

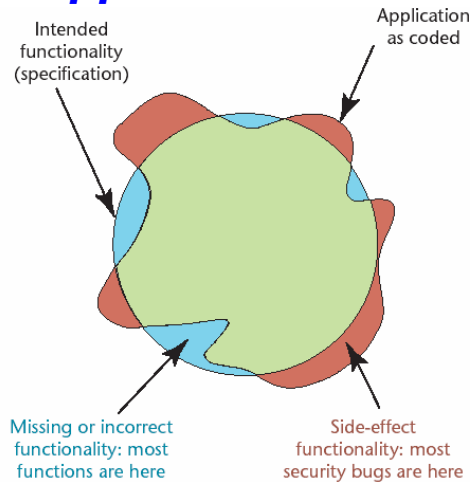
Intro to Security Testing

Michael Gegick
North Carolina State University
mcgegick@ncsu.edu

NC STATE UNIVERSITY

© Laurie Williams 2006

Security Testing: Testing for What It's NOT supposed to do



NC STATE UNIVERSITY

Thompson, Herbert, *, IEEE Security and Privacy, July/Aug 2003, pp. 83-86. Williams 2006

Security Testing

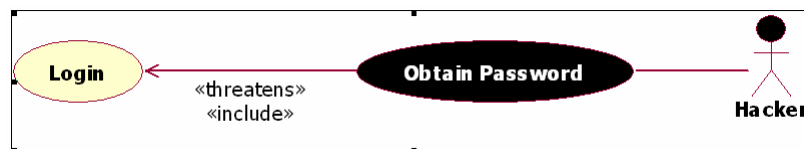
- **Security vulnerabilities can have a larger impact than traditional software bugs**
 - **Litigation**
 - **Brand**
 - **Regulatory requirements**
- **Functional security testing**
- **Risk-based security testing**
- **Penetration testing**

Functional Security Testing

- **Test the security mechanisms**
 - **Do they behave properly?**
 - **Same techniques as functional testing**
 - **Based on requirements**
 - **Example requirements**
 - **Disable user after three failed login attempts**
 - **All network traffic must be encrypted**
- **Can be done early in software process**
- **White hat**
 - **Defensive**

Risk-Based Security Testing

- **Know your threats**
 - Abuse/Misuse cases
 - Architectural risk analysis
 - Attack patterns
- **Test the software to make sure it does not misbehave**
 - Performs an attack
- **Let the risk analysis drive testing**
- **Can be done early in software process**
 - Unit-level
- **Black hat oriented**
 - Destructive

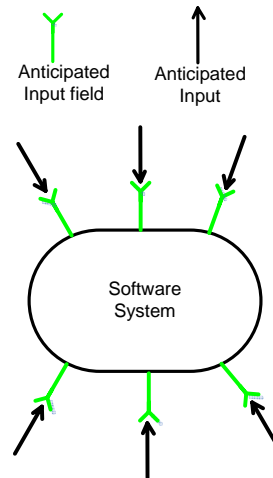


Injection Attacks

- **Largest problem in software security today**
- **Attackers inject malicious input into *input vectors***
 - The path into a software system

Developer-View of Software System

- Feature oriented
- All input will be valid
- Input will only be injected in anticipated input fields

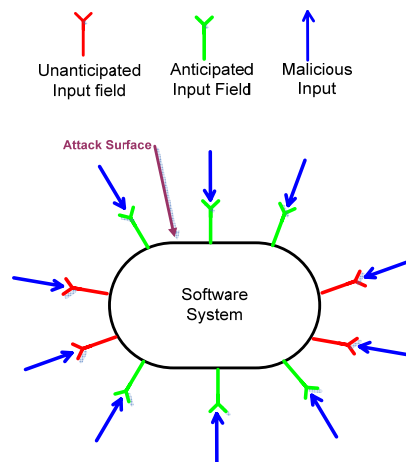


NC STATE UNIVERSITY

© Laurie Williams 2006

Attacker-View of Software System

- Exploit assumptions made by developers
- Inject malicious input into input fields
- Inject malicious input into fields the developers assumed would receive no input



NC STATE UNIVERSITY

© Laurie Williams 2006

Penetration Testing

Test the environment/configuration

- Usually occurs when code is complete and in its operational environment
- Some configuration problems are easy/quick to fix
- **Performed at the system/integration-level**
- **Modern penetration testing covers injection attacks**
- **Primarily black hat oriented**

Penetration Testing (2)

- **Use tools to automate the testing**
- **SPIDynamics**
 - **WebInspect**
- **Disadvantages**
 - **Penetrate-and-patch method**
- **Mostly black box testing**
 - **some white box testing**

Testing Results

- When you have exhausted your test cases, then are you secure?

References

- McGraw, G., Is Penetration Testing a Good Idea?, IT Architect, <http://www.itarchitect.com/shared/article/showArticle.jhtml?articleId=164901611>, 2005.
- McGraw, G., *Software Security: Building Security In*. Boston: Addison-Wesley, 2006.
- Michael, C., Radosevich, W., Risk-Based and Functional Security Testing, <https://buildsecurityin.us-cert.gov/daisy/bsi/articles/best-practices/testing/255.html?branch=1&language=1>, 2005.

Exercise

- Identify technical risks of iTrust
- Ranks your risks
- Develop a test plan to test the risks
- What types of testing should you use and why?
- Indicate how you want to incorporate those tests in the software process
- Use <http://cwe.mitre.org> for information on weaknesses