

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

Common Motivations of Intruders

1. Peeping Tom
 - Casual prying by nontechnical users
2. Insider threat
 - Disgruntled insiders, programmer's backdoor
3. Extortion
 - Make money
4. Espionage/Intelligence gathering
 - Commercial or military or government
5. Hacktivism
 - Political or social motivation
6. Sometimes motivations may overlap
 - Was Snowden incident 2? 4? 5? All?

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

Something the user knows: Passwords

LOGIN: ken
PASSWORD: FooBar
SUCCESSFUL LOGIN

(a)

LOGIN: carol
INVALID LOGIN NAME
LOGIN:

(b)

LOGIN: carol
PASSWORD: Idunno
INVALID LOGIN
LOGIN:

(c)

(a) A successful login

(b) Login rejected after name entered

(c) Login rejected after name and password typed

Storing passwords

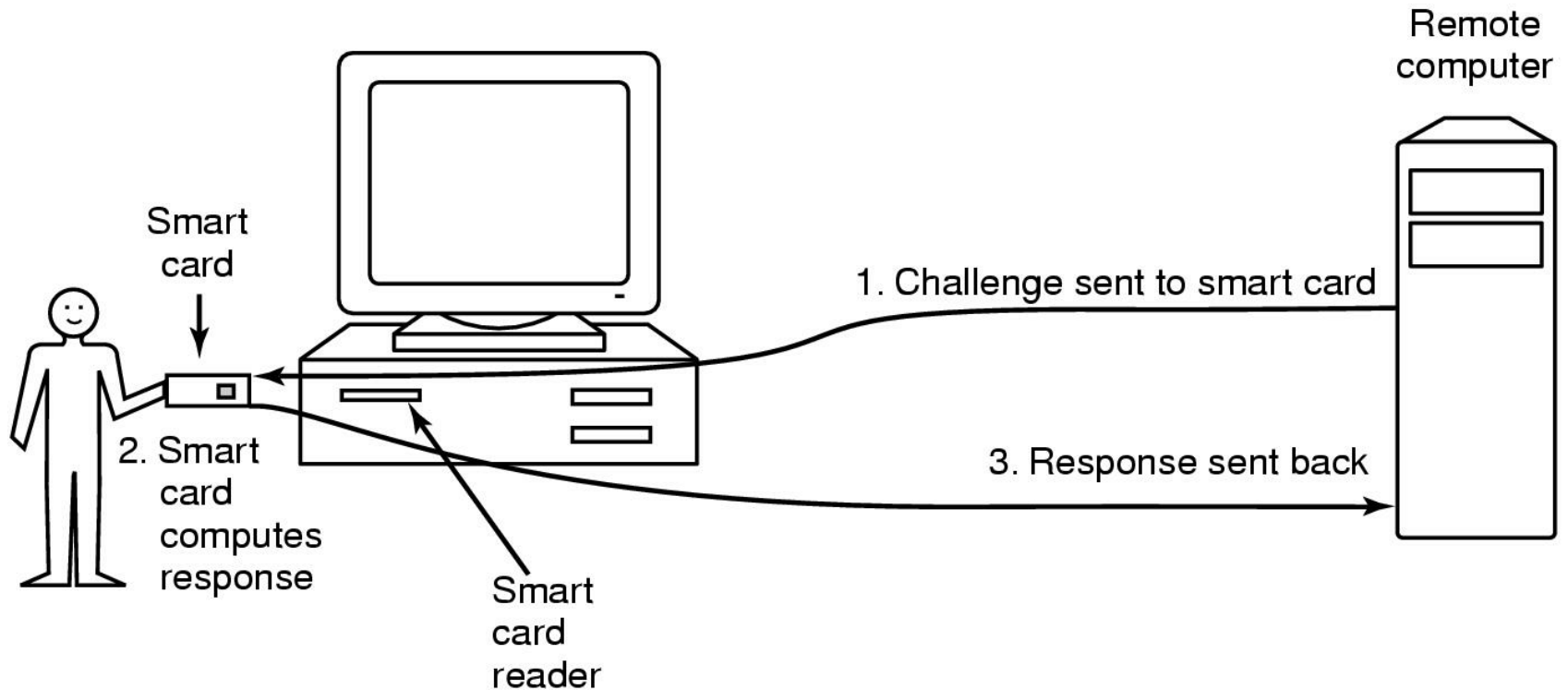
- Originally stored in plaintext in a “secure” file.
 - Secure only as long as root account is not compromised
 - Also, users may not want sysadmins to know their passwords, which usually contain private data.
- Now these are hashed using one-way functions
 - Given password input x
 - easy to evaluate $y = f(x)$
 - But given y
 - computationally infeasible (or at least non-trivial) to compute $x = f^{-1}(y)$

Challenge-Response Authentication

- Forgot password?
- Ask user something that no one else would know.
 - Poor choices: Mother's maiden name, where you were born, first girlfriend/boyfriend, pet's name, high school, childhood street etc.
 - All easily guessed
 - Not sure why websites still do this
- Ask user to compute something
 - “What is the fifth smallest prime number?”
 - Assuming the question can't be understood by a program.
- Attack: Beg customer service to help a poor user who forgot password

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

Countermeasures against authentication attacks

- Limiting times when someone can log in
 - “Sorry: You can’t log in at 2am”
- Limited number of login tries
 - “Too many invalid logins. Your account is now LOCKED.”
- Two-factor authentication
 - Password + Automatic callback/SMS at number prespecified
- Logging: Tracking all logins and locations of login
 - “Your last login: from Timbuktu yesterday.”
- Ask user to recognize text in a figure
 - CAPTCHA = Completely Automated Public Turing test to tell Computers and Humans Apart
- Honeypot accounts: Simple login name/password as a trap
 - security personnel notified when attacker bites
 - Only if you want to track/catch the intruder (sometimes don’t care)

Common Attacks and Countermeasures

Trojan Horses

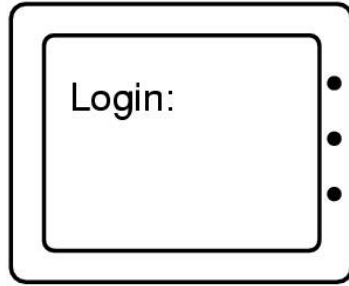
- Malicious email attachments
 - CM: Don't open. Open in a VM. Use a cloud-based reader.
- Malicious websites that exploit browser vulnerabilities
 - Visit and get hacked
 - CM: Turn off Flash plugins, Javascript. Affects usability.
- “Free” program made available to unsuspecting user
 - Actually contains code to do harm
 - CM: Run in VM. Don't download.
- Place altered version of utility program on victim's computer
 - Trick user into running that program
 - CM: Administrator must strictly control file permissions

Virus and Worm

- Virus
 - program that can reproduce itself by attaching its code to another program
 - requires human intervention to spread to another machine
- Worms
 - spread across machines
 - automatically, or with human assistance

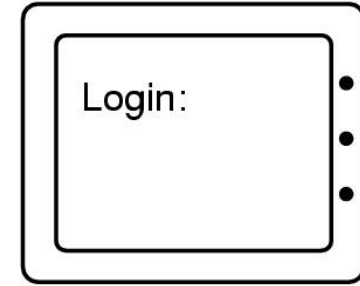
Login Spoofing

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



(a)

Correct login Screen



(b)

Phony login Screen

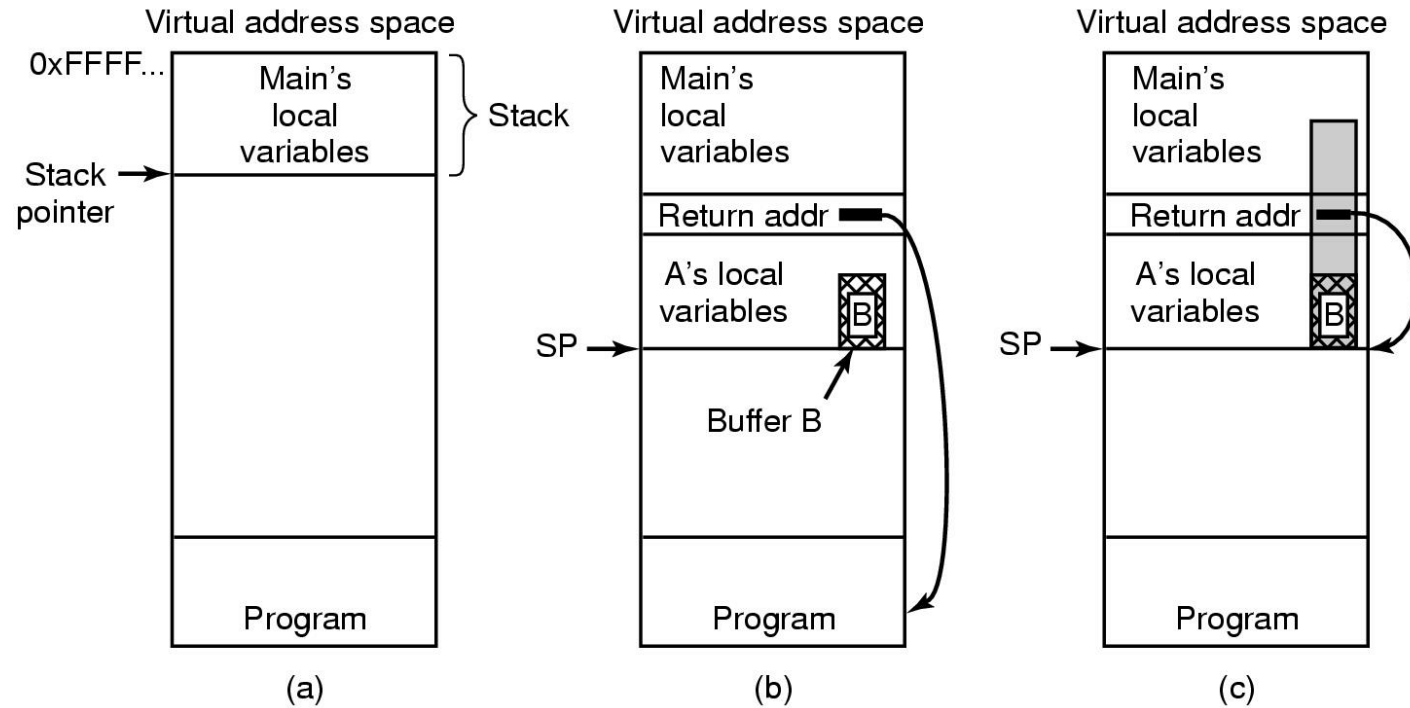
Countermeasures:

- 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.

Logic Bombs

- Company programmer writes a program with potential to do harm
- OK as long as he/she enters password daily
- If programmer fired, no password and bomb “explodes”
- CM: Log all activity
 - Easy to detect and correlate.
- Don’t try this at work!
 - Your employer may be smarter than you give them credit for.

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.
- CM:
 - 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.

Other ways to gain access

- Trying privileged system calls to see what happens
- Doing specified DO NOTs
 - “Only authorized personnel beyond this point”
- Convince a system programmer to add a backdoor

Design Principles for Security

- Default should be no access
- Check for current authority
- Give each process least privilege possible
- Protection mechanism should be
 - simple
 - uniform
 - in lowest layers of system
- Scheme should be simple and psychologically acceptable
 - If its too hard, users will get around it.
 - Like using post-it notes on the monitor.

Sandboxes

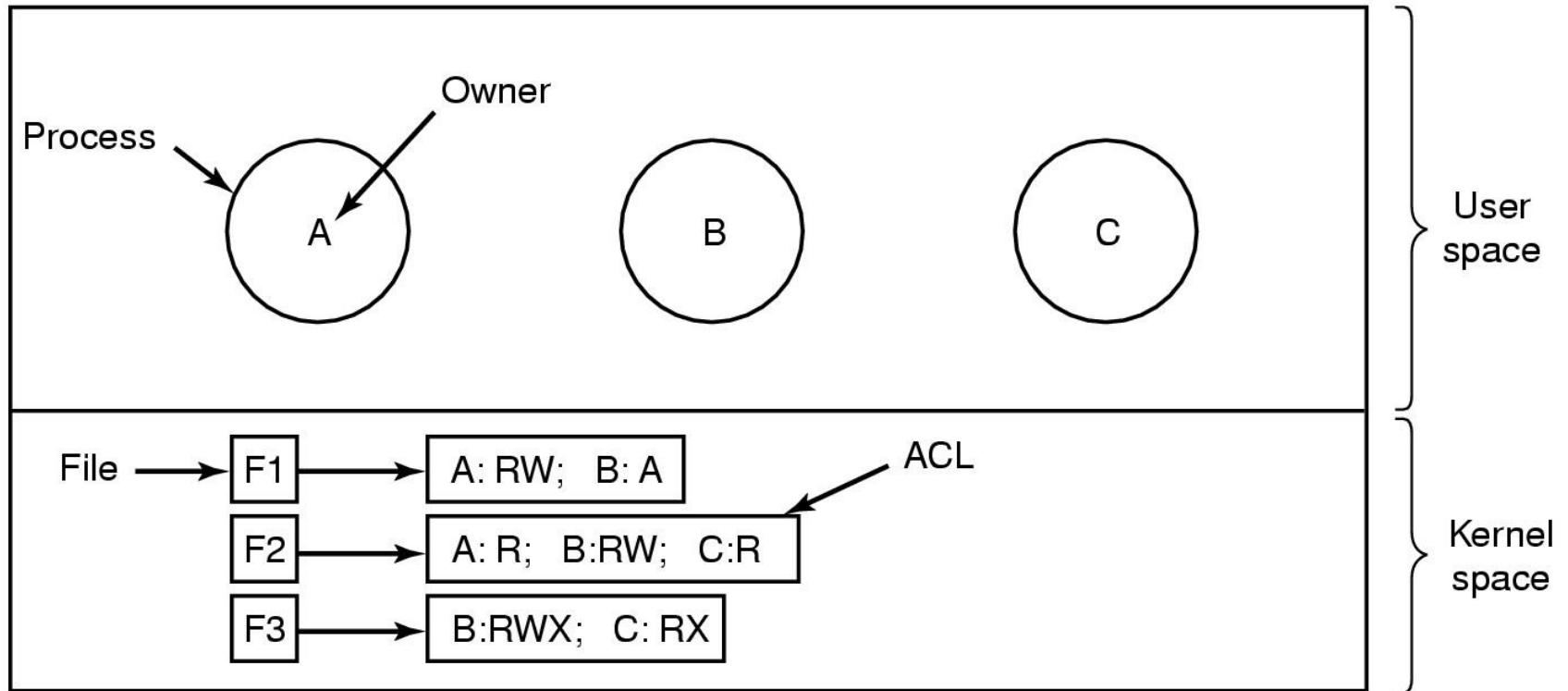
- Run downloaded code/browser in a VM or a “Jail”.
- Isolate trojans/viruses, worms
- Effectiveness of isolation only as effective as the security of the Sandbox.
- VM Escapes and Jail-breaks are possible.
 - Usually due to implementation bugs in the hypervisor or runtime

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”

Access Control Lists

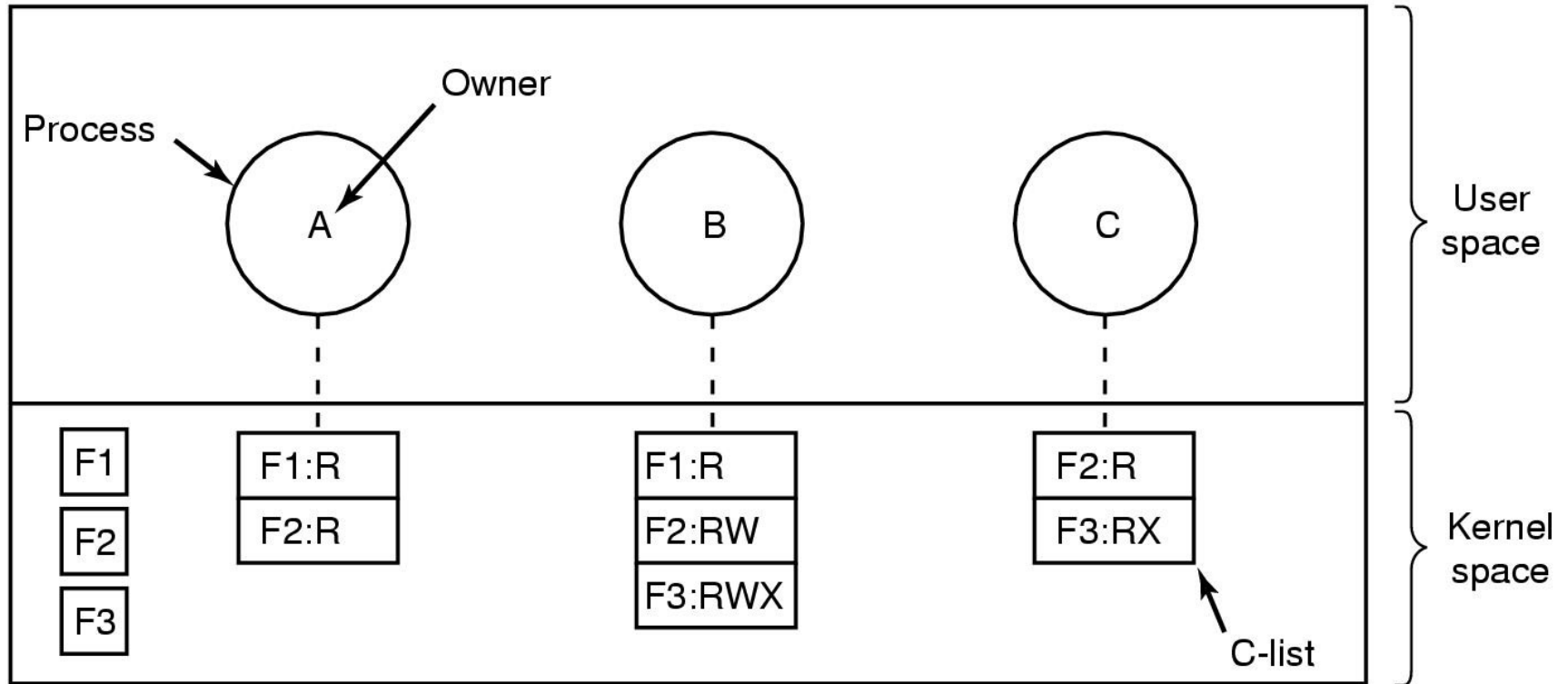
“Who can access what”



Use of access control lists of manage file access

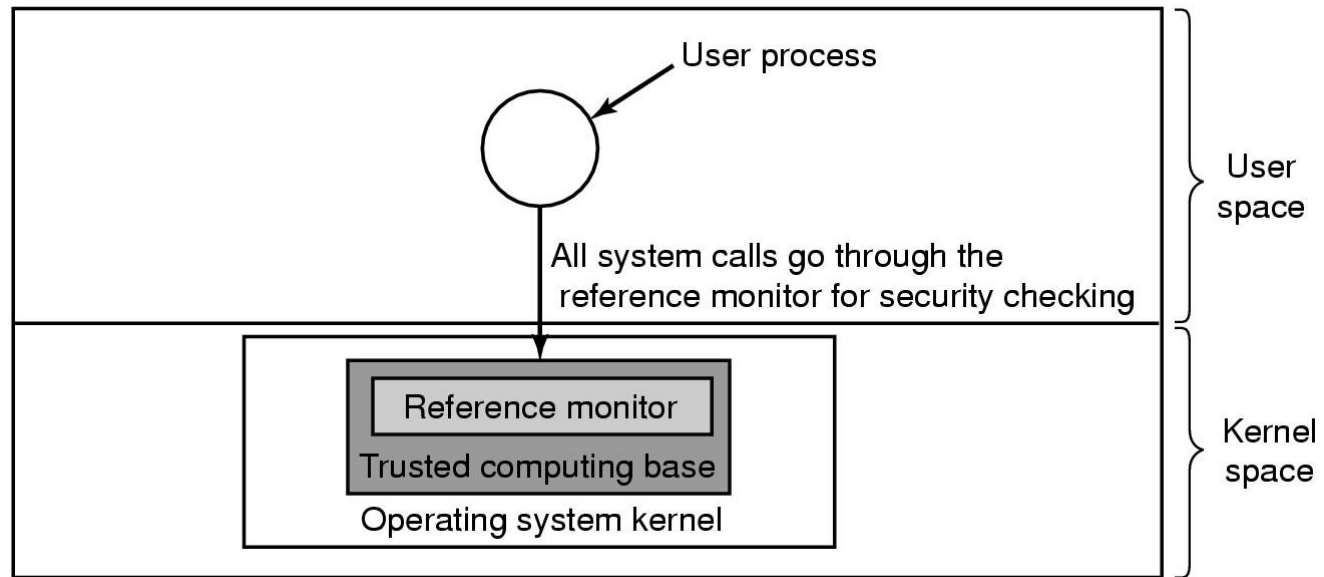
Capabilities

“Who can do what”



Each process has a capability list

Reference Monitor and Trusted Computing Base

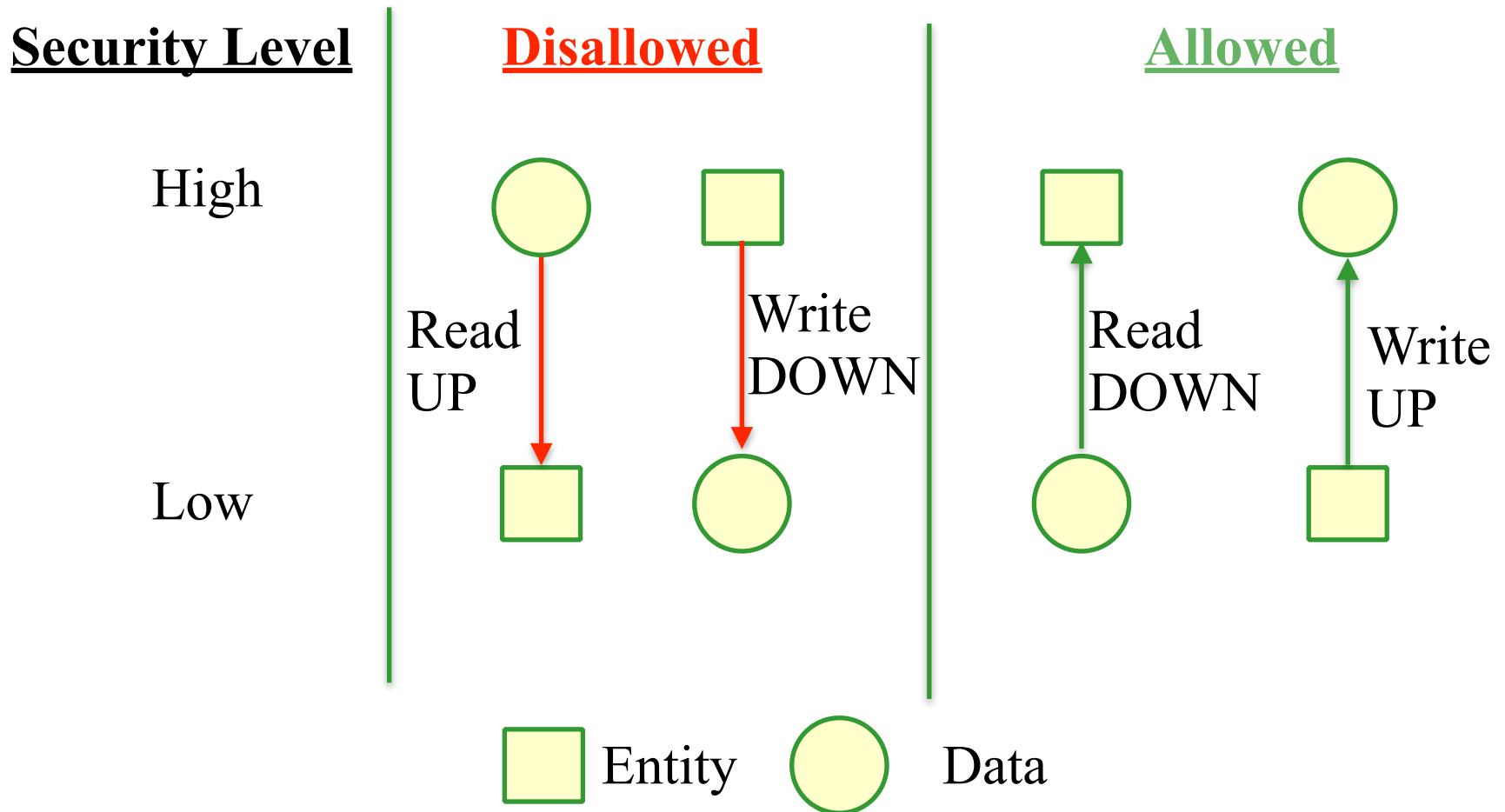


- A reference monitor, enforces access control/capabilities.
 - also called “security kernel”
- Its “trusted” because it MUST work correctly to ensure rest of the system is secure.
- Usually small, so it can be verified easily.
- Verification: either manual or automated. Hard either way.

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



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.