Course code: **SEC/WEB**

Course title: **Security testing of modern Web applications**

Days: 5

Description:

Course intended for:

The training is aimed at testers, programmers, application administrators, auditors and anyone interested in issues of security, wanting to acquire comprehensive knowledge in the field of penetration tests, reverse engineering and its use for verification of security of the systems developed.

Course objective:

The training objective is to deliver knowledge, which will turn the participant into a security verifier, using knowledge bases and ready exploits, at the same time serving as a basis for becoming a security researcher, able to work with unknown applications and protocols and to search for new susceptibilities in them. The training strengths include the .NET (stack Microsoft) and Java (stack Oracle and open source) application examples – from the field of banking and Android and iOS mobile applications. The participants perform tasks associated with designing of penetration tests, using a laboratory in form of a virtualized environment simulating the typical problems of a complex infrastructure.

Requirements:

The participants are expected to have experience in work with Web applications – preferably as programmers or implementers, or to have experience in the field of security of such solutions. Familiarity with basic concepts of Java and .NET, as well as basic administration in Windows and Linux system, will allow the participants to complete all labs smoothly.

Course parameters:

Duration: 5*8 hours (5*7 net hours)

Length:

Introduction, 1st phase of attack, Web application attacks – 2 days Reverse engineering - 1 day Exploit development - 1 day Wi-Fi, Cards, Reporting, Cloud - 1 day

Course curriculum:

1. Types of penetration tests
    - Difference between a penetration test and an audit
    - Penetration test methodologies
    - Test types: whitebox, blackbox, greybox
    - Penetration test phases
    - Threat estimation and modeling and attack trees
    - The scope of the test
    - R&D projects
    - Checklists in penetration tests, encoding standards: : CIS, CERT
2. Susceptibility kinds, susceptibility types according to various classifications
    - Susceptibility classification according to OWASP and CWE
    - what is CVE? – open and closed susceptibility bases
3. Tools for analysis of the network and identification phase, gathering of information on the attack target
    - sniffers
    - active tools
    - passive tools
    - configuration verification
    - use of: : p0f, nmap, Maltego
    - Google dorks
    - Shodan
4. Web application attacks
    - SQL Injection

Sages sp. z o.o., ul. Nowogrodzka 62c, 02-002 Warszawa
tel. +48 22 203 56 00 fax: +48 22 203 56 01
e-mail: office@sages.io web: www.sages.io

- *Injection
- XSS
- CSRF
- Direct access to data and objects
- XXE
- Spring EL Injection
- Shellshock
- Session management
- Typical problems with user registration and password recovery

5. Tools for proxy type manual tests
    - Use of Burp Suite, OWASP ZAP

6. Automatic security scanners
    - Use of Nessus, Nexpose, Burp Suite Scan, Skipfish, Arachni

7. Exploit sets
    - Use of Metasploit

8. Scripts and automation of security tests
    - Use of ZAP and Mozilla ZEST

9. Reverse engineering
    - Executable files for Windows and Linux
    - Mobile and in-built platforms
    - Decompilation of JVM and .NET executable files

- Reversing of protocols and file formats
- Tools: IDA Pro, OllyDbg, Immunity Debugger, Windbg, Radare, objdump, Ronin

10. Mobile application tests: Android, iOS
11. Equipment
    - In-built equipment tests
    - Internet of things
    - SDR (Software Defined Radio)
12. Wireless network tests
    - Network identification
    - Gaining access
13. Testing of WebServices
    - XXE
14. Cryptography
    - SSL
    - POODLE
    - Man-in-the-middle
    - Weaknesses in cryptography implementation
15. Intelligent cards
16. DoS and DDoS attacks
    - Application logic attacks: ReDOS, XML Bomb, Flood
17. Development of exploits – not only exploitdb ;)
    - Use of:

I. Buffer overflow
II. Race condition
III. Backdoor
IV. Use-After-Free and Heap Spraying
V. Return Oriented Programming

- Tools for static analysis for JVM and .NET
- fuzzing
- shellcode operation principles
- overriding of ALSR protection modes
- creation of new modules for Metasploit

18. methods of acquisition and maintenance of access
    - passwords
    - Trojan horses
    - Pivoting
    - Use of meterpreter
    - Concealing of presence using modules and rootkits
19. Physical acquisition of access
    - implants
    - wiretaps
20. Cloud
    - Cloud specific issues
    - Cloud use in tests: DoS, DDoS
21. Enterprise class issues

- Security specific solutions: API Gateway, Oracle DB, Imperva, SAP, DB2
- BYOD, MDM
- Management of roles and users: IDM
- VPN
- Backup

22. Management of information during penetration tests
    - Building of knowledge bases and attack bases
    - Dradis Framework
    - Magic Tree
23. Report development
    - What should a good penetration test report contain?
    - How to formulate recommendations and overrides?
    - How to deliver a report securely to the client?
    - How to describe susceptibility and obtain CVE?
24. Planning and management of a penetration test project
    - What to do to avoid a failure?
    - Formalities in cooperation with the client
    - Management of scope
25. The pentester image
    - Networking
    - Business cards