Week 11 - Friday

COMP 4290

Collect Passwords and Secret Phrases

Last time

- What did we talk about last time?
- Started privacy

Questions?

Project 3

Assignment 4

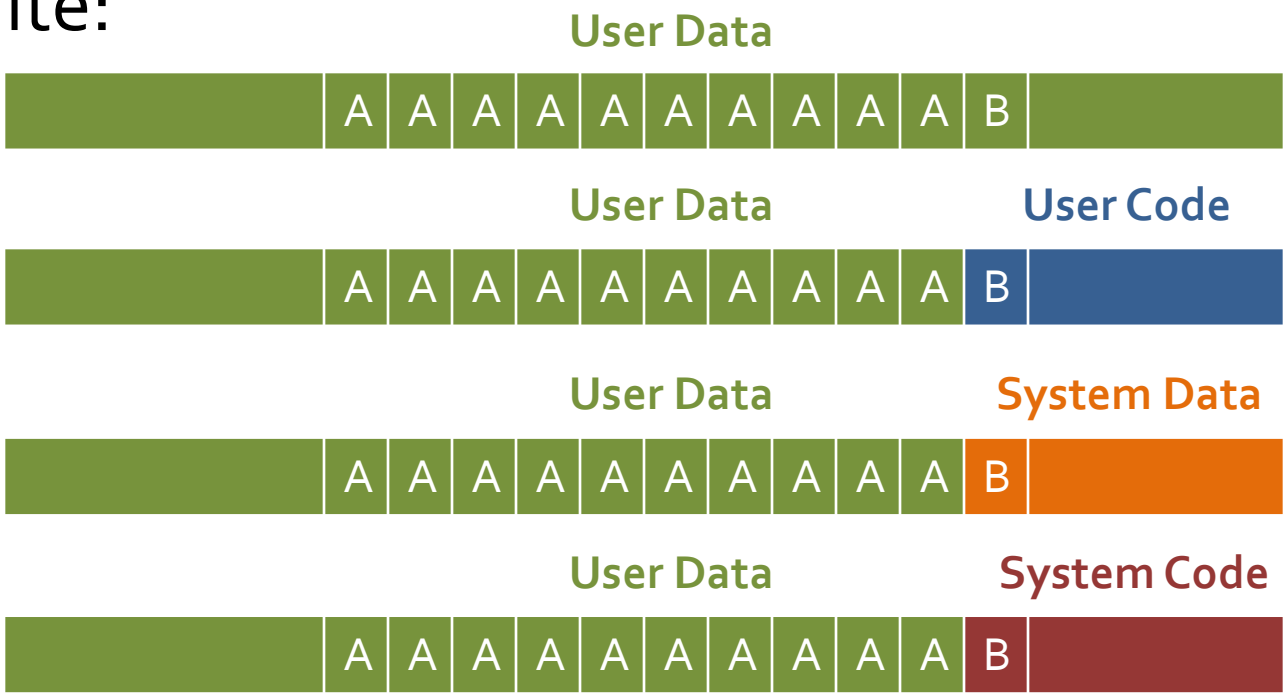
Olivia Crespo Presents

Review

Program Security

Buffer overflow

- A **buffer overflow** happens when data is written past the end (or beginning) of an array
- It could overwrite:
  - User data
  - User code
  - System data
  - System code



Incomplete mediation

- **Incomplete mediation** happens with a system does not have complete control over the data that it processes
- Example URL:
  - `http://www.security.com/query.php?date=2012March20`
- Wrong URL:
  - `http://www.security.com/query.php?date=2000Hyenas`
- The HTML generates the URL, but the URL can be entered manually

Time-of-check to time-to-use

- A **time-of-check to time-to-use flaw** is one where one action is requested, but before it can be performed, the data related to the action is changed
- The book's example is a man who promises to buy a painting for $100 who puts five $20 bills on the counter and pulls one back when the clerk is turning to wrap up the painting
- In this flaw, the first action is authorized, but the second may not be

Viruses

- Terminology is inconsistent
- Popular culture tends to call everything a virus
- Sometimes we will too, but here are some other terms:

Type	Characteristics
Virus	Attaches itself to a program and propagates copies of itself to other programs
Trojan horse	Contains unexpected, additional functionality
Logic bomb	Triggers action when condition occurs
Time bomb	Triggers action when specified time occurs
Trapdoor	Allows unauthorized access to functionality
Worm	Propagates copies of itself through a network
Rabbit	Replicates itself without limit to exhaust resources

- Almost all of these are, by definition, Trojan horses
- Worms differ from viruses primarily because they spread across networks

Where Viruses Live

- One-time execution
- Boot sector
  - The part of a hard drive that says what code to load to start your OS
- Memory resident
  - Sometimes called TSR (terminate and stay resident)
- Inside documents
- A few other places that are sensible:
  - Applications
  - Libraries
  - Compilers (infect programs as you create them)
  - Antivirus software

Virus Signatures

- Storage patterns
  - The size of a file
  - Compare against a hash digest for the program
- Execution patterns
  - Viruses are also suspicious because of the way they execute
  - The functioning of the code compared to some standard
  - Suspicious execution patterns (weird JUMP commands)

Polymorphic viruses

- Because virus scanners try to match strings in machine code, virus writers design **polymorphic viruses** that change their appearances
- **No-ops**, code that doesn't have an impact on execution, can be used for simple disguises
- Clever viruses can break themselves apart and hide different parts in randomly chosen parts of code
  - Similar to code obfuscation
- Advanced polymorphic viruses called **encrypting viruses** encrypt parts of themselves with randomly chosen keys
  - A scanner would have to know to decrypt the virus to detect it
- Virus scanners cannot catch everything

Targeted malicious code

- Trapdoors
  - A way to access functionality that is not documented
  - Often inserted during development for testing purposes
- Salami attacks
  - Steal tiny amounts of money when a cent is rounded in financial transactions
  - Or, steal a few cents from millions of people
- Rootkits
- Privilege escalation
- Keystroke logging

Testing to prevent programming flaws

- **Unit testing** tests each component separately in a controlled environment
- **Integration testing** verifies that the individual components work when you put them together
- **Function** and **performance tests** sees if a system performs according to specification
- **Acceptance testing** give the customer a chance to test the product you have created
- The final **installation testing** checks the product in its actual use environment

Testing methodologies

- **Regression testing** is done when you fix a bug or add a feature
  - We have to make sure that everything that used to work still works after the change
- **Black-box testing** uses input values to test for expected output values, ignoring internals of the system
- **White-box** or **clear box testing** uses knowledge of the system to design tests that are likely to find bugs
- **You can only prove there are bugs. It is impossible to prove that there aren't bugs.**

Secure design principles

- Saltzer and Schroeder wrote an important paper in 1975 that gave 8 principles that should be used in the design of any security mechanisms
  1. Least privilege
  2. Fail-safe defaults
  3. Economy of mechanism
  4. Complete mediation
  5. Open design
  6. Separation of privilege
  7. Least common mechanism
  8. Psychological acceptability

Web Security – User Side

Browser security issues

- Browsers are how most of the world interacts with the Internet
- There are lots of problems when trying to maintain security:
  - Browsers often connect to more than just the URL listed in the address bar
  - Fetching a page automatically fetches lots of other data
  - If the browser is corrupted, you have no protection
  - Most browsers support plug-ins, which can be malicious or badly implemented
  - Browsers can access data on the user computer
  - The user does not know what data the browser is sending

Man-in-the-Browser

- The browser controls all the interactions with the world wide web
- If your browser has been compromised, it doesn't matter how good your encryption is
- The browser sees all the data before it is encrypted
- SilentBanker is an example of a plug-in that stole bank information
  - The banking websites still worked!

Page-in-the-middle

- A page-in-the-middle attack is one in which you are redirected to a page that looks like the one you wanted
  - For example, a copy of your banking website
- Such a page might be arrived at because of a phishing link or DNS cache poisoning
- A browser-in-the-middle attack is worse, since your browser is compromised and no websites can be trusted

Program download substitution

- A page could trick you into downloading a file that appears to be an application you want
  - In reality, it's a virus, Trojan horse, or other malware
- How do you know what you're downloading?
- Often, there's no way to be sure

User-in-the-middle

- A user-in-the-middle attack tricks an unsuspecting user to do something only a human can do, like solve a CAPTCHA
- Spam and porn companies often have the same owners
- People get offers for free porn in their e-mail, provided that they fill out a CAPTCHA
- This attack is not very damaging to the individual, but it wastes time and fills the world with more spam

Browser authentication issues

- We've already talked about how people authenticate
- One of the problems here is that **computers** are failing to authenticate
  - You're not sure that the site you're connecting to is really your bank
- The problem is hard because computers authenticate based almost entirely on what they know
  - It's possible to eavesdrop on such information

- Some banks let you to pick a picture and a caption

GOAT POLITICS

Authentication approaches

- Web authentication can be done with approaches beyond or in addition to a password
- Shared secret
  - Secret questions asked earlier
- One-time password
  - Password provided by a SecurID
- Out-of-band communication
  - Sending a PIN and a credit card in separate mailings
  - Texting a one-time password to a registered cell phone

Defaced web site

- **Website defacement** is when an attacker changes the content of a legitimate website
- Usually, this is done by exploiting a weakness in authentication of the people who are allowed to update content
- These attacks can be pranks
- They can be done to demonstrate that security is poor
  - Often to embarrass government websites
- They can be done to show political disagreement with the website or the agency behind the website
- The changes could be subtle enough that the change is not noticed for a while

Fake website

- Websites are easy to fake
- By their nature, the HTML, JavaScript, CSS, and images used to create a website are all publically available
  - It's even possible to link to current images on the real website
- This attack is usually designed to trick users into entering private information into the malicious website

Protecting websites

- Detecting that a change has occurred on a website can be difficult
- One approach is to make a hash value of the website
  - Store the hash elsewhere, securely
  - Hash the contents of the website periodically to see if it still matches
  - This approach only works if the data doesn't constantly change
- **Digital signatures** allows companies to sign code to verify that they did originate the code
  - Example: ActiveX controls
  - You shouldn't be running this kind of code anyway

Web bugs

- Only a website you visit can leave a cookie or run JavaScript, right?
  - Sure, but how many sites do you visit?
- Images that are linked to other websites (especially ads) count as visiting other websites
- Visiting a single page could store cookies from every ad on the page (and more!)
- **Web bugs** are images that are usually 1 x 1 pixels and clear
  - They make it impossible to know how many sites could be storing cookies

Clickjacking

- **Clickjacking** is when you think you're clicking on one button, but you're really clicking on another
- It could be that you're agreeing to download or install a program that you don't think you are
  - Called a **drive-by download**
- It could be that you think you're entering data into a real website, but it's just a front for a malicious one
- These attacks are possible because web pages can have transparent frames, allowing you to see something that you're not really interacting with

Obtaining user or website data

- The inherently unsecure model used for web interactions has a number of weak points
- Some ways that website data can be leaked include:
  - Cross-site scripting
  - SQL injection
  - Dot-dot-slash
  - Server-side includes

Fake e-mail

- There is lots of fake e-mail out there
- The book calls **spam** fake or misleading e-mail
- Several kinds of spam are rising
  - Fake "Your message could not be delivered" messages
  - Fake social networking messages
  - Current events messages
  - Shipping notices

Why do people send spam?

- Advertising black- or graymarket pharmaceuticals
- Pump and dump – artificially inflating the price of a stock
- General advertising
- Malware in the e-mail or in links from the e-mail
- Advertising sites (such as porn) that might be illegal
- Cost is virtually nothing

Dealing with spam

- Legal approaches
  - US CAN-SPAM act
  - Directive 2002/58/EC in Europe
  - It's hard to define what is and isn't spam
  - Most laws require an opt-out mechanism, but enforcement is hard
- IP addresses are easy to spoof, but the next generation Internet might change that
- Screening programs try to filter out spam (with both false positives and false negatives)
- Some web hosting companies enforce volume limitations on how many e-mails can be sent per day
- Paying postage per e-mail?

E-mail spoofing

- SMTP is the protocol for sending e-mail
- It's very straight-forward
- The **from** field is easy to spoof
- There are protocols with authentication built in, but regular SMTP is entrenched how
- You can never trust header information in an e-mail

Phishing

- **Phishing** is when an e-mail tries to trick someone into giving out private data or doing something else unsafe
- **Spear phishing** is phishing that targets a specific individual
  - Details about that user's life or accounts might be included
- **Whaling** is a term used for spear phishing of rich people or celebrities
  - They have more money
  - Many of their personal details could be public

OS Security

Separation

- OS security is fundamentally based on separation
  - **Physical separation:** Different processes use different physical objects
  - **Temporal separation:** Processes with different security requirements are executed at different times
  - **Logical separation:** Programs cannot access data or resources outside of permitted areas
  - **Cryptographic separation:** Processes conceal their data so that it is unintelligible

Memory protection

- Protecting memory is one of the most fundamental protections an OS can give
  - All data and operations for a program are in memory
  - Most I/O accesses are done by writing memory to various locations
- Techniques for memory protection
  - Fence
  - Base/bounds registers
  - Tagged architectures
  - Segmentation
  - Paging

Storing access control information

- Directory based approaches
  - Create a directory that lists all the objects a given user can access and their associated rights:
  - Problems:
    - Directories can become large
    - How is access revoked?
    - What if two files in different locations in the system have the same name?
- Access control lists
  - List all the users that have rights for a specific object
  - Most objects only have a few legal users
  - Wild cards can make the situation easier
- Access control matrices
  - Both directories and access control lists are equivalent
  - We can also imagine a matrix that holds all subjects and all objects
  - It is too inefficient for most systems to be implemented this way, but security researchers sometimes use this model for theoretical purposes

Access control matrix example

	Objects			
Subjects	file 1	file 2	process 1	process 2
process 1	*read, write, own*	*read*	*read, write, execute, own*	*write*
process 2	*append*	*read, own*	*read*	*read, write, execute, own*

Bell-LaPadula overview

- Confidentiality access control system
- Military-style classifications
- Uses a linear clearance hierarchy
- All information is on a **need-to-know** basis
- It uses **clearance** (or **sensitivity**) levels as well as project-specific **compartments**

Security clearances

- Both subjects (users) and objects (files) have security clearances
- Below are the clearances arranged in a hierarchy

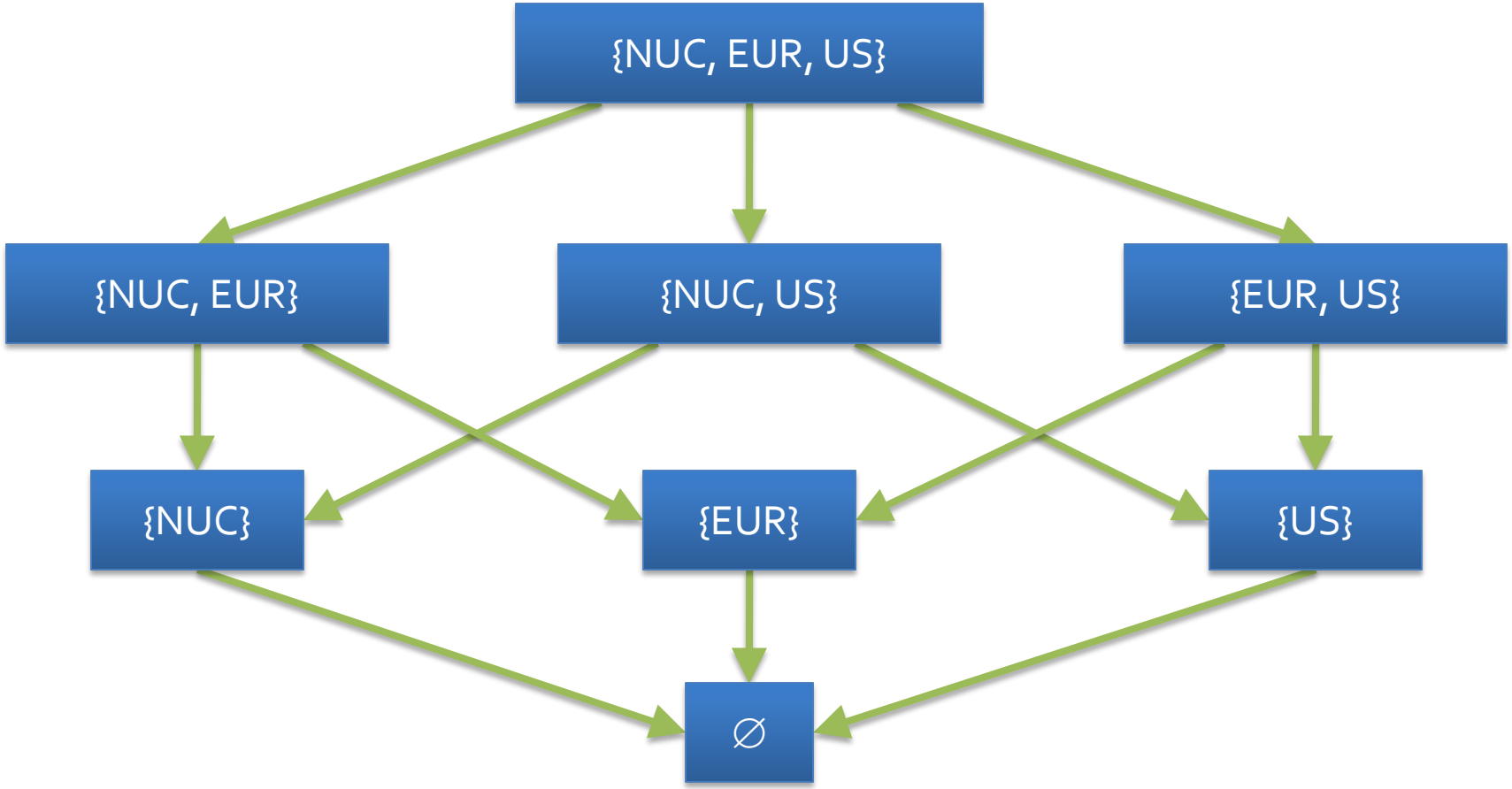
Clearance Levels	Sample Subjects	Sample Objects
Top Secret (TS)	Tamara, Thomas	Personnel Files
Secret (S)	Sally, Samuel	E-mail Files
Confidential (C)	Claire, Clarence	Activity Log Files
Restricted (R)	Rachel, Riley	Telephone List Files
Unclassified (UC)	Ulaley, Ursula	Address of Headquarters

Adding compartments

- We add compartments such as NUC = Non-Union Countries, EUR = Europe, and US = United States
- The possible sets of compartments are:
  - ∅
  - {NUC}
  - {EUR}
  - {US}
  - {NUC, EUR}
  - {NUC, US}
  - {EUR, US}
  - {NUC, EUR, US}
- Put a clearance level with a compartment set and you get a **security level**
- The literature does not always agree on terminology

Romaine lattice

- The subset relationship induces a **lattice**



Bell-La Padula properties

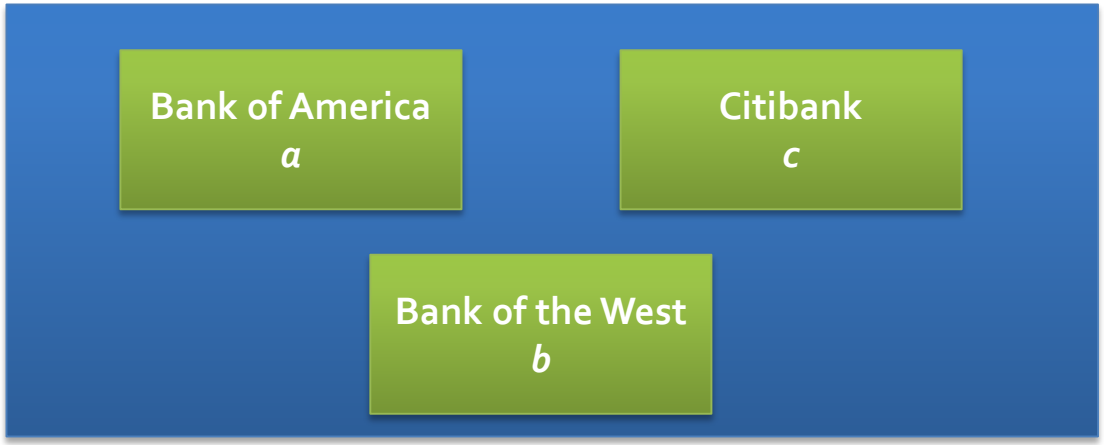
- Let L be a security level and C be a category
- We say that security level (L, C) **dominates** security level (L', C') if and only if $L' \leq L$ and $C' \subseteq C$
- Simple security requires (L_S, C_S) to dominate (L_O, C_O) and S to have read access
  - **Read down**
- *-property now requires (L_O, C_O) to dominate (L_S, C_S) and S to have write access
  - **Write up**

Chinese Wall model

- The Chinese Wall model respects both confidentiality and integrity
- It's very important in business situations where there are conflict of interest issues
- Real systems, including British law, have policies similar to the Chinese Wall model
- Most discussions around the Chinese Wall model are couched in business terms

COI Examples

Bank COI Class



Gasoline Company COI Class

Chinese Wall overview

- We can imagine the Chinese Wall model as a policy controlling access in a database
- The **objects** of the database are items of information relating to a company
- A **company dataset** (CD) contains objects related to a single company
- A **conflict of interest** (COI) class contains the datasets of companies in competition
- Chinese Wall rules prevent people from reading and writing data from CDs in different COIs

Biba model

- Integrity based access control system
- Uses integrity levels, similar to the clearance levels of Bell-LaPadula
- Precisely the **dual** of the Bell-LaPadula Model
- That is, we can only read up and write down
- Note that integrity levels are intended only to indicate integrity, **not** confidentiality
- Actually a measure of accuracy or reliability

Mandatory and discretionary access control

- **Mandatory access control** (**MAC**) means that the controls are enforced by rules in the system, not by user choices
  - Bell-La Padula is a perfect example of MAC
- **Discretionary access control** (**DAC**) means that the user has control over who can access the objects he or she owns
  - Linux and Windows are largely DAC systems
- Most real systems have elements of both

Network Security

Packet switched vs. circuit switched

- The Internet is a packet switched system
- Individual pieces of data (called packets) are sent on the network
  - Each packet knows where it is going
  - A collection of packets going from point A to point B might not all travel the same route
- Phone lines are circuit switched
  - This means that a specific circuit is set up for a specific communication
  - Operators used to do this by hand
  - Now it is done automatically
  - Only one path for data

Network strength

- If a single cut can case a network to go down, that network is vulnerable to a **single point of failure**
- Most important networks like electrical systems have redundancy so that this doesn't happen to a whole city
  - **Resilience** or **fault tolerance**

Terminology

- A **computer network** is at least two computers connected together
  - Often one is a **server** and the other is a **client**
- A computer system in a network is called a **node**
- The processor in a node is called a **host**
- A connection between two hosts is a **link**

Network characteristics

- **Anonymity**: We don't know who we're dealing with
- **Automation**: Communication may be entirely between machines without human supervision
- **Distance**: Communications are not significantly impacted by distance
- **Opaqueness**: It is hard to tell how far away other users are and to be sure that someone claiming to be the same user as before is

Transmission media

- Copper wire
  - **Twisted pair** is a pair of insulated copper wires
  - **Coaxial cable** has a single wire surrounded by an insulation jacket covered by a grounded braid of wire
  - **Repeaters** or **amplifiers** are needed periodically to prevent signal degradation
- Optical fiber
  - Carries light instead of electricity
  - Higher bandwidth and less signal degradation than copper
  - Replacing aging copper lines
- Wireless
  - Good for short distance
  - Uses radio signals
- Microwave
  - Strong signals
  - Requires line of sight
- Infrared
  - Similar to microwave but weaker signals
- Satellites
  - Need **geosynchronous** orbits
  - Secure applications need smaller **footprints** than broadcasts

Layers

- Protocols and standards define each layer
- Not every layer is always used
- Sometimes user errors are referred to as Layer 8 problems

Layer	Name	Activity	Example
7	**Application**	User-level data	HTTP
6	**Presentation**	Data appearance, some encryption	TLS
5	**Session**	Sessions, sequencing, recovery	IPC and part of TCP
4	**Transport**	Flow control, end-to-end error detection	TCP
3	**Network**	Routing, blocking into packets	IP
2	**Data Link**	Data delivery, packets into frames, transmission error recovery	Ethernet
1	**Physical**	Physical communication, bit transmission	Electrons in copper

TCP/IP

- The OSI model is conceptual
- Most network communication uses TCP/IP
- We can view TCP/IP as four layers:

Layer	Action	Responsibilities	Protocol
Application	Prepare messages	User interaction	HTTP, FTP, etc.
Transport	Convert messages to packets	Sequencing, reliability, error correction	TCP or UDP
Internet	Convert packets to datagrams	Flow control, routing	IP
Physical	Transmit datagrams as bits	Data communication	

TCP/IP

- **Transmission Control Protocol (TCP)**
  - Creates a reliable communication session
  - Wraps information into packets
  - Uses **port** numbers to connect processes to information streams
- **Internet Protocol (IP)**
  - Allows for unreliable transport
  - Wraps packets into datagrams
  - Uses IP addresses for routing
- **User Datagram Protocol (UDP)**
  - Alternative to TCP that is unreliable but has low overhead

Reconnaissance

- A smart attacker learns everything he or she can about the system before attacking it
- Useful methods for reconnaissance of a network include:
  - Port scans
  - Social engineering
  - Dumpster diving
  - OS and application fingerprinting
  - Background research

Eavesdropping and wiretapping

- **Eavesdropping** means overhearing private information without much effort
  - Administrators need to periodically monitor network traffic
- **Wiretapping** implies that more effort is being used to overhear information
- **Passive wiretapping** is only listening to information
- **Active wiretapping** means that you may adding or changing information in the stream

Wiretapping

- If you are on the same LAN, you can use a **packet sniffer** to analyze packets
- **Inductance** allows you to measure the signals inside of a wire without a direct physical connection
- Wireless is broadcast
  - Easy to intercept, but can be protected by WPA or WPA2 encryption (and hardly at all by WEP)
- Microwave is easy to intercept
  - Heavy multiplexing makes it hard to untangle individual signals
- Satellites are similar (unsecure but heavily multiplexed)
- Optical fiber is very difficult to tap
  - Cutting a single fiber means recalibrating the network
  - Repeaters and taps that connect the fiber are the best places to attack

Authentication issues

- Passwords are often easy to guess
  - Because we're bad at picking passwords
  - Because the user may not have realized that the machine would be exposed to network attacks
- Passwords are sent in the clear
- Bad hashes can give information about the password
- Sometimes buffer overflows can crash the authentication system
- Sometimes authentication is not needed
  - **`.rhosts`** and **`.rlogin`** files in Unix
  - Guest accounts
- Default passwords on routers and other devices that never get changed

Authentication attacks

- **Spoofing** is when an attacker carries out one end of a networked exchange
- A **masquerade** is spoofing where a host pretends to be another host
  - URL confusion: someone types **`hotmale.com`** (don't go there!) or **`gogle.com`**
- **Phishing** is a form of masquerading
- **Session hijacking** (or **sidejacking**) is carrying on a session started by someone else
  - Login is encrypted, the rest of the data often isn't
  - Firesheep allows you to log on to other people's Facebook and Twitter accounts in, say, the same coffeeshop
- **Man-in-the-middle** attacks

Confidentiality threats

- Misdelivery
  - Data can have bad addresses, occasionally because of computer error
  - Human error (e.g. James Hughes (student) instead of James Hughes (professor)) is more common)
- Exposure of data can happen because of wiretapping or unsecure systems anywhere along the network
- Traffic flow analysis
  - Data might be encrypted
  - Even so, it is very hard to hide where the data is going to and where it is coming from
  - Tor and other anonymization networks try to fix this

Integrity threats

- Attackers can falsify some or all of a message, using attacks we've talked about
  - Parts of messages can be combined
  - Messages can be redirected or deleted
  - Old messages can also be replayed
- Noise can degrade the signals
  - All modern network protocols have error correction built in
- Malformed packets can crash systems
- Protocols often have vulnerabilities

Denial of service

- Networks are one of the best places to launch an attack on availability
- In this setting, these are usually called **denial of service** (DoS) attacks
- Transmission failure can happen because a line is cut or because there is too much noise
- Flooding is a common technique
  - Ask for too many connections
  - Request too many of some other service
- **Distributed denial of service** (DDoS) attacks are common (often using zombies or botnets) to make a more damaging and hard to trace attack

Denial of service attacks

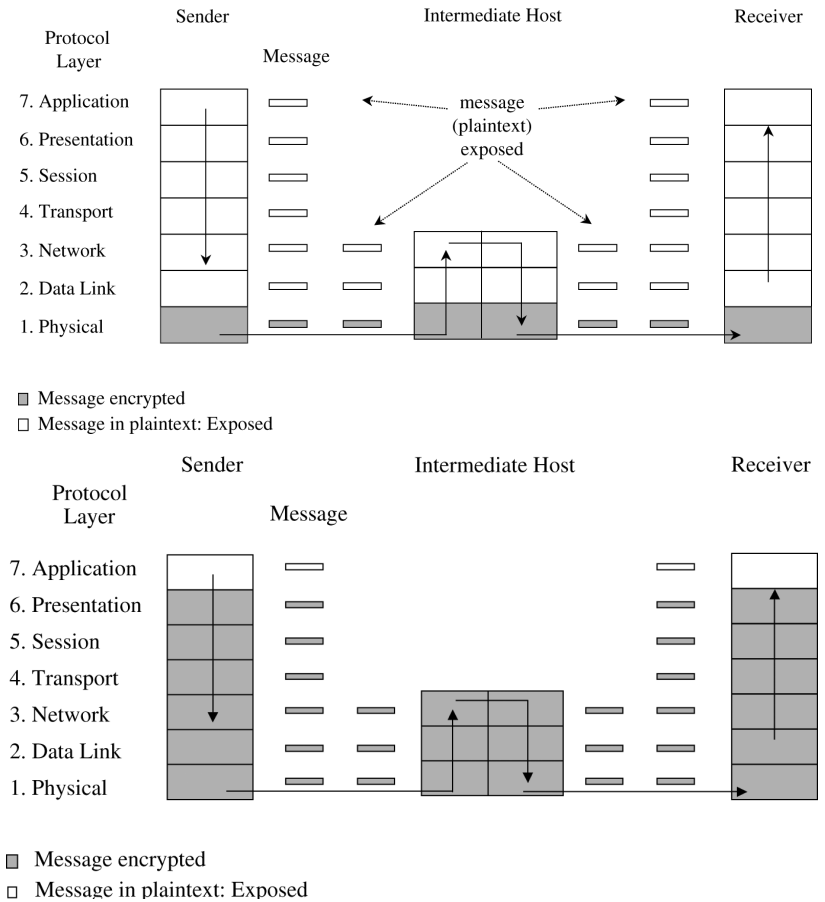
- TCP SYN floods
  - Exploit the three-way handshake
- Echo-chargen
  - Chargen sets up a stream of packets for testing
  - Echo packets are supposed to be sent back to the sender
  - If you can trick a server into sending echo packets to itself, it will respond to its own packets forever
- Ping of death
  - A ping packet requests a reply
  - If you can send more pings than a server can handle, it goes down
  - Only works if the attacker has more bandwidth than the victim (DDoS helps)
- Smurf
  - A ping packet is broadcast to everyone, with the victim spoofed as the originator
  - All the hosts try to ping the victim
  - The real attacker is hidden
- Teardrop
  - A teardrop attack uses badly formed IP datagrams
  - They claim to correspond to overlapping sequences of bytes in a packet
  - There's no way to put them back together and the system can crash

DNS attacks

- The Domain Name System (DNS) uses Domain Name Servers (also DNS) to convert user readable URLs like **`google.com`** to IP addresses
- Taking control of a server means that you get to say where google.com is
- For efficiency, servers cache results from other servers if they didn't know the IP
  - **DNS cache poisoning** is when an attacker gives a good server a bad IP address

Network encryption

- Encryption is important for network security
- **Link encryption** encrypts data just before going through the physical communication layer
  - Each link between two hosts could have different encryption
  - Message are in plaintext within each host
  - Link encryption is fast and transparent
- **End-to-end encryption** provides security from one end of the transmission to the other
  - Slower
  - Responsibility of the user
  - Better security for the message in transit



Database Security

What is a database?

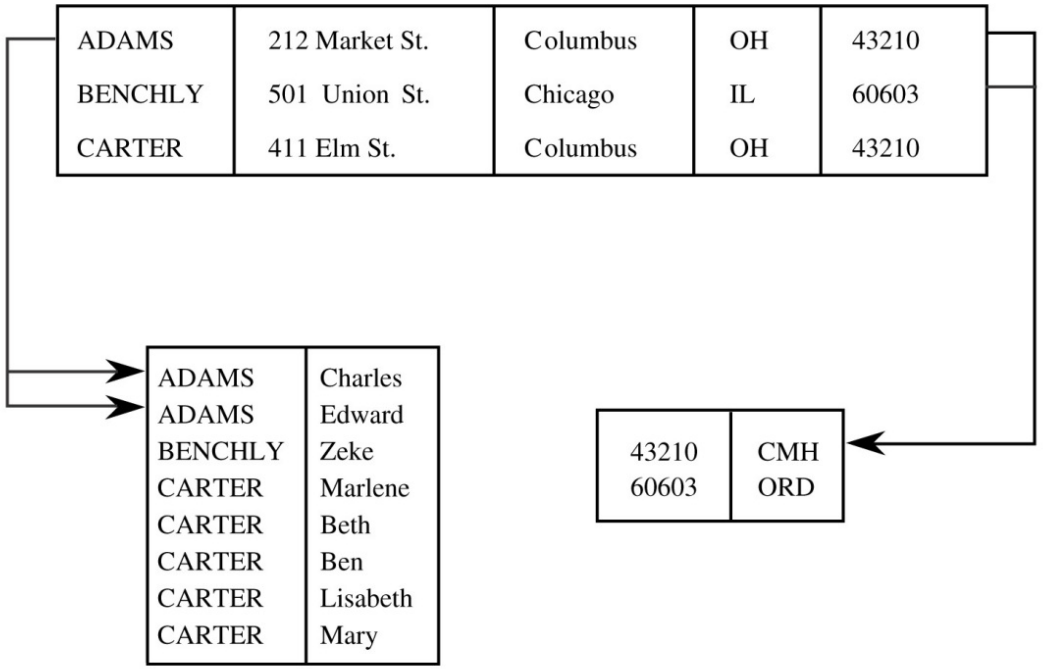
- A **database** is a collection of data and a set of rules to organize the data by relationships
- A **database administrator** makes the rules and controls access
- A **database management system** (**DBMS**) is the program through which the user interacts with the database

Database components

- Almost all modern databases use the relational database model
  - The fundamental unit of organization is a table
  - An older format for databases was hierarchical, like a tree
- A table consists of **records**
- A record consists **fields** or **elements**, which are each a specific item of data

Schemas

- The tables in a database are usually related to each other in some way
- The logical structure of a database is called a **schema**
- A user may only see part of it, called a **subschema**
- An **attribute** is the name of a column
- A **relation** is a set of columns



Queries

- A **query** is the name of a command given to a database by a user
- Queries can:
  - Retrieve
  - Modify
  - Add
  - Delete
- Most databases allow commands to be issued through a variant of SQL

Database security requirements

- Because they are a central part of modern business, several aspects of database security are crucial:
  - Physical database integrity
  - Logical database integrity
  - Element integrity
  - Access control
  - User authentication
  - Availability

Reliability and integrity

- **Reliability** is a measure of how long a software system can run without failing
  - Reliability is often quoted in terms of uptime percentage
  - Or mean time between failures
- Database reliability and integrity has three aspects:
  - Database integrity
    - Is the database as a whole protected from disk failure or corruption
  - Element integrity
    - Are only authorized users allowed to change elements
  - Element accuracy
    - Are the values in the elements correct

Two-phase update

- A key problem for database integrity is what happens if the system fails in the middle of an update
  - Then the database is inconsistent
- A two-phase update is a common solution
  - During the **intent** phase, the DBMS computes the results needed for the update, but does not change the database
  - During the **commit** phase, it changes all of the fields to the values computed in the intent phase
  - If the intent phase fails, the DBMS can start over from the beginning
  - If the commit phase fails, the DBMS can try to write all the data from the intent phase again

Disclosure of sensitive data

- The most serious disclosure of sensitive data is its exact value
- Bounds can also be disclosed
  - Example: highest salary and lowest salary
  - If the user can manipulate the bounds, he or she can search for specific values
- Negative result
  - Felonies is not zero
  - Visits to the oncology ward is not zero
- Existence
  - Knowing that a field even exists means someone is using it
- Probable value
  - How many people are in Bob's dorm room? 2
  - How many people in Bob's dorm room pirate movies? 1
  - There's a 50% chance that Bob pirates movies

Direct attack

- In a direct attack on sensitive information, a user will try to determine the values of a sensitive field by finding the right query
- Sometimes an unusual query will be used to bypass checks

Indirect attack

- To avoid leaking sensitive data, some DBMSs allow statistics to be reported
- Each of the following statistics can be attacked in different ways:
  - Sum
  - Count
  - Mean
  - Median

Protecting against inference

- Suppress obviously sensitive information
  - Easy, but incomplete
- Track what the user knows
  - Expensive in terms of computation and storage requirements
  - Analysis may be difficult
  - Multiple users can conspire together
- Disguise the data
  - Data is hidden
  - Users who are not trying to get sensitive data get slightly wrong answers

Integrity and confidentiality

- Integrity is difficult, but we can assign levels of trust
  - It is necessarily not going to be as rigorous as Biba
- Confidentiality
  - Difficult and causes redundancies since top secret information cannot be visible in any way to low clearance users
  - Worse, we don't want to leak any information by preventing a record from being added with a particular primary key (because there is a hidden record that already has that primary key)
  - **Polyinstantiation** means that records with similar or identical primary keys (but different data) can exist at different security levels

Upcoming

Next time…

- Exam 2 is on Monday

Reminders

- **Exam 2 is Monday**
- Work on Project 3
- Work on Assignment 4
  - **Due next Friday**