„KV Web Security“

Seminar thesis:

- The thesis should be written preferably in English (or German); programming tasks must have their explanation in English
- Hand- in as a PDF
- Length: 10 pages (excluding title page, ToC, ...)
- Presentation in course for 10-15 minutes (same language as thesis); 24.5.2013
- Single work (each student separately)

Topics: 7 Theoretical, 20 Programming

Content:

1. Summarize and explain: ClickJacking for mobile devices
   http://seclab.stanford.edu/websec/framebusting/tapjacking.pdf
2. Summarize and explain: Cross Channel Scripting/Embedded webservers
   http://seclab.stanford.edu/websec/embedded/embedded-ccs09-paper.pdf
3. Summarize and explain: node.js security
   http://lab.cs.ttu.ee/dl93
4. Summarize and explain: HTML5 securits issues
   http://media.hacking-lab.com/hlnews/HTML5_Web_Security_v1.0.pdf
5. Explanation of “tainting” and description of a system (e.g. in PHP)
6. Static vulnerability analysis: Examples and limitations
7. Securing web sessions: Where to store/how to pass session identifiers, handling them, session fixation attacks etc. (English only, as webpages for SampleWebServer)

SampleWebServer:

1. SampleWebServer: Differential view (automatic) for vulnerable and secure code (includes reformatting existing vulnerabilities to minimize output)
2. Escaping framework for HTML: Example for all kinds of locations to insert data without (including what you can then do; form to enter data which is then inserted as tag, text, Javascript, … content) and with it
3. SSO “Integrated Windows Authentication”: Sample application for IE, Firefox and Chrome and Apache webserver on Linux
4. XPath injection example (based on reading a “database” stored as an XML file)
5. MySpace Worm (simplified; trivial site with personal pages)
6. Error messages as information leak (but see A6-12!)
7. Log injection && Log overflow
8. Session fixation attack (show link which is then “sent” by the user to be entered; actually: copied via clipboard)
9. Captcha reriding: http://blog.opensecurityresearch.com/2012/02/captcha-re-riding-attack.html (Captcha aus einer Bibliothek nehmen)
10. Integrating a simple mailserver for sending mail (including configuration to send mail directly or via another host, potentially with authentication; but should remain simple) + Mail header injection example
11. Phishing website example (select a bank and rebuild it; incl. E-Mail to send, …)
12. Forging request headers via flash (e.g. referrer) – requires flash programming!

13. DoS attack on the webserver (separate Java application; keeping connections open and other approaches without third-party sites)
14. Converting Security Sample Server to a virtual Linux machine, with a second machine to connect from and a network (including IPv6) between them. Example attacks should involve IPv6 (so not directly at server) and DNS poisoning (going to example server 2 instead of 1; web servers should now run on different IPs instead of different ports).
15. Implementation of a JavaScript virtual keyboard with anti-screenshot functionality (see http://ijcsi.org/papers/IJCSI-8-5-3-534-537.pdf).
16. JavaScript Calender (or something similar: AJAX), which uses "eval" on the client to access the "JSON" data passed from the server. Server does not validate properly and allows sending some data from client back to them → Create link which leads to JavaScript execution on different client clicking on it
17. Create "secret" authentication token to be sent by SMS on webpage by JavaScript and send it to server for passing it to SMS gateway along username etc. Requires entering token on next page to authenticate user as controlling this telephone number.
18. Implement default "404" and "500" webpages for SecuritySampleServer, which an application can change. Model it after Apache with lots of useful information. Example application which crashes and produces a 500 with lots of useful data for attackers.
19. Store session information (three different ones!) in Cookie, form field and URL and compare it on server: Implement an attack to pass on these tokens to another server who then impersonates the first one.
20. Sample application with password lockout and increasing delays: JavaScript and server-side.