Access Control

Access Control

- Two parts to access control
- **Authentication:** Are you who you say you are?
    - Determine whether access is allowed
    - Authenticate human to machine
    - Or authenticate machine to machine
- **Authorization:** Are you allowed to do that?
    - Once you have access, what can you do?
    - Enforces limits on actions
- Note: "access control" often used as synonym for authorization

Are You Who You Say You Are?

- How to authenticate human a machine?
- Can be based on…
  - Something you **know**
    - For example, a password
  - Something you **have**
    - For example, a smartcard
  - Something you **are**
    - For example, your fingerprint

Something You Know

- Passwords
- Lots of things act as passwords!
  - PIN
  - Social security number
  - Mother's maiden name
  - Date of birth
  - Name of your pet, etc.

Why Passwords?

- Why is "something you know" more popular than "something you have" and "something you are"?
- **Cost**: passwords are free
- **Convenience**: easier for admin to reset pwd than to issue a new thumb

Keys vs Passwords

- **Crypto keys**
- Spse key is 64 bits
- Then 2^{64} keys
- Choose key at random…
- …then attacker must try about 2^{63} keys

- **Passwords**
- Spse passwords are 8 characters, and 256 different characters
- Then $256^8 = 2^{64}$ pwds
- Users do not select passwords at random
- Attacker has far less than 2^{63} pwds to try **(dictionary attack)**

Good and Bad Passwords

- Bad passwords
  - frank
  - Fido
  - password
  - 4444
  - Pikachu
  - 102560
  - AustinStamp
- Good Passwords?
  - jfIej,43j-EmmL+y
  - 09864376537263
  - P0kem0N
  - FSa7Yago
  - 0nceuP0nAt1m8
  - PokeGCTall150

Password Experiment

- Three groups of users — each group advised to select passwords as follows
  - **Group A:** At least 6 chars, 1 non-letter
  - winner → **Group B:** Password based on passphrase
  - **Group C:** 8 random characters
- Results
  - **Group A:** About 30% of pwds easy to crack
  - **Group B:** About 10% cracked
    - Passwords easy to remember
  - **Group C:** About 10% cracked
    - Passwords hard to remember

Password Experiment

- User compliance hard to achieve
- In each case, 1/3rd did not comply
  - And about 1/3rd of those easy to crack!
- Assigned passwords sometimes best
- If passwords not assigned, best advice is…
  - Choose passwords based on passphrase
  - Use pwd cracking tool to test for weak pwds
- Require periodic password changes?

Attacks on Passwords

- Attacker could…
  - Target one particular account
  - Target any account on system
  - Target any account on any system
  - Attempt denial of service (DoS) attack
- Common attack path

  Upgrading level of privilege
  - Outsider → normal user → administrator
  - May only require **one** weak password!

Password Retry

- Suppose system locks after 3 bad passwords. How long should it lock? What are +'s and -'s of each?
  - 5 seconds
    - ..circle all accounts...
  - 5 minutes
    - ...DoS...
  - Until Admin restores service

Password File?

- Bad idea to store passwords in a file
- But we need to verify passwords
- Cryptographic solution: **hash** the pwd
  - Store $y = h(\text{password})$
  - Can verify entered password by hashing
  - If Trudy obtains "password file," she does not obtain passwords
- But Trudy can try a forward search
  - Guess x and check whether $y = h(x)$

Dictionary Attack

- Trudy pre-computes $h(x)$ for all x in a **dictionary** of common passwords
- Suppose Trudy gets access to password file containing hashed passwords
  - She only needs to compare hashes to her pre-computed dictionary
  - After one-time work, actual attack is trivial
- Can we prevent this attack? Or at least make attacker's job more difficult?

Salt

- Hash password with **salt**
- Choose random salt s and compute
$y = h(\text{password}, s)$
 and store (s,y) in the password file
- Note: The salt s is not secret
- Easy to verify salted password
- But Trudy must re-compute dictionary hashes for each user
    - Lots more work for Trudy!

Other Password Issues

- Too many passwords to remember
  - Results in password reuse
  - Discovered one discovered all (even slightly modified)
- Who suffers from bad password?
  - Login password vs ATM PIN (all the system falls for one weak password)
- Failure to change default passwords
- Social engineering (34% just for asking)
- Error logs may contain "almost" passwords
- Bugs, keystroke logging, spyware, etc.

Passwords

- The bottom line…
- **Password cracking is too easy**
  - One weak password may break security
  - Users choose bad passwords
  - Social engineering attacks, etc.
- Trudy has (almost) all of the advantages
- All of the math favors bad guys
- Passwords are a **BIG** security problem
  - And will continue to be a big problem

Password Cracking Tools

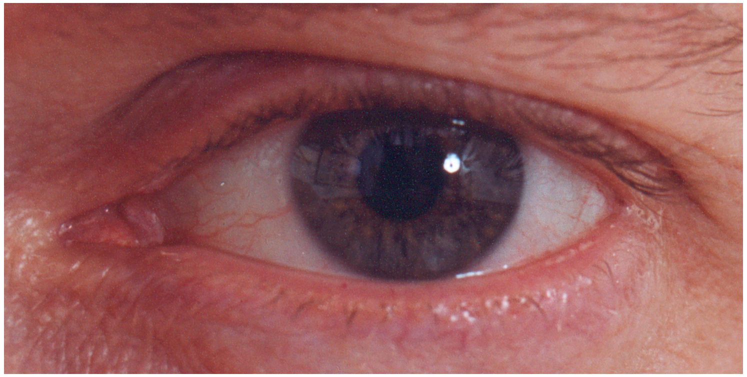
- Popular password cracking tools
  - Password Crackers
  - Password Portal
  - L0phtCrack and LC4 (Windows)
  - John the Ripper (Unix)
- Admins should use these tools to test for weak passwords since attackers will

Password Manager

- Password Manager?
  - Yes, please

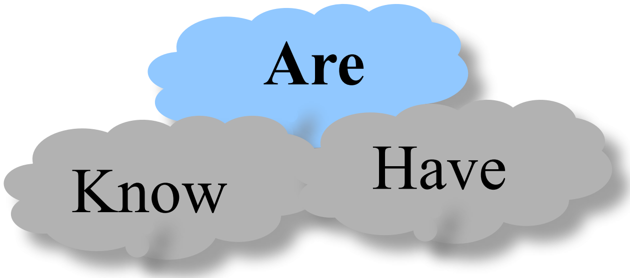
- https://keepass.info/download.html

Biometrics



Something You Are

- Biometric
  - **"You are your key"**—Schneier
- Examples
  - Fingerprint
  - Handwritten signature
  - Facial recognition
  - Speech recognition
  - Gait (walking) recognition
  - "Digital doggie" (odor recognition)
  - Many more!



Why Biometrics?

- More secure replacement for passwords
- Cheap and reliable biometrics needed
  - Today, an active area of research
- Biometrics **are** used in security today
  - Thumbprint mouse
  - Palm print for secure entry
  - Fingerprint to unlock car door, etc.
- But biometrics not too popular
  - Has not lived up to its promise (yet?)

Ideal Biometric

- **Universal** — applies to (almost) everyone
  - In reality, no biometric applies to everyone
- **Distinguishing** — distinguish with certainty
  - In reality, cannot hope for 100% certainty
- **Permanent** — physical characteristic being measured never changes
  - In reality, OK if it to remains valid for long time
- **Collectable** — easy to collect required data
  - Depends on whether subjects are cooperative
- Also, safe, user-friendly, etc., etc.

Biometric Modes

- **Identification** — Who goes there?
  - Compare **one-to-many**
  - Example: The FBI fingerprint database
- **Authentication** — Are you who you say you are?
  - Compare **one-to-one**
  - Example: Thumbprint mouse
- Identification problem is more difficult
  - More "random" matches since more comparisons
- We are interested in authentication

Enrollment vs Recognition

- Enrollment phase
  - Subject's biometric info put into database
  - Must carefully measure the required info
  - OK if slow and repeated measurement needed
  - Must be very precise
  - May be weak point of many biometric
- Recognition phase
  - Biometric detection, when used in practice
  - Must be quick and simple
  - But must be reasonably accurate

Cooperative Subjects?

- Authentication — cooperative subjects
- Identification — uncooperative subjects
- For example, facial recognition
  - Used in Las Vegas casinos to detect known cheaters (terrorists in airports, etc.)
  - Often do not have ideal enrollment conditions
  - Subject will try to confuse recognition phase
- Cooperative subject makes it much easier
  - We are focused on authentication
  - So, subjects are generally cooperative

Biometric Errors

- **Fraud rate** versus **insult rate**
  - Fraud —Trudy mis-authenticated as Alice
  - Insult —Alice not authenticated as Alice
- For any biometric, can decrease fraud or insult, but other one will increase
- For example
  - 99% voiceprint match ⇒low fraud, high insult
  - 30% voiceprint match ⇒ high fraud, low insult
- **Equal error rate:** rate where fraud == insult
  - A way to compare different biometrics

Fingerprint Comparison

- Examples of **loops**, **whorls**, and **arches**
- Minutia extracted from these features

Loop (double)

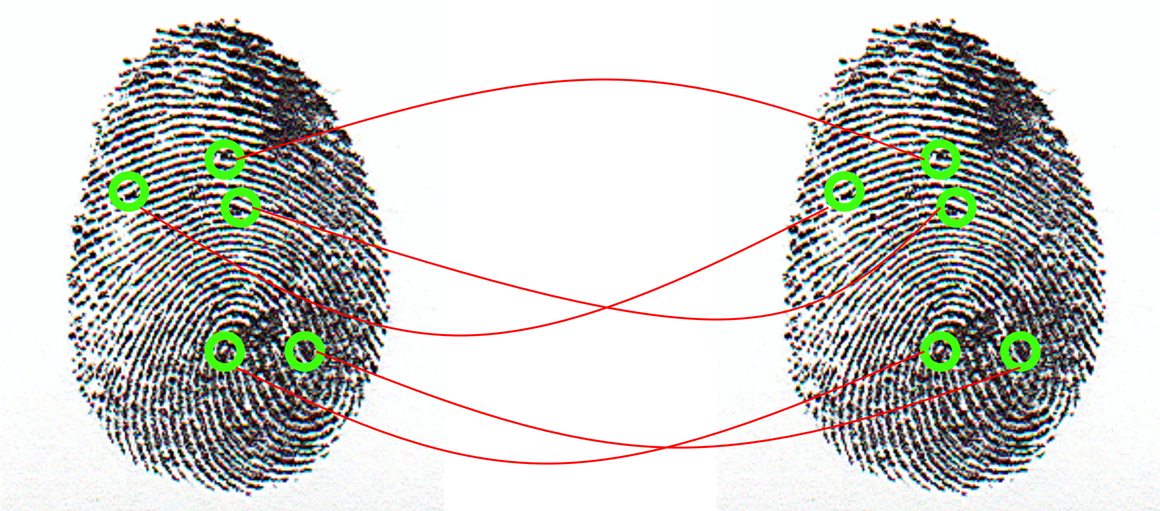
Whorl

Arch

Fingerprint: Enrollment

- Capture image of fingerprint
- Enhance image
- Identify points

Fingerprint: Recognition



- Extracted points are compared with information stored in a database
- Is it a statistical match? (with some pre-determined level of confidence)
- Aside: [Do identical twins' fingerprints differ]?

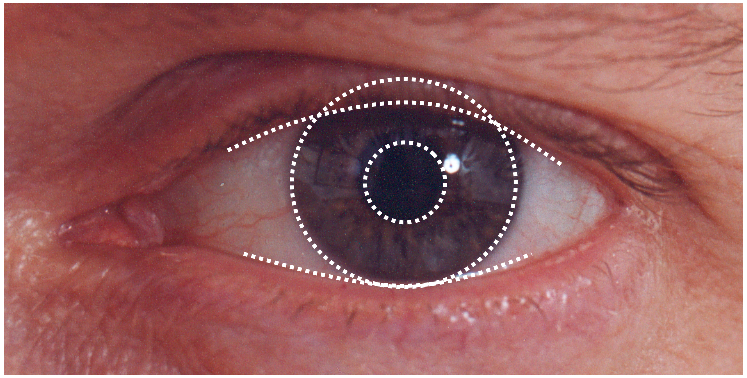
Hand Geometry

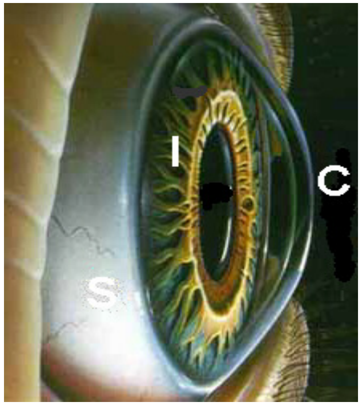
- A popular biometric
- Measures shape of hand
  - Width of hand, fingers
  - Length of fingers, etc.
- Human hands not unique
- Hand geometry sufficient for many situations
- OK for authentication
- Not useful for ID problem

Hand Geometry

- Advantages
  - Quick — 1 minute for enrollment, 5 seconds for recognition
  - Hands are symmetric — so what?
- Disadvantages
  - Cannot use on very young or very old
  - Relatively high equal error rate

Iris Patterns





- Iris pattern development is "chaotic"
- Little or no genetic influence
- Different even for identical twins
- Pattern is stable through lifetime

Attack on Iris Scan

- Good **photo** of eye can be scanned
  - Attacker could use photo of eye
- Afghan woman was authenticated by iris scan of old photo
  - Story is [here]
- To prevent attack, scanner could use light to be sure it is a "live" iris

Equal Error Rate Comparison

- Equal error rate (EER): fraud == insult rate
- **Fingerprint** biometric has EER of about 5%
- **Hand geometry** has EER of about 10^{-3}
- In theory, **iris scan** has EER of about 10^{-6}
  - But in practice, may be hard to achieve
  - Enrollment phase must be extremely accurate
- Most biometrics much worse than fingerprint!
- Biometrics useful for authentication…
  - …but identification biometrics almost useless today

Biometrics: The Bottom Line

- Biometrics are hard to forge
- But attacker could
  - Steal Alice's thumb
  - Photocopy Bob's fingerprint, eye, etc.
  - Subvert software, database, "trusted path" …
- And how to revoke a "broken" biometric?
- **Biometrics are not foolproof**
- Biometric use is limited today
- That should change in the (near?) future

Something You Have

- Something in your possession
- Examples include following…
  - Car key
  - Laptop computer (or MAC address)
  - Password generator
  - ATM card, smartcard, etc.

2-factor Authentication

- Requires any 2 out of 3 of
  - Something you know
  - Something you have
  - Something you are
- Examples
  - ATM: Card and PIN
  - Credit card: Card and signature
  - Password generator: Device and PIN
  - Smartcard with password/PIN

Authentication vs Authorization

- Authentication — Are you who you say you are?
  - Restrictions on who (or what) can access system
- **Authorization** — Are you allowed to do that?
  - Restrictions on actions of authenticated users
- Authorization is a form of **access control**
- Classic authorization enforced by
  - Access Control Lists (ACLs)
  - Capabilities (C-lists)

Lampson's Access Control Matrix

- **Subjects** (users) index the rows
- **Objects** (resources) index the columns

	OS	Accounting program	Accounting data	Insurance data	Payroll data
Bob	rx	rx	r	---	---
Alice	rx	rx	r	rw	rw
Sam	rwx	rwx	r	rw	rw
Accounting program	rx	rx	rw	rw	rw

Are You Allowed to Do That?

- **Access control matrix** has **all** relevant info
- Could be 1000's of users, 1000's of resources
- Then matrix with 1,000,000's of entries
- How to manage such a large matrix?
- Need to check this matrix before access to any resource is allowed
- How to make this efficient?

Access Control Lists (ACLs)

- ACL: store access control matrix by **column**
- Example: ACL for **insurance data** is in **blue**

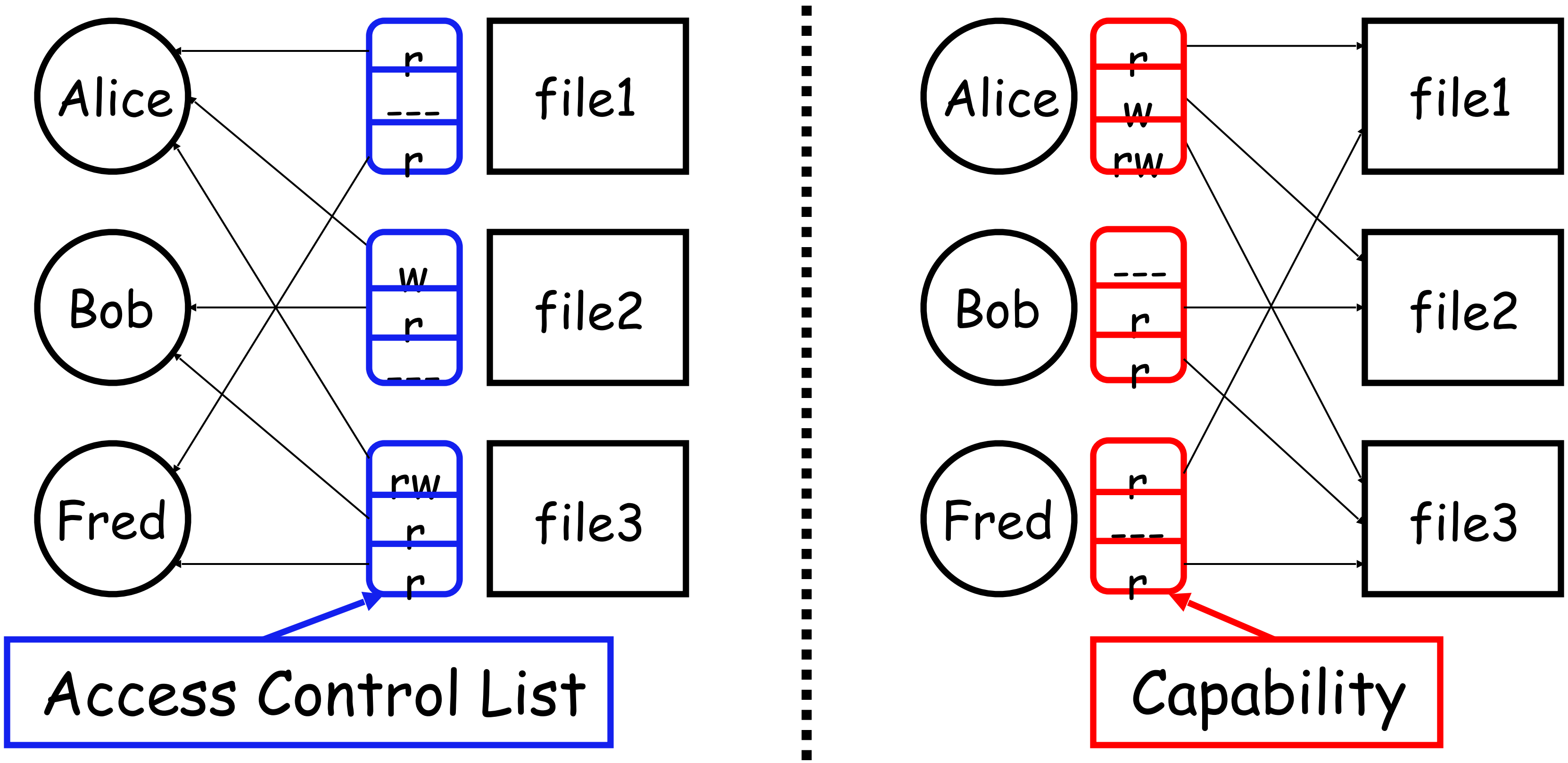
	OS	Accounting program	Accounting data	Insurance data	Payroll data
Bob	rx	rx	r	---	---
Alice	rx	rx	r	rw	rw
Sam	rwx	rwx	r	rw	rw
Accounting program	rx	rx	rw	rw	rw

Capabilities (or C-Lists)

- Store access control matrix by **row**
- Example: Capability for **Alice** is in **red**

	OS	Accounting program	Accounting data	Insurance data	Payroll data
Bob	rx	rx	r	---	---
Alice	**rx**	**rx**	**r**	**rw**	**rw**
Sam	rwx	rwx	r	rw	rw
Accounting program	rx	rx	rw	rw	rw

ACLs vs Capabilities



- Note that arrows point in opposite directions…
- With ACLs, still need to associate users to files

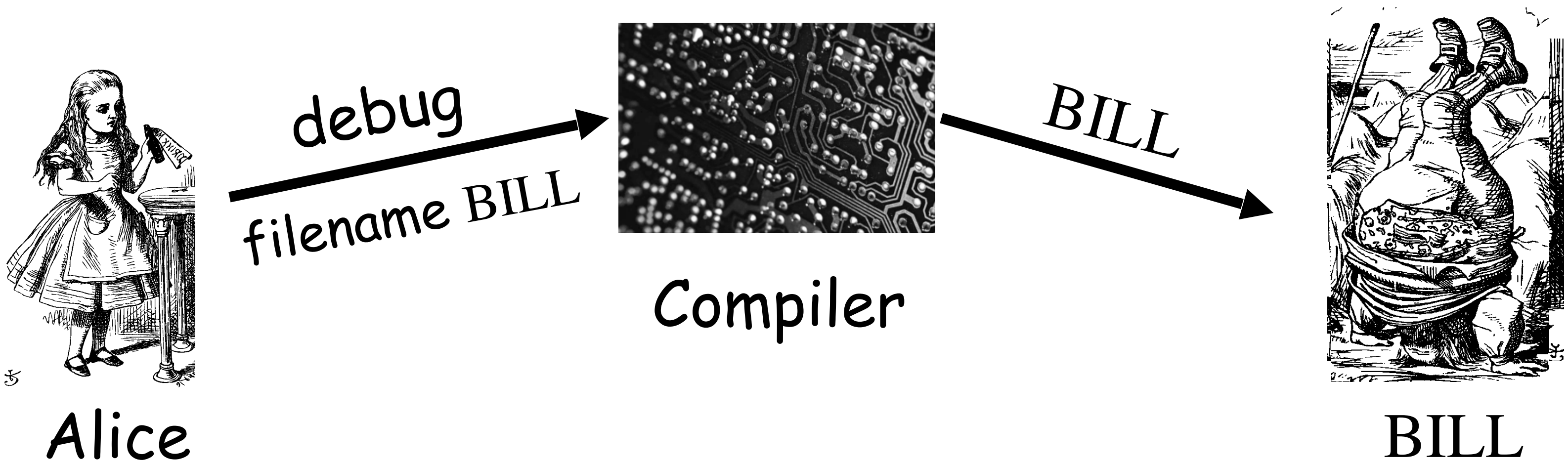
Confused Deputy

- Two resources
  - Compiler and BILL file (billing info)
- Compiler can write file BILL
- Alice can invoke compiler with a debug filename
- Alice not allowed to write to BILL

- Access control matrix

	Compiler	BILL
Alice	x	r
Compiler	rx	rw

ACL's and Confused Deputy



- Compiler is **deputy** acting on behalf of Alice
- Compiler is **confused**
  - Alice is not allowed to write/overwrite BILL
- Compiler has confused its rights with Alice's

Confused Deputy

- Compiler acting for Alice is confused
- There has been a separation of **authority** from the **purpose** for which it is used
- With ACLs, difficult to avoid this problem
- With Capabilities, easier to prevent problem
  - Must maintain association between authority and intended purpose
  - Capabilities make it easy to **delegate** authority

ACLs vs Capabilities

- ACLs
  - Good when users manage their own files
  - Protection is data-oriented
  - Easy to change rights to a resource
- Capabilities
  - Easy to delegate---avoid the confused deputy
  - Easy to add/delete users
  - More difficult to implement
- Capabilities loved by academics (they are more secure!)
  - Capability Myths Demolished

Multilevel Security (MLS) Models

Classifications and Clearances

- **Classifications** apply to **objects**
- **Clearances** apply to **subjects**
- US Department of Defense (DoD) uses 4 levels:

  TOP SECRET

  SECRET

  CONFIDENTIAL

  UNCLASSIFIED

Clearances and Classification

- Practical classification problems
  - Proper classification not always clear
    - Users might have different views
  - Level of granularity to apply classifications
    - It's possible to construct a document where each paragraph, taken individually, is UNCLASSIFIED, but the overall document is TOP SECRET
  - Aggregation — flipside of granularity
    - Discover TOP SECRET info from a careful analysis of UNCLASSIFIED documents

Subjects and Objects

- Let O be an **object**, S a **subject**
  - O has a classification
  - S has a clearance
  - Security **level** denoted L(O) and L(S)
- For DoD levels, we have the equation:

TOP SECRET>SECRET>CONFIDENTIAL >UNCLASSIFIED

MLS Applications

- Classified government/military systems
- **Business example:** info restricted to
  - Senior management only, all management, everyone in company, or general public
- Network firewall (keep intruder at low level to limit the damage)
- Confidential medical info, databases, etc.

MLS Security Models

- MLS models explain **what** needs to be done
- Models **do not** tell you **how** to implement
- Models are descriptive, not prescriptive
  - That is, high level description, not an algorithm
- There are many MLS models
- We'll discuss simplest MLS model
  - Other models are more realistic
  - Other models also more complex, more difficult to enforce, harder to verify, etc.

Bell-LaPadula

- BLP security model designed to express essential requirements for MLS
- BLP deals with **confidentiality**
  - To prevent unauthorized reading
- Recall that O is an object, S a subject
  - Object O has a classification
  - Subject S has a clearance
  - Security level denoted L(O) and L(S)

Bell-LaPadula

- BLP consists of
  - **Simple Security Condition**: S can read O if and only if $L(O) \leq L(S)$
  - ***-Property** (**Star Property**): S can write O if and only if $L(S) \leq L(O)$

  TOP SECRET info cannot be written in SECRET document
- **No read up, no write down**

McLean's Criticisms of BLP

- McLean: BLP is "so trivial that it is hard to imagine a realistic security model for which it does not hold"
- McLean's "system Z" allowed administrator to reclassify object, then "write down"
- Is this fair?
- Violates spirit of BLP, but **not** expressly forbidden in statement of BLP
- Raises fundamental questions about the nature of (and limits of) modeling

B and LP's Response

- BLP enhanced with **tranquility property**
  - Strong tranquility: security labels never change
  - Weak tranquility: security label can only change if it does not violate "established security policy"
- Strong tranquility impractical in real world
  - Often want to enforce "**least privilege**"
  - Give users lowest privilege for current work
  - Then upgrade as needed (and allowed by policy)
  - This is known as the **high water mark** principle
- Weak tranquility allows for **least privilege** (high water mark), but the property is vague

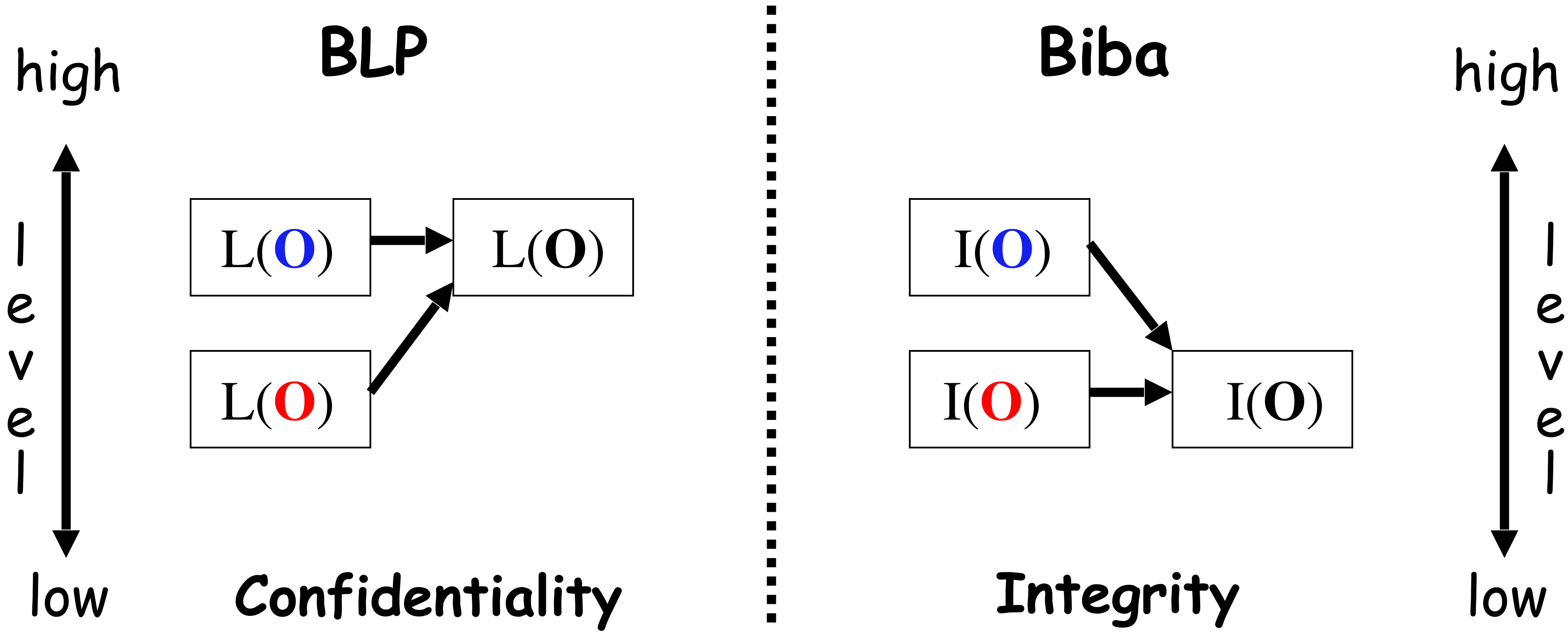
Biba's Model

- BLP for confidentiality, Biba for **integrity**
  - Biba is to prevent unauthorized writing
- Biba is (in a sense) the dual of BLP
- Integrity model
  - Spse you trust the integrity of **O** but not **O**
  - If object **O** includes **O** and **O** then you cannot trust the integrity of **O**
- Integrity level of O is minimum of the integrity of any object in O
- **Low water mark** principle for integrity

Biba

- Let I(O) denote the integrity of object O and I(S) denote the integrity of subject S
- Biba can be stated as
  - **Write Access Rule:** S can write O if and only if $I(O) \leq I(S)$
    - (if S writes O, the integrity of O $\leq$ that of S)
  - **Biba's Model:** S can read O if and only if $I(S) \leq I(O)$
    - (if S reads O, the integrity of S $\leq$ that of O)
- Often, replace Biba's Model with
  - **Low Water Mark Policy:** If S reads O, then $I(S) = \min(I(S), I(O))$

BLP vs Biba



High water mark principle

Low water mark principle

Compartments

Compartments

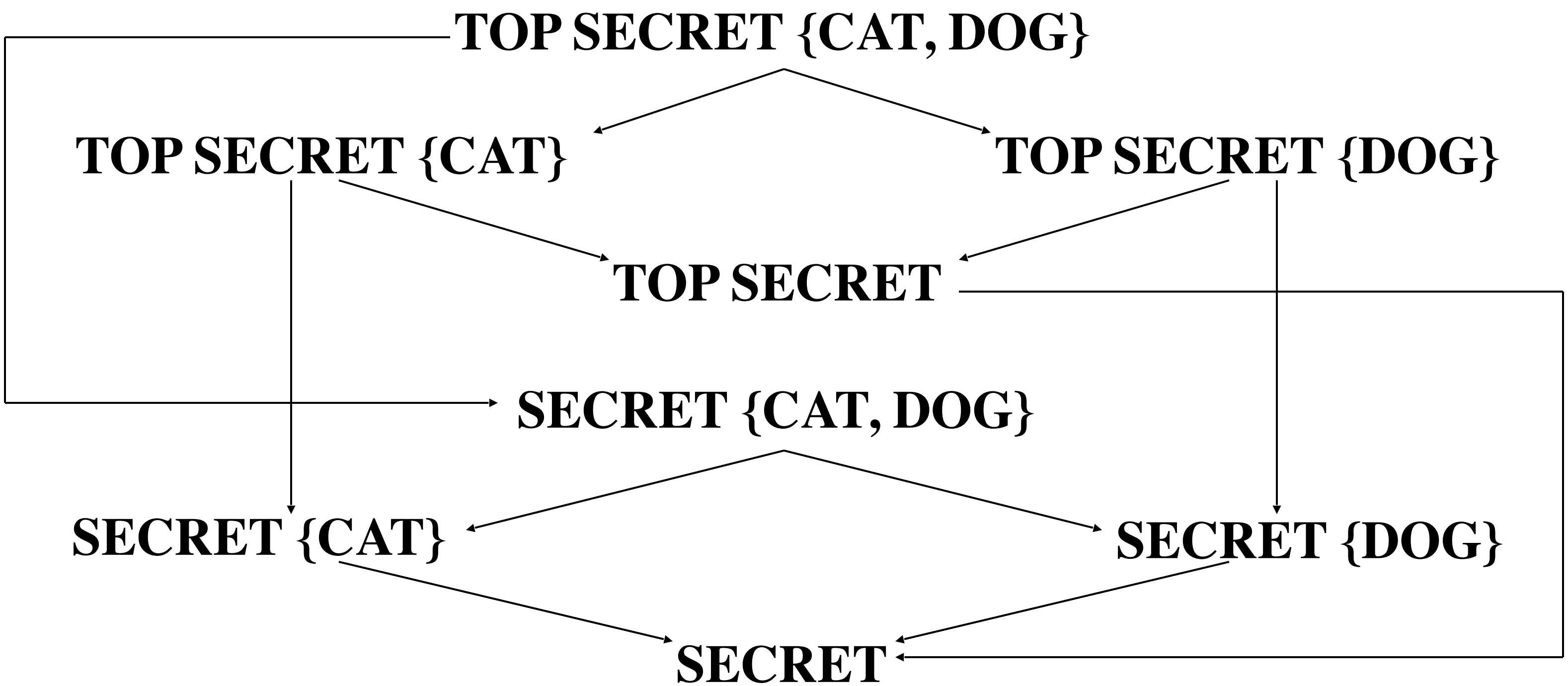
- Multilevel Security (MLS) enforces access control **up and down**
- Simple hierarchy of security labels is generally notflexible enough
- Compartments enforces restrictions **across**
- Suppose **TOP SECRET** divided into **TOP SECRET {CAT}** and **TOP SECRET {DOG}**
- Both are **TOP SECRET** but information flow restricted across the **TOP SECRET** level
- You need a specific Clearance to access them

Compartments

- Why compartments?
  - Why not create a new classification level?
  - For example, the info could be not comparable
- May not want either of
  - **TOP SECRET {CAT}≥TOP SECRET {DOG}**
  - **TOP SECRET {DOG}≥TOP SECRET {CAT}**
  - **These equations may not hold**
- Compartments designed to enforce the **need to know** principle
  - Regardless of clearance, you only have access to info that you need to know to do your job, not all of them (example, all the TOP SECRET info)

Compartments

- Arrows indicate "≥" relationship



- Not all classifications are comparable, e.g.,

TOP SECRET {CAT} vs SECRET {CAT, DOG}

MLS vs Compartments

- MLS can be used without compartments
  - And vice-versa
- But, MLS almost always uses compartments
- Example
  - MLS mandated for protecting medical records of British Medical Association (BMA)
  - AIDS was **TOP SECRET**; prescriptions **SECRET**
  - But, everyone with SECRET clearance can deduce the AIDS patients by the prescriptions!!
  - Everything tends toward **TOP SECRET**
  - Defeats the purpose of the system!
- Compartments-only approach used instead

Covert Channel

Covert Channel

- MLS designed to restrict legitimate channels of communication
- May be other ways for information to flow
- For example, resources shared at different levels could be used to "signal" information
- **Covert channel**: a communication path not intended as such by system's designers

Covert Channel Example

- Alice has **TOP SECRET** clearance, Bob has **CONFIDENTIAL** clearance
- Suppose the file space shared by all users
- Alice creates file FileXYzW to signal "1" to Bob, and removes file to signal "0"
- Once per minute Bob lists the files
  - If file FileXYzW does not exist, Alice sent 0
  - If file FileXYzW exists, Alice sent 1
- Alice can leak **TOP SECRET** info to Bob!

Covert Channel Example

Alice:	Create file	Delete file	Create file		Delete file
Bob:	Check file	Check file	Check file	Check file	Check file
Data:	1	0	1	1	0
Time:					

Covert Channel

- Other possible covert channels?
  - Print queue
  - ACK messages
  - Network traffic, etc.
- When does covert channel exist?
  1. Sender and receiver have a shared resource
  2. Sender able to vary some property of resource that receiver can observe
  3. "Communication" between sender and receiver can be synchronized

Covert Channel

- So, covert channels are everywhere
- "Easy" to eliminate covert channels:
  - Eliminate all shared resources…
  - …and all communication
- Virtually impossible to eliminate covert channels in any useful system
  - DoD guidelines: **reduce covert channel capacity** to no more than 1 bit/second
  - Implication? DoD has given up on eliminating covert channels!

Covert Channel

- Consider 100MB **TOP SECRET** file
  - Plaintext stored in **TOP SECRET** location
  - Ciphertext (encrypted with AES using 256-bit key) stored in **UNCLASSIFIED** location
- Suppose we reduce covert channel capacity to 1 bit per second
- It would take more than 25 years to leak entire document thru a covert channel
- But it would take less than 5 minutes to leak 256-bit AES key thru covert channel!

Inference Control

Inference Control Example

- Suppose we query a database
  - Question: What is average salary of Italian CS professors at SJSU?
  - Answer: $1,000,000
  - Question: How many Italian CS professors at SJSU?
  - Answer: 1
- Specific information has leaked from responses to general questions!

Inference Control and Research

- For example, medical records are private but valuable for research
- How to make info available for research and protect privacy?
- How to allow access to such data without leaking specific information?

Naïve Inference Control

- Remove names from medical records?
- Still may be easy to get specific info from such "anonymous" data
- Removing names is not enough
  - As seen in previous example
- What more can be done?

Less-naïve Inference Control

- Query set size control
  - Don't return an answer if set size is too small
  - …but research on rare medical conditions?
- N-respondent, k% dominance rule
  - Do not release statistic if k% or more contributed by N or fewer
  - Example: Avg salary in Bill Gates' neighborhood
  - This approach used by US Census Bureau

Less-naïve Inference Control

- Randomization
  - Add small amount of random noise to data
  - … but problems with rare medical conditions
- Many other methods — none satisfactory… for now.

Inference Control

- Robust inference control may be impossible
- Is weak inference control better than nothing?
  - **Yes**: Reduces amount of information that leaks
- Is weak covert channel protection better than nothing?
  - **Yes**: Reduces amount of information that leaks
- Is weak crypto better than no crypto?
  - **Probably not**
  - Encryption indicates important data. May be easier to filter encrypted data

CAPTCHA

Turing Test

- Proposed by Alan Turing in 1950
- Human asks questions to another human and a computer, without seeing either
- If questioner cannot distinguish human from computer, computer passes the test
- The **gold standard** in artificial intelligence
- No computer can pass this today
  - But some claim to be close to passing

CAPTCHA

- CAPTCHA
  - Completely Automated Public Turing test to tell Computers and Humans Apart
- Automated — test is generated and scored by a computer program
- Public — program and data are public
- Turing test to tell… — humans can pass the test, but machines cannot pass
  - Also known as HIP == Human Interactive Proof
- Like an inverse Turing test (well, sort of…)

CAPTCHA Paradox?

- "…CAPTCHA is a program that can generate and grade tests that it itself cannot pass…"
  - "…much like some professors…"
- Paradox — computer creates and scores test that it cannot pass!
- CAPTCHA used so that only humans can get access (i.e., no bots/computers)
- CAPTCHA is for **access control**

CAPTCHA Uses?

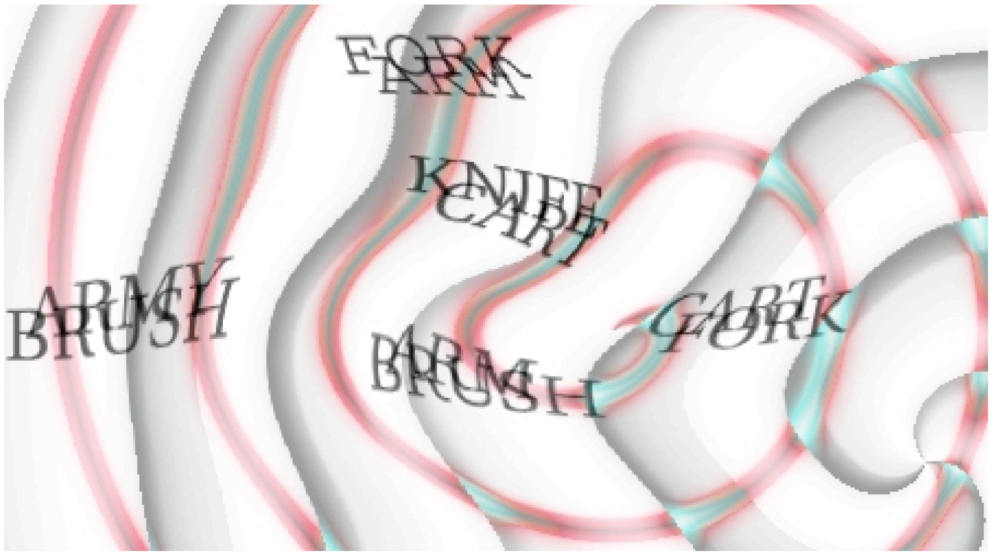
- Original motivation: automated bots stuffed ballot box in vote for best CS grad school
  - SJSU vs Stanford?
- Free email services — spammers like to use bots to sign up for 1000's of email accounts
  - CAPTCHA employed so only humans get accounts
- Sites that do not want to be automatically indexed by search engines
  - CAPTCHA would force human intervention

CAPTCHA: Rules of the Game

- Easy for most humans to pass
- Difficult or impossible for machines to pass
  - **Even with access to CAPTCHA software**
- From Trudy's perspective, the only unknown is a random number
  - Analogous to Kerckhoffs' Principle (Encryption Algorithm is know, only Key unknown)
- Desirable to have different CAPTCHAs in case some person cannot pass one type
  - Blind person could not pass visual test, etc.

Do CAPTCHAs Exist?

- Test: Find 2 words in the following



- Easy for most humans
- A (difficult?) OCR problem for computer
  - OCR == Optical Character Recognition

CAPTCHAs

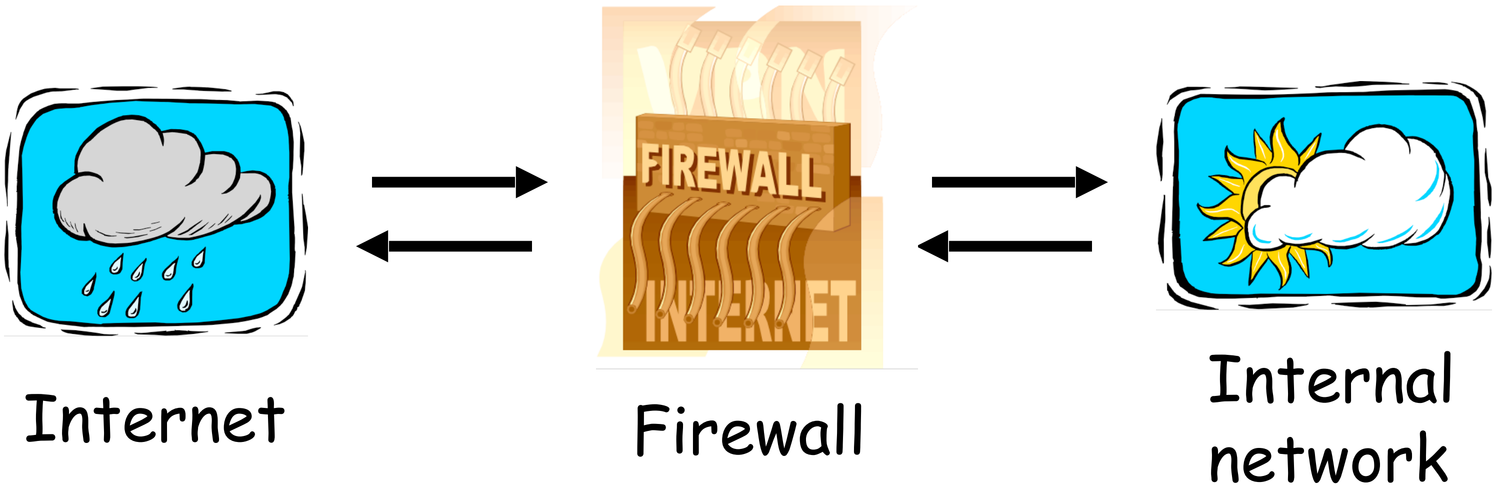
- Current types of CAPTCHAs
  - Visual — like previous example
  - Audio — distorted words or music
- No text-based CAPTCHAs
  - No Captcha reCAPTCHA from Google

CAPTCHA's and AI

- OCR is a challenging AI problem
  - Hard part is the **segmentation problem** (the ability to separate one letter from another)
  - Humans good at solving this problem
- Distorted sound makes good CAPTCHA
  - Humans also good at solving this
- Hackers who break CAPTCHA have solved a hard AI problem
  - So, putting hacker's effort to good use!
- Other ways to defeat CAPTCHAs???

Firewalls

Firewalls



- Firewall decides what to let in to internal network and/or what to let out
- **Access control** for the network

Firewall as Secretary

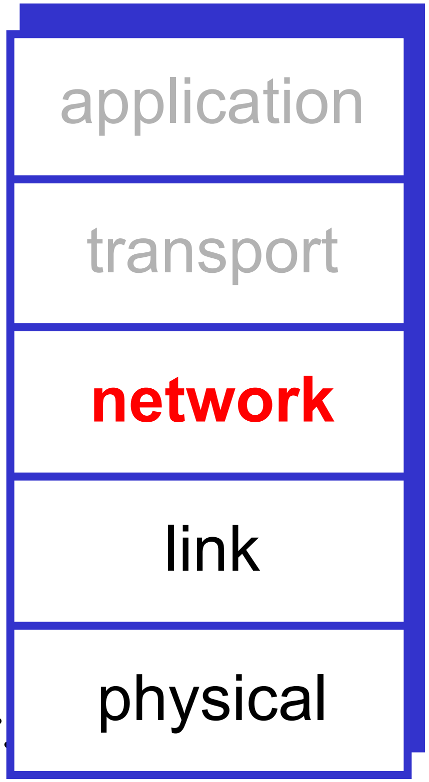
- A firewall is like a **sécretary**
- To meet with an executive
  - First contact the secretary
  - Secretary decides if meeting is important
  - So, secretary filters out many requests
- You want to meet chair of CS department?
  - Secretary does some filtering
- You want to meet the President of the US?
  - Secretary does lots of filtering

Firewall Terminology

- No standard firewall terminology
- Types of firewalls
  - **Packet filter** — works at network layer
  - **Stateful packet filter** — transport layer
  - **Application proxy** — application layer
- Other terms often used
  - E.g., “deep packet inspection”

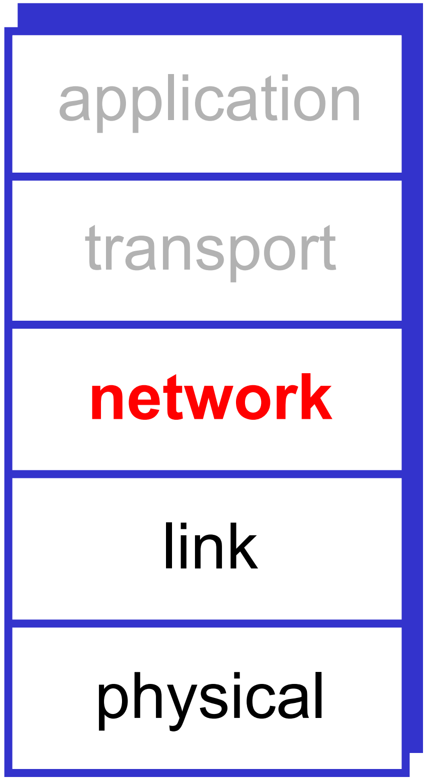
Packet Filter

- Operates at network layer
- Can filters based on…
  - Source IP address
  - Destination IP address
  - Source Port
  - Destination Port
  - Flag bits (SYN, ACK, etc.)
  - Egress or ingress (different filtering rules for incoming and outgoing packets)



Packet Filter

- Advantages?
  - Efficiency
    - packets only need to be processed up to the network layer
    - only header information is examined
- Disadvantages?
  - No concept of state (each packet is independent)
  - Cannot see TCP connections
  - Blind to application data (where many malware reside!!)



Packet Filter

- Configured via Access Control Lists (ACLs)
  - Different meaning than the one previously seen

Action	Source IP	Dest IP	Source Port	Dest Port	Protocol	Flag Bits
Allow	Inside	Outside	Any	80	HTTP	Any
Allow	Outside	Inside	80	> 1023	HTTP	ACK
Deny	All	All	All	All	All	All

- **Q**: Intention?
- **A**: Restrict traffic to Web browsing
- Outbound Web traffic (port 80) allowed
- Restrict incoming packets to Web responses

Port Scanning

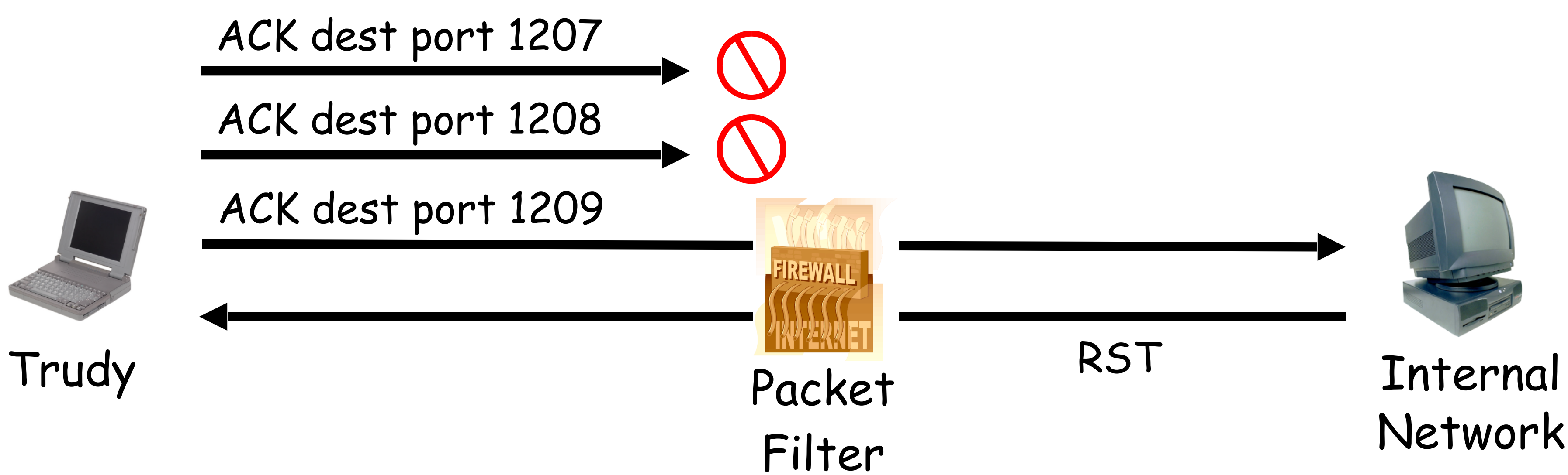
- An attacker has to:

1. Check which ports are listening
2. Recognize the service using that port
3. Use a specific exploit to attack that service

TCP ACK Scan

- Attacker scans for open ports thru firewall
  - Port scanning is first step in many attacks
- Attacker sends packet with ACK bit set, **without** prior 3-way handshake
  - Violates TCP/IP protocol (the initial packet must have the SYN bit set)
  - ACK packet pass thru packet filter firewall
  - Appears to be part of an ongoing connection (no concept of state)
  - The recipient recognizes that the packet is not part of an established connection and respond with a RST packet to terminate the connection

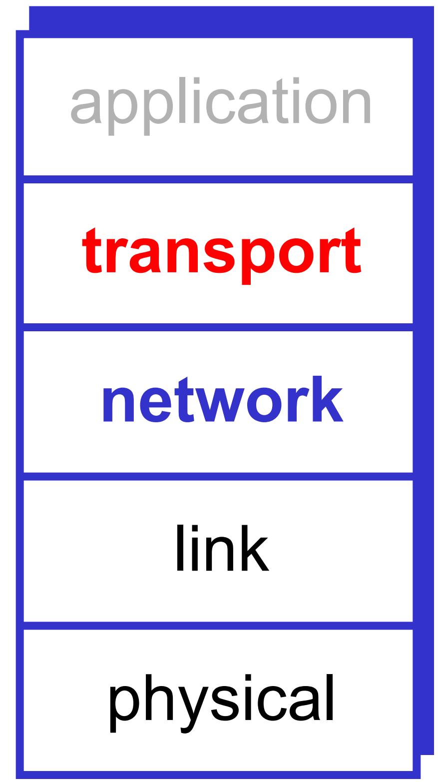
TCP ACK Scan



- Attacker knows port 1209 open thru firewall
- A **stateful packet filter** can prevent this
  - Since scans not part of established connections

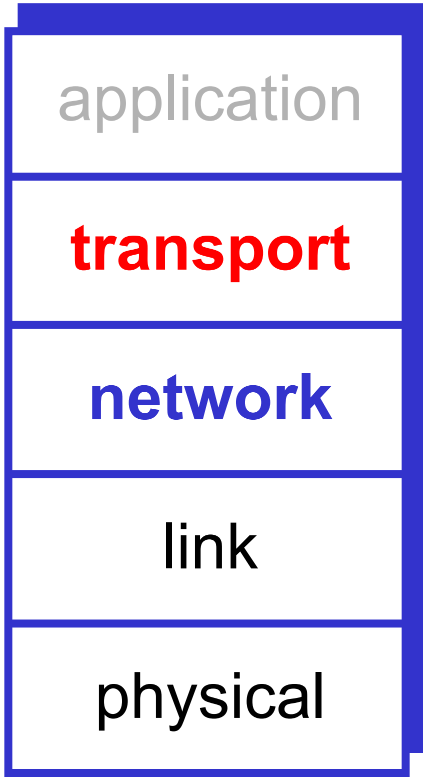
Stateful Packet Filter

- Adds **state** to packet filter
- Operates at transport layer
- **Remembers** TCP connections, flag bits, etc.
- Can even remember UDP packets (e.g., DNS requests)



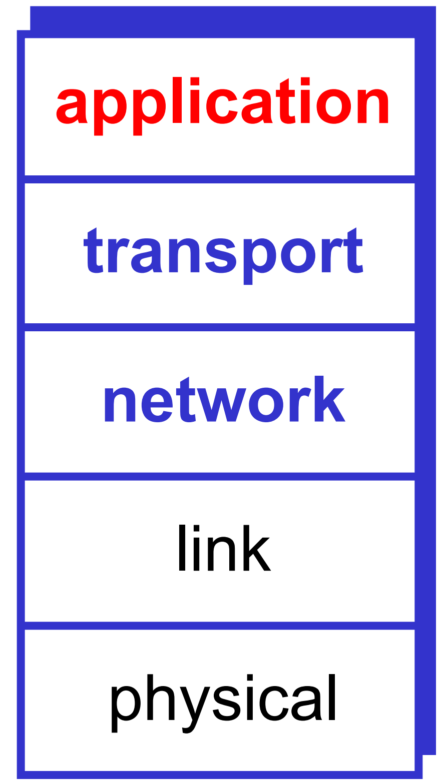
Stateful Packet Filter

- Advantages?
  - Can do everything a packet filter can do plus…
  - Keep track of ongoing connections (so prevents TCP ACK scan)
- Disadvantages?
  - Cannot see application data
  - Slower than packet filtering



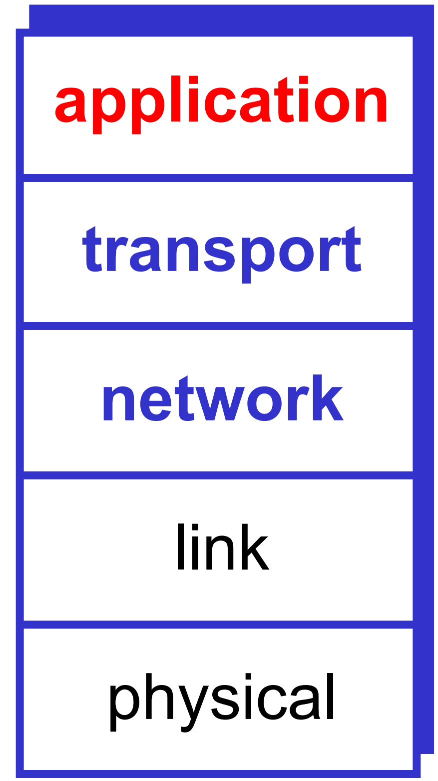
Application Proxy

- A **proxy** is something that acts on your behalf
- Application proxy looks at incoming application data
- Verifies that data is safe before letting it in



Application Proxy

- Advantages?
  - Complete view of connections and applications data
  - Filter bad data at application layer (viruses)
- Disadvantages?
  - Speed

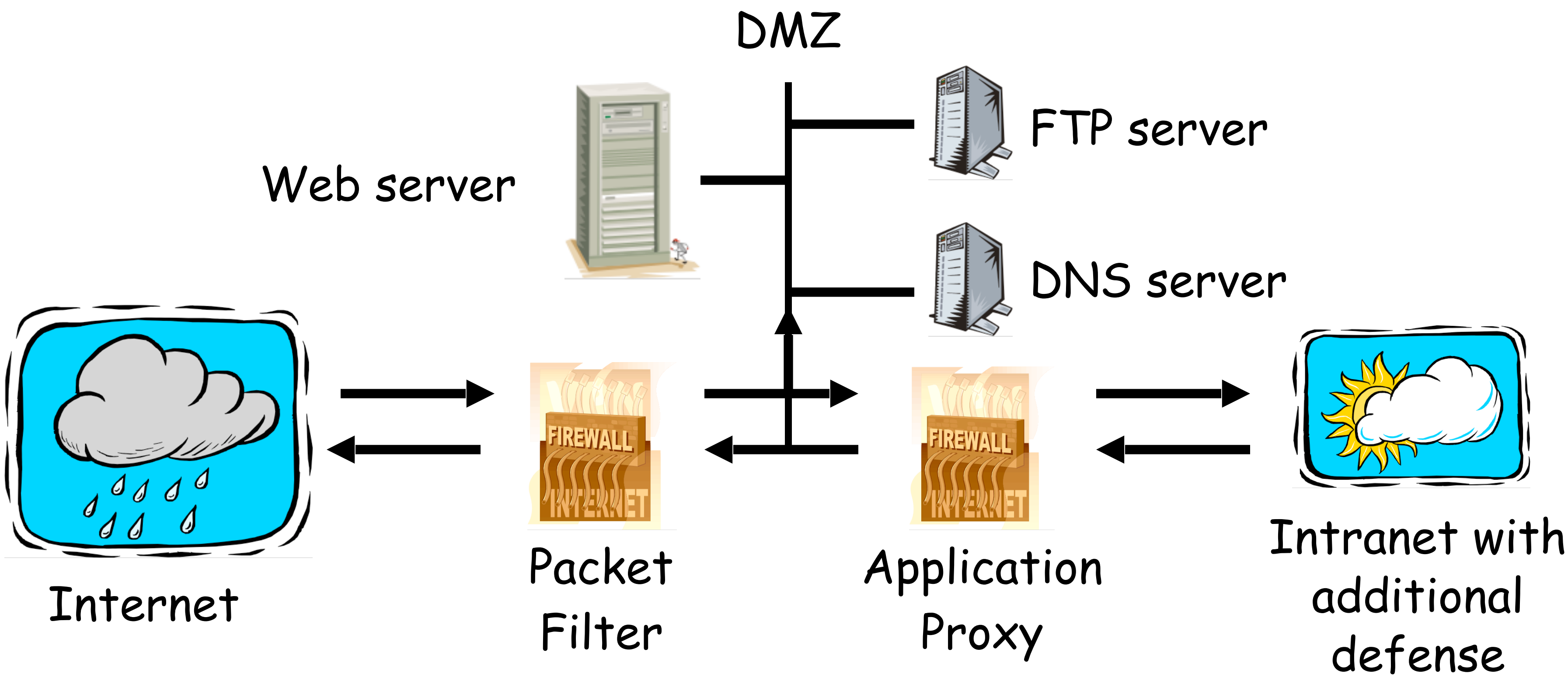


Application Proxy

- Creates a new packet before sending it thru to internal network
- Attacker must talk to **proxy** and convince it to forward message
- Proxy has complete view of connection
- Prevents some scans stateful packet filter cannot (destroying and recreating packets)

Firewalls and Defense in Depth

- Typical network security architecture



Intrusion Detection Systems

Intrusion Detection

- In spite of intrusion prevention, bad guys will sometime get in
- Intrusion detection systems (**IDS**)
  - Detect attacks in progress (or soon after)
  - Look for unusual or suspicious activity
- IDS evolved from log file analysis
- IDS is currently a **hot** research topic
- How to respond when intrusion detected?
  - We don't deal with this topic here…

Intrusion Detection Systems

- Who is likely intruder?
  - May be outsider who got thru firewall
  - May be evil insider
- What do intruders do?
  - Launch well-known attacks
  - Launch variations on well-known attacks
  - Launch new/little-known attacks
  - "Borrow" system resources
  - Use compromised system to attack others. etc.

IDS

- Intrusion detection **approaches**
  - Signature-based IDS
  - Anomaly-based IDS
- Intrusion detection **architectures**
  - Host-based IDS
    - Buffer overflow, escalation of privilege, …
  - Network-based IDS
    - DoS, Network Probes, …
- Any IDS can be classified as above
  - In spite of marketing claims to the contrary!

Signature Detection Example

- Failed login attempts may indicate password cracking attack
- IDS could use the rule "N failed login attempts in M seconds" as **signature**
- If N or more failed login attempts in M seconds, IDS warns of attack
- Note that such a warning is specific
  - Admin knows what attack is suspected
  - Easy to verify attack (or false alarm)

Signature Detection

- Suppose IDS warns whenever N or more failed logins in M seconds
  - Set N and M so false alarms not common
  - Can do this based on "normal" behavior
- But, if Trudy knows the signature, she can try N−1 logins every M seconds…
- Then signature detection slows down Trudy, but might not stop her

Signature Detection

- Many techniques used to make signature detection more robust
- Goal is to detect "**almost**" signatures
- For example, if "about" N login attempts in "about" M seconds
  - Warn of possible password cracking attempt
  - What are reasonable values for "about"?
  - Can use statistical analysis, heuristics, etc.
  - Must not increase false alarm rate too much

Signature Detection

- Advantages of signature detection
  - Simple
  - Detect known attacks
  - Know which attack at time of detection
  - Efficient (if reasonable number of signatures)
- Disadvantages of signature detection
  - Signature files must be kept up to date
  - Number of signatures may become large
  - Can only detect known attacks
  - Variation on known attack may not be detected

Anomaly Detection

- Anomaly detection systems look for unusual or abnormal behavior
- There are (at least) two challenges
  - What is normal for this system?
  - How "far" from normal is abnormal?
- No avoiding statistics here!
  - **mean** defines normal
  - **variance** gives distance from normal to abnormal

How to Measure Normal?

- How to measure normal?
  - Must measure during "representative" behavior
  - Must not measure during an attack…
  - …or else attack will seem normal!
  - Normal is statistical **mean**
  - Must also compute **variance** to have any reasonable idea of abnormal

How to Measure Abnormal?

- Abnormal is relative to some "normal"
  - Abnormal indicates possible attack
- Statistical discrimination techniques include
  - Bayesian statistics
  - Linear discriminant analysis (LDA)
  - Quadratic discriminant analysis (QDA)
  - Neural nets, hidden Markov models (HMMs), etc.
- Fancy modeling techniques also used
  - Machine Learning
  - Artificial immune system principles
  - Many, many, many others

Anomaly Detection (1)

- Spse we monitor use of three commands:

open, read, close

- Under normal use we observe Alice:

open, read, close, open, open, read, close, …

- Of the six possible ordered pairs, we see four pairs are normal for Alice,

(open,read), (read,close), (close,open), (open,open)

- Can we use this to identify unusual activity?

Anomaly Detection (1)

- We monitor use of the three commands

open, read, close

- If the ratio of abnormal to normal pairs is "too high", warn of possible attack
- Could improve this approach by
  - Also use expected frequency of each pair
  - Use more than two consecutive commands
  - Include more commands/behavior in the model
  - More sophisticated statistical discrimination

Anomaly Detection (2)

- Over time, Alice has accessed file F_n at rate H_n

H_0	H_1	H_2	H_3
.10	.40	.40	.10

- Recently, "Alice" has accessed F_n at rate A_n

A_0	A_1	A_2	A_3
.10	.40	.30	.20

- Is this normal use for Alice?
- We compute $S = (H_0-A_0)^2+(H_1-A_1)^2+\ldots+(H_3-A_3)^2 = .02$
  - We consider $S < 0.1$ to be normal, so this is normal
- How to account for use that varies over time?

Anomaly Detection (2)

- To allow "normal" to adapt to new use, we update averages: $H_n = 0.2A_n + 0.8H_n$
- In this example, H_n are updated… $H_2 = .2*.3+.8*.4 = .38$ and $H_3 = .2*.2+.8*.1 = .12$
- And we now have

H_0	H_1	H_2	H_3
.10	.40	.38	.12

Anomaly Detection (2)

- The updated long term average is

H_0	H_1	H_2	H_3
.10	.40	.38	.12

- Suppose new observed rates…

A_0	A_1	A_2	A_3
.10	.30	.30	.30

- Is this normal use?
- Compute $S = (H_0-A_0)^2+\ldots+(H_3-A_3)^2 = .0488$
  - Since $S = .0488 < 0.1$ we consider this normal
- And we again update the long term averages:

$$H_n = 0.2A_n + 0.8H_n$$

Anomaly Detection (2)

- The starting averages were:

H_0	H_1	H_2	H_3
.10	.40	.40	.10

- After 2 iterations, averages are:

H_0	H_1	H_2	H_3
.10	.38	.364	.156

- Statistics slowly evolve to match behavior
- This reduces false alarms for SA
- But also opens an avenue for attack…
  - Suppose Trudy **always** wants to access F_3
  - Can she convince IDS this is normal for Alice?

Anomaly Detection (2)

- To make this approach more robust, must incorporate the variance

- Can also combine N stats S_i as, say,

$$T = (S_1 + S_2 + S_3 + \ldots + S_N) / N$$

to obtain a more complete view of "normal"

Anomaly Detection Issues

- Systems constantly evolve and so must IDS
  - Static system would place huge burden on admin
  - But evolving IDS makes it possible for attacker to (slowly) convince IDS that an attack is normal
  - Attacker may win simply by "going slow"
- What does "abnormal" really mean?
  - Indicates there may be an attack
  - Might not be any specific info about "attack"
  - How to respond to such vague information?
  - In contrast, signature detection is very specific

Anomaly Detection

- Advantages?
  - Chance of detecting unknown attacks
- Disadvantages?
  - Cannot use anomaly detection alone…
  - …must be used with signature detection
  - Reliability is unclear
  - May be subject to attack
  - Anomaly detection indicates "something unusual", but lacks specific info on possible attack

Anomaly Detection: The Bottom Line

- Anomaly-based IDS is active research topic
- Many security experts have high hopes for its ultimate success
- Often cited as key future security technology
- Hackers are not convinced!
  - Title of a talk at Defcon: "Why Anomaly-based IDS is an Attacker's Best Friend"
- Anomaly detection is difficult and tricky
- As hard as AI?

Access Control Summary

- Authentication and authorization
  - Authentication — who goes there?
    - Passwords — something you know
    - Biometrics — something you are (you are your key)
    - Something you have

Access Control Summary

- Authorization — are you allowed to do that?
  - Access control matrix/ACLs/Capabilities
  - MLS/Multilateral security
  - BLP/Biba
  - Covert channel
  - Inference control
  - Firewalls
  - IDS

Coming Attractions…

- Security protocols
    - Generic authentication protocols
    - SSH
    - SSL
    - IPSec
    - Kerberos
    - …
- We'll see lots of crypto applications in the protocol part