Hochschule Konstanz
Fakultät Informatik

Software Security

AIN

Hanno Langweg

05 Security Testing

Testing

- White box
  - All manufacturer's knowledge available
- Grey box
- Black box testing
  - No insider knowledge
- Verify **presence** of vulnerabilities
  - In practice, testing will be incomplete, i.e. there may be vulnerabilities not detected by testing
- **Automation** enables large number of test cases

Testing

https://www.eviltester.com/2007/04/not-all-testers-are-evil.html

- Uncover security-related defects
- Test **bias**
  - Developer's perspective
  - Attacker's perspective
- **Sources** for **test cases**
  - Knowledge of **typical** faults
  - Misuse cases, threat trees, security requirements
- Techniques
  - Creation of input, edge cases, sequences of events
  - Degradation/modification of execution environment


YOU ELIMINATED THE BUDGET FOR AUTOMATED TEST SOFTWARE. HOW ARE WE GOING TO TEST OUR NEW CODE?
GO WRITE SOME AUTOMATED TEST SOFTWARE, YOU BIG BABY. I ALREADY PAY YOU, SO IT'S FREE.
TODAY I LEARNED YOU CAN END ANY CONVER-SATION BY CALLING THE OTHER PERSON A BIG BABY.
WAA-WAA! DO YOU WANT YOUR BOTTLE?
www.dilbert.com scottadams@aol.com
©2007 Scott Adams, Inc./Dist. by UFS, Inc.

Vulnerability analysis

- Common Criteria AVA_VAN (vulnerability analysis)
  - Assessment to determine if potential vulnerabilities could allow adversary to violate security requirements
  - **Potential vulnerabilities** identified during software development, operation of a product, flaw hypotheses, quantitative/statistical analysis of security mechanisms etc.
- Levels of vulnerability analysis
  - Survey
  - Analysis
  - Focused analysis
  - Methodical analysis
  - Advanced methodical analysis

Vulnerability analysis

- Vulnerability survey
  - Survey of **publicly available information** to determine vulnerabilities in a specific product
  - Only addresses vulnerabilities that are **known and easily found** by an adversary
  - Evaluator performs own tests assuming *Basic* attack potential
    - Based on collected potential vulnerabilites

Vulnerability analysis

- Vulnerability analysis
  - Requires **functional specification** for security mechanisms
  - Evaluator performs own tests assuming *Basic* attack potential
    - Based on **own flaw hypotheses** generated from documentation/specification
- In evaluations often 50% of effort spent on vulnerability analysis

Vulnerability analysis

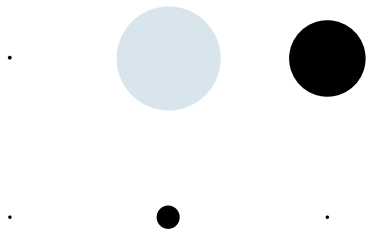
- Focused vulnerability analysis (>=EAL4 ➔ VAN.3)
  - Requires **source code** at least for security mechanisms
  - Evaluator performs own tests assuming *Enhanced-Basic* attack potential
    - Based on own **flaw hypotheses including code inspections**
- Methodical vulnerability analysis
  - Same prerequisites as focused vulnerability analysis
  - Evaluator performs own tests assuming *Moderate* attack potential
- Advanced methodical vulnerability analysis (VAN.5)
  - Evaluator performs own tests assuming *High* attack potential

Fuzzing

- Software testing technique developed in late 1980s
  - Randomly generate data as program input
  - Observe program behaviour
- Input can be textual, graphical, network requests, parameter values passed to library functions etc.
- Detect whether program responds inappropriately or even crashes

Fuzzing

- Generation of data randomly and free from assumptions
  - Covers large range of different inputs
  - Low cost
- Improves security and reliability
- Sometimes templates used to generate random input in specific format
  - More targeted testing
  - Misses creative inputs adversary might use

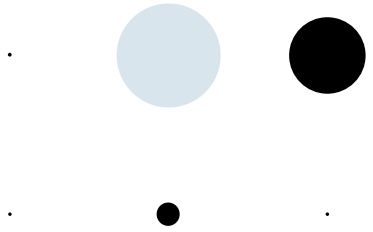


Fuzzing

- Has been used for large software projects
  - Operating systems
  - Open source software
  - Commercial software
- Often uncovers underspecified interfaces
- Tools focus on
  - Web applications
  - Network protocols, IPC
  - Command-line arguments, environment variables

Fuzzing

- Limitations
  - Identifies only simple faults with handling of input
  - Does not address well bugs that are triggered by a small number of input values
  - Does not address well sequences of input



Summary

- Testing can
  - **confirm presence** of vulnerabilities
  - (in practice) **not prove absence** of vulnerabilities
- Vulnerability analysis often 50% of evaluation effort
- Methodical development of flaw hypotheses
- Fuzzing
  - Covers large variety of inputs
  - Only suitable for simple bugs