CIS 418/518 – Secure Software Engineering
Software Security Testing

Jagadeesh Nandigam

School of Computing
Grand Valley State University
nandigaj@gvsu.edu

Outline

What is Security Testing?

- Security testing is a form of testing that is performed with the intention of revealing flaws in security mechanisms and finding the vulnerabilities or weaknesses of software applications.
- The goal of security testing to ensure that the core properties of confidentiality, integrity, and avaialability of a software application are not compromised by its users.
- Security testing is about making sure bad things don't happen.

Security Testing Approaches

- Outside ⇒ In vs. Inside ⇒ Out Approaches
  - Outside ⇒ In Approaches
    - Late-lifecycle testing efforts made after software is completed and in its operational environment.
    - Example: penetration testing
  - Inside ⇒ Out Approaches
    - Early-lifecycle testing efforts made prior to softwre release.
    - Use of static/dynamic analysis tools, fuzz testing, and tests based on abuse/misuse cases and architectural risk analysis.
- Functional Security vs. Adversarial Security Testing

Functional Security Testing

- Testing security mechanisms/features and other security requirements to ensure their functionality is properly implemented.
  - Component/Unit level testing
  - System level testing
  - White-box & Black-box testing

Adversarial Security Testing

- Testing motivated by understanding and simulating the attacker's perspective.
- Based on risk analysis results, abuse cases, attack patterns, attack surface analysis, threat models, security requirements, etc.

Dynamic Analysis and Fuzz Testing

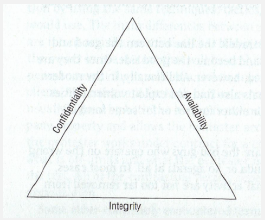
- Run-time verification of software using dynamic analysis tools is necessary to ensure that a program's functionality works as designed.
- Dynamic analysis tools monitor application behavior for memory corruption, user privilege issues, and other critical security problems.
- *Fuzz testing* is a specialized form of dynamic analysis used to induce program failure by deliberately introducing malformed or random data to an application.
- Fuzz testing involves inputting large amounts of random, semi-valid inputs, called fuzz, to the application in an attempt to make it crash.
- Fuzzers are used to generate inputs for the program under test.
- Fuzz testing is effective in discovering vulnerabilities that can be exploited by buffer overflows, SQL injection, denial of service, and XSS.

Penetration Testing

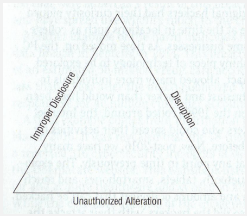
- Penetration testing can be defined as a legal and authorized attempt to locate and successfully exploit vulnerabilities in computer systems for the purpose of making these systems more secure.
- An activity that happens once software is complete and installed in its operational environment.
- A penetration tester conducts tests to survey, assess, and test the security of an application by using the same techniques, tactics, and tools as an attacker.
- Proper penetration testing always ends with specific recommendations for addressing and fixing the issues discovered during the test.
- Some commonly used terms for penetration testing are pen testing, ethical hacking, white-hat hacking.

The CIA and Anti-CIA Triads

- Any organization that is security minded is trying to maintain the core principles of confidentiality, integrity, and availability (the CIA triad).
- A pentester's job is to find holes in the client's environment that would disrupt the CIA triad and how it functions (the anti-CIA triad).
- Before the penetration testing is performed, make sure the client always is aware of the potential risks to their business and make sure they have made backups and put other measures in place in case a catastrophic failure occurs.



The CIA Triad



The Anti-CIA Triad

Types of Penetration Testing

- Black-Box Testing
  - Performed from a remote location much like from a real attacker
  - Extremely limited in your information about the system under test
- Gray-Box Testing
  - Provided with limited information on some critical resources, but untouchable, ahead of time
- White-Box Testing
  - Provided with full knowledge of the structure and makeup of the target enviroment
  - Commonly performed by internal teams as a means for them to quicxly detect problems and fix them before an external party locates and exploits them
  - Time and cost required to find and resolve the security vulnerabilities is comparably less than with the black-box approach

Penetration Testing Methodology (PTM)

1. Gaining Permission via a Contract
2. Gathering Intelligence (aka Reconnaissance)
3. Scanning and Enumeration
4. Penetrating the Target (aka Exploitation)
5. Maintaining Access
6. Covering Your Tracks
7. Reporting Findings

PTM – Gaining Permission via a Contract

- Get a clear and unambiguous permission, via a signed contract, to conduct the pen test.
- Some of the items that may be included in the contract:
  - Systems to be evaluated or targets of evaluation
  - Perceived risks
  - Timeframe
  - Level of systems knowledge necessary
  - Actions to be performed when a serious problem is discovered (i.e., continue testing or to contact the client right away)
  - Deliverables – vulnerability scanner reports, important vulnerabilities to address, counter measures to implement

PTM – Gathering Intelligence

- Gathering Intelligence – a meticulous process through which you are locating information that may be useful when carrying out later phases of pen testing.
  - Active vs. Passive information gathering methods (with or without engaging the target)
  - Examining a company's Web presence
  - Viewing a Website offline (with website downloaders or crawlers)
  - Finding an older version of an existing Website
  - Gathering information with search engines
  - Targeting employees with people searches
  - Discovering location
  - Social engineering
  - Looking via financial services
  - Investigating job boards
  - Searching email
  - Extracting technical information using `whois` utility

PTM – Scanning and Enumeration

- Scanning is the process of identifying live systems and the services that exist on those systems.
  - *Ping scan or sweeping* to check for live systems using tools such as `ping`, `Angry IP`, and `nmap`.
  - *Port scanning* to identify the ports that are open and closed and the services running on the open ports.
  - *Vulnerability scanning* to identify weaknesses or problems in the software or system configuration using tools such as `Nessus` and `OpenVAS`.

PTM – Scanning and Enumeration

- Enumeration is the process of extracting meaningful information from the openings and information found during scanning.
  - Enumeration requires that connections be actively opened to the target to extract meaningful information.
  - Discretion and patience must be observed at this state to avoid detection.
  - Information gathered during this phase includes: usernames, group information, hostnames, share names, services, application data, routing tables, auditing and service settings, SNMP and DNS details.

PTM – Penetrating the Target

- Using information from scanning, information gathering, and enumeration, the pen tester attempts to gain access to the system and exploit vulnerabilities.
- This stage includes:
  - Cracking passwords
  - Escalating privileges
  - Executing applications
  - Hiding files
  - Covering tracks
  - Concealing evidence

PTM – Maintaining Access

- Once access is gained to the system, the attacker tries to retain and maintain access by:
  - Installing a backdoor
  - Launching a virus, worms, or spyware
  - Inserting trojans
  - Installing a rootkit

PTM – Covering Your Tracks

- Covering your tracks and cleaning up after yourself is important for these reasons:
  - Evading detection for as long as possible gives you time to carry out your attack.
  - Keeping track of what you have done and then removing or reversing the changes after the test is important. What you leave behind could leave the system in an insecure state and be potentially dangerous to your client.

PTM – Reporting Findings

- The final step in penetration testing is to generate a report for the client. The contents of this report should include:
  - Overview of the pen testing process used
  - Summary of any successful penetration scenarios
  - Detailed listing of all information gathered
  - Detailed listing of all vulnerabilities found
  - Description of all vulnerabilities found
  - Suggestions and techniques to resolve vulnerabilities found

References

- Gary McGraw, *Software Security – Building Security In*, Chapters 6 & 7, Addison Wesley, 2006.
- Sean-Philip Oriyano, *Penetration Testing Essentials*, Johh Wiley, 2017.
- Patrick Engebretson, *The Basics of Hacking and Penetration Testing*, Second Edition, Syngress, 2013.