Midterm Review

Logistics

- In class:
  - On time: 4:10-5:30pm Wed
- 1 8x11 page cheat sheet allowed
- Special requirements: see TA

Scope

- Part I, II, III
  - Software Security
  - Secure Architecture Principles
  - Cryptography
- Material covered in lectures & labs
- Exam will cover breadth

Part I Software Security (I)

- Buffer overflow vulnerability and attack
  - what is a buffer overflow vulnerability?
  - what types of programming errors and issues in programming language design that cause buffer overflow vulnerabilities?
  - how is a buffer overflow vulnerability exploited?
  - what is a control hijacking attack?
  - what are the important steps for a control hijacking attack?
  - what are the different types of control hijacking attacks?
  - what is a NOP slide?
  - what is the difference btw code injection and arc injection?
  - what is data hijacking?

Part I Software Security (II)

- Memory corruption defenses
  - what is a NX-bit?
  - what is a stack canary?
  - what is ASLR?
  - what are the pros and cons of each defense mechanism? Know which defenses are applicable to code injection / arc injection, stack / heap / exception handlers

- Know about the following vulnerabilities:
  - Format string vulnerabilities
  - User after free
  - Double free
  - Integer overflow
  - Implicit cast

Part I Software Security (III)

- Vulnerability / Memory Safety Analysis Techniques
  - Overview
    - How do the different techniques differ in terms of soundness and completeness? Why?
    - How do you ensure memory safety by inserting assertions in code?
  - Blackbox Fuzzing
    - what is the purpose of fuzzing?
    - how do blackbox fuzzers work?
    - Code coverage metrics: line / branch / path coverage
  - Dynamic Symbolic Execution (DSE)
    - how does DSE systematically explore paths in a program?
    - Given an explored path, what is its path constraint formula in Static-Single Assignment (SSA) form?
    - how does a whitebox fuzzer (based on DSE) detect a vulnerability?
    - what are the common corner cases where bugs arise (e.g., arithmetic overflow, most negative integer, etc).

Part I Software Security (IV)

- Static Analysis - Abstract Interpretation
  - How do you use the interval domain to analyze a program?
  - How and when does it introduce false positives?
- Manual program verification / manual code reasoning
  - Given a program or a statement, what is its precondition and postcondition?
  - Given a loop, what is its loop invariant?
  - How do you use preconditions/postconditions/loop invariants to prove an assertion is correct (or not)?

Part II: Secure Architecture Principles (I)

- What is the principle of least privilege? why is it an important security design principle?
- Access Control and Capabilities
  - What is an access control list (ACL)?
  - What is a capabilitiy?
  - When is access control based on ACL vs capability?
  - How does the Unix access control (file permissions) work?
- How does setuid work? how is it used to allow a program to drop privilege?
- Privilege separation:
  - What is privilege separation? what are the components in privilege separation?
- What is TOCTOU vulnerability?

Part II: Secure Architecture Principles (II)

- Isolation
  - what is a reference monitor?
  - Isolation through Jail
    - What is the purpose of chroot and jail?
    - What do they guarantee and not guarantee?
    - How jailbreak can happen
  - Isolation through system call interposition
    - What is the purpose of system call interposition?
    - How is system call interposition implemented?
    - What are the potential implementation pitfalls?
  - Software based fault Isolation
    - What is software based fault isolation?
    - When do you need SFI?
    - how do you implement SFI for RISC architecture?
    - how do you verify a piece of code is correctly compiled with SFI?

Part III Cryptography (I)

- Symmetric Key Cryptography
  - What is encryption?
  - What is authentication?
  - one-time pad re-use
  - What is an initialization vector?
  - CTR mode
  - What happens when IV repeats?
  - CBC mode
  - What happens when IV repeats?
  - ECB mode
  - MAC, HMAC (at a high level, not construction)

Part III Cryptography (II)

- Secure Communication
  - Know how to construct a secure channel (e.g. encryption + auth + nonce)
    - What is a nonce?
  - What are the capabilities of a passive attacker?
  - What crypto functions are necessary to secure against a passive attacker?
  - What are the capabilities of an active attacker?
  - What crypto functions are necessary to secure against an active attacker?

Part III Cryptography (III)

- Public Key Cryptography
  - RSA construction
  - Key distribution problem
  - Digital signatures, certificates, certificate revocation

Feedback Survey

- Give us feedback
  - Improve the class together