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. **Module Introduction**
    a. Overview of the day

2. **Exploitation Techniques Fundamentals**
    a. A set of categories of software's vulnerabilities
        i. Memory Corruptions
            1. Buffer Overflow
                a. Stack buffer overflow
                    i. LAB - An example of stack 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

3. **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 Vulnerabilites
        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

4. **Fuzzing bugs - how to write a simple fuzzer**
   a. The history of fuzz testing
   b. What “to fuzz” means
   c. Even a dumb fuzzer can give you a crash
      i. LAB - Example of dumb, random fuzzing of files
         1. Charlie Miller’s 5 lines
   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
         2. LAB - Write a simple fuzzer (FTP) - EIP = 41414141
      ii. File format fuzzing with Minifuzz by Microsoft
         1. LAB - File fuzzing with Microsoft Minifuzz

DAY 02

5. **Recap of the previous day**

6. **Module introduction**
    1. Overview of the first day

7. **OWASP Top 10 2013**
    1. Top 10 is a “concept” that can be extended to other contexts (e.g., mobile, cloud)
    2. Security issues related to web application and technologies
        1. Web application as a gateway to the corporate internal network
    3. Risk definition and adopted methodology
        1. Likelihood
        2. Impact
            1. Technical
            2. Business
    4. For each item in the Top 10
        1. The theory behind the vulnerability
        2. Attack scenario(s)
            1. Focus on the impact of the related attack
        3. Live examples
            1. **LAB - Vulnerable code examples and exploitation (ASP.NET)**

DAY 03

8. **Recap of the previous day**

9. **Module introduction**
    1. Overview of the first day

10. **Source code auditing**
    1. What source code auditing is?
        1. Vertical and horizontal approaches
        2. Theory from OWASP Code Review guide
    2. Manual vs automated review
        1. Theory, limitations and common issues or pitfalls
        2. Manual and automated tools

11. **Client-side vs Server-side attacks**
    1. Defining Server-side attacks
        1. Examples and strategies
    2. Defining client-side attacks
        1. Examples and strategies

12. **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

13. **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

14. **Recap of the previous day**

15. **Module introduction**
    a. Overview of the first day

16. **Web Application Exploit Development**
    a. Why exploiting web applications
        i. **LAB - SQL Injection exploiting**
        ii. **LAB - Cross-Site Scripting exploiting**
    b. Framework methods to develop a professional web exploit
        i. **LAB - CSRF exploiting**

17. **Reference and tools**