51.505 – Foundations of Cybersecurity

Week 1 – Introduction

Created by **Martin Ochoa** (2017)
Modified by **Jianying Zhou** (2018)

Last updated: 31 Aug 2018

Instructor & TA

Prof. Jianying Zhou

http://jianying.space/

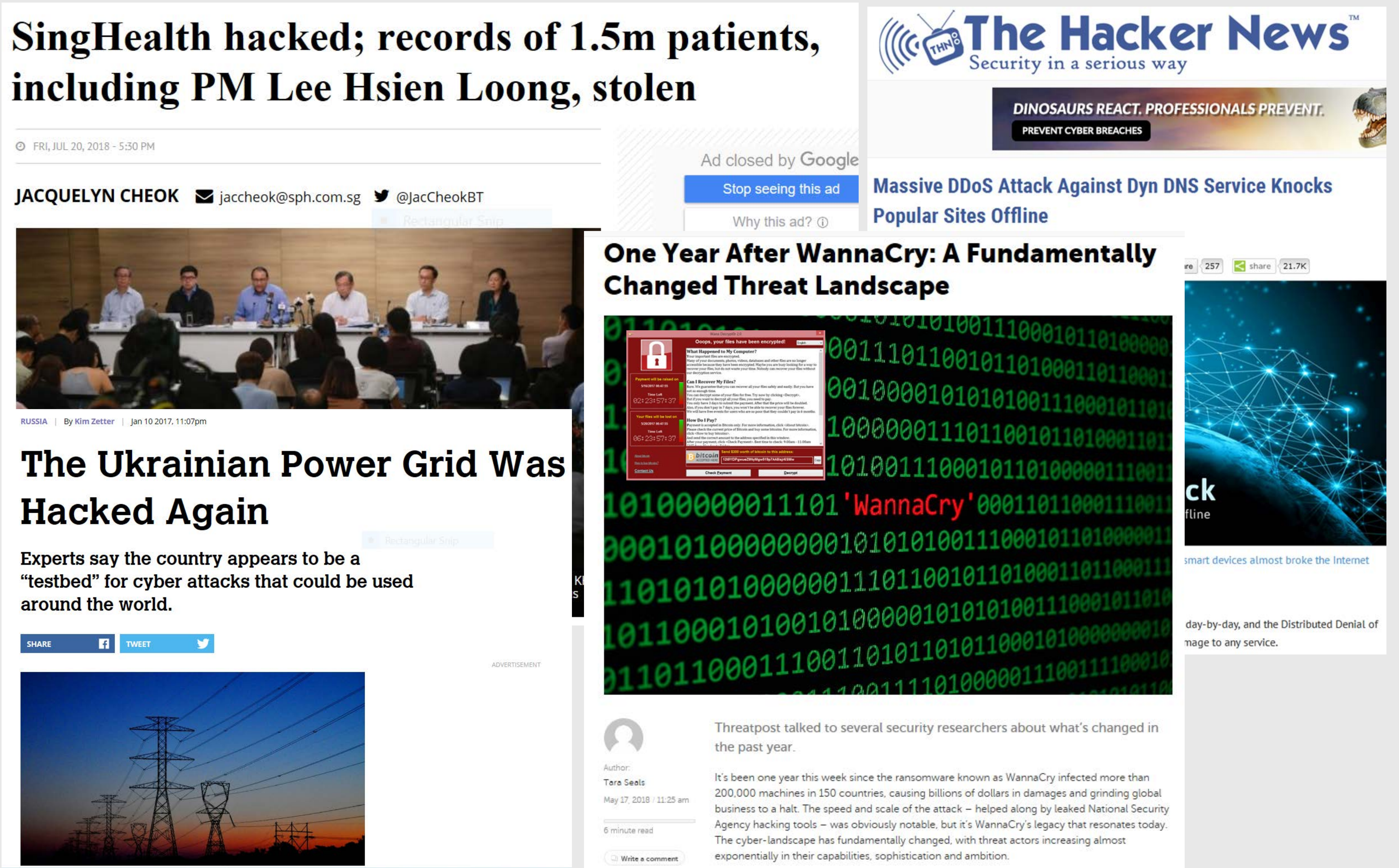
jianying_zhou@sutd.edu.sg

Office: 1.302.03

Edwin Franco

edwin_franco@sutd.edu.sg

Cyber Attacks in Real Life



SingHealth hacked; records of 1.5m patients, including PM Lee Hsien Loong, stolen

FRI, JUL 20, 2018 - 5:30 PM

JACQUELYN CHEOK jaccheok@sph.com.sg @JacCheokBT

RUSSIA | By Kim Zetter | Jan 10 2017, 11:07pm

The Ukrainian Power Grid Was Hacked Again

Experts say the country appears to be a "testbed" for cyber attacks that could be used around the world.

SHARE TWEET

The Hacker News™
Security in a serious way

Massive DDoS Attack Against Dyn DNS Service Knocks Popular Sites Offline

257 share 21.7K

smart devices almost broke the Internet

day-by-day, and the Distributed Denial of
mage to any service.

One Year After WannaCry: A Fundamentally Changed Threat Landscape

Threatpost talked to several security researchers about what's changed in the past year.

Author:
Tara Seals
May 17, 2018 / 11:25 am
6 minute read
Write a comment

It's been one year this week since the ransomware known as WannaCry infected more than 200,000 machines in 150 countries, causing billions of dollars in damages and grinding global business to a halt. The speed and scale of the attack – helped along by leaked National Security Agency hacking tools – was obviously notable, but it's WannaCry's legacy that resonates today. The cyber-landscape has fundamentally changed, with threat actors increasing almost exponentially in their capabilities, sophistication and ambition.

Learning Objectives

1. Define and explain the concepts of confidentiality, integrity and availability.
2. Model, analyze, and apply cryptographic primitives in standard situations.
3. Classify and describe common attacker models.
4. Select and discuss suitable countermeasures given an expected attacker model.
5. Evaluate the security of existing system designs respect to different attacker models.
6. Contrast efficiency vs. security trade-offs.
7. Examine and demonstrate an advanced cybersecurity topic based on a recent scientific publication or technical report.

Course Overview

Part I: Foundations (4 classes)

- Week 1: Security concepts and access control
- Week 2: Confidentiality and integrity policies
- Week 3: Information flow policies and enforcement
- Week 4: Availability and concurrency in distributed systems

Course Overview

Part II: Cryptography (4 classes)

- Week 5: Symmetric crypto
- Week 6: Mid-term exam
- Week 7: Recess
- Week 8: Hashes and authentication
- Week 9: Secure channel and randomness
- Week 10: Public-key crypto

Course Overview

Part III: Advanced topics (3 classes)

- Week 11: Security protocols
- Week 12: Public-key infrastructures
- Week 13: PhD student presentations
  Q&A
- Week 14: Final exam

Course Overview

Recommended textbooks

- [MB]: *Computer Security: Art and Science*. Matt Bishop. ISBN: 978-0-201-44099-7, 2002. (2nd edition to be released in 2018)
- [RA]: *Security Engineering: A Guide to Building Dependable Distributed Systems*. Ross Anderson, 2nd edition, ISBN: 978-0470068526, 2008.
- [FSK]: *Cryptography Engineering: Design Principles and Practical Applications*. Niels Ferguson, Bruce Schneier, and Tadayoshi Kohno, ISBN: 978-0-470-47424-2, 2010.

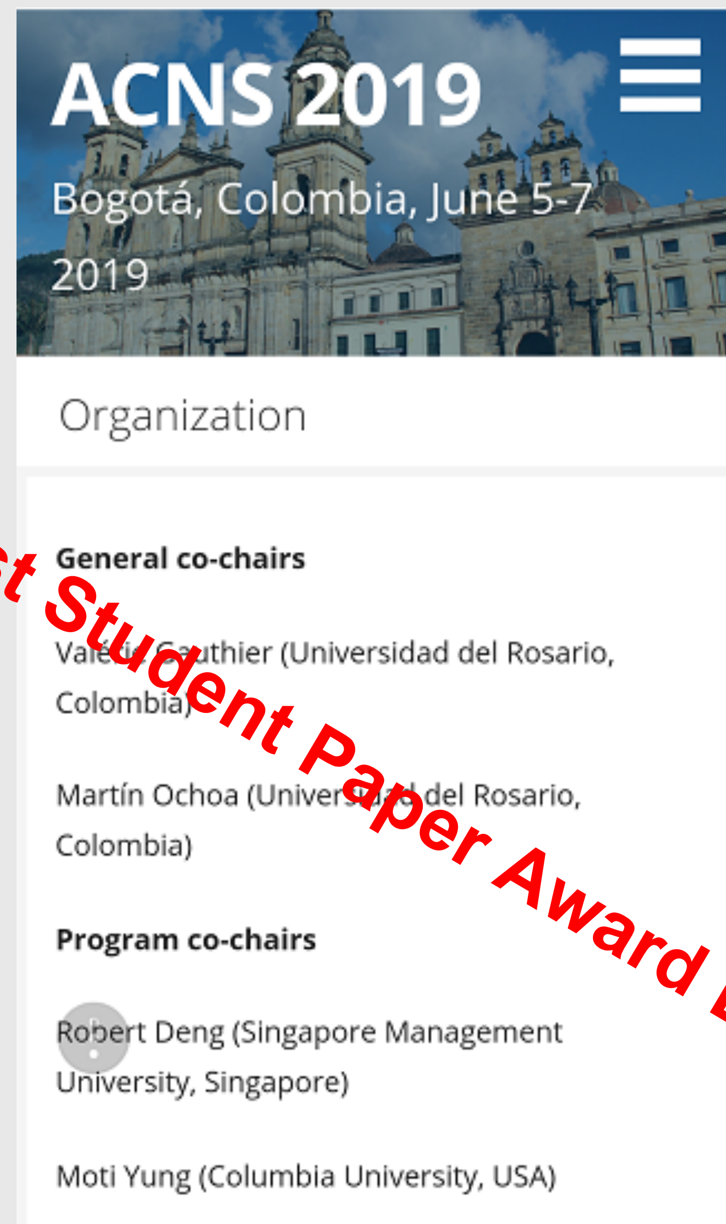
Top Cybersecurity Conferences

- http://jianying.space/conference-ranking.html

Conference	CIF (2017)	AR (2008-2017)	PR (2008-2017)	CR (2017)
1. IEEE S&P	3.79	**12.6%** = 41.1 / 326	**9.2%** = 41.1 / 448.9	**4.6%** (109)
2. Usenix Sec	3.21	**16.2%** = 49.6 / 306	**9.6%** = 49.6 / 517.9	**5.4%** (93)
3. ACM CCS	2.58	**18%** = 93.9 / 520.9	16.1% = 93.9 / 584.7	**4.6%** (109)
4. Eurocrypt	2.55	21.8% = 43.4 / 199.5	**11.4%** = 43.4 / 379.5	6% (84)
5. NDSS	2.41	**16.8%** = 42 / 250.1	19.8% = 42 / 212	**4.9%** (102)
6. Crypto	2.39	22.7% = 53.6 / 236	**13.5%** = 53.6 / 396.2	**5.6%** (90)
7. CHES	2.31	24.2% = 30.7 / 126.7	**8.7%** = 30.7 / 354.4	10.4% (48)
8. ACSAC	1.99	20.1% = 43.2 / 214.8	18.6% = 43.2 / 232.7	11.6% (43)
9. Asiacrypt	1.98	**20%** = 49.8 / 248.9	21.1% = 49.8 / 236.3	9.4% (53)
10. RAID	1.73	25.7% = 21.3 / 82.9	17.9% = 21.3 / 119	14.3% (35)
11. IEEE/IFIP DSN	1.66	23.1% = 54.7 / 236.7	25% = 54.7 / 218.6	12.2% (41)
12. FC	1.64	27.1% = 29.7 / 109.6	26.1% = 29.7 / 113.6	7.6% (66)
13. FSE	1.63	31.9% = 28.6 / 89.6	18.7% = 28.6 / 153.2	10.9% (46)
14. PKC	1.57	25.6% = 32.8 / 128.1	28.5% = 32.8 / 115.2	9.6% (52)
15. ESORICS	1.52	20.2% = 48.1 / 238.4	34% = 48.1 / 141.5	11.6% (43)
16. ACM WiSec	1.44	27.4% = 23.5 / 85.7	29.6% = 23.5 / 79.3	12.5% (40)
17. ACNS	1.44	20.4% = 32.8 / 160.9	36% = 32.8 / 91	13.2% (38)
18. IEEE CSF	1.35	28.4% = 25.8 / 91	27.9% = 25.8 / 92.6	17.9% (28)
19. ACM AsiaCCS	1.34	23.8% = 55.3 / 232	41.3% = 55.3 / 133.9	9.8% (51)
20. TCC	1.33	34.5% = 39.6 / 114.8	32.5% = 39.6 / 121.9	8.3% (60)

ACNS 2019

- http://www.acns19.com/ ; **ACNS Home (**http://jianying.space/acns/**)**



Best Student Paper Award Euro 1000 !

Bureaucracy

We have three hours of which:

- About half the time recap of theory and discussions.
- About half the time class exercises and discussions.
- About 10-minute break in between.

Bureaucracy

- *Classwork*: Groups of max 3 members (15% MSSD, 10% PhD)
- *Homework*: Groups of max 3 members (25% MSSD, 20% PhD)
- *Mid-term exam:* Individual (20%) – Part I
- *Final exam*: Individual (40%)

Bureaucracy

Additional work for PhD students (10%):

- Each student selects one paper from this year's top cyber security conferences by **Week 8**:
  - ✓ IEEE S&P, ACM CCS, Usenix Security, NDSS, ACM AsiaCCS, ESORICS, ACNS.
- Prepare presentation explaining the paper to the class in Week 13.
  - ✓ Analysis: pros & cons
  - ✓ **Bonus:** if discovering flaws, or proposing improvements

Bureaucracy

Course platform:

- eDimensioin – https://edimension.sutd.edu.sg/
- Lecturing slides will be available at eDimension.
- Submit your classwork and homework to eDimension.
- Announcements and interactions via eDimension.

Bureaucracy

Groups for classwork & homework:

- Each group has max 3 members.
- Two options to decide group members:
  - ✓ You choose your own group members.
  - ✓ Randomly assign group members.

You vote !

Once upon a time…

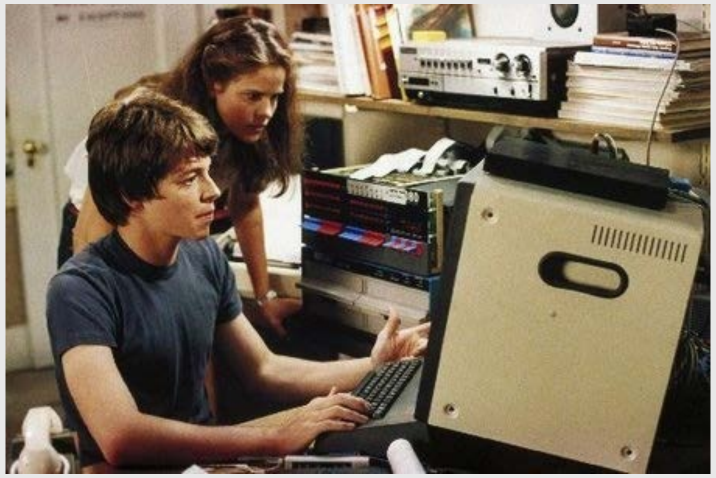
- Secrets always exist !
  - ✓Roots of cryptography go back a long way
- Physical security
  - ✓Castles, walls, locks etc.
- What has changed ?

Turing Machines

- Alan Turing invented computers theoretically in the 1930's.
  - ✓ "*On Computable Numbers, with an Application to Entscheidungsproblem*" (1936)
- They were implemented practically during WWII.
- What is it ?

Information Security

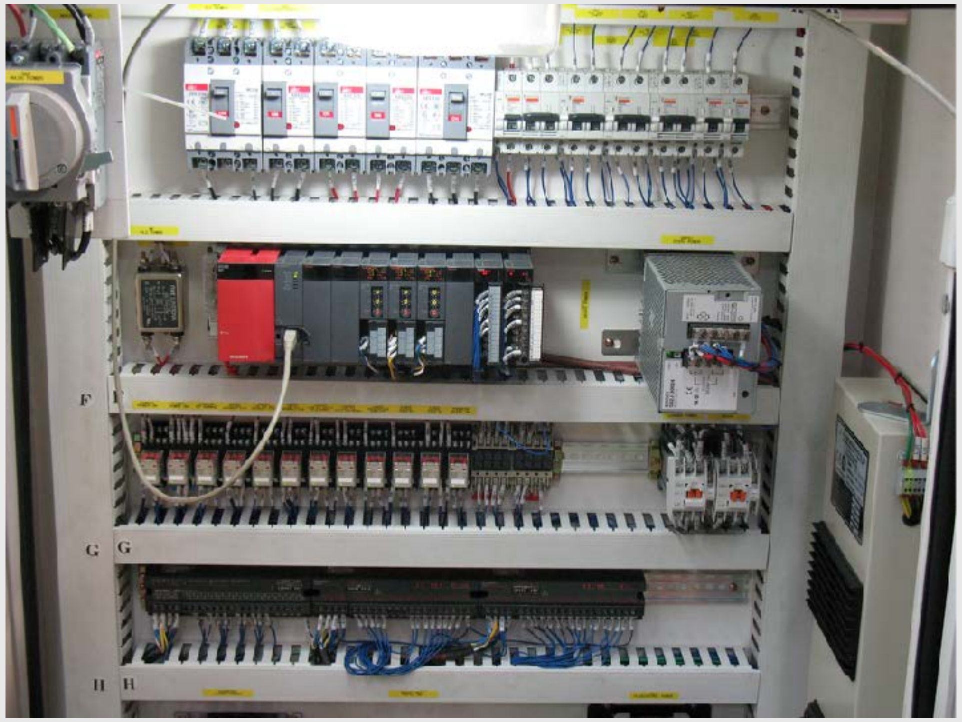
- Applications of computing to both civilian and military.
- Increase in connectivity, specially after mid 1980’s.
- How to protect information?
- What has changed in terms of threats?



Cyber-Physical Systems

- Turing machines control and sense directly the physical world.

- What has changed in terms of impact of attacks?



Security is broad !

- To talk about cybersecurity of a system involves
  - ✓ Software Applications
  - ✓ Operating Systems
  - ✓ Hardware
  - ✓ Communication Networks
  - ✓ Domain specificity of system (Banking, Water, Power etc.)
- Different perspectives (design, implementation, testing)

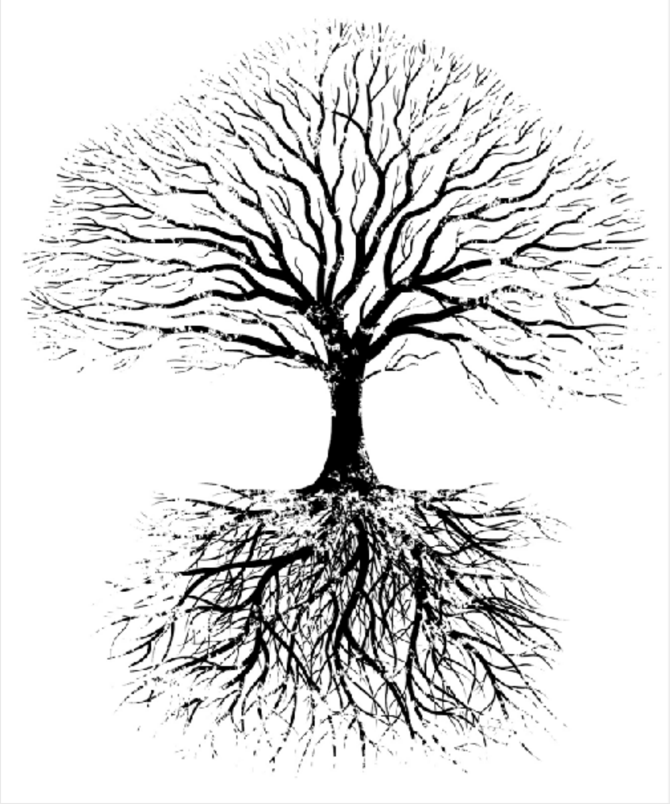
What is this course about ?

- Will review fundamental concepts of cybersecurity.

  ✓ Abstract concepts, that apply to many settings.

  ✓ Concrete building blocks that can solve problems in many settings too.

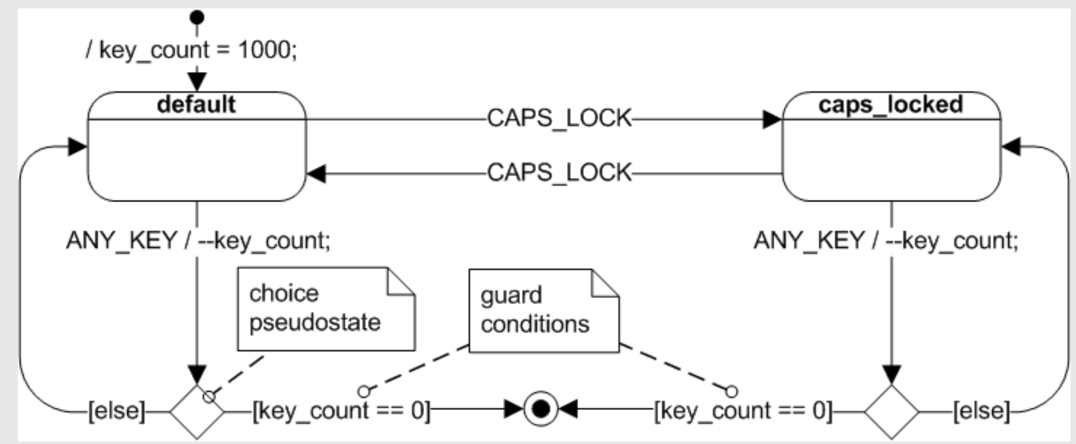
  ✓ Applications of building blocks to well known general issues.

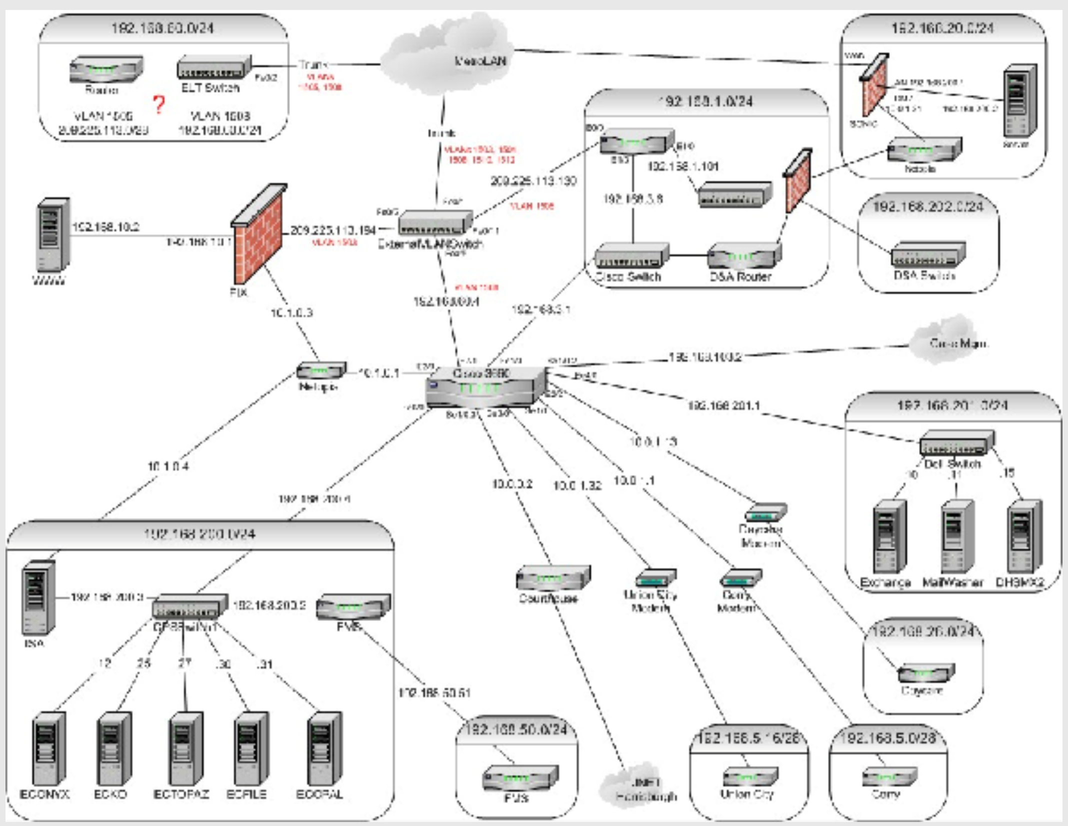


- Fundamental $\neq$ Easy.

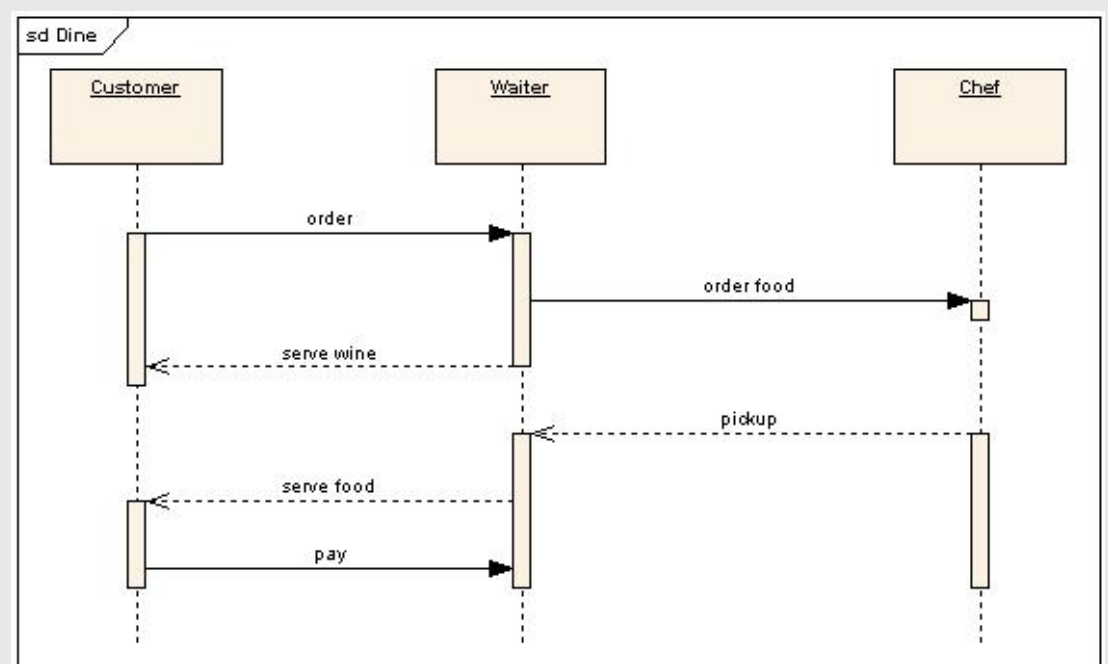
System Models

- To talk about the security of a system, we need to have a model of what the system is, and what it does.





Static



Behavioural/Semantic

Attacker Models

- Security is a negative property: *No attacker can do X*.
- Is good to characterise what “attacker” means, what can they do?
- Examples:
  - ✓ *Dolev-Yao attacker model*: Can intercept, modify and block all messages in a communication channel.
  - ✓ *Man-at-the-end attacker model*: Has access to an end-host, can debug, decompile, patch application binaries.

Basic Security Concepts

CIA $\neq$

Confidentiality

- What is confidentiality?

  ✓ *"the **property**, that information is not made available or disclosed to unauthorized individuals, entities, or processes"* (ISO27000).

- Ability to distinguish groups of users is crucial!

- Example of attacks?

- How to enforce confidentiality?

Integrity

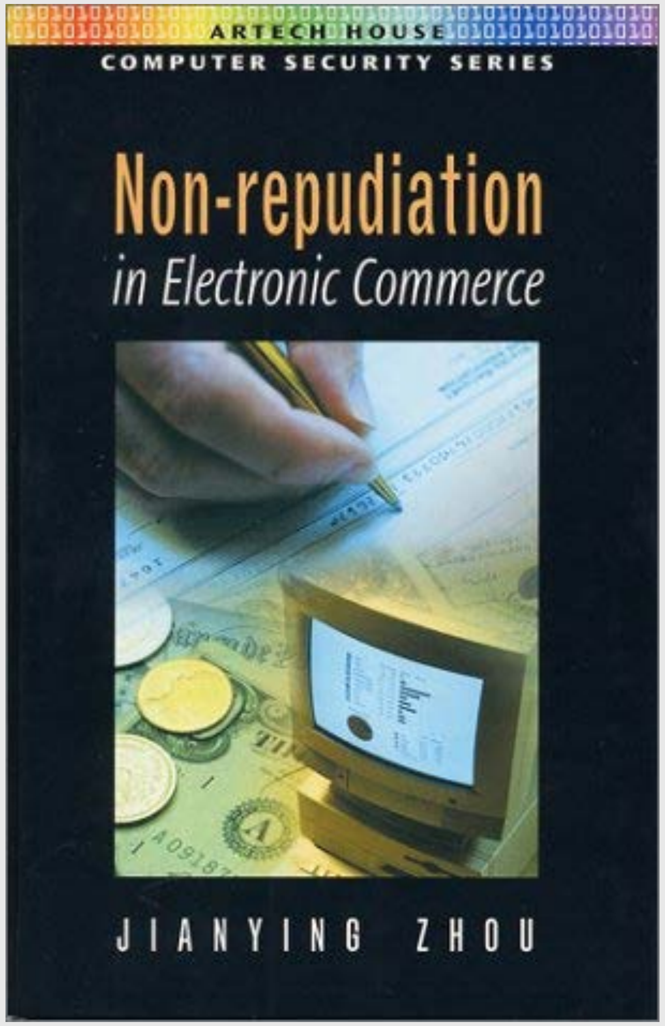
- What is integrity?
  - ✓ *“Integrity involves maintaining the consistency, accuracy, and trustworthiness of data over its entire life cycle.” –* ***data integrity***
  - ✓ *If it is about* ***origin integrity*** *(the source of the data, often called* ***authentication****).*
- In cyber-physical systems this can be extended, how?
- Example of attacks?
- How to enforce integrity?

Availability

- What is <u>availability</u>?
  - ✓ *"Availability refers to ensuring that authorized parties are able to access systems/information when needed."*
- Example of attacks?
- How to enforce availability?

Non-repudiation *

- What is non-repudiation?
  - ✓ *"the **ability** to prove that an event occurred or an action was carried out by an entity."*
- Example of attacks?
- How to enforce non-repudiation?



Basic Terminology

- A threat is "*A potential cause of an incident, that may result in harm of systems and organizations*" (ISO 27005) (i.e., a hacker, an unintended leakage of information)
- A vulnerability is "*A weakness of an asset or group of assets that can be exploited by one or more threats*" (ISO 27005) (i.e., an SQL-i vulnerability)
- An exploit is …hey, there are no good definitions around! But usually is some code or series of steps to take advantage of a vulnerability an carry out an attack.
- An attack is "*an assault on system security that derives from an intelligent threat, i.e., an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system*" (IETF) (i.e., a DoS attack on a server exploiting a buffer overflow).

Security Policy & Mechanism

- *A* <u>security policy</u> *is a statement of what is, and is not, allowed.*
  - ✓ Security policies can be thought of as requirements to a system, or refinements of more abstract properties.
- *A* <u>security mechanism</u> *is a method, tool or procedure for enforcing a security policy.*
  - ✓ Mechanisms are ways to enforce policies. Broadly 3 classes:
    - ❖ Prevention
    - ❖ Detection
    - ❖ Recovery

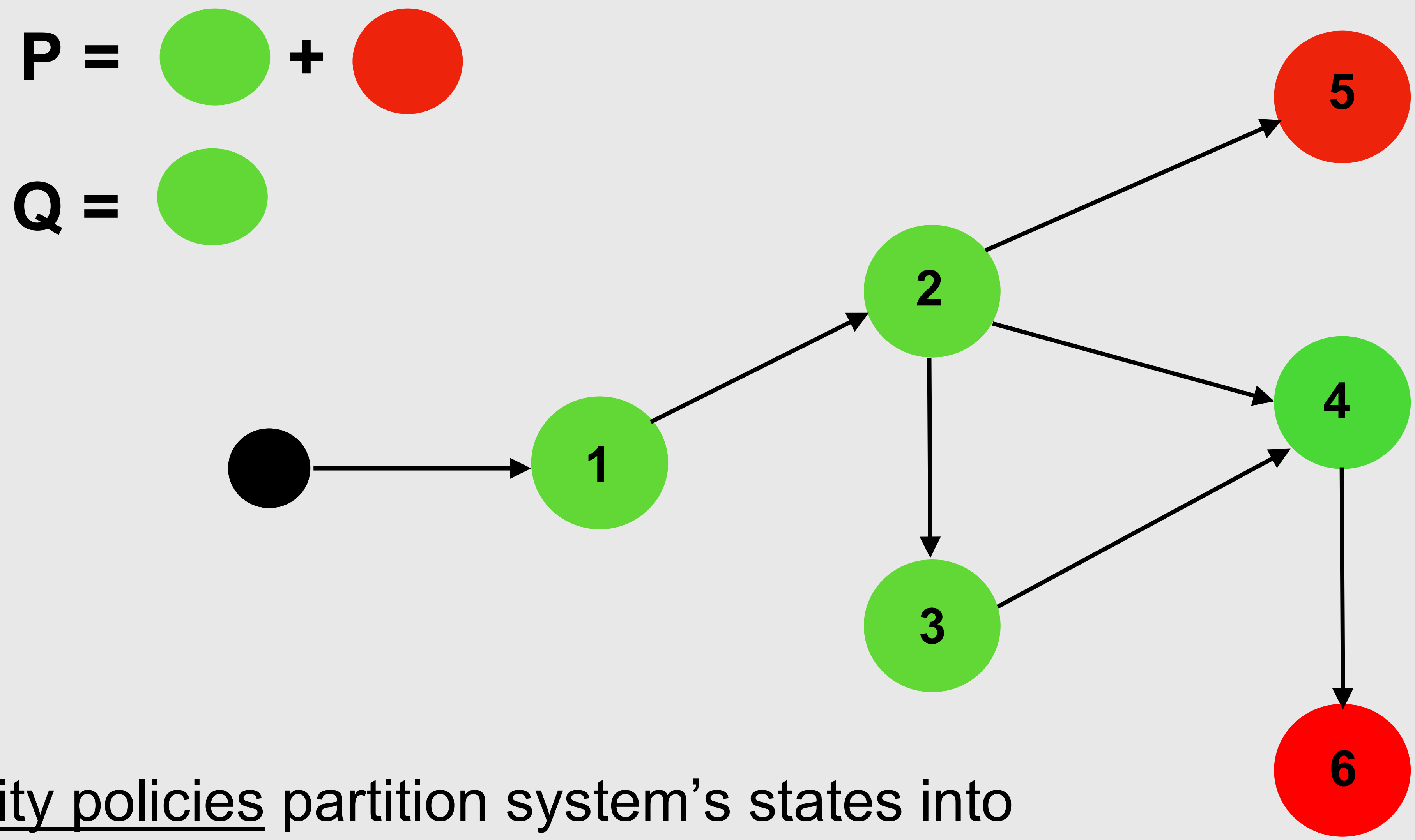
Security Policy & Mechanism

- *Example informal requirement:* Homework of individual students should be confidential.
- *Example policy:* In a system, students cannot copy files from other student’s homework folder.
- *Example mechanism:* The system has an access control monitor that blocks copy attempts between students. Students need to properly configure the settings of the system for the mechanism to work.

Assumptions & Trust

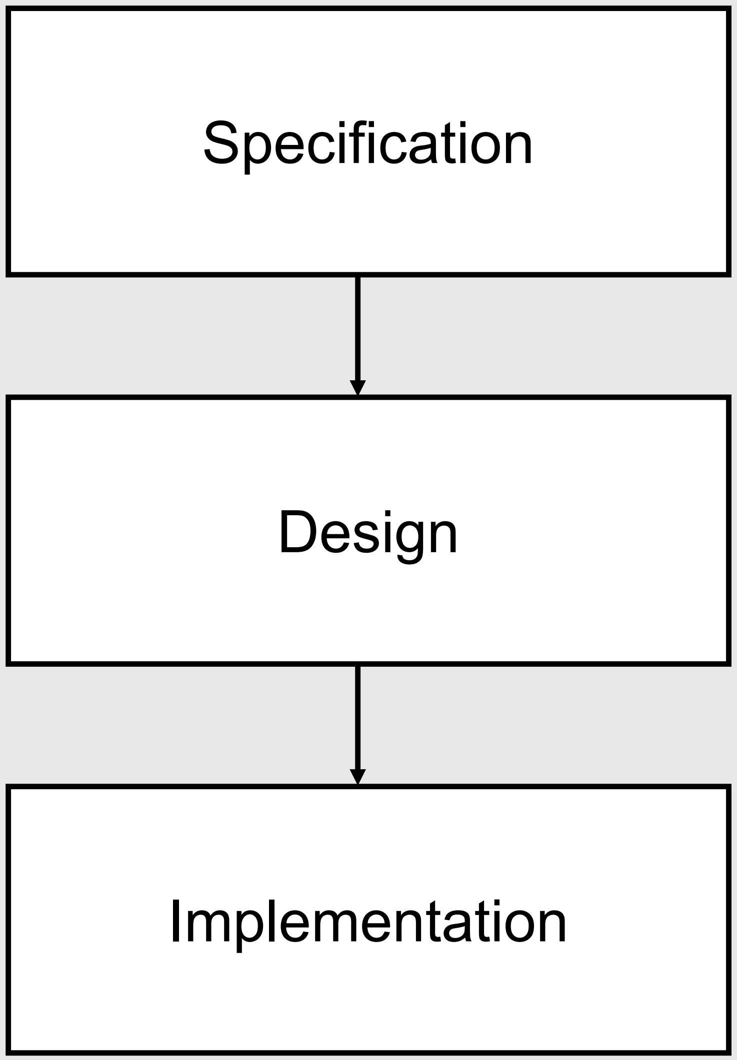
- How much do we trust a policy?
- How much do we trust a mechanism?
- Let P all states, Q secure states by policy, R restricted states by mechanism. A mechanism is
  - ✓ _Secure_ (or sound) if $R \subseteq Q$
  - ✓ _Precise_ if $R = Q$
  - ✓ _Broad_ (not sound) if $R \cap (P \setminus Q) \neq \emptyset$

Policies & System's States



- Security policies partition system's states into 'good' and 'bad'.
- Security mechanisms enforce systems within 'good' states.

Assurance



Assuming specification is good.

- How good is the design?
- How good is the implementation?
- How to know?
  - ✓ Mathematical proofs (verification)
  - ✓ Testing

Operational Issues

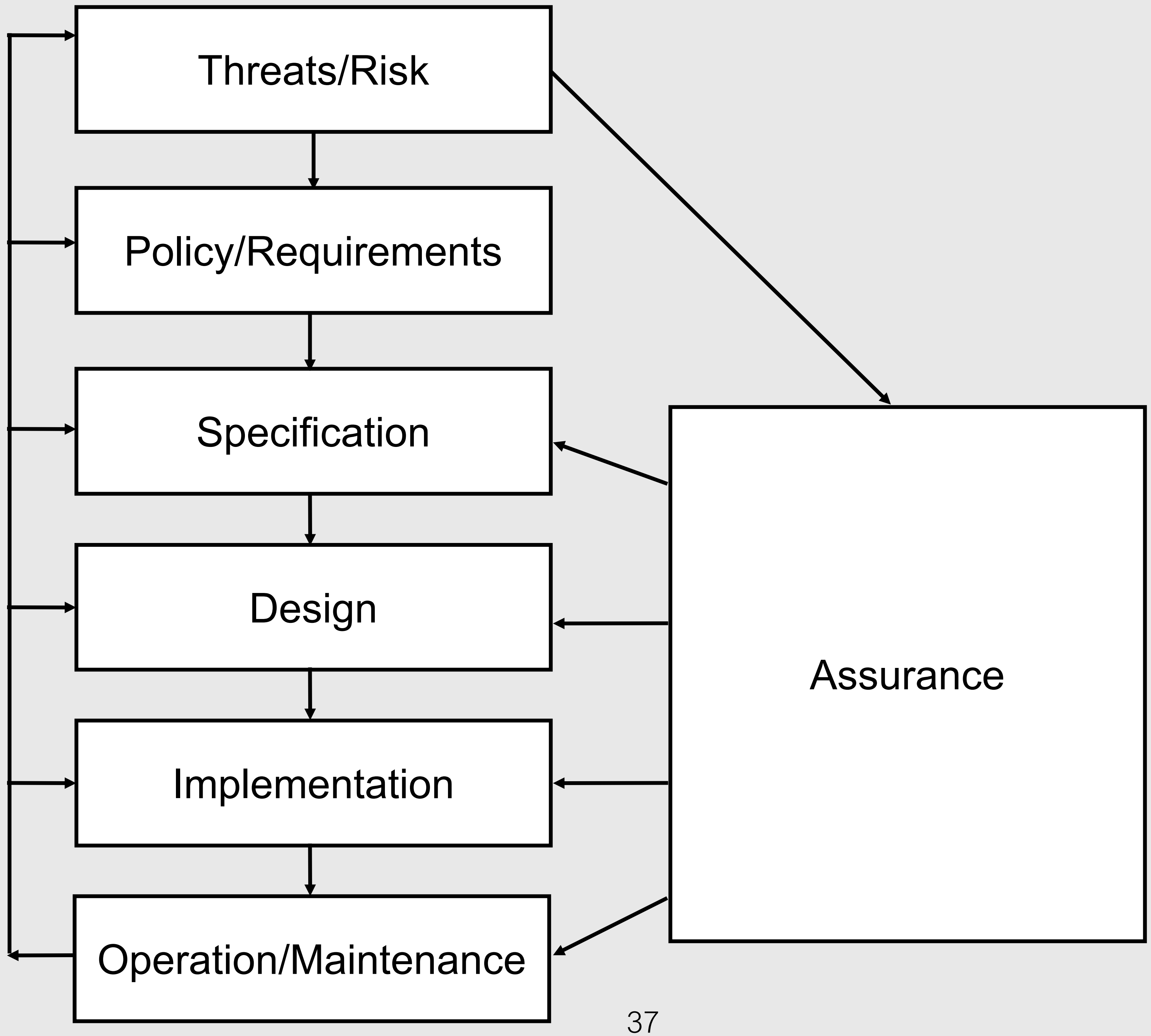
- Cost-benefit analysis.
  - ✓ High-assurance is expensive. Is it worth it?
- Risk analysis.
  - ✓ Who is attacking, why, what are the odds?
  - ✓ Impact?
- Laws and regulations.

Human Issues

- (Often) weakest link in the security of systems.
  - ✓ Social engineering
  - ✓ Malicious insiders

https://www.youtube.com/watch?v=opRMrEfAIiI

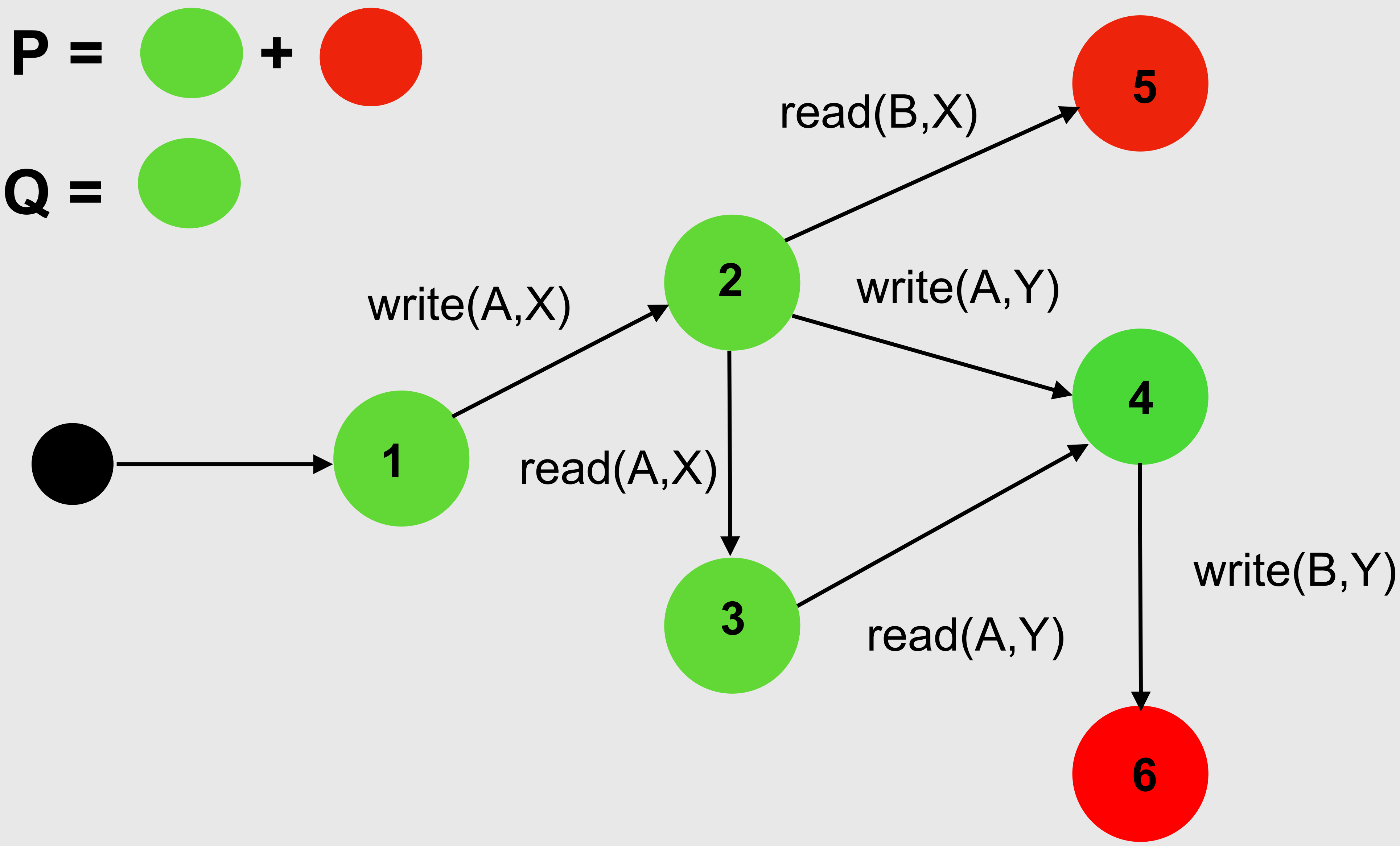
Security Life-cycle



Subjects & Objects

- Usually in systems there is a set of well defined <u>*subjects*</u> *(**S**),* i.e.:
  - ✓ Authenticated users
  - ✓ Processes
- Also, there is a set of well defined <u>*objects*</u> *(**O**),* i.e.:
  - ✓ Files
  - ✓ Database tables
- <u>Access right</u>: *a(s,o) where* $s \in S$, $o \in O$

Protection State



"Only A should have access to X and Y"

Access Control Matrix

- Defines protection state. For instance in *operating system*:

	file 1	file 2	process 1	process 2
process 1	read, write, own	read	read, write, execute, own	write
process 2	append	read, own	read	read, write, execute, own

Access Control Matrix

- Or in a *database*:

	table 1	table 2
user 1	insert, delete	select
user 2	update	select, insert

Access Controlled by History

- Consider a *statistical* database where only queries on sums or counts can be made.

name	position	age	salary
Celia	teacher	45	$40,000
Heidi	aide	20	$20,000
Holly	principal	37	$60,000
Leonard	teacher	50	$50,000
Matt	teacher	33	$50,000

Access Controlled by History

- Why is the following bad?
  - ✓ *Query 1*: sum_salary('position is teacher')
  - ✓ *Query 2*: count('age less than 40 and teacher')
  - ✓ *Query 3*: sum_salary('age greater than 40 and teacher')

Access Controlled by History

- How to protect against this? One idea: **query-set-overlap control**. For r = 2:

  ✓ *Query 1*: sum_salary('position is teacher')
  ➔ Involved records: {Celia, Leonard, Matt}

  ❖ Intersection with all previous queries = **0**

  ✓ *Query 2*: count('age less than 40 and teacher')
  ➔ Involved records: {Matt}

  ❖ Intersection with all previous queries = **1**

- *Query 3*: sum_salary('age greater than 40 and teacher')
  ➔ Involved records: {Celia, Leonard}

  ❖ Intersection with all previous queries = **2**

Protection State Transitions

- Primitive commands to evolve the protection state:
  - ✓ `create subject` **`s`**
  - ✓ `create object` **`o`**
  - ✓ `enter right` **`r`** `into matrix`
  - ✓ `delete right` **`r`** `from matrix`
  - ✓ `destroy subject` **`s`**
  - ✓ `destroy object` **`o`**

Protection State Transitions

- For instance in UNIX-like systems, a process (***p***) creates a file (***f***), with owner read (*r*) and write (*w*) permission:

```
command create_file(p,f)
    create object f;
    enter own into a[p,f];
    enter r into a[p,f];
    enter w into a[p,f];
end
```

Protection State Transitions

- <u>Conditional command</u>: suppose a process (***p***) wishes to give another process (***q***) the right to read a file (***f***) <u>if ***p*** owns ***f***</u>.

```
command grant_read_file(p,f,q)
    if own in a[p,f]
    then
      enter r into a[q,f];
end
```

Delegation

- How can the protection system evolve?
  - ✓ Subjects must be allowed to execute primitive or composed commands.
- Can we delegate permissions to other users?
  - ✓ In principle yes, if the following holds:
  - ✓ *Principle of Attenuation of Privilege*: A subject may not give away rights it does not possess.

Delegation

- Usually *owners* can add/remove permissions to the objects they own, and also grant permissions to others.

- For instance `chmod` command in UNIX-like OSs.

```
chmod u=rwx,g=rx,o=r myfile
```

```
chmod 754 myfile
```

(4 = read, 2 = write, 1 = execute, 0 = no permission)

Delegation

- In some systems, subjects may have an extra "copy right" (or "grant right") to delegate rights to others:

  ✓ For instance if user *A* has rights *read (***r***), write (***w***), copy* (**c**) for file *f*, he can assign *read, write* rights for file *f* to user *B*.

```
command grant_read_write_rights(A,f,B)
      if r in a[A,f] and c in a[A,f]
      then
        enter r into a[B,f];
      if w in a[A,f] and c in a[A,f]
      then
        enter w into a[B,f];
  end
```

Delegation

- The “own right” is a special right that enables an owner to
  - ✓ add or delete privileges for himself
  - ✓ grant rights to others

- Does the following example violate *Principle of Attenuation of Privilege*?
  - ✓ User *A* has rights *read (**r**), own* (***o***) for file *f*, he assigns *write* right for file *f* to user *B*.

Key Points

- Fundamental concepts: CIA
- System models & attacker models
- Security policy & mechanism
- Access control matrix
  - ✓ Protection state transitions
  - ✓ Delegation

Exercises & Reading

- Classwork (Exercise Sheet 1): due on Fri Sept 14, 10:00 PM
- Homework (Exercise Sheet 1): due on Fri Sept 21, 6:59 PM
- Reading: MB [Ch1 & Ch2], RA [Ch1]

End of Slides for Week 1