PHP Security

Level: medium

Length: 28 hours

Who can attend: PHP programmers who want to improve the security of their applications

Required infrastructure: VGA projector, whiteboard, personal computer connected to Internet

Course objectives: realize the importance of the security of PHP applications, identify and fix the vulnerabilities.

Prerequisites: basic knowledge regarding Internet: browsers, web servers, HTTP protocol, HTML, CSS, Javascript, PHP and database programming.

Related courses: Web Technologies - HTML, CSS, JavaScript & JQuery, Building Web Applications with Apache, PHP & MySQL, Introduction to Linux

Description:

As one of the most popular language for building web applications, PHP has become a preffered target for hackers. Security sites and forums describe a growing number of attacks on the PHP websites.

Any programmer must be aware of the importance of security. Building security into PHP applications should be an everyday task.

During the course we will study the most common security vulnerabilities of PHP itself but also the vulnerabilities of the web server or the database server. Specific solutions will be presented for each vulnerability.

The course is focused on practice. The trainee will learn about security of the websites by breaking into a vulnerable website, identifying the vulnerability and fixing it in the end.

Bibliography:

- Pro PHP Security (2nd Edition), 2010, Chris Snyder, Thomas Myer, Michael Southwell, Apress
- Hardening Apache, 2004, Tony Mobily, Apress
- https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

Content:

1. What is security and why do we need it?
   Meeting the attackers and their objectives.
   Open Web Application Security Project (OWASP)
   Risk management

2. OWASP Top 10 (the most frequent security vulnerabilities)
   A1. Injection: SQL injection, code & command injection, SSI injection
   A2. Broken Authentication an Session Management
   A3. Cross Site Scripting (XSS)
   A4. Insecure direct object reference
   A5. Security misconfiguration
   A6. Sensitive data exposure
   A7. Missing function level access control
   A8. Cross-site Request Forgery (CSRF)
   A9. Using components with known vulnerabilities
   A10. Invalidated redirects and forwards

3. Prevention methods
   a. Authentication Techniques
   b. HTTP vs HTTPS
   c. Storing confidential data (hashing vs. encrypting)
   d. Filtering/Validation/Escaping Techniques
   e. Controlling access: ACL, RBAC
   f. Using captcha
   g. Tokens and Session Management
   h. Securing File Uploads
   i. Hiding and logging errors
   j. Web server and database server security