Computer Security (COM-301)

Mandatory Access Control
Confidentiality security models

Carmela Troncoso

SPRING Lab

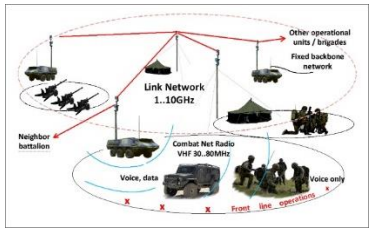
carmela.troncoso@epfl.ch

Some slides/ideas adapted from: George Danezis

We talked about DAC

Mandatory Access Control (MAC)

Central security policy *assigns permissions*



Discretionary Access Control (DAC)

Object owners *assign permissions*

STRAVA

Theoretical lecture ahead!

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

Mandatory Access Control

Access to and operations on resources **are determined** by the security policy

- “owner” may not exist or not have power to set permissions against policy
- the security policy **must** be enforced despite subjects trying to subvert it

Security models

SECURITY MODEL: a **design pattern** for a specific security property or set of properties

When faced with a standard security problem → use well-known model!

Security models

SECURITY MODEL: a **design pattern** for a specific security property or set of properties

When faced with a standard security problem → use well-known model!

The devil is in the details!

Many aspects not covered by the model!

- who are the subjects?
- what are the objects?
- what mechanisms to use to implement it?

Bell-La Padula (BLP) model: Protecting confidentiality

Subjects **S** and objects **O** associated to a **level** of confidentiality

Subjects access rights are defined by four attributes:

Execute: the subject cannot see or modify the object, but can run it

Read: the subject can only see the object but cannot modify it

Append: the subject cannot read the object, but can add attach new content

Write: the subject can see the object and add content or modify existing content

These access rights are defined in an access control matrix

D. Bell and L. LaPadula. "Secure computer systems: Unified exposition and Multics interpretation". Technical Report ESD-TR-75-306, MITRE Corp., March 1976.

Level function for objects: **Classification**

Objects are associated to a **Security Level**
(they have a **label**, and belong to one or more **categories**)

Security Level = (Classification, {set of categories})

Classification - total order of **labels** *(e.g., Unclassified, Confidential, Secret, Top Secret)*

Categories – compartments of objects with a common topic (*e.g., Nuclear, NATO, Crypto*)

Classification: dominance relationship

DOMINANCE RELATIONSHIP

A security level (l1,c1) “dominates” (l2, c2) *if and only if* l1 >= l2 **and** c2 is a subset of c1

Labels: **A**dmin < **N**urse < **S**urgeon < **D**octor
Categories: DEMOGRAPHICS, ANALYSIS, RESULTS

Classification: dominance relationship

DOMINANCE RELATIONSHIP

A security level (l1,c1) “dominates” (l2, c2) *if and only if* l1 >= l2 **and** c2 is a subset of c1

Labels: **A**dmin < **N**urse < **S**urgeon < **D**octor
Categories: DEMOGRAPHICS, ANALYSIS, RESULTS

Which statements are true?

(D, {}) dominates (S, {})

(S, {}) dominates (N, {RESULTS})

(S, {DEMOGRAPHICS,RESULTS}) dominates (N, {DEMOGRAPHICS})

(D, {ANALYSIS,RESULTS}) dominates (S, {DEMOGRAPHICS})

What level dominates them all?
What level dominates only itself?

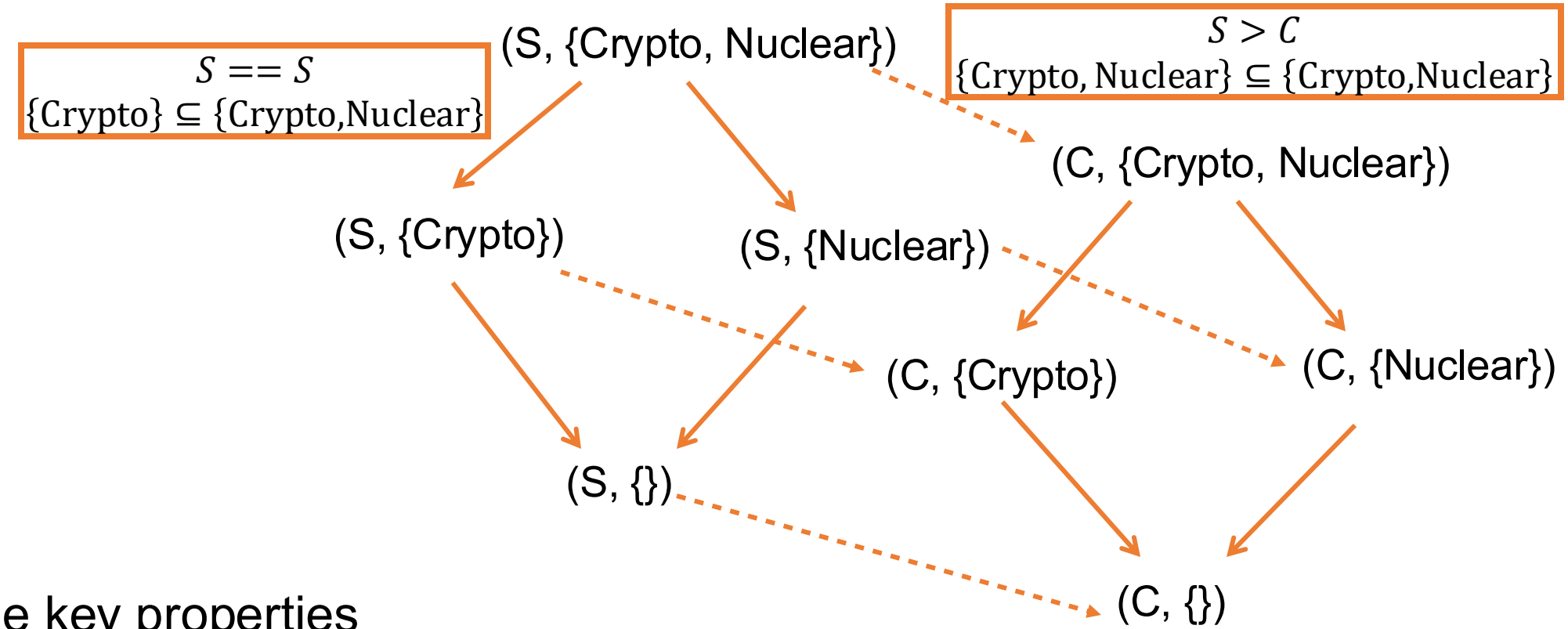
Dominance lattice

DOMINANCE RELATIONSHIP

A level (c1, l1) "dominates" (c2, l2)

iff c1 >= c2 and l2 is a subset of l1

Labels: C < S **Categories**: Crypto, Nuclear



Three key properties

(a) Dominates is transitive.

(b) Top and bottom elements.

(c) Only **partial** order.

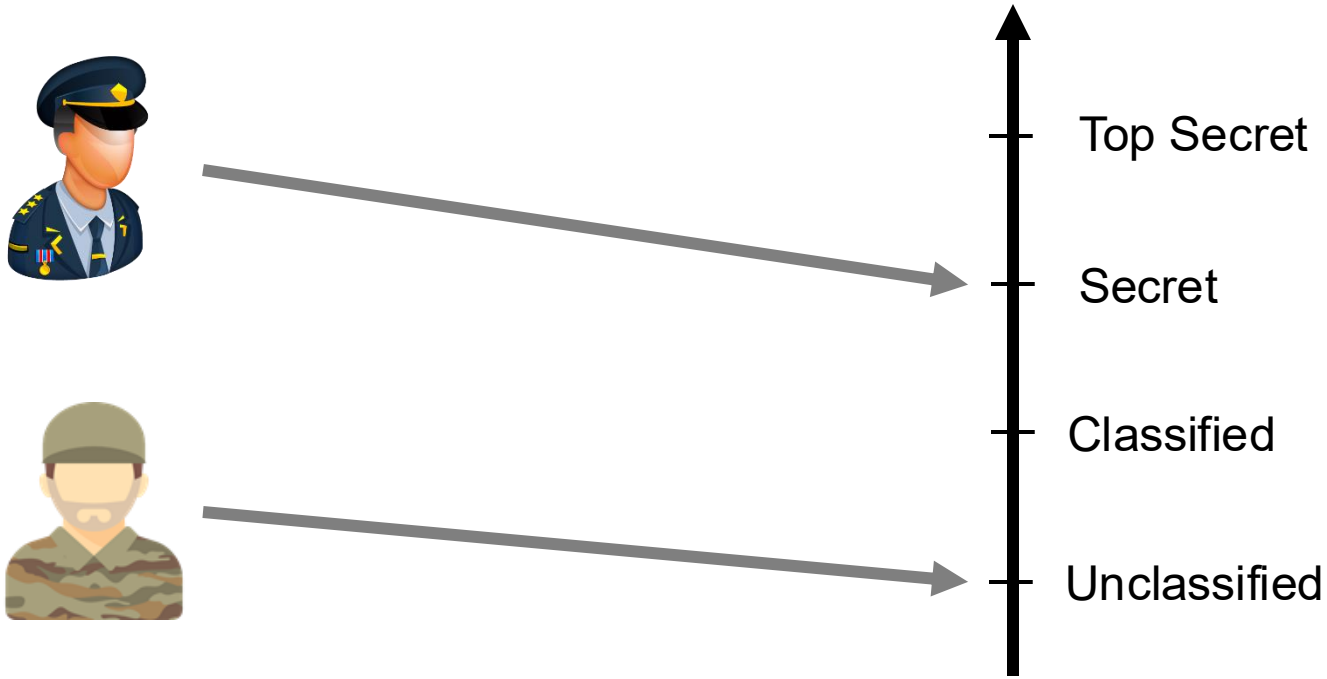
Level function for subjects: **Clearance level**

BLP calls this also “classification”

Clearance – maximum security level a subject has been assigned: ***clearance level(S_i)***

Current security level – subjects can operate at lower security levels: ***current-level(S_i)***

*level(S_i) **must** dominate current-level(S_i) !!!*



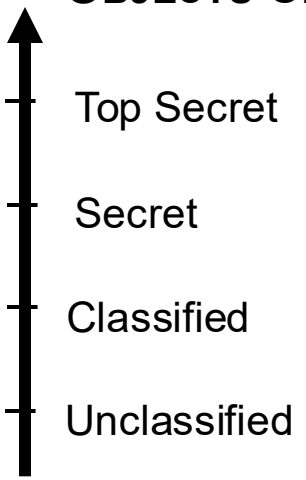
BLP System: ss-property

SIMPLE SECURITY PROPERTY (SS-PROPERTY)

If (subject, object, r) is a current access, then level(subject) dominates level(object)

CLEARANCE: SECRET

OBJECTS CLASSIFICATION

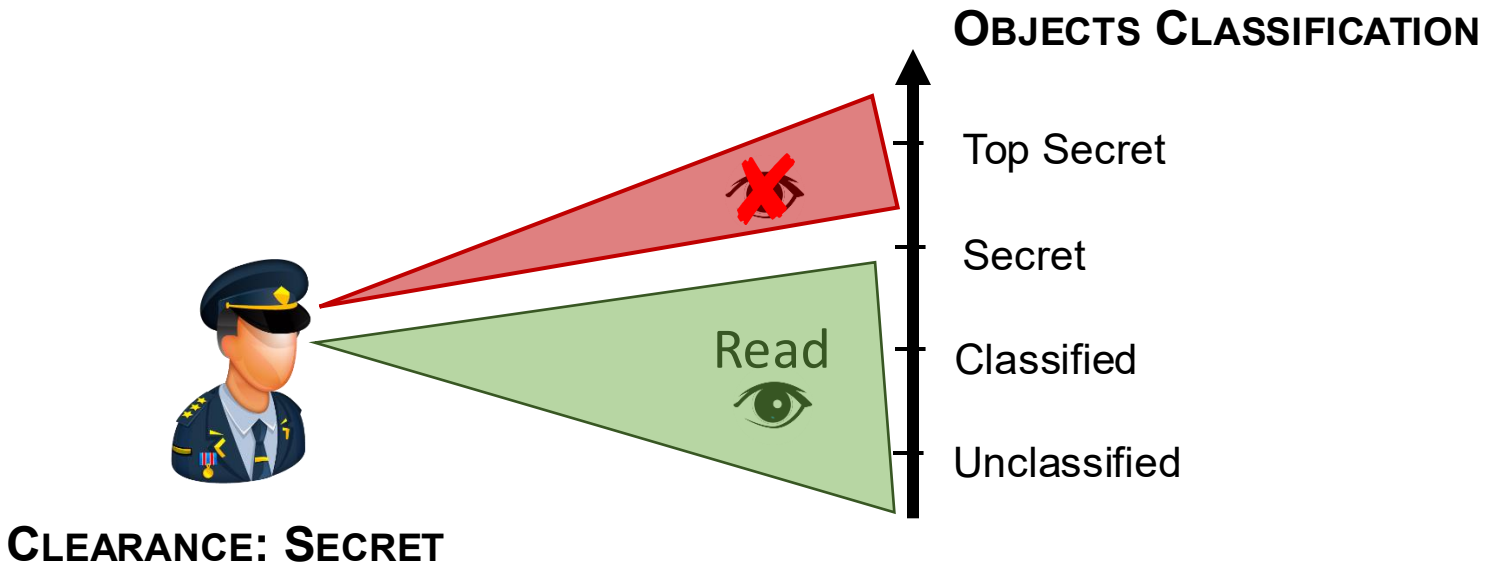


BLP System: ss-property

SIMPLE SECURITY PROPERTY (SS-PROPERTY)

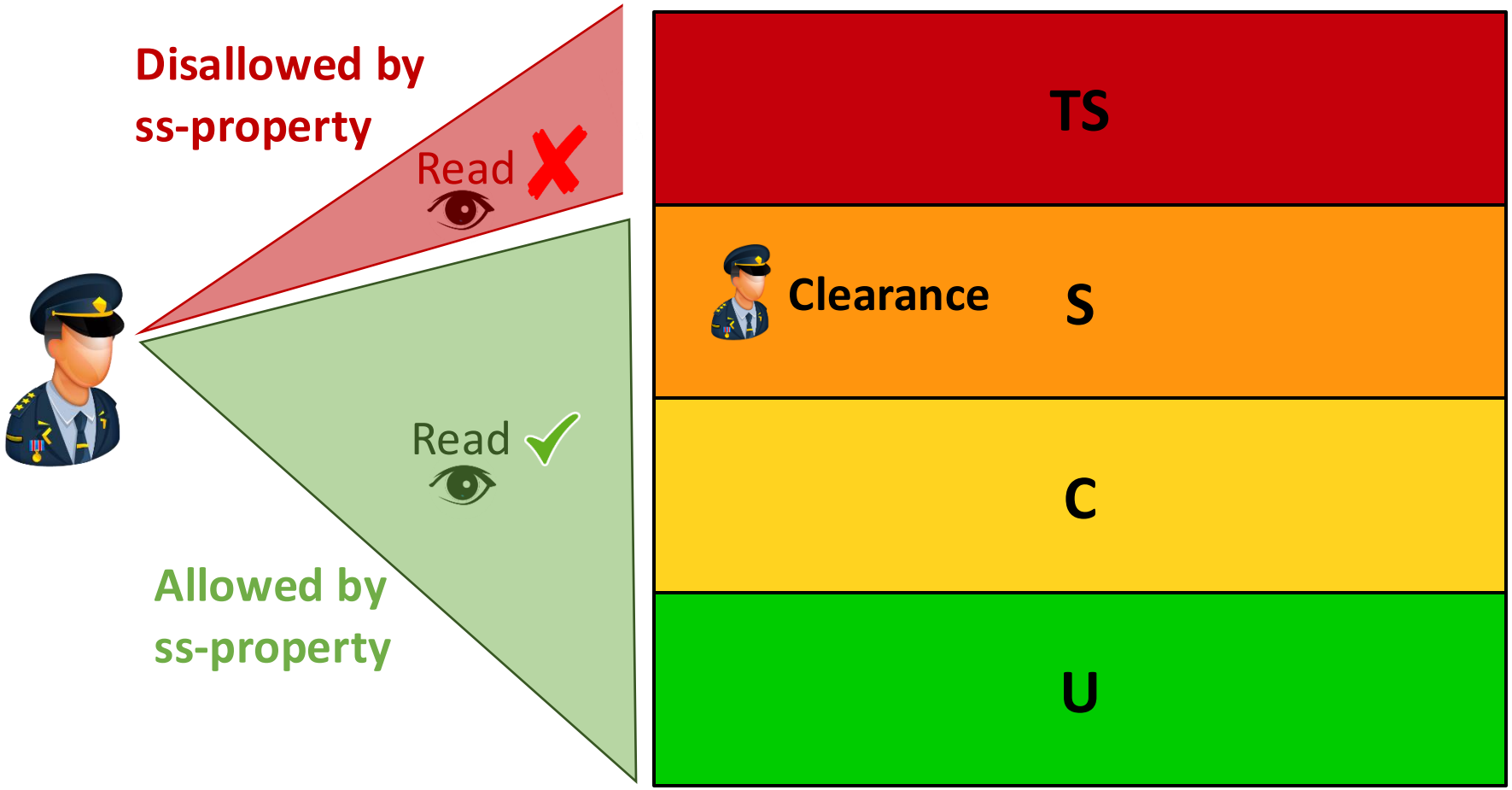
If (subject, object, r) is a current access, then level(subject) dominates level(object)

No Read Up (NRU)



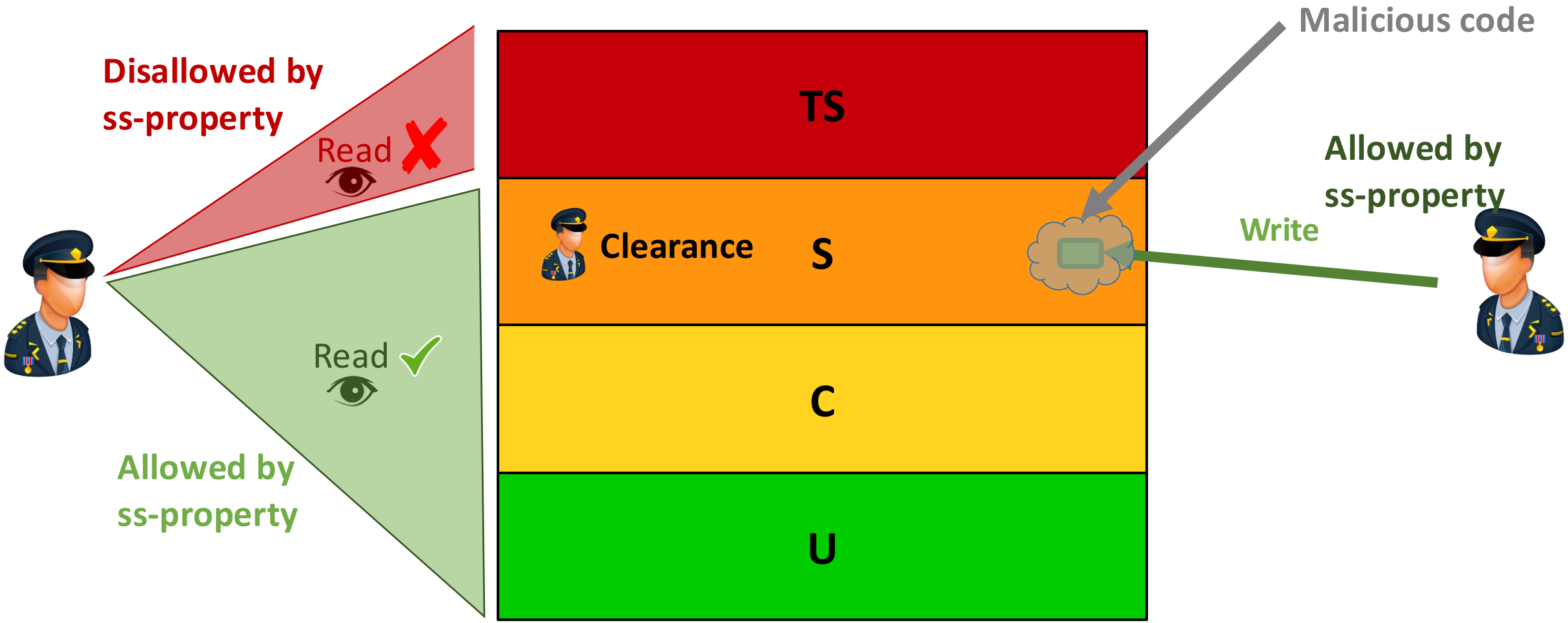
Why is the ss-property not sufficient?

No Read Up (NRU)



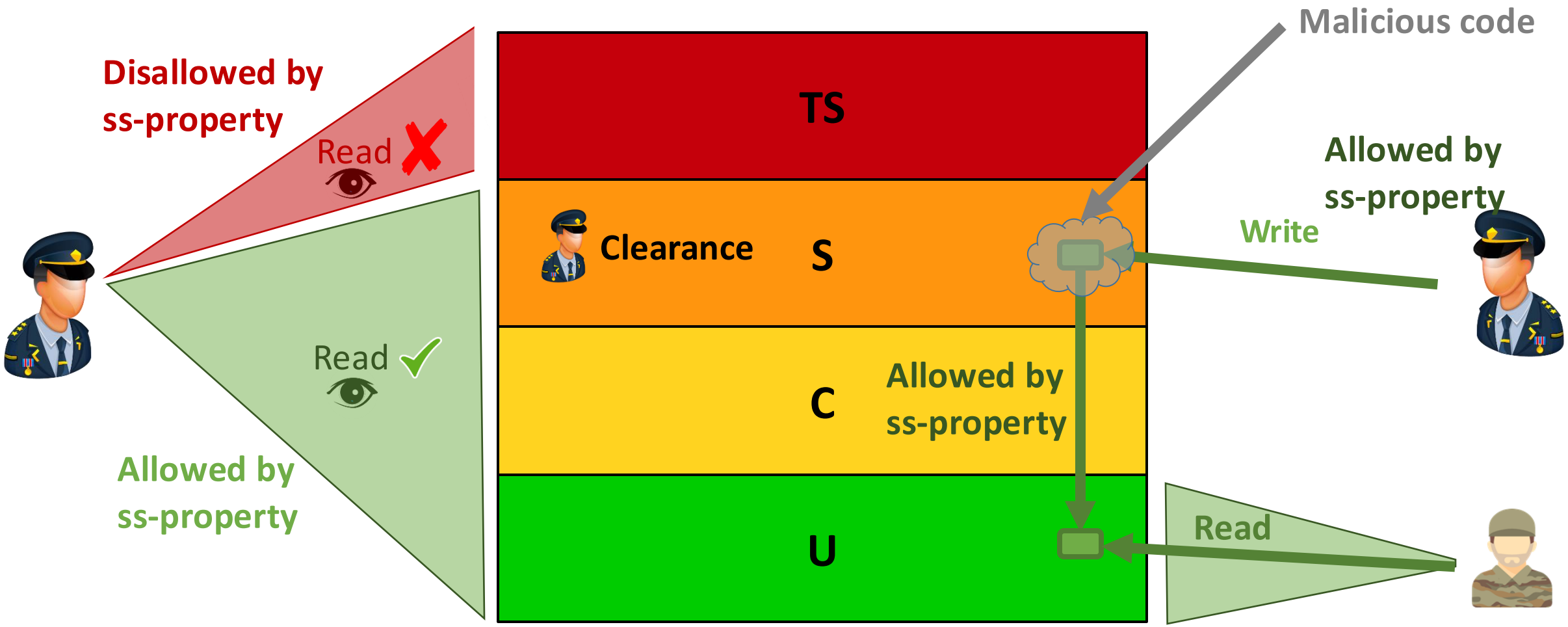
Why is the ss-property not sufficient?

No Read Up (NRU)



Why is the ss-property not sufficient?

No Read Up (NRU)



BLP System: *-property

STAR PROPERTY (*-PROPERTY)

if a subject has simultaneous "observe" (r,w) access to O_1
and "alter" (a,w) access to O_2 then level (O_2) dominates level (O_1)

CLEARANCE: SECRET

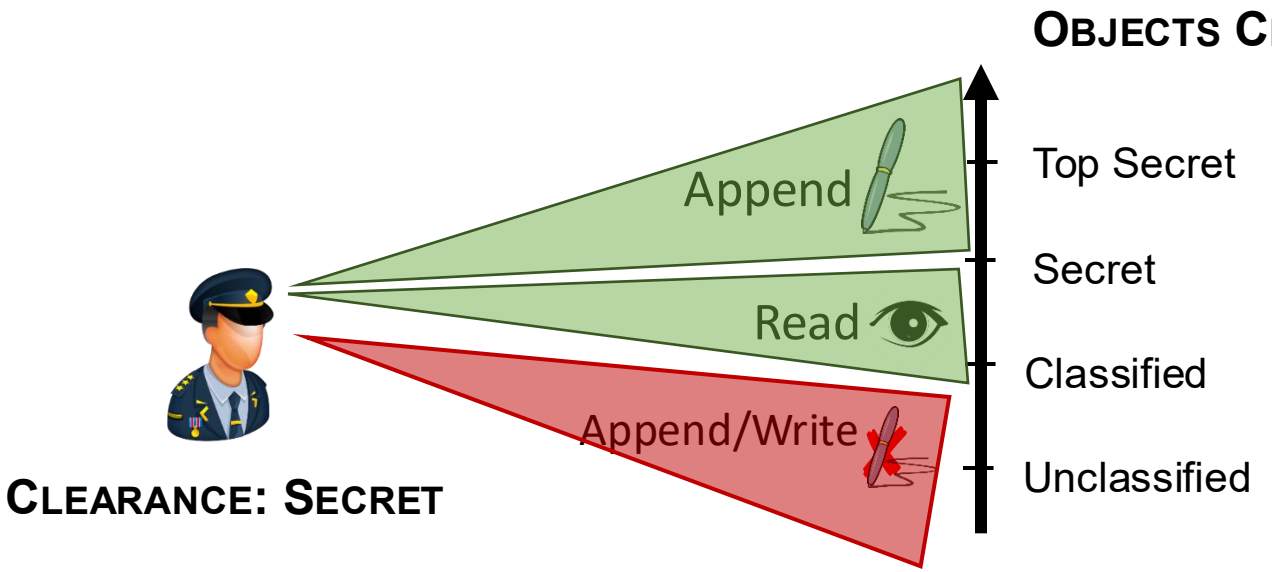
OBJECTS CLASSIFICATION

- Top Secret
- Secret
- Classified
- Unclassified

BLP System: *-property

STAR PROPERTY (*-PROPERTY)

if a subject has simultaneous "observe" (r,w) access to O_1
and "alter" (a,w) access to O_2 then level (O_2) dominates level (O_1)



No Write Down (NWD)

BLP System: ds-property

DISCRETIONARY PROPERTY (DS-PROPERTY)

if an access (subject, object, action) takes place it must be in the access control matrix

Information should only be accessed on a “need-to-know” basis

MAC needs **DAC** (least privilege inside the Security Level)

Also important to protect integrity

BLP: Basic Security Theorem

BASIC SECURITY THEOREM

if all state transitions are secure, and the initial state is secure, then every subsequent state is secure regardless of the inputs

If for any individual access:

(1) the ss-property holds.

(2) the *-property holds.

(3) the ds-property holds.

… then for any sequential composition security holds!

A system can be analyzed in terms of single step transitions of states!!

But… these properties are not enough

Assume

- level(s_1) is TS (Top Secret)
- level(s_2) is C (Confidential)

Sequence of events

1) s_2 creates o_2 → level(o_2) = C

But… these properties are not enough

Assume

- level(s_1) is TS (Top Secret)
- level(s_2) is C (Confidential)

Sequence of events

1) s_2 creates o_2 → level(o_2) = C
2) s_1 reads C and either:
    - changes the object level → level(o_2) = TS
    - leaves object level untouched → level(o_2) = C

But… these properties are not enough

Assume

- level(s_1) is TS (Top Secret)
- level(s_2) is C (Confidential)

Sequence of events

1) s_2 creates o_2 → level(o_2) = C
2) s_1 reads C and either:
    - changes the object level → level(o_2) = TS
    - leaves object level untouched → level(o_2) = C
3) s_2 attempts to access to o_2 in C → success or failure leaks 1 bit of information!

Covert channels

COVERT CHANNEL

any channel that allows information flows contrary to the security policy

Storage channels

e.g. shared counters, ID fields, file meta-data, etc.

Timing channels

e.g. use of CPU, load to memory (cache), queuing time, etc.

Covert channels

COVERT CHANNEL

any channel that allows information flows contrary to the security policy

Storage channels

e.g. shared counters, ID fields, file meta-data, etc.

Timing channels

e.g. use of CPU, load to memory (cache), queuing time, etc.

Principle 7

Least common mechanism

The more resources are shared, the harder it is to eliminate covert channels

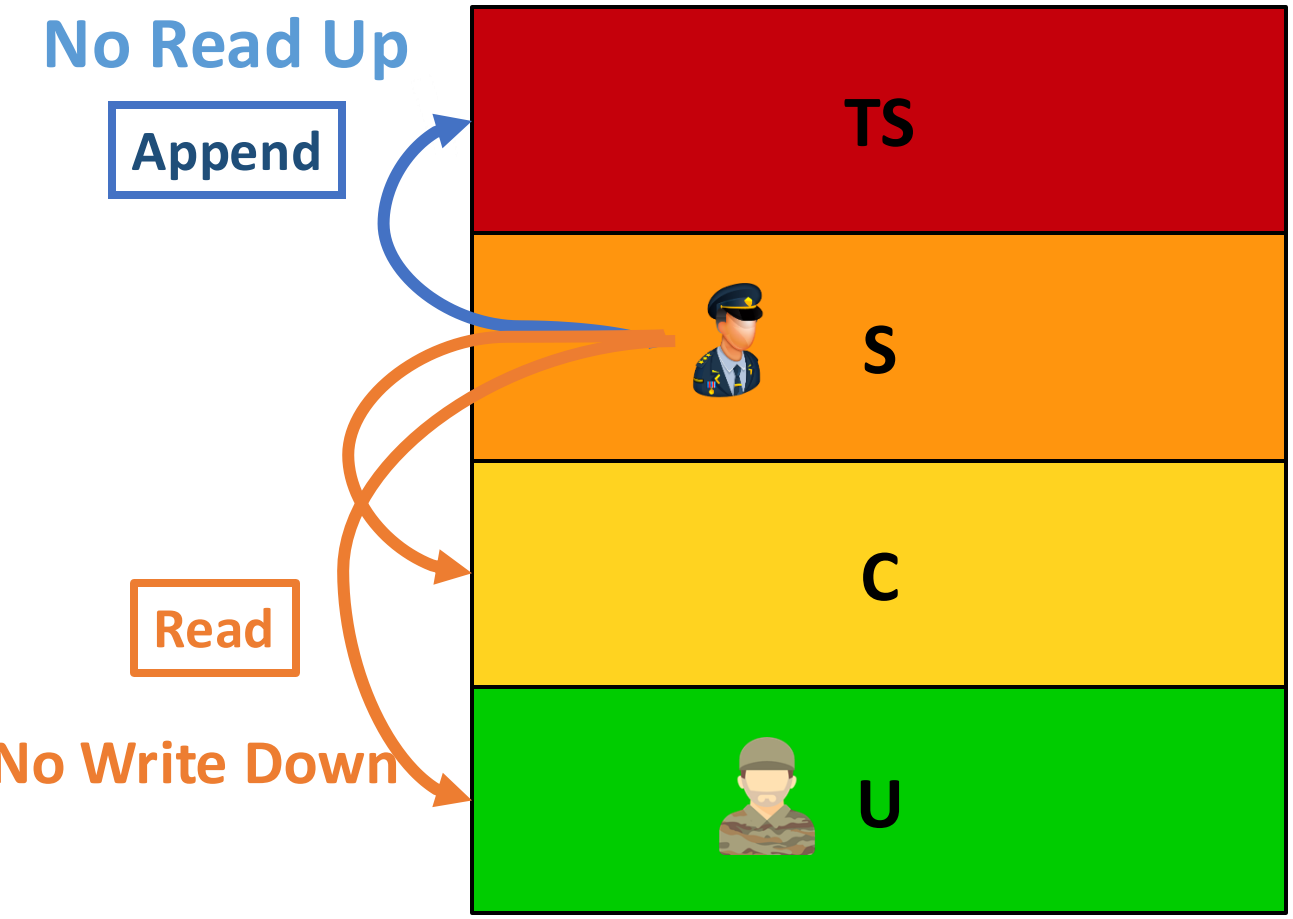
Mitigating Covert channels

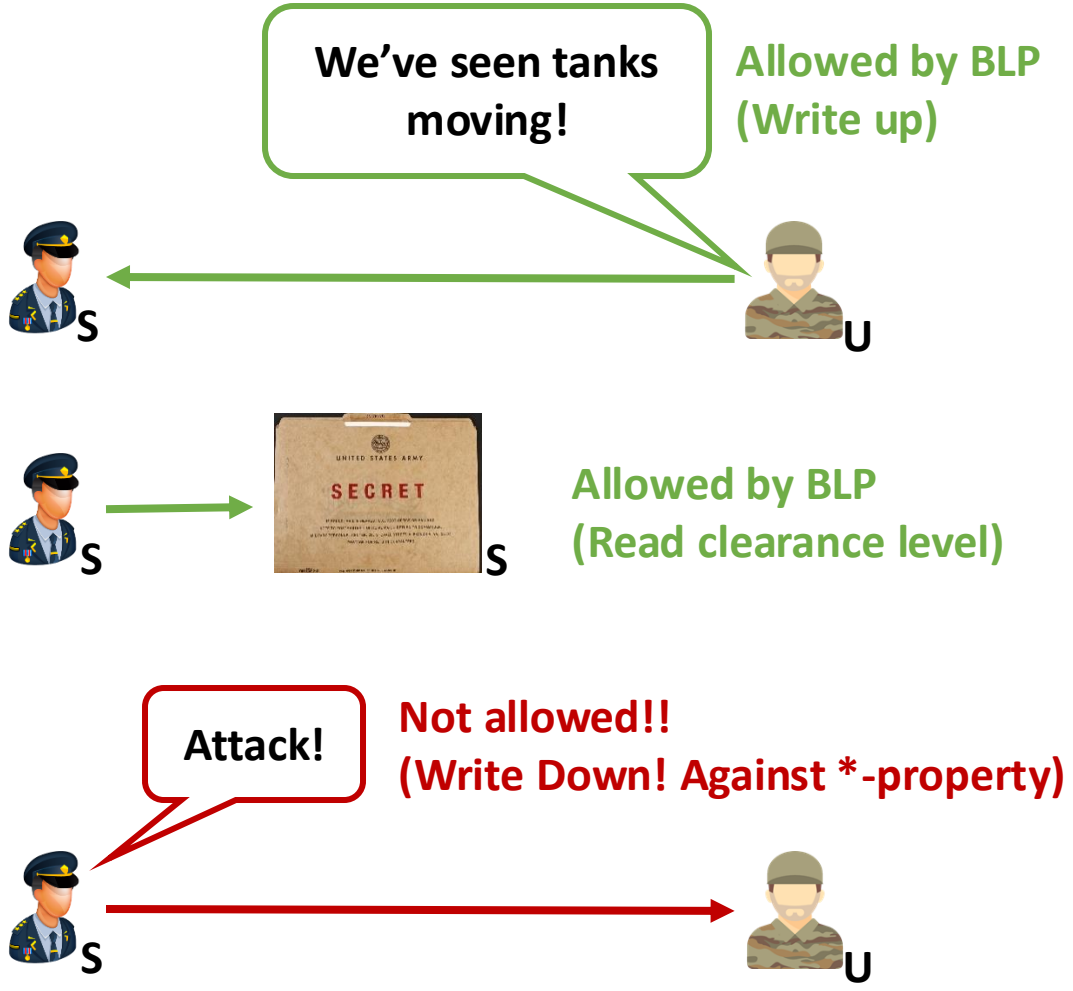
COVERT CHANNEL

any channel that allows information flows contrary to the security policy

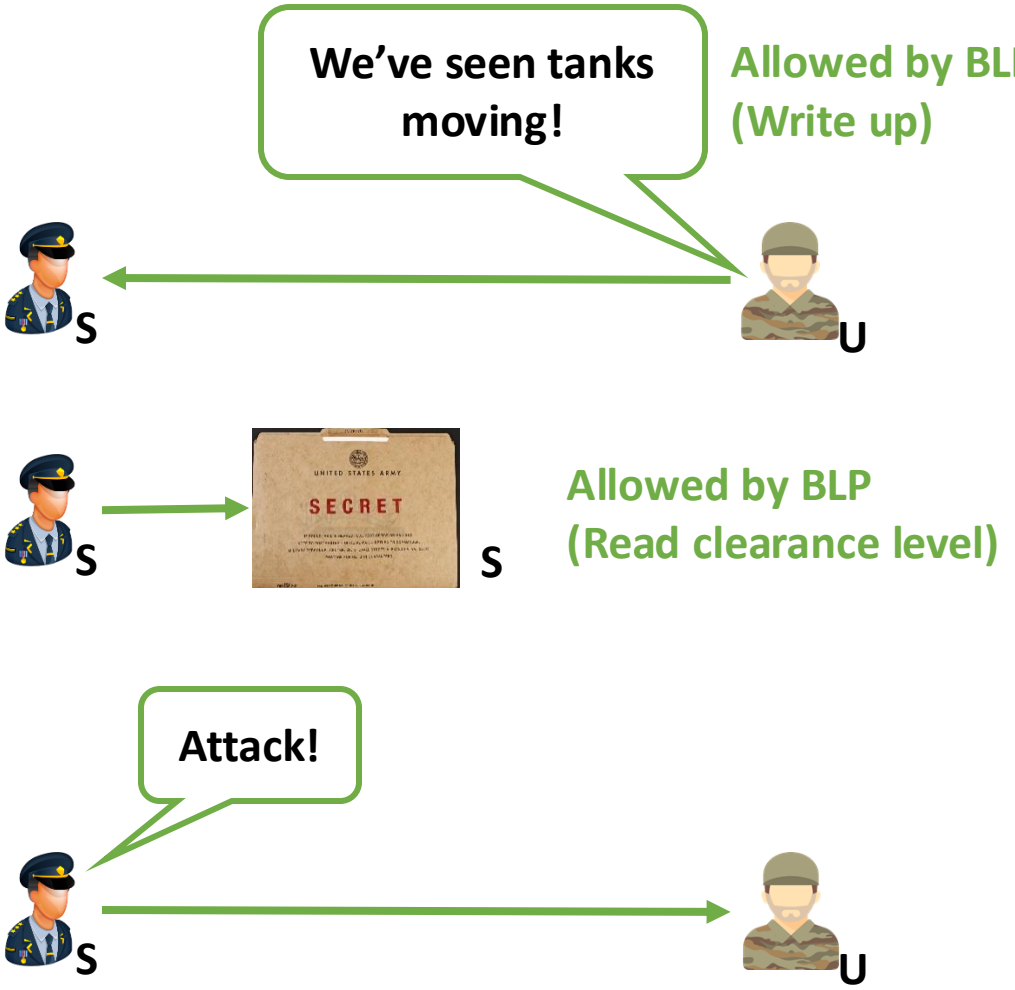
Mitigation: isolation (communication with low level not possible) or add of noise to communication.

- Hard to achieve less than 1 bit / sec
- OK for documents, **NOT** OK for cryptographic keys
    - DoD policy: cryptographic keys must always be stored on dedicated hardware.


No Read Up
Append
TS
S
C
U
Read
No Write Down


We've seen tanks moving!
Allowed by BLP (Write up)
S
U
S
UNITED STATES ARMY
SECRET
S
Allowed by BLP (Read clearance level)
Attack!
Not allowed!! (Write Down! Against *-property)
S
U

Declassification



DECLASSIFICATION

remove classification label

It is very typical and necessary

Under the control of the security policy.

- It **cannot** be made inherently safe

(*manual process*)

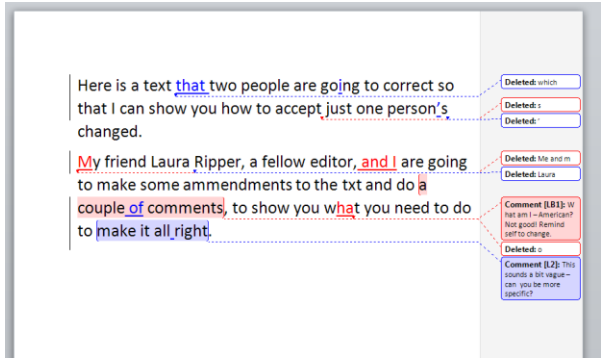
-Rules about archives, historical records

Hard to rule out covert channels.

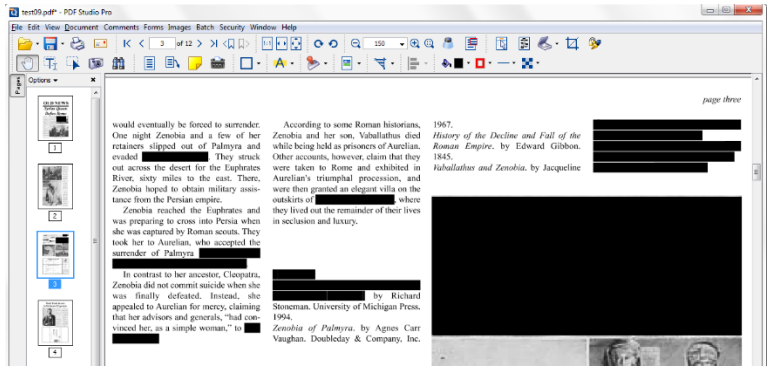
How to know the object does not contain secrets?

Difficulty of Declassification in practice

- Microsoft Word revision history retains deleted text



- Portable Document Format (PDF) redaction by overlaying graphical elements (usually black rectangles) → the text is on the file!



Strategic adversary!

https://www.slideshare.net/ange4771/pdf-secrets

Hill, S., Zhou, Z., Saul, L., & Shacham, H. On the (in) effectiveness of mosaicing and blurring as tools for document redaction. *Proceedings on Privacy Enhancing Technologies*, *2016*.

Bell La Padula: limitations

- Confidentiality-oriented: does not consider integrity or availability
- State-based + single transition model: too low-level, not expressive
- The 3 security properties are not sufficient to ensure confidentiality…
  - Changes in clearance and classification can create covert channels
  - A static system without changes is impractical

Computer Security (COM-301)

Mandatory Access Control
Integrity Security models

Carmela Troncoso

SPRING Lab

carmela.troncoso@epfl.ch

Some slides/ideas adapted from: George Danezis

Protecting integrity

Bell-La Padula focuses on **confidentiality**. Relevant for military / government environments

What about **commercial services?**

Banking, Stock and sales inventory, stock exchange, land registry, student grades database, electronic contracts, payments, …

Preventing fraud is about **protecting integrity**: the adversary has not influenced the result

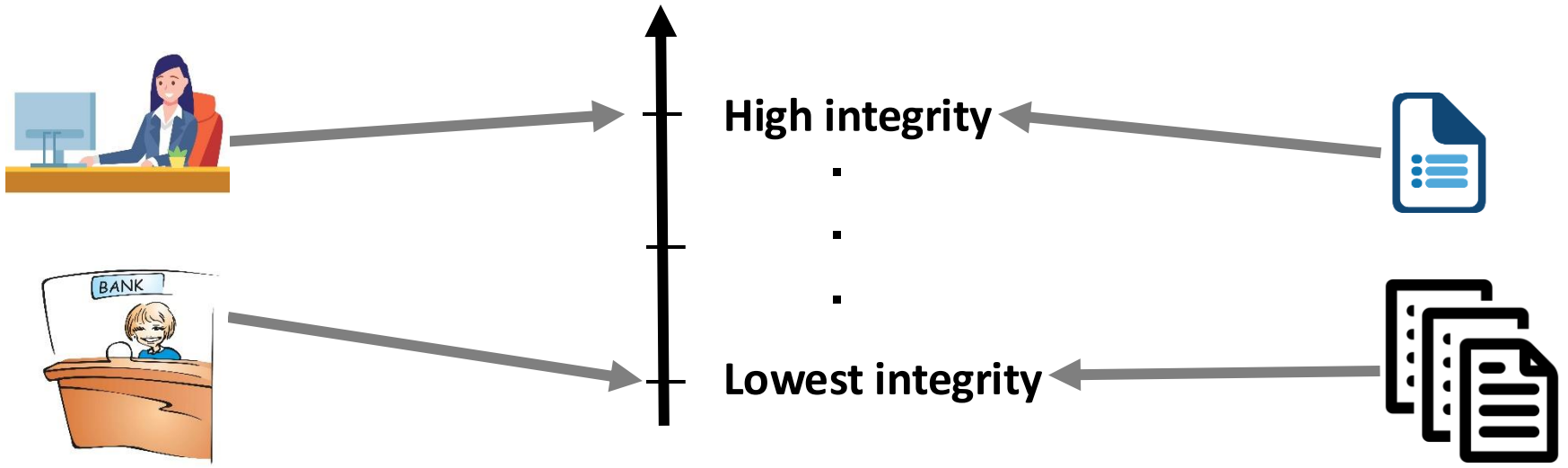
Confidentiality is either *secondary* or *unnecessary*.

Integrity is key for computer security in general!!

TCB has to have high integrity.

Public key cryptography requires high-integrity for confidentiality

The BIBA model for integrity



Two operations:

"read" (observe)

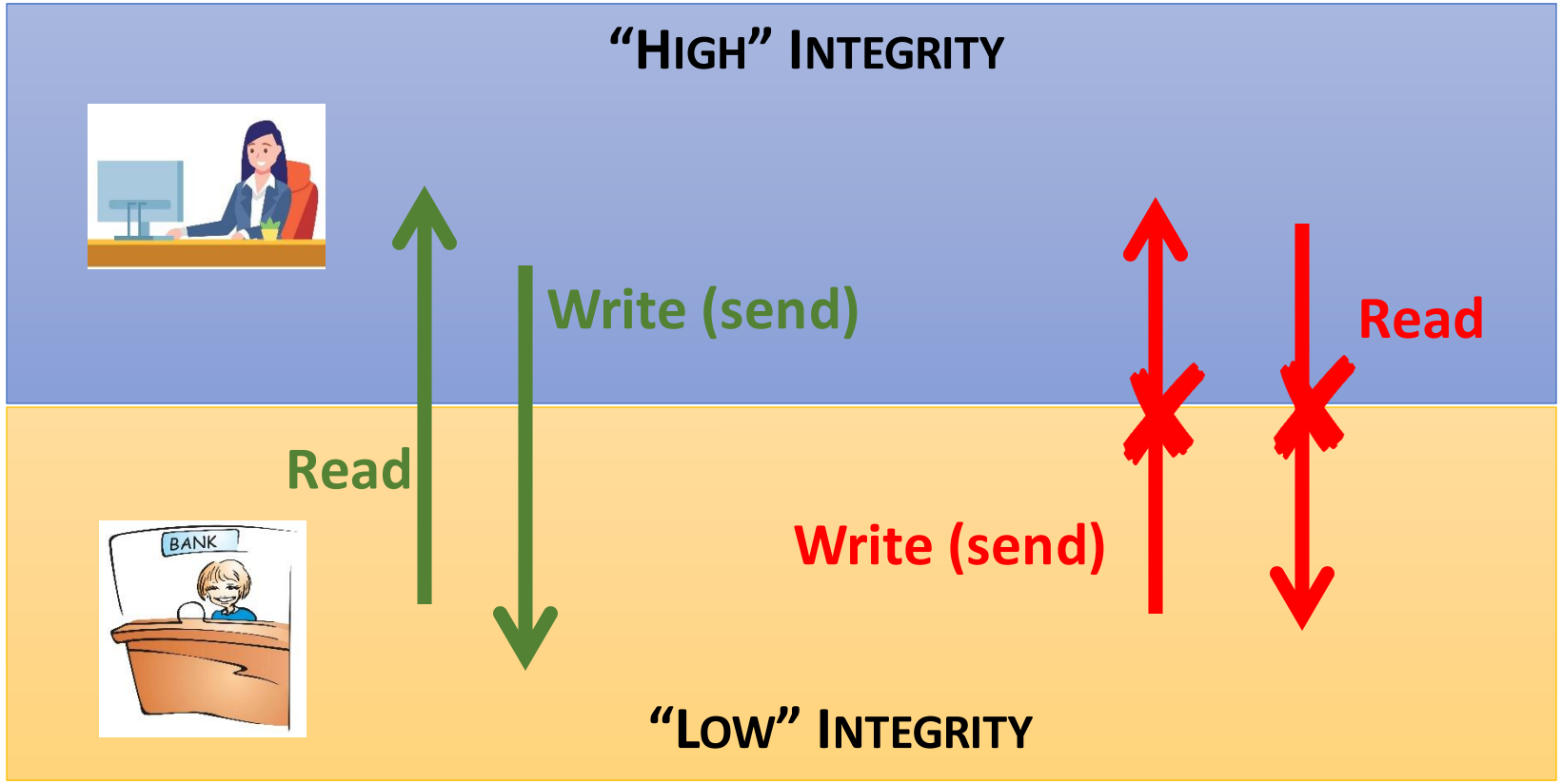
"write" (modify)

Two key rules (strict)

- ***simple integrity (no read-down)***: protects higher integrity principals from being corrupted by lower integrity level data

- ****-integrity (no write-up)***: prevents lower integrity principals from corrupting high integrity data

BIBA illustrated



EXAMPLES

In the Bank:
Director can establish a rule and every employee reads. Employees cannot rewrite rules

In the computer:
Web application open in the browser should not write to the file system (at most /tmp)

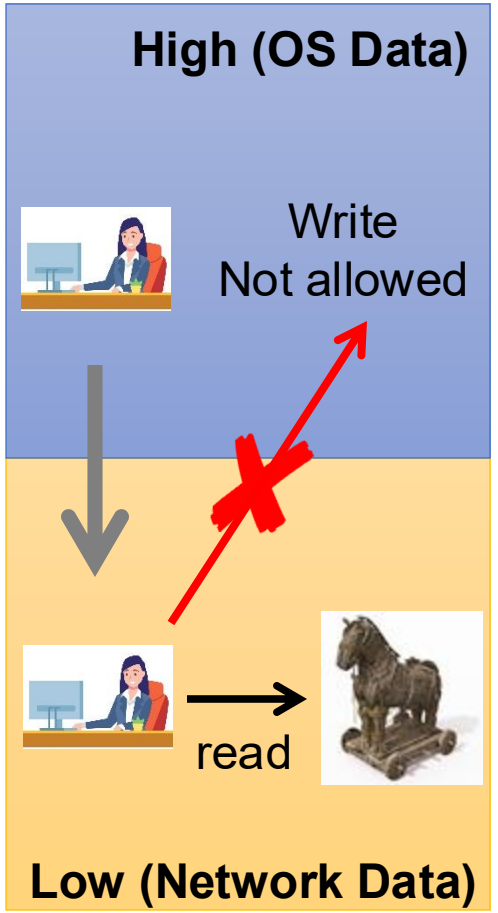
BIBA variant 1: Low-water-mark for subjects

Low-water-mark policy for subjects

- Subjects start processes at their highest integrity level.
- When accessing an object, its current level is lowered to the lowest of the two: current-level(s) and level(o)

Temporary downgrade for the session

- Example: mitigate impact of a network Trojan
- Hard to avoid label creep



BIBA variant 2: Low-water-mark for objects

Low-water-mark policy for objects

- Once an object has been written to by a subject, it assumed the lowest level of the object or subject.

A high-integrity database written to by a process with access to the network (low integrity) is labelled at "low" integrity

What is the effect?

Dangerous! only allows for integrity violation detection

Mitigation: replicate objects & sanitize / erase

BIBA Additional actions: Invoke

Simple Invocation

Only allow subjects to invoke subjects with a label they dominate

+ protect high integrity data from misuse by low integrity principals

- what level is the output?

Controlled Invocation

Only allow subjects to invoke subjects that dominate them

+ prevents corruption of high integrity data

- hard to detect polluting information

Sanitization

SANITIZATION

Process of taking objects with “low” integrity and “lifting them” to “high integrity”

“Sanitization” problems are the root cause of large classes of real-world security vulnerabilities

Malformed “low” (user) input can influence “high” (service) data and code

EXAMPLES

Web security: web server (high) accepts input from web client (low)
→ SQL interpreter → SQL injection vulnerability

OS Security: UNIX suid program (high) accepts input from a user (low)
→ short buffer → buffer overflow

Fundamental principle of sanitization

PRINCIPLE 2: FAIL-SAFE DEFAULT

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

Positively verify that "low" objects are within a valid set before elevating their integrity to "high".

- White list: check that all properties of good objects hold.
- Do not blacklist: do not just check for bad objects or properties.

Insert a photo in a web album? Ensure caption is from a restricted set of Unicode, or apply to it a transform to "escape" / "encode" any characters not from that safe set into it. Do not simply check it does not contain "<script>". (XSS Attack)

COVERT CHANNELS DIFFICULT TO CATCH!

Director, here is this object

Is it in the Universe of good things? Accept!

Universe of good things

Do this ...

... not that!

Is it in the set of bad things? Reject!

Bad things ... (large, undefined set)

Principles to support integrity

Three principles to guide your choices:

- **Separation of duties**: Require multiple principals to perform an operation

(harder for an adversary to tamper with the system as they have to corrupt two principals)

- **Rotation of duties**: Allow a principal only a limited time on any particular role and limit other actions while in this role

(harder for an adversarial insider to tamper with the system)

- **Secure Logging**: Tamper evident log to recover from integrity failures. Consistency of log across multiple entities is key.

(harder to make an integrity breach durable)

Chinese Wall model

Inspiration: UK rules about handling “conflicts of interest” in the financial sector.

- A separation must exist at all times, even within the same firm, between people engaging in activities that conflict with each other.
- Cost of failure: large fines and reputation

David FC. Brewer and Michael J. Nash. “The Chinese Wall Security Policy.” in IEEE SSP 1989

Chinese Wall model: Entities and Basic Concepts

1. All objects are associated with a label denoting their origin

   "Pepsi Ltd.", "Coca-Cola Co.", "Microsoft Audit", "Microsoft Investments"

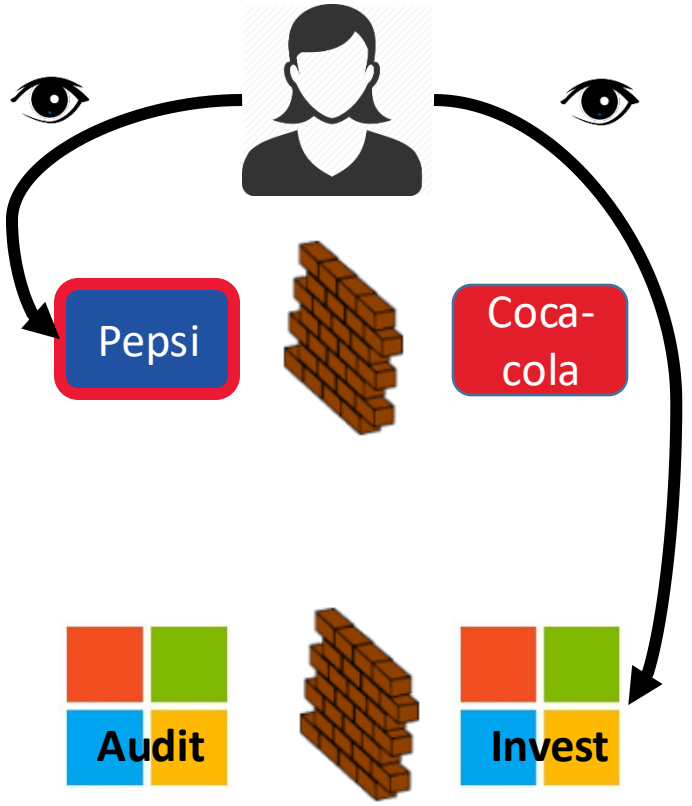
2. The originators define "conflict sets" of labels

   {"Pepsi Ltd.", "Coca-Cola Co."}, {"Microsoft Audit", "Microsoft Investments"}

3. Subjects are associated with a history of their accesses to objects, and in particular their labels.

ALICE

1. *Pepsi*
2. *Microsoft Invest*

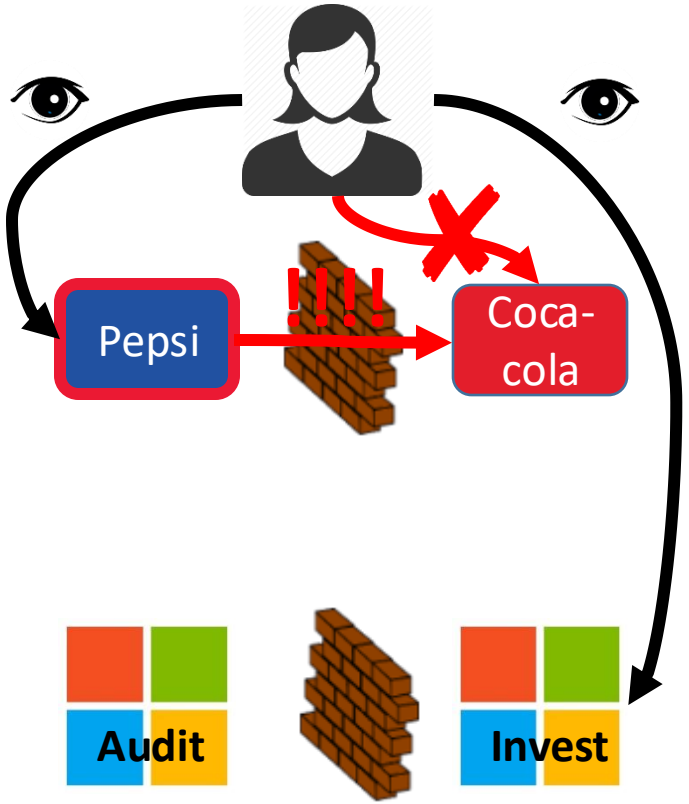


Chinese Wall model: Access rules

A subject can read an object (for either read or write) if the access ***does not allow an information flow*** between items with labels in the same conflict set

Alice starts her first day at work

1) She accesses files of “Pepsi Ltd” (OK)
2) She accesses files of “Microsoft invest” (OK)
3) She tries to access files of “Coca-cola Co.” (access denied!)

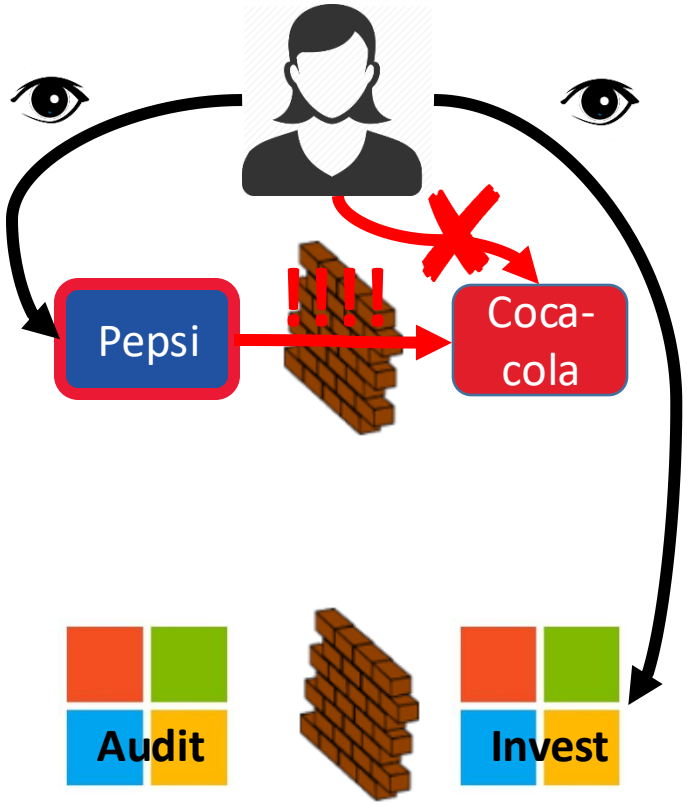


Chinese Wall model: Access rules

A subject can read an object (for either read or write) if the access ***does not allow an information flow*** between items with labels in the same conflict set

Alice starts her first day at work

1) She accesses files of “Pepsi Ltd” (OK)
2) She accesses files of “Microsoft invest” (OK)
3) She tries to access files of “Coca-cola Co.” (access denied!)



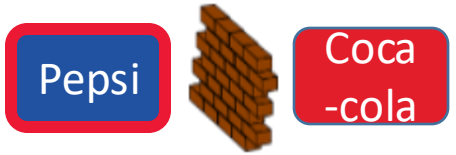
Why? She has already accessed files from “Pepsi Ltd” thus an information flow between those and “Coca-cola Co” might happen (She could work again with “Pepsi”)

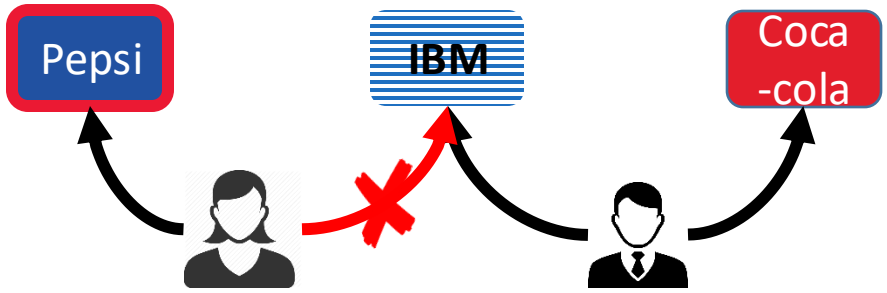
Chinese Wall model: Indirect flows

Direct flow within a conflict set is easy to detect! What about indirect?

Alice and Bob start together

1) Alice is assigned to “Pepsi Ltd” (OK)
2) Bob is assigned to “Coca-cola Co.” and “IBM Co.” (OK)
3) Alice tries to access files of “IBM Co.” (access denied!)



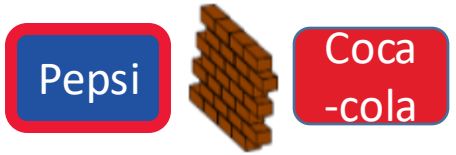


Chinese Wall model: Indirect flows

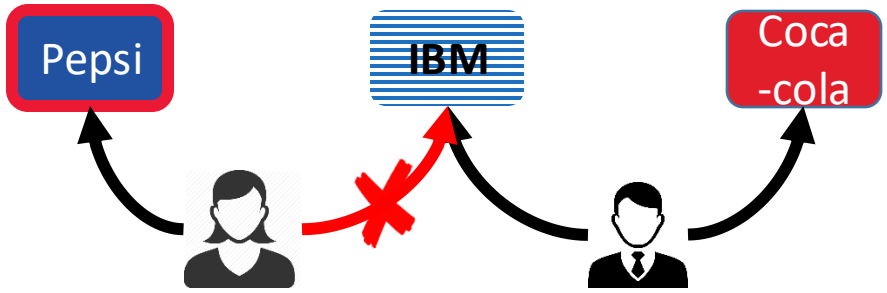
Direct flow within a conflict set is easy to detect! What about indirect?

Alice and Bob start together

1) Alice is assigned to “Pepsi Ltd” (OK)
2) Bob is assigned to “Coca-cola Co.” and “IBM Co.” (OK)
3) Alice tries to access files of “IBM Co.” (access denied!)



Why? If she writes in IBM with her knowledge of Pepsi, then the information ***may*** flow to Coca-cola.

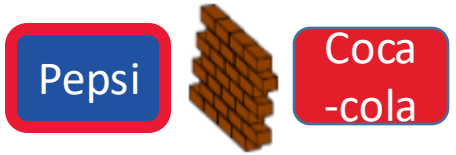


Chinese Wall model: Indirect flows

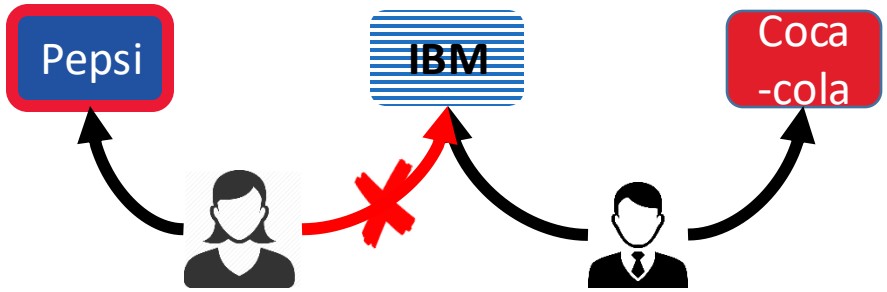
Direct flow within a conflict set is easy to detect! What about indirect?

Alice and Bob start together

1) Alice is assigned to “Pepsi Ltd” (OK)
2) Bob is assigned to “Coca-cola Co.” and “IBM Co.” (OK)
3) Alice tries to access files of “IBM Co.” (access denied!)



Why? If she writes in IBM with her knowledge of Pepsi, then the information ***may*** flow to Coca-cola.



SANITIZATION is necessary for business

“Un-label” some items as long as the information cannot lead to any conflict of interest, e.g., extract some “general market information”

Summary of the lecture

- **Security models**: patterns to design MAC policies

- **BLP**: Confidentiality
  - Key concept: Declassification
- **BIBA**: Integrity
  - Can bootstrap: high confidentiality (PKI) or High availability (replication)
  - Can lead to: low confidentiality or low availability
  - Key concept: Sanitization
- **Chinese Wall:** Conflicts of interest (confidentiality & integrity)
- **Multilateral security:** conflicting properties