
ZROC102



Module 3: Testing Common Web Application Vulnerabilities (4) – CSRF

CSRF (CROSS-SITE REQUEST FORGERY)

What is CSRF?

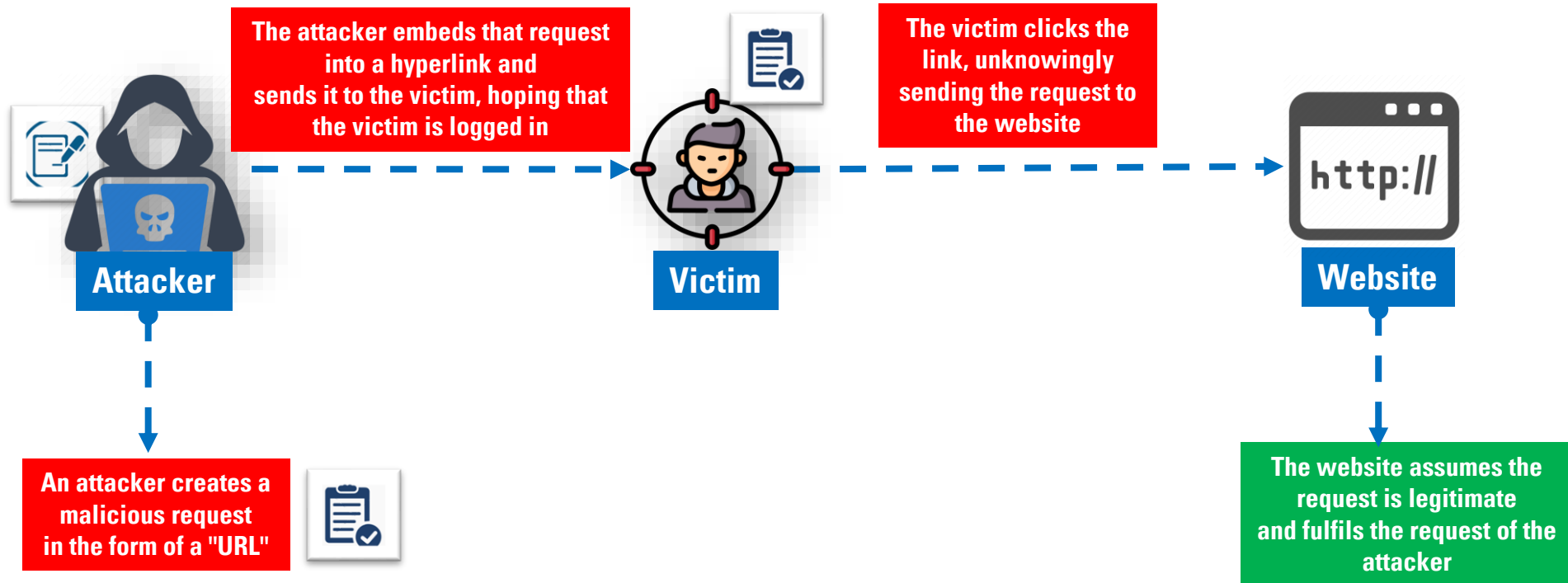
Cross-Site Request Forgery (also known as CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated.

With the attacker using social engineering (such as sending a link via email or chat), an attacker may trick the users of a web application into executing actions of the attacker's choosing. If the victim is a normal user, a successful CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address, and so forth. If the victim is an administrative account, CSRF can compromise the entire web application.

Overview

For most sites, browser requests automatically include any credentials associated with the site, such as the user's session cookie, IP address, Windows domain credentials, and so forth. Therefore, if the user is currently authenticated to the site, the site will have no way to distinguish between the forged request sent by the victim and a legitimate request sent by the victim.

HOW CSRF WORKS



CSRF (EXAMPLE)

Example 1:

Alice wishes to transfer \$100 to **Bob** using the bank.com web application that is vulnerable to CSRF. **Empress**, an attacker, wants to trick Alice into sending the money to Empress instead. The attack will comprise the following steps:

- Building an exploit URL or script
- Tricking Alice into executing the action with Social Engineering

GET scenario

If the application was designed to primarily use GET requests to transfer parameters and execute actions, the money transfer operation might be reduced to a request like:

```
GET http://bank.com/transfer.do?acct=BOB&amount=100 HTTP/1.1
```

Empress now decides to exploit this web application vulnerability using **Alice** as the victim. **Empress** first constructs the following exploit URL which will transfer \$100,000 from Alice's account to **Empress's** account. The attacker takes the original command URL and replaces the beneficiary's name with herself, raising the transfer amount significantly at the same time:

```
http://bank.com/transfer.do?acct=Empress&amount=100000
```

CSRF (EXAMPLE) CONTD.

The social engineering aspect of the attack tricks Alice into loading this URL when Alice is logged into the bank application. This is usually done with one of the following techniques:

- Sending an unsolicited email with HTML content
- Planting an exploit URL or script on pages that are likely to be visited by the victim while they are also doing online banking

The exploit URL can be disguised as an ordinary link, encouraging the victim to click it:

```
<a href="http://bank.com/transfer.do?acct=MARIA&amount=100000">View my Pictures!</a>
```

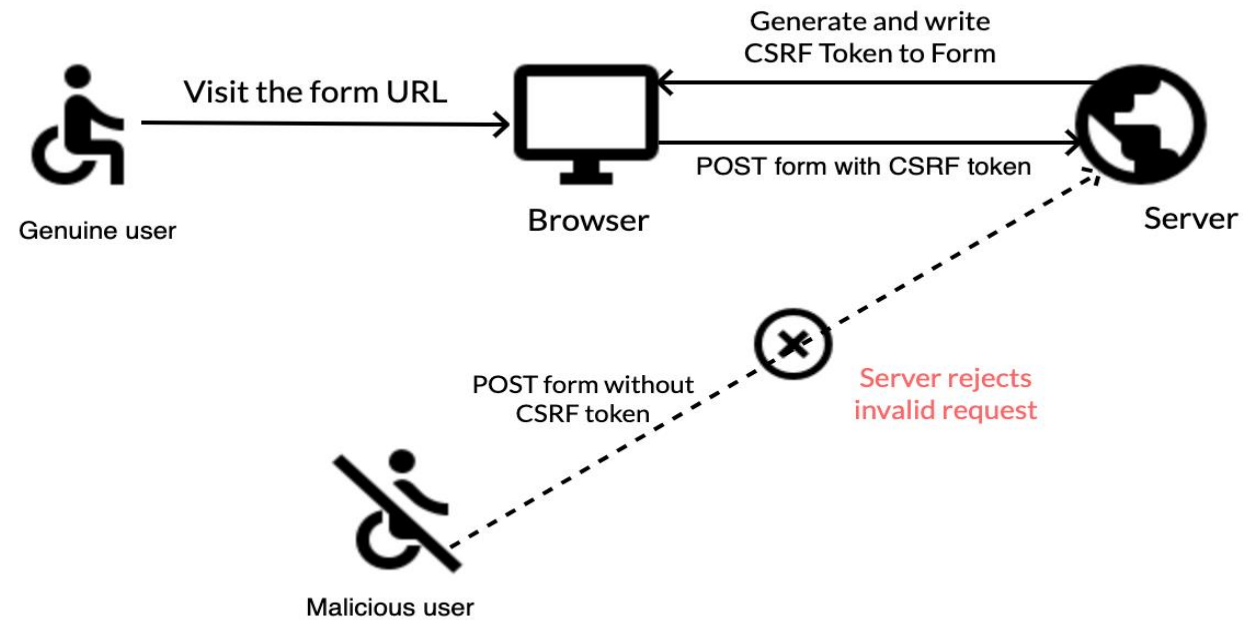
The browser will still submit the request to bank.com

CSRF MITIGATIONS

Steps to be taken?

- ☐ **REST:** Follow these design principles and assign actions to different HTTP verbs. A hacker can't use a **GET** request to grab data, as it will be specific to viewing.
 - ☐ **Anti-forgery tokens:** The server should request this code for any subsequent requests, and if it's not there, it should drop those requests.
 - ☐ **SameSite Cookie:** This Google-created anti-CSRF tool can help block cookies sent along with requests from third parties.
 - ☐ **Verify:** The user will confirm login details before taking a sensitive action.
-

CSRF PROTECTION EXPLAINED

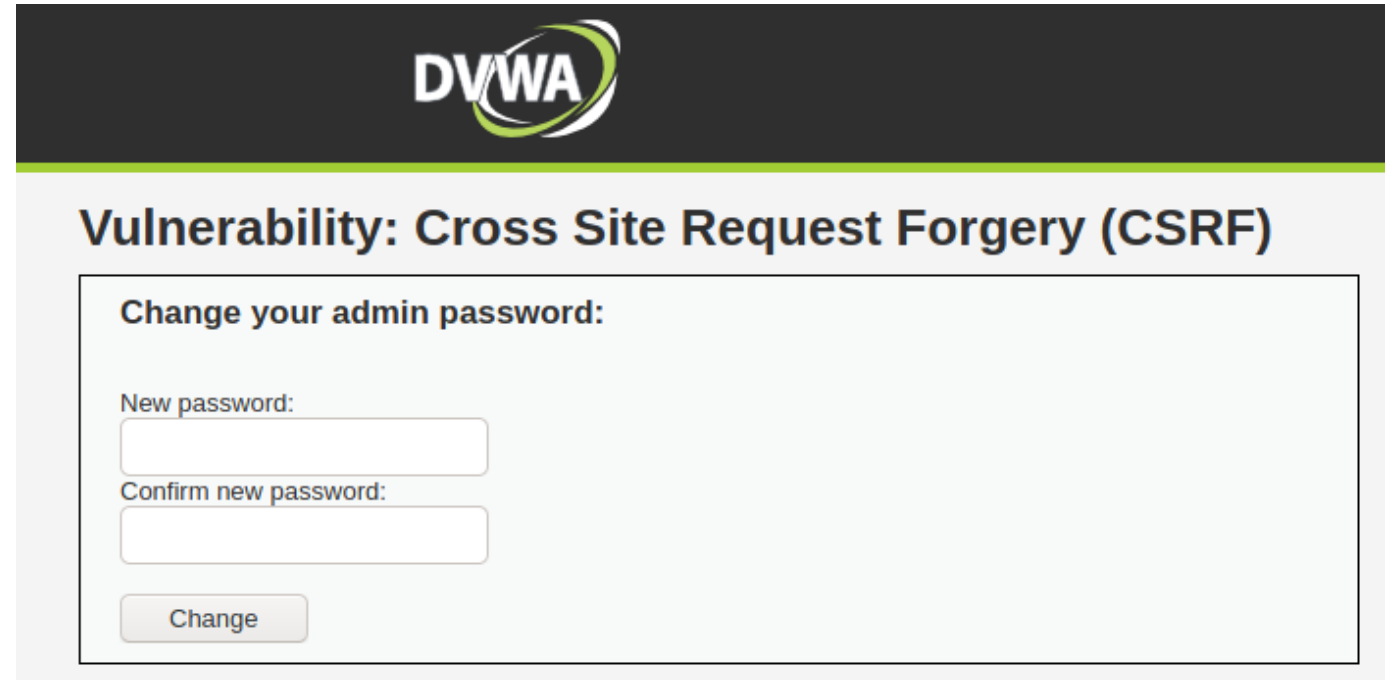


DVWA (CRSF) - PRACTICAL

Practical (Hands-on)

URL:

<http://127.0.0.1:445/vulnerabilities/csrf/>



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. At the top, there is a dark header with the DVWA logo. Below the header, the page title is "Vulnerability: Cross Site Request Forgery (CSRF)". The main content area is a light gray box with a black border. Inside this box, the text "Change your admin password:" is displayed. Below this text, there are two input fields: "New password:" and "Confirm new password:". At the bottom of the box, there is a "Change" button.

DVWA

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:

Confirm new password:

Change

SOURCES

CSRF

<https://owasp.org/www-community/attacks/csrf>

<https://portswigger.net/web-security/csrf>
