



Computer Security and Privacy

What is computer security?

Definitions and basic vocabulary

Thomas Bourgeat

Goals

What should you learn today?

- Understand **what is security engineering** and why it is different than most other topics you have studied
- Learn **basic vocabulary** to speak about security and learn **how to characterize the adversary**
- Learn what a **security mechanism** is and how security engineers **show systems are secure**
 - Understand that in security, **composition is not trivial**
 - Understand that **security is relative to the considered adversary**

Why a course on computer security?

What makes **security problems** special?

When we design systems / programs we seek:

- **Correctness**: for a given input, provide expected output
- **Safety**: well-formed programs cannot have bad (even dangerous) outputs
- **Robustness**: cope with errors (input and execution)

We consider what could go wrong, and try to take it into account

What is computer security?

COMPUTER SECURITY

Properties of a computer system must hold
in the presence of a **resourced strategic adversary**

What **Properties**?

TRADITIONAL PROPERTIES

- **Confidentiality** — prevention of unauthorized disclosure of information
(e.g. The adversary should not be able to read my bank statement)
- **Integrity** — prevention of unauthorized modification of information
(e.g. The adversary should not be able to change my bank balance)
- **Availability** — prevention of unauthorized denial of service
(e.g. The adversary should not prevent me accessing my bank account)

What **Properties**?

NOT-SO-TRADITIONAL (BUT IMPORTANT) PROPERTIES

Authenticity

Anonymity

Isolation

Non-repudiation

....

There are many more complex properties relevant in modern security

The Security policy

SECURITY POLICY: a high-level description of the *security properties* that must hold in the system in relation to *assets* and *principals*.

- **Assets (objects):** anything with value (e.g., data, files, memory) that needs to be protected.
- **Principals (subjects):** people, computer programs, services,... (*may not contain the adversary*)

Examples of *security properties* in terms of **principals** and **assets**

Confidentiality	prevention of unauthorized disclosure of information	< authorized users may read a file
Integrity	prevention of unauthorized modification of information	< authorized programs may write a file
Availability	prevention of unauthorized denial of service	< authorized services can access a file

The Resourced Strategic Adversary?

COMPUTER SECURITY

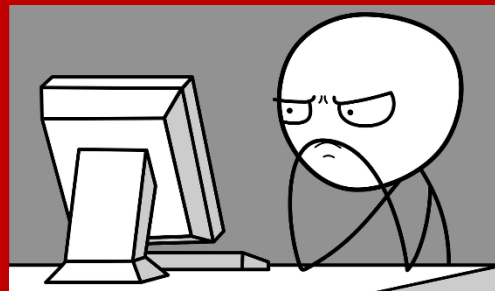
Properties of a computer system must hold in presence of a resourced strategic adversary

THREAT MODEL: describes the **resources** available to the **adversary** and the adversary's capabilities (observe, influence, corrupt,...)

The **adversary** is a malicious entity aiming at breaching the security policy

The **adversary** is **strategic**: the adversary will choose the **optimal** way to use her resources to mount an attack that violates the security properties

THREAT MODELLING IS A VERY HARD TASK!!



Key vocabulary to talk about the adversary

THREAT MODEL

Technical term to define the adversary's capabilities.

The adversary can observe my connection

The adversary can corrupt my machine

The adversary controls a bank employee

VULNERABILITY

Specific weakness that could be exploited by adversaries with interest in a lot of different assets

The banking API is not protected

The password appears in plain text in my screen

THREAT

What is the feared event, the goal of the adversary that we don't want materialized

A hacker wants to retrieve money breaking into the bank's system

A student wants to learn my password by looking over my shoulder

HARM

The bad thing that happens when the threat materializes

The adversary steals money

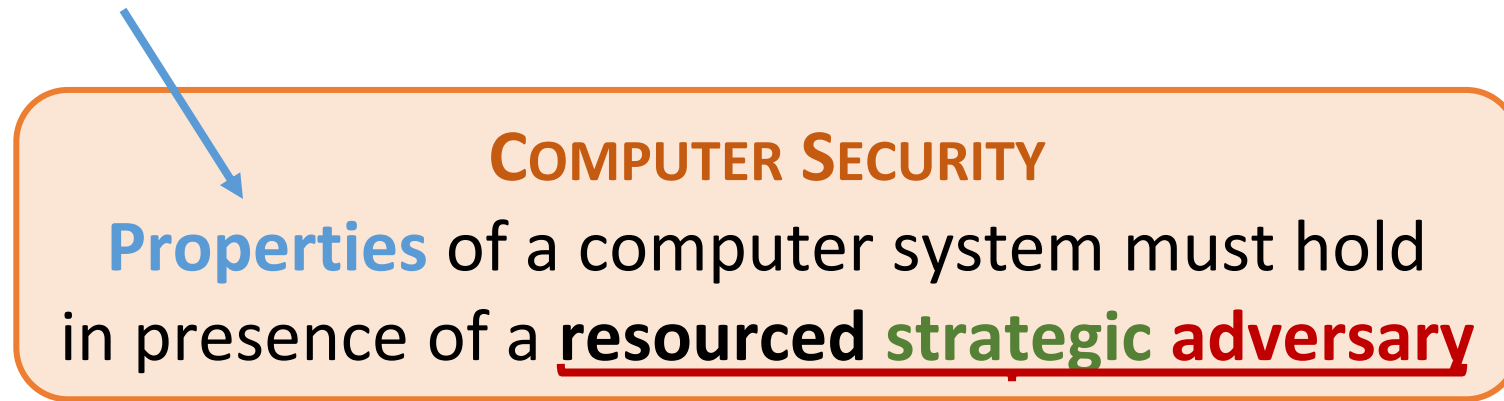
The adversary blocks access to the bank

The adversary learns my password

The adversary reads the message

All together

Defined by the SECURITY POLICY



Described by the THREAT MODEL

If the **properties** hold within the **threat model**, it means that there are no **vulnerabilities** that can be exploited to materialize **threats**, and no **harm** happens

Basic concepts of security engineering

Securing a system

SECURITY MECHANISM: Technical mechanism used to ensure that the security policy is not violated by an adversary within the threat model.

Securing a system

SECURITY MECHANISM: Technical mechanism used to ensure that the security policy is not violated by an adversary within the threat model.

Software (programs) + Hardware + Maths (cryptography) & Distributed systems, people and procedures

Security mechanisms can be engineered!!

Securing a system

SECURITY MECHANISM: Technical mechanism used to ensure that the security policy is not violated by an adversary within the threat model.

Software (programs) + Hardware + Maths (cryptography) & Distributed systems, people and procedures

Security mechanisms can be engineered!!

Example 1

Policy: ensure the log of transactions is not tampered with by a single employee

Mechanism: keep a copy of the log on multiple computers, such that no single employee has access to all of them

Example 2

Policy: ensure messages cannot be read by anyone but the sender and the receiver

Mechanism: encrypt the message before sending

Systems are big! Need security mechanismS

Security **does not necessarily increase linearly** with the number of mechanisms!

Two ways of composing



Defence in depth
As long as one remains
Security policy ✓

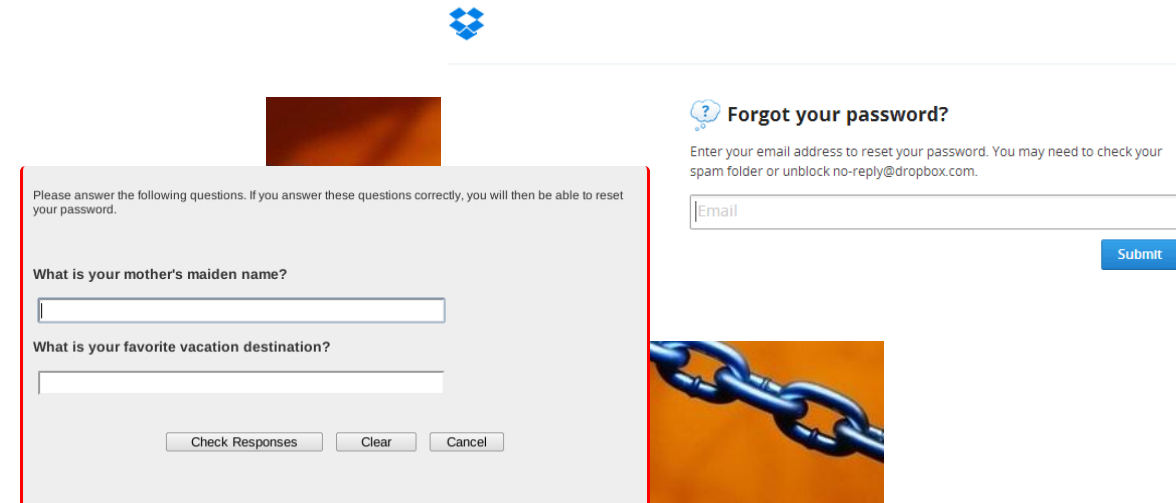


Weakest Link
If any one fails
Security policy ✗

Systems are big! Need security mechanisms



Defence in depth
As long as one remains
Security policy ✓



Weakest Link
If any one fails
Security policy ✗

Humans are part of the system

Social Engineering

Phishing attacks

Bad use of passwords

Weak
Written down
Repeated



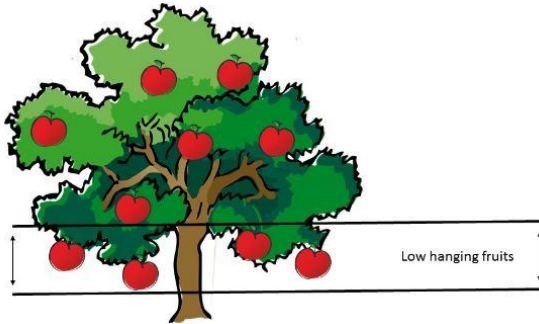
It does not mean you should not care about the rest!!

http://www.slate.com/blogs/future_tense/2016/01/22/calling_humans_the_weakest_link_in_computer_security_is_dangerous.html

How do we show systems are secure?

ATTACKER

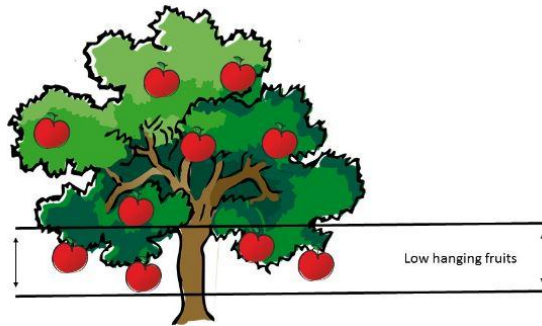
*Just **one way** to violate **one** security property is enough!*
(within the threat model)



How do we show systems are secure?

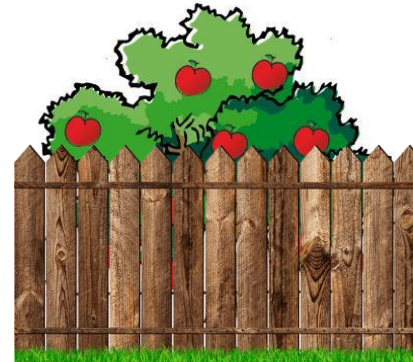
ATTACKER

*Just **one way** to violate **one** security property is enough!*
(within the threat model)



DEFENDER

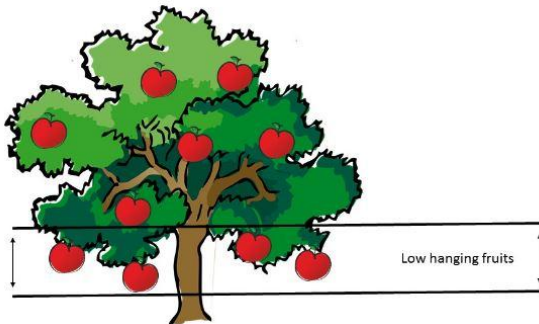
No adversary strategy can violate the security policy



How do we show systems are secure?

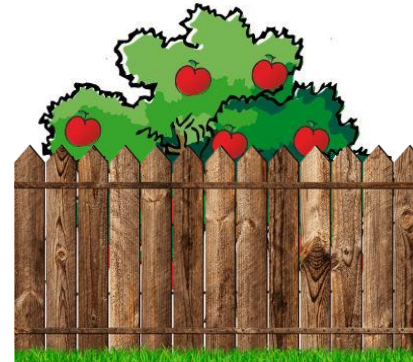
ATTACKER

*Just **one way** to violate **one** security property is enough!*
(within the threat model)



DEFENDER

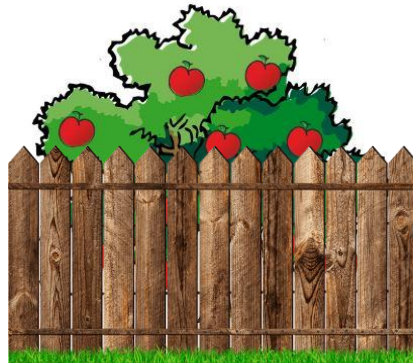
No adversary strategy can violate the security policy



“One of the major problems right now is script kiddies. These are people who just download open source tools that are meant for good, and they point them at whatever they want, press ‘Go,’ and it fires a suite of exploits at a system hoping one of them will work.”

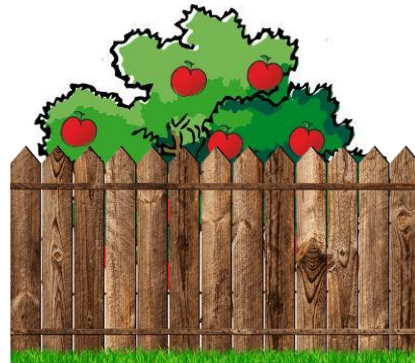
Richard Moore. Security Specialist (IBM)

Is this system secure?



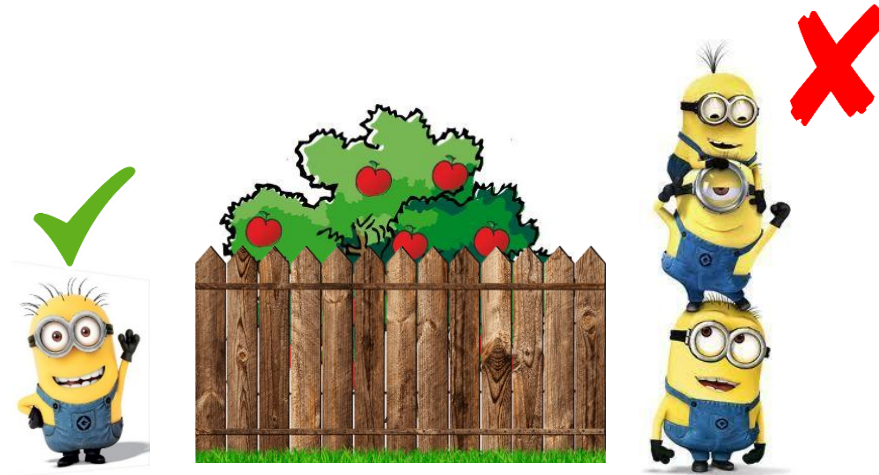
~~Is this system secure?~~

Is this system secure under **this** threat model?



~~Is this system secure?~~

Is this system secure under **this** threat model?



A system is “secure” if an adversary constrained by a **specific threat model** cannot violate the security policy

How do we show systems are secure?

SECURITY ARGUMENT: rigorous argument that the security mechanisms in place are indeed effective in maintaining the security policy (*verbal* or *mathematical*).

Subject to the assumptions of the threat model.

How do we show systems are secure?

SECURITY ARGUMENT: rigorous argument that the security mechanisms in place are indeed effective in maintaining the security policy (*verbal* or *mathematical*).

subject to the assumptions of the threat model.

FOR A THREAT MODEL TO BE USEFUL

The model **must** constrain the adversary, otherwise we cannot make a security argument

How do we show systems are secure?

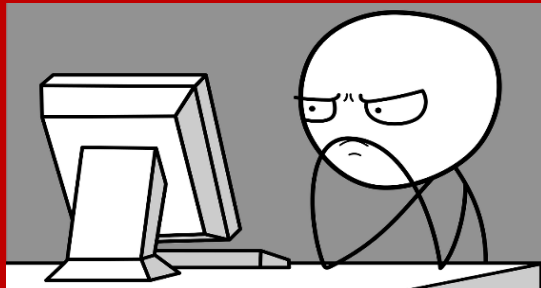
SECURITY ARGUMENT: rigorous argument that the security mechanisms in place are indeed effective in maintaining the security policy (*verbal* or *mathematical*).

subject to the assumptions of the threat model.

FOR A THREAT MODEL TO BE USEFUL

The model **must** constrain the adversary, otherwise we cannot make a security argument

SECURITY ARGUMENTS FOR COMPOSITION OF MECHANISMS ARE VERY HARD TO GET RIGHT!!



Security engineering

1.- High-level specification

- Define the **architecture** of the system (e.g., high level block diagram)
- Define the **security policy** (principals, assets, security properties)
- Define the **threat model** **why is this very important?**

2.- Security design

- Select / Design **security mechanisms**
- State your **security argument**: which controls maintain which properties?

3.- Secure implementation

- **Implement** mechanisms
- Ensure they **conform** to the design model
- Security **testing**

Summary

Security problems always involve an **adversary**

The adversary is **strategic**, will take the most damaging approach

The adversary's capabilities define a **threat model**

Security mechanisms aim at fulfilling a **security policy** within a **threat model**

Showing security implies providing a **security argument**

Goals

What should you learn today?

- Understand **what is security engineering** and why it is different than most other topics you have studied
- Learn **basic vocabulary** to speak about security and learn **how to characterize the adversary**
- Learn what a **security mechanism** is and how security engineers **show systems are secure**
 - Understand that in security, **composition is not trivial**
 - Understand that **security is relative to the considered adversary**