







Secure Software Development

Penetration Testing

Objectives

- Compare and contrast the artifacts and goals for QA testing and security testing 
- Compare and contrast the types of testing used for security testing 
- Explain the purpose for penetration testing and the activities performed during penetration testing
- Compare and contrast the mindset of a hacker and the mindset of a penetration tester
- Explain how penetration testing can be improved 
- Explain what aspects need to be considered when planning penetration tests 

Brain Teaser



Security
Screen Door

Dead bolt

Glass
window

Discussion

- What do we test and how?

Functionality \Rightarrow Does it work?
 \Rightarrow Features

Security is **Not**
a feature or set of features

How is security testing versus security functionality

Vulnerabilities tend to come from unexpected but intentional misuses of the system.

Security testing is -testing for negatives.

Trying to demonstrate that the system does not have bugs.

Types of Testing

- White Box

⇒ Know the inside

- Black Box ←

⇒ No idea of the inside.

Definitions

- Vulnerability (Security Flaw):
 - Specific failure of the system to guard against unauthorized access or actions. It can be procedures, technology (SW or HW), or management.
- Exploiting a vulnerability
 - Using the failure of the system to violate the site security policy is called exploiting the vulnerability
- Penetration Study
 - is a test for evaluating the strengths of all security controls on the computer system. It intends to find all possible security holes and provides suggestions for fixing them.
- Penetration Testing
 - An authorized attempt to violate specific constraints stated in the form of a security or integrity policy.
 - a testing technique for discovering, understanding, and documenting all the security holes that can be found in a system.
- What is the difference between penetration testing and hacking/intrusion?

Authorization

Penetration Testing

- Most common of software security best practices
- Testing performed with a mindset of trying to break into a deployed system
 - Requires complete, deployed system
 - Uncovers vulnerabilities, but may not aid in quantifying risk
- “Badnessometer”
 - It is not a proof techniques. It can never prove the absence of security flaws. It can only prove their presence.

Goals of penetration Testing

- Gaining of read or write access to specific objects, files, or accounts;

- Gaining of specific privileges

↳ Root

- Disruption or denial of the availability of objects.

↳ shut the system down

Success Dependencies

Skill and Knowledge of testers

Ad hoc testing

⇒ Not checklist based

⇒ Exploratory testing

Results not always reproducible
between teams.

Access to real production
systems.

Better Penetration Testing

- Should emphasize issues discovered and tracked from the beginning of the software lifecycle
 - requirements ✓
 - Architectural risk analysis ✓
- Should be based on perceived risk

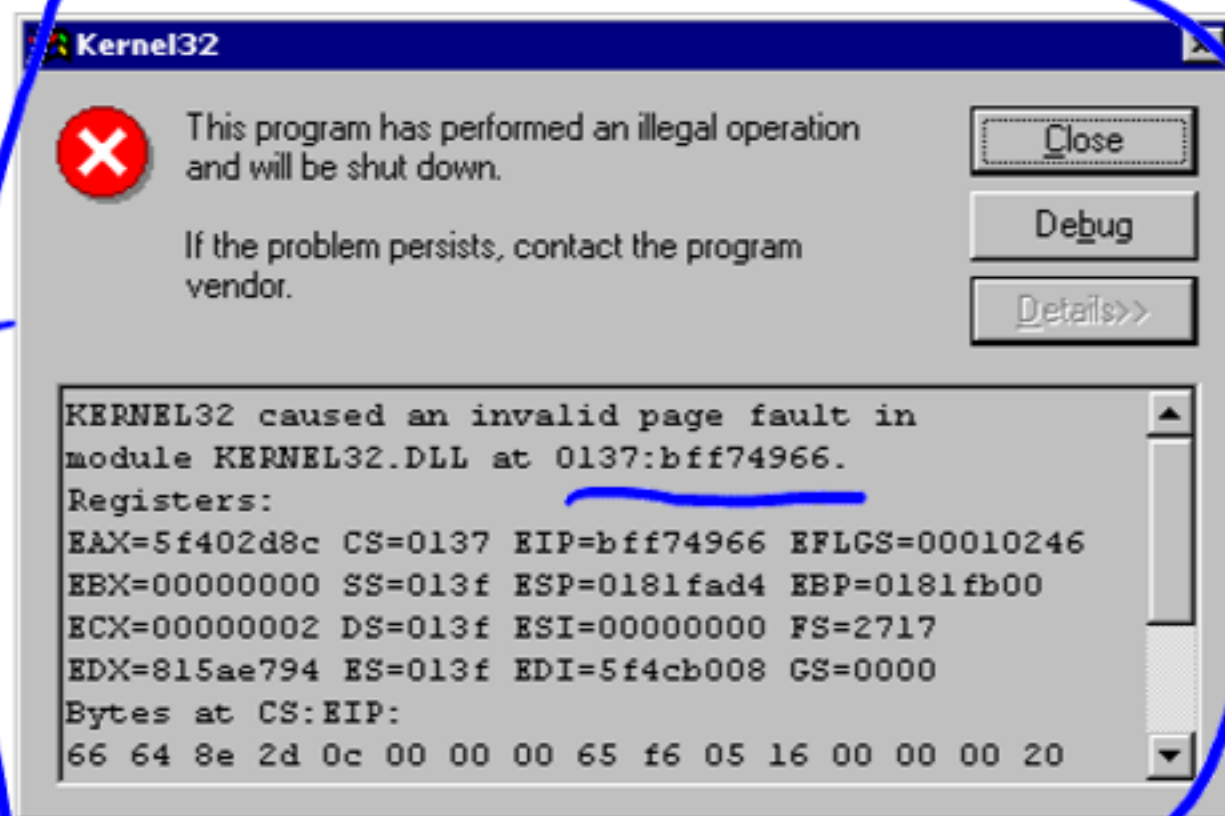


Better Penetration testing

- Disassemblers and compilers
- Control flow and coverage tools — *Where does code execute?*
- DLL Analyzers
- Debuggers
- Rootkits

Planning

- about verifying the absence of insecure functionality.
 - No software development artifacts for these behaviors.
- Starting points
 - interfaces between the software and its external environment.
 - User interfaces
 - network interfaces
 - Error messages and user alert dialogs



Layering of Tests

1. External attacker with no knowledge of the system.
2. External attacker with access to the system.
3. Internal attacker with access to the system.



Penetration Testing Types

- Environment Attacks
 - Scripts, plugins, other installed materials
 - Registry entries
- Input Attacks
 - network protocols and sockets
 - exposed remote functionality
 - DCOM, remote procedure calls (RPCs) and Web services,
 - data files (binary or text)
 - temporary files
 - UI controls allowing direct user input,
- Data and Logic Attacks
 - Denial of service attacks

Hacking Methodology



Fuzz testing

- Fuzz testing is a method of finding software security holes by feeding purposely invalid and ill-formed data as input to program interfaces
- *Inputs include:*
 - Files
 - Network ports
 - APIs

Advantages of Fuzz Testing

- High volume of testing
- Highly automated
- Finds many problems related to reliability
 - Many of which are potential security holes
 - Over 70% of security vulnerabilities Microsoft patched in 2006 were discovered by fuzzing
- Fuzz testing does not typically validate proper reaction to invalid data