Computer System Security (TCS591)

B. Tech CSE V Semester

Instructor:

Dr. Mohammad Wazid

Professor, Department of CSE

Head of Cyber security and IoT research group

Graphic Era (Deemed to be University), Dehradun, India

Email: wazidkec2005@gmail.com

Homepage: https://sites.google.com/site/mwazidiiith/home

Cross site request forgery attack

Overview

- Cross site request forgery (CSRF), also known as XSRF, Sea Surf or Session Riding, is an attack that tricks a web browser into executing an unwanted action in an application to which a user is logged in.
- It can result in unauthorized fund transfers, changed passwords and data theft (i.e., stolen session cookies).
- It is conducted using malicious social engineering, such as an email or link that tricks the victim into sending a forged request to a server.
- As the unsuspecting user is authenticated by their application (i.e., online banking) at the time of the attack, it’s impossible to distinguish a legitimate request from a forged one.

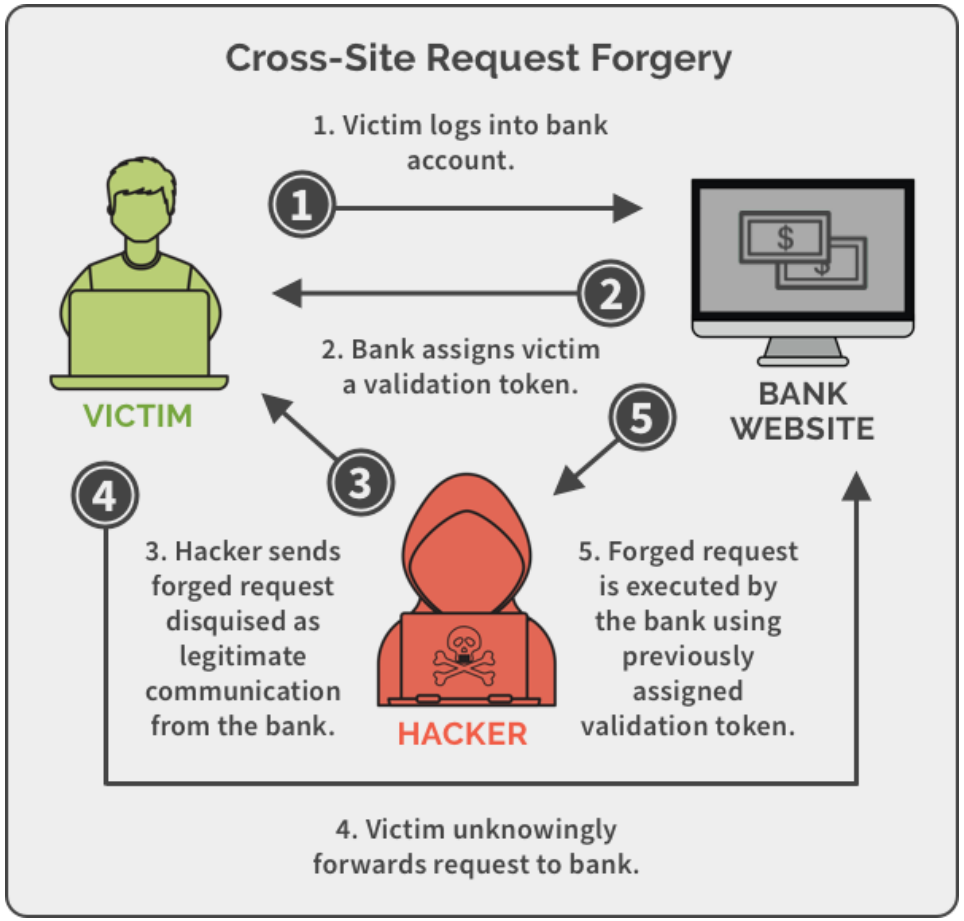

Fig. Steps involved in CSRF
Image source: spanning.com

Example

- Before executing an assault, a perpetrator typically studies an application in order to make a forged request appear as legitimate as possible.
- For example, a typical GET request for a $100 bank transfer might look like:

GET http: //netbank.com /transfer.do? acct=PersonB&amount=$100 HTTP/1.1

Example

- A hacker can modify this script so it results in a $100 transfer to their own account. Now the malicious request might look like:

GET http:// netbank.com/ transfer.do?acct=AttackerA&amount=$100 HTTP/1.1

- A bad actor can embed the request into an innocent looking hyperlink:

Read more!

Example

- Next, he can distribute the hyperlink via email to a large number of bank customers. Those who click on the link while logged into their bank account will unintentionally initiate the $100 transfer.

CSRF Protection

- There are two primary approaches to prevent Cross-site Request Forgery (CSRF)

1. Synchronizing the cookie with an anti-CSRF token that has already been provided to the browser or

2. Preventing the browser from sending cookies to the web application in the first place.

- The details of these methods are as follows.

CSRF Protection

1. Anti-CSRF Tokens

- Anti-CSRF token, sometimes referred to as a synchronizer token or just simply a CSRF token.
- When a user submits a form or makes some authenticated request that requires a cookie, a random token should be included in the request.
- The web application will then verify the existence and correctness of this token before processing the request.
- If the token is missing or incorrect, the request can be rejected.

CSRF Protection

1. Anti-CSRF Tokens

- ***The characteristics of a well designed anti-CSRF system involve the following attributes:***
- The anti-CSRF token should be unique for each user session.
- The session should automatically expire after a suitable amount of time.
- The anti-CSRF token should be a cryptographically random value of significant length.

CSRF Protection

1. Anti-CSRF Tokens

- The anti-CSRF token should be cryptographically secure, that is, generated by a strong pseudo-random number generator (PRNG) algorithm.
- The server should reject the requested action if the anti-CSRF token fails validation.
- **If attacker tries to send the request using the stolen cookie then he/she will not be validated by the server. Because anti-CSRF token uses random number and in each session it is different. So the validation of attacker will be failed.**

CSRF Protection

2. SameSite Cookies

- The SameSite cookie attribute is a new attribute that can be set on cookies to instruct the browser to disable third-party usage for specific cookies.
- The SameSite attribute is set by the server when setting the cookie and requests the browser to only send the cookie in a first-party context.

CSRF Protection

2. SameSite Cookies

- Therefore, request has to be originated from the same origin.
- Means requests made by third-party sites (i.e., attacker site) will not include the SameSite cookie.
- Then that server (i.e., online banking server) will not accept the request comes from attacker site.
- This effectively eliminates Cross-site Request Forgery attacks.

Best practices for CSRF protection

- Logging off web applications (i.e., online banking) when not in use.
- Securing usernames and passwords (try to update them regularly).
- Not allowing browsers to remember passwords.
- Avoiding simultaneously browsing while logged into an application (i.e., online banking). First finish your banking work and then logout and then start browsing for other websites.

References

1. CSRF protection, information available at: https://www.acunetix.com/blog/ articles/cross-site-request-forgery/

2. CSRF procedure, information available at:

https://spanning.com/blog /cross-site-forgery-web-based-application-security-part-2/

3. CSRF procedure, information available at:

https:// www.imperva.com/learn/ application-security/csrf-cross-site-request-forgery/

4. Textbook: Security in Computing, 5th Edition by C. P. Pfleeger, S. L. Pfleeger, J. Margulies