Security Properties

slides stolen from David Walker (Princeton) and Andrew Myers (Cornell)

Outline

- What is computer security?
  - Protecting against worms and viruses?
  - Making sure programs obey their specifications?
  - Still plenty of security problems even if these problems are solved…

Acknowledgments: Andrew Myers, Steve Zdancewic, Fred Schneider

What is security?

- Security: prevent bad things from happening
  - Confidential information leaked
  - Important information damaged
  - Critical services unavailable
  - Clients not paying for services
  - Money stolen
  - Improper access to physical resources
  - System used to violate law
  - Loss of *value*

… or at least make them less likely

- Versus an adversary!

Attack Sampler #1: Morris Worm

1988: Penetrated an estimated 5 to 10 percent of the 6,000 machines on the internet.

Used a number of clever methods to gain access to a host.

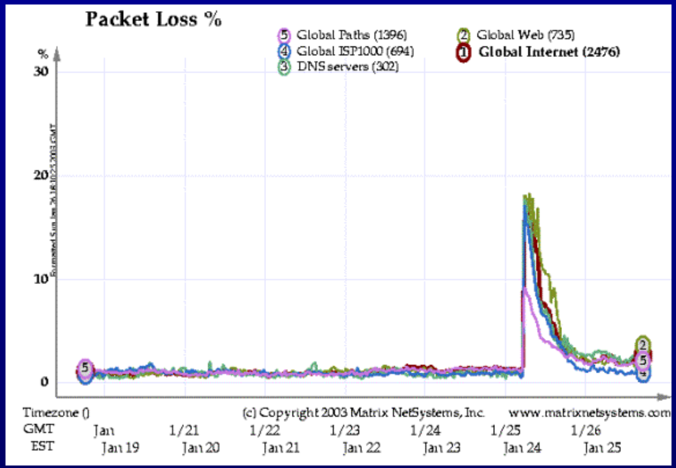
- brute force password guessing
- bug in default sendmail configuration
- X windows vulnerabilities, rlogin, etc.
- buffer overrun in fingerd

Remarks:

- System diversity helped to limit the spread.
- “root kits” for cracking modern systems are easily available and largely use the same techniques.

2002: MS-SQL Slammer worm

- Jan. 25, 2002: SQL and MSDE servers on Internet turned into worm broadcasters
  - YABO
  - Spread to most vulnerable servers on the Internet in less than 10 min!



- Denial of Service attack
  - Affected databases unavailable
  - Full-bandwidth network load ⇒ widespread service outage
  - “Worst attack ever” – brought down many sites, not Internet
- Can’t rely on patching!
  - Infected SQL servers at Microsoft itself
  - Owners of most MSDE systems didn’t know they were running it…support for extensibility

Attack sampler #2: Love Bug, Melissa

- 1999: Two email-based viruses that exploited:
  - a common mail client (MS Outlook)
  - trusting (i.e., uneducated) users
  - VB scripting extensions within messages to:
    - look up addresses in the contacts database
    - send a copy of the message to those contacts
- Melissa: hit an estimated 1.2 million machines.
- Love Bug: caused estimated $10B in damage.
- Remarks:
  - no passwords, crypto, or native code involved

Attack sampler #3: Hotmail

- 1999: All Hotmail email accounts fully accessible by anyone, without a password
- Just change username in an access URL (no programming required!)
- Selected other Hotmail headlines (1998-99)

  Hotmail bug allows password theft

  Hotmail bug pops up with JavaScript code

  Malicious hacker steals Hotmail passwords

  New security glitch for Hotmail

  Hotmail bug fix not a cure-all

Attack sampler #4: Yorktown

- 1998: “Smart Ship” USS Yorktown suffers propulsion system failure, is towed into Norfolk Naval Base
- Cause: computer operator accidentally types a zero, causing divide-by-zero error that overflows database and crashes all consoles
- Problem fixed two days later

Attack sampler #5: insiders

- Average cost of an outsider penetration is $56,000; average insider attack cost a company $2.7 million (Computer Security Institute/FBI)
- 63 percent of the companies surveyed reported insider misuse of their organization's computer systems. (WarRoom Research)
- Some attacks:
  - Backdoors
  - “Logic bombs”
  - Holding data hostage with encryption
  - Reprogramming cash flows
- Attacks may use legitimately held privileges!
- Many attacks (90%?) go unreported

More attack samplers & vulnerabilities

- Take a look at
  - Attacking Malicious Code: A report to the Infosec Research Council.
    Gary McGraw and Greg Morrisett.
  - A list of recent online attacks and defenses
- Take a look at
  - Trust in Cyberspace Fred Schneider editor
  - state of security and vulnerability in power grid and communications infrastructure

Terminology

- Vulnerability

  Weakness that can be exploited in a system

- Attack

  Method for exploiting vulnerability

- Threat / Threat model

  The power of the attacker (characterizes possible attacks)

  - E.g., attacker can act as an ordinary user, read any data on disk, and monitor all network traffic.

- Trusted Computing Base

  Set of system components that are depended on for security

  - Usually includes hardware, OS, some software, …

Who are the attackers?

- Operator/user blunders.
- Hackers driven by intellectual challenge (or boredom).
- Insiders: employees or customers seeking revenge or gain
- Criminals seeking financial gain.
- Organized crime seeking gain or hiding criminal activities.
- Organized terrorist groups or nation states trying to influence national policy.
- Foreign agents seeking information for economic, political, or military purposes.
- Tactical countermeasures intended to disrupt military capability.
- Large organized terrorist groups or nation-states intent on overthrowing the US government.

What are the vulnerabilities?

- Poorly chosen passwords
- Software bugs
  - unchecked array access (buffer overflow attacks)
- Automatically running active content: macros, scripts, Java programs
- Open ports: telnet, mail
- Incorrect configuration
  - file permissions
  - administrative privileges
- Untrained users/system administrators
- Trap doors (intentional security holes)
- Unencrypted communication
- Limited Resources (i.e. TCP connections)

What are the attacks?

- Password Crackers
- Viruses:
  - ILoveYou (VBscript virus), Melissa (Word macro virus)
- Worms
  - Code Red: Port 80 (HTTP), Buffer overflow in IIS (Internet/Indexing Service)
- Trojan Horses
- Social Engineering:
  - “Hi, this is Joe from systems, I need your password to do an upgrade”
- Packet sniffers: Ethereal
- Denial of service: TCP SYN packet floods

Social engineering attacks

E-mail this to a friend   Printable version

Passwords revealed by sweet deal

More than 70% of people would reveal their computer password in exchange for a bar of chocolate, a survey has found.

Security crumbles in the face of sweet bribes

It also showed that 34% of respondents volunteered their password when asked without even needing to be bribed.

A second survey found that 79% of people unwittingly gave away information that could be used to steal their identity when questioned.

Security firms predict that the lax security practices will fuel a British boom in online identity theft.

Security vs. fault tolerance

- Attacks vs. faults
- Reliability community often assumes benign, random faults
  - Failstop failures = system halts
  - Byzantine failure = system behaves arbitrarily badly (under control of adversary)
- Attackers go for the weakest link!
  - It doesn't help to have a $1000 lock on your door if the window is open.

Assumptions and abstraction

- Arguments for security always rest on assumptions:
  - “the attacker does not have physical access to the hardware”
  - “the code of the program cannot be modified during execution”
- Assumptions are vulnerabilities
  - Sometimes known, sometimes not
- Assumptions arise from **abstraction**
  - security analysis only tractable on a simplification (abstraction) of actual system
  - Abstraction hides details (assumption: unimportant)

Risk management

- Cost-benefit: high security may be more expensive than benefits obtained
  - Security measures interfere with intended use

security ⟷ functionality / efficiency / cost

- Preventing problems may be infeasible, unnecessary; deterrence may be sufficient
  - Remove the incentive to attack
  - Make it easier to attack someone else
  - Make it too costly to attack

When to enforce security

Possible times to respond to security violations:

- Before execution:
  analyze, reject, rewrite
- During execution:
  monitor, log, halt, change
- After execution:
  roll back, restore, audit, sue, call police

Policy vs. mechanism

- What is being protected (and from what) vs.
- How it is being protected

  (access control, cryptography, …)

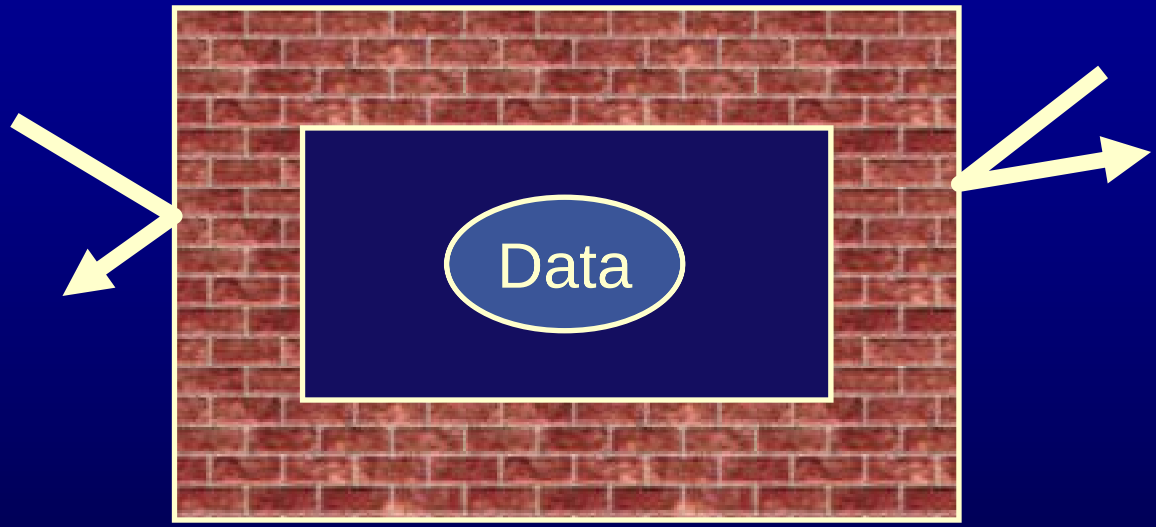
- Want:
  - To know what we need to be protected from
  - Mechanisms that can implement many policies

What is being protected?

- Something with value
- Information with (usually indirect) impact on real world
- Different kinds of protection are needed for different information : ensure different **security properties**
  - **Confidentiality**
  - **Integrity**
  - **Availability**

Properties: Integrity

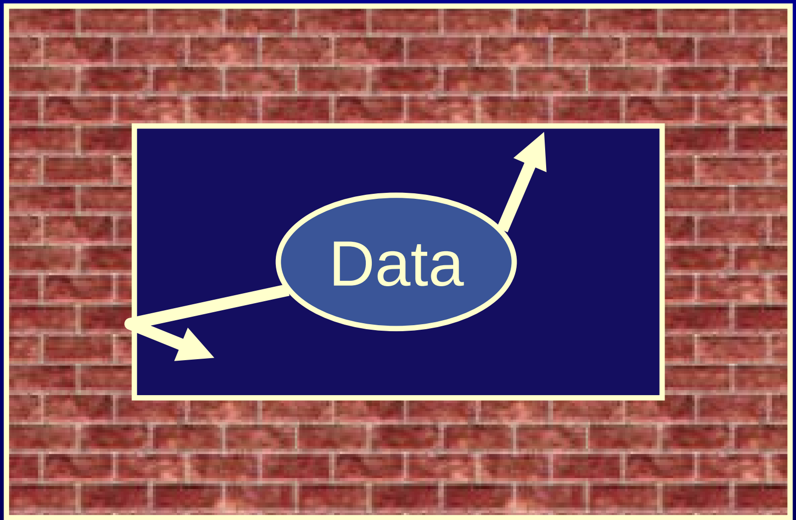
- No improper modification of data



- E.g., account balance is updated only by authorized transactions, only you can change your password
- Integrity of security mechanisms is crucial
- Enforcement: access control, digital signatures,…

Properties: Confidentiality

- Protect information from improper release



- Limit knowledge of data or actions
- E.g. D-Day attack date, contract bids
- Also: secrecy
- Enforcement: access control, encryption,…
- Hard to enforce after the fact…

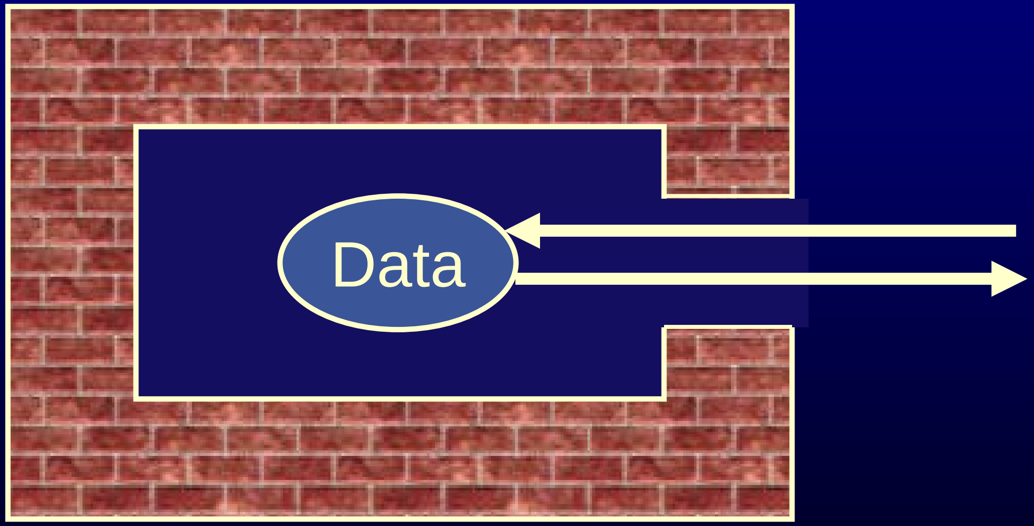
Properties: Privacy, anonymity

- Related to confidentiality
- Privacy: prevent misuse of personal information
- Anonymity: prevent connection from being made between identity of actor and actions
  - Keep identity secret
  - Keep actions secret

Properties: Availability

- Easy way to ensure confidentiality, integrity: unplug computer

- Availability: system must respond to requests



Properties: Nonrepudiation

- Ability to convince a third party that an event occurred (e.g., sales receipt)
- Needed for external enforcement mechanisms (e.g., police)
- Related to integrity

Properties: Safety

- “Nothing bad ever happens” (at a particular moment in time)
- A property that can be enforced using only history of program
- Amenable to purely run-time enforcement
- Examples:
  - access control (e.g. checking file permissions on file open)
  - memory safety (process does not read/write outside its own memory space)
  - type safety (data accessed in accordance with type)

Properties: Liveness

- "Something good eventually happens"
- Example: availability
  - "The email server will always respond to mail requests in less than one second"
- Violated by denial of service attacks
- Can't enforce purely at run time – stopping the program violates the property!
- Tactic: restrict to a safety property
  - "web server will respond to page requests in less than 10 sec or report that it is overloaded."

Security Property Landscape

"System does exactly what it should--and no more"

Privacy

Digital rights

Noninterference (confidentiality, integrity)

Mandatory access control

Byzantine Fault Tolerance

Discretionary access control

Reference confinement

Fault Tolerance

Type safety

Memory safety

Availability

Memory protection

Safety properties

Liveness properties

Security Mechanisms

slides stolen from David Walker (Princeton) and Andrew Myers (Cornell)

Topics

- Fundamental enforcement mechanisms
- Design principles for secure systems
- Operating system security mechanisms

Mechanisms: Authentication

- If system attempts to perform action X, should it be allowed? (e.g., read a file)
  - authentication + authorization

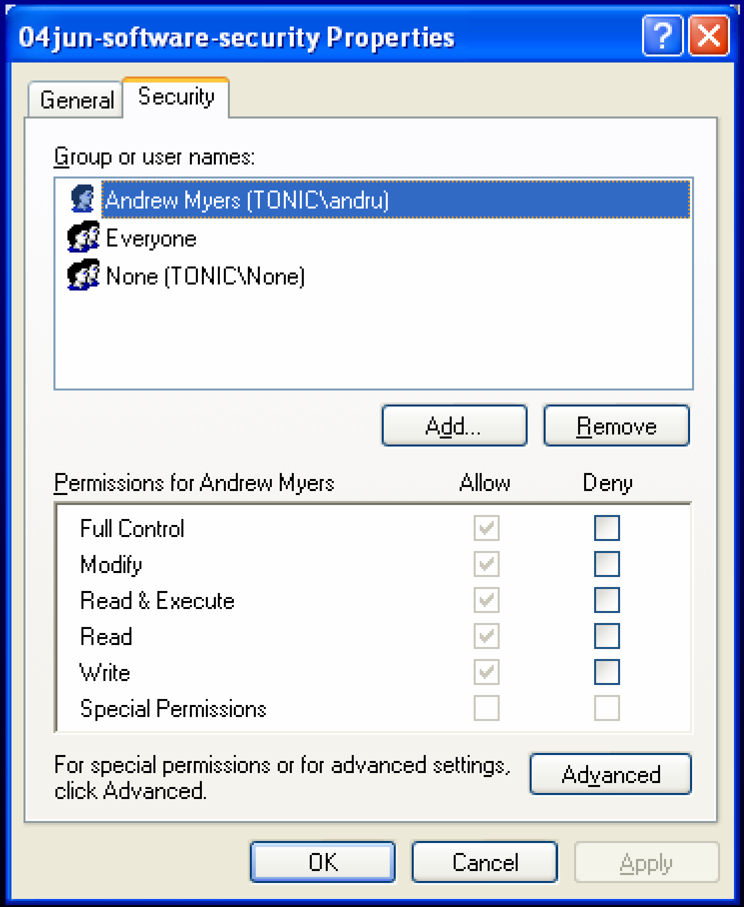
- **Authentication:** what principal p is system acting on behalf of? Is this an authentic request from p?
  - Passwords, biometrics, certificates…

Principals

- A principal is an identity; an abstraction of privileges
  - Process uid
  - E.g., a user (Bob), a group of users (Model airplane club), a role (Bob acting as president)

Mechanisms: Authorization

- **Authorization:** is principal p authorized to perform action A?
- Access control mediates actions by principals
- Example: file permissions (ACLs)



Mechanisms: Auditing

- For after-the-fact enforcement, need to know what happened: auditing
- Audit log records security-relevant actions (who, what, when)
- Authorization + Authentication + Audit = “The gold (Au) standard” : classic systems security

- A fourth kind of mechanism: program analysis and verification
  - Needed for extensible systems and strong security properties… more later

Principle: Complete Mediation

- Common requirement: system must have ability to mediate all security-relevant operations
  - Dangerous to assume op is not security-relevant..
  - Many places to mediate: hardware, compiler, …
- Assumption: mediation mechanism cannot be compromised (TCB)
- Example: operating system calls
  - Kernel interface mediates access to files, memory pages, etc.
  - No other way to create/manipulate resources
  - One problem: covert timing channels

Principle: Minimize TCB

- The trusted computing base (TCB): those portions of the system that absolutely must be correct in order for the system to be "secure"
  - Observation: Complex things are more likely not to work correctly
  - Consequence: the TCB should be as small as possible
  - Consequence: Economy of Mechanism – don't use three mechanisms when one will do

"Things should be made as simple as possible–but no simpler."
-- A. Einstein

Principle: Failsafe Defaults

- One of the most common programming or design errors is “forgetting to handle a case”
  - as systems get larger, it becomes easier and easier to forget to consider one element of the design
  - If access is off by default, “forgetting a case” results in denial of service
    - detected quickly by legitimate users and corrected
  - If access is on by default, “forgetting a case” results in the possibility of illegitimate access
    - not detected by legitimate users and lingers as security vulnerability

Principle: Least Privilege

- A principal should be given only those privileges needed to accomplish its tasks.
  - No more, no less.
- What is the minimal set of privileges?
- What is the granularity of privileges?
  - Separation of privileges (read vs. write access)
- How & when do the privileges change?

- Example violation: UNIX sendmail has root privilege

Principle: Open Design

- Success of mechanism should not depend on it being secret
  - “No security through obscurity”
  - Inevitably, the secret gets out
  - Insiders will know the secret
  - Increased assurance if many critics.
  - Some form of secret is necessary. Make these secrets replaceable data rather than the algorithm itself. eg: cryptographic keys
- An age-old controversy:
  - Open design makes critics’ jobs easier, but also attackers’ job.
  - Analysis tends to concentrate on core functionality; vulnerabilities remain off the beaten path. (Ergo: small TCB)
  - Sometimes there are economic reasons to keep secrets

Principle: Security is a Process

- Every system has vulnerabilities
  - Impossible to eliminate all of them
  - Goal: assurance
- Systems change over time
  - Security requirements change over time
  - Context of mechanisms changes over time
- Secure systems require maintenance
  - Check for defunct users
  - Update virus software
  - Patch security holes
  - Test firewalls