HT103 - Vulnerability Detection and Exploitation

You will learn how to apply the theory and practice of **code auditing**, how to **dissect an application**, how to discover security vulnerabilities and assess the danger each vulnerability presents. You will **run vulnerability scans and observe exploits** to better secure networks, servers and workstations. This course is valuable for those involved in securing enterprise systems: network and system administrators, computer security personnel, officers with direct involvement in security and those involved in cyber security measures and implementation.

Course Agenda

DAY 01

1. **Introduction**
    a. Course objectives
    b. Course structure
2. **Module Introduction**
    a. Overview of the day
3. **Exploitation Techniques Fundamentals**
    a. A set of categories of software's vulnerabilities
        i. Memory Corruptions
            1. Buffer Overflow
                a. Stack/Heap buffer overflow
                b. Off-by-one (stack/heap)
                c. Modern memory protection mechanism (e.g., DEP and ASLR)
        ii. Format String Bugs
        iii. Logical flaw
        iv. Configuration flaw
4. **Public Vulnerabilities & 0-Days**
    a. Vulnerability Definition
        i. CIA Paradigm
        ii. Definition of Vulnerability
        iii. Definition of Exploit
    b. Public and Private Vulnerabilities
        i. Public Vulnerabilities
        ii. CVE
        iii. 0-day and 1-day Vulnerabilities
        iv. Common methods for vulnerabilities identification

1. Fuzzing
2. Code review
3. Reversing

v. Malware analysis e patch analysis
1. 1-day vulnerabilities

c. Exploits
i. An exploit at work
1. techniques, payloads, injection and execution
ii. Exploit (technical) taxonomy
1. Local Exploit
2. Remote Exploit
3. Userland exploit
4. Kernel exploit
iii. Private
iv. Publics
1. Public exploit repositories
v. Exploit Markets
1. White market
a. iDefense and ZDI
b. Bug bounty programs
i. Google, Mozilla, Facebook, Microsoft
ii. Bugcrowd
c. Other initiatives
i. PWN2OWN
ii. Pwnium
2. Black market
3. Gray market

5. Fuzzing bugs - how to write a simple fuzzer

a. The history of fuzz testing
b. What “to fuzz” means
c. Even a dump fuzzer can give you a crash
i. LAB - Example of dump, random fuzzing of files
d. How to create a fuzzer
i. Random fuzzing
ii. Specification based fuzzing (e.g. RFC-based fuzzing, (E)BNF fuzzing)
e. Let’s write a fuzzer
i. LAB - We use Metasploit
1. LAB - Introducing the framework and the modules structure
ii. File format fuzzing with Minifuzz by Microsoft

DAY 02

6. **Recap of the previous day**
7. **Module introduction**
    a. Overview of the first day
8. **OWASP Top 10 2013, PART I**
    a. Top 10 is a “concept” that can be extended to other contexts (e.g., mobile, cloud)
    b. Security issues related to web application and technologies
        i. Web application as a gateway to the corporate internal network
    c. Risk definition and adopted methodology
        i. Likelihood
        ii. Impact
            1. Technical
            2. Business
    d. For each item in the Top 10
        i. The theory behind the vulnerability
        ii. Attack scenario(s)
            1. Focus on the impact of the related attack
        iii. Live examples
            1. **LAB - Vulnerable code examples and exploitation (ASP.NET)**

9. **Source code auditing**
    a. What source code auditing is?
        i. Vertical and horizontal approaches
        ii. Theory from OWASP Code Review guide
    b. Manual vs automated review
        i. Theory, limitations and common issues or pitfalls
        ii. Manual and automated tools

DAY 03

10. **Recap of the previous day**
11. **Module introduction**
    a. Overview of the first day
12. **Client-side vs Server-side attacks**
    a. Defining Server-side attacks
        i. Examples and strategies
    b. Defining client-side attacks
        i. Examples and strategies

13. **Mobile Vulnerabilities and Weakness**
    a. OWASP TOP 10 for Mobile 2014
        i. For each item in the top 10
            1. A theoretical introduction will be provided

14. **Modify Exploit Code**
    a. Not always an exploit works out-of-the-box
        i. A real world example
            1. **LAB - Jboss Invoker Deploy exploit provided by Metasploit failed, even if it worked on a test VM with the same vulnerable Jboss version installed.**
            2. Execution vs comprehension: understanding the vulnerability is more important than run an exploit
                a. **LAB - Google for retrieve an exploit source code and modify it a bit**
                b. **LAB - Modify, run and hack the target machine**

DAY 04

15. **Recap of the previous day**
16. **Module introduction**
    a. Overview of the first day

17. **Web Application Exploit Development**
    a. Why should I use Metasploit for web exploiting ?
    b. Framework methods to develop a professional web exploit
        i. **LAB - Exploit development based on proposed Owasp TOP 10**
        ii. **LAB - SQL Injection exploiting**
        iii. **LAB - Cross-Site Scripting exploiting**
        iv. **LAB - Directory traversal exploiting**
        v. **LAB - CSRF exploiting**

18. **Reference and tools**