti-CSRF Test Form

CSRF testing functionality via "Generate Anti-CSRF Test Form"

Automate POC code creation to discover CSRF flaws via POST

Implementation:
Find possibly vulnerable POST
Right-click POST in ZAP and select "Generate Anti-CSRF Test FORM
Submit

If the form submits successfully the app is vulnerable.

To send a ZAP CSRF Test Form without a referer, save as an HTML file and there will be no referer set.

Note: When using ZAP Anti-CSRF Test Form should be authenticated in vulnerable app.

Logic Attacks

Business logic is found within client-code, which the execution of is controlled by the client. The purpose is to call functionality on server.

Attacker **manually calls** functions in a different order, since many automated tools lack this functionality.

Example Normal Logic:
1. Add item to cart
2. Total cost
3. Authorize card
4. Check out

Example of Attacker Logic:
1. Total Cart
2. Authorize credit card
3. Add items
4. Check out

Because the app stores the state of each step the attacker is able to call steps out of order circumventing paying.

Discovering Logic Flaws is a manual process because most tools only test functionality.

Application mapping is crucial to discovery.

Because logic is integral to architecture it is harder to fix.