<https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)>

<https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)_Prevention_Cheat_Sheet>

<https://cwe.mitre.org/data/definitions/352.html>

<https://www.owasp.org/index.php/Reviewing_code_for_Cross-Site_Request_Forgery_issues>

* Overview
  + Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated. CSRF attacks specifically target state-changing requests, not theft of data, since the attacker has no way to see the response to the forged request. With a little help of 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.
* Introduction
  + CSRF is not the same as XSS (Cross Site Scripting), which forces malicious content to be served by a trusted website to an unsuspecting victim. Injected text is treated as executable by the browser, hence running the script. Used in Phishing, Trojan upload, Browser vulnerability weakness attacks…..
  + Cross-Site Request Forgery (CSRF) (C-SURF) (Confused-Deputy) attacks are considered useful if the attacker knows the target is authenticated to a web based system. They only work if the target is logged into the system, and therefore have a small attack footprint. Other logical weaknesses also need to be present such as no transaction authorization required by the user.
  + In effect CSRF attacks are used by an attacker to make a target system perform a function (Funds Transfer, Form submission etc..) via the target’s browser without the knowledge of the target user, at least until the unauthorized function has been committed. A primary target is the exploitation of “ease of use” features on web applications (One-click purchase), for example.
* Description
  + Csrf là kĩ thuật tấn công lừa người dùng gửi request độc hại. Hacker lợi dụng danh tính và đặc quyền của user để thực hiện các chức năng không mong muốn. Trình duyệt gửi request tự động kèm theo các xác thực liên quan tới website như session cookie, địa chỉ IP,… Từ đó nếu user vẫn đang được xác thực bởi website đó thì website sẽ không phân biệt được đâu là request giả mạo và đâu là request hợp lệ do user gửi đi.
  + Attacker sẽ nhằm vào các request làm thay đổi trạng thái của user trên server như thay đổi email,password,chuyển khoản, thanh toán cái gì đó.
  + Attacker sẽ build ra một URL để khai thác. Sau đó gửi tới cho user, khi user click vào URL đó thì request độc hại sẽ tự động được gửi đi mà người dùng có thể sẽ ko biết là mình vừa gửi request.
  + URL có thể được giấu sau các thẻ html như img, iframe, … mà user có thể không để ý và click vào đó.
* Prevention measures that do NOT work

https://www.owasp.org/index.php/Cross-Site\_Request\_Forgery\_(CSRF)

* Examples
* Demo
  + <https://www.mojomarketplace.com/>
  + <https://www.exploit-db.com/exploits/42648/>
  + <http://www.websecgeeks.com/2016/05/web2py-vulnerabilities-2145-lfixsscsrf.html>
  + <http://www.websecgeeks.com/2014/06/referrer-csrf-bypass-not-effective-but.html>
  + <https://scotthelme.co.uk/csrf-is-dead/>
  + <http://blog.mazinahmed.net/2015/06/facebook-messenger-multiple-csrf.html>
  + <https://hethical.io/paypal-bug-bounty-updating-the-paypal-me-profile-picture-without-consent-csrf-attack/>
  + <https://whitton.io/articles/messenger-site-wide-csrf/>
  + <https://github.com/ngalongc/bug-bounty-reference>
  + <https://superevr.com/blog/2011/csrf-flash-307-redirect-game-over>
  + <https://www.owasp.org/images/2/2d/FLASH_Security.pdf>
  + <https://media.blackhat.com/eu-13/briefings/Lundeen/bh-eu-13-deputies-still-confused-lundeen-wp.pdf>
  + <https://zseano.com/tutorials/5.html>
  + <https://www.slideshare.net/0ang3el/neat-tricks-to-bypass-csrfprotection>