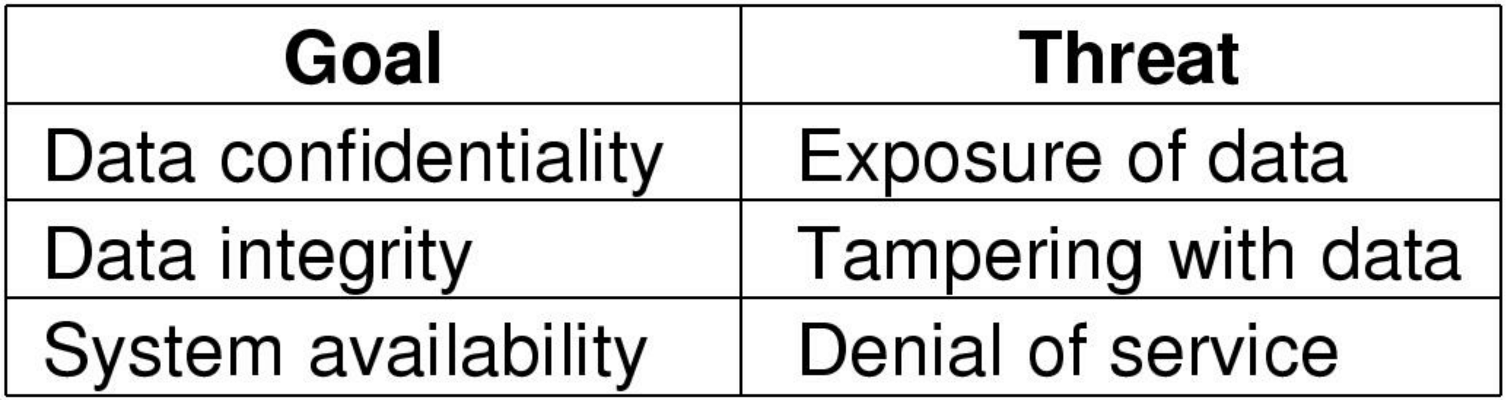
Operating System and Security

Kartik Gopalan

From: http://www.syslog.com/~jwilson/pics-i-like/kurios119.jpg

What is Security

- C.I.A



Goal	Threat
Data confidentiality	Exposure of data
Data integrity	Tampering with data
System availability	Denial of service

- Preventing unauthorized users from executing undesirable actions, such as
  - Stealing your data (C)
  - Giving you fake data/Tampering your data (I)
  - Preventing you from doing your work (A)

Securing what?

- Securing the OS from users
  - OS-level mechanisms
- Securing one user from another
  - Access control, isolation
- Securing users from OS!
  - Yes, sometimes the OS is not trusted by the user.
  - E.g. in a cloud users may not trust the cloud platform’s OS.

Security mechanisms in OS and hardware

- CPU Execution privileges ("Who can access?")
  - Part of CPU state
  - x86 privilege rings (0,1,2,3) in EFLAGS
  - VTx provides root and non-root modes
- Memory protection ("What can be accessed?")
  - Protection bits in segment descriptors
  - Protection bits in page-table registers
  - Virtual Memory (naming)
- File system privileges ("What can be accessed?")
  - User accounts
  - Access permissions

User Authentication

- Verifying that you are who you claim you are.

- File permissions and user’s rights are set according to user’s identity, which is established by authentication.

- Basic Principles. Authentication must identify:
  - Something the user knows
  - Something the user has
  - Something the user is

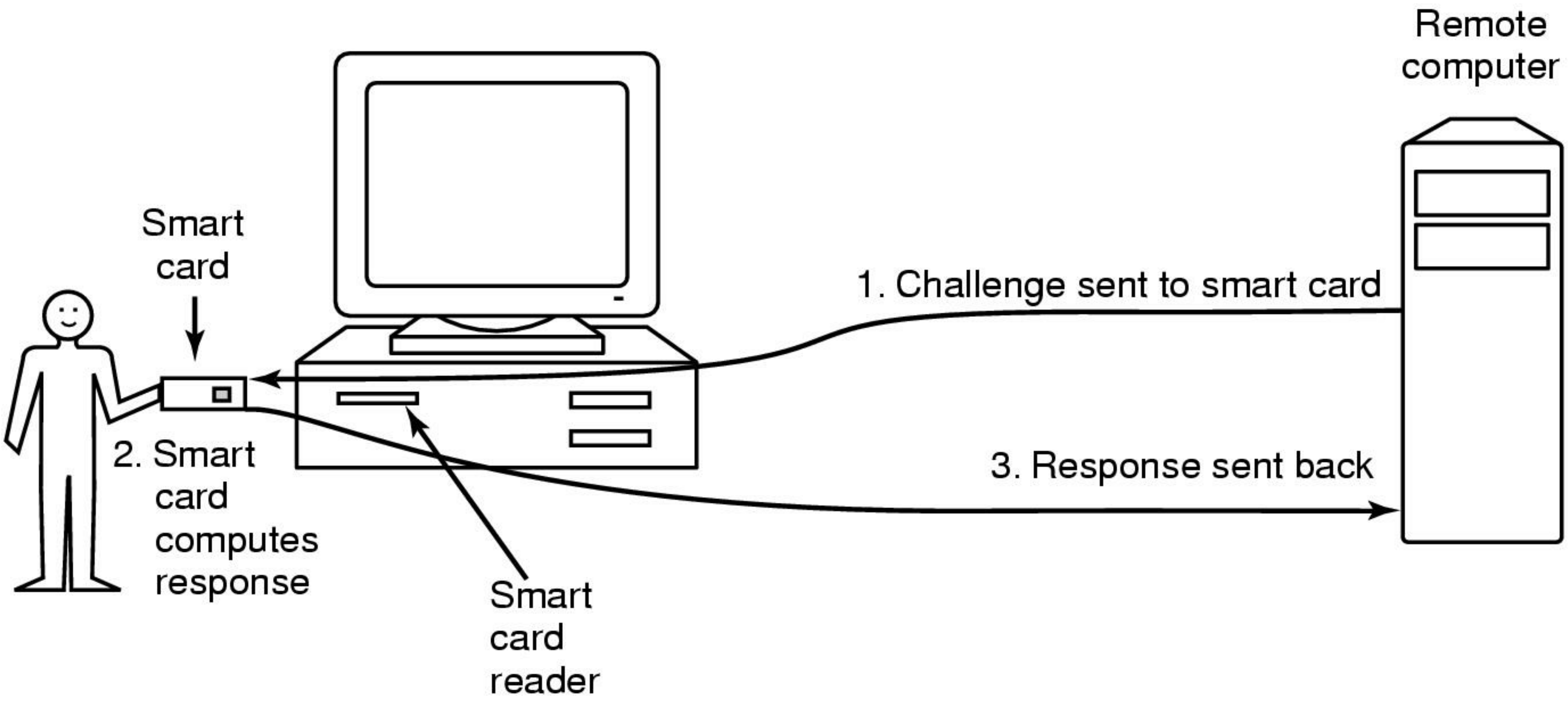
- This is done before user can use the system

Storing passwords

- Originally stored in plaintext in a “secure” file.
  - Secure only as long as root account is not compromised
  - Also, users may not want sysadmins to know their passwords, which usually contain private data.

- Now these are hashed using one-way functions
  - Given password input x
    - easy to evaluate $y = f(x)$
  - But given y
    - computationally infeasible (or at least non-trivial) to compute $x = f^{-1}(y)$

Something the user has: Authentication Using a Physical Object



- – magnetic stripe cards, chip cards: stored value cards, smart cards

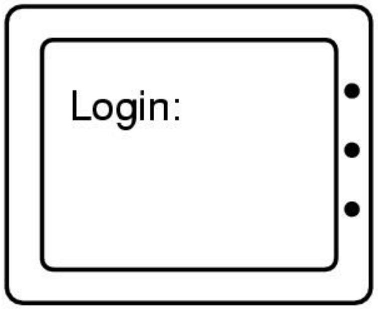
Something the user is: The user’s body

- Biometrics:
  - voice
  - face
  - fingerprint
  - iris scan
  - typing style

- These have both false-positives and false-negatives
- Susceptible to spoofing attacks

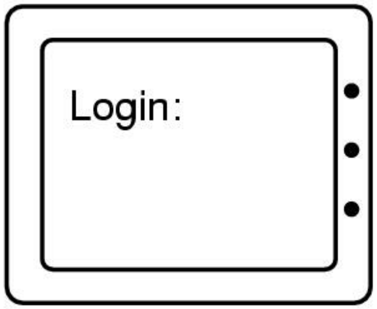
Login Spoofing and Trusted Path

“I’m sure I entered the right password. What happened?”



(a)

Correct login Screen



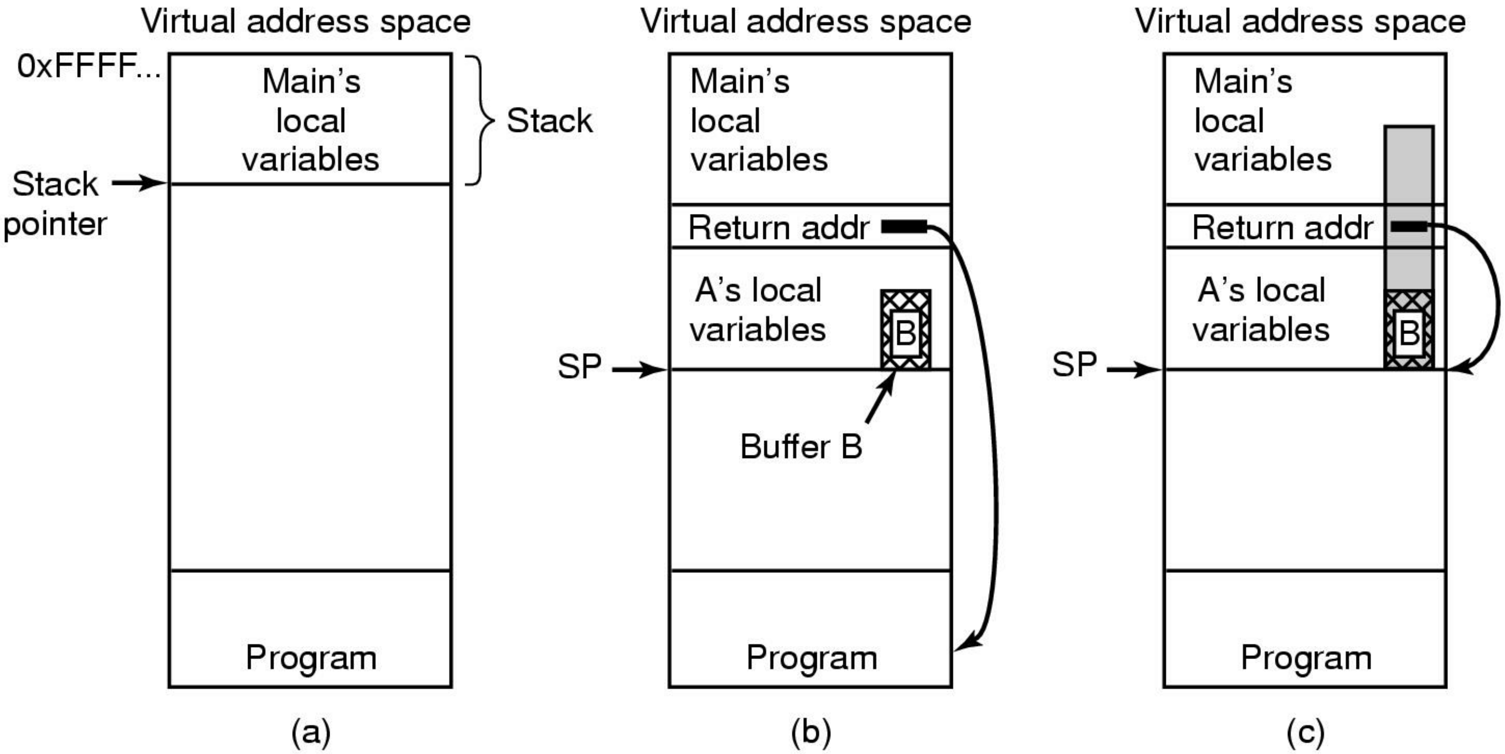
(b)

Phony login Screen

Countermeasures:

- Cautious user can intentionally enter a fake password the first (few) time(s).
- Use “Trusted Path”
  - A sequence of user actions that is guaranteed to give control to the OS.
  - E.g. pressing Ctrl-Alt-Del could guarantee that legitimate login (or logout) screen will show up.

Buffer Overflow



- (a) Situation when main program is running
- (b) After function *A* called
- (c) Buffer overflow shown in gray

Memory reuse — Dumpster Diving

- Request memory, disk space, tapes
- Don’t write. Just read and interpret existing data.
- May find passwords, ssh keys, emails, personal information, browsing history, etc.
- Countermeasure:
  - Scrub memory/storage before allocating to user.
  - Encrypt data. Throw away the key once done.
  - Disadvantage: Takes more time.

Logging

- Logs: A time-wise record of system activity.
  - Events always appended. “Never” erased.
- Logs must be analyzed often to detect suspect activity
- What to log?
  - Too much logging
    - takes up storage
    - slows down normal operations.
    - Slows down analysis.
  - Too little logging and you miss critical events.
- Privacy risk
  - Can break laws.
  - Or violate user’s perception of privacy. (sometimes more important)

Sandboxes

- A restricted execution environment for untrusted programs.
    - Run dowloaded apps in a system VM/process VM/container.
    - Isolate trojans/viruses, worms

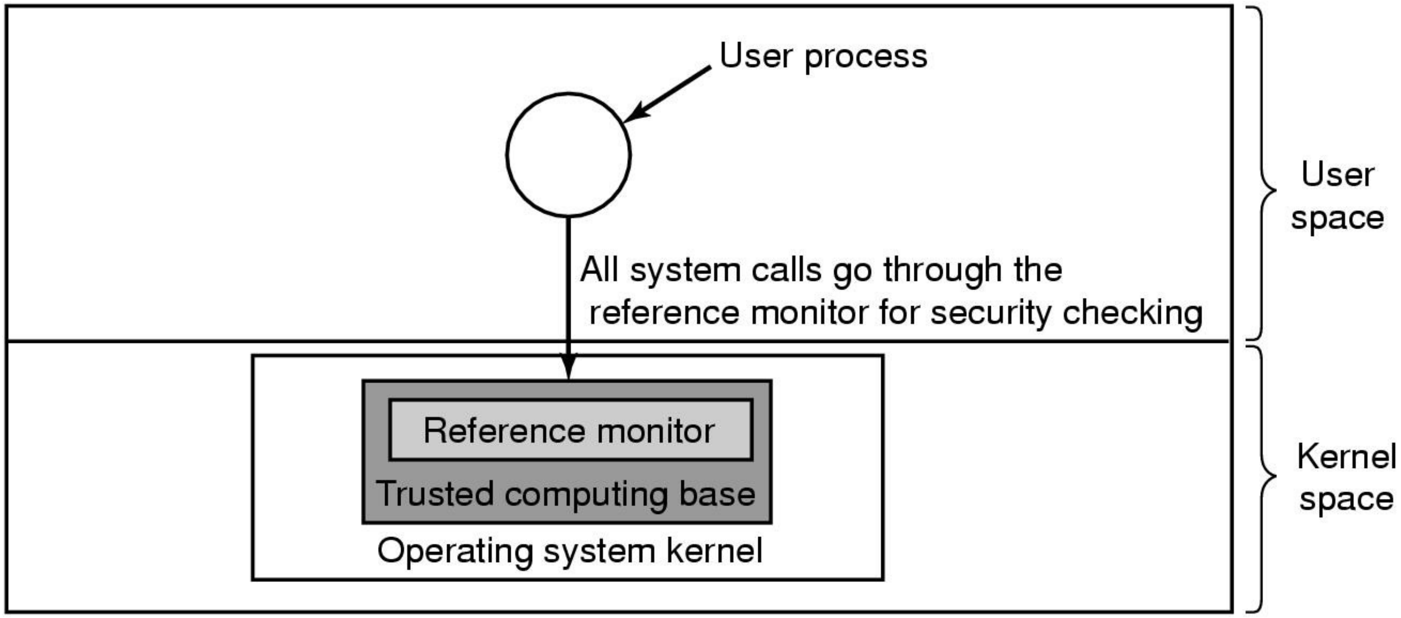
- Effectiveness of isolation only as effective as the implementation of the Sandbox
    - “Gates” must be accompanied by effective “fencing”.

- VM Escapes and Jail-breaks are possible.
    - Usually due to implementation bugs in the hypervisor or runtime

Access control

- **Discretionary access control (DAC)**
  - “John can access X. Alice can do Y.”
  - Commodity systems
- **Mandatory access control (MAC)**
  - Military/spy systems
  - More later
- **Role-based access control (RBAC)**
  - “CEO can do X. Software Engineer can do Y. Secretary can do Z”.
  - Enterprise systems
- **Administrative Role-based Access Control**
  - “Dean can allow department chair to do X. Dept chair can allow secretary to do Y”

Reference Monitor and Trusted Computing Base



- A reference monitor, enforces access control/capabilities.
  - also called “security kernel”
- Its “trusted” because it MUST work correctly to ensure rest of the system is secure.
  - “Trusted” doesn’t automatically mean “secure”.
  - A “Trusted” system means that user has no choice but to assume that the system is secure.
- Reference monitor is usually small, so it can be verified easily.
- Verification can be either manual or automated. Hard to verify either way.

Multi-level Security

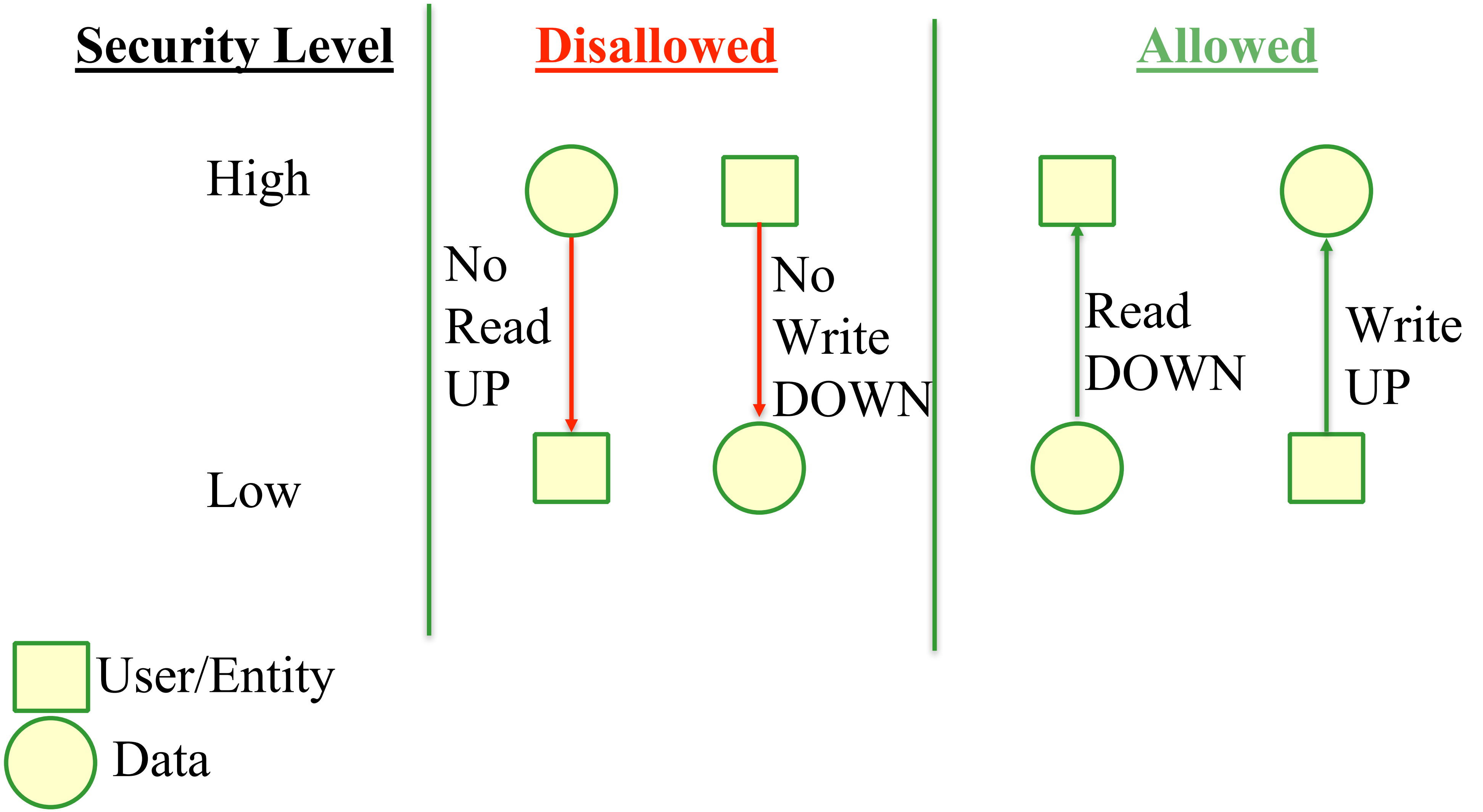
- Also called Mandatory Access Control (MAC)
  - As opposed to Discretionary Access Control (DAC) in commodity systems.

- Data objects are classified at different levels
  - Top secret, secret, confidential, unclassified etc
  - Sometimes additional compartments: Crypto, Subs, NoForn

- People (and computers) have clearances

- Informally: To see a data object, you must have clearance for that level and for that compartment.

MLS: No Read UP, No Write DOWN



Security Level

Disallowed

Allowed

High

Low

No Read UP

No Write DOWN

Read DOWN

Write UP

□ User/Entity

○ Data

No Read UP: Lower classification level should not read data from higher-level.

No Write DOWN: Higher level should not write data to lower level.

MLS Pump

- In practice, to get things done, upper-level must at least acknowledge the receipt of data from lower level.
  - But acks create a backdoor for covert channels (surreptitious communication)

- An MLS Pump
  - Allows acks from higher to lower levels,
  - but at such a low data rate that covert channels become impractical.