Security and Trust I:
1. Introduction

Dusko Pavlovic

UHM ICS 355
Fall 2014

Outline

Announcements

What is security?

Structure of the course

Outline

Announcements

What is security?

Structure of the course

Contacts

- Dusko Pavlovic
- email: dusko@hawaii.edu
- office: 311B
  - hours: TW 4:30pm, F 9am

Contacts

- Depeng Li
- email: depengli@hawaii.edu
- office: 314D

Contacts

- Nancy Mogire
- email: nmogire@hawaii.edu
- office: 311A
  - hours: TW 4:30pm, F 9am

Credits

3

- class participation and presentations: 25%
- 3 homework assignments: 25%
- midterm exam: 25%
- final exam: 25%

Course web page

asecolab.org/courses/ICS355/

Readings

- Dorothy Denning, *Cryptography and Data Security* **Chapters 4–5.** Addison-Wesley 1983
- Dieter Gollmann, *Computer Security* **not Part Three.** Wiley 2011
- Matt Bishop, *Computer Security: Art and Science* **Parts 1–3.** Addison-Wesley 2005

What shall we study?

- What do you expect from the course?

What shall we study?

- What do you expect from the course?
- Why security?

We study Computer Science

. . . in modern CS security is the main problem

age	*ancient times*	*middle ages*	*modern times*
platform	computer	operating system	network
applications	Quicksort, compilers	MS Word, Oracle	WWW, botnets
requirements	correctness, termination	liveness, safety	trust, privacy
tools	programming languages	specification languages	scripting languages

Paradigm shifts in computation

What shall we study?

- What is security?

Outline

Announcements

What is security?

- Security requirements
- Security types and properties
- Security, networks and protocols
- Honesty and trust
- Security and Privacy
- Phases and implementations of security
- Security is a process

Structure of the course

Home sweet home

The Flintstone family owned a cave house.

Home sweet home

Their house was **lively** and **functional.**

Home sweet home

For **safety** from the storms

Home sweet home

For **safety** from the storms
the house had a door.

Home sweet home

For **security** from the thieves

Home sweet home

For **security** from the thieves
the door had a lock, and the house had a fence

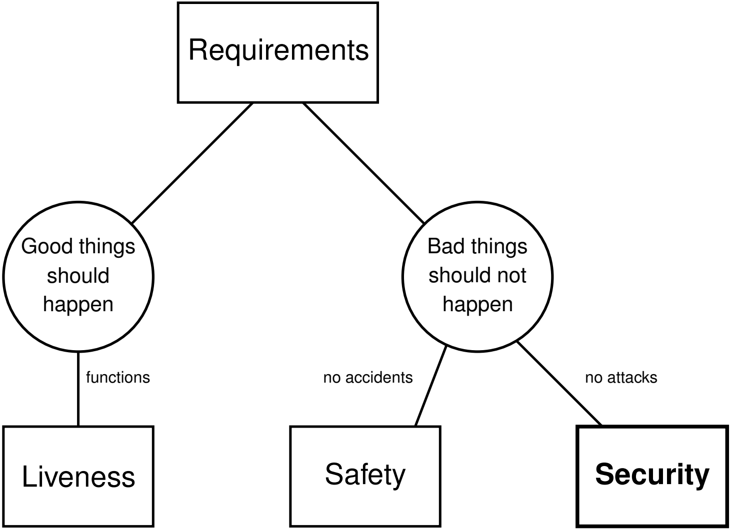
Home sweet home

For **security** from the thieves
the door had a lock, and the house had a fence
and the security experts patrolled in the neighborhood.

What do you require for a good life?

What does a software system require?

What does a software system require?



What does a software system require?

Liveness vs Safety vs Security

Liveness: A dwelling to perform the *functions* of life.

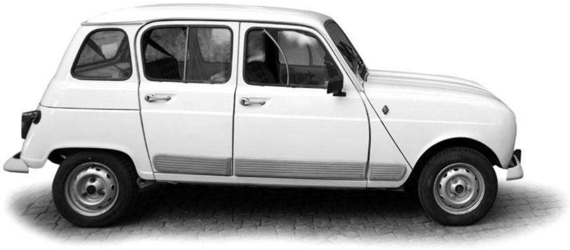
Liveness vs Safety vs Security

Safety: A door for protection from *natural hazards*.

Liveness vs Safety vs Security

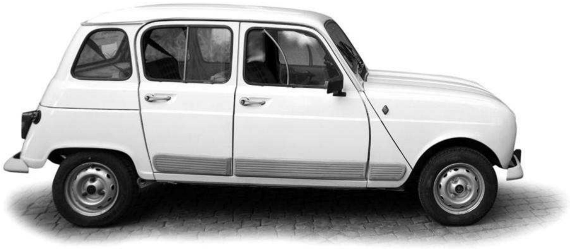
Security: A lock for protection from *intentional intruders*.

Liveness vs Safety vs Security



- **car liveness (functionality)**: driving
- **car safety**: no accidents
- **car security**: no theft

Liveness vs Safety vs Security



- **car liveness (functionality)**: driving ⇜ engine
- **car safety**: no accidents ⇜ brakes
- **car security**: no theft ⇜ locks

Logical form of security requirements

On a mountain

- *positive requirements*: reach the peak

  liveness: climb up the mountain

- *negative requirements*: do not fall

  safety: do not slip on ice

  security: do not let someone push you

Logical form of security requirements

In a crypto system

- *positive requirements*: encryption and decryption

  liveness: $D(k, E(k, m)) = m$

- *negative requirements*: ***only*** decryption with key

  safety: no bugs in the implementation

  security: if $A(E(k, m)) = m$ then $A(y) = D(k, y)$

Logical form of security requirements

On the airport

- *positive requirements*: route the traffic

  liveness: board passengers to and from planes

- *negative requirements*: ***only*** route the traffic

  safety: do not leave the floor slippery

  security: prevent theft and terrorism

Logical form of security requirements

In a kitchen

- *positive requirements*: food

  liveness: prepare and eat food

- *negative requirements*: ***only*** good food

  safety: do not bite your tongue or swallow a fork

  security: resist malicious advertising and food baiting

Logical form of security requirements

So there is always the same pattern

- *positive requirements*: . . . (something you need)

  liveness: . . . (what you do to get it)

- *negative requirements*: . . . (avoid trouble)

  safety: . . . (natural hazards)

  security: . . . *(intentional attacks)*

Logical form of security requirements

This pattern is everywhere

- Almost anything can become a security problem
- Is there any system to it?
- What types of security problems are there?
- What types of security solutions?

What do we secure and how?

Security tasks and tools fall into the same types

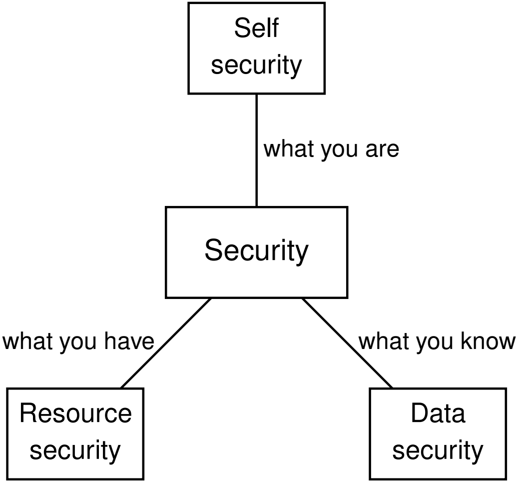
- **data and information**: what you know
- **objects and resources**: what you have
- **subjects and self/(id)entity**: what you are

What do we secure and how?

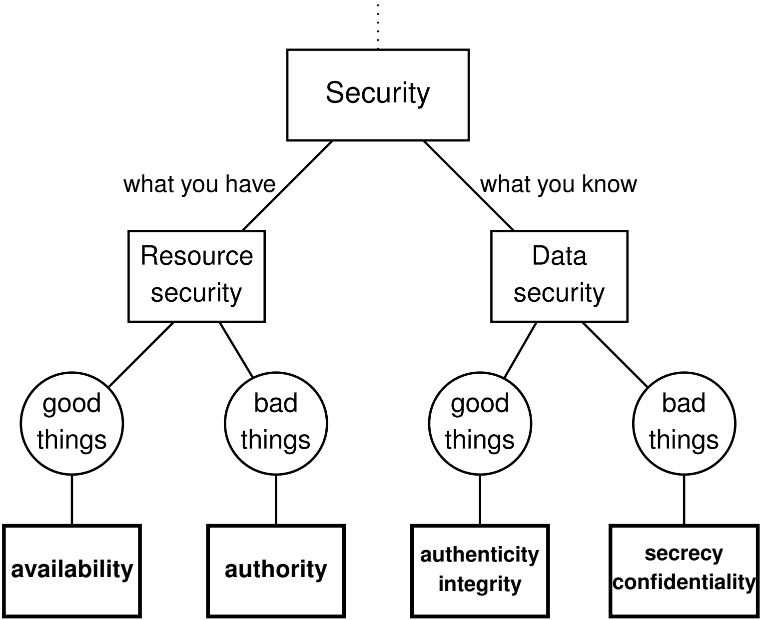
Security tasks and tools fall into the same types

- **data and information**: what you know
  - can copy
  - can give away
  - (and then still know: *password, digital key*. . .)
- **objects and resources**: what you have
  - cannot copy
  - can give away
  - (but not have any more: *smartcard, physical key*. . .)
- **subjects and self/(id)entity**: what you are
  - cannot copy
  - cannot give away
  - (you always are yourself: *fingerprint, handwriting*. . .)

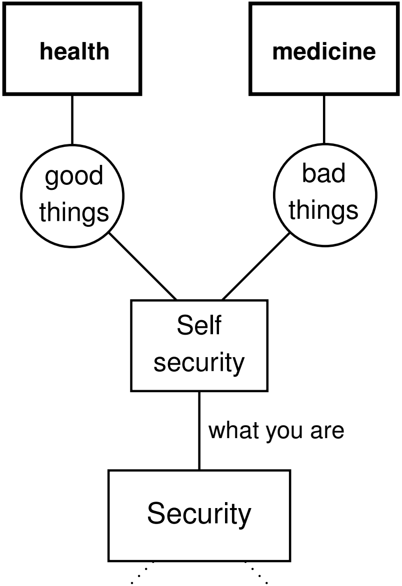
Three types of security tasks



Our *data* and *resources* are secured together



Our *selves* are secured separately



Remaining questions

- What is privacy?
  - How is it related with security?

Remaining questions

- What is privacy?
  - How is it related with security?
- What is trust?
  - How is it related with security?

Remaining answers

- To answer these questions, we need to take a closer look at the **security processes**

Remaining answers

- To answer these questions, we need to take a closer look at the **security processes**
- What kind of a process is security?

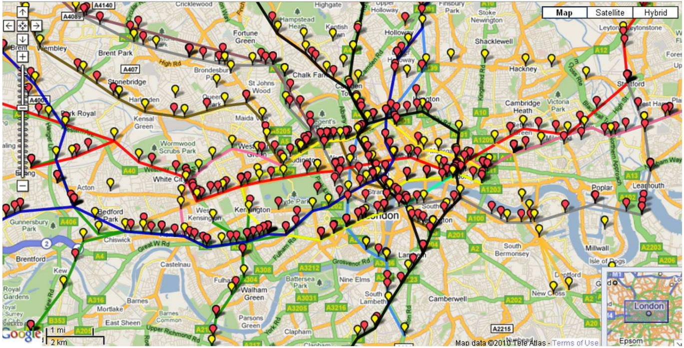
Remaining answers

- To answer these questions, we need to take a closer look at the **security processes**
- What kind of a process is security?
- What is its space and time?

Map of London

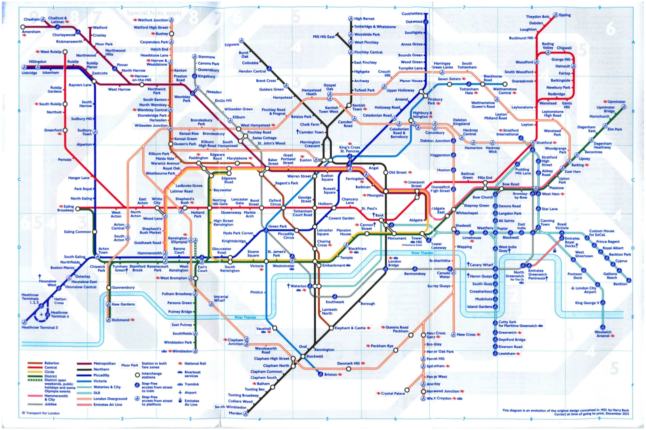
A view of space inhabited by people

Map of London Tube stations



Display some type of interactions,
abstract away the irrelevant details

Network of London Tube



Abstract space of interactions

What is a network?

Network is an abstraction of space

consisting of

- **nodes**: all local actions are at the nodes
  - (You can only enter or exit a train at stations ⇝ nodes.)
- **links**: all non-local interactions are along the links
  - (The trains only move along the rails ⇝ links.)

What is a protocol?

$$\frac{\text{protocol}}{\text{network}} = \frac{\text{program}}{\text{computer}}$$

Roles and actors

Protocol assigns roles to computational actors: Alice, Bob,. . .

Honesty

- An actor Bob is honest if he acts according to a given protocol

Trust

- Trust is Alice's belief that Bob is honest
  - i.e. that he will act according to a specified protocol

Trust

Examples

- shopping: Bob will deliver goods
- marketing: Bob will pay for goods
- access control: Bob will not abuse resources
- key infrastructure: Bob's keys are not compromised

Trust

Examples

- shopping: Bob will deliver goods
- marketing: Bob will pay for goods
- access control: Bob will not abuse resources
- key infrastructure: Bob's keys are not compromised
- Prisoners' Dilemma: Bob will not defect
- Centipede game: ...
- ...social cooperation is possible

Privacy

Privacy is the right to be left alone
(with all your possessions)

Warren and Brandeis
Harvard Law Review 1890

Security vs Privacy

- Security is the **requirement**
  to be protected from dishonest attackers and intruders
  - thieves, enemies, spies...
  - breaking protocols
  - — but rational, predictable
- Privacy is the **right**
  to be protected from honest participants
  - government, merchants, parents, friends...
  - expected to obey some explicit or implicit protocols
  - — but curious, sometimes unreliable

Security and privacy implementations

Three phases of security

- **prevention:** security properties cannot be breached
  - firewalls, cryptography
- **detection:** security breaches are detected
  - intrusion detection, digital forensics
- **deterrence:** recovery, penalties, incentives
  - legal measures (RIAA, MPAA), economics of security (cost of an attack must be higher than the expected profit of success)

Security and privacy implementations

Three phases of security

- **prevention:** security properties cannot be breached
  - firewalls, cryptography
- **detection:** security breaches are detected
  - intrusion detection, digital forensics
- **deterrence:** recovery, penalties, incentives
  - legal measures (RIAA, MPAA), economics of security (cost of an attack must be higher than the expected profit of success)

Security implementations are specified as **policies**

Warning about terminology

- Security is many things to many people
  - software engineer, government, school, beehive. . .
- Security terms and concepts vary from context to context
  - Different purposes justify different concepts
- We fix the glossary for the purposes of this course
  - The other usages are not less, or more correct
  - They may be less useful, or more useful

Warning about security

- Security is a process

Warning about security

- Security is a process
  - All systems become insecure eventually

Process of Science

If we have a definite theory, from which we can compute the consequences which can be compared with experiment, then in principle we can prove that theory wrong.

Process of Science

. . . But notice that we can never prove it right.

Suppose that you invent a theory, calculate the consequences, and discover every time that the consequences agree with the experiment. The theory is then right? No, it is simply not proved wrong. In the future you could compute a wider range of consequences, there could be a wider range of experiments, and you might then discover that the thing is wrong.

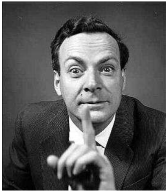
Process of Science

That is why laws like Newton's laws for motion of planets last such a long time. He guessed the law of gravitation, and it took several hundred years before the slight error in the motion of Mercury was observed. During all that time, the theory had not been proven wrong, and could be taken temporarily to be right.

Process of Science

We never are definitely right;
we can only be sure when we are wrong.

Richard Feynman
Lectures on the Character of Physical Law



The best kept secret of Science

- Science does not provide persistent laws
- Science only provides methods to improve theories

Religion

Religion says: This is the truth about the world.

- You can rely upon it.

Religion, Art

Religion says: This is the truth about the world.

- You can rely upon it.

Art says: This is a story about the world.

- You can relax and play with it.

Religion, Art, and Science

Religion says: This is the truth about the world.

- You can rely upon it.

Art says: This is a story about the world.

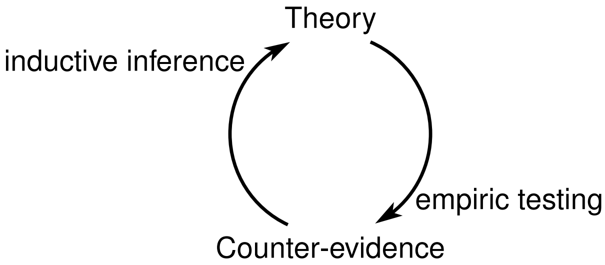
- You can relax and play with it.

Science says: This a theory about the world.

- You shouldn't rely upon it too much.
- You shouldn't relax, but work to improve it.

Upshot

Process of Science



Science never settles on a theory.

It loops through theories and counter-evidence forever.

Upshot

Security is like science:

it never settles

"Richard Feynman on Security"

If we have a precisely defined security claim about a system, from which we can derive the consequences which can be tested, then in principle we can prove that the system is insecure.

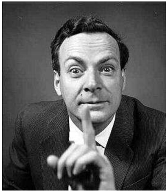
"Richard Feynman on Security"

. . . But we can never prove that it is secure.

Suppose that you design a system, calculate some security claims, and discover every time that the system remains secure under all tests. The system is then secure? No, it is simply not proved insecure. In the future you could refine the security model, there could be a wider range of tests and attacks, and you might then discover that the thing is insecure.

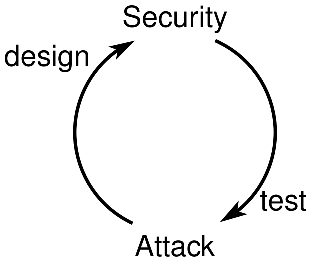
"Richard Feynman on Security"

We never are definitely secure;
we can only be sure when we are insecure.



Upshot

Process of Security



Security never settles.
Every security claim has a lifetime.

Outline

Software engineering

Program dependability

- **safety:** "bad things (actions) don't happen"
- **liveness:** "good things (actions) do happen"

Software engineering

Program dependability

- **safety:** "bad things (actions) don't happen"
- **liveness:** "good things (actions) do happen"

In sequential computation

- all first order constraints are dependability properties

Security engineering: Systems

Resource security (access control)

- **authorization:** "bad *resource calls* don't happen"
- **availability:** "good *resource calls* do happen"

In an operating or a computer system

- all resource constraints are security properties

Security engineering: Systems

Information security

- **secrecy:** "bad *information flows* don't happen"
- **authenticity:** "good *information flows* do happen"

In network computation

- all information flow constraints are security properties

Security engineering: Networks

Social choice (voting) and market economy

- **neutrality:** "bad *data aggregations* don't happen"
- **fairness:** "good *data aggregations* do happen"

In social data processing

- all aggregation constraints are security properties

Security vs dependability

processing	dependability	**security**
System	centralized	**distributed**
observations	global	**local**
Environment	neutral	**adversarial**
threats	accidents	**attacks**

Some terminology

Information security

- **secrecy:** "bad *information flows* don't happen"
- **authenticity:** "good *information flows* do happen"

In network computation

- all information flow constraints are security properties

We could also say

Information security

- **confidentiality:** "bad *information flows* don't . . . "
- **integrity:** "good *information flows* do. . . "

Although not synonymous

- secrecy, and confidentiality
- authenticity and integrity

are used interchangeably

Security speak

(overheard at a security conference)

Speaker: Isn't it terrifying that on the Internet we have no privacy?

Charlie: You mean *confidentiality*. Get your terms straight.

Radia: Why do security types insist on inventing their own language?

Mike: It's a denial-of-service attack.

Charlie: You mean *chosen cyphertext attack*. . .

Variants

(a possible assignment of meanings)

Bad information flows

- **secret information:** disclosure prevented
  - e.g., by cryptography
- **private information:** disclosure when authorized
  - information privately owned
- **confidential information:** disclosure restricted
  - penalized when detected

Variants

(a possible assignment of meanings)

Bad information flows about resources

- **secret funds:** it is secret that they exist
  - secret ceremony, secret lover. . .
- **confidential report:** some details confidential
  - content can be disclosed, but not the source
- **private funds:** access restricted by protocol
  - private ceremony, private resort. . .

Variants

(a possible assignment of meanings)

Good information flows

- **authenticity** of a painting, of a letter, of testimony
  - the source of the message is who it says it is
- **integrity** of evidence, of a person
  - the content of the message not been altered, tampered with, compromised

Structure of the course

- Resource security
  - Access control
  - Security models
- Channel security
  - Machines and channels
  - Shared machines and covert channels
  - Information flow security
- Privacy and trust