Seminar session 6

Intro to secure web programming with Django

Django is a high-level web framework. The software that makes it easy to build dynamic websites.

advantages:

- ➔ Provides an abstract solution to common problems of web development by providing "shortcuts" to frequent programming tasks.
- ➔ Dynamic website as opposed to static pages – web application.
- ➔ Can retrieve form a database or carry out a task based on user input
- ➔ Provides a mechanism of mapping requested URLs to code that handles requests – code to specific to username.
- ➔ HTML forms can easily be displayed, validated and redisplayed.
- ➔ Supports conversion of user- submitted input data types for easy manipulation.
- ➔ Ability to spate content from presentation – a template system for look & feel modifications.
- ➔ Provides a mechanism to easily integrate with storage layers.

common development frameworks:

Smarty:

- ➔ Focuses on separation application logic and presentation
- ➔ Uses a temporary caching to facilitate delivering pages

Yii framework:

- ➔ focuses on reducing SQL statements and facilitates integration with other platforms
- ➔ Provides security, authentication, and professional features to create robust projects rapidly.

Zend framework:

- ➔ Popular for enterprise-level application and focuses on modularity, performance and maintenance
- ➔ Allows continues extension and integration

Django:

- ➔ A high-level Python-based framework for rapid web application development
- ➔ Supports the properties of common frameworks

Specific properties:

- ➔ Django encourages loose coupling:
  application should communicate with each other through APIs:
  - template
  - Database-access system
  - HTTP request/response layer
  - Caching
- ➔ Django supports the MVC architecture

Issues with traditional frameworks:

- ➔ The presentation is tied to the code
- ➔ The database code is tied to the business logic
- ➔ The database connection parameters and backend are hard-coded
  - Hardcoding is in general a bad solution. Because only the developers are familiar to the code.
  - Because it is in plaintext it is vulnerable for attackers.

PHP tradidtion

- ➔ code is kept under web server's document root ie /vear/www
- ➔ code is exposed – not secure!

Django:

- ➔ Python code can be kept outside document root e.g /home/mycode
- ➔ Secured – web server not exposed