ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

SPRING
SECURITY AND PRIVACY ENGINEERING LABORATORY

# Computer Security (COM-301)
## Principles of computer security

**Carmela Troncoso**

SPRING Lab

carmela.troncoso@epfl.ch

Some slides/ideas adapted from: Philippe Oechslin, George Danezis, Emiliano de Cristofaro, Gianluca Stringhini

# About this course - Aim

- Understand **basic concepts and principles of security** design and engineering that will **outlast current technology**”

- **Model threats** and **think critically** about security problems

- **Basic security mechanisms:** purpose and limitations

# Why makes **security problems** special?

- **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)

# Why makes **security problems** special?

- **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)

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

# What Properties?

## Traditional (CIA)

- **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)*

## Other

Authenticity, anonymity, non-repudiation, forward secrecy

Some properties have no (official) name!!

– security *games* expressing that the system is doing exactly what expected

# What Properties?

## Traditional (CIA)

- **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)*

## Other

Authenticity, anonymity, non-repudiation, forward secrecy

Some properties have no (official) name!!

– security *games* expressing that the system is doing exactly what expected

# What Properties?

## SECURITY GAMES


# What Properties?

## Security Games

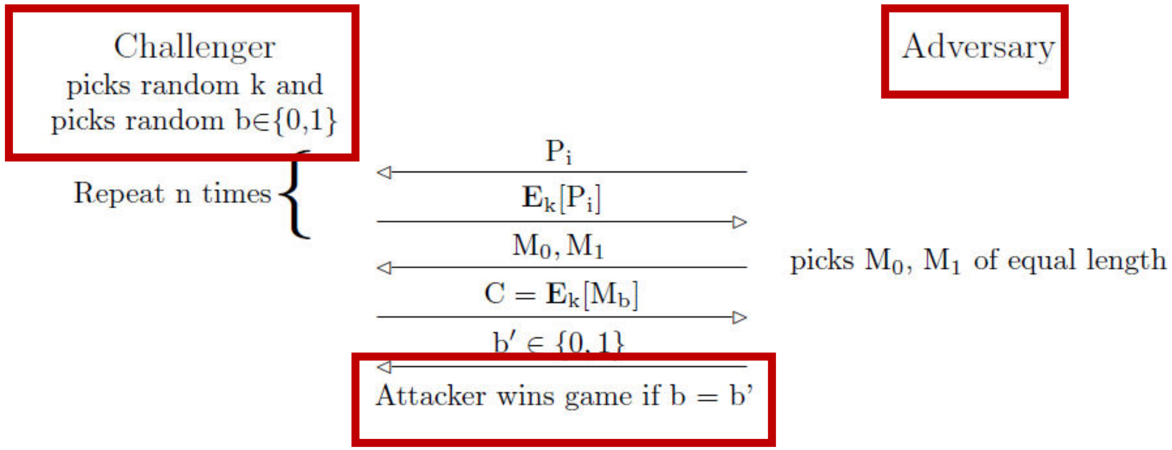


| Challenger | | Adversary |
|---|---|---|
| picks random k and picks random b∈{0,1} | | |
| Repeat n times { | $\xleftarrow{P_i}$ | |
| | $\xrightarrow{E_k[P_i]}$ | |
| | $\xleftarrow{M_0, M_1}$ | picks $M_0$, $M_1$ of equal length |
| | $\xrightarrow{C = E_k[M_b]}$ | |
| | $\xleftarrow{b' \in \{0,1\}}$ | |
| | Attacker wins game if b = b' | |

# What Properties?

## Security Games





# What Properties? – The security policy

A high level description of the **principals**, **assets** and **security properties** that must hold in the system.

- **Principals (subjects)**: people, computer programs, services (entities that can be authenticated) *(may not contain the adversary)*

- **Assets (objects)**: anything with value that needs to be protected.

- **Properties**: usually defined in relation to Principals + Assets

# What Properties? – The security policy

A high level description of the **principals**, **assets** and **security properties** that must hold in the system.

- **Principals (subjects)**: people, computer programs, services (entities that can be authenticated) *(may not contain the adversary)*

- **Assets (objects)**: anything with value that needs to be protected.

- **Properties**: usually defined in relation to Principals + Assets

- **Confidentiality** prevention of unauthorized disclosure of information **< authorized principals may read**
- **Integrity** prevention of unauthorized modification of information **< authorized principals may write**
- **Availability** prevention of unauthorized denial of service **< authorized principals can access the system**

# What Properties? – The security policy

A high level description of the **principals**, **assets** and **security properties** that must hold in the system.

- **Principals (subjects)**: people, computer programs, services (entities that can be authenticated) *(may not contain the adversary)*

- **Assets (objects)**: anything with value that needs to be protected.

- **Properties**: usually defined in relation to Principals + Assets

Requires a high-level idea of the architecture and requirements of the system

- **Confidentiality** prevention of unauthorized disclosure of information **< authorized principals may read**
- **Integrity** prevention of unauthorized modification of information **< authorized principals may write**
- **Availability** prevention of unauthorized denial of service **< authorized principals can access the system**

# What Properties? – The security policy

How is it established?

**Factors**

- Security Engineering
- Business and Marketing
- Risk Management
- Legal and Compliance

Policies need **not** to be static!!

# The Strategic Adversary?

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

**THREAT MODEL**: what are the resources available to the adversary?

What can the adversary**…**

- Observe
- Corrupt / control
- Influence or modify

# The Strategic Adversary?

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

**THREAT MODEL**: what are the resources available to the adversary?

What can the adversary**…**

- Observe
- Corrupt / control
- Influence or modify

**ATTACK**: an intended act against a system or a population of systems that violates a given security policy

**Black vs. white hacker**

# The Strategic Adversary?

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

## THREAT MODEL: what are the resources available to the adversary?

What can the adversary**…**

- Observe
- Corrupt / control
- Influence or modify

**ATTACK**: an intended act against a system or a population of systems that violates a given security policy

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

# The Strategic Adversary?

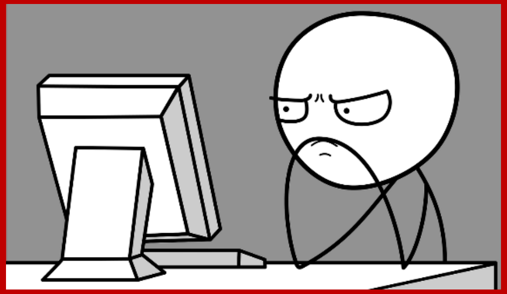
Properties of a computer system must hold
in presence of a resourced **strategic adversary**

## THREAT MODEL: what are the resources available to the adversary?

What can the adversary**…**

- Observe
- Corrupt / control
- Influence or modify

**THREAT MODELLING IS A VERY HARD TASK!!**



**ATTACK**: an intended act against a system or a population of systems that violates a given security policy

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

# The Strategic Adversary?

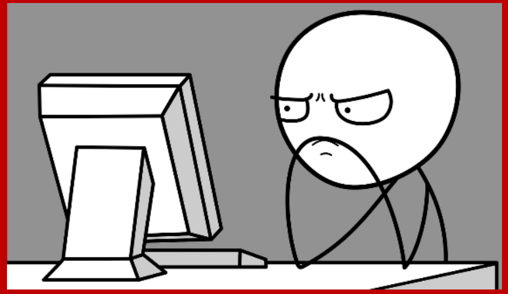
Q: Are Adversary motives important?

**THREAT MODEL**: what are the resources available to the adversary?

What can the adversary**…**

- Observe
- Corrupt / control
- Influence or modify

**THREAT MODELLING IS A VERY HARD TASK!!**



**ATTACK**: an intended act against a system or a population of systems that violates a given security policy

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

# Threat model, threat, harms, and vulnerabilities

**THREAT MODEL**

The adversary's capabilities. Very technical term!!

*The adversary can observe my connection*

*The adversary can corrupt my machine*

*The adversary controls a bank employee*

# Threat model, threat, harms, and vulnerabilities

### THREAT MODEL

The adversary's capabilities. Very technical term!!

*The adversary can observe my connection*

*The adversary can corrupt my machine*

*The adversary controls a bank employee*

### THREAT

Who might attack which assets, using what resources, with what goal, how, and with what probability

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

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

# Threat model, threat, harms, and vulnerabilities

### THREAT MODEL

The adversary's capabilities. Very technical term!!

*The adversary can observe my connection*

*The adversary can corrupt my machine*

*The adversary controls a bank employee*

### THREAT

Who might attack which assets, using what resources, with what goal, how, and with what probability

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

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

### 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 model, threat, harms, and vulnerabilities

### Threat model

The adversary's capabilities. Very technical term!!

*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

Who might attack which assets, using what resources, with what goal, how, and with what probability

*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*

# Example I: State-level adversary

SUSPECTED SPYING

## NSA accused of tapping Swisscom phone lines

*By swissinfo.ch and agencies*

IN DEPTH: NSA SPYING

MAY 27, 2015 - 17:42

**The Swiss Federal Prosecutor's Office is investigating whether America's National Security Agency (NSA) tapped Swisscom phone lines. The accusation comes from Austrian parliamentarian and whistleblower Peter Pilz.**

According to Pilz, Germany's foreign intelligence service gathered data from several countries on behalf of the NSA. Pilz presented a number of relevant documents in Bern on Wednesday, including a list of key transmission lines. A list from 2005 showed nine Swisscom lines ending in Switzerland: seven in Zurich, two in Geneva.


Swisscom phone lines have allegedly been tapped by the NSA

- What is the system under attack?
- Who are the principals?
- What are their assets?
- What are the security properties to maintain?

- What is the **threat model**?
- What is the **security policy**?

https://www.swissinfo.ch/eng/suspected-spying_nsa-accused-of-tapping-swisscom-phone-lines/41454902

# Example II: Solo young hackers

Vaud — 16 février 2018 11:34; *Act: 16.02.2018 11:53*

## Mots de passe et photos intimes dérobés à l'Unil

*Un étudiant a piégé des ordinateurs publics, notamment à la bibliothèque de l'Unithèque, pour accéder aux comptes et télécharger des fichiers privés.*


Un étudiant a obtenu des centaines de photos intimes en piratant des ordinateurs publics de l'Unil. (Photo: Keystone)

on off i

J'aime

0 Partager

Des keyloggers, dispositifs permettant d'enregistrer ce que les utilisateurs tapent sur leur clavier, avaient été installés sur plusieurs ordinateurs de l'Université de Lausanne. Le pirate, un étudiant, se servait ensuite des informations collectées pour accéder frauduleusement à quelque 2700 comptes personnels appartenant à ses petits camarades et en télécharger les fichiers privés.

- What is the system under attack?
- Who are the principals?
- What are their assets?
- What are the security properties to maintain?

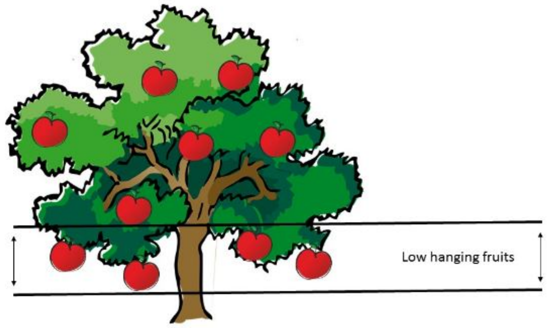
- What is the **threat model**?
- What is the **security policy**?

https://www.20min.ch/ro/news/vaud/story/Mots-de-passes-et-photos-intimes-derobes-a-l-UNIL-23553124?httpredirect

# Asymmetry adversary vs. defender

**ATTACKER**

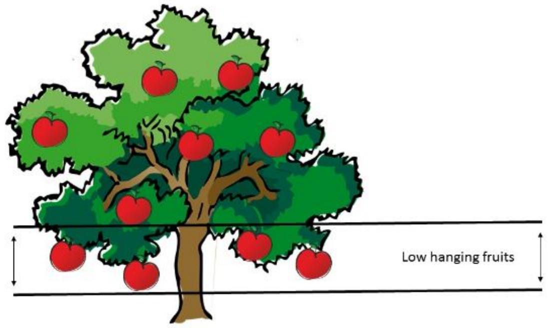
*Just* ***one*** **way** to violate ***one*** security property is enough!
(within the threat model)



# Asymmetry adversary vs. defender

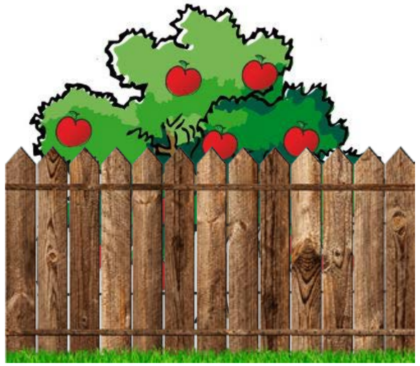
**ATTACKER**

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




**DEFENDER**

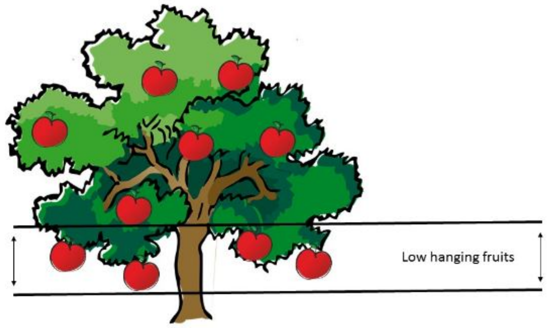
**No adversary strategy** can violate the security policy



# Asymmetry adversary vs. defender

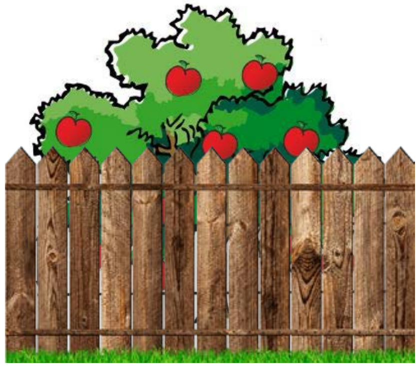
**ATTACKER**

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




**DEFENDER**

**No adversary strategy** can violate the security policy

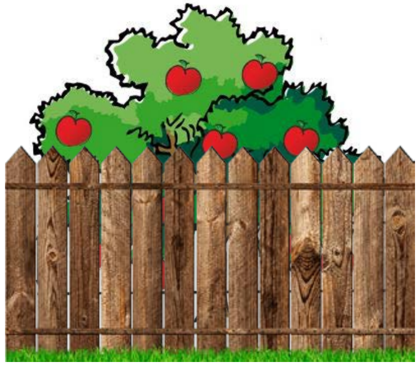


*"One of the major problems right now is script kiddies. These are people who just download open source tools that are meant for good, and they point them at whatever they want, press 'Go,' and it fires a suite of exploits at a system hoping one of them will work."*

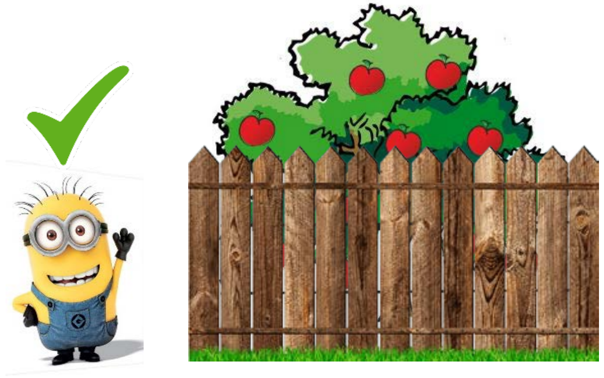
Richard Moore. Security Specialist (IBM)

https://securityintelligence.com/security-gamification-engineer-richard-moore-proves-that-anyone-can-be-a-hacker/

# Is this system secure?



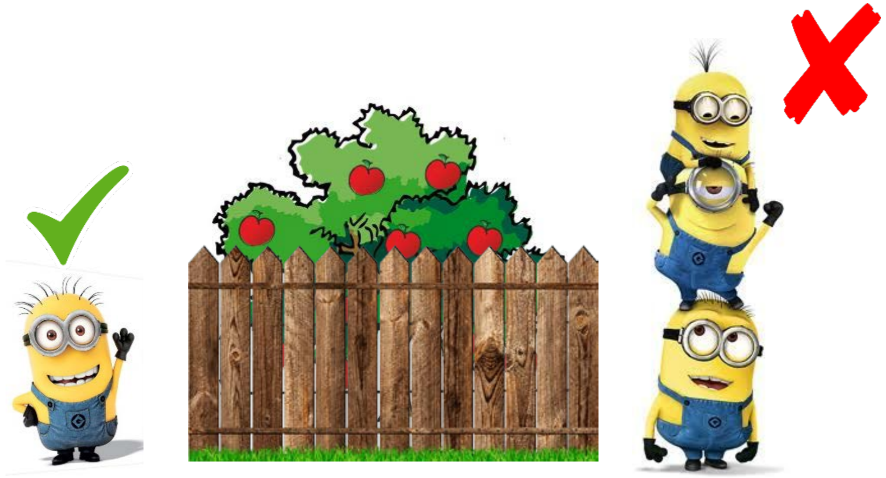
# Is this system secure?



# Is this system secure?

# ~~Is this system secure?~~

# Is this system secure under **this** thread model?



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

**Exercise**: Observe security systems around you and:

- What is the security policy?
- What is the threat model?
- How / why could it fail?

# How do we show it is secure?

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

# How do we show it is 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.

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

# How do we show it is 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.

**USEFUL MODELS**

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

# How do we show it is secure?

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

(as such they can be engineered!!)

**Example**:

- <u>Policy</u>: ensure the log of transactions is not tampered with by a single employee

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

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

# How do we show it is secure?

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

(as such they can be engineered!!)

**Example**:

- Policy: ensure the log of transactions is not tampered with by a single employee **(+ secret from any employee?)**
- Mechanism: keep a copy of the log on multiple computers, such that no single employee has access to all of them

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

# Basic principles to build Security Mechanisms

**READING**: J. Saltzer and M. Schroeder. *The Protection of Information in Computer Systems*. Fourth ACM Symposium on Operating Systems Principles (October 1973)
(Intro & Section 1)

*“The term "security" describes techniques that control who may use or modify the computer or the information contained in it.”* ← ***Security mechanisms***

https://www.acsac.org/secshelf/papers/protection_information.pdf

# Basic principles to build Security Mechanisms

**READING**: J. Saltzer and M. Schroeder. *The Protection of Information in Computer Systems*. Fourth ACM Symposium on Operating Systems Principles (October 1973)
(Intro & Section 1)

*“The term "security" describes techniques that control who may use or modify the computer or the information contained in it.”* ← ***Security mechanisms***

*“Principles **guide** the design and contribute to an implementation without security flaws”*

https://www.acsac.org/secshelf/papers/protection_information.pdf

# Basic principles to build Security Mechanisms

**READING**: J. Saltzer and M. Schroeder. *The Protection of Information in Computer Systems*. Fourth ACM Symposium on Operating Systems Principles (October 1973)
(Intro & Section 1)

*"The term "security" describes techniques that control who may use or modify the computer or the information contained in it."* ← ***Security mechanisms***

*"Principles **guide** the design and contribute to an implementation without security flaws"*

Not must-do, but yes must-try
Need good reasons to not follow them

https://www.acsac.org/secshelf/papers/protection_information.pdf

# 1 - Economy of mechanism

**"Keep the [security mechanism / implementation] design as simple and small as possible" [SS75]**

# 1 - Economy of mechanism

**“Keep the [security mechanism / implementation] design as simple and small as possible” [SS75]**

- Operational testing is not appropriate to evaluate security.

[Penetration testing may be valuable]

- Easier to audit and verify.

**homework**

( U.S. Navy 60s )

# 1 - Economy of mechanism

**"Keep the [security mechanism / implementation] design as simple and small as possible" [SS75]**


**homework**
( U.S. Navy 60s )

- Operational testing is not appropriate to evaluate security.
[Penetration testing may be valuable]

- Easier to audit and verify.

"**Trusted Computing Base" (TCB)**: Every component of the system on which the security policy replies upon

# 1 - Economy of mechanism

**"Keep the [security mechanism / implementation] design as simple and small as possible" [SS75]**

- Operational testing is not appropriate to evaluate security.

[Penetration testing may be valuable]

- Easier to audit and verify.


**homework**
( U.S. Navy 60s )

**"Trusted Computing Base" (TCB):** Every component of the system on which the security policy replies upon

# The “Trusted Computing Base”

**Every component** of the system on which the security policy relies.

→Hardware, firmware, software

# The "Trusted Computing Base"

**Every component** of the system on which the security policy relies.

→Hardware, firmware, software

if it goes wrong → Security policy may be violated

And if something goes wrong outside?

# The "Trusted Computing Base"

**Every component** of the system on which the security policy relies.

→Hardware, firmware, software

if it goes wrong

Security policy may be violated

And if something goes wrong outside? The policy holds!

# The “Trusted Computing Base”

**Every component** of the system on which the security policy relies.

→Hardware, firmware, software

**TRUSTED?** Economic of mechanism → to ease verification → Needs to be kept small!!!

# The "Trusted Computing Base"

**Every component** of the system on which the security policy relies.

→Hardware, firmware, software

Minimal TCB:
minimize **attack surface**!

**TRUSTED?** Economic of mechanism → to ease verification → Needs to be kept small!!!

# The "Trusted Computing Base"

**Every component** of the system on which the security policy relies.

→Hardware, firmware, software

Minimal TCB:
minimize **attack surface**!

**TRUSTED?**

Economic of mechanism → to ease verification → Needs to be kept small!!!

Only proper use of the verb "to trust" in Security Engineering: "X trusts Y will do Z"

# 1 - Economy of mechanism

**“Keep the [security mechanism / implementation] design as simple and small as possible” [SS75]**

- Operational testing is not appropriate to evaluate security.

[Penetration testing may be valuable]

- Easier to audit and verify.

THE KISS PRINCIPLE KISS

**homework**
( U.S. Navy 60s )

**“Trusted Computing Base” (TCB):** Every component of the system on which the security policy replies upon

Needs to be kept small!!!

# 2 – Fail-safe defaults

***“Base access decisions on permission rather than exclusion”*****[SS75]**

# 2 – Fail-safe defaults

***"Base access decisions on permission rather than exclusion"*[SS75]**

- Goal: if something fails, be as secure as if it does not fail
  → errors / uncertainty should err on the side of the security policy

- Whitelist, do not blacklist
  → lack of permission is easy to detect and solve

**- Do not** try to fix!!

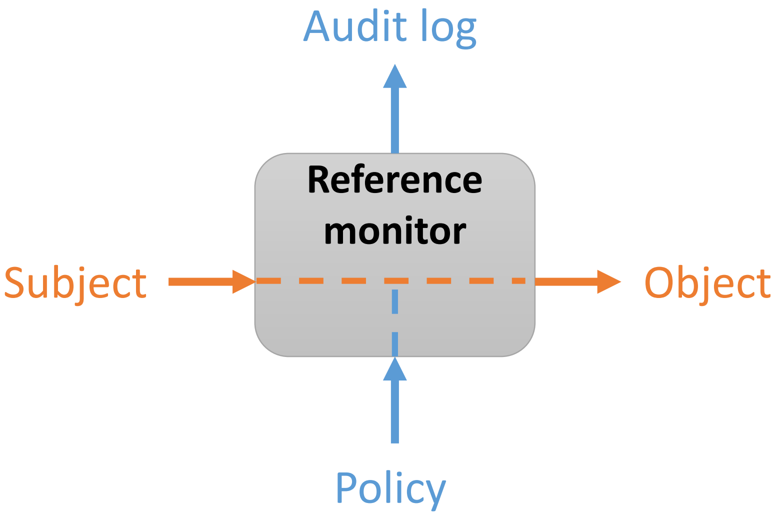
- Examples:
  - Automated doors: if cannot close, open
  - [Integrity] Form input: if no permission to write in X, do not write anywhere

# 3 – Complete mediation

**“Every access to every object must be checked for authority” [SS75]**

# 3 – Complete mediation

**“Every access to every object must be checked for authority” [SS75]**




mediates **ALL** actions and ensures they are according to the policy

**Difficult to implement**

- Performance?
  - Boosting reduces security
- Time to check vs. time to use
- Modern distributed systems
  - You can only check what you see!

# 4 – Open design

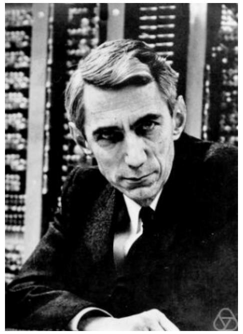
**“The design should not be secret” [SS75]**

*“The design of a system should not require secrecy”*

**Kerckhoffs**
*La Cryptographie Militaire*
(*1883*)

*"The enemy knows the system"*

*"one ought to design systems under the assumption that the enemy will immediately gain full familiarity with them”*



**Shannon**
*Communication Theory of Secrecy Systems*
(*1949*)

*“The Paradox of the Secrecy About Secrecy”*

**Baran**
*Security, secrecy, and tamper-free considerations*
(*1964*)

https://www.rand.org/pubs/research_memoranda/RM3765/RM3765.chapter2.html

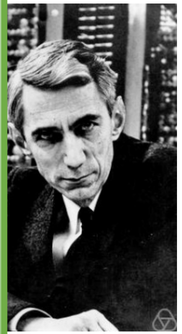
# 4 – Open design

**“The design should not be secret” [SS75]**

*“The design of a system should not require secrecy”*

**Kerck**
*La Cryptogra*
(*18*

**Crypto**: only the key must be secret
**Authentication**: only keep password secret
**Obfuscation**: only keep noise generation parameter secret



**Shannon**
*Communication Theory of Secrecy Systems*
(*1949*)

*“The Paradox of the Secrecy About Secrecy”*

**Baran**
*Security, secrecy, and tamper-free considerations*
(*1964*)

https://www.rand.org/pubs/research_memoranda/RM3765/RM3765.chapter2.html

# 4 – Open design

**“The design should not be secret” [SS75]**

- Open design = better & easier auditing

- Secrecy is unrealistic!!
  Way to build a bad threat model!

- Famous failures:
  - DVD encryption
  - GSM encryption

**Linus’ law**: *"given enough eyeballs, all bugs are shallow"*

**Raymond**
*The Cathedral and the Bazaar* (*1997*)

# 4 – Open design

**“The design should not be secret” [SS75]**

- Open design = better & easier auditing

- Secrecy is unrealistic!!
  Way to build a bad threat model!

- Famous failures:
  - DVD encryption
  - GSM encryption

**Linus’ law**: *"given enough eyeballs, all bugs are shallow"*

**Raymond**
*The Cathedral and the Bazaar*
(*1997*)

**Key principle behind the academic discipline devoted to understanding computer security**

# 5 – Separation of privilege

**“No single accident, deception, or breach of trust is sufficient to compromise the protected information” [SS75]**

# 5 – Separation of privilege

*A **privilege** allows a user to perform an action on a computer system, e.g., create a file in a directory, access a device, write to a socket for communicating over the Internet.*

> **"No single accident, deception, or breach of trust is sufficient to compromise the protected information" [SS75]**

- Require multiple conditions to execute an action
    - Two keys to open a safe
    - Two-factor authentication

- Problems
    - Availability?
    - Responsibility?
    - Complexity!

# 6 – Least privilege

**“Every program and every user of the system should operate using the least set of privileges necessary to complete the job” [SS75]**

# 6 – Least privilege

**"Every program and every user of the system should operate using the least set of privileges necessary to complete the job" [SS75]**

- Rights added as needed, discarded after use

- Damage control
  - Minimize high privilege actions & interactions

- Need-to-know principle
  - Guest accounts @ EPFL
  - Data minimization principle (Data Protection)

# 6 – Least privilege

**“Every program and every user of the system should operate using the least set of privileges necessary to complete the job” [SS75]**

- Rights added as needed, discarded after use

- Damage control
  - Minimize high privilege actions & interactions

**Related to what other principle?**

- Need-to-know principle
  - Guest accounts @ EPFL
  - Data minimization principle (Data Protection)

# 7 – Least common mechanism

**“Minimize the amount of mechanism common to more than one user and depended on by all users” [SS75]**

*“Every shared mechanism represents a potential information path between users and must be designed with great care to be sure it does not unintentionally compromise security”*

**COVERT CHANNELS**

- Storage (/tmp)
- Timing (shared CPU, queue, cache)

# 7 – Least common mechanism

**“Minimize the amount of mechanism common to more than one user and depended on by all users” [SS75]**

*“Every shared mechanism represents a potential information path between users and must be designed with great care to be sure it does not unintentionally compromise security”*

- Economy of mechanism
    - (Design) Interactions make validation of security design hard.
    - (Implementation) Interactions may lead to unintentional leaks of information.

**COVERT CHANNELS**

- Storage (/tmp)
- Timing (shared CPU, queue, cache)

# 7 – Least common mechanism

**“Minimize the amount of mechanism common to more than one user and depended on by all users” [SS75]**

*“Every shared mechanism represents a potential information path between users and must be designed with great care to be sure it does not unintentionally compromise security”*

- Economy of mechanism
  - (Design) Interactions make validation of security design hard.
  - (Implementation) Interactions may lead to unintentional leaks of information.

**COVERT CHANNELS**
- Storage (/tmp)
- Timing (shared CPU, queue, cache)

**Isolation**
**Virtual Machines**

# 7 – Least common mechanism

**“Minimize the amount of mechanism common to more than one user and depended on by all users” [SS75]**

*“Every shared mechanism represents a potential information path between users and must be designed with great care to be sure it does not unintentionally compromise security”*

- Economy of mechanism
  - (Design) Interactions make validation of security design hard.
  - (Implementation) Interactions may lead to unintentional leaks of information.

**Cautionary Note:**
**Mechanism != Code!!!**

**Isolation**
**Virtual Machines**

**COVERT CHANNELS**
- Storage (/tmp)
- Timing (shared CPU, queue, cache)

# 8 – Psychological acceptability

**“It is essential that the human interface be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly” [SS75]**

# 8 – Psychological acceptability

**“It is essential that the human interface be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly” [SS75]**

- Hide complexity introduced by security mechanisms
  - The mechanisms should not make the resource more difficult to access than if it was not present

- Mental model of the (honest) users must match security policy and security mechanisms

- Cultural acceptability:
  - (Authentication) Photographs that must uncover faces.
  - (Safety) Register of everyone who sleeps in a dorm.

# Two extra principles from physical security
# 9 - Work factor

**“Compare the cost of circumventing the mechanism with the resources of a potential attacker” [SS75]**

# Two extra principles from physical security
# 9 - Work factor

DIFFICULT TO TRANSPOSE TO COMPUTER SECURITY!!

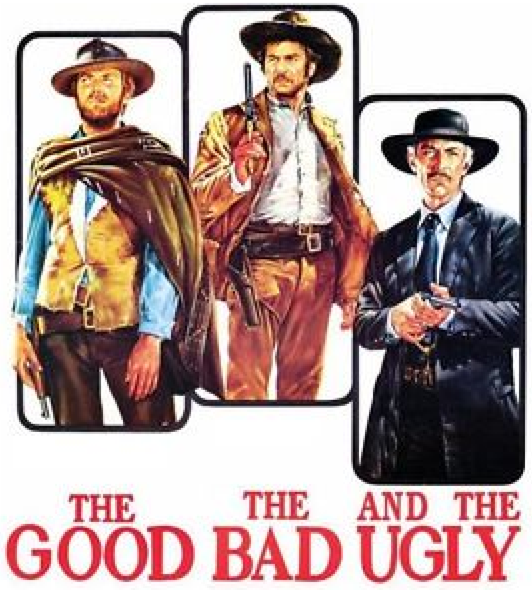
**“Compare the cost of circumventing the mechanism with the resources of a potential attacker” [SS75]**

# Two extra principles from physical security
# 9 - Work factor

**DIFFICULT TO TRANSPOSE TO COMPUTER SECURITY!!**

**"Compare the cost of circumventing the mechanism with the resources of a potential attacker" [SS75]**

It helps **refining** the threat mode!




**Difficult to quantify**

Defining **cost?**

- cost of compromising insiders?
- cost of finding a bug?
- monetization?

# Two extra principles from physical security
# 10 - Compromise recording

**“Reliably record that a compromise of information has occurred [...] in place of more elaborate mechanisms that completely prevent loss” [SS75]**

# Two extra principles from physical security
# 10 - Compromise recording

DIFFICULT TO TRANSPOSE TO COMPUTER SECURITY!!

**“Reliably record that a compromise of information has occurred [...] in place of more elaborate mechanisms that completely prevent loss” [SS75]**

# Two extra principles from physical security
# 10 - Compromise recording

DIFFICULT TO TRANSPOSE
TO COMPUTER SECURITY!!

**"Reliably record that a compromise of information has occurred [...] in place of more elaborate mechanisms that completely prevent loss" [SS75]**

Keep **tamper-evidence logs**

May enable recovery (integrity)


Logs **are not magic**:

- What if you cannot recover? (Confidentiality)
- How to keep integrity? (Blockchain!)
- Logs may be a vulnerability (Privacy)?
- Logging the log? (Availability)

**Detecting the compromise may be difficult (or expensive)**

# Summary of the day

- Security problems always involve an **adversary**
    - The adversary is always **strategic**
    - The adversary's capabilities define a **threat model**
    - **Security mechanisms** aim at fulfilling a **security policy** <u>**within**</u> **a threat model**

- Principles allow us to identify safe and unsafe *patterns* in the security engineering process
    - Do not use principles as a blind checklist!
    - Use principles as tools to weight design decisions.
        - Having examples and counter examples help

# Basic principles to build Security Mechanisms

1. Economy of mechanism
2. Fail-safe defaults
3. Complete mediation
4. Open Design
5. Separation of Privilege
6. Least Privilege
7. Least Common Mechanism
8. Psychological Acceptability

\+ Work Factor
\+ Compromise Recording