Comp 4350 - Software Engineering II

Lecture 12: Software Security and Testing

Dr. Shaowei Wang

Administrative Items

- Technical Seminar (Week of Oct 31)
  - Sharing knowledge/experience among teams
  - Topics: technology, design choices, challenges, solutions, lessons learned
  - Format: 15 min presentation (13 min talk + 2 min Q&A)
  - Evaluation: usefulness (70% teams, 30% instructor), presentation quality
  - Schedule: Google Sheet
  - Submission: UMLearn -> Assignments -> Technical seminar

Software Security Overview

Security Issues in Code Repositories

- 4 million security flaws in public repos (Source)
- Large scale breaches (e.g., Yahoo! incidents)

Causes of Data Breaches

- Old/unpatched vulnerabilities
- Human errors (e.g., weak passwords, data sharing)
- Malware
- Insider abuse
- Physical theft

Software Security as Quality

- Key Qualities
  - Confidentiality
  - Integrity
  - Authentication
  - Authorization
  - Availability
  - Non-repudiation

Security Levels

- System-wide security safeguards are required; can't be confined to a few levels.

Software Security Assurance (SSA)

- Vital throughout all development phases
- Includes vulnerability screening for both in-house and external (library) software

---

Vulnerabilities, Common Vulnerabilities and Exploits

- Definition: Weaknesses, design flaws, or implementation bugs
- Results: Financial loss, data breach, risk to human life
- Detection methods: Static analysis, penetration testing, code reviews

Common Vulnerabilities

- SQL injection
- Cross-Site Scripting (XSS)
- Unsecured object references
- Inadequate security configuration
- Unsecure cryptographic storage

OWASP Foundation

- Provides information on common vulnerabilities
- Reference: OWASP

SQL Injection and Prevention

- Vulnerable code might execute attacker-provided SQL
- Prevention Techniques
  - Validation of user inputs
  - Parametrized queries (e.g., Java Prepared Statement)

Cross-Site Scripting (XSS)

- Attack delivery via legitimate web pages/applications
- Injection of malicious script into web forums, message boards, pages with comments
- Prevention: Input sanitization and HTML escaping

Penetration Testing (PT)

Definition and Types

- A simulated authorized attack to unveil vulnerabilities and exploit severity
- Types: Vulnerability scanning, security audit, penetration tests

Vulnerability Scanning - Static Analysis

- Early detection of vulnerabilities

- Techniques include data flow analysis, control flow graphs

Rule-Based and Machine Learning-Based Analysis

- Rule-based: Application of expert-designed patterns to detect anti-patterns
- Machine learning-based: Use of algorithms for vulnerability detection (e.g., DeepVD)

Static Analysis - Pros and Cons

- Pros: Automatable, supports DevOps (CI/CD), useful for easily detectable vulnerabilities
- Cons: False positives, failure to detect certain types of vulnerabilities, difficulty with dependency resolution

Criteria for Selecting Tools

- Language support
- Types of detectable vulnerabilities
- Dependency resolution capability
- Integration with IDEs
- Ease of installation and use

Defense Mechanisms

Self-Protective Software

- Proposals for self-protective measures include RASP (Runtime Application Self-Protection) and ABSP (Architecture Based Self-Protection).

Review Best Practices

- Establish threat models
- Combine automated tools with expert manual analysis
- Review for security, not for risk assessment

Vulnerability Discovery and Assessment

- Example findings: Weak/default passwords, reused passwords, outdated patches, misconfigured servers

---

Security Testing Process (ST)

- Aligns with the Software Development Life Cycle (SDLC)
- Derivation of security requirements from threat models
- Importance of early testing and rigorous process

OWASP Testing Framework

During SDLC:

- **Before development**: Define SDLC, review policies, develop metrics, ensure traceability
- **Design**: Review security requirements, revise designs/architecture, create UML and threat models
- **Development**: Conduct code reviews and inspections, perform static analysis
- **Deployment**: Execute penetration tests, test configuration management
- **Maintenance/Operations**: Review operational management, periodically check system health, verify changes

---

Penetration Testing (PT) Phases

1. **Reconnaissance**: Gather target system information
2. **Scanning**: Use tools to gather more detailed information (e.g., open ports)
3. **Gaining Access**: Exploit vulnerabilities
4. **Maintaining Access**: Remain in the system and extract information
5. **Covering Tracks**: Erase evidence of compromise
6. **Escalation**: Use compromised information to exploit further resources

Examples of Attacks

- Insecure direct object references
- Failure to restrict URL access

Scenario and Report Examples

- Offensive Security - Example: MegaCorp One's compromised system detailed through attack phases

---

Conclusion and Reminders

- The importance of software security cannot be understated, and it has to be an ongoing process throughout the software lifecycle. The critical takeaway is the need for proactive security assurance mechanisms, coupled with effective vulnerability detection and rigorous penetration testing methodology. Remember to utilize the provided tools and frameworks, such as those offered by OWASP, to enhance the security and integrity of software systems.

Class Notes

Slide 1

Title: Comp 4350 Software Engineering II Lecture 12
Dr. Shaowei Wang

Slide 2

Title: Administrative items
Technical seminar (the week of Oct 31)

The seminar is to share knowledge, experiences, and challenges faced by teams, lasting 15 minutes. Each team will be evaluated in two aspects:

1. Usefulness of the seminar
2. Quality of the presentation
   Google Sheet schedule: Link
   Submit slides to UMLearn -> Assignments -> Technical seminar

Slide 3

Title: Agenda
Software Security
Static Analysis
Security Tests

Slide 4

Title: Security issues are very prevalent in online code repositories

- 4 million security flaws in public repos.
  From Spiceworks

Slide 5

The first announced breach, reported in September 2016, affected over 500 million Yahoo! user accounts.
A separate data breach in August 2013 impacted all 3 billion Yahoo! user accounts.
Both breaches are considered the largest discovered in the history of the Internet.

Slide 6

Old, unpatched vulnerabilities.
Human errors: bugs, weak passwords, private data sharing.
Malware: cause minor problems but in large quantities.
Insider abuse: the most dangerous vulnerabilities requiring internal security measures.
Physical theft: personal devices that must be physically protected too.

Slide 7

Title: Software Security As Quality

Slide 8

Security is the quality that controls the system's data access according to authorization levels.
Software security as quality

Slide 9

Security must be guaranteed at various system levels to prevent vulnerabilities from affecting higher levels.

Slide 10

Security must be ensured throughout development phases for both our software and external libraries.

Slide 11

Title: Vulnerabilities

Slide 12

A vulnerability is a hole or a weakness in the application that allows attackers to cause harm.
Vulnerabilities can result in lost money, private data, or worse.

Slide 13

SQL injection
Cross Site Script (XSS)
Unsecured object references
Bad security configuration
Unsecure cryptographic storage
Common Vulnerabilities at OWASP

Slide 14

Title: SQL injection

Slide 15

This SQL query selects all customers where the username is blank or equals true.

Slide 16

Title: Generated vulnerable code from ChatGPT
Results from ChatGPT on 2/12/2023
SQL injection: UserId = "105 OR 1=1", returns ALL rows in the "users" table.

Slide 17

Title: SQL injection prevention techniques
Using parametrized queries

Slide 18

Title: Parametrized queries
Pre-compile SQL statements with placeholders for parameters.

Slide 19

Safe Java Prepared Statement Example:

Slide 20

Slide 21

Title: Cross-site Scripting (XSS)
Client-side code injection attack.

Slide 22

Title: Stored XSS

Slide 23

Attacker injects malicious code to run in a victim’s browser, gaining access.

Slide 24

Title: Cross-site Scripting Attack Vectors

Slide 25

Slide 26

Slide 27

Title: How to Prevent XSS
Sanitize user input to prevent code execution.

Slide 28

Slide 29

Title: Escaping HTML
Convert characters into their HTML entity equivalents.

Slide 30

Title: Unsecured object references

Slide 31

Use proper access controls and validate user input to prevent unauthorized access.

Slide 32

Slide 33

Slide 34

Title: Security tests
Processes to ensure security.

Slide 35

Security testing ensures controls for confidentiality, integrity, authentication, etc.

Slide 36

Vulnerability scanning
Security Audit
Penetration tests

Slide 37

Title: Vulnerability Scanner – static analysis
Code scanning before deployment.

Slide 38

Static analysis techniques:

- Data flow analysis
- Control Flow Graph

Slide 39

Slide 40

Title: Rule-based
Based on pre-defined rules developed by experts.

Slide 41

Slide 42

Slide 43

Slide 44

Advantages: Can run multiple software, helps DevOps, detects easy vulnerabilities.
Disadvantages: Several types of vulnerabilities are hard to detect, increased false positives, configuration issues not detected.

Slide 45

Criteria for selecting static analysis tools.

Slide 46

Slide 47

Slide 48

Slide 49

Slide 50

Title: Code security review
Ensuring security checks are implemented.

Slide 51

Preparation for code security review.

Slide 52

Slide 53

Title: Penetration tests (PT)

Slide 54

Slide 55

Best practices for reviews.

Slide 56

Reports from PT scenarios:

- Offensive Security
- MegaCorp One

Slide 57

Scenario detailing an attack and the compromised system.

Slide 58

Results from PT reports - identified vulnerabilities and risks.

Slide 59

Title: Tutorial for Penetration Tests

Slide 60

Security testing process follows SDLC. Early testing is crucial.

Slide 61

Derive and elicit security requirements: positive, negative, and abuse cases.

Slide 62

Self-protective software faces challenges in software security.

Slide 63

Self-protection can be implemented via code insertion or external systems.

Slide 64

Advantages and disadvantages of self-protection.

Slide 65

PT phases: Recognition, Scanning, Get access, Maintain access, Cover tracks, Climbing.

Slide 66

Insecure direct object references: modifying URLs for unauthorized access.

Slide 67

Failure to restrict URL access: passing roles or privileges via URLs without verification.

Slide 68

OWASP Testing Framework stages across SDLC phases.