ThoughtWorks®

Security Testing of Web Applications

Agenda

- Security Testing, Web Application, and Web Security Testing
- Security testing approach
- Q&A

Security testing

- Security testing is a process to determine that an information system protects data and maintains functionality as intended.

Web Application

- At the core of every web application is the fact that all of its functionality is communicated using HTTP, and its results are typically formatted in HTML. Inputs are communicated using GET, POST, and similar methods.

Web Security Testing

- Web security testing tells us whether Web based applications requirements are met when they are subjected to malicious input data.

OWASP Top 10 Web Application Security Risks for 2010

- A1: Injection
- A2: Cross-Site Scripting (XSS)
- A3: Broken Authentication and Session Management
- A4: Insecure Direct Object References
- A5: Cross-Site Request Forgery (CSRF)
- A6: Security Misconfiguration
- A7: Insecure Cryptographic Storage
- A8: Failure to Restrict URL Access
- A9: Insufficient Transport Layer Protection
- A10: Unvalidated Redirects and Forwards

Security testing approach

- Password cracking
- URL Manipulation
- SQL injection
- XSS (Cross Site Scripting)
- Vulnerability
- Spoofing

Password cracking

- Guessing username and password
- Commonly using dictionary attack
- Cookies can also leak authentication information
- Keyboard recorder

What we can do

- Enforce a complex password
- CAPTCHA (Completely Automated Public Turing test to Tell Computers and Humans Apart)
- Encrypting cookies
- Software keyboard

URL Manipulation

- Some web applications communicate additional information between the client (browser) and the server in the URL. Changing some information in the URL may sometimes lead to unintended behavior by the server.

What we can do

- Change the parameters (or even delete some sub-string of link) in URL and check the web application's behavior
- Better to change the link parameter into GUID or encrypt it
- The parameter should not be predictable (.../user/... and .../admin...)

SQL injection

- This is the process of inserting SQL statements through the web application user interface into some query that is then executed by the server.

What we can do

- Each input area should be tested
- Special characters (include characters of different languages) should be handled/escaped properly in such cases
- ASCII code should also be tested
- Input string which can let system to divide zero or execute a long time need to be tested

XSS (Cross Site Scripting)

- When a user inserts HTML/ client-side script in the user interface of a web application and this insertion is visible to other users, it is called XSS.

What we can do

- Html tag and other reserved keywords/strings need to be tested as parameter
- Script need to be tested as parameter

Vulnerability

- This is a weakness in the web application. The cause of such a “weakness” can be bugs in the application, an injection (SQL/ script code) or the presence of viruses.

What we can do

- Client/Server cache
- Session time-out
- Even the weakness of Operation System
- Connectivity attack such as DoS (denial-of-service) and DDoS (distributed denial-of-service)

Spoofing

- The creation of hoax look-alike websites or emails is called spoofing

What we can do

- Authentication, such as e-signature and certification
- Encrypted data transfer protocols: SSL, SSH, PKI, SET and so on

References

- Web Security Testing Cookbook, 1st Editionby Paco Hope; Ben Walther
- http://www.softwaretestinghelp.com/security-testing-of-web-applications
- http://www.owasp.org/index.php/OWASP_Top_Ten_Project
- Wikipedia