

Applied Cryptography and Network Security

Adam J. Lee
adamlee@cs.pitt.edu
6111 Sennott Square

Lecture #1
January 7, 2014



University of Pittsburgh



Administrivia [UPDATE]

Applied Cryptography and Network Security

- CS 1653
- TH 11:00 - 12:15
- SENSQ 6110

Instructor:

Prof. Adam J. Lee
adamlee@cs.pitt.edu
6111 Sennott Square
OH: TH 2:30 – 4:00 PM

Teaching Assistant:

Bill Garrison
bill@cs.pitt.edu
6503 Sennott Square
OH: *TBD*

Web:

<http://www.cs.pitt.edu/~adamlee/courses/cs1653>



Why should I take this course?

My goal is to help you answer the following types of questions:

- How do various types of cryptosystems work?
 - Symmetric key
 - Public key
 - Threshold
- What sorts of cool things can we do with cryptography?
- What does it mean to be secure? Private?
- How can we design functional **and** secure systems?

(Some) Questions that we will **not** answer:

- How do I design new cryptographic algorithms?
- How do I break into the *[your favorite software]* system?
- ...



Topic Outline

Introduction

- CIA properties
- Vulnerabilities, threats, and attacks
- Design principles

Cryptographic tools

- Symmetric key algorithms
- Hashing
- Public key cryptography
- Secret sharing and threshold cryptography

Applications of cryptography

- User authentication
- Digital certificates
- Key distribution
- Protecting digital transactions
- Etc.



Topic Outline

OS and Application Security

- Integrity of mechanism and OS protection
- SQL injection
- Malicious logic

Network Security

- Attacking protocols
- Denial of service
- Intrusion detection

Research Issues in Privacy

- Private messaging
- Private routing
- Data privacy



The web is your friend!

CS 1653: Applied Cryptog

people.cs.pitt.edu/~adamlee/courses/cs1653/lectures.html

Lecture Schedule (Subject to Change)

[Home] [Policies] [Homework] [Project]

Notice: Students are responsible for reading assigned materials prior to the lecture in which they will be discussed. Copies of the lecture slides are available over AFS in the [/afs/cs.pitt.edu/usr0/adamlee/public/cs1653/lectures](afs/cs.pitt.edu/usr0/adamlee/public/cs1653/lectures) folder. As per the [course policies](#), lecture slides are for your personal use only.

Lecture #	Date	Topics	Readings	Other
1	1/7 (Tue)	Administrivia and course introduction	-	-
2	1/09 (Thu)	Design Principles	[SS75] pp. 1279-1283	<i>Project 1 out</i>
3	1/14 (Tue)	Cryptography basics and classical cryptography	KPS 2	-
4	1/16 (Thu)	Symmetric key cryptography, modes of operation	KPS 3.1-3.2, 4	-
5	1/21 (Tue)	Case study: AES	KPS 3.5	<i>Project 1 due Project 2 out</i>
6	1/23 (Thu)	Hashing	KPS 5	-
7	1/28 (Tue)	Public key cryptography background	KPS 6.1-6.2, 6.4, 6.8	-
8	1/30 (Thu)	Case study: RSA	KPS 6.3, 6.6, 7	-
9	2/4 (Tue)	RSA wrap-up; secret sharing	[S79]	-
10	2/6 (Thu)	Threshold cryptography; User authentication	KPS 9, 10	-
11	2/11 (Tue)	Handshakes I	KPS 11	<i>Project 2 due Project 3 out</i>
12	2/13 (Thu)	Handshakes II	KPS 12, <i>Optional:</i> [L81]	-
13	2/18 (Tue)	Strong password protocols	-	-
14	2/20 (Thu)	Kerberos	KPS 13, 14	-
15	2/25 (Tue)	PKI models	KPS 15	-
16	2/27 (Thu)	Real-time communication security	KPS 16	-
17	3/04 (Tue)	Midterm (in class)	-	-
18	3/06 (Thu)	SSL	KPS 19	<i>Project 3 due</i>
-	3/11 (Tue)	Spring break. No class.	-	-
-	3/13 (Thu)	Spring break. No class.	-	-
19	3/18 (Tue)	Review/Discussion	-	<i>Project 4 out</i>
20	3/20 (Thu)	Case study: Breaking MD5	Optional: [SS+081]	-



How will my grade be calculated?

Overall Breakdown:

- 25% Midterm
- 25% “Cumulative” Final
- 30% Project
- 10% Homework and in-class quizzes
- 10% Attendance/Participation/Subjective

*All material before
spring break*

*Note that this is
worth more than
your exams...*

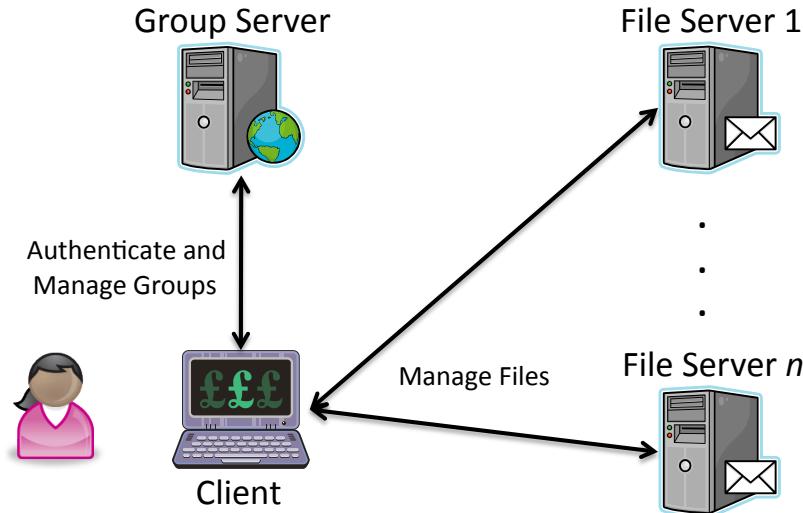
In general, you can expect that:

- Exams will test how well you think about problems, not how well you memorize facts
- Homework assignments will be few and far between.
 - Short programming assignments
 - Experience reports
 - Etc.

Our project will take a security engineering approach to applying the material covered in class



The Scenario:



The Gist:

- Groups of 2-3 students
- Five phases (i.e., mini-projects) that build upon one another
 - Apply the concepts that we cover in class
 - Test your ability to think about problems faced during the development of “real” products
 - Cultivate your ability to work in groups

How will my final project grade be calculated?



Description	Tentative Due Date	Percent of Grade
Requirements spec.	January 21	10%
Prototype implementation	February 11	25%
Security features I	March 6	25%
Security features II	April 3	25%
Attack and defend	April 17	15%

Important notes:

- Choose your group carefully
 - You **cannot** “quit” your group part way through the semester
 - But you **can** (and must!) report on your division of labor
- This project is progressive in nature
 - That is, later phases build on earlier phases
 - Working hard early will pay dividends later
 - What if your early phases don’t work out so well?



Class Introductions





Why study computer security?





The good ol' days

Security was much less complicated years ago...

Early desktops had

- A single user
- A single address space
- No permissions
- No network
- Limited data-processing abilities



The leading threat in those days was computer viruses



Boot-sector viruses



Executable viruses

Both threats could be controlled reasonably well by using anti-virus software, and basic “software hygiene”



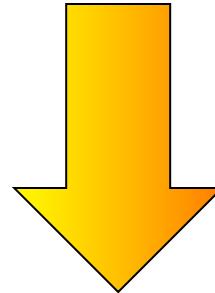
The times, they are a changing



The Internet

Changes introduced

- Constant data exchange
 - Email, web, etc
- Machines no longer isolated
- Looooooooong uptimes



Results

- Faster virus propagation
 - Email: Days to weeks
 - Active worms: Minutes
- Active attacks against end-user hosts



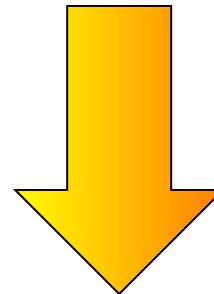
The times, they are a changing



Software/Data Complexity

Changes introduced

- Data formats more complex
- Software functionality bloating
- Boundary between data and executable content is blurred



Results

- “Data hygiene” not as easy
- Plethora of new vulnerabilities
- Vulnerabilities more easily exploited



The times, they are a changing



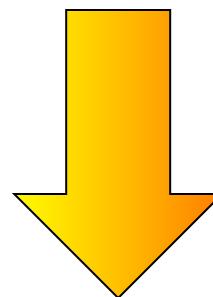
Attacker Motivation

Old days

- Attacks on end hosts not really valuable
- Get “fame” for viruses, etc.

Today

- **Everything** is online!
- Attackers can benefit financially from viruses, worms, and compromised hosts



Results

- Spam and botnets
- Organized online crime

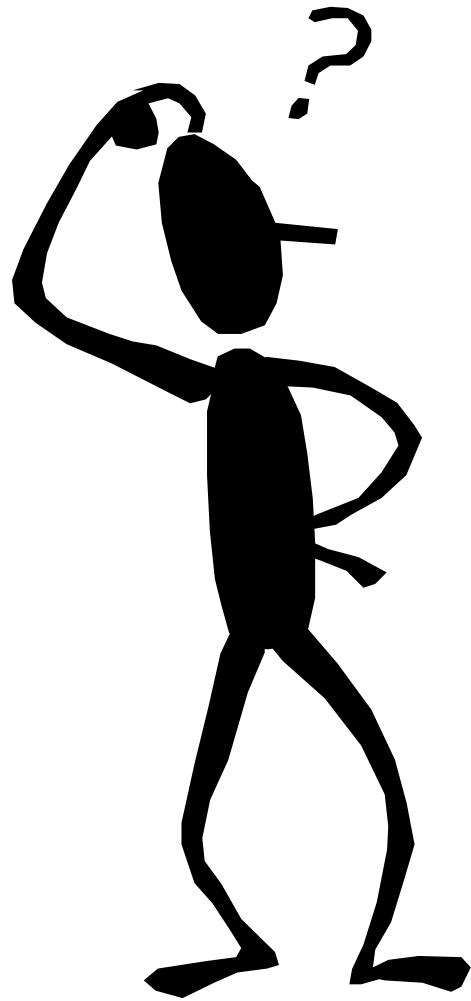
Studying security is more important now than ever before



Security is a challenging problem



What is computer security?





Computer security is typically defined with respect to three types of properties

Can I trust the services that I use?



Integrity



Confidentiality

How do I ensure that my secrets remain secret?



Availability

Am I able to do what I need to do?

Confidentiality refers to the need to conceal information or resources



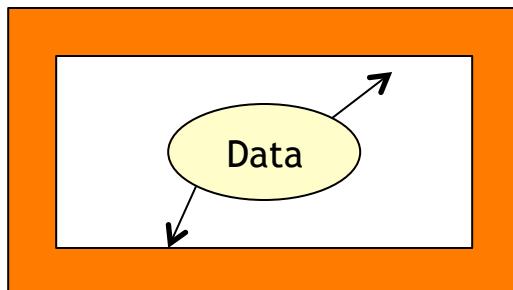
The need for confidentiality arises in both military and civilian information systems

- **Military:** Classified information processing and intelligence
- **Civilian:** Proprietary data, sensitive records, etc.

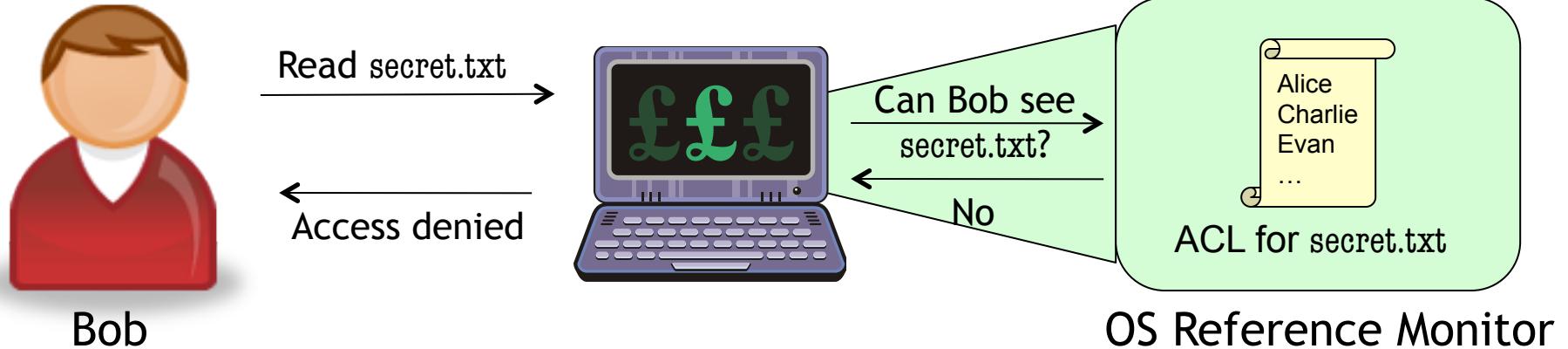
There are many flavors of confidentiality. For example:

- **Data:** “No one should know my bank account balance”
- **Existence:** “No one should know that I shop at XYZ.com”
- **Configuration:** “No one should know what software I run”

The precise notion of confidentiality used in a system depends on the environment in which the system will be used



Access control systems are designed to preserve data confidentiality



Note: We must **trust the OS kernel** in order for the reference monitor approach to work.

Can you think of a solution that avoids this trust?

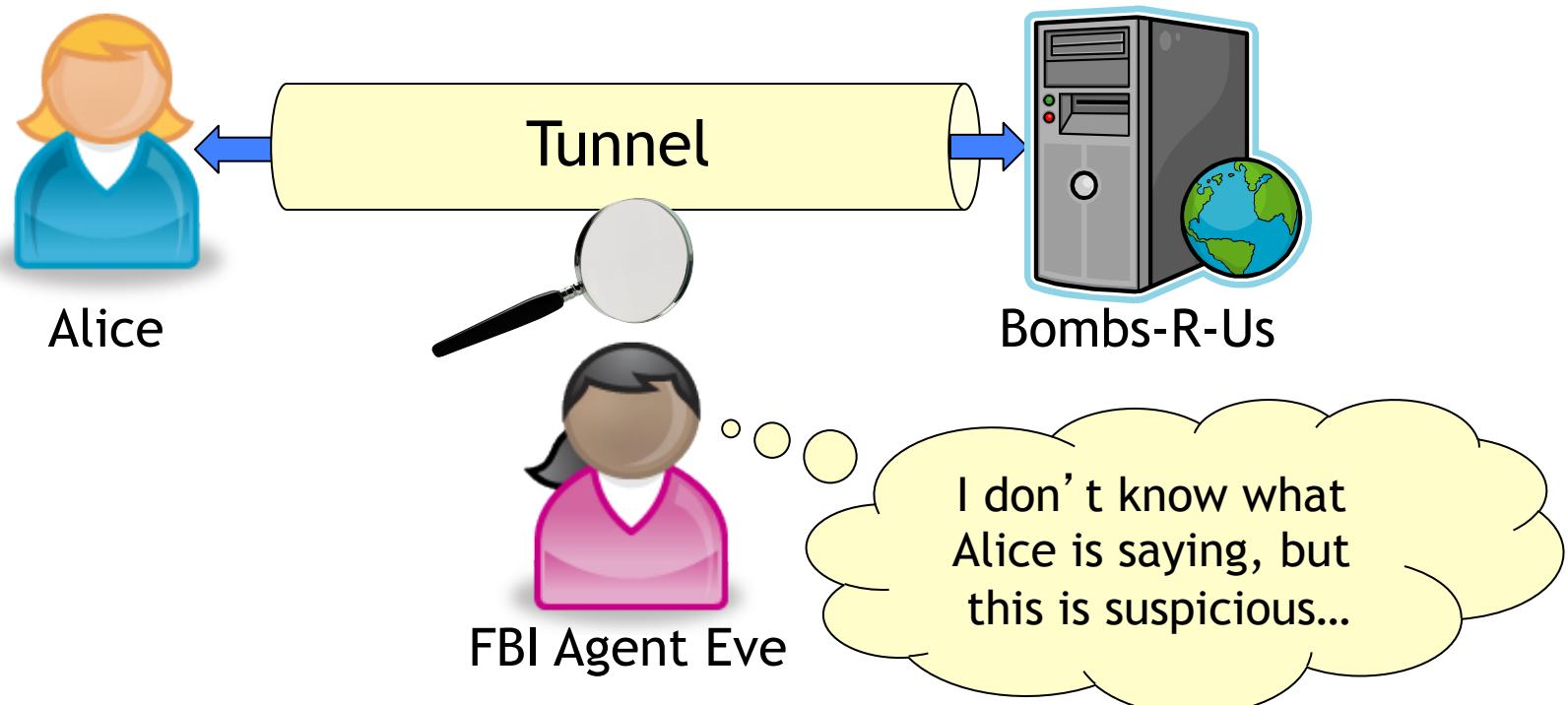
- Cryptographic protection!

Does cryptography guarantee confidentiality?



While cryptography does provide **data** confidentiality, it does not always provide **existence** confidentiality.

Example: SSL





Integrity refers to the trustworthiness of information or resources

Systems are typically concerned with two types of integrity

- **Data integrity:** “Is my bank account balance correct”
- **Origin integrity:** “Was this application really written by Microsoft?”

Integrity mechanisms can either **prevent** or **detect** violations



Pro: Keep system consistent

Con: Difficult to design!

- “Easy” to keep bad guys out
- But how do we identify malicious insiders?

Pro: Easier to design in some cases

- File checksums
- Transaction consistency checks
- Etc.

Con: Causes of violations often remain unknown...



Availability refers to the ability to use information or resources

Systems are usually designed with the **expectation** of certain patterns of usage

- Electricity demand is higher during the work day
- Roads are more utilized during commute hours
- Web sites experience peak traffic during certain hours
- Etc.

By violating these assumptions, the security properties of the system can be altered!

- **Example:** Poorly-protected backup servers

Interesting: Detecting malicious availability violations is non-trivial

- E.g., Botnet-based DDoS attack or “Slashdotted” article?



Discussion

Which of the CIA properties do you feel is most important? Why? Are there other properties that are as important or more important than these?

If security is characterized by the CIA properties, how do we characterize insecurity?



A **threat** is a potential violation of security. I.e., a threat is something that you want to prevent from happening.

Note: A violation **need not occur** in order for a threat to exist!

Threats are sometimes broadly classified into four categories:

- **Disclosure:** Information leakage
- **Deception:** Acceptance of false information
- **Disruption:** Interruption or prevention of correct operation
- **Usurpation:** Unauthorized control of some part of a system

These four classes encompass many, many threats...



A few examples...

Snooping: Unauthorized interception of information

- **Category:** Disclosure
 - **Defenses:** Confidentiality mechanisms
-

Modification or Alteration: Unauthorized change of information

- **Category:** Deception, Disruption, and/or Usurpation
 - **Defenses:** Integrity mechanisms
-

Spoofing: Impersonation of one entity by another

- **Category:** Deception and/or Usurpation
- **Defenses:** Integrity mechanisms (e.g., crypto or authentication services)

Threats simply define the types of insecurity that we are concerned with...



Vulnerabilities are situations or conditions that allow a threat to be realized.

For example:

- Incorrect assumptions about operating environment
- Implementation bugs
- Backdoors
- Incompletely specified policies (e.g., firewalls)
- Corrupted hardware
- Etc.

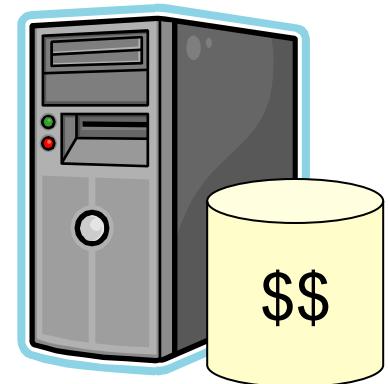
There are many online databases, such as CERT/CC, that list vulnerabilities in widely-used software packages

An **attack** is the exploitation of a vulnerability to realize a threat.



An Example: Bob's Bank

Bob's Bank runs an online banking site that allows its users to check their account balances and pay bills online. Bob's Bank wants to make sure that clients can only see the data in their old account. Due to financial woes, the banking portal is run using an out-of-date web server.



Threat: Seeing another user's bank records

Vulnerability: Unsafe string operations in C code of server

Attack: A buffer overflow that gives the adversary root-level access to the system

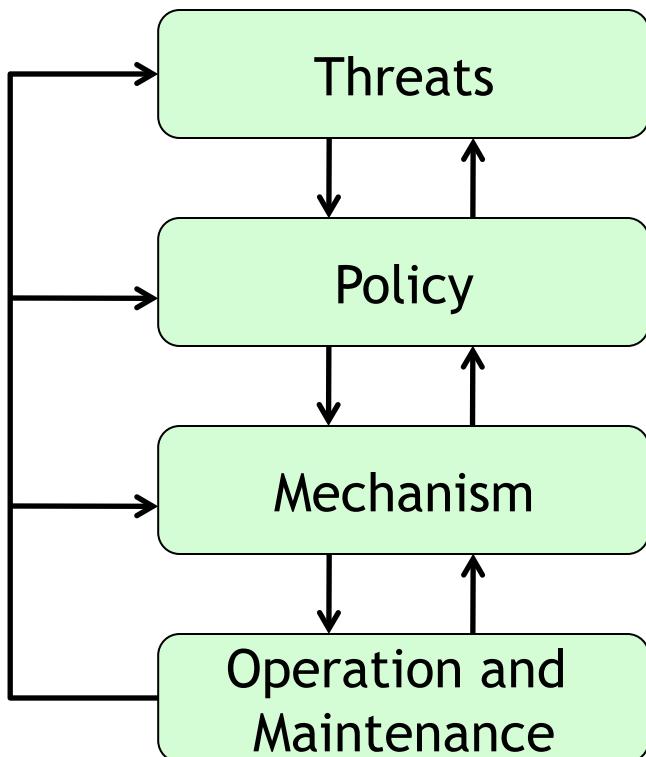


So what?

Security is not an
absolute property!



Security is a process!



Steps:

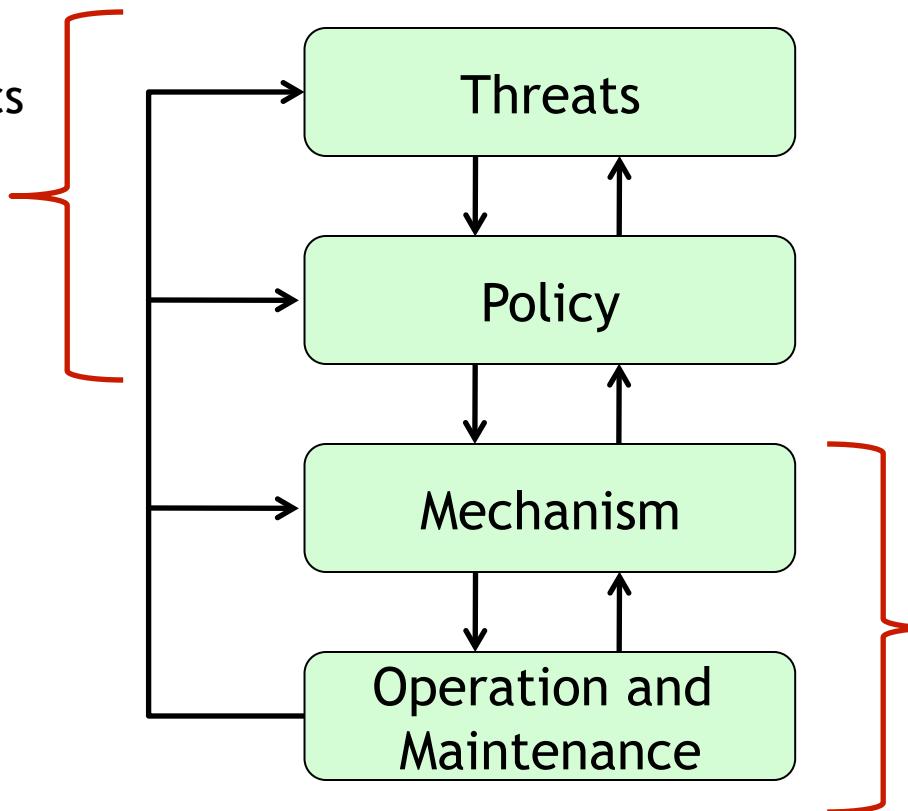
1. Identify threats for the domain of interest
2. Define policies to protect against these threats
 - High-level organizational policies
 - Low-level logical policies
 - Everything in between!
3. Develop mechanisms to enforce these policies
4. Wash, rinse, repeat



Recap and Preview

Today:

- Security basics
- Terminology
- What is security?



Next time:

- Assurance
- System design principles
- Operational issues

Onward: Details, details, details...



Discussion

What are your opinions regarding online vulnerability databases? Do they represent a helpful tool to the community, or simply provide attackers with an easy means of exploiting systems? What do you feel should be the “protocol” used with regard to newly discovered vulnerabilities?