Possible Security problems

Inhaltsverzeichnis

1.1 OWASP Top 10 – 2010 [http://www.owasp.org/index.php/Top10]

1. **Injection (e.g. SQL, OS and LDAP)**
   - occur when untrusted data is sent to interpreter as part of command or query
   - attacker's hostile data can trick interpreter into executing unintended commands or accessing unauthorized data
2. **Cross-Site Scripting (XSS)**
   - occur whenever application takes untrusted data and sends it to a web browser without proper validation and escaping
   - allows attackers to excecute scripts in victim's browser

     → hijack user session

     → deface web site

     → redirect the user to malicious sites
3. **Broken Authentication and Session Management**
   - application function related to authentification and session management not implemented correctly → allows attacker to compromise passwords, keys ...
4. **Insecure Direct Object References**
   - occurs when developer exposes a reference to an internal implementation object, such as file, directory or DB key
   - without access control check / protection, attacker can manipulate these references to access unauthorized data
5. **Cross-Site Request Forgery (CSRF)**
   - forces a logged-on victim's browser to send forged HTTP request, including session cookie and any other automatically included authentication information, to a vulnerable web application
6. **Security Misconfiguration**
   - secure configuration defined an deployed for the application framework, application server, web server, DB server and plattform
   - keeping all software up to date, including all code libraries used by application
7. **Insecure Cryptografic Storage**
   - many web application do not properly protect sensitive data with appropriate encryption or hasing
   - attacker's may steal or modify such weakly protected data

8. **Failure to Restricte URL Access**
   - check URL access rights before rendering protected links and buttons
   - application needs to perform similar access control checks each time these pages are accessed or attackers will be able to forge URLs to access hidden pages
9. **Insufficient Transport Layer Protection**
   - applications frequently fail to authenticate, encrypt and protect the confidentiality and integrity of sensitive network traffic
   - they sometimes support weak algorithms, use expired or invalid certificates or do not use them correctly
10. **Unvalidated Redirects and Forwards**
   - Web applications frequently redirect an forward users to other pages and websites and use untrusted data to determine the destination pages
   - without proper validation, attackers can redirect victims to phishing or malware sites or use forwards to access unauthorized pages

1.2 Risks and solutions for CoALA

Main risks: No 1, 2, (4), 5, 8, 10

1.2.1 User Login

- **Risk**: Injections and buffer overflow (corrupt execution stack)
- **Solutions:**
  - check input data
    - if input data include e.g. ';' → show user error message
  - limit input fields
  - use secure login (SSL)

1.2.2 User logged in

- **Risk**: We shall use URL to allow other applications / modules to directly interact with CoALA

  → Risks of injections, XSS, (Insecure Direct Object References), CSRF, Failure to Restricted URL Access, Unvalidated Redirectes and Forwards
- **Solutions**:
  - checked EVERY time when data are requested wheather supplied values are valid and user is authorized
  - escape special characters and all untrusted data based on HTML context
  - use parameterized APIs (e.g. Prepare Statements) and use them carefully
  - „white list“ input validation
  - use of unique token in URL
  - avoid redirects and forwards