Intro to IT Security

CS306C—Fall 2022

Prof. Antonio R. Nicolosi

Antonio.Nicolosi@stevens.edu

Introduction to IT Security

Grading (tentative)

10%: In-class quizzes & class attendance

20%: Lab attendance & performance

20%: Homeworks

25%: Midterm

25%: Final

- No make-up or substitute exams!

List of Topics

We'll survey a broad range of computer security issues

- General concepts
- Cryptography
- Network Security
- Software Security

Introduction to Computer Security

- General concepts
  - Vulnerabilities, threat models, adversaries, security goals

Cryptography

- Information security goals
  - Secrecy, integrity, authentication
- Basic primitives
  - Hash functions, one-way functions
- Symmetric primitives
  - Encryption, message authentication codes
- Public-key primitives
  - Asymmetric encryption, digital signatures
- Cryptographic protocols
  - Key exchange, key distribution, authentication

Network Security

- Networking background
- Network security threats
  - Virus, worm, spoofing, snooping, denial of service
- Network security defenses
  - Crypto (SSL/TLS, IPSec), firewalls, intrusion detection systems

Software Security

- Buffer overflow and other implementation flaws
- Isolation and sandboxing techniques
- Software engineering best practices
  - Defensive programming, fail-safe defaults, least-privilege and privilege-separation principles

Additional Topics

If time permits, we will cover some additional topics

- Anonymity and privacy
- Cloud computing
- E-voting, e-cash, e-commerce
- Security and the law; ethical issues
- Special requests?

What this course is and is not about

- This course is not about
  - How to “hack” computer systems
- This course is about
  - How to analyze, understand and evaluate the security of computer systems
  - How to effectively use cryptography in the system you build

Definitions

- Security: Techniques to control who can access/modify system
- Principal: Unit of accountability in a system (*e.g.,* a user)
- Access control: Techniques to restrict operations to certain principals
  - Authentication: Verification of the identity of the principals making a request
  - Authorization: The granting of a request to a principal

Attacks on Security

- Violation of secrecy (a.k.a. confidentiality)
  - Attacker reads data without authorization
- Violation of integrity
  - Attacker modifies data without authorization
  - *e.g.,* Attacker modifies data on disk
  - *e.g.,* Attacker modifies network reply to “read file” request
- Violation of availability (a.k.a denial of service)
  - Overload the system; cause a deadlock
  - Attacker makes system unavailable to legitimate users
  - Trigger security mechanism (*e.g.,* wrong bank PIN 3 times)

Security is a Negative Goal

- Ensure nothing happens without authorization
  - How do you reason about what a system will not do?
- Must first specify who is authorized to do what
  - One must specify a well-defined policy
- Trusted computing base (TCB):
  Collection of software upon which security relies
  - Should have well-specified TCB
  - Minimize TCB–less code in which to find security holes

Protection Mechanisms

- Hardware protection
  - ROM and/or read-only disks can provide booting
  - Processor separates user and kernel code
  - User code cannot access devices, write kernel memory, . . .
  - Kernel boundary separates TCB
- Software protection
  - Kernel or server sanity-checks and restricts call arguments
  - Bytecode interpreter restricts functionality (e.g., JVM)
- Protection of network traffic with cryptography

Policy

- Policy: The goal of the computer security effort
  - Human intent—originates from outside the system
- Often expressed in terms of subjects and objects
  - Subject: Entity making requests (= principal)
  - Object: Abstraction to which access is requested (e.g., a file, a page of memory, a serial port)
  - Each object supports different kinds of access (*e.g.,* read or write file, change permissions, . . .)
- Access control: Should operation be allowed?
  - What principal is making the request? (Authentication)
  - Is the operation permitted to the principal? (Authorization)

Examples of Access Control

- Un-networked machine in a locked room
  - Policy: Only users with keys can access the computer
    - Don't overlook old-fashioned solutions if they apply!
- Bank ATM card
  - Policy: Can only withdraw money if it's in your account
  - Authentication: Owner must possess card & know PIN
  - Authorization: Database tracks account balances
- Private Unix file
  - Policy: only owner can read
  - Authentication: Password to run software as user
  - Authorization: Kernel checks permissions bits on file

Authentication Mechanisms

- User provides password to server
- User proves possession of device (e.g., smart card)
- Cryptographic protocol proves possession of private key
- Biometrics (e.g., fingerprint)—Often misused
  - Should your laptop send your fingerprint to the server?
- Don’t forget server-to-user authentication!
  - Fake login screen or ATM machine gets user’s password

Authorization Mechanisms

- Access control lists on objects
  - A list of principals and permitted operations for each
- Self-authenticating capabilities
- Signed digital certificates
- Object access in type-safe language
  - Authentication: Some procedure returns a pointer to object
  - Authorization: Possession of pointer allows access
- Issues of authorization mechanisms
  - Can those with access futher delegate access?
  - How easy is it to revoke access to an object?

Interactions between objects

- Must consider any operations could violate policy
- Example: Only professor can change grades
  - OS refuses write requests from others to grades file (easy)
  - But can attacker change program professor runs to edit file?
  - Can attacker change directory so owner reads different file?