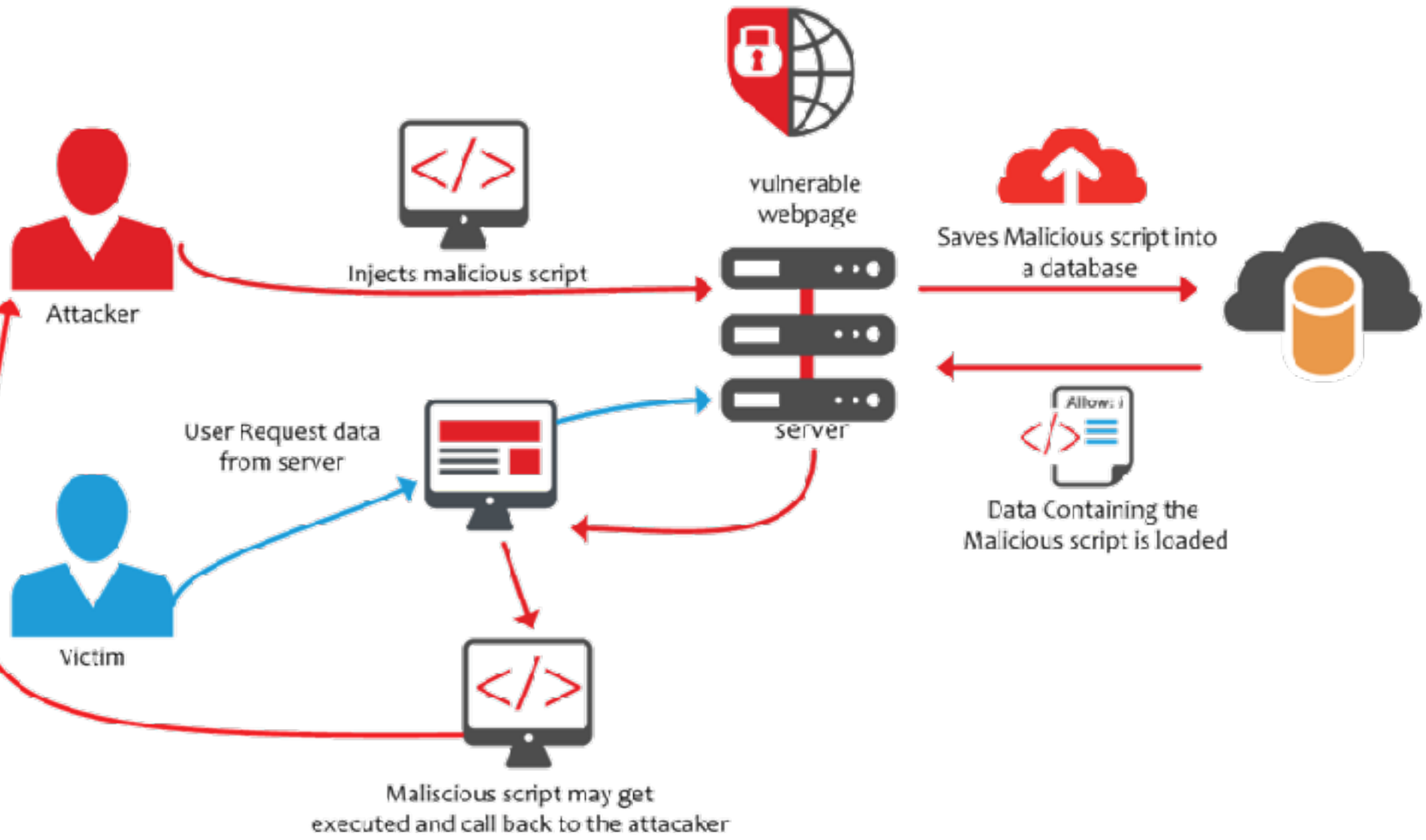
PHP Security-3

Jupiter Zhuo

Prevent The Most Common Form of Attacks

- XSS
- Session Hijack Or CSRF
- Remote Code Injection

Cross Site Scripting
XSS Attacks



What We Covered

- XSS Stored
- XSS Reflected
- Protecting Form

XSS Stored

- Filtering
  - force the data type
    - strip_tags(),preg_replace()
- Validation
  - strlen , preg_match()
- https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet

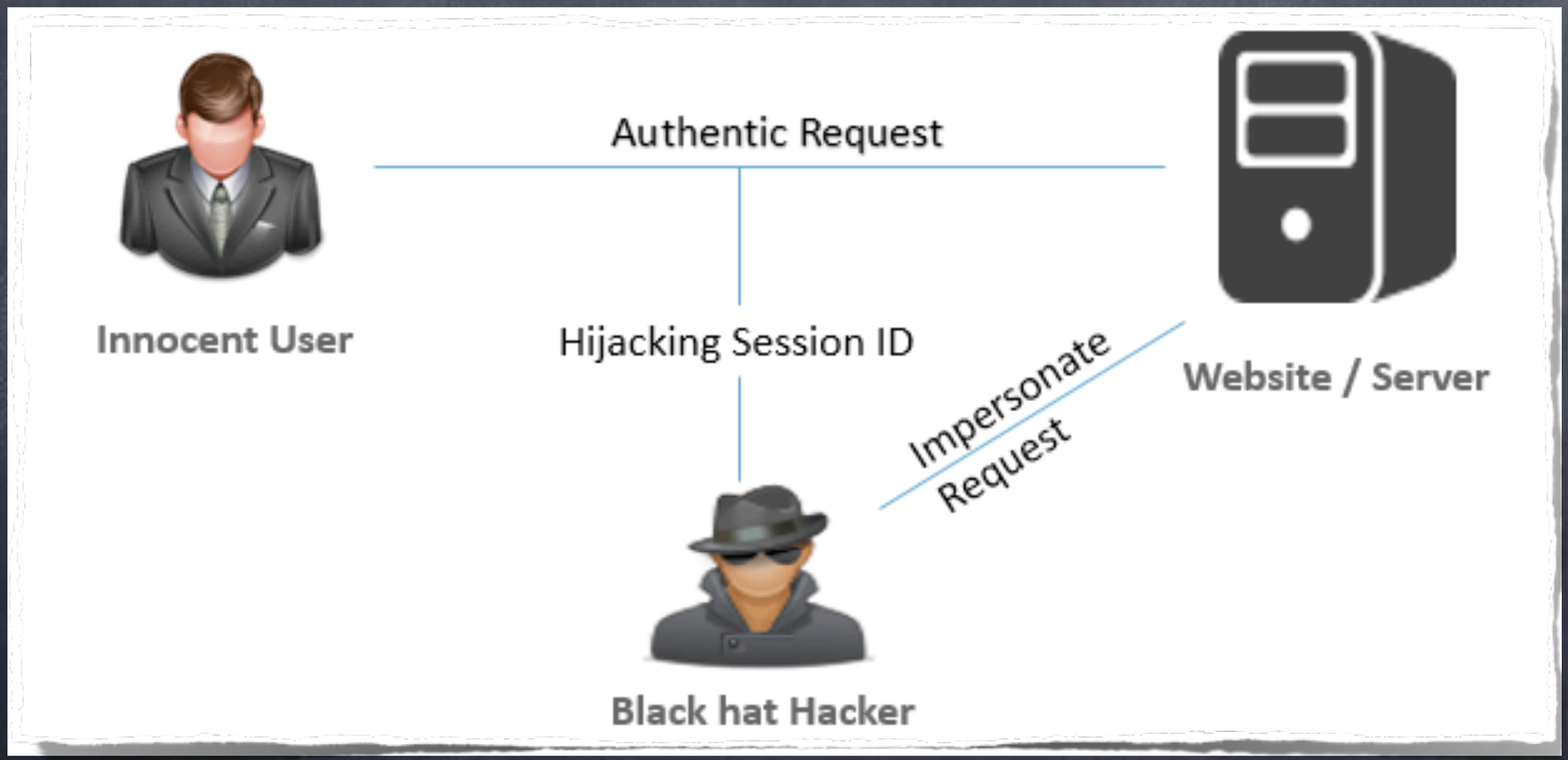
XSS Reflected

- escape all user output
  - htmlspecialchar()
  - htmlentities()
- Setting the outbound Charset

XSS Protecting Form

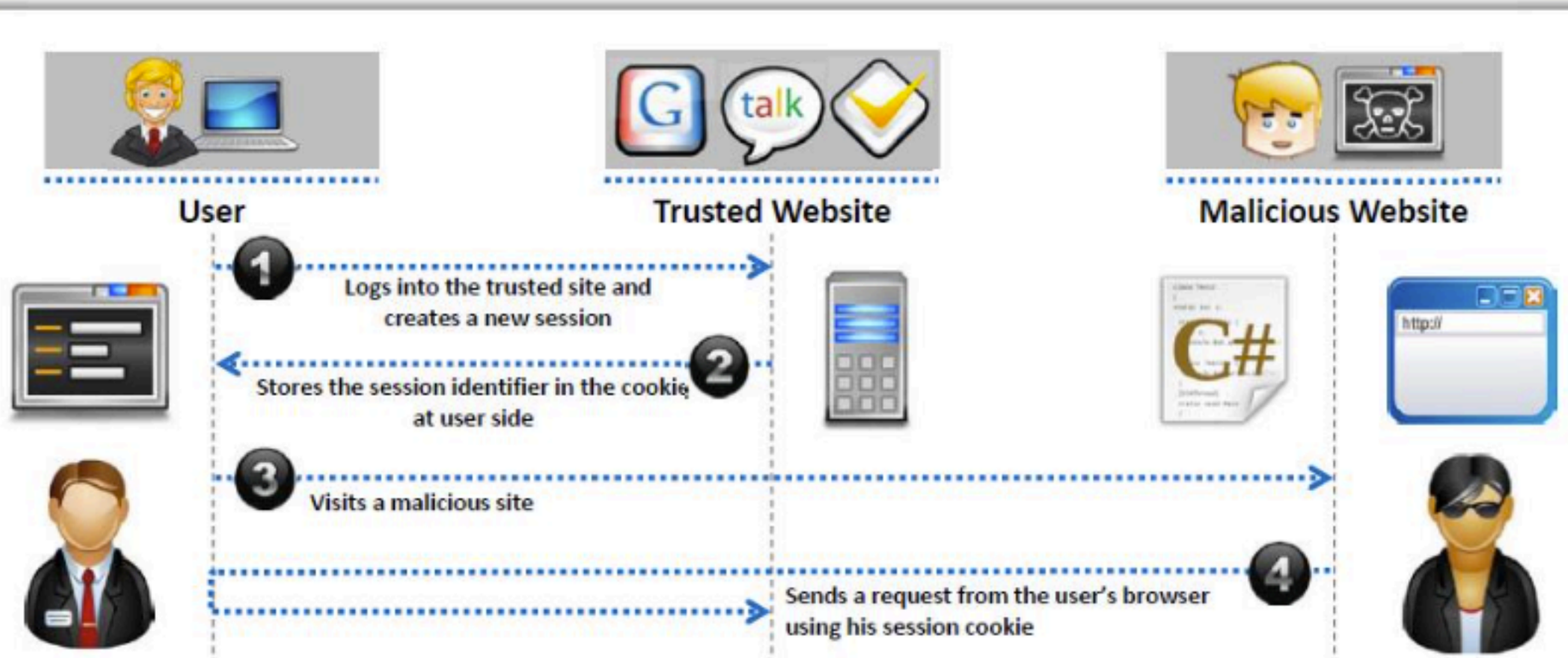
- Single Hash
- CaPTCHA
- HTML 5 Form Feature + Filtering & Validation (Server Side)

session hijack



CSRF Attack

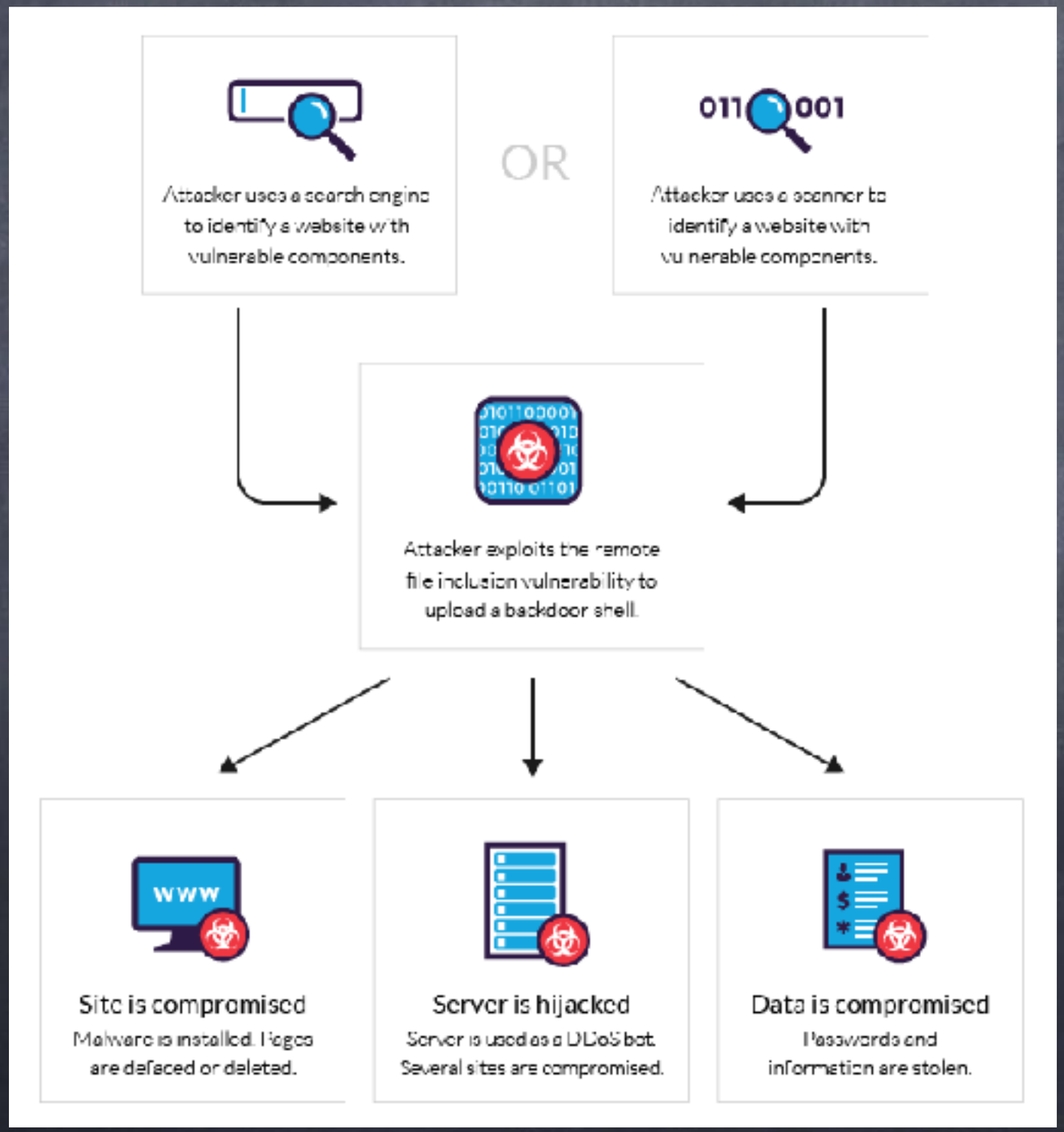
- Cross-Site Request Forgery (CSRF) attacks exploit web page vulnerabilities that allow **an attacker to force an unsuspecting** user's browser to send malicious requests they did not intend
- If user establishes a session with the trusted site and if he simultaneously visits a **malicious site**; then a malicious HTTP request is sent to the trusted site using user session that was established with the trusted site, this results in compromising the **integrity** of the application



How To Preventing Session Hijack & CRSF

- Regenerating Session ID
- Providing Logout Option
- Keep Session Short Time
- Session Profile (Multiple Login)
- using SSL

Remote File Inclusion Attack



How To Preventing RFI

- Filter & Validate include file
- build safe command using __DIR__
- Settinng php.ini
  - allow_url_fopen =Off
  - allow_url_include =Off
  - disable_functions eval,phpinfo,exec,passthru,shell_exec,system,proc_open, popen,curl_exec,curl_multi_exec,parse_ini_file,show_source

Lab

- Protect Shopping cart against XSS Attack
  - Purchase Item click shopping cart
  - Enter notes field with javascript
    - open view/cart.php implementing filtering,validation and output escaping
- Protect against char based Attack
  - open index.php
  - set Char encoding to UTF-8

Lab

- Secure the login process against session hijacking
  - open index.php
    - regenerate session id
  - open view/login.php
    - Filter, validate, Escape all Input
    - Add One time hash
    - Add Captcha
    - Attempt hijacking using another browser

Lab

- Protect Against Remote Code Injection
  - open index.php
  - create an array of allowed pages can be include
  - validate all incoming file request
  - Test using contact us to upload phpinfo file after you fixed security bugs