KV Web Security

1. Aufgabe

Es soll ein Mini-Webserver verwendet und ausgebaut werden, sodass ein bestimmter Angriff (je Person unterschiedlich) möglich ist. Dazu müssen Webseiten erzeugt und ev. (nur als allerletzte Möglichkeit; dann aber jede einzelne Änderung genau dokumentieren!) der Server angepasst werden.
Abzugeben ist eine schriftliche Ausarbeitung in Form von Webseiten: Ein Anleitung, wie der Angriff genau durchgeführt werden kann (+Screenshots), eine Erklärung wie man ihn beheben kann, und die entsprechenden Änderungen der Web-Applikation. In Einzelfällen kann es problematisch sein, weil zu viel am Server geändert werden müsste, dann kann notfalls nach Rücksprache auch eine Bibliothek eingesetzt werden oder andere Änderungen sind möglich.
Die „Applikation", die erforderlich ist, muss als neue Klasse von „ApplicationBase" abgeleitet werden (und ist daher auch in Java zu schreiben!). Es sollten hierbei keine JSP-Seiten, Template-Systeme oder ähnliches verwendet werden, sondern es sollen minimale Webseiten aus Dateien eingelesen oder per „println" erzeugt werden.
Erzeugen sie ihre Webseiten in einem Unterverzeichnis bzw. auf einem eigenen Webserver mit dem Namen ihrer Aufgabe (z.B. 127.0.0.1/A1_1/ oder www.a1_1.com für SQL Injection). Die „sicherheitsbereinigte" Version der Applikation soll eine separate Applikation sein (Kopie + notwendige Anpassungen, diese genaue markiert).

2. Umfang der schriftlichen Ausarbeitung

Äquivalent von ca. 5-10 Seiten A4.

3. Inhalte für Seminararbeiten

Vergleiche hierzu näher die OWASP – Webseiten (http://www.owasp.org/index.php/Top_10_2010-Main) sowie entsprechende CWEs (http://cwe.mitre.org/index.html)!

A1 - Injection
1. SQL injection (SQL-Server is only simulated; define a few exploits which can be parsed and simulated)
2. OS command injection (limited to Windows)

A2 – Cross Site Scripting (XSS)
3. Stored XSS
4. Reflected XSS

A3 – Broken Authentication and Session Management
5. Authentication by Cleartext-Cookie
6. Authentication through replay of MD5 value of password (client-side calculated by JavaScript)

A4 – Insecure Direct Object References
7. Direct object reference (list of "allowed" objects with their direct ID)
8. Path traversal attack (read file)

A5 – Cross Site Request Forgery
9. CSRF by an image within a hidden iframe
10. CSRF adds an admin user through a URL

A6 – Security Misconfiguration
11. Password in configuration file, can be accessed via Web
12. Too extensive error messages (Stack trace, ...)

A7 – Insecure Cryptographic Storage
13. Username+Password in cleartext in cookie
14. Using XOR encryption for password
15. Fixed salt + MD5

A8 – Failure to Restrict URL Access
16. Bypassing authentication through directly accessing a deep URL
17. Using JavaScript on the client for access control

A9 – Insufficient Transport Layer Protection
18. Password recovery by sending it in an E-Mail in cleartext (simulate sending an E-Mail by storing it in a file, which can be accessed via the web server)
19. Using an HTTPS cookie without setting the secure flag --> Access also by non-encrypted connection (simulate an HTTPS connection through "special" URLs)

A10 – Unvalidated Redirects and Forwards
20. Redirect to a site passed in as a parameter
21. Bypass access control through a redirection to a deep URL

Alternative task:
- Extending the servers to support TLS and one appropriate attack

4. Abgabetermin

Die Ausarbeitungen sind bis zum 20.6.2011 per E-Mail an „sonntag@fim.uni-linz.ac.at" mit der Überschrift „Abgabe Web Security - <Name>" abzugeben.