Exam #2 Study Guide

CPSC 348 – Computer Security
Fall 2020

Injection/Application Security

- **Injection.** Describe injection attacks and explain why they are so bad in terms of detectability, exploitability, prevalence, and impact.
- **Sanitation and validation.**
  - Describe sanitation, validation, and the similarities and differences between the two.
  - Compare and contrast the two in terms of security vs. usability.
- **Security design principles.**
  - Explain the importance of the principle of least privilege as a mitigation against injection attacks.
  - Explain the importance of the principle of fail-safe defaults with regard to whitelists vs. blacklists.
- **Attacks.** For each of the following attacks:
  - Explain what it is.
  - Explain how it works.
  - Given the description of a system, identify which attack it is vulnerable to.
  - Given a simple code sample that is vulnerable to the attack, give a malicious input that will execute the attack, and explain what will happen when the program processes your malicious input.
  - Explain one of the consequences we discussed in class. What is the attacker able to do? What does she gain from the attack?
  - Modify the code to protect against the attack.
  - Attacks:
    - *Injection attacks*:
      - SQL injection
      - OS command injection
      - Path traversal
    - *Web-based attacks*:
      - Cross-site scripting (XSS)
      - Cross-site request forgery (CSRF)
    - *Other attacks*:
      - Buffer overflow
      - Integer overflow

Security Configuration

- **Terms.** Understand and explain the terms “hardening”, “attack surface”, and “attack vector”.
  - Explain how the concept of hardening relates to security vs. usability.
- **Mitigating vulnerabilities with server configuration.** For each of the following attacks, explain which server configuration protects against it and how.

    - OS command injection
    - Path traversal
    - XSS
    - CSRF
    - Session hijacking
    - Protocol downgrade
    - Denial of service (DoS)
    - Credential stuffing
- **HTTP security headers.** For each of the following HTTP headers, explain what it does and which vulnerabilities it protects against.
    - Content-Security-Policy
    - Strict-Transport-Security
- **HTTP cookie security attributes.** For each of the following HTTP cookie attributes, explain what it does and which vulnerabilities it protects against.
    - Secure
    - HttpOnly
    - SameSite
- **Encrypted connections.** Explain one of the ways in which servers can force encrypted connections and disallow unencrypted connections.
- **General server security configurations.** For each of the following server configurations, explain how it makes it more difficult for an attacker to detect or exploit any vulnerability.
    - Proper error-handling
    - Firewalls
    - Removing unnecessary apps, services, features, and user accounts
- **Security design principle: Defense-in-depth.** Explain what defense-in-depth is, give an example, and explain why it matters.
- **Automation.** Explain why automation is so important in security configuration, and give a basic explanation of what can be automated and what can't.
    - Secure configuration of new systems
    - Security monitoring of existing systems
    - Patching
- **Patch management.**
    - Explain what patch management is and why it is crucial for maintaining security.
    - Explain why systems aren't configured to automatically apply patches as soon as they are released.
    - Explain how standards organizations play a key role in patch management.
    - Explain the best practices for patch management.