

AN INTRODUCTION TO COMPUTER SECURITY

Richard A. Kemmerer
Computer Science Department
University of California
Santa Barbara, California, U.S.A.

Email: kemm@cs.ucsb.edu

Overview of Security

CS177 2011

1

Computer Security

- What is computer security?
 - The field of computer science that analyzes the security properties of computer systems
 - The protection of resources (including data and programs) from accidental or malicious modification, destruction, or disclosure
- Why is it important?
 - Information is power and money
 - Computer systems manage information and provide mission-critical support for business, government, and financial institutions

Overview of Security

CS177 2011

2

How did we get here?

- Computer security has existed as long as computer systems existed
 - Alan Turing was part of the group that worked on breaking the Nazi's encryption algorithm, Enigma, using automated methods
- Every time a new system is introduced new security issues are introduced as well

Overview of Security

CS177 2011

3

How did we get here?

- Today's uber-connected, ubiquitous computing exacerbates the problem
 - Botnets, D-Dos, Viruses, Spoofing, Hackers, Spam, Scams, Brute-forcing, Trojan Horses, Frauds, Rootkits, Web Attacks, Overflow, Scanners, Hijacking, Social Engineering, Password Cracking, Phishing, Eavesdropping, Man-in-the-middle attacks, Identity Theft, Blackmail...

Overview of Security

CS177 2011

4

How bad is it?

September 2001 - Nimda worm spread nationwide in less than an hour and attacked 86,000 computers

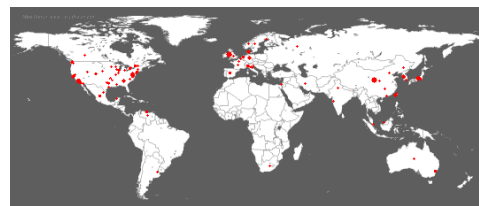
January 2003 - Sapphire/Slammer SQL worm was able to spread nationwide in less than 10 minutes, doubling in size every 8.5 seconds. At its peak (3 minutes after its release) it scanned at over 55 million IP addresses per second, infecting 75,000 victims

Overview of Security

CS177 2011

5

How bad is it?



Thu Jul 19 00:00:00 2001 (UTC)
Victims: 159
<http://www.caida.org/>
Copyright (C) 2001 UC Regents, Jeff Brown for CIDA/UCSD

Geographic spread of Code Red worm

Overview of Security

CS177 2011

6

Why is it so bad?

Computers are everywhere
Computer systems constantly grow in complexity (and size)
Today's networks are very heterogeneous, and critical components are often connected (maybe in indirect ways) to non-critical, poorly managed computer systems
People make mistakes in both the development and the deployment of computer systems

Overview of Security

CS177 2011

7

Why is it so bad?

Home Users Increase Vulnerabilities

Today most homes are connected, particularly with the advent of DSL and cable modems

Most home users:

- are unaware of vulnerabilities
- don't use firewalls
- think they have nothing to hide or don't care if others get their data
- don't realize their systems can serve as jump off points for other attacks (*zombies or bots*)

Overview of Security

CS177 2011

8

Why is it so bad?

Computer security is reactive

- usually reacting to latest attack
- offense is easier than defense

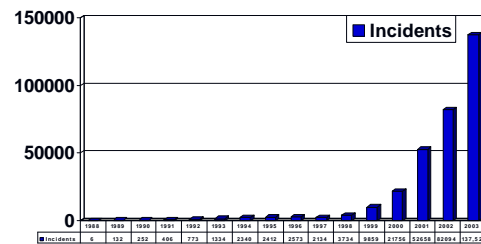
Security is expensive both in dollars and in time
There is not now, and never will be, a system with perfect security

Overview of Security

CS177 2011

9

Security Incidents



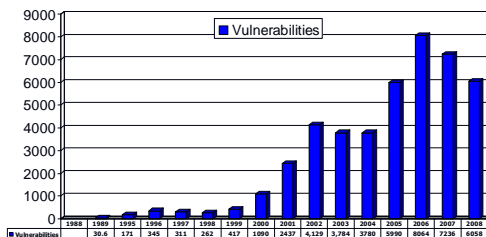
Source: CERT

Overview of Security

CS177 2011

10

Security Vulnerabilities



Source: CERT
<http://www.cert.org/stats/>

Overview of Security

CS177 2011

11

Who are the attackers?

Script kiddies download malicious software from hacker web sites
Hackers trying to prove to their peers that they can compromise a specific system
Insiders are legitimate system users who access data that they have no rights to access
Organizational level attackers use the full resources of the organization to attack

Overview of Security

CS177 2011

12

Who are the attackers?

After September 11, 2001 the idea of nation state level cyber attacks being carried out by terrorists became a big concern

More recently, most attacks are financially motivated. There is a complete cyber underground economy

Outline

Examples of known security threats

Classification of security threats

Security policies

Protection mechanisms

Techniques for assuring system security

Most Common Threat Password Guessing

- Exhaustive search for passwords
- Lists of commonly used passwords
- Distributed default passwords
- Password cracking programs readily available on the Internet

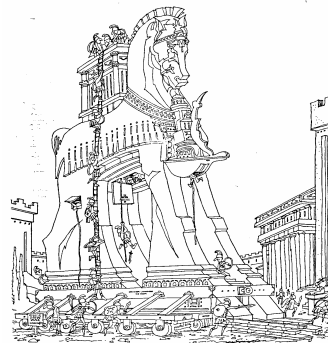
Spoofing

Duping a user into believing that he is talking to the system and revealing information (e.g., password)

Browsing

After an intruder has gained access to a system he may peruse any files that are available for reading and glean useful information for further penetrations

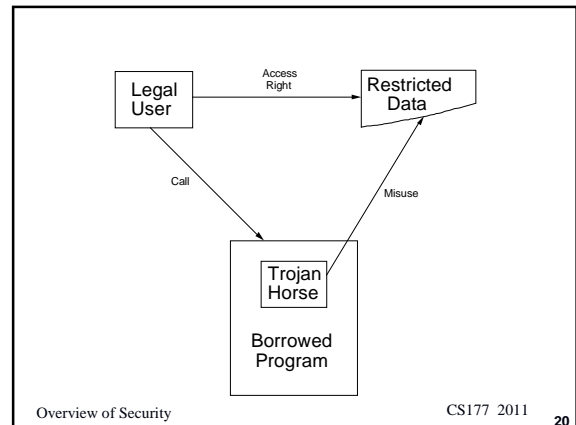
- Often done by legitimate users



Trojan Horse

A program that does more than it is supposed to do

- More sophisticated threat
- A text editor that sets all of your files to be publicly readable in addition to performing editing functions
- Every unverified program is suspect (especially games)



Trap Door

A system modification installed by a penetrator that opens the system on command

- May be introduced by a system developer
- Bogus system engineering change notice

Virus

A program that can infect other programs by modifying them to include a possibly evolved copy of itself

Examples

Amiga Virus

Resident on boot block

IBM Christmas Virus

Names and netlog files

Denial of service

Census Bureau

County and City Data Book CD-ROM

WWW Pages Containing Applets

MIME-encoded Mail

Code Red Worm

Blaster

Sasser

Statistical Database

A statistic is *sensitive* if it discloses confidential information about some individual, organization, or company

Nonsensitive statistics may lead to the disclosure of sensitive data

Inference of Sensitive Data From Nonsensitive Information

Can detect information about an individual by querying about a group where the individual is the only member in the group or the only one not in the group

For example:

If Smith is the only foreign worker, one can deduce information about Smith by querying about non-foreigners

Overview of Security

CS177 2011

25

Example Database

Name	Sex	Major	Class	SAT	GP
Bruno	F	CS	2008	600	3.2
Alley	F	EE	2008	520	2.5
Lasta	M	EE	2009	630	3.5
Gise	F	CS	2009	800	4.0
Kies	M	BIO	2007	500	2.2
Costo	M	EE	2010	580	3.0
Kraig	M	CS	2009	700	3.8
Good	F	PSY	2007	580	2.8
Islay	M	CS	2010	600	3.2
Farel	F	BIO	2007	750	3.8
Pfau	F	PSY	2010	500	2.5
Ghezzi	M	EE	2009	600	3.0
Boyer	M	CS	2007	650	3.5

Overview of Security

CS177 2011

26

THREAT CLASSIFICATION

Overview of Security

CS177 2011

27

Security

Confidentiality - Keeping data and resources hidden or protected from unauthorized disclosure

Integrity - ensures that the data and programs are modified or destroyed only in a specified and authorized way

- Data integrity (integrity)
- Origin integrity (authentication)

Availability - ensures that the resources of the system will be usable whenever they are needed by an authorized user

Overview of Security

CS177 2011

28

Computer Security Threats

Browsing
Leakage
Inference

Tampering
Accidental destruction

Masquerading

Denial of services

Overview of Security

CS177 2011

29

Browsing

Searching through main and secondary memory for residue information

Leakage

Transmission of data to an unauthorized user from a process that is allowed to access the data

Inference

Deducing confidential data about an individual by correlating unrelated statistics about groups of individuals

Overview of Security

CS177 2011

30

Tampering

Making unauthorized changes to the value of information

Accidental Data Destruction

Unintentional modification of information

Masquerading

Gaining access to the system under another user's account

Denial of Service

Prevention of authorized access to computer resources or the delaying of time-critical operations

Bishop Threat Definitions

Threat is a potential violation of security

Attacks are those actions which could cause a threat to occur

Attackers are those who execute an attack

Confidentiality, integrity, availability are enforced to counter the threats to the security of a system and foil attacks

Cerias Definitions

Vulnerability is a flaw in a system that allows a policy to be violated

Exploit is the act of exercising a vulnerability
Also used to refer to an actual program, binary or script that automates an attack

Exposure is an information leak that may assist an attacker

Bishop Classes of Threats

- Disclosure (unauthorized access to information)
 - Snooping, wiretapping, eavesdropping
 - These threats are mostly addressed by confidentiality services
- Deception (acceptance of false data)
 - Modification, spoofing, repudiation of origin, denial of receipt
 - These threats are mostly addressed by integrity services
- Disruption (interruption or prevention of correct operation)
 - Modification
 - These threats are mostly addressed by integrity services
- Usurpation (unauthorized control of some part of the system)
 - Modification, spoofing, delay, denial of service
 - These threats are addressed by integrity and availability services

Security Policy

A security policy is a statement of what is and what is not allowed

It defines the concept of “security” for a computer system

It can be defined formally or informally

When defined formally, it provides a precise characterization of the secure states of a system

Simple Example Policy

- Policy disallows cheating
 - Includes copying homework, with or without permission
- CS class has students do homework on computer
- Anne forgets to read-protect her homework file
- Bill copies it

- Who cheated?
 - Anne, Bill, or both?

Answer Part 1

Bill cheated

- Policy forbids copying homework assignment
- Bill copied
- System entered unauthorized state (Bill having a copy of Anne's assignment)

Answer Part 2

- Anne didn't protect her homework
 - Not required by security policy
- She didn't breach security
- If policy said students had to read-protect homework files, then Anne would have breached security
 - She didn't

Access Control

A means of limiting a user's access to only those entities that the policy determines should be accessed

Subjects - Active entities in the system (e.g. , users, processes, programs)

Objects - Resources or passive entities in the system (e.g. , files, programs, devices)

Access Modes - Read, write, execute, append, update

Access Control Mechanisms - Determine for each subject what access modes it has for each object

Access Control

Discretionary Access Control (DAC)

The owner specifies to the system what other users can access his files
(Access is at the user's discretion)

Mandatory Access Control (MAC)

The system determines whether a user can access a file based on the fixed security attributes of the user and of the file
(Non-discretionary access)

Access Control Matrix

Subjects	Objects				
	O1	O2	O3	O4	O5
S1			W	RW	W
S2	R	E		R	
S3		RW	E		
S4	RE		RW		RE

Access Control List (Authorization List)

- Associated with each object
- Contains subject name and type of access allowed
- Corresponds to column in the matrix

Capability List (C-list)

- Associated with each subject
- Contains object name and type of access allowed
- Corresponds to a row in the matrix
- Defines the environment or domain that the subject may access

Mandatory Control Policy

- Each subject has an access class (authorization)
- Each object has an access class (classification)
- Access class made up of
 - * level
 - * category set
- Comparison of access classes
 - * equal (=)
 - * less than (<)
 - * greater than (>)
 - * not comparable (NC)

Example Mandatory Controls

- Three security levels
Unclassified, Confidential, Secret
- Three security categories
Crypto, Nuclear, Intelligence

Comparisons

SECRET/ {CRYPTO} = SECRET/ {CRYPTO}
SECRET/ {CRYPTO} > CONFIDENTIAL/ {CRYPTO}
SECRET/ {CRYPTO} < SECRET/ {CRYPTO, NUCLEAR}
SECRET/ {CRYPTO} NC SECRET/ {NUCLEAR}

Access Rules

Simple security property

Read permission if:

Access class (subject) \geq Access class (object)

*- *Property*

Write permission if:

Access class (subject) \leq Access class (object)

Policies and Mechanisms

- A security policy is a statement of what is, and what is not, allowed
- A security mechanism is a method, tool, or procedure to enforce a security policy
 - Different mechanisms can be used to enforce the same policy
- For example:
 - Policy: students should not copy other students' assignments
 - Mechanism: % chmod 700 ~

Goals of Security

- Given a security policy's definition of secure and insecure states, the corresponding security mechanisms can perform different functions
- Prevention
 - Prevent attackers from violating security policy
 - Example: use of passwords
- Detection
 - Detect attackers' violations of security policy
 - Example: use of logging of sensitive operations
- Recovery
 - Stop attack, assess, and repair damage
 - Example: use of checkpointing and virtualization
 - Continue to function correctly even if attack succeeds

Overview of Security

CS177 2011 49

Approaches to Security

- Procedural
- Functions and Mechanism
- Assurance

Overview of Security

CS177 2011 50

Procedural Approaches

Prescribe appropriate behavior for a user interacting with the system

- periods processing
- guidelines for managing passwords
- appropriate handling of removable storage devices
- electronic voting systems

Overview of Security

CS177 2011 51

Periods Processing

Split the day into periods and run different classification jobs in each period

Overview of Security

CS177 2011 52

Guidelines for Choosing Passwords

- Long (8 character minimum)
- Non-obvious
- Not written in an obvious place
- Changed at appropriate intervals
- Not shared
- Not stored

Many guidelines can be enforced by the system

Overview of Security

CS177 2011 53

Appropriate Handling of Hardware

Management of removable media

Disposal of hardware

- study showed that confidential information is often left in hardware to be salvaged (*IEEE Security & Privacy* magazine, January 2003)

Overview of Security

CS177 2011 54

Functions and Mechanisms

Enforce security policy

Examples are the 3As

- **Authentication:** assures that a particular user is who he/she claims to be
- **Access control:** a means of limiting a user's access to only those entities that the policy determines should be accessed
- **Audit:** a form of transaction record keeping. The data collected is called an audit log

Authentication Mechanisms

Authenticates users at login time

- Secure attention key
- One way functions

Secure Attention Key

- Foils attempts at spoofing
- Guarantees trusted path to the system
- User must use it

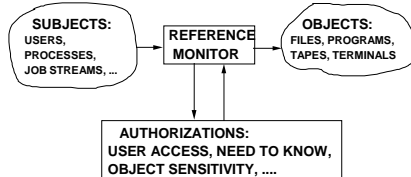
One-Way Function

A function whose inverse is computationally infeasible to determine

- Enciphered passwords are stored in a password file
- At login time password presented by the user is enciphered and compared to what is in the password file

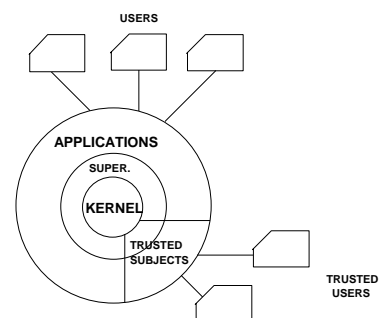
Reference Monitor

Provides mediation of all accesses to assure that the access control policy is enforced



- Reference Monitor must be
- Invoked on every reference
 - Tamperproof
 - Subject to analysis/test whose completeness can be assured

Security Kernel



Kernel must handle parts of the operating system that manage resources shared by multiple users

Supervisor contains functions that provide useful common facilities but do not manage anything shared among users

Trusted subjects are used to extend the security policy

- May perform actions not permitted by the access checks
- Must be subject to analysis and test just like the security kernel

ASSURANCE TECHNIQUES

Assumptions and Trust

- A policy describes the security of a system in a certain environment and under certain assumptions
 - A policy that states that students should not copy other students' files, which is enforced by using file system access control mechanisms, is valid under the assumption that students don't share passwords and that they set file access privileges correctly
- Assumptions about policies
 - A policy unambiguously partitions a system's states into secure and nonsecure
 - A policy correctly captures the security requirements of the real world

Assumptions and Trust

- Assumptions about mechanisms
 - The security mechanisms enforce the policy and prevent the system from entering a nonsecure state
 - The mechanisms can be trusted
 - Are implemented correctly
 - Are installed and administered correctly

Assurance

- Assurance is a measure of how well the system meets its requirements
 - In other words, how much one can trust the system to do what it is supposed to do
- Assurance is derived by analyzing the specification, design, and implementation of a system

Assurance Techniques

Penetration analysis

Covert channel analysis

Formal verification

Penetration Analysis

Uses a collection of known flaws, generalizes the flaws, and tries to apply them to the system being analyzed

- Penetration team known as "Tiger Team"
- Demonstrates the presence not the absence of protection failures

Covert Channels

Security analysis of both overt and covert channels is necessary

Overt channel – Uses the system's protected data objects to transfer information

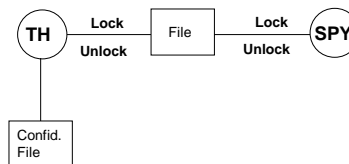
Covert channel – Uses entities not normally viewed as a data object to transfer information

Two Types of Covert Channels

Storage channels – the sender alters the value of a data item and the receiver detects and interprets the altered value to receive information covertly

Timing channels – the sender modulates the amount of time required for the receiver to perform a task or detect a change in an attribute, and the receiver interprets the delay or lack of delay to receive information covertly

File Lock Example

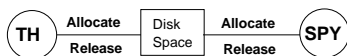


Trojan Horse locks a file if it wants to signal a 1 and doesn't lock it if it wants to signal a 0

Spy attempts to lock the file

- If lock fails spy interprets it as a 1
- If lock succeeds spy interprets it as a 0

Disk Quota Example



Trojan Horse allocates or releases disk space to send a 1 or a 0

Spy attempts to allocate disk space

- allocate fails if no more space is available

This is a resource exhaustion channel

IPC Quota Example

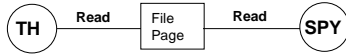


Trojan Horse reads a message to release a message slot or does nothing

Spy sends a message

- Send fails if the queue is full
- Send succeeds if there is space in the queue

Paging Example



Trojan Horse reads a page or does nothing

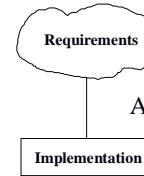
Spy reads a page

The read operation

- returns immediately if the page is in main memory
- loads page in main memory, with a noticeable delay, if the page has not been read lately

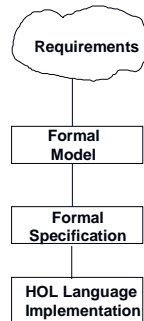
This is a timing channel

Problem



Are these consistent?

Formal Specification and Verification



Models

Access Control

Considers subjects and objects requirements:

- 1) If subject s has read access to object o , then
 $Security_level(s) \geq Security_level(o)$
- 2) If subject s has write access to object o , then
 $Security_level(s) \leq Security_level(o)$

Formal Specifications

State Machine

Relates values of variables before and after each state transition

e.g.

Exchange (x,y)

New_value(x) = y
& New_value(y) = x

Formal Specifications

Algebraic

Relates results of sequences of operations

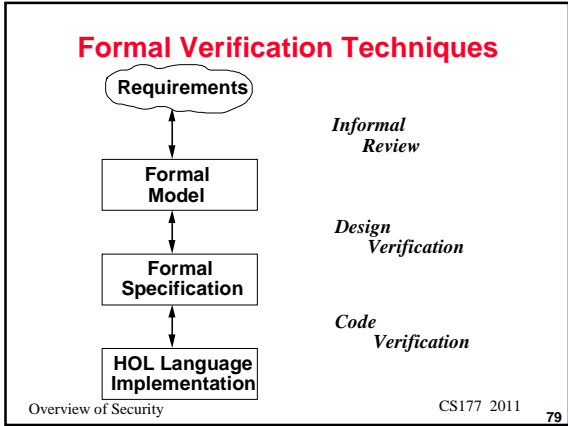
e.g.,

Axioms:

Exchange $(Exchange(x,y)) = x,y$

First $(Exchange(x,y)) = Last(x,y)$

Last $(Exchange(x,y)) = First(x,y)$



Design Verification

Consistency between the model and the specification

Assumes:

- Model is appropriate
- Specification is complete

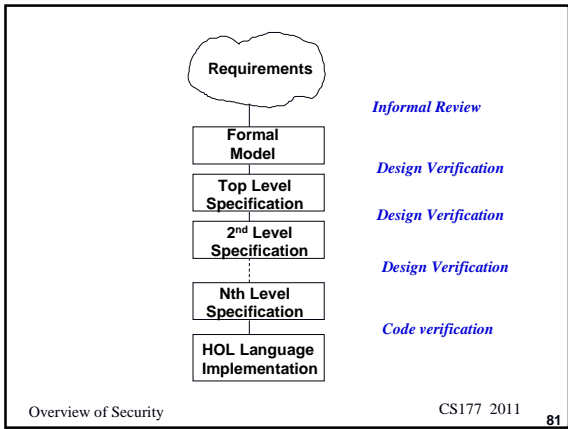
Code Verification

Consistency between specification and the implementation

Assumes:

- Specification is appropriate
- Implementation language is correctly defined

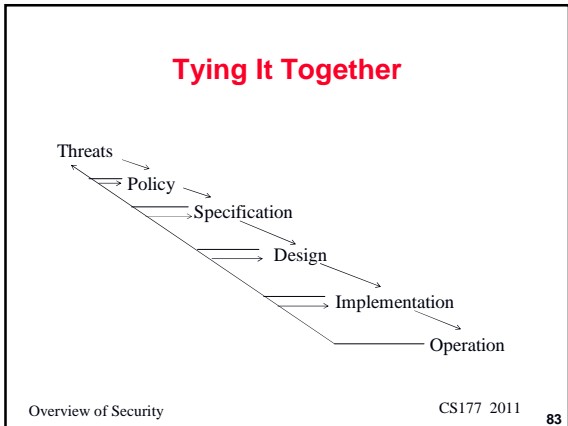
Overview of Security CS177 2011 80



Operational Issues

- Useful, real-world policies have to take into account other factors other than the enforcement of confidentiality, integrity, and availability
- Cost-Benefit Analysis
 - How much does security cost?
 - Is it cheaper to prevent or recover?
- Risk Analysis
 - From what should we protect the assets?
 - External attacker with no access
 - External attacker with partial access
 - Internal attacker
 - How much should we protect?
- Laws and Customs
 - Are the desired security measures illegal?
 - Are the proposed security mechanisms acceptable?
 - Is the resulting system user-friendly?

Overview of Security CS177 2011 82



What About Privacy?

Confidentiality - ensures that sensitive information is not disclosed to unauthorized recipients

Integrity - ensures that the data and programs are modified or destroyed only in a specified and authorized way

Availability - ensures that the resources of the system will be usable whenever they are needed by an authorized user

Privacy - ensures that only the information that an individual wishes to disclose is disclosed

Overview of Security CS177 2011 84

Other Privacy Concerns

Privacy is more than just confidentiality

Privacy advocates consider it important to be able to verify the integrity of personal information, especially when that information can be used against them (e.g., credit reports)

Internet Privacy

- The ability to control what information one reveals about oneself over the Internet, and to control who can access that information.
- These concerns include whether email can be stored or read by third parties without consent, or whether third parties can track the web sites someone has visited.
- Another concern is whether web sites which are visited collect, store, and possibly share personally identifiable information about users.

<http://en.wikipedia.org/wiki/Privacy#Informational>

Web Sites and Mailing Lists

- History of computer security
 - <http://csrc.nist.gov/publications/history/>
- Security surveys/statistics
 - CERT statistics:
http://www.cert.org/stats/cert_stats.html
 - CSI/FBI survey:
<http://www.gocsi.com/awareness/fbi.jhtml>

More Web Sites and Mailing Lists

- SecurityFocus.com
 - Bugtraq
 - Focus-ids
 - ...
- Phrack.org
- Zone-h.org