Operating System and Security

Kartik Gopalan



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

What is Security

• C.I.A

Goal

Data confidentiality

Data integrity

System availability

- 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)

Threat
Exposure of data
Tampering with data
Denial of service

Securing what? From what?

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



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

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?") \bullet
 - 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 \bullet

Something the user knows: Passwords

LOGIN: ken PASSWORD: FooBar SUCCESSFUL LOGIN

(a)

(a) A successful login(b) Login rejected after name entered(c) Login rejected after name and password typed

LOGIN: carol INVALID LOGIN NAME LOGIN:

(b)

LOGIN: carol PASSWORD: Idunno INVALID LOGIN LOGIN: (c)

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 might 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? lacksquare
- Ask user something that no one else would know. ullet
 - \bullet pet's name, high school, childhood street etc.
 - All easily guessed ullet
 - Not sure why websites still do this lacksquare
- Ask user to compute something lacksquare
 - "What is the fifth smallest prime number?" \bullet
 - Assuming the question can't be understood by a program. lacksquare
- Attack: Beg customer service to help a poor user who forgot password \bullet
 - Examples: Former CIA director, Twitter CEO \bullet

Poor choices: Mother's maiden name, where you were born, first girlfriend/boyfriend,

Something the user has:



– 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

itives and false-negatives tacks

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

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

es I human assistance

Login Spoofing "I'm sure I entered the right password. What happened?"



(a) Correct login Screen

Countermeasures:

- time(s).
- Use "Trusted Path"

 - logout) screen will show up.



Careful user can intentionally enter a fake password the first (few)

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

Logic Bombs

- to do harm
- OK as long as he/she enters password daily
- CM: Log all activity
 - Easy to detect and correlate.
- Don't try this at work!
 - credit for.

• Company programmer writes a program with potential

• If programmer fired, no password and bomb "explodes"

• Your employer may be smarter than you give them

Buffer Overflow



- \bullet
- (b) After function A called
- (c) Buffer overflow shown in gray

(a) Situation when main program is running

- 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.

Memory reuse — Dumpster Diving

Logging

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

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 necessary • 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 dowloaded 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"

AC) do Y."

 \mathbf{C}

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.



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
 - but at such a low data r become impractical.