Computer Security

Vermont Technical College

CIS-4040

Peter C. Chapin

What We Cover

- This is a fundamentals course
  - Basic concepts of security
  - Important security tools (cryptography)
  - Important security protocols
  - Network vs host security
- This course is not…
  - … about specific vulnerabilities or attack methods
  - … about specific tools or techniques

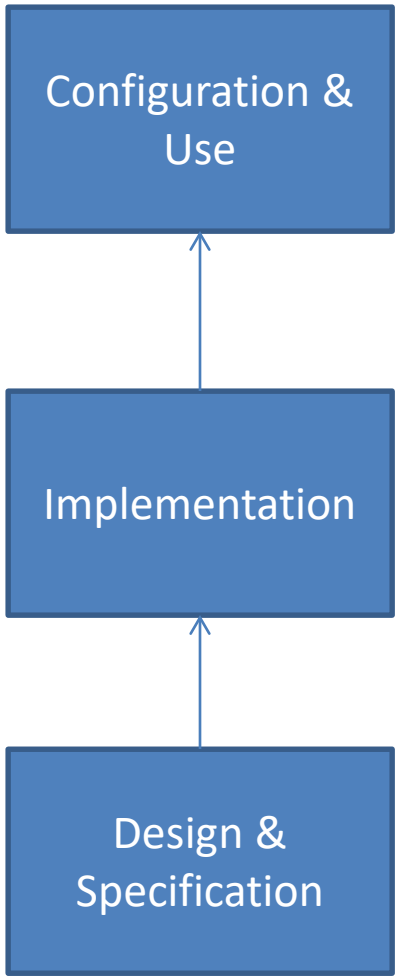
What is Computer Security?

- *A computer system is secure if unexpected behavior cannot occur or is not problematic*.
  - Unexpected behavior includes
    - Revealing data to unauthorized entities.
    - Letting unauthorized entities modify data.
    - Ordinary software faults.
    - Hardware failure.
  - **Very broad definition**! It overlaps software engineering and system administration.

What is Computer Security?

- *A computer system is secure if it behaves as expected when attacked by an unauthorized intelligence*.
  - Focuses on the issue of **malicious attack** (bad guy)
  - Random errors are not a security problem unless they introduce an exploitable vulnerability
  - **More intuitive definition**, but it overlooks a practical reality: *Data loss is data loss no matter what causes it*.

Security Layers

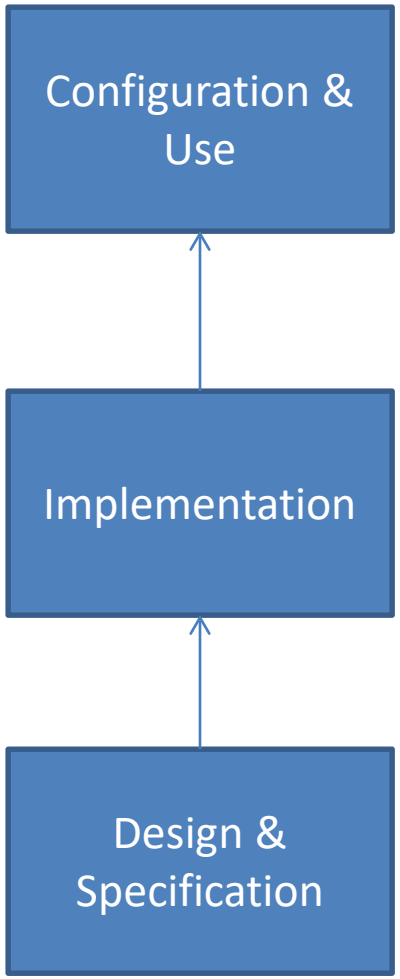


Is the system configured and used properly?
(Administration)

Is the system built correctly?
(Engineering)

Is the system secure in principle?
(Security theory)

Complexity is Bad for Security



Complex systems are hard to use.

Complex systems are hard to build.

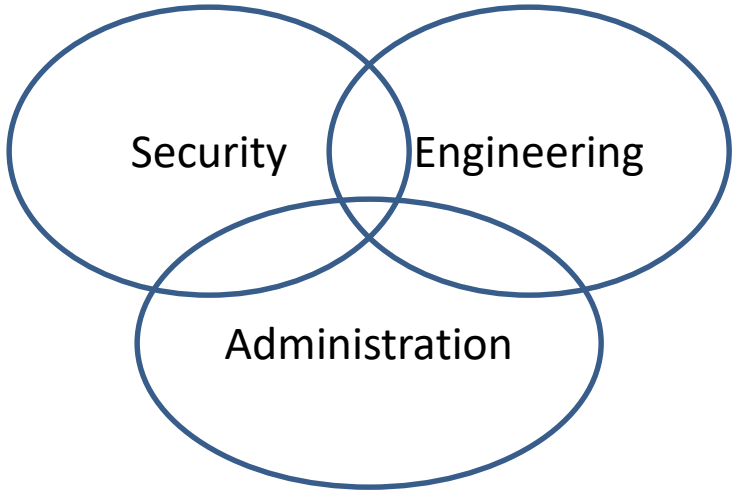
Complex systems are hard to understand.

Examples of Excessive* Complexity

- NTFS (Windows file system) permissions
  - Many complex interacting options
- IPsec (IP Security Protocol)
  - Too many ways of doing essentially the same thing. Too many interacting options.
- Linux `iptables` configuration
  - Like many firewalls, provides a large number of features. How can administrator be sure it's ok?

* Complexity is "excessive" if the corresponding capability is either unnecessary OR can be obtained in another, simpler way.

Security, Engineering, Administration



- Security + Engineering
  - Design
  - Reliability
- Security + Admin
  - Software configuration
  - Contingency planning
- Engineering + Admin
  - Administrative tools
  - Hardware support

Notes on Terminology

- “Insecure” vs “Unsecure” vs “Unsecured”
  - Insecure is an emotional state. Unsecure is not a word. Unsecured implies no security activated.
- “Hacker”
  - A hacker used to be a good guy. Then hackers became bad. Now they are good again. *Don’t use the term, the meaning is ambiguous.*
- “Adversary” vs “Attacker”
  - Adversary is more neutral.

Alice and Bob

- Security community has traditionally used “Alice” and “Bob” instead of A and B in examples. I will continue this tradition.

Security Services

- “Service” in this context is a type of security, not a server
- Q: “Is your system secure?”
  - Wrong answer: “Yes” (or “No”)
  - Right answer: “Secure in what sense?”
- Security is **not** a Boolean attribute.
  - Many possible security services exist.
  - A system might be strong in some ways, weak in others.
  - *Match the security services you use to your needs!*

“Big Two” Security Services

- **Confidentiality**
  - The property of blocking unauthorized users from reading data. (Common tool: Encryption)
- **Data Integrity**
  - The property of blocking unauthorized users from writing data. (Common tool: Digital signatures)

These two services are *duals* of one another. They have an intimate theoretical relationship that we will explore as the course progresses.

Other Security Services

- Authentication
  - The ability to determine the *identity of a principal*
- Authorization
  - Determining *what a principal can do* once authenticated
- Anti-Replay
  - The ability to detect when an old transaction is inappropriately resubmitted for processing
- Sequence Control
  - The ability to detect when the ordering of events has been rearranged
- Availability
  - The ability to continue working despite attempts to shut you down

Example

- *Alice sends Bob packets over the network. Alice encrypts and signs the packets so…*
  - Confidentiality, data integrity, and authentication are provided.
- BUT…
  - Without sequence control an attacker could rearrange the packets
  - Without anti-replay an attacker could send the packets again

Attacker Models

- Passive
  - Attacker only able to look at data, but not touch it. “Observe, but do not interfere.”
- Active
  - Attacker able to modify, insert, remove, reorder data. “Go ahead and interfere.”

It’s important to use the right model when analyzing a security system. **Be realistic**. *No security is sufficient against an attacker with god-like power!*

Dolev-Yau Attacker Model

- Common when analyzing network protocols
  - Attacker can…
    - … read every message everywhere on the network
    - … modify any message anywhere on the network
    - … block or reorder messages at will
  - Attacker cannot…
    - … break any cryptographic methods in use
    - … access any information on any of the hosts
  - “Attacker carries the message”

Don’t Use “Security Though Obscurity”

- *Always assume adversary has full knowledge of the methods and algorithms used.*
  - They will figure them out eventually no matter what you do
  - Assuming your methods and algorithms remain secret is “security through obscurity.”

Summary

- *A computer system is secure if* it behaves in the expected way.
- *Security concerns overlap with* those from engineering and system administration.
- *Security must be applied at all levels*: Design, implementation and deployment.
- *Complexity is bad for security*. Unfortunately feature-hungry users gravitate toward complex systems.
- *Security services define the kinds of security* one might be interested in having.
- Before analyzing security, *understand the assumed capabilities of the attacker*.
- *Always assume the attacker knows your methods*.