Facebook Security Breach Exposes Accounts of 50 Million Users

One of the challenges for Facebook's chief executive Mark Zuckerberg is convincing users that the company handles their data responsibly. Josh Edelson/Agence France-Presse — Getty Images

By Mike Isaac and Sheera Frenkel

Sept. 28, 2018

288

SAN FRANCISCO — Facebook, already facing scrutiny over how it handles the private information of its users, said on Friday that an attack on its computer network had exposed the personal information of nearly 50 million users.

Huge Facebook breach leaves thousands of other apps vulnerable

The breach affecting 50m accounts took advantage of 'tokens', a system used by third-party platforms such as Spotify

most viewed

Three key Republicans condemn Trump for mocking Christine Blasey Ford

Stephen Hawking's first wife intensifies attack on The Theory of Everything

Facebook Security Breach Exposes Accounts of 50 Million Users

One of the challenges for Facebook's chief executive Mark Zuckerberg is convincing users that the company handles their data responsibly. Josh Edelson/Agence France-Presse — Getty Images

By Mike Isaac and Sheera Frenkel

Sept. 28, 2018

288

SAN FRANCISCO — Facebook, already facing scrutiny over how it handles the private information of its users, said on Friday that an attack on its computer network had exposed the personal information of nearly 50 million users.

job Sign in Search International edition

Sport Culture Lifestyle More

The Guardian

al development Football **Tech** Business Environment Obituaries

Huge Facebook breach leaves thousands of other apps vulnerable

The breach affecting 50m accounts took advantage of 'tokens', a system used by third-party platforms such as Spotify

most viewed

Three key Republicans condemn Trump for mocking Christine Blasey Ford

Stephen Hawking's first wife intensifies attack on The Theory of Everything

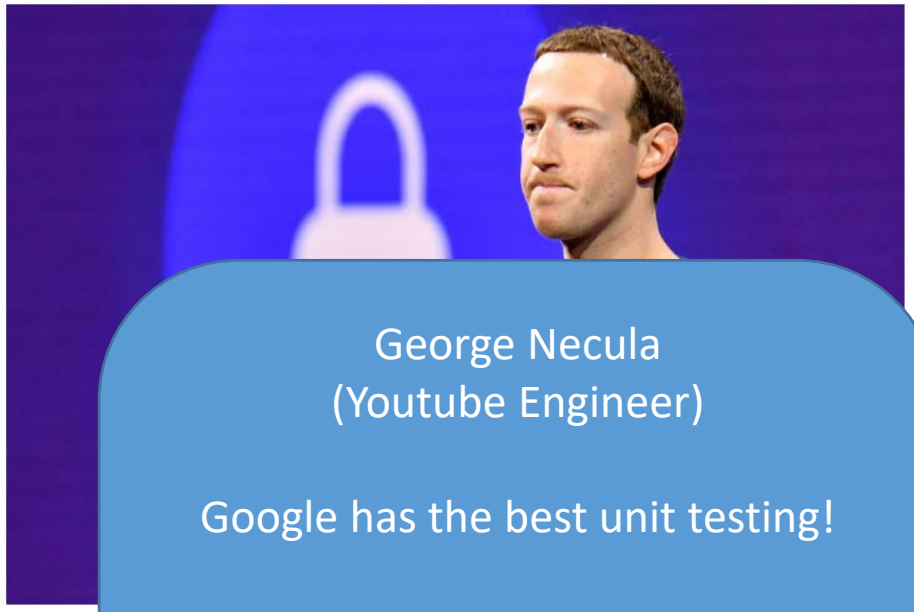
~~Least~~ Most common mechanism…

Complete mediation

Time to check vs. time to use

Checking bugs cannot **prove** there are no bugs

Facebook Security Breach Exposes Accounts of 50 Million Users



One of the
company h

By M

Sept

SA
har
on i
50 m

George Necula
(Youtube Engineer)

Google has the best unit testing!

Hard to do more complex testing!!
(not that much bigger test)

2 billion lines of code
40k commits, 15Mlines in 50K files

Reuse code -> Lots dependencies
(thousands per file)

~~Least~~ Most common mechanism…

Complete mediation

Time to check vs. time to use

Checking bugs cannot **prove** there are no bugs

Last week

ACCESS CONTROL

Security mechanism that ensures that

"all accesses and actions on system objects by principals are WITHIN the security policy"

Operating System

- Objects: files, devices, OS operations, ...
- Subjects: principals are processes, pipes, ...

Middleware

- Objects: tables, records, rows, columns, …
- Subjects: DB specific, e.g. stored in USERS table

Hardware

- Objects: Memory pages, privileged instructions
- Subjects: processor mode, protection domains

Applications

- Objects: Photos, posts, messages
- Subjects: users, groups

Mixing domains is meaningless!!

OS access control cannot restrict access to a certain row of a Database.

but they build on top of each other:

OS access control required to restrict access to the *whole* DB file.

Last week

ACCESS CONTROL
Security mechanism that ensures that
"all accesses and actions on system objects by principals are WITHIN the security policy"

Operating System

- Objects: files, devices, OS operations, ...
- Subjects: principals are processes, pipes, ...

George Necula
Software Engineer at YouTube
GDPR -> Access per row/table…

Hardware

- Objects: Memory pages, privileged instructions
- Subjects: processor mode, protection domains

Applications

- Objects: Photos, posts, messages
- Subjects: users, groups

Mixing domains is meaningless!!

OS access control cannot restrict access to a certain row of a Database.

but they build on top of each other:

OS access control required to restrict access to the *whole* DB file.

Last week

ACCESS CONTROL

Security mechanism that ensures that

"all accesses and actions on system objects by principals are WITHIN the security policy"

Operating System

- Objects: files
- Subjects: pri

Middleware

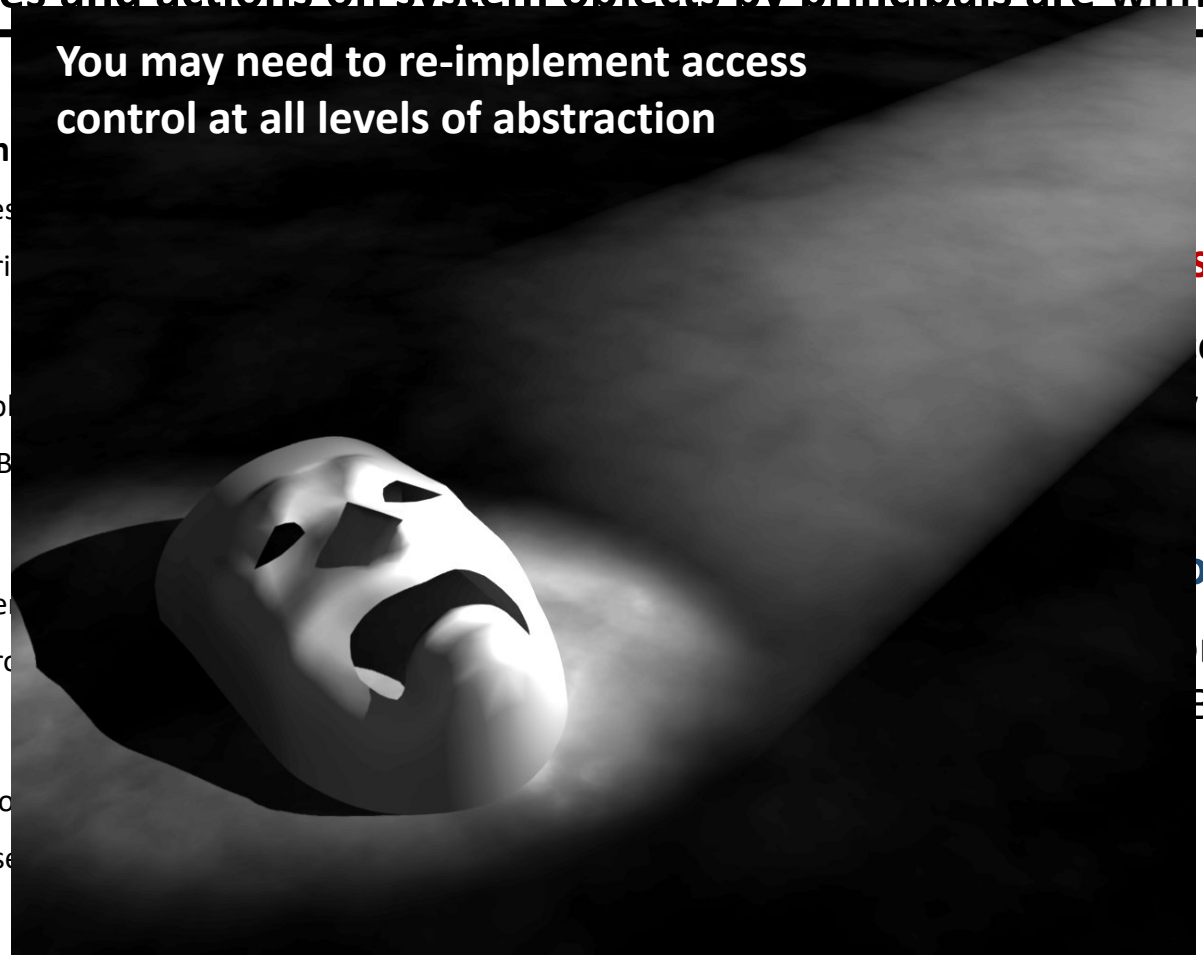
- Objects: tabl
- Subjects: DB

Hardware

- Objects: Me
- Subjects: pro

Applications

- Objects: Pho
- Subjects: use



s is meaningless!!

ontrol cannot restrict access to a
of a Database.

n top of each other:

ntrol required to restrict access to
B file.

Last week

ACCESS CONTROL MATRIX

Abstract representation that characterizes the rights of each subject with respect to every object in a system

	file1	file2	file3
Alice	read write		read
Bob		read write	read write

Last week

ACCESS CONTROL MATRIX

Abstract representation that characterizes the rights of each subject with respect to every object in a system

	file1	file2	file3
Alice	read write		read
Bob		read write	read write

Cannot be implemented directly!!

Last week

ACCESS CONTROL MATRIX

Abstract representation that characterizes the rights of each subject with respect to every object in a system

	file1	file2	file3
Alice	read write		read
Bob		read write	read write

Cannot be implemented directly!!

ACCESS CONTROL LISTS (BY COLUMN): permission associated to **objects**

```
file1: {(Alice,read/write)}
file2: {(Bob, read/write)}
file3: {(Alice,read), (Bob,read/write)}
```

CAPABILITIES (BY ROW): permissions associated to **subjects**

```
Alice: {(file1,read/write), (file3,read)}
Bob: {(file2,read/write), (file3,write)}
```

Last week

ACCESS CONTROL MATRIX

Abstract representation that characterizes the rights of each subject with respect to every object in a system

	file1	file2	file3
Alice	read write		read
Bob		read write	read write

Cannot be implemented directly!!

ACCESS CONTROL LISTS (BY COLUMN): permission associated to **objects**

```
file1: {(Alice,read/write)}
file2: {(Bob, read/write)}
file3: {(Alice,read), (Bob,read/write)}
```

CAPABILITIES (BY ROW): permissions associated to **subjects**

```
Alice: {(file1,read/write), (file3,read)}
Bob: {(file2,read/write), (file3,write)}
```

AMBIENT AUTHORITY

CALLING A FUNCTION WITHOUT SUBJECT

ACL → USES EXECUTING USER PERMISSIONS

CAPABILITIES → USES CAPABILITIES USER'S PERMISSION

Last week

ACCESS CONTROL MATRIX

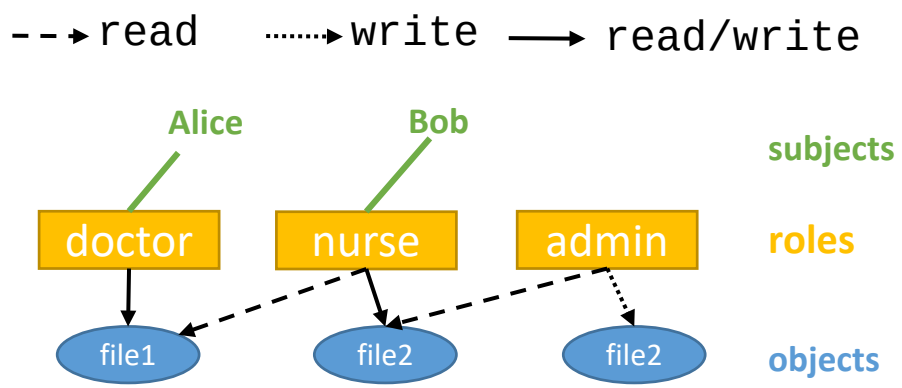
Abstract representation that characterizes the rights of each subject with respect to every object in a system

	file1	file2	file3
Alice			
Bob			

Cannot be implemented directly!!

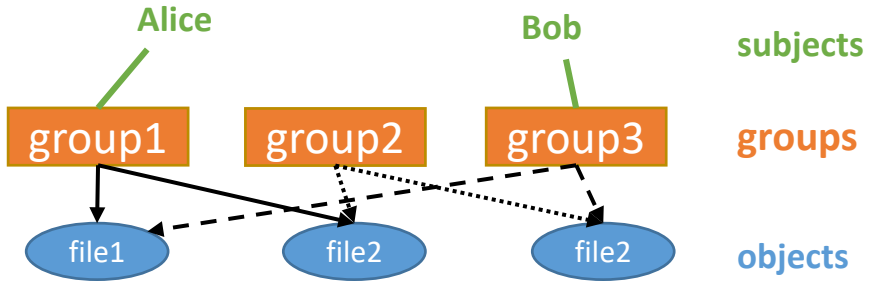
ROLE BASED ACCESS CONTROL: group **subjects** that have similar requirements

```
doctors:    {(file1,read/write)}
nurses:     {(file1,read), (file2,read/write)}
admin:      {(file2,read), (file3,write)}
```



GROUP BASED ACCESS CONTROL: group **objects** that have similar requirements

```
group1:     (read/write){file1,file2}
group2:     (write){file2,file3}
group3:     (read){file1,file3}
```



Last week

ACCESS CONTROL MATRIX

Abstract representation that characterizes the rights of each subject with respect to every object in a system

	file1	file2	file3
Alice			
Bob			

Cannot be implemented directly!!

ROLE BASED ACCESS CONTROL: group **subjects** that have similar requirements

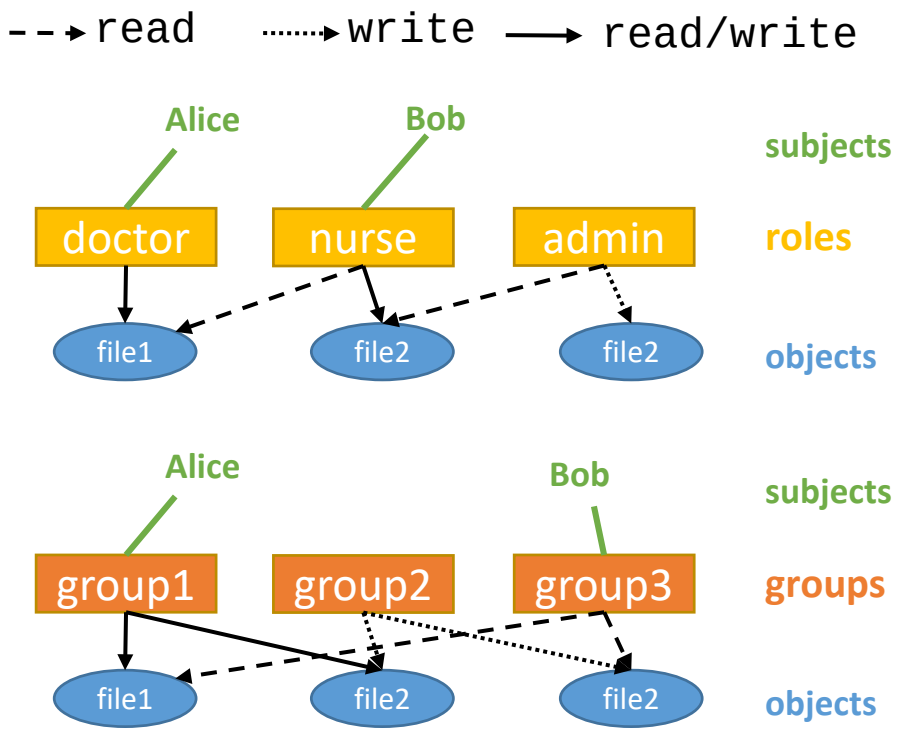
```
doctors:    {(file1,read/write)}
nurses:     {(file1,read), (file2,read/write)}
admin:      {(file2,read), (file3,write)}
```

Negative permissions

```
Alice: not (read,file1)
```

GROUP BASED ACCESS CONTROL: group **objects** that have similar requirements

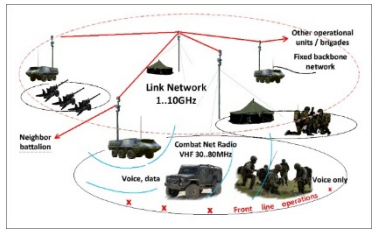
```
group1:     (read/write){file1,file2}
group2:     (write){file2,file3}
group3:     (read){file1,file3}
```



Last week: who sets the policy?

MANDATORY ACCESS CONTROL (MAC)

Central security policy *assigns permissions*



DISCRETIONARY ACCESS CONTROL (DAC)

Object owners *assign permissions*

STRAVA

MAC
OR
DAC

Are there things that “creators” or “owners” of resources are not allowed to do? (by policy not mechanism)

YES! it is a MAC

NO! it is a DAC

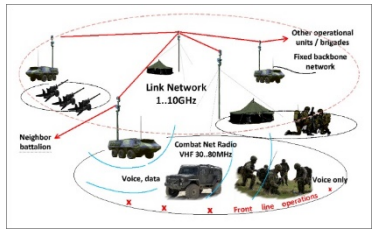
Last week: who sets the policy?

ACL

```
-rwxrwxrw- 1 catronco catronco 8600 Sep 15 15:20 hello
```

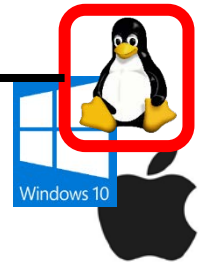
MANDATORY ACCESS CONTROL (MAC)

Central security policy *assigns permissions*



DISCRETIONARY ACCESS CONTROL (DAC)

Object owners *assign permissions*



STRAVA

Are there things that “creators” or “owners” of resources are not allowed to do?
(by policy not mechanism)

YES! it is a MAC

NO! it is a DAC

What about Windows?

Principals = users, machines, groups,…

Objects = files, Registry keys, printers, …

Access control:

Each object has a ***discretionary access control list*** (DACL)

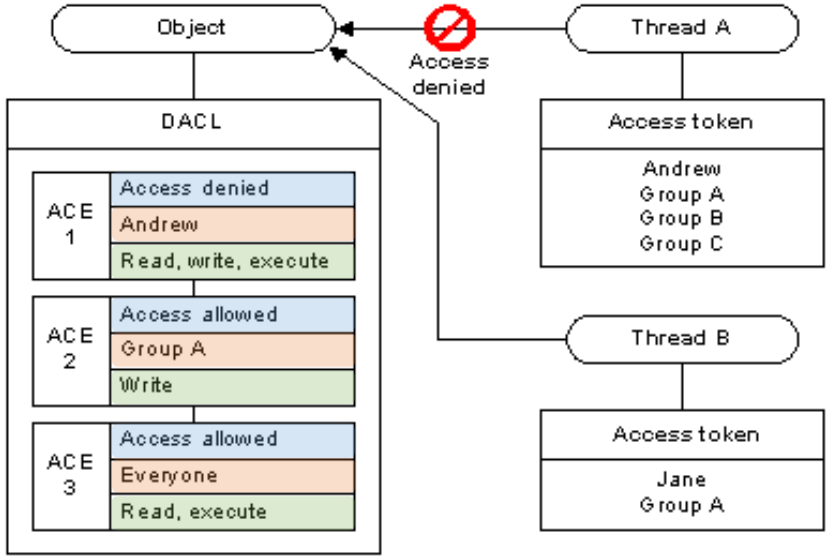
Each process (or thread) has an access token:

- Login user account (process "runs as" this user)
- All groups in which the user is a member(recursively!)
- All privileges assigned to these groups

Compare DACL with the process' access token when creating a handle to the object

What about Windows? DACL

List of Access Control Entries (ACEs)

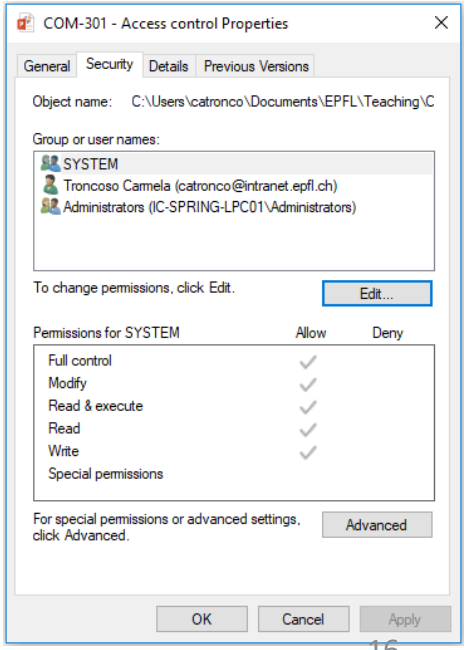


Type: negative / positive

Principal

Permissions: more fine grained than UNIX

+ Flags and others…



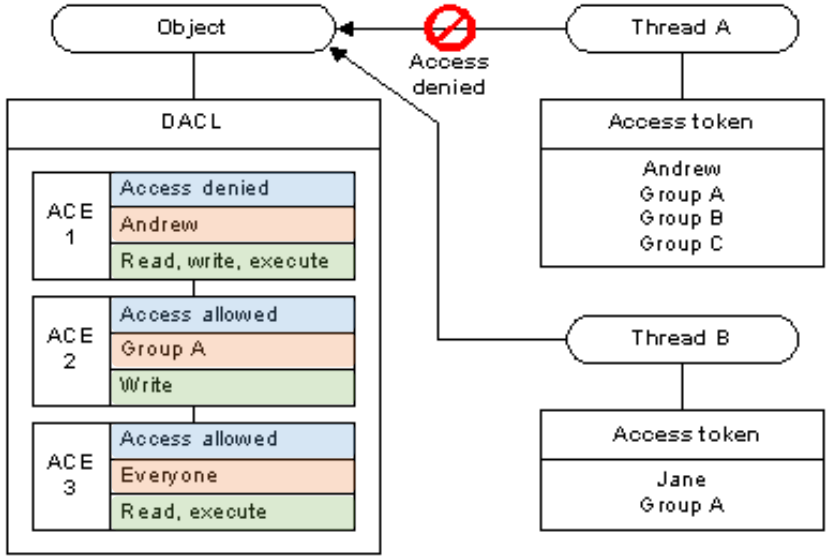
Least Privilege by default

Run as administrator

https://docs.microsoft.com/en-us/windows/desktop/secauthz/dacls-and-aces

What about Windows? DACL

List of Access Control Entries (ACEs)



Why negative first?

Type: negative / positive

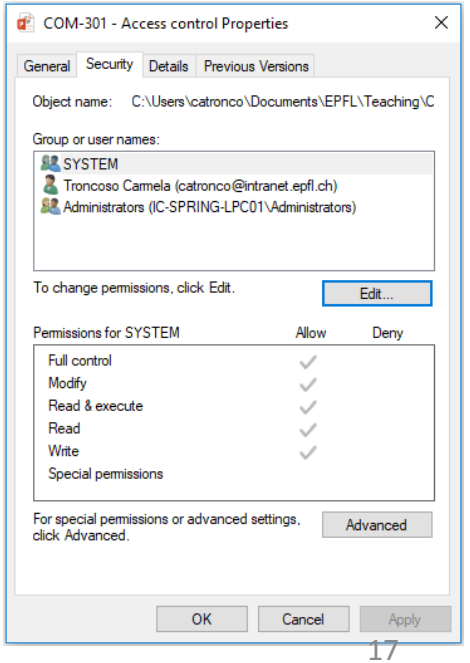
Principal

Permissions: more fine grained than UNIX

+ Flags and others…

Least Privilege by default

Run as administrator



https://docs.microsoft.com/en-us/windows/desktop/secauthz/dacls-and-aces

This week: 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 users trying to subvert it

How to build these policies?

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

SPRING
SECURITY AND PRIVACY ENGINEERING LABORATORY

Computer Security (COM-301)

Security models

Carmela Troncoso
SPRING Lab
carmela.troncoso@epfl.ch

Some slides/ideas adapted from: George Danezis

Before we start - Why this topic?

Reminder Course aim:

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

What are Security models?

A security model is a **design pattern** for a specific security property or set of properties

- Example of a “security model”
  - **ORANGE BOOK**: covers (mainly) needs of government for confidentiality.
    ↳ Trusted Computer System Evaluation Criteria

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

What are Security models?

A security model is a **design pattern** for a specific security property or set of properties

- Example of a “security model”
  - **ORANGE BOOK**: covers (mainly) needs of government for confidentiality.

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

The devil is in the details!

who are the subjects

what are the objects

what mechanisms to use to implement it?

Many aspects not covered by model!

PART I: Security models for confidentiality

References

Ross Anderson “Security Engineering”:

- Chapter 8 “Multilevel Security”

Dieter Gollmann “Computer Security”:

- Chapter 3 “Security models”
  - Bell-LaPadula Model

History

1950-1960s - Mainframe computer

- punch cards, paper tape, and/or magnetic tape
- **No interaction**, batch processes

IBM650 (1954)

1960s-1970s - Terminals connected to the mainframe

- Computers are deployed in Gov. and military.
- Computer Security is concerned with keeping **secrets**: confidentiality.

IBM 2260 (1964)

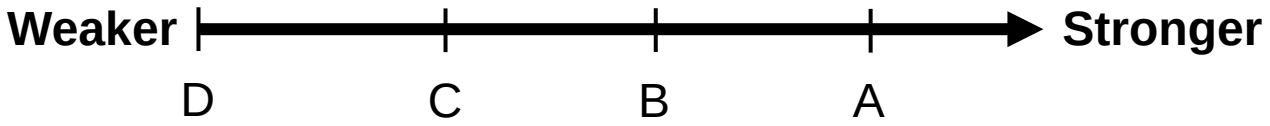
1983 and 1985 - "Orange Book"

- Trusted Computer System Evaluation Criteria (TCSEC) (US DoD).
- Assessing effectiveness of computer security. Requirements to process gov. secrets.
- Classes: D, C, B, A

1980s-1990s - Personal computers and the Internet!

A bit more on the Orange Book

Trusted Computer System Evaluation Criteria



D – Minimal Protection

Evaluated but failed!

C – Discretionary Protection

C1: Discretionary Security Protection

DAC, authentication, identification

C2: Controlled Access Protection

DAC + audit trail, accountability, resource isolation

B – Mandatory Protection

B1: Labelled Security Protection

*informal security policy, selective MAC, **labels***

B2: Structured Protection

*full security policy, full MAC and DAC coverage, **covert storage channels** mitigated*

B3: Security Domains

*robust reference monitor, minimal TCB, audit, IDS, timing **covert channels***

A – Verified Protection

A1: Verified design

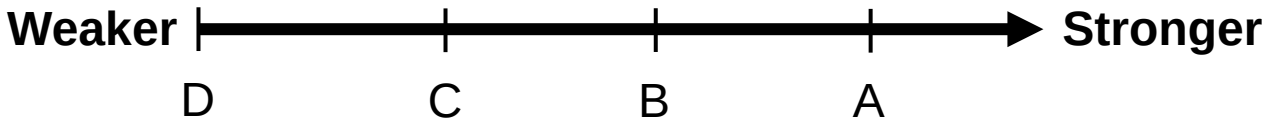
B3 + formal specification, design and verification

Beyond A1

strength of mechanism

A bit more on the Orange Book

Trusted Computer System Evaluation Criteria



D – Minimal Protection

Evaluated but failed!

C – Discretionary Protection

C1: Discretionary Security Protection

DAC, authentication, identification

C2: Controlled Access Protection

DAC + audit trail, accountability, resource isolation

Why is such a model unrealistic?

B – Mandatory Protection

B1: Labelled Security Protection

*informal security policy, selective MAC, **labels***

B2: Structured Protection

*full security policy, full MAC and DAC coverage, **covert storage channels** mitigated*

B3: Security Domains

*robust reference monitor, minimal TCB, audit, IDS, **timing covert channels***

A – Verified Protection

A1: Verified design

B3 + formal specification, design and verification

Beyond A1

strength of mechanism

The Bell-LaPadula Model (BLP)

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

(Section II: model description)

Goal: Enable one to show that a computer system can securely process classified information (**Confidentiality**)

BLP System Model

Subjects S_i in S

Objects O_i in O

Access Attributes { `e(xecute)`, `r(ead)`, `a(ppend)`, `w(rite)` }

Actions Semantics

	No Alteration	Alteration
No observation	`execute`	`append`
Observation	`read`	`write`

BLP System Model

Subjects S_i in S

Objects O_i in O

Access Attributes { e(xecute), r(ead), a(ppend), w(rite)

Actions Semantics

	No Alteration	Alteration
No observation	execute	append
Observation	read	write

System state

- **Current** access set: authorized (subject, object, action)
- **Level function** mapping subject/objects to classifications

  level(O_j) level(S_i) current-level(S_i)

-Access control matrix M

Multilevel Security MLS

Level function for objects: **Classification**

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

Categories – grouped in sets called compartments (*e.g., Nuclear, NATO, Crypto*)

(Classification, {set of categories})

DOMINANCE RELATIONSHIP

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

All objects O_j are assigned a security level called their classification: ***classification level(O_j)***

True or false?

DOMINANCE RELATIONSHIP

A level (l1, c1) “dominates” (l2, c2)

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

Labels: U < C < S < TS

Categories: SIGINT, CRYPTO, NOFORN

(TS, {}) dominates (C, {})

(S, {}) dominates (C, {NOFORN})

(C, {}) dominates (C, {SIGINT})

(S, {SIGINT, CRYPTO}) dominates (C, {CRYPTO})

(TS, {CRYPTO, SIGINT}) dominates (S, {NOFORN})

True or false?

DOMINANCE RELATIONSHIP

A level (l1, c1) “dominates” (l2, c2)

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

Labels: U < C < S < TS

Categories: SIGINT, CRYPTO, NOFORN

(TS, {}) dominates (C, {})

(S, {}) dominates (C, {NOFORN})

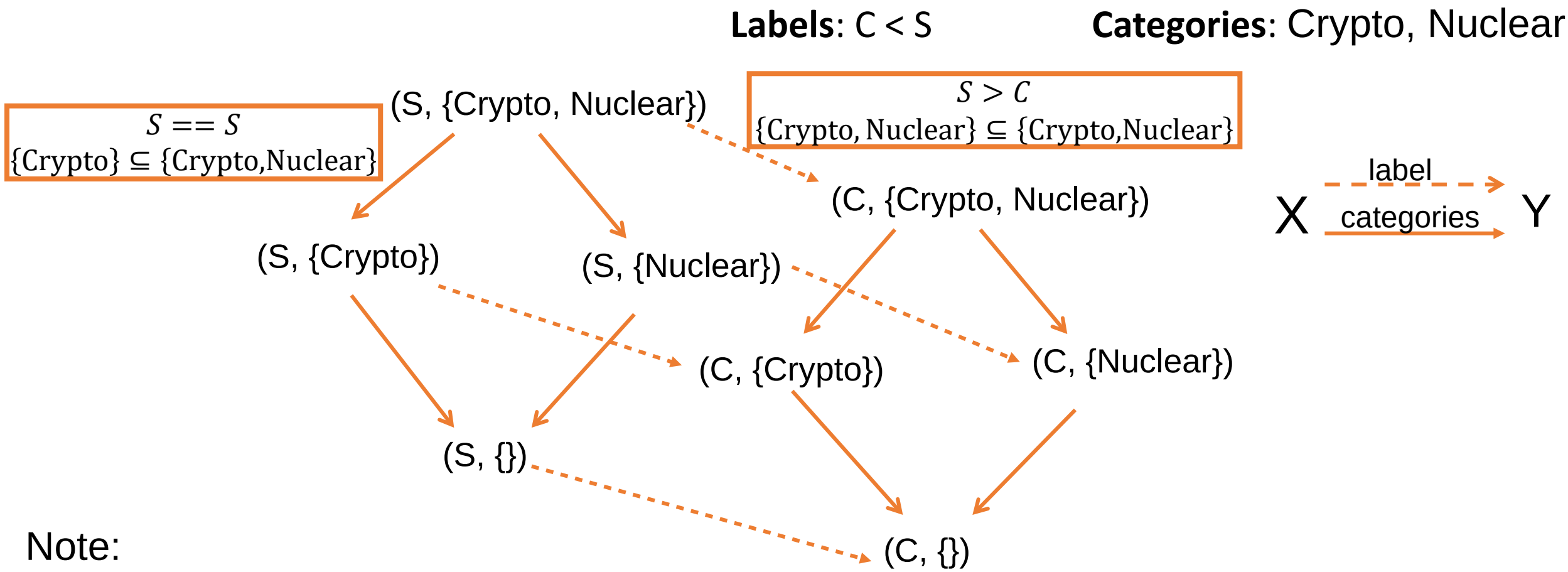
(C, {}) dominates (C, {SIGINT})

(S, {SIGINT, CRYPTO}) dominates (C, {CRYPTO})

(TS, {CRYPTO, SIGINT}) dominates (S, {NOFORN})

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

Establishing dominance: Lattice



Note:
(a) Dominates is transitive.
(b) Top and bottom elements.
(c) Only **partial** order.

DOMINANCE RELATIONSHIP

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

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

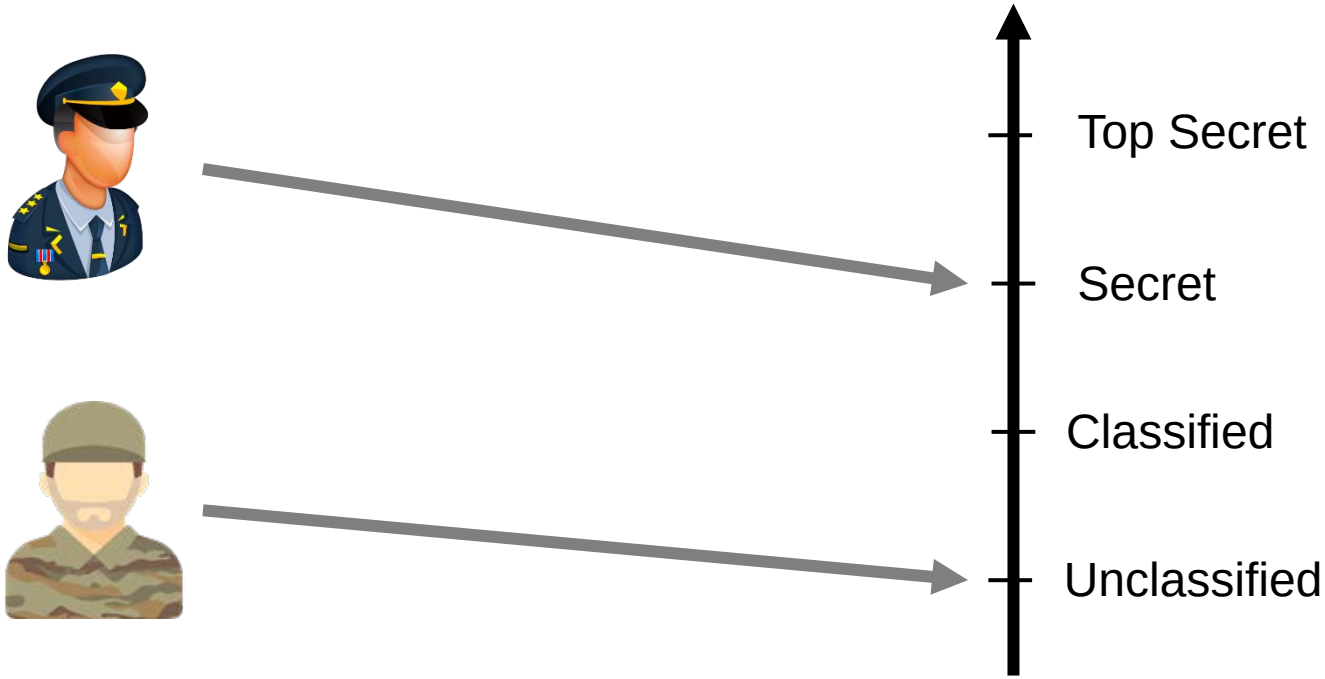
Level function for subjects: **Clearance**

BLP calls this “classification” too

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)***

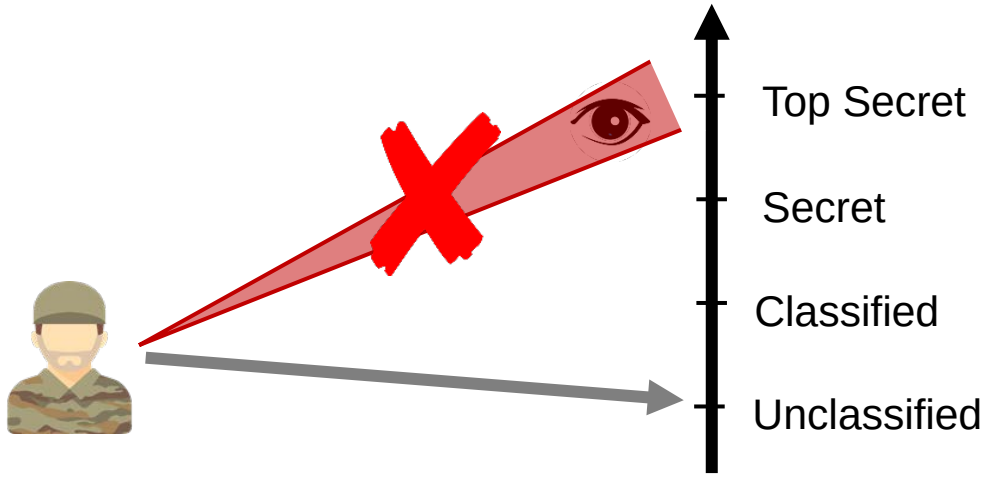
*level(S_i) **must** dominate current-level(S_i) !!!*



BLP System Properties:

SIMPLE SECURITY PROPERTY (SS-PROPERTY)

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

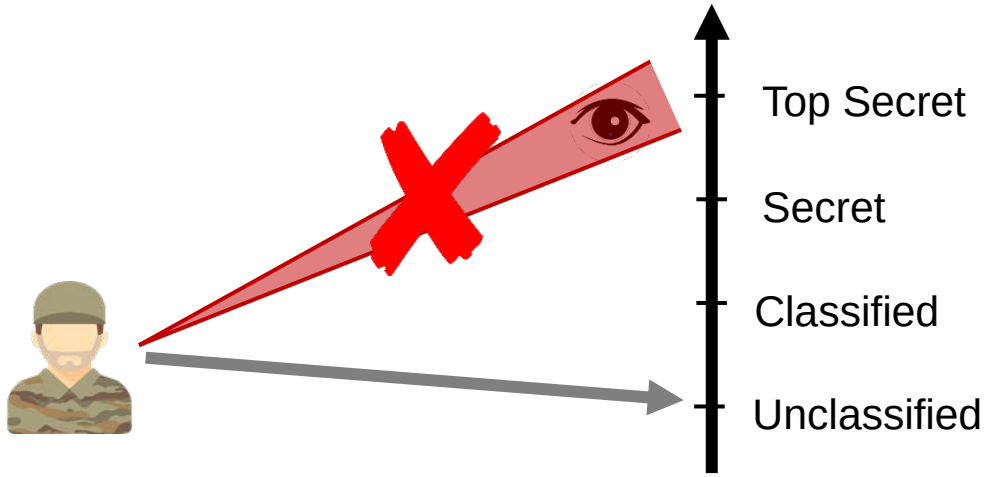


This used to be the whole policy. Is it enough?

BLP System Properties:

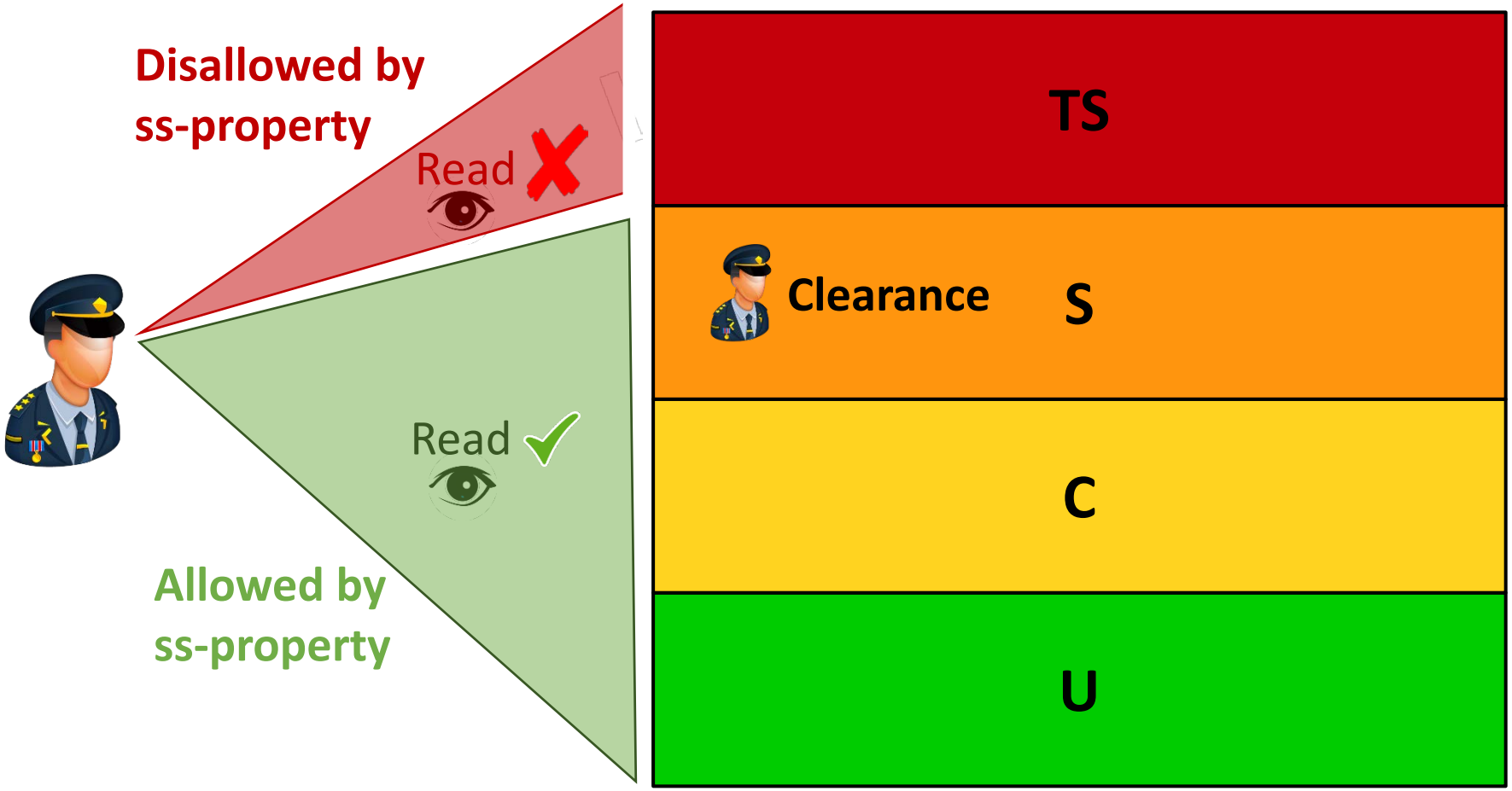
SIMPLE SECURITY PROPERTY (SS-PROPERTY)

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



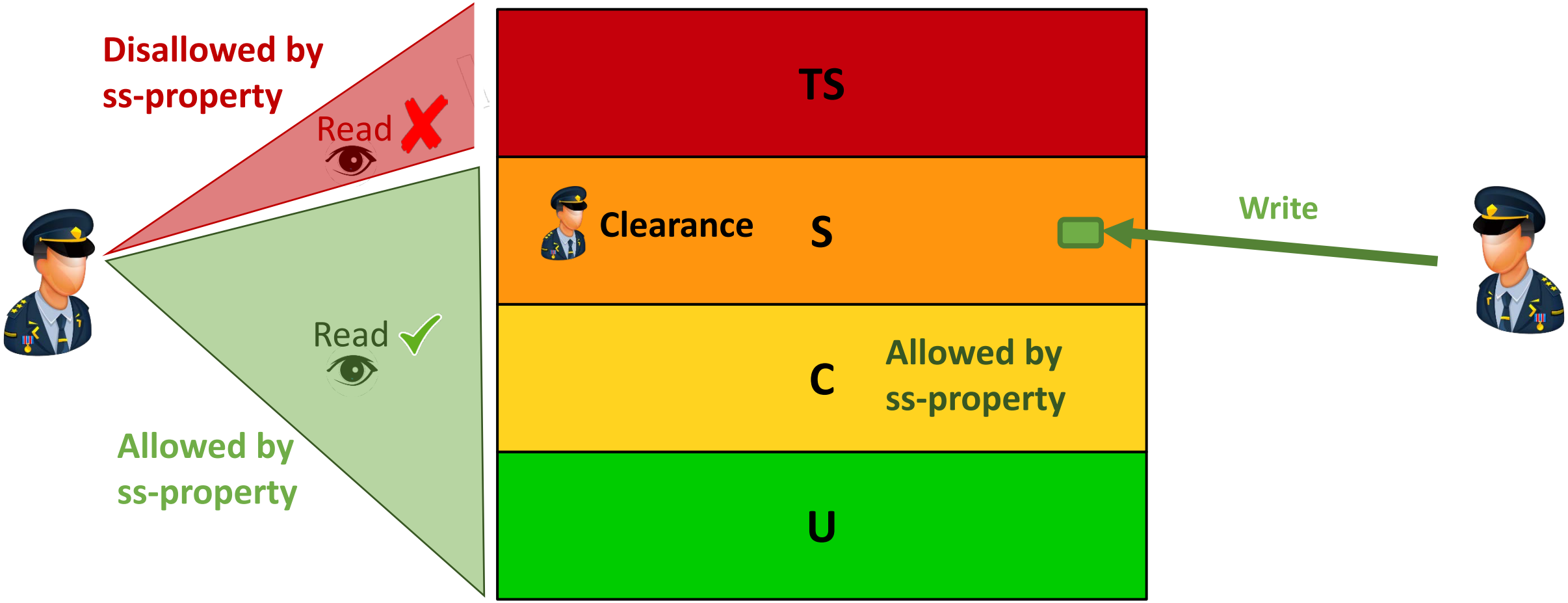
Why is the ss-property not sufficient?

No Read Up (NRU)



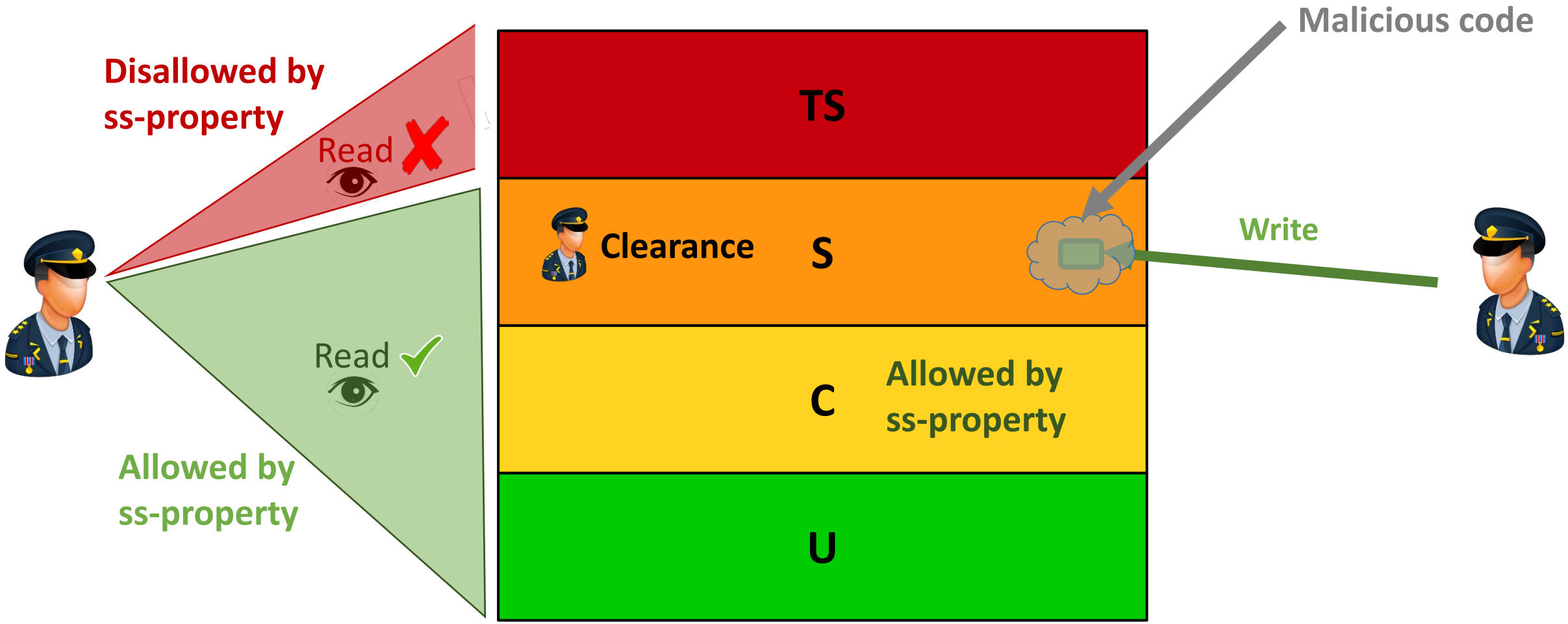
Why is the ss-property not sufficient?

No Read Up (NRU)



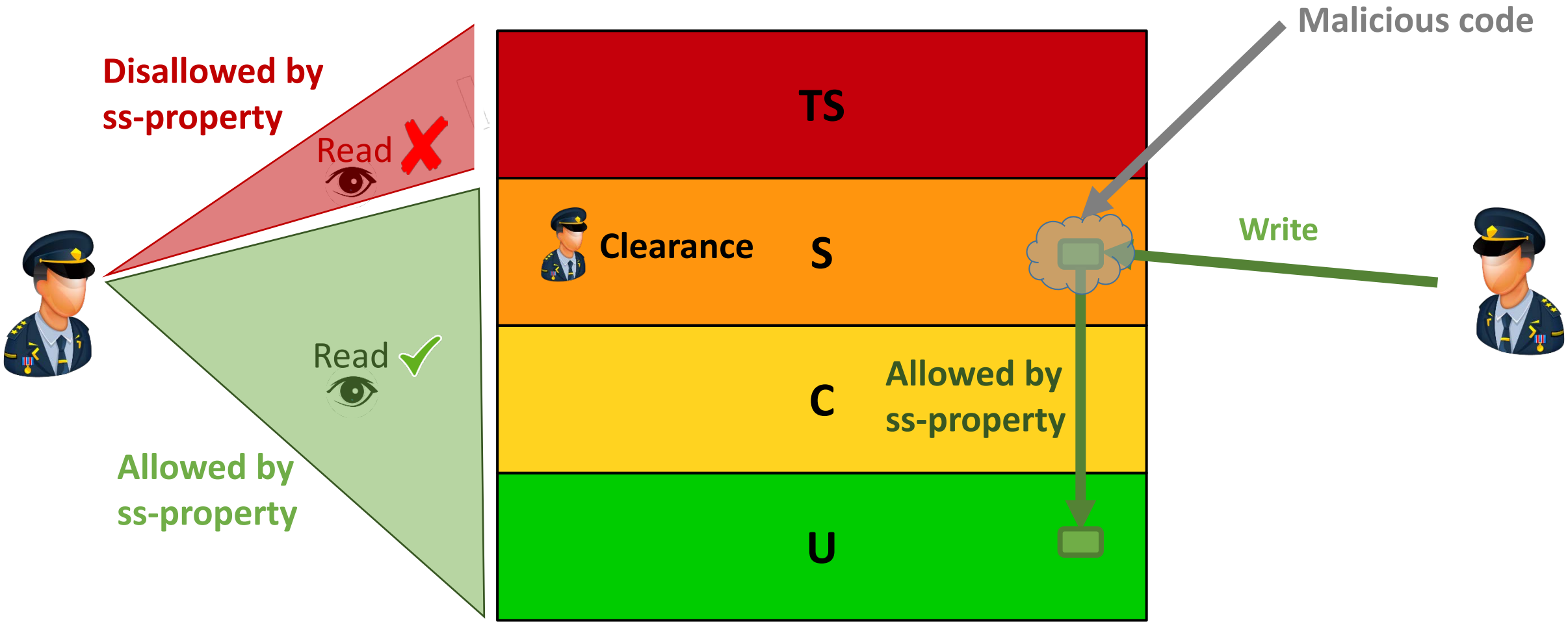
Why is the ss-property not sufficient?

No Read Up (NRU)



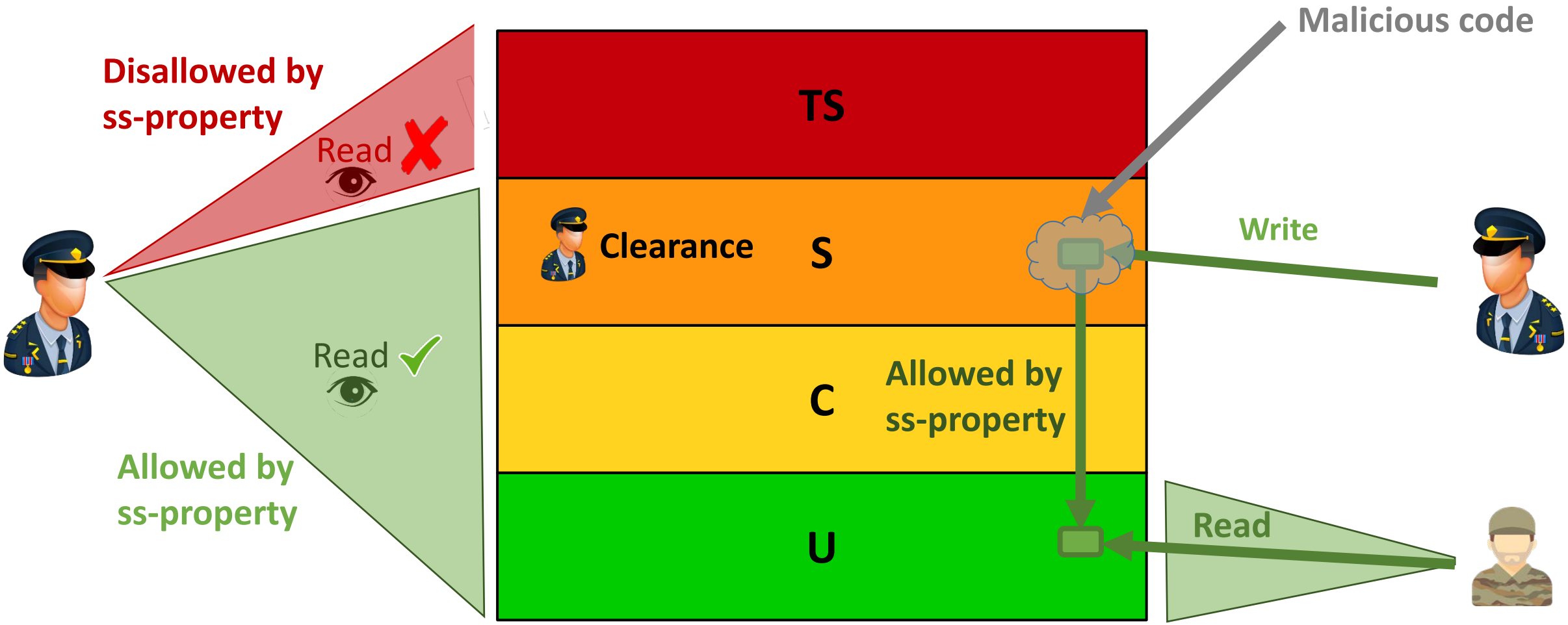
Why is the ss-property not sufficient?

No Read Up (NRU)



Why is the ss-property not sufficient?

No Read Up (NRU)



BLP System Properties

	No Alteration	Alteration
No observation	execute	append
Observation	read	write

STAR PROPERTY (*-PROPERTY)

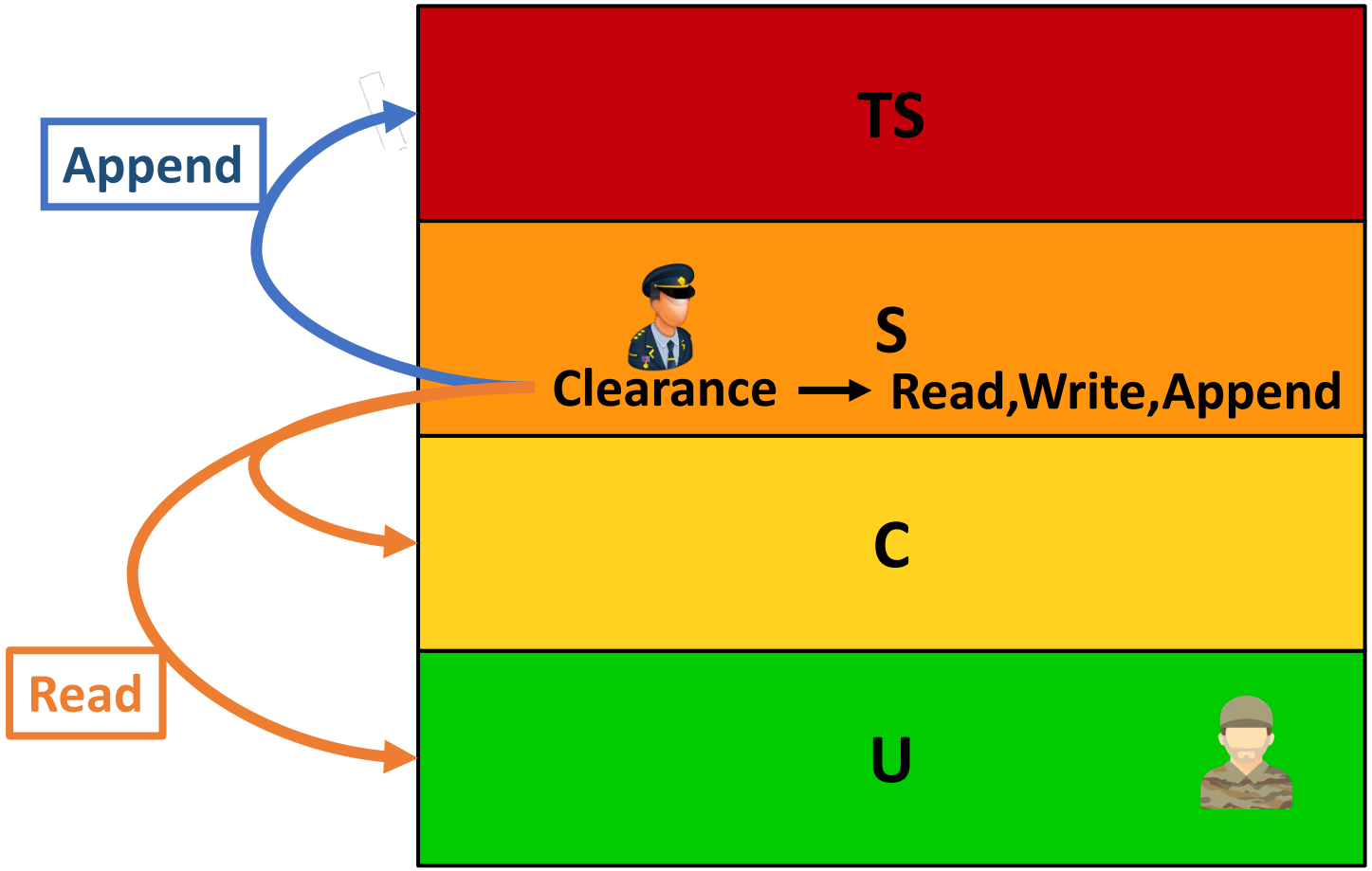
if a subject has simultaneous "observe" (r,w) access to O_1
and "alter" (a,w) access to O_2 then level (O_2) dominates level (O_1)

level(a-object) dominates level(w-object)
level(w-object) equals level(w-object)
level(w-object) dominates level(r-object)

Subjects ***current-level(s)*** {
level(object) dominates *current-level(subject)* if "append"
level(object) equals *current-level(subject)* if "write"
level(object) is dominated by *current-level(subject)* if "read

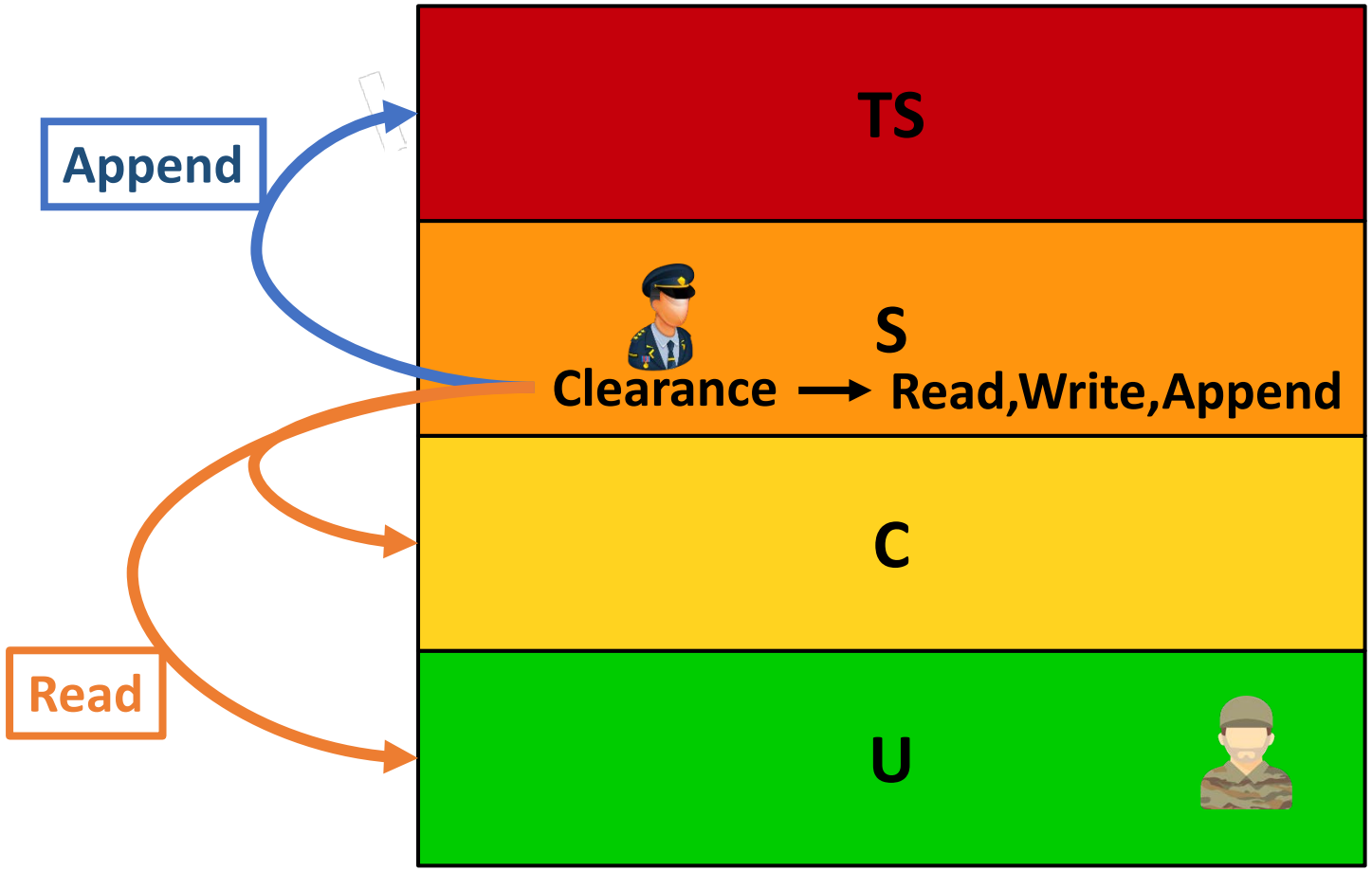
ss-property and *-property

No Read Up (NRU) **No Write Down (NWD)**



ss-property and *-property

No Read Up (NRU) **No Write Down (NWD)**



Discretionary security property (ds-property)

MAC: Levels (ss-property and *-property)

Discretionary security property (ds-property)

MAC: Levels (ss-property and *-property)

6 – Least privilege principle

Discretionary security property (ds-property)

MAC: Levels (ss-property and *-property)

6 – Least privilege principle

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

Also needs **DAC**!

DISCRETIONARY PROPERTY (DS-PROPERTY)

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

Useful for protecting integrity!!

Full BLP – Basic Security Theorem

BLP

Classification of objects

Clearances of subjects

Properties

- ss-property
- *-property
- ds-property (matrix)

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

Full BLP – Basic Security Theorem

BLP

Classification of objects

Clearances of subjects

Properties

- ss-property
- *-property
- ds-property (matrix)

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!

Full BLP – Basic Security Theorem

BLP

Classification of objects

Clearances of subjects

Properties

- ss-property
- *-property
- ds-property (matrix)

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.

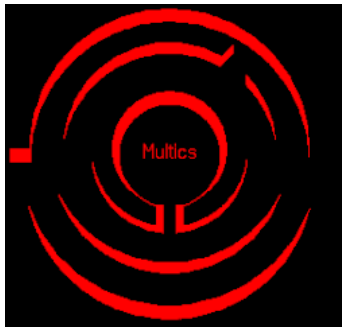
(2) the *-property holds.

(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!!

State transitions? Example: Multics OS



MIT, Bell Labs, General Electric
1965

Origin – UNIX, Bell-LaPadula model

- Get access and release access (Access Set)
- Change level or current-level (Levels)
- Give permission or take permission (Matrix)
- Create object and delete object

Claim: if these operations follow the 3 security properties, then overall "security" cannot be violated

https://multicians.org/multics-data-security.html

BLP – Problems

- Confidentiality-oriented (integrity? availability?)

- (practical) State-based + single transition model: too low-level, not expressive

- The 3 security properties are not sufficient to ensure security of underlying system
  - Covert channels!

BLP – Problems: illegal information flows

Assume

- level(s_1), current-level(s_1), level(o_1) is TS
- level(s_2), current-level(s_2), level(o_2) is C

Sequence of events

- s_1 get access o_1, read, release access.
- s_1 change current-level to C
- s_1 get access to o_2, write, release access.

BLP – Problems: illegal information flows

Assume

- level(s_1), current-level(s_1), level(o_1) is TS
- level(s_2), current-level(s_2), level(o_2) is C

Sequence of events

- s_1 get access o_1, read, release access.
- s_1 change current-level to C
- s_1 get access to o_2, write, release access.

Every state and transition is “secure” but illegal information flow could exist

BLP – Problems: illegal information flows

Assume

- level(s_1) is TS
- level(s_2) is C

Sequence of events

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

BLP – Problems: illegal information flows

Assume

- level(s_1) is TS
- level(s_2) is C

Sequence of events

- s_2 creates o_2 → level(o_2) = C
- s_1 reads C and either:
  - changes the object level → level(o_2) = TS
  - leaves object level untouched → level(o_2) = C **Existence itself may be a problem!**
- s_2 attempts to access to o_2 → access success leaks 1 bit of information!

Every state and transition is “secure” but illegal information flow could exist

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

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

Mitigation: isolation and addition of noise.

- 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.

One other problem…

What transition type enables these illegal flows?

What happens if we . . .

downgrade all subjects to lowest security level
downgrade all objects to lowest security level
enter all access rights in the ACM

Is the system secure? It satisfies every security property of BLP!

"A system which can be brought to a state with no restrictions cannot be secure"

McLean

"This is application dependent. If the users need it, it should be possible. Otherwise it should not be implemented"

Bell

J. McLean: Reasoning About Security Models. IEEE Symposium on Security and Privacy 1987: 123-133

One other problem… The tranquillity property

BLP assumes static!!

TRANQUILITY

classification / clearance does not change during execution

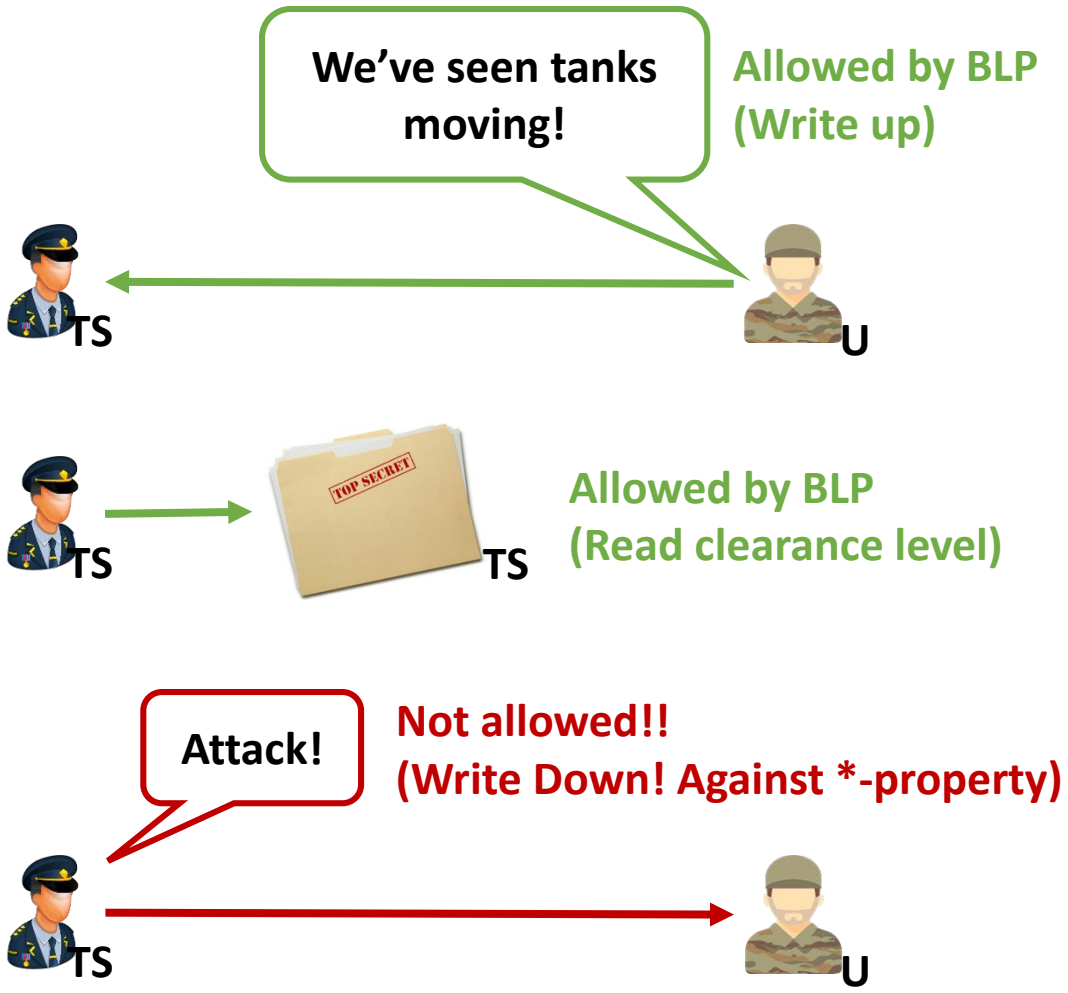
"transition-by-transition" does not allow for robust security arguments

Also… static is **not** practical !

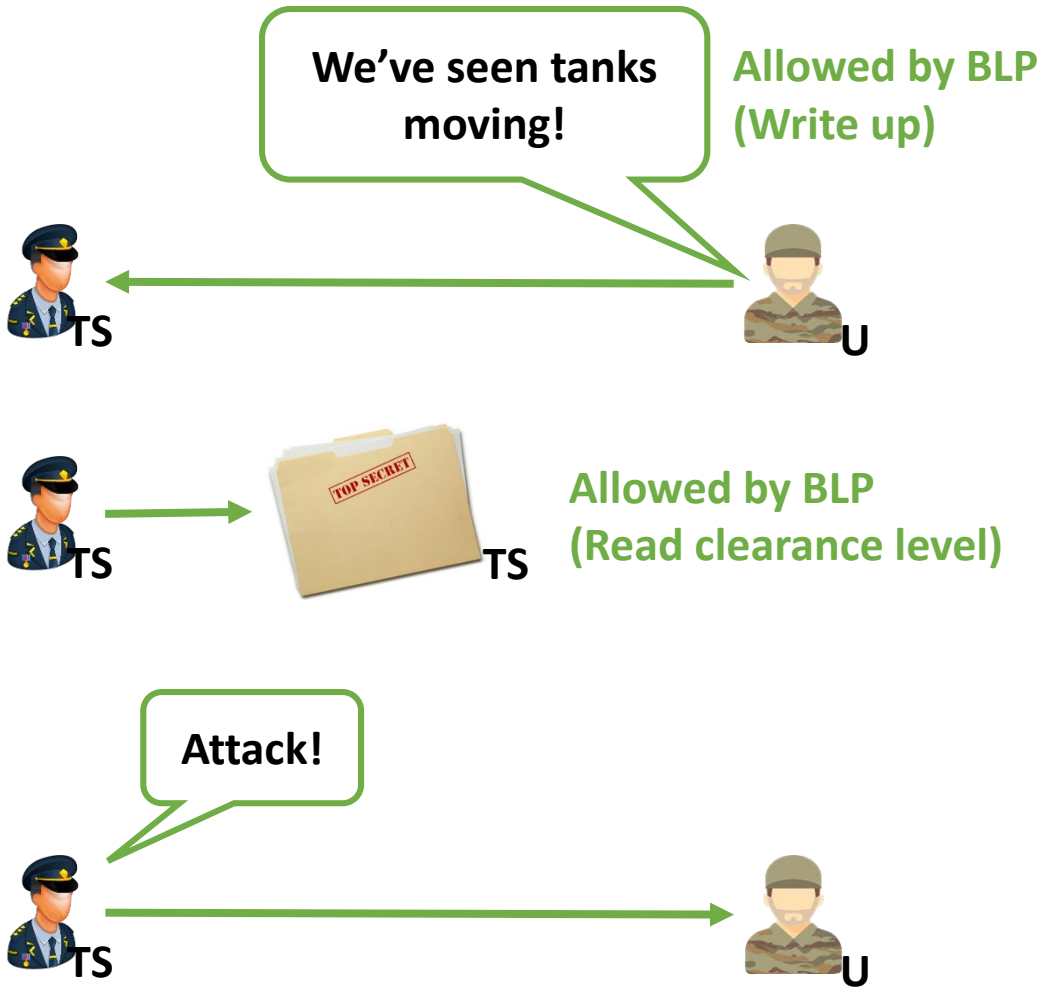
Declassification

Declassification

Declassification



Declassification



DECLASSIFICATION

remove classification label

Under the control of the security policy.

- It cannot be made inherently safe

(*manual process*)

-Rules about archives, historical records

It happens often

In practice… Declassification and Covert Channels!

Declassifying → from Secret to Unclassified.

You look at an object and think it is not sensitive any more

How do you know that there is not more information hidden in the object than what you can see?

Example: steganography

Steganography

STEGANOGRAPHY

discipline of **hiding secret information into a cover medium**

Image steganography

Hide messages in photos

(a) Hide message into the least significant bits of an image

Steganography

STEGANOGRAPHY

discipline of **hiding secret information into a cover medium**

Image steganography

Hide messages in photos

(a) Hide message into the least significant bits of an image

Strategic adversary!

Steganography

STEGANOGRAPHY

discipline of **hiding secret information into a cover medium**

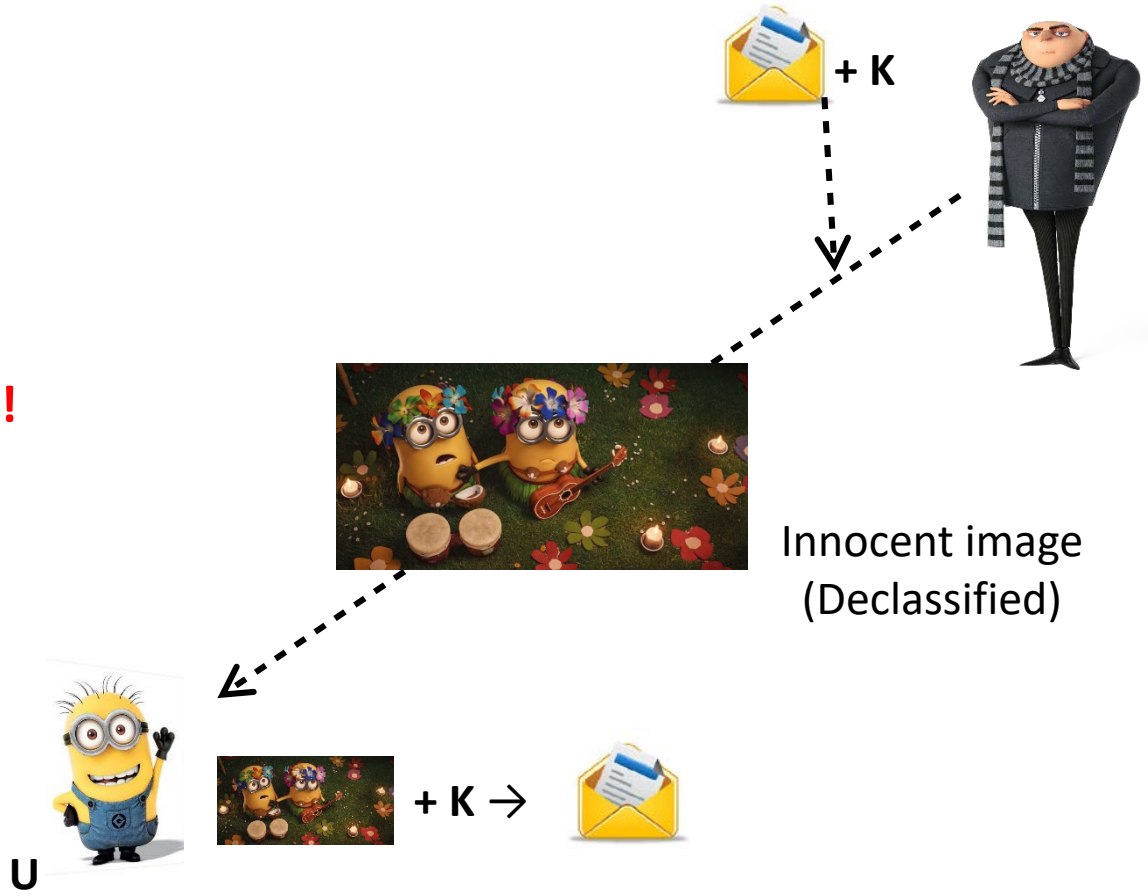
Image steganography

Hide messages in photos

(a) Hide message into the least significant bits of an image

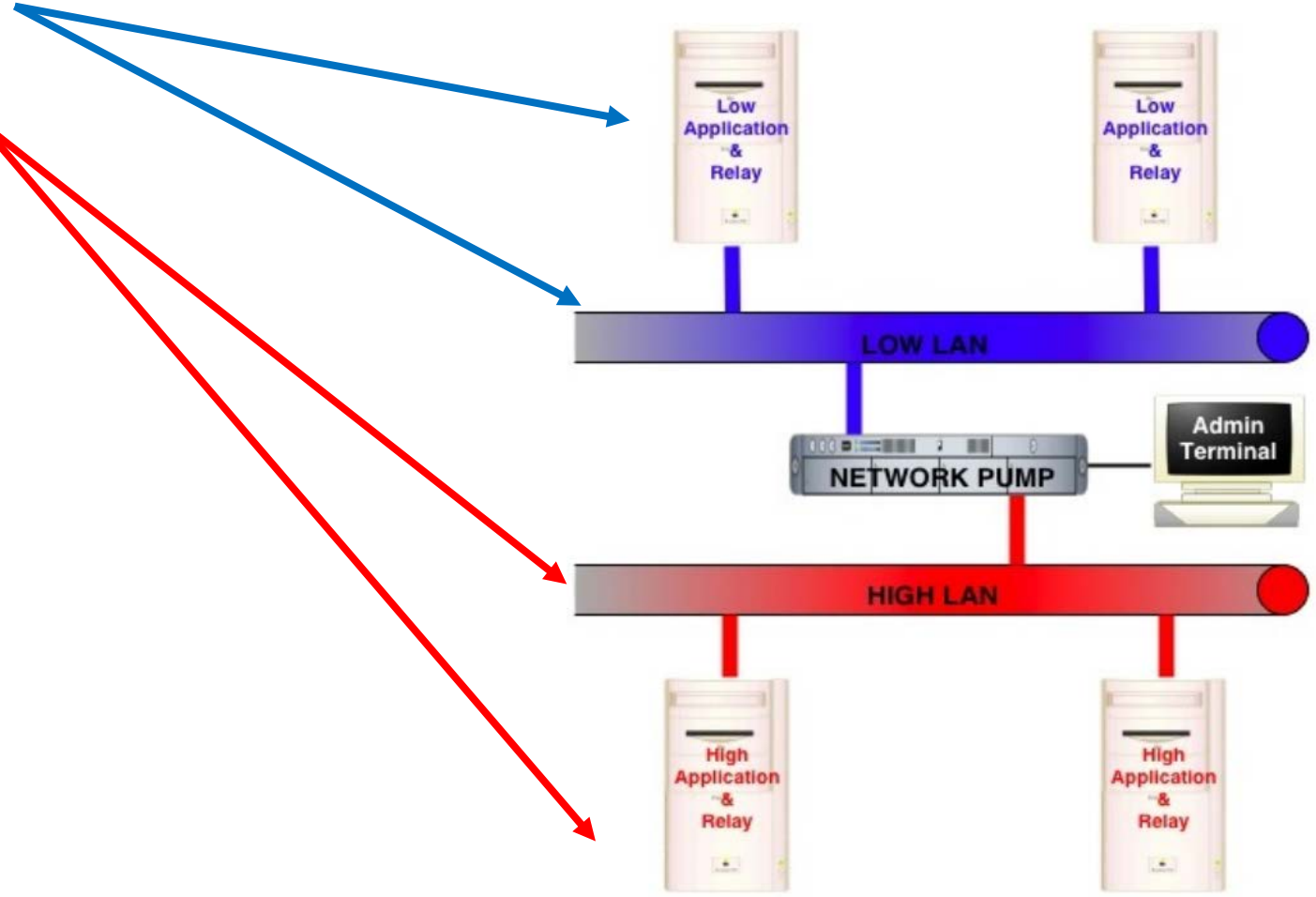
Strategic adversary!

(b) Use a key to encode/decode. Take advantage of photo natural variance to hide the existence of the message.



Real-world MLS systems: NRL pump (1990s)

"**High**" and "**Low**" on separate machines and separate networks



https://www.nrl.navy.mil/itd/chacs/sites/edit-www.nrl.navy.mil.itd.chacs/files/files/networkPumpBrochure_0.pdf

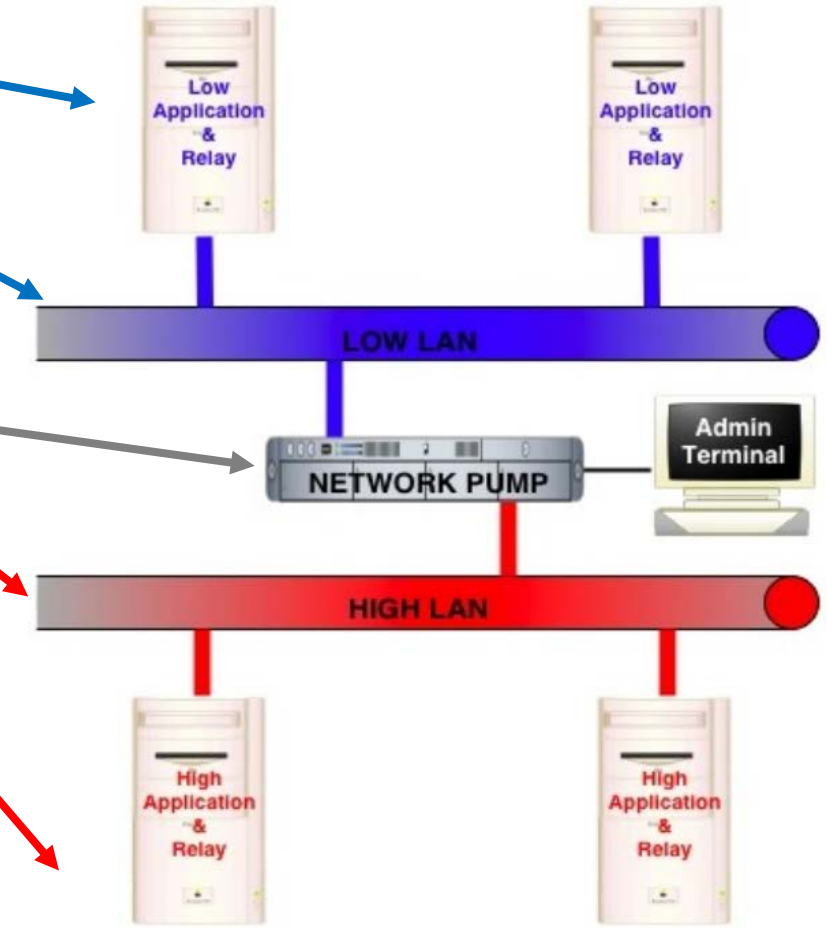
Real-world MLS systems: NRL pump (1990s)

"**High**" and "**Low**" on separate machines and separate networks

"**Pump**" network device:

- **High** can read **Low**
- **High** **cannot** go to **Low**

physical isolation!!



https://www.nrl.navy.mil/itd/chacs/sites/edit-www.nrl.navy.mil.itd.chacs/files/files/networkPumpBrochure_0.pdf

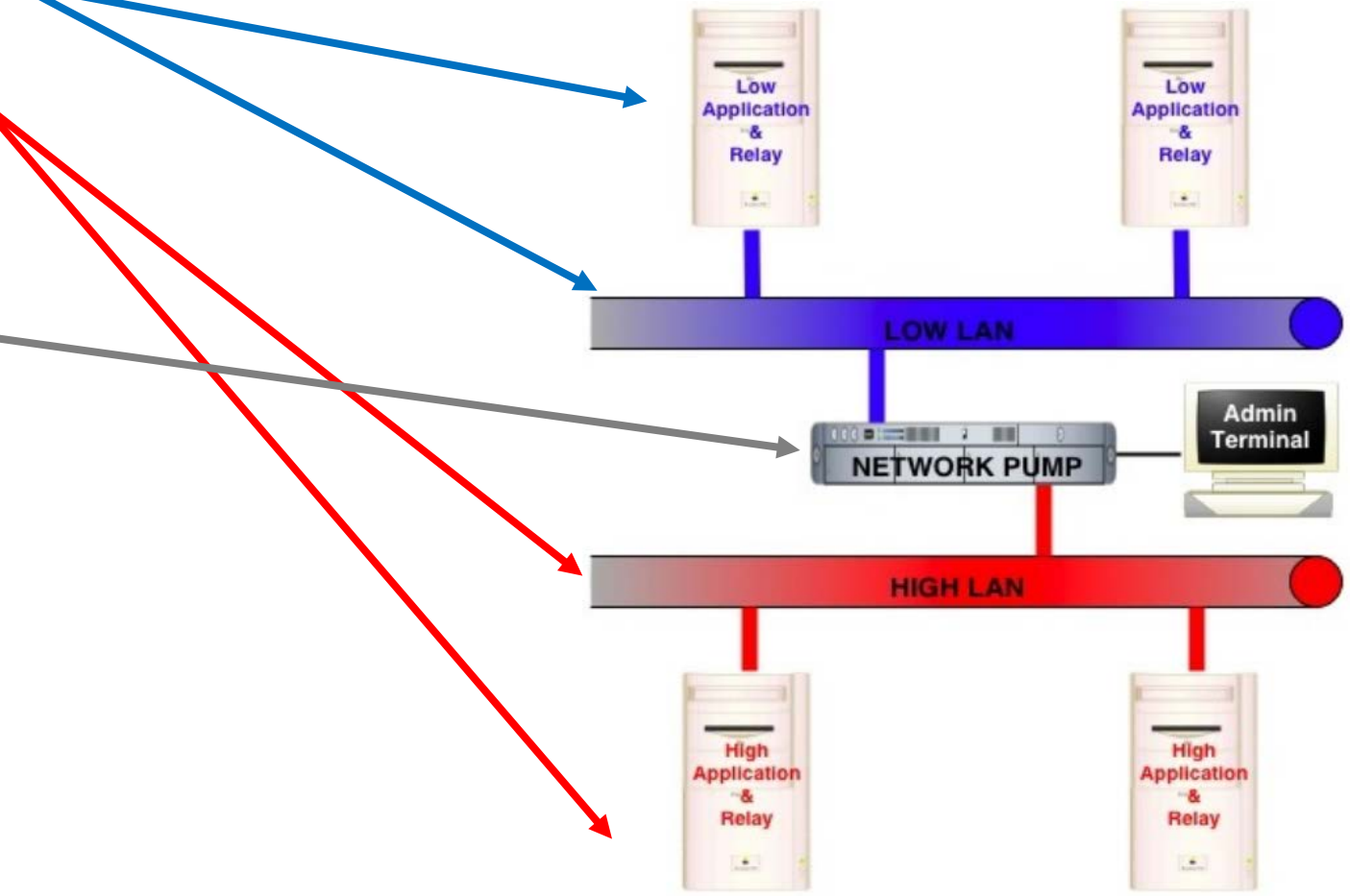
Real-world MLS systems: NRL pump (1990s)

"**High**" and "**Low**" on separate machines and separate networks

"**Pump**" network device:

- **High** can read **Low**
- **High** **cannot** go to **Low**

physical isolation!!



Network acknowledgements?

https://www.nrl.navy.mil/itd/chacs/sites/edit-www.nrl.navy.mil.itd.chacs/files/files/networkPumpBrochure_0.pdf

Real-world MLS systems: SE-Linux (2000s)

NSA contributed Linux kernel module that supports MAC.

- Confines programs to least privilege
- No “root” superuser
- Attaches domains to users and files, and only allows accesses according to a policy (e.g., lattice)
- Permissive mode allows debugging

Mostly used to confine “daemons”

Real-world MLS systems: SE-Linux (2000s)

NSA contributed Linux kernel module that supports MAC.

- Confines programs to least privilege.
- No “root” superuser.
- Attaches domains to users and files, and only allows accesses according to a policy (e.g., lattice)
- Permissive mode allows debugging

Mostly used to confine “daemons”

Covert channels? Is this still the worry?

PART II: Security models for integrity

References

Ross Anderson “Security Engineering”:

- Chapter 8 “Multilevel Security”
- Chapter 9 “Multilateral Security”

Dieter Gollmann “Computer Security”:

- Chapter 3 “Security models”
  - Biba, Chinese Wall

Why integrity matters

Military / government: secrecy and confidentiality → BLP, Lattices

Commercial services

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

Preventing fraud is about **protecting integrity**.

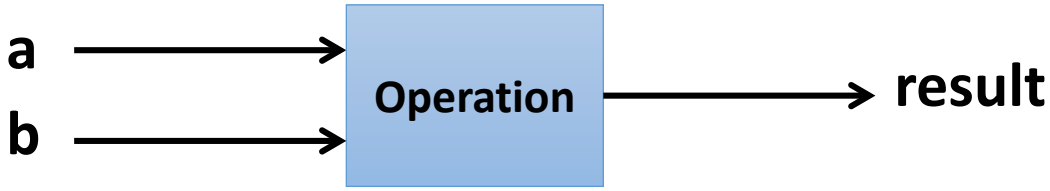
Confidentiality is either *secondary* or *unnecessary*.

Integrity is **key** for security in general!!

TCB has to have high integrity.

Public key cryptography requires high-integrity for confidentiality

What is integrity



Confidentiality: hide a, b and / or result

Integrity: is a, b or result correct given the operation?

- e.g., Adversary could influence inputs a and b from honest parties
- e.g., Adversary could exchange Operation for Operation'
- e.g., Adversary could change result
- e.g., Adversary could perform this operation more than once

system operates as if there was no adversary at all

Principles of Integrity mechanisms

1. SEGREGATION / SEPARATION OF DUTIES / DUAL CONTROL
Require multiple principals to perform an operation

Examples:

- *Accountant records payments / income and stores checks deliveries/sales. The two must match!*
- *Both sides of a transaction need to keep a record, and the records must match (legal receipts)*
- *Two officers are required to launch a missile*
- *Developers should not also be operators*

Principles of Integrity mechanisms

2. ROTATION OF DUTIES

Allow a principal only a limited time on any particular role
& Limit other actions while in this role

Additional deterrent to fraud

No one entity has control over an entire process

Reduces risk of compromise (new observers to find irregularities)

Examples:

- *Guards appointed for a single (random) shift to guard the bank safe*
- *Tellers changed over time so that they cannot fiddle with accounts*

Principles of Integrity mechanisms

3. SECURE LOGGING

Tamper evident log to recover from integrity failures. Consistency of log across multiple entities is key to the above

Examples:

- *Log of transactions inside the ATM machine.*
- *Cash till has an internal log.*
- *Notaries maintain logs of transactions and contracts.*
- *Bitcoin!*

The BIBA model

Multilevel security approach to tackle the integrity problem

A family of models subject to refinements and extensions

Reminder:

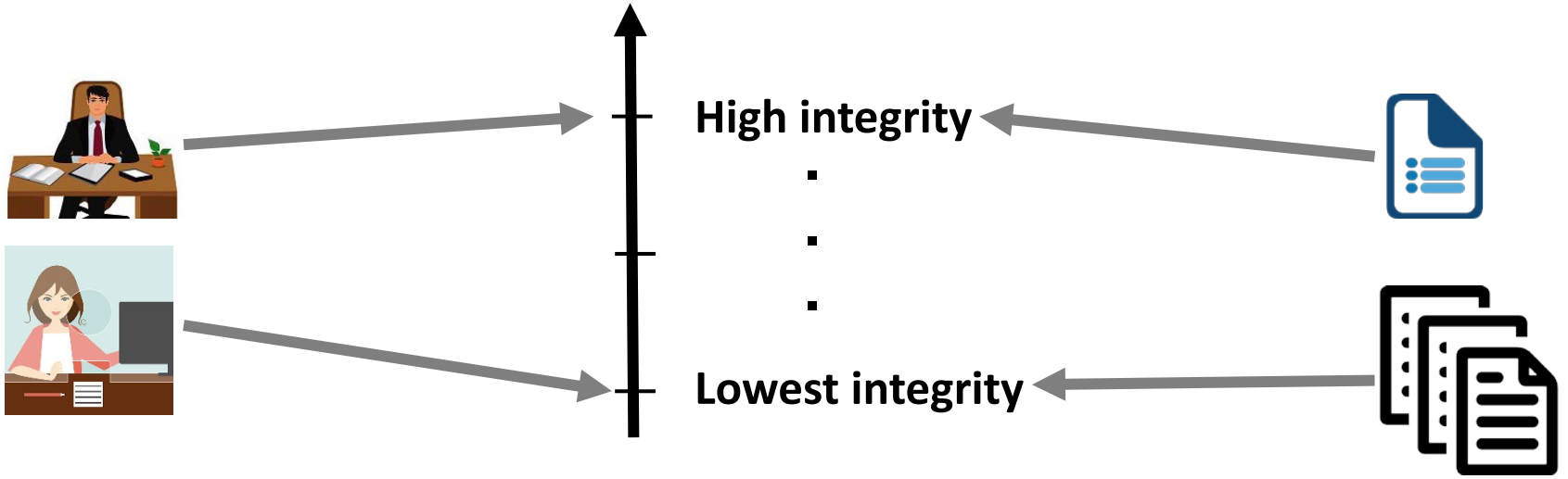
A security model is a **design pattern** to achieve a certain property

It *might, or might not,* be appropriate for a specific security policy

It *might need specializing* and adapting to solve specific problems

Biba, K. J. "*Integrity Considerations for Secure Computer Systems*", MTR-3153, The Mitre Corporation, April 1977

BIBA is a multilevel security policy

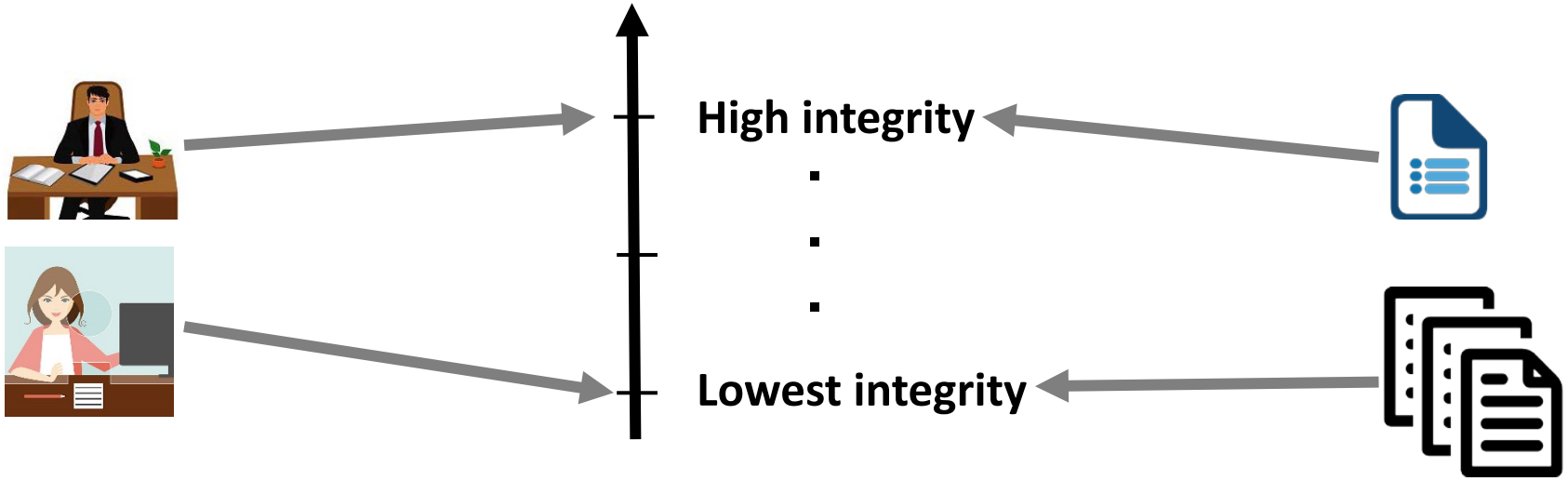


Two operations:

"read" (observe)

"write" (modify)

BIBA is a multilevel security policy



Two operations:

"read" (observe)

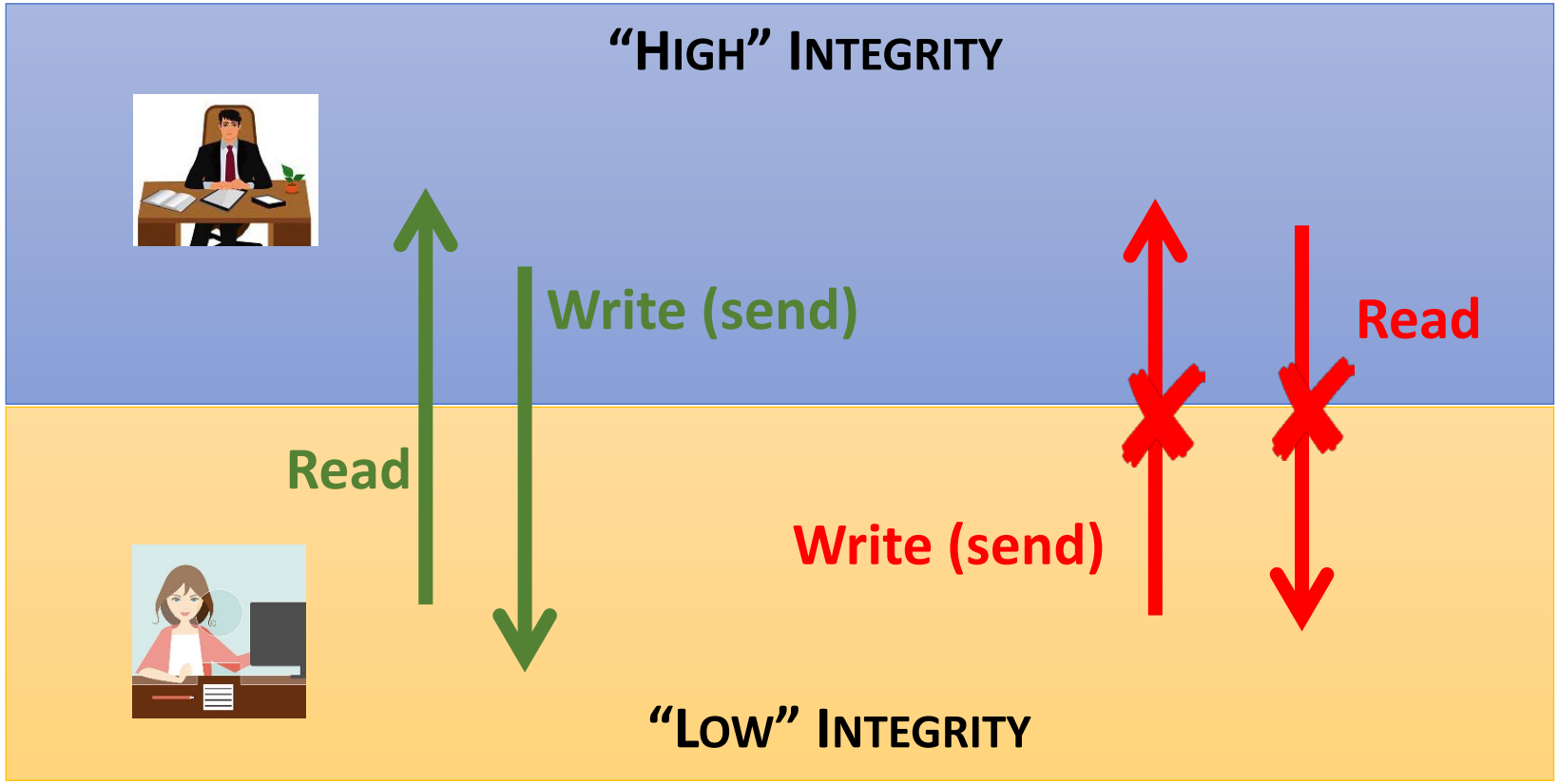
"write" (modify)

Two key rules (strict)

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

- ***No write-up (*-integrity)***: 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 and BLP are dual

Both assume a partial order of labels for subjects and objects.

- **BIBA**: integrity labels
- **BLP**: confidentiality labels

Rules are inverse:

- **BIBA**: no-write (alter) up
- **BIBA**: no-read down

BLP: no-read up.

BLP: no-write (alter) down.

BIBA variants: 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 (Label creep !)

- Example: mitigate impact of a network trojan

BIBA variants: 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 (Label creep !)

- Example: mitigate impact of a network trojan

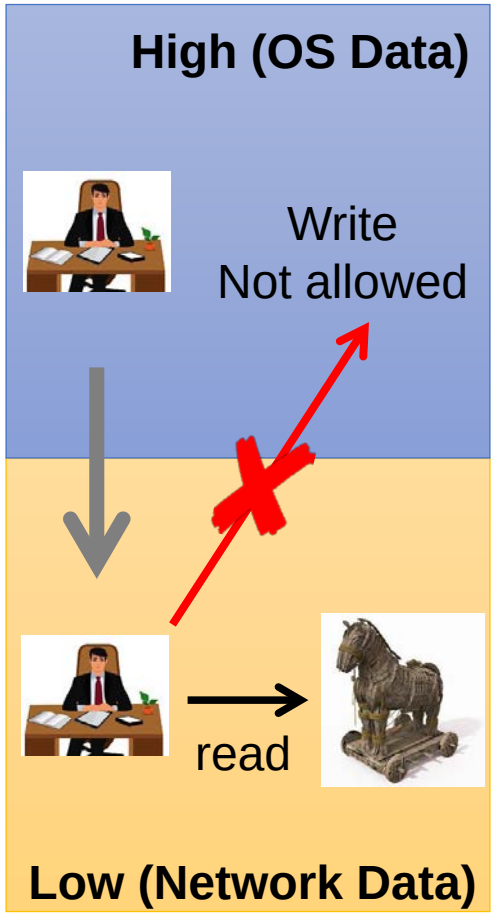
BIBA variants: Low-water-mark for subjects

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 (Label creep !)

- Example: mitigate impact of a network trojan

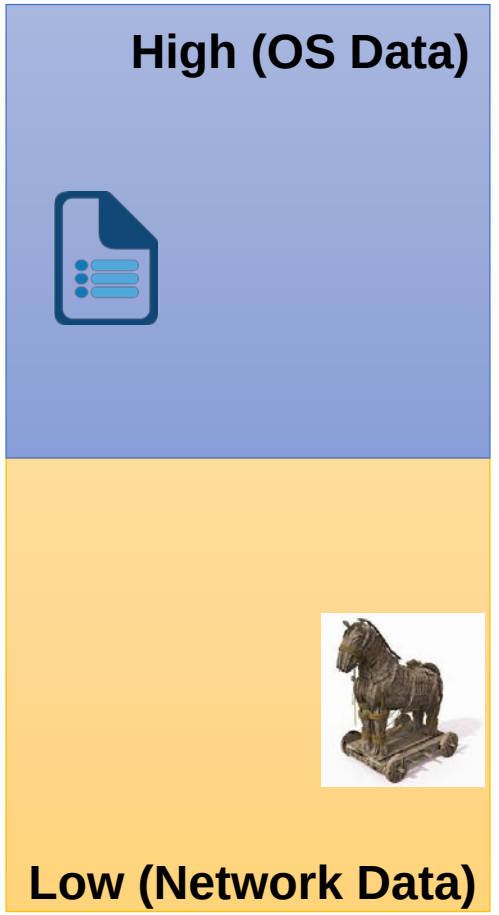


BIBA variants: Low-water-mark for objects

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 & sanitize / erase, selective objects

BIBA variants: 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.

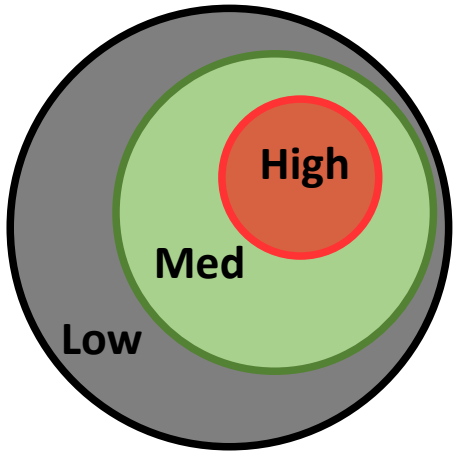
A high-integrity database written to by a process with access to the network (low integrity) is labelled at "low" integrity

What is the effect?

Dangerous! only allows for integrity violation detection

Mitigation: replicate & sanitize / erase, selective objects

BIBA variants: Ring Policy

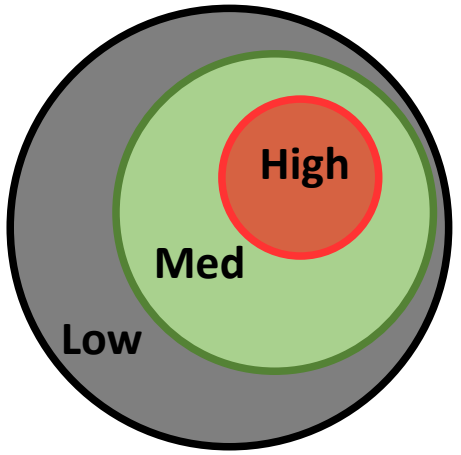


Rules of “ring” policy

- Integrity levels of objects and subjects are fixed.
- No write up (*classic BIBA*).
- A subject **may read an object at any level**.
- Ring invocation property: a principal can “invoke” a program with a high integrity label, that itself may be able to write at higher levels

!= classic BIBA

BIBA variants: Ring Policy



Rules of “ring” policy

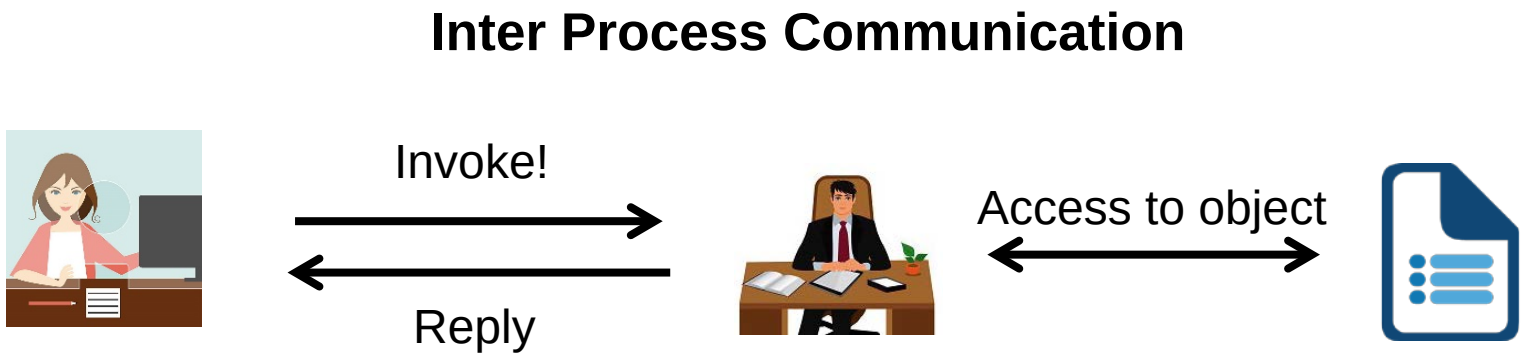
- Integrity levels of objects and subjects are fixed.
- No write up (*classic BIBA*).
- A subject **may read an object at any level**.
- Ring invocation property: a principal can “invoke” a program with a high integrity label, that itself may be able to write at higher levels

!= classic BIBA

Principals and their programs are trusted to keep the integrity invariants at their level and below despite being able to read from all levels, and write on behalf of others

Confused deputy problems!

BIBA Additional actions: Invoke



The right of a subject to run and communicate with another subject

e.g., Director at “High” invoked by Teller's software at “Low”

Who is allowed to invoke what?

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 (Ring Invocation)

Only allow subjects to invoke subjects that dominate them

+ prevents corruption of high integrity data

- what about imperfect sanitization?

Who is allowed to invoke what?

Simple Invocation

Only allow subjects to invoke subjects with a label they dominate

+ protect high integrity data from misuse by low integrity principals

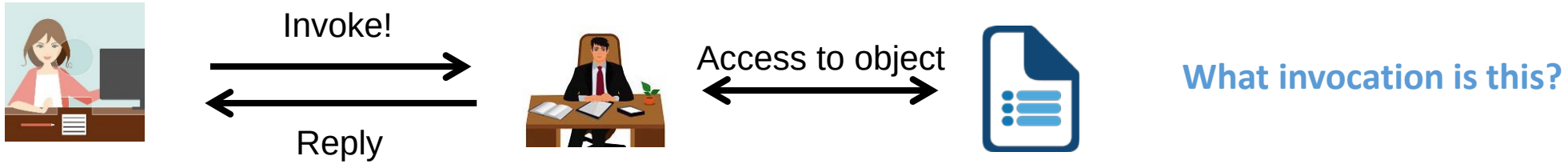
- what level is the output?

Controlled Invocation (Ring Invocation)

Only allow subjects to invoke subjects that dominate them

+ prevents corruption of high integrity data

- what about imperfect sanitization?



What invocation is this?

Sanitization

SANITIZATION

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

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

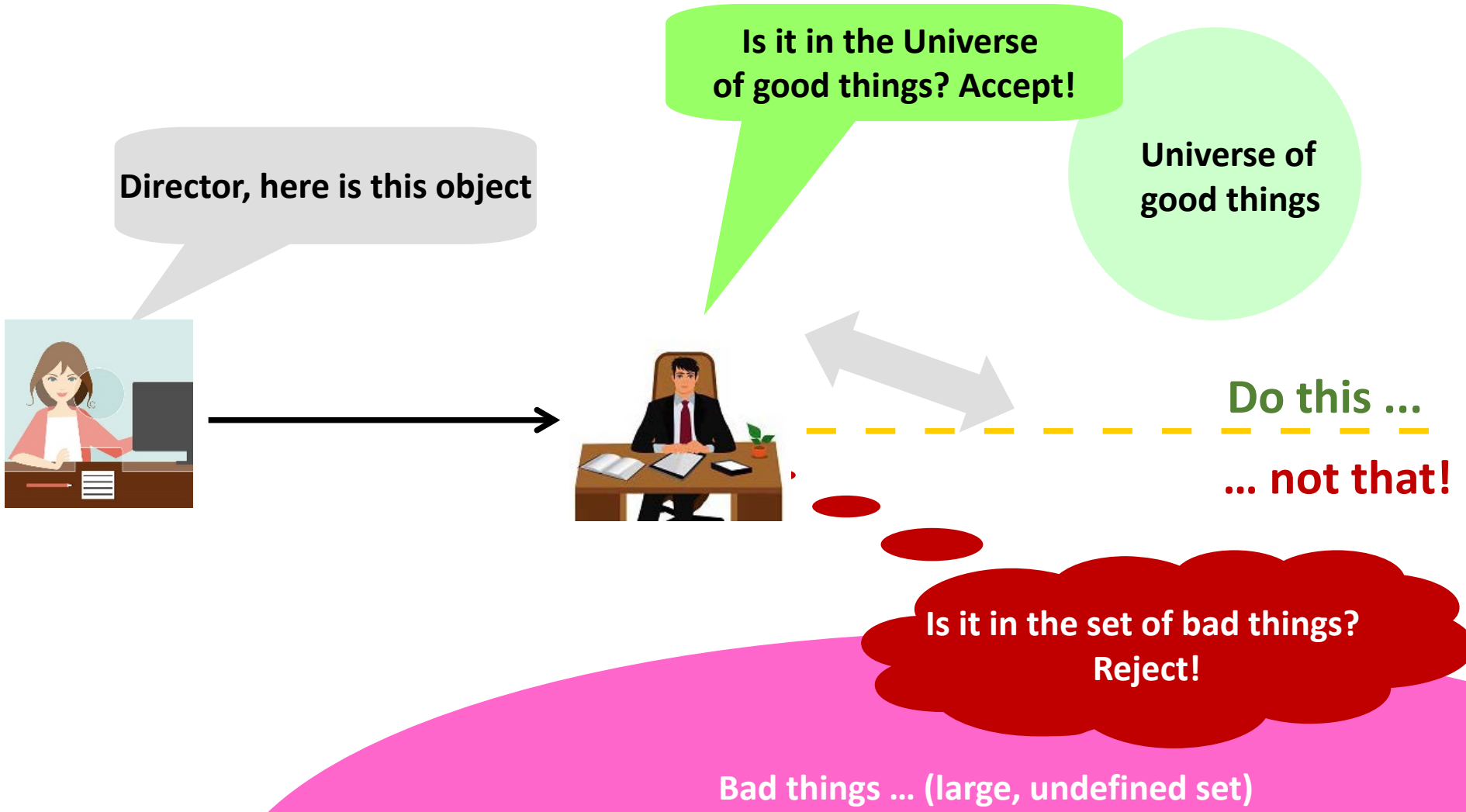
"Base access decisions on permission rather than exclusion" **[SS75]**

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.

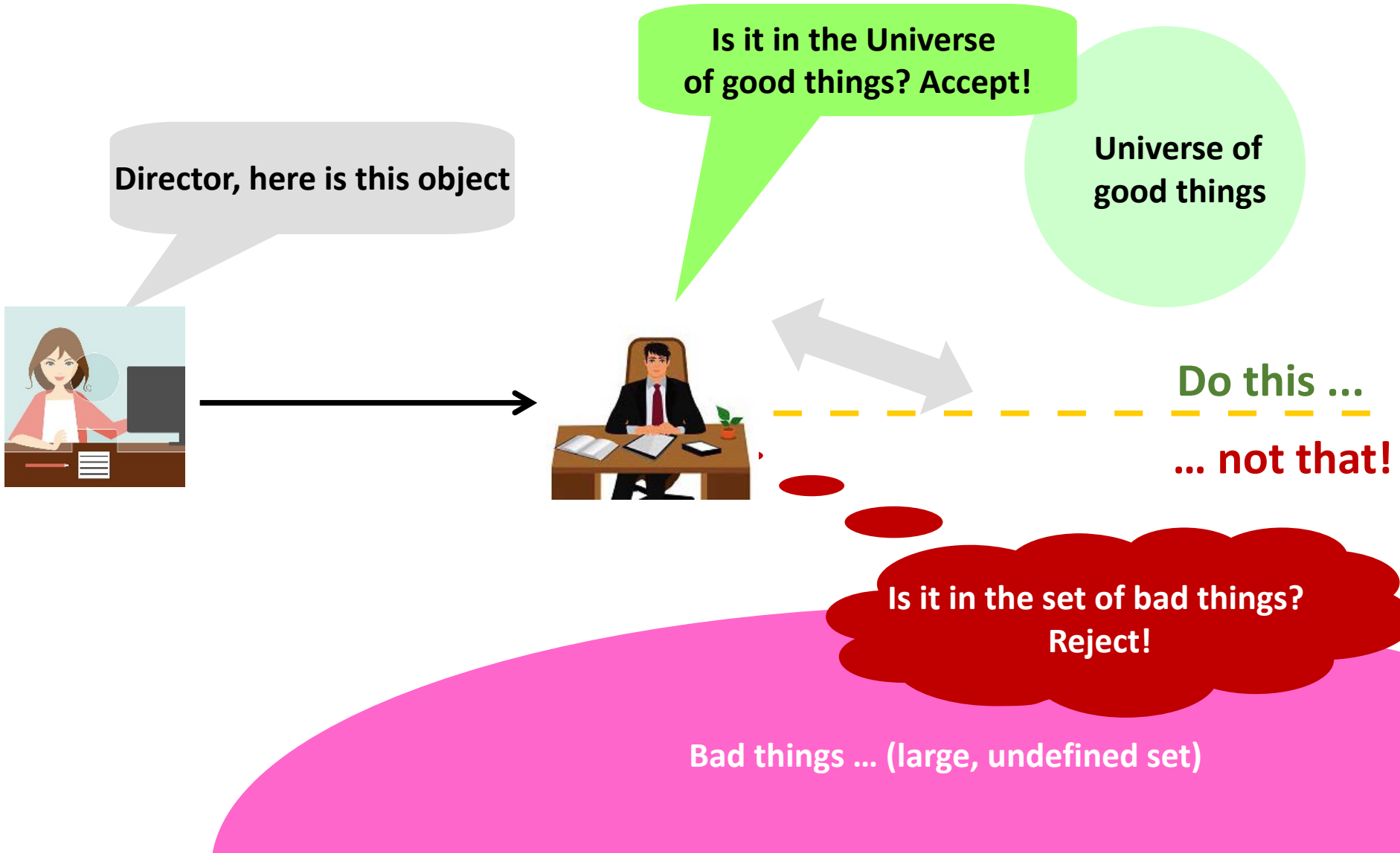
Insert a photo in a web album? Ensure caption is from a restricted set of Unicode, or apply to it a transform to "escape" / "encode" any characters not from that safe set into it. Do not simply check it does not contain "<script>". (XSS Attack)

Principle illustrated



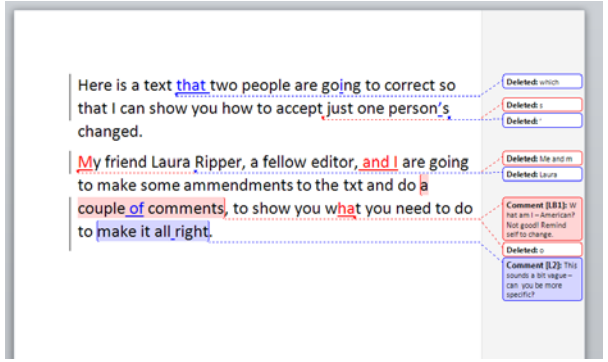
Principle illustrated

COVERT CHANNELS DIFFICULT TO CATCH!

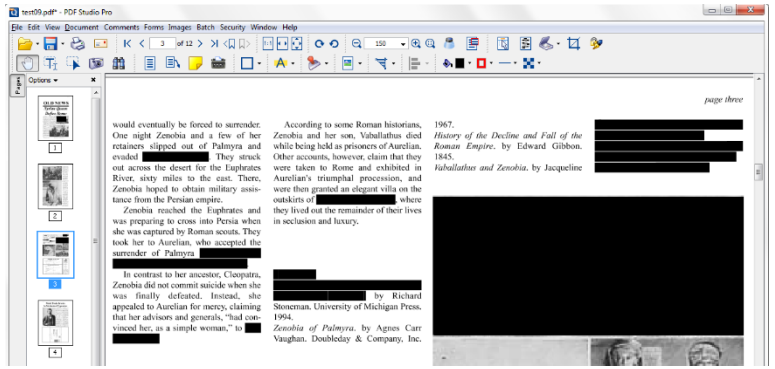


Declassification and Sanitization!

- 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*.

Combining security properties

Secure composition of mechanisms is hard!

Composing confidentiality and integrity

- BLP: confidentiality, no integrity
- BIBA: integrity, no confidentiality
- Example we study: Chinese Wall Model

From multi-level to multi-lateral security

- “Different” entities seek different properties.
- These properties may be opposed to each other.

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

Consultancy services for different clients

Financial advice and auditing of same client

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

Chinese Wall model: Entities and Basic Concepts

All objects are associated with a label denoting their origin

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

The originators define "conflict sets" of labels

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

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

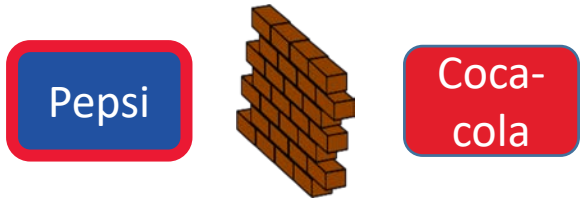
Chinese Wall model: Entities and Basic Concepts

All objects are associated with a label denoting their origin

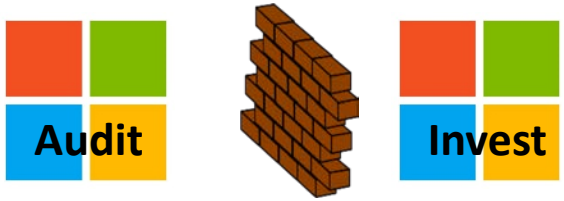
"Pepsi Ltd.", "Coca-Cola Co.", "Microsoft Audit", "Microsoft Investments"

The originators define "conflict sets" of labels

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



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



Chinese Wall model: Entities and Basic Concepts

All objects are associated with a label denoting their origin

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

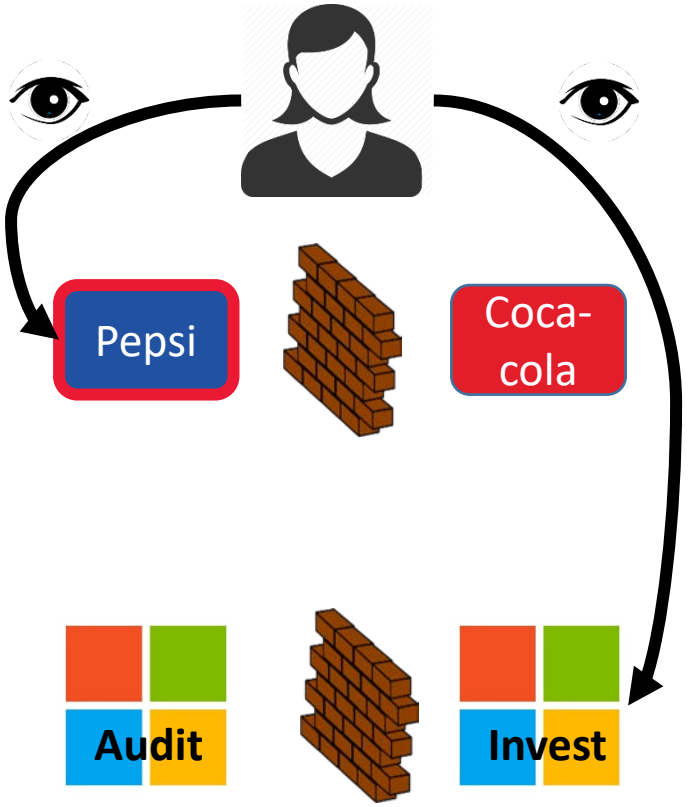
The originators define "conflict sets" of labels

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

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

ALICE

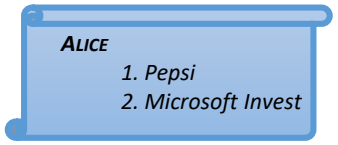
1. *Pepsi*
2. *Microsoft Invest*



Chinese Wall model: Access rules

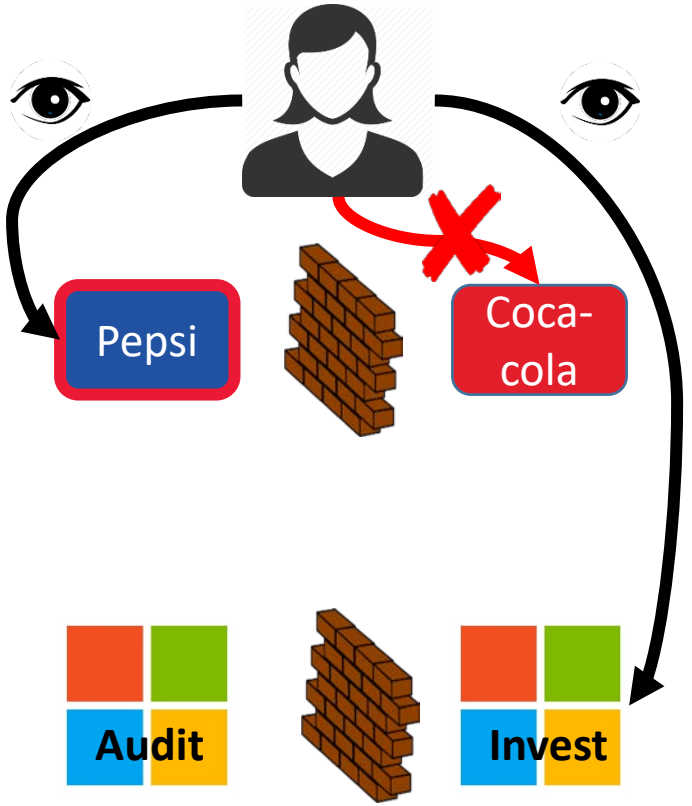
A subject can read an object (for either read or write) if the access ***does not allow an information flow*** between items with labels in the same conflict set

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!)

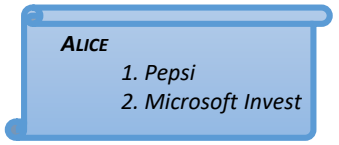
Why?



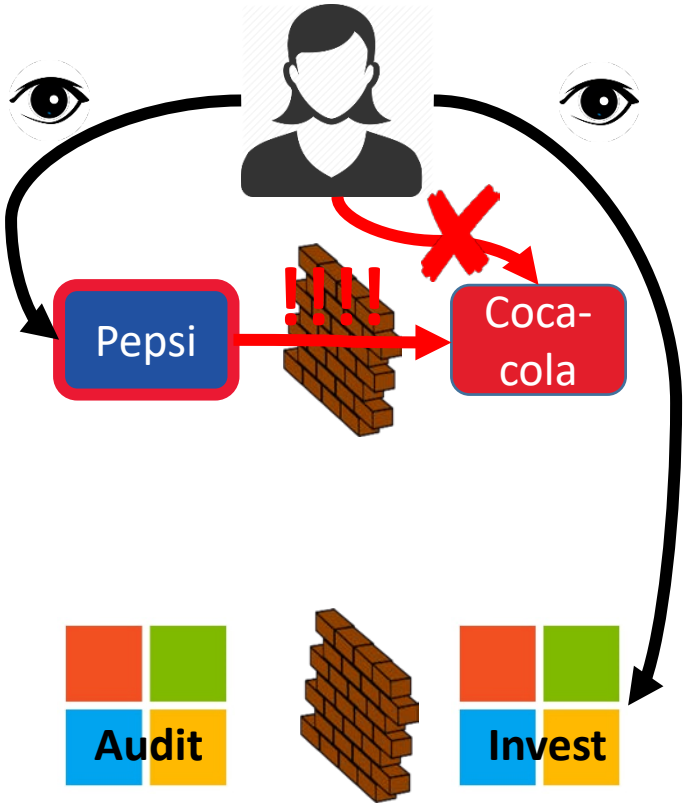
Chinese Wall model: Access rules

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!)



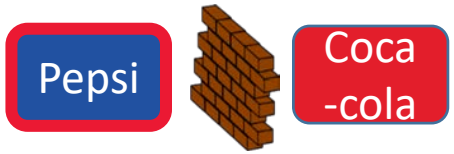
Why? She has already accessed files from “Coca-cola Co.” thus an information flow between those and “Pepsi Ltd” might happen

Chinese Wall model: Indirect flows

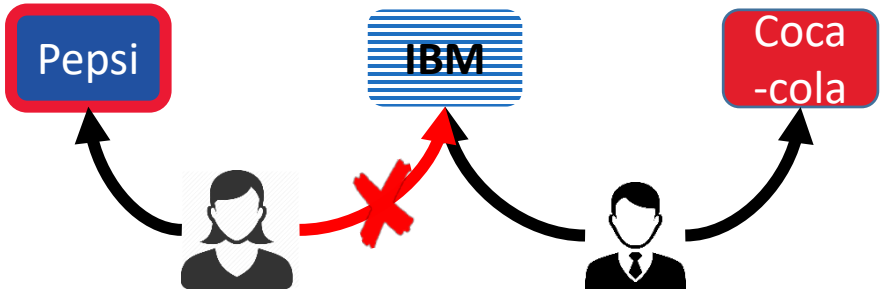
Direct flow within a conflict set is easy to detect! What about indirect?

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”

Multilateral security

At least two principals require two different, “incompatible” or even conflicting security properties

- Both Alice and Bob want financial privacy, but they wish to compute who is richest

- Alice wants privacy of her location, but her insurer Bob want to make sure she is not driving much on rural roads (integrity)

- Users of an on-line service want anonymity, but the service wants to identify who is committing fraud

- Elections: need to both provide privacy and integrity (even if you lose)

- In telephony: users want privacy, the network wants to be paid for communications

Who secures the TCB?

How to deal with multilateral security

1. Use a trusted third party

- They need to be trusted by all parties to enforce the security properties all parties care about.
- 5Cs: Cost, Compulsion, Collusion, Corruption, Carelessness.

2. Use some form of secure hardware

- One party provides the hardware, that is used by the other.

  SIM card in your mobile phone - Keeps the authentication keys away from YOU

  Digital Rights Management controls in DVD players - Keeps you away from some functions of the equipment.

- Single point of failure - manufacturer?

3. Modern Cryptography.

- Can enforce any multilateral security property.
- At what cost? Advanced Privacy Enhancing Technologies master course!

More security models

- The BMA model (Anderson)
- The Clark Wilson model (Anderson or Gollmann)
- Non-interference models (Gollmann)

- Common Criteria (Gollmann)

Summary of the day

- **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