Access Control

Access Control

- Two parts to access control
- **Authentication:** Who goes there?
  - Determine whether access is allowed
  - Authenticate human to machine
  - 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

Authentication

Who Goes There?

- How to authenticate a human to 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.

Trouble with Passwords

- "Passwords are one of the biggest practical problems facing security engineers today."
- "Humans are incapable of securely storing high-quality cryptographic keys, and they have unacceptable speed and accuracy when performing cryptographic operations. (They are also large, expensive to maintain, difficult to manage, and they pollute the environment. It is astonishing that these devices continue to be manufactured and deployed.)"

Why Passwords?

- Why is "something you know" more popular than "something you have" and "something you are"?
- **Cost**: passwords are free
- **Convenience**: easier for SA to reset pwd than to issue user 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
  - 0986437653726 3
  - POkemON
  - FSa7Yago
  - OnceuPOnAt1m8
  - 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
  - Outsider → normal user → administrator
  - May only require **one** weak password!

Password Retry

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

Password File

- Bad idea to store passwords in a file
- But need a way to verify passwords
- Cryptographic solution: **hash** the passwords
  - Store y = h(password)
  - Can verify entered password by hashing
  - If attacker obtains password file, he does not obtain passwords
  - But attacker with password file can guess x and check whether y = h(x)
  - If so, attacker has found password!

Dictionary Attack

- Attacker pre-computes h(x) for all x in a **dictionary** of common passwords
- Suppose attacker gets access to password file containing hashed passwords
  - Attacker only needs to compare hashes to his pre-computed dictionary
  - Same attack will work each time
- Can we prevent this attack? Or at least make attacker's job more difficult?

Password File

- Store hashed passwords
- Better to hash with **salt**
- Given password, choose random s, compute

  $$y = h(\text{password}, s)$$

  and store the pair (s,y) in the password file
- Note: The salt s is **not secret**
- Easy to verify password
- Attacker must recompute dictionary hashes for each user — lots more work!

Password Cracking: Do the Math

- Assumptions
- Pwds are 8 chars, 128 choices per character
  - Then $128^8 = 2^{56}$ possible passwords
- There is a **password file** with 2^{10} pwds
- Attacker has **dictionary** of 2^{20} common pwds
- Probability of 1/4 that a pwd is in dictionary
- **Work** is measured by number of hashes

Password Cracking

- Attack 1 password without dictionary
  - Must try $2^{56}/2 = 2^{55}$ on average
  - Just like exhaustive key search
- Attack 1 password with dictionary
  - Expected work is about

$$1/4\ (2^{19}) + 3/4\ (2^{55}) = 2^{54.6}$$

  - But in practice, try all in dictionary and quit if not found — work is at most 2^{20} and probability of success is 1/4

Password Cracking

- Attack any of 1024 passwords in file
- **Without** dictionary
  - Assume all 2^{10} passwords are distinct
  - Need 2^{55} comparisons before expect to find password
  - If no salt, each hash computation gives 2^{10} comparisons $\Rightarrow$ the expected work (number of hashes) is $2^{55}/2^{10} = 2^{45}$
  - If salt is used, expected work is 2^{55} since each comparison requires a new hash computation

Password Cracking

- Attack any of 1024 passwords in file
- **With** dictionary
  - Probability at least one password is in dictionary is $1 - (3/4)^{1024} = 1$
  - We ignore case where no pwd is in dictionary
  - If no salt, work is about $2^{19}/2^{10} = 2^9$
  - If salt, expected work is less than 2^{22}
  - Note: If no salt, we can precompute all dictionary hashes and amortize the work

Other Password Issues

- Too many passwords to remember
  - Results in password reuse
  - Why is this a problem?
- Who suffers from bad password?
  - Login password vs ATM PIN
- Failure to change default passwords
- Social engineering
- Error logs may contain "almost" passwords
- Bugs, keystroke logging, spyware, etc.

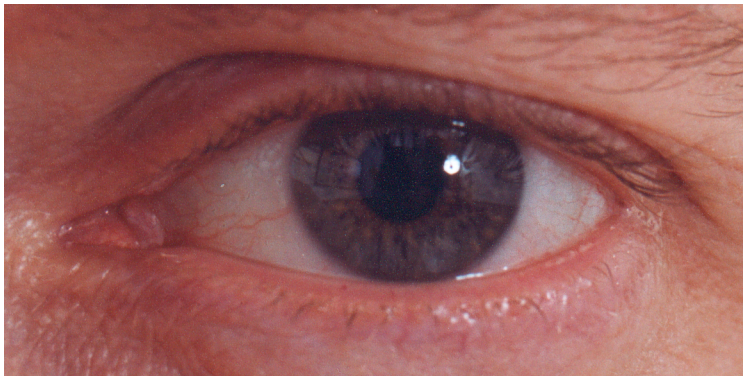
Passwords

- The bottom line
- **Password cracking is too easy!**
  - One weak password may break security
  - Users choose bad passwords
  - Social engineering attacks, etc.
- The bad guy has all of the advantages
- All of the math favors bad guys
- Passwords are a **big** security problem

Password Cracking Tools

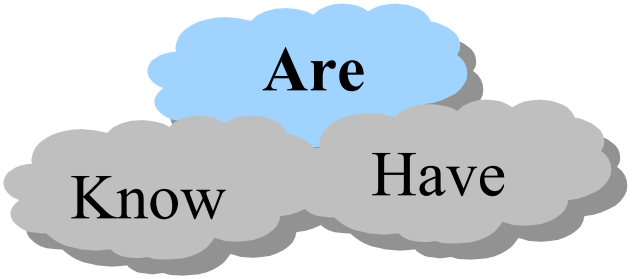
- Popular password cracking tools
  - Password Crackers
  - Password Portal
  - L0phtCrack and LC4 (Windows)
  - John the Ripper (Unix)
- Admins should use these tools to test for weak passwords since attackers will!
- Good article on password cracking
  - Passwords - Conerstone of Computer Security

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?

- Biometrics seen as desirable replacement for passwords
- Cheap and reliable biometrics needed
- Today, a very 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, want it to remain valid for a long time
- **Collectable** — easy to collect required data
  - Depends on whether subjects are cooperative
- Safe, easy to use, etc., etc.

Biometric Modes

- **Identification** — Who goes there?
  - Compare one to many
  - Example: The FBI fingerprint database
- **Authentication** — Is that really you?
  - Compare one to one
  - Example: Thumbprint mouse
- Identification problem 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 for good recognition
  - A weak point of many biometric schemes
- Recognition phase
  - Biometric detection when used in practice
  - Must be quick and simple
  - But must be reasonably accurate

Cooperative Subjects

- We are assuming cooperative subjects
- In identification problem often have uncooperative subjects
- For example, facial recognition
  - Proposed for use in Las Vegas casinos to detect known cheaters
  - Also as way to detect terrorists in airports, etc.
  - Probably do not have ideal enrollment conditions
  - Subject will try to confuse recognition phase
- Cooperative subject makes it much easier!
  - In authentication, subjects are cooperative

Biometric Errors

- **Fraud rate** versus **insult rate**
  - Fraud — user A mis-authenticated as user B
  - Insult — user A not authenticate as user A
- For any biometric, can decrease fraud or insult, but other 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
  - The best measure for comparing biometrics

Fingerprint History

- 1823 — Professor Johannes Evangelist Purkinje discussed 9 fingerprint patterns
- 1856 — Sir William Hershel used fingerprint (in India) on contracts
- 1880 — Dr. Henry Faulds article in *Nature* about fingerprints for ID
- 1883 — Mark Twain's *Life on the Mississippi* a murderer ID'ed by fingerprint

Fingerprint History

- 1888 — Sir Francis Galton (cousin of Darwin) developed classification system
  - His system of "minutia" is still in use today
  - Also verified that fingerprints do not change
- Some countries require a number of points (i.e., minutia) to match in criminal cases
  - In Britain, 15 points
  - In US, no fixed number of points required

Fingerprint Comparison

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

Loop (double)

Whorl

Arch

Fingerprint Biometric

- Capture image of fingerprint
- Enhance image
- Identify minutia

Fingerprint Biometric

- Extracted minutia are compared with user's minutia stored in a database
- Is it a statistical match?

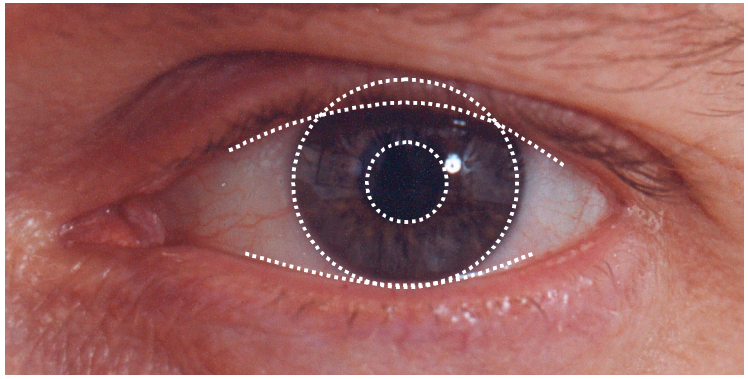
Hand Geometry

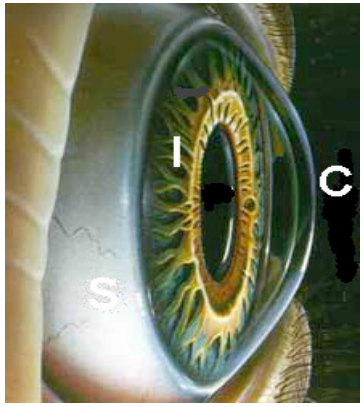
- Popular form of biometric
- Measures shape of hand
  - Width of hand, fingers
  - Length of fingers, etc.
- Human hands not unique
- Hand geometry sufficient for many situations
- Suitable for authentication
- Not useful for ID problem

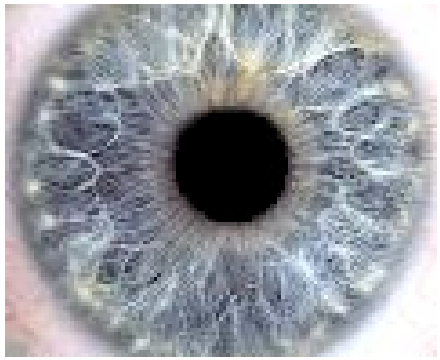
Hand Geometry

- Advantages
  - Quick
  - 1 minute for enrollment
  - 5 seconds for recognition
  - Hands symmetric (use other hand backwards)
- Disadvantages
  - Cannot use on very young or very old
  - Relatively high equal error rate

Iris Patterns







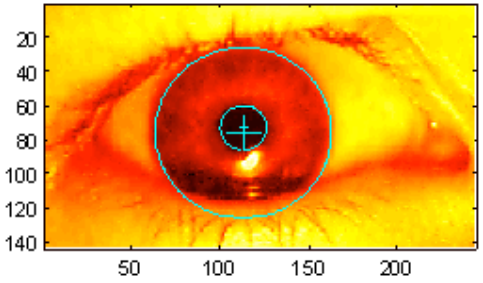
- Iris pattern development is "chaotic"
- Little or no genetic influence
- Different even for identical twins
- Pattern is stable through lifetime

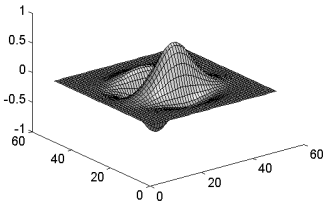
Iris Recognition: History

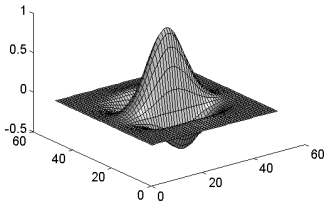
- 1936 — suggested by Frank Burch
- 1980s — James Bond films
- 1986 — first patent appeared
- 1994 — John Daugman patented best current approach
  - Patent owned by Iridian Technologies

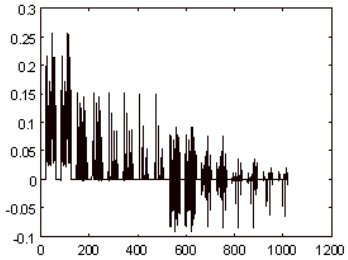
Iris Scan

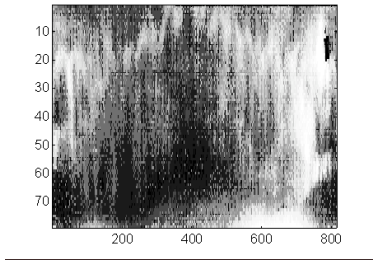
- Scanner locates iris
- Take b/w photo
- Use polar coordinates…
- Find 2-D wavelet trans
- Get 256 byte iris code











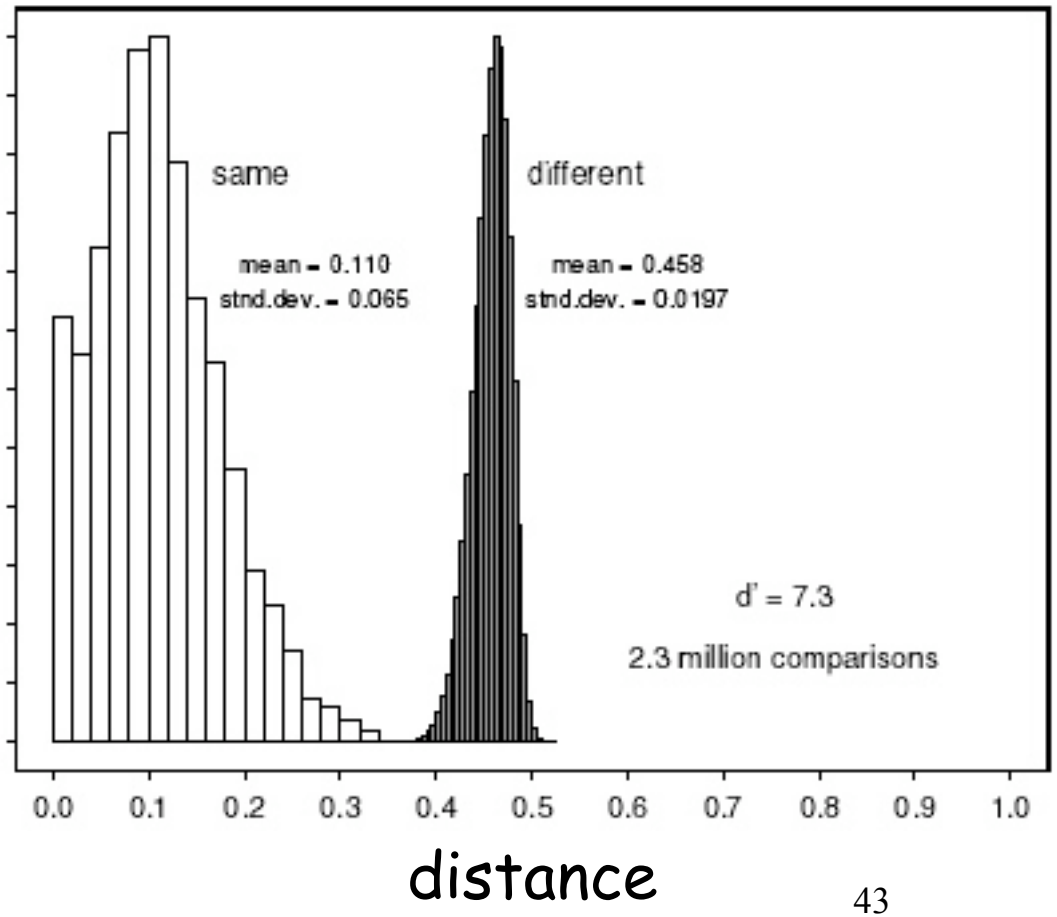
Measuring Iris Similarity

- Based on Hamming distance
- Define d(x,y) to be
  - # of non match bits/# of bits compared
  - d(0010,0101) = 3/4 and d(101111,101001) = 1/3
- Compute d(x,y) on 2048-bit iris code
  - Perfect match is d(x,y) = 0
  - For same iris, expected distance is 0.08
  - At random, expect distance of 0.50
  - Accept as match if distance less than 0.32

Iris Scan Error Rate

distance	Fraud rate
0.29	1 in $1.3*10^{10}$
0.30	1 in $1.5*10^{9}$
0.31	1 in $1.8*10^{8}$
0.32	1 in $2.6*10^{7}$
0.33	1 in $4.0*10^{6}$
0.34	1 in $6.9*10^{5}$
0.35	1 in $1.3*10^{5}$

: equal error rate



Attack on Iris Scan

- Good **photo** of eye can be scanned
- And attacker can use photo of eye
- Afghan woman was authenticated by iris scan of old photo
  - Story is here
- To prevent photo 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, hard to achieve
  - Enrollment phase must be extremely accurate
- Most biometrics much worse than fingerprint!
- Biometrics useful for authentication…
- But ID biometrics are almost useless today

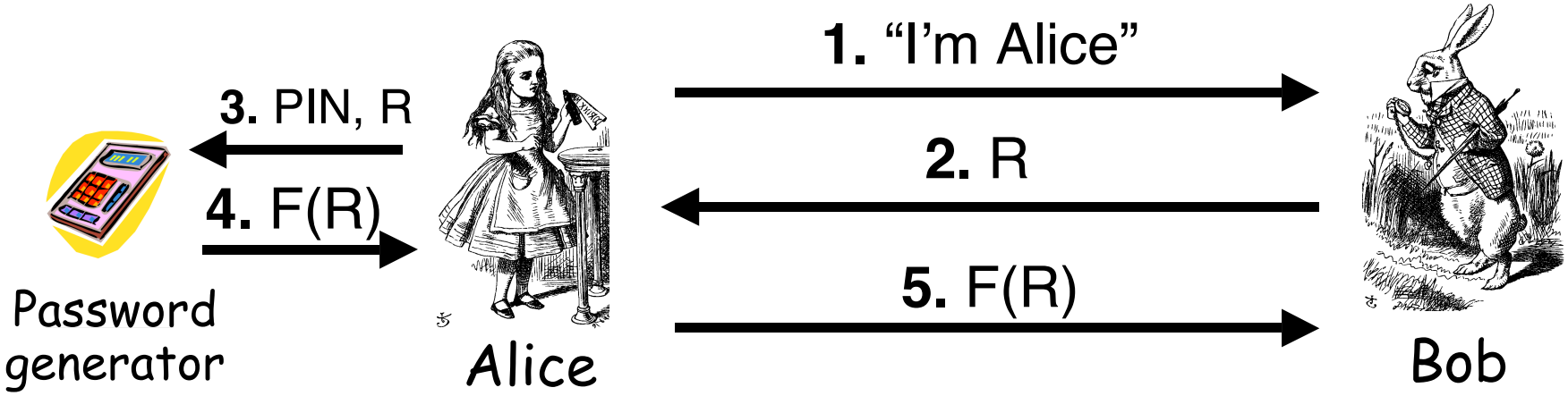
Biometrics: The Bottom Line

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

Something You Have

- Something in your possession
- Examples include
  - Car key
  - Laptop computer
    - Or specific MAC address
  - Password generator
    - We'll look at this next
  - ATM card, smartcard, etc.

Password Generator



- Alice gets "challenge" R from Bob
- Alice enters R into password generator
- Alice sends "response" back to Bob
- Alice **has** pwd generator and **knows** PIN

2-factor Authentication

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

Single Sign-on

- A hassle to enter password(s) repeatedly
  - Users want to authenticate only once
  - "Credentials" stay with user wherever he goes
  - Subsequent authentication is transparent to user
- Single sign-on for the Internet?
  - Microsoft: **Passport**
  - Everybody else: **Liberty Alliance**
  - Security Assertion Markup Language (**SAML**)

Web Cookies

- Cookie is provided by a Website and stored on user's machine
- Cookie indexes a database at Website
- Cookies **maintain state** across sessions
- Web uses a stateless protocol: HTTP
- Cookies also maintain state within a session
- Like a single sign-on for a website
  - Though a very weak form of authentication
- Cookies and privacy concerns

Authorization

Authentication vs Authorization

- Authentication — Who goes there?
  - 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**
- Authorization enforced by
  - Access Control Lists
  - Capabilities

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
- But how to manage a large access control (AC) matrix?
- Could be 1000's of users, 1000's of resources
- Then AC matrix with 1,000,000's of entries
- Need to check this matrix before access to any resource is allowed
- Hopelessly inefficient

Access Control Lists (ACLs)

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

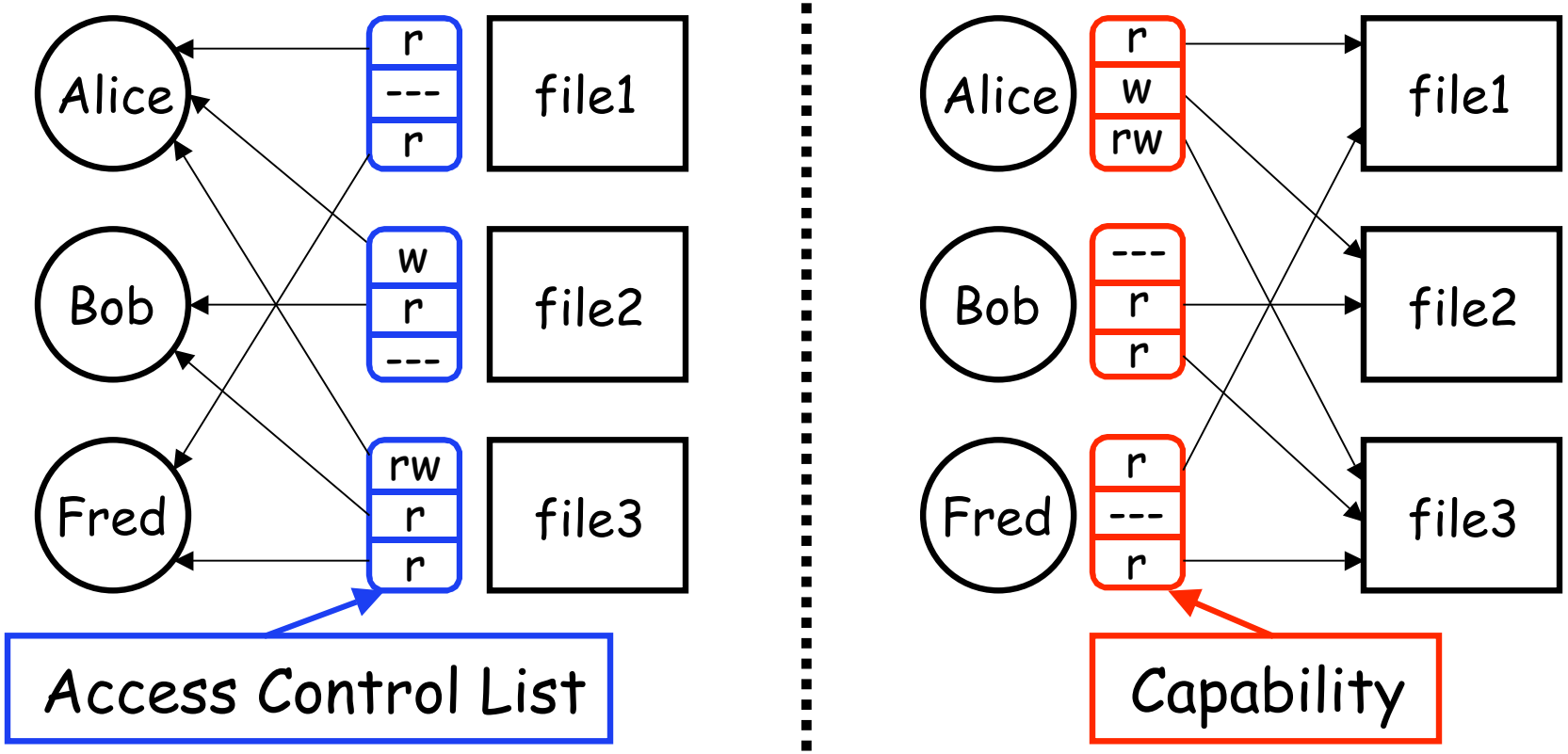
	OS	Accounting program	Accounting data	Insurance data	Payroll data
Bob	rx	rx	r	---	---
Alice	rx	rx	r	rw	rw
Sam	rwx	rwx	r	rw	rw
Accounting program	rx	rx	rw	rw	rw

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

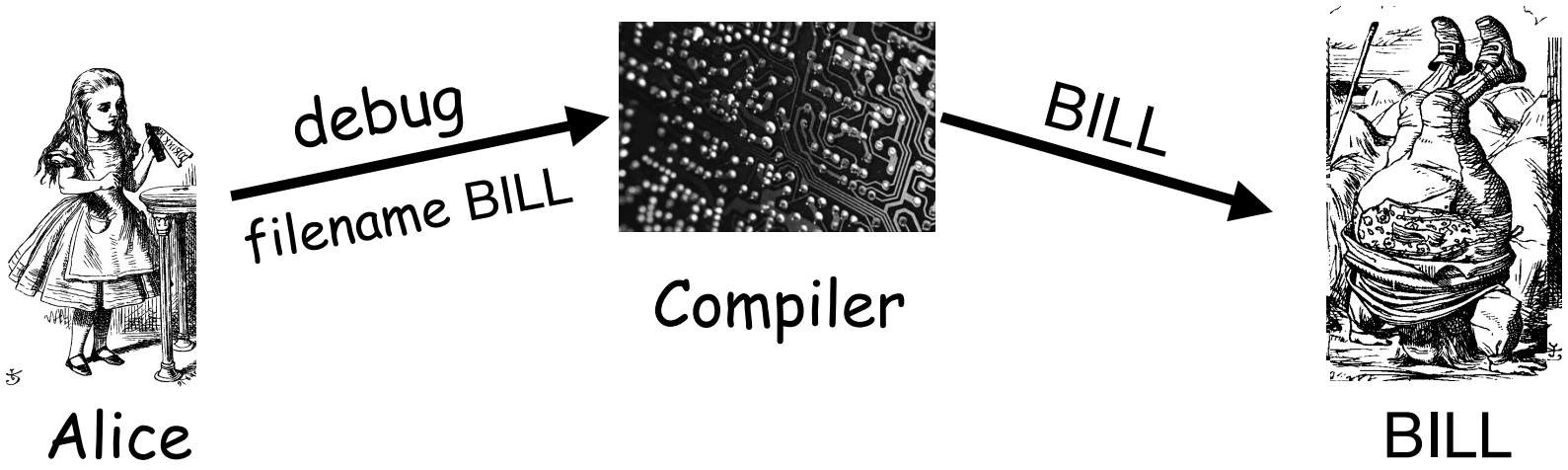
Confused Deputy

- 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	---
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 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
  - Easy to add/delete users
  - Easier to avoid the confused deputy
  - More difficult to implement
  - The "Zen of information security"
- Capabilities loved by academics
  - Capability Myths Demolished

Multilevel Security (MLS) Models

Classifications and Clearances

- **Classifications** apply to **objects**
- **Clearances** apply to **subjects**
- US Department of Defense uses 4 levels of classifications/clearances
  - **TOP SECRET**
  - **SECRET**
  - **CONFIDENTIAL**
  - **UNCLASSIFIED**

Clearances and Classification

- To obtain a **SECRET** clearance requires a routine background check
- A **TOP SECRET** clearance requires extensive background check
- Practical classification problems
  - Proper classification not always clear
  - Level of granularity to apply classifications
  - Aggregation — flipside of granularity

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

  TOP SECRET > SECRET > CONFIDENTIAL > UNCLASSIFIED

Multilevel Security (MLS)

- MLS needed when subjects/objects at different levels use same system
- MLS is a form of **Access Control**
- Military/government interest in MLS for many decades
  - Lots of funded research into MLS
  - Strengths and weaknesses of MLS relatively well understood (theoretical and practical)
  - Many possible uses of MLS outside military

MLS Applications

- Classified government/military information
- **Business example:** info restricted to
  - Senior management only
  - All management
  - Everyone in company
  - General public
- Network firewall
  - Keep intruders at low level to limit 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
  - 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) ≤ L(S)

  ***-Property** (**Star Property**): S can write O if and only if L(S) ≤ L(O)

- **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 property: security labels never change
  - Weak tranquility property: 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 needed for current work
  - Then upgrade privilege 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

BLP: The Bottom Line

- BLP is simple, but probably too simple
- BLP is one of the few security models that can be used to prove things about systems
- BLP has inspired other security models
  - Most other models try to be more realistic
  - Other security models are more complex
  - Other models difficult to analyze and/or apply in practice

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 $I(O) \leq I(S)$

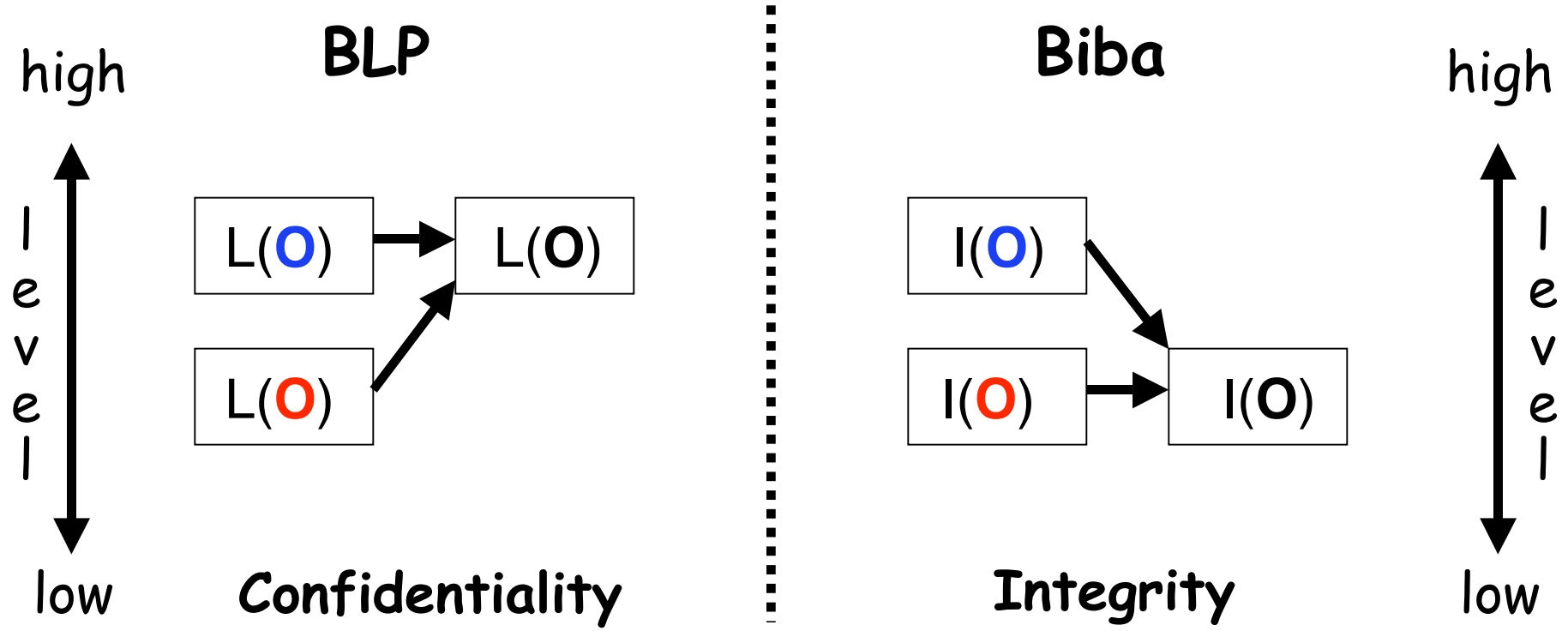
  (if S writes O, the integrity of O ≤ that of S)

  Biba's Model: S can read O if and only if $I(S) \leq I(O)$

  (if S reads O, the integrity of S ≤ 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


BLP
high
level
low
L(O)
L(O)
L(O)
Confidentiality
Biba
high
level
low
I(O)
I(O)
I(O)
Integrity

Multilateral Security (Compartments)

Multilateral Security

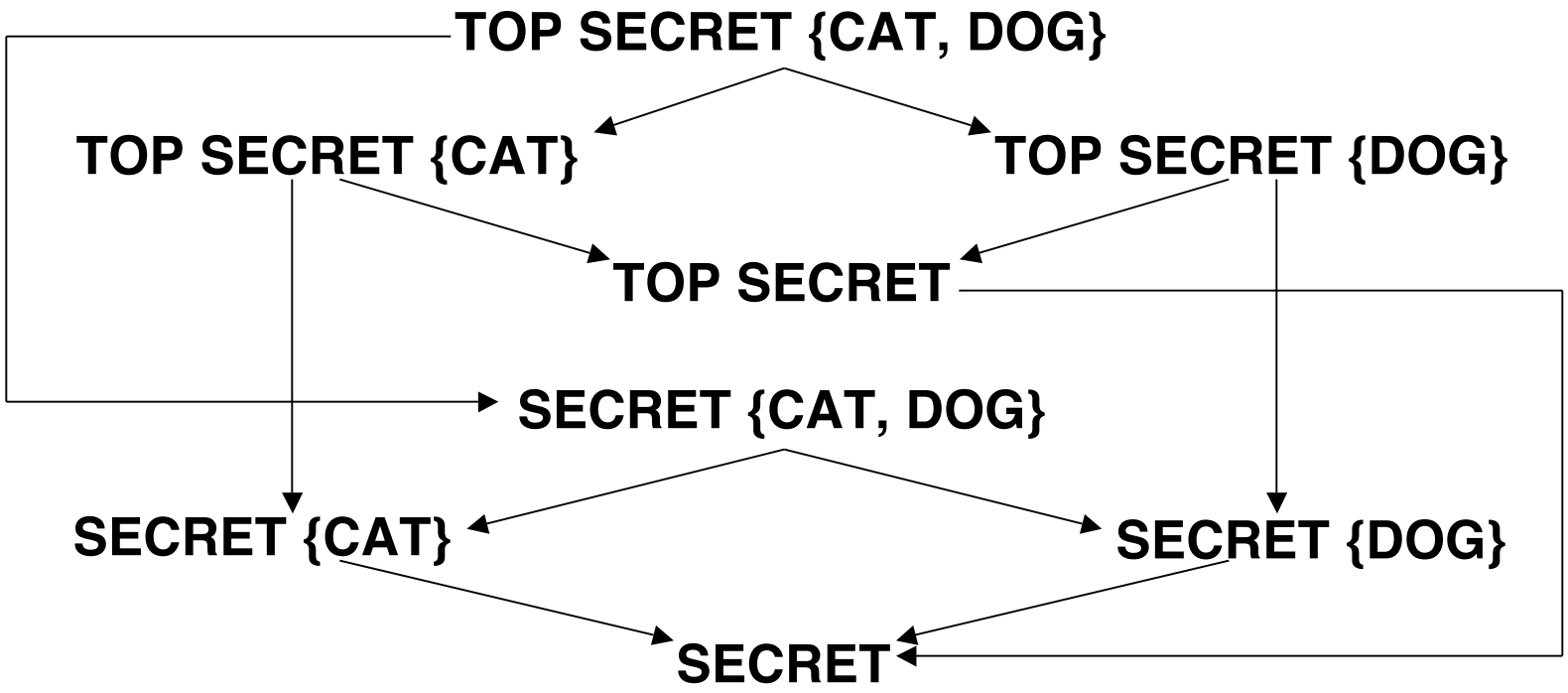
- Multilevel Security (MLS) enforces access control **up and down**
- Simple hierarchy of security labels may not be flexible enough
- Multilateral security enforces access control **across** by creating compartments
- Suppose **TOP SECRET** divided into **TOP SECRET {CAT}** and **TOP SECRET {DOG}**
- Both are **TOP SECRET** but information flow restricted across the **TOP SECRET** level

Multilateral Security

- Why compartments?
  - Why not create a new classification level?
- May not want either of
  - **TOP SECRET {CAT} ≥ TOP SECRET {DOG}**
  - **TOP SECRET {DOG} ≥ TOP SECRET {CAT}**
- Compartments allow us to enforce the **need to know** principle
  - Regardless of your clearance, you only have access to info that you need to know

Multilateral Security

- Arrows indicate "≥" relationship



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

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

MLS vs Multilateral Security

- MLS can be used without multilateral security or vice-versa
- But, MLS almost always includes multilateral
- Example
  - MLS mandated for protecting medical records of British Medical Association (BMA)
  - AIDS was **TOP SECRET**, prescriptions **SECRET**
  - What is the classification of an AIDS drug?
  - Everything tends toward **TOP SECRET**
  - Defeats the purpose of the system!
- Multilateral security was used instead

Covert Channel

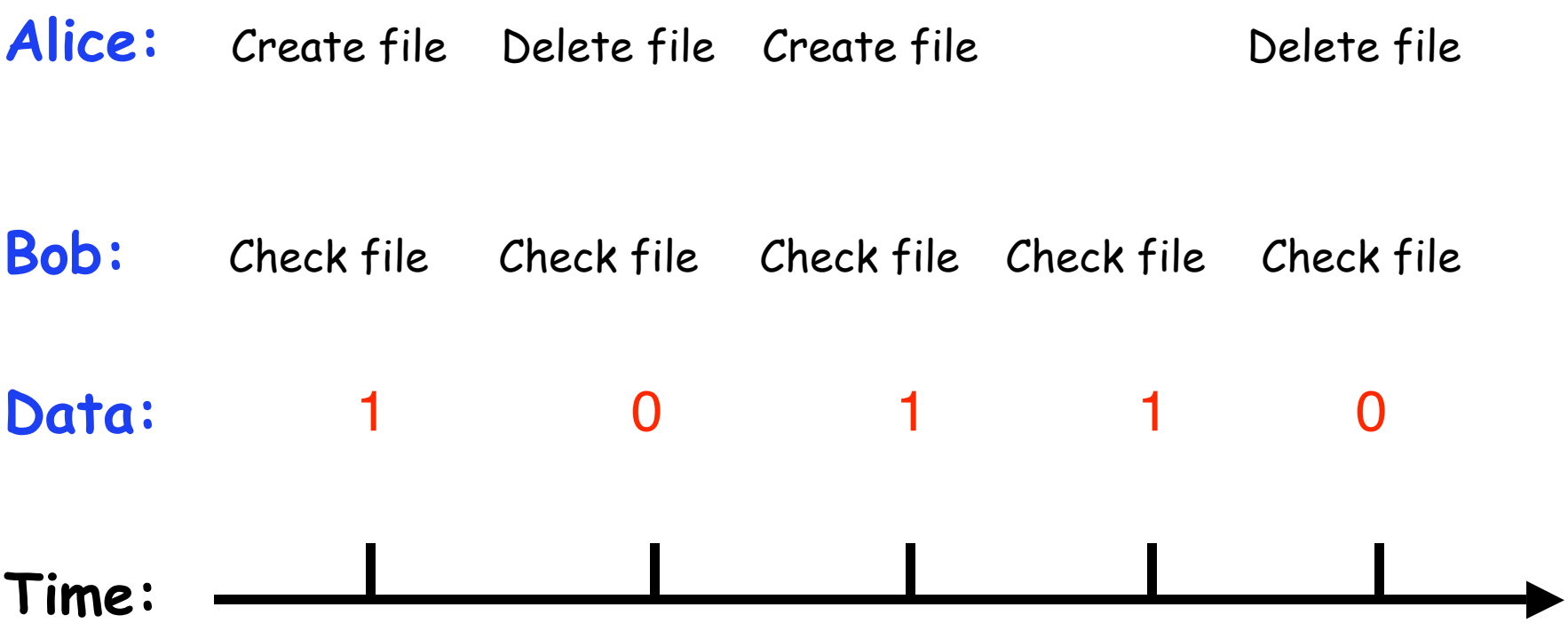
Covert Channel

- MLS designed to restrict legitimate channels of communication
- May be other ways for information to flow
- For example, resources shared at different levels may signal information
- **Covert channel**: "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 each 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



Covert Channel

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

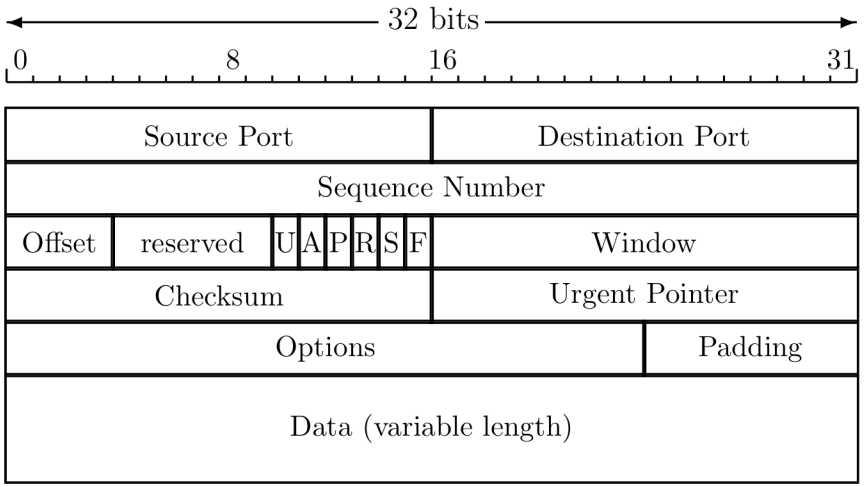
Covert Channel

- Covert channels exist almost everywhere
- Easy to eliminate covert channels…
  - Provided you eliminate all shared resources and all communication
- Virtually impossible to eliminate all covert channels in any useful system
  - DoD guidelines: goal is to **reduce covert channel capacity** to no more than 1 bit/second
  - Implication is that DoD has given up trying to eliminate covert channels!

Covert Channel

- Consider 100MB **TOP SECRET** file
  - Plaintext version stored in **TOP SECRET** place
  - Encrypted with AES using 256-bit key, ciphertext 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!

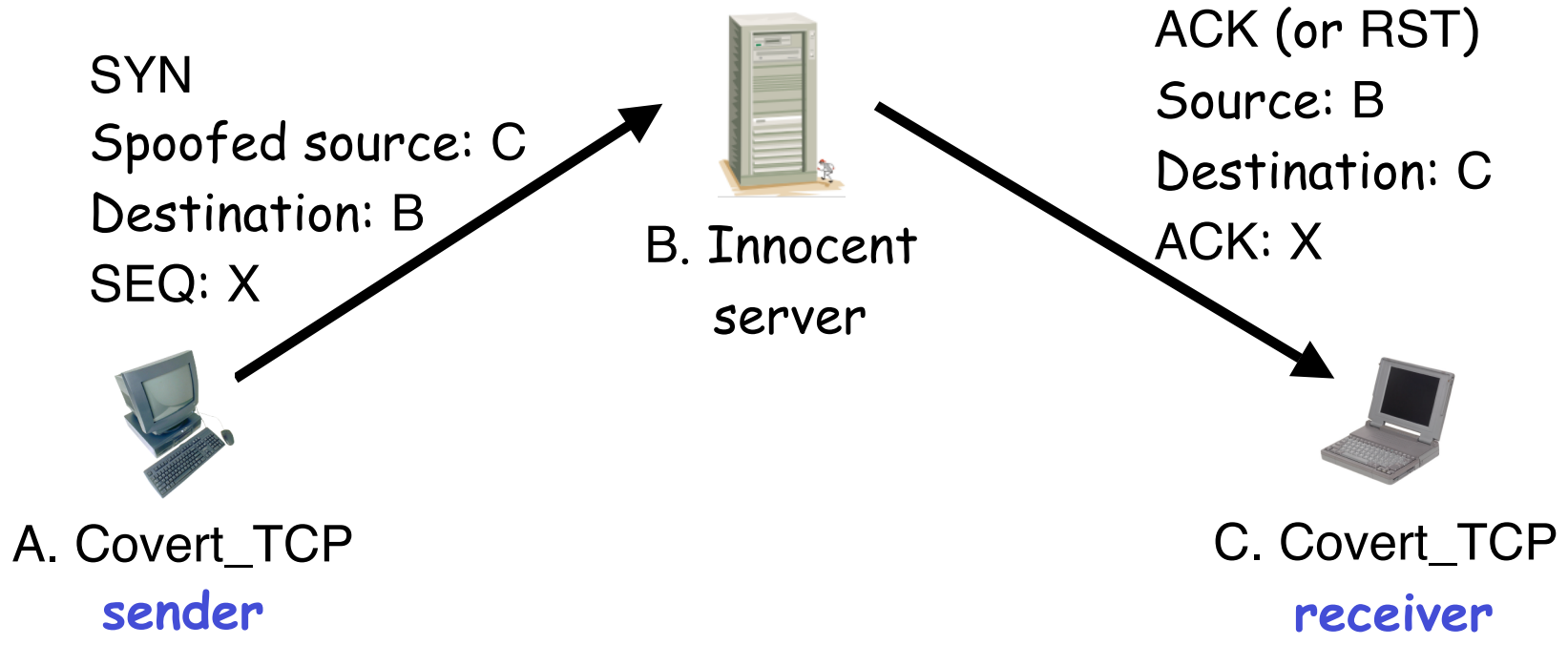
Real-World Covert Channel



- Hide data in TCP header "reserved" field
- Or use covert_TCP, tool to hide data in
  - Sequence number
  - ACK number

Real-World Covert Channel

- Hide data in TCP sequence numbers
- Tool: covert_TCP
- Sequence number X contains covert info



Inference Control

Inference Control Example

- Suppose we query a database
  - Question: What is average salary of female CS professors at SJSU?
  - Answer: $95,000
  - Question: How many female 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
- 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
  - Used by the US Census Bureau
- Randomization
  - Add small amount of random noise to data
- Many other methods — none satisfactory

Inference Control: The Bottom Line

- Robust inference control may be impossible
- Is weak inference control better than no inference control?
  - **Yes**: Reduces amount of information that leaks and thereby limits the damage
- 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 one other human and one computer (without seeing either)
- If human questioner cannot distinguish the human from the computer responder, the computer passes the test
- The gold standard in artificial intelligence
- No computer can pass this today

CAPTCHA

- **CAPTCHA** — **C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part
- **A**utomated — test is generated and scored by a computer program
- **P**ublic — program and data are public
- **T**uring test to tell… — humans can pass the test, but machines cannot pass the test
- Like an inverse Turing test (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 to restrict access to resources to humans (no computers)
- CAPTCHA useful for **access control**

CAPTCHA Uses?

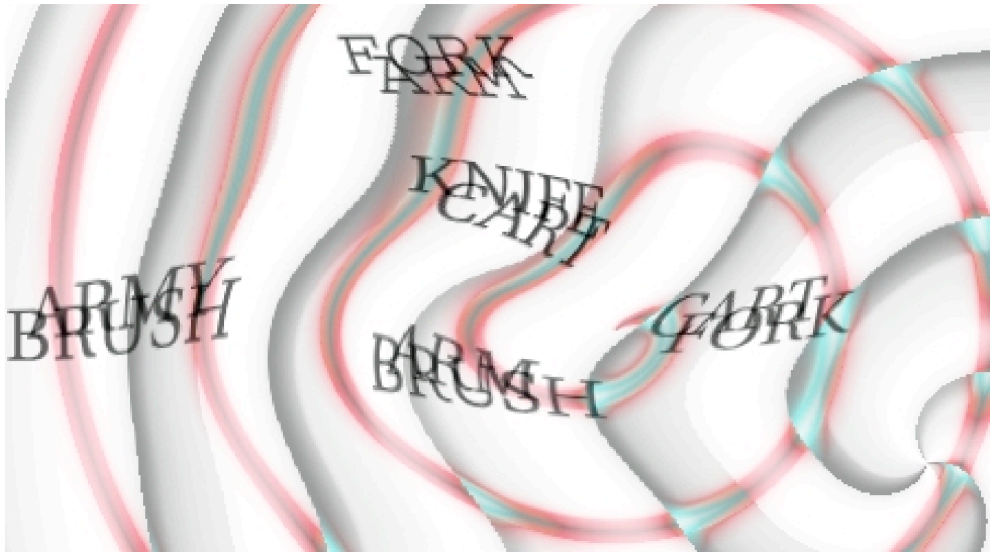
- Original motivation: automated "bots" stuffed ballot box in vote for best CS school
- Free email services — spammers used bots sign up for 1000's of email accounts
  - CAPTCHA employed so only humans can get accts
- Sites that do not want to be automatically indexed by search engines
  - HTML tag only says "please do not index me"
  - CAPTCHA would force human intervention

CAPTCHA: Rules of the Game

- Must be easy for most humans to pass
- Must be difficult or impossible for machines to pass
  - **Even with access to CAPTCHA software**
- The only unknown is some random number
- 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
- Difficult for computers (OCR problem)

CAPTCHAs

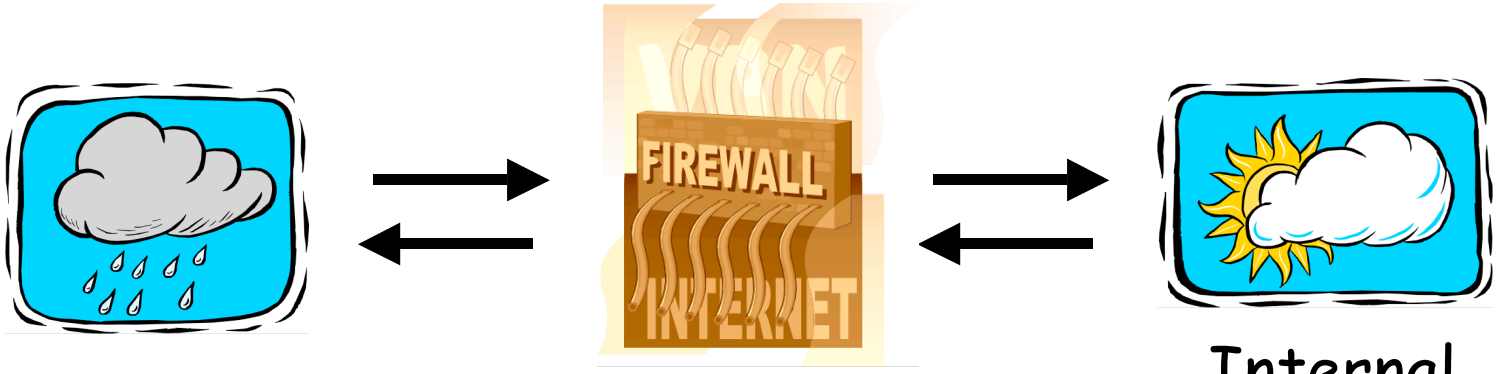
- Current types of CAPTCHAs
  - Visual
    - Like previous example
    - Many others
  - Audio
    - Distorted words or music
- No text-based CAPTCHAs
  - Maybe this is not possible…

CAPTCHA's and AI

- Computer recognition of distorted text is a challenging AI problem
  - But humans can solve this problem
- Same is true of distorted sound
  - Humans also good at solving this
- Hackers who break such a CAPTCHA have solved a hard AI problem
- Putting hacker's effort to good use!

Firewalls

Firewalls



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

Firewall as Secretary

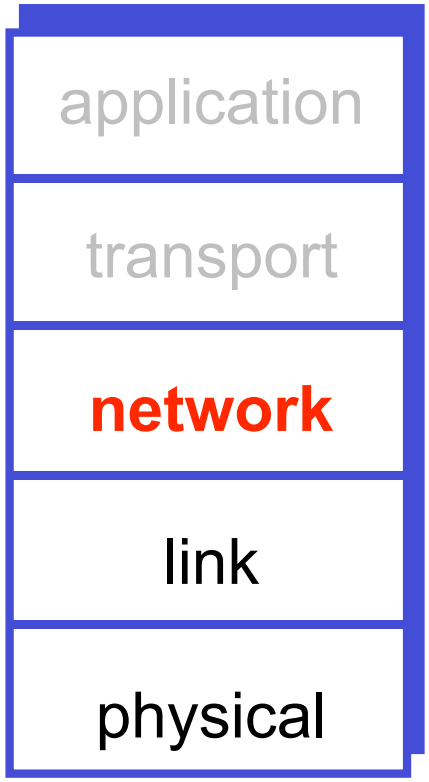
- A firewall is like a **secretary**
- To meet with an executive
  - First contact the secretary
  - Secretary decides if meeting is reasonable
  - Secretary filters out many requests
- You want to meet chair of CS department?
  - Secretary does some filtering
- You want to meet President of US?
  -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

Firewall Terminology

- No standard terminology
- Types of firewalls
  - **Packet filter** — works at network layer
  - **Stateful packet filter** — transport layer
  - **Application proxy** — application layer
  - Personal firewall — for single user, home network, etc.

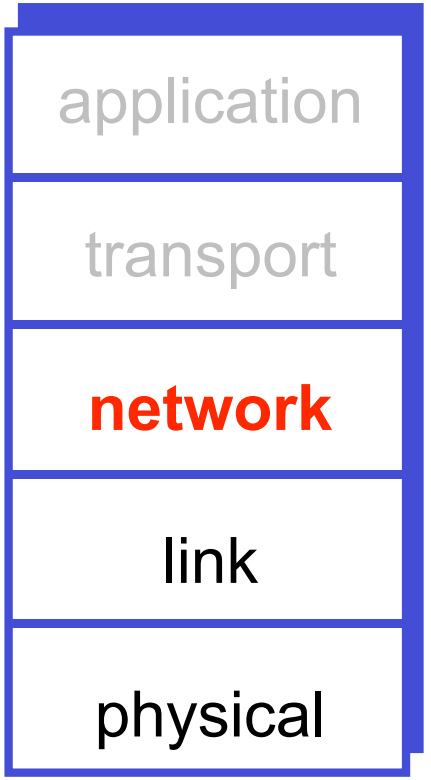
Packet Filter

- 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



Packet Filter

- Advantage
  - Speed
- Disadvantages
  - No state
  - Cannot see TCP connections
  - Blind to application data



Packet Filter

- Configured via Access Control Lists (ACLs)
  - Different meaning of ACL than previously

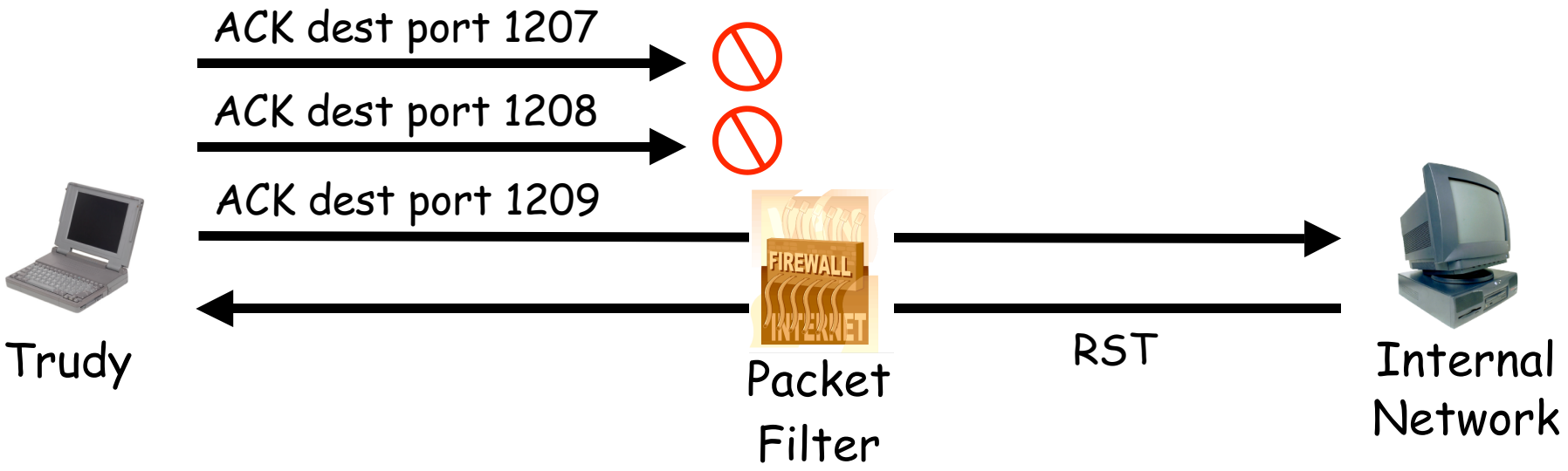
Action	Source IP	Dest IP	Source Port	Dest Port	Protocol	Flag Bits
Allow	Inside	Outside	Any	80	HTTP	Any
Allow	Outside	Inside	80	> 1023	HTTP	ACK
Deny	All	All	All	All	All	All

- Intention is to restrict incoming packets to Web responses

TCP ACK Scan

- Attacker sends packet with ACK bit set, **without** prior 3-way handshake
- Violates TCP/IP protocol
- ACK packet pass thru packet filter firewall
  - Appears to be part of an ongoing connection
- RST sent by recipient of such packet
- Attacker scans for open ports thru firewall

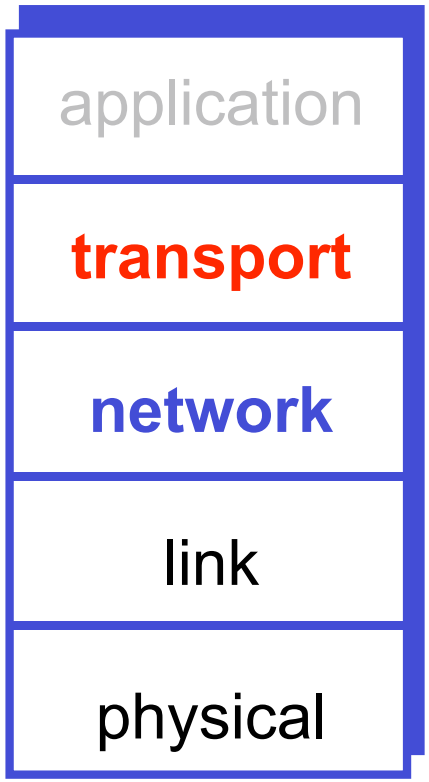
TCP ACK Scan



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

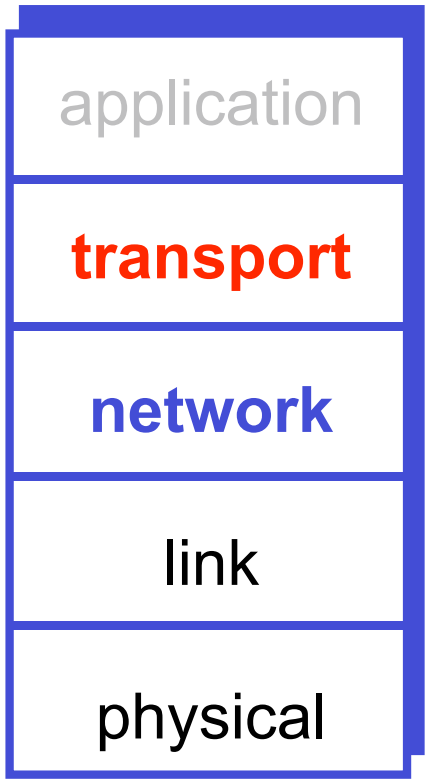
Stateful Packet Filter

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



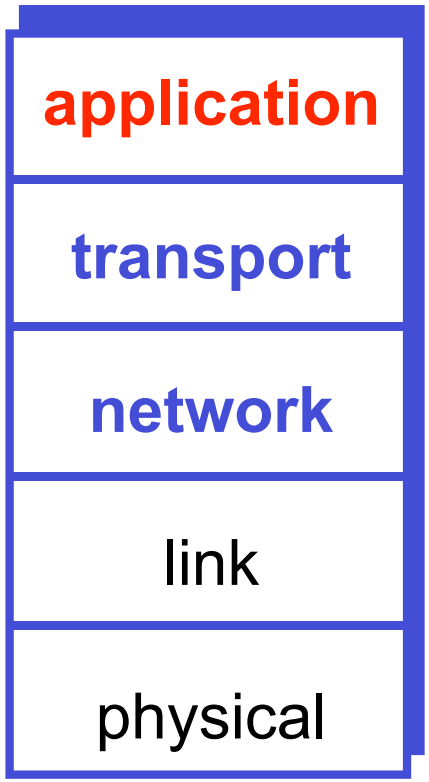
Stateful Packet Filter

- Advantages
  - Can do everything a packet filter can do plus...
  - Keep track of ongoing connections
- Disadvantages
  - Cannot see application data
  - Slower than packet filtering



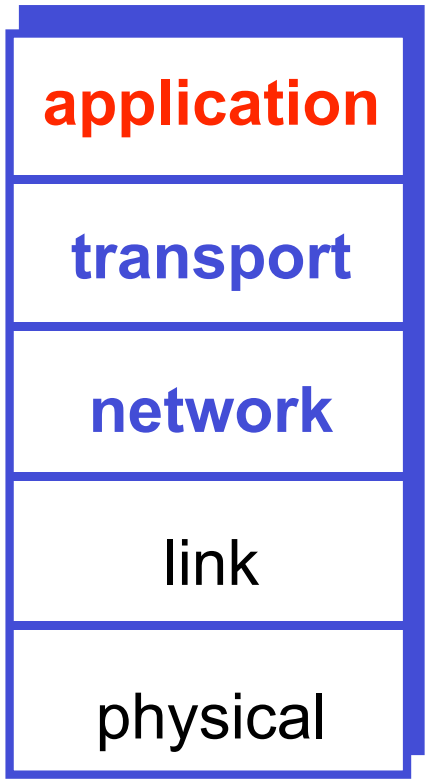
Application Proxy

- 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, Word macros)
- Disadvantage
  - Speed



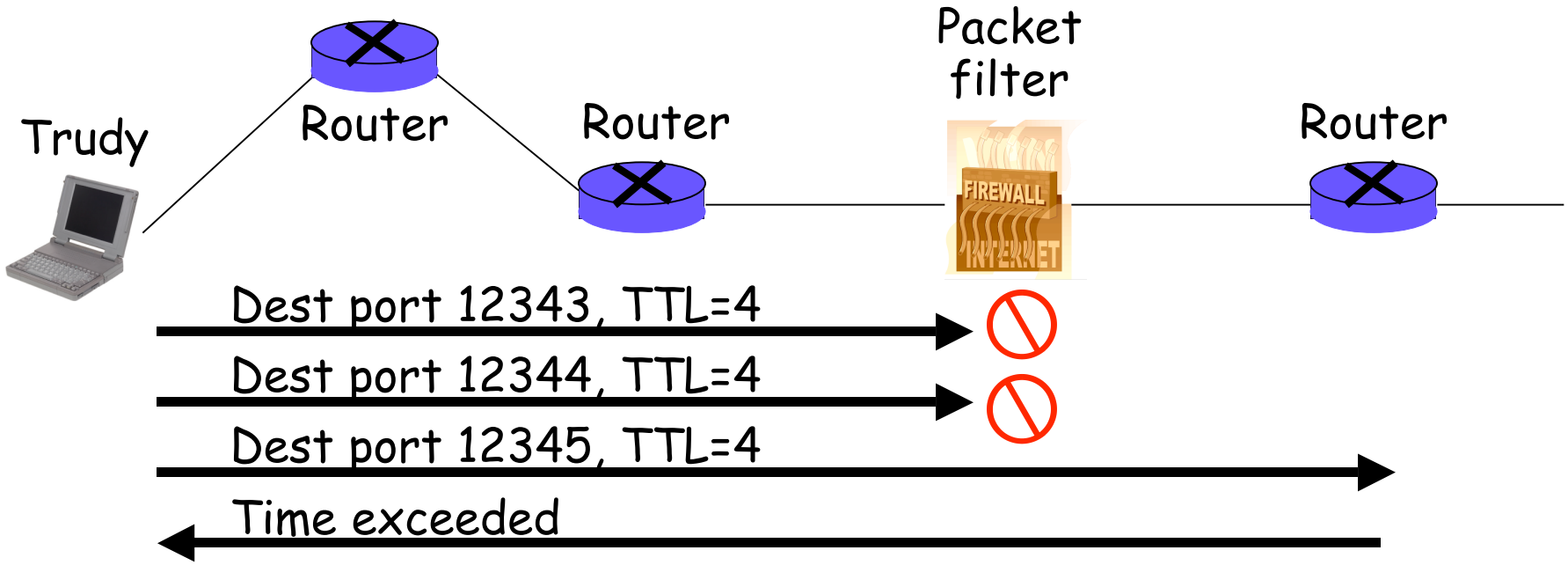
Application Proxy

- 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 attacks stateful packet filter cannot — see next slides

Firewalk

- Tool to scan for open ports thru firewall
- Known: IP address of firewall and IP address of one system inside firewall
  - TTL set to 1 more than number of hops to firewall and set destination port to N
  - If firewall does not let thru data on port N, no response
  - If firewall allows data on port N thru firewall, get time exceeded error message

Firewalk and Proxy Firewall



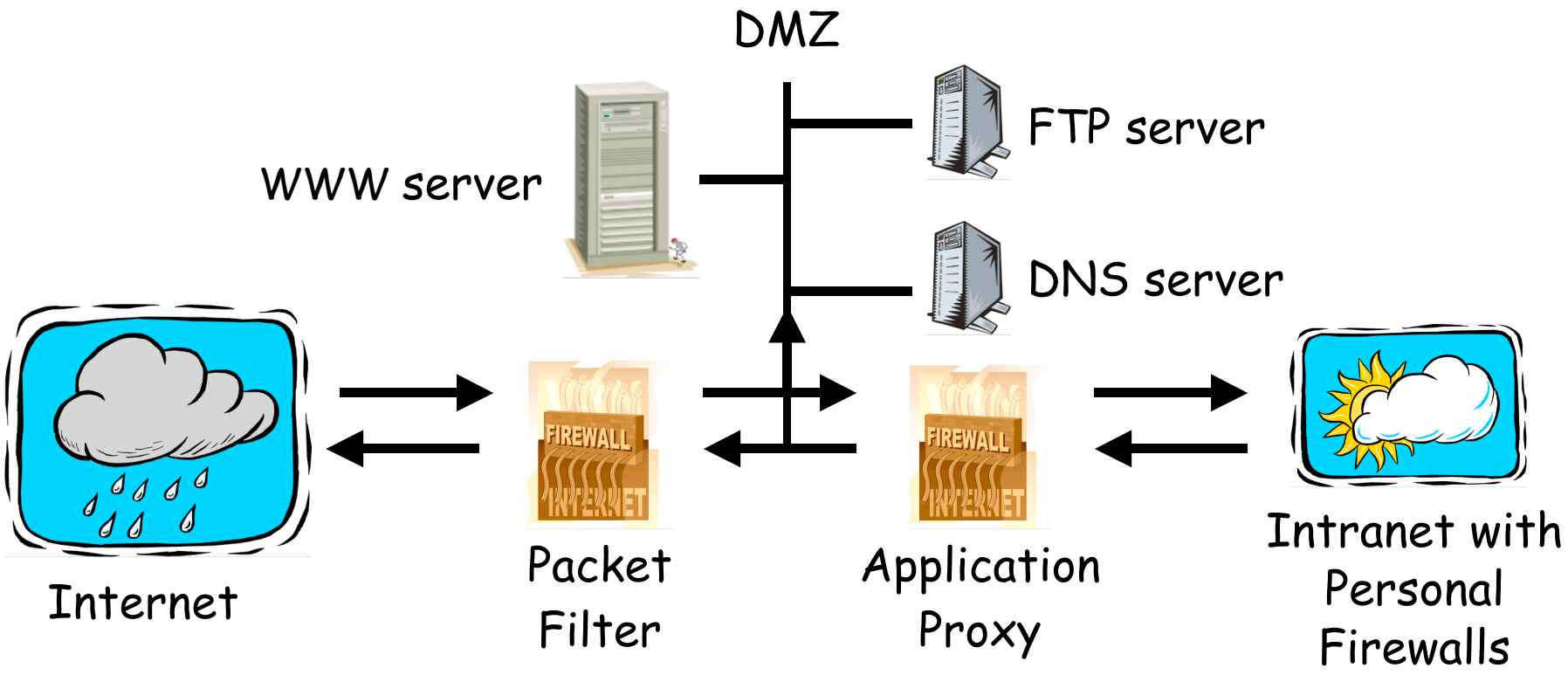
- This will **not** work thru an application proxy
- The proxy creates a new packet, destroys old TTL

Personal Firewall

- To protect one user or home network
- Can use any of the methods
  - Packet filter
  - Stateful packet filter
  - Application proxy

Firewalls and Defense in Depth

- Example security architecture



Intrusion Detection Systems

Intrusion Prevention

- Want to keep bad guys out
- **Intrusion prevention** is a traditional focus of computer security
  - Authentication is to prevent intrusions
  - Firewalls a form of intrusion prevention
  - Virus defenses also intrusion prevention
- Comparable to locking the door on your car

Intrusion Detection

- In spite of intrusion prevention, bad guys will sometime get into system
- Intrusion detection systems (IDS)
  - Detect attacks
  - Look for "unusual" activity
- IDS developed out of log file analysis
- IDS is currently a very **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 or little-known attacks
  - Use a system to attack other systems
  - Etc.

IDS

- Intrusion detection **approaches**
  - Signature-based IDS
  - Anomaly-based IDS
- Intrusion detection **architectures**
  - Host-based IDS
  - Network-based IDS
- Most systems can be classified as above
  - In spite of marketing claims to the contrary!

Host-based IDS

- Monitor activities on hosts for
  - Known attacks or
  - Suspicious behavior
- Designed to detect attacks such as
  - Buffer overflow
  - Escalation of privilege
- Little or no view of network activities

Network-based IDS

- Monitor activity on the network for
  - Known attacks
  - Suspicious network activity
- Designed to detect attacks such as
  - Denial of service
  - Network probes
  - Malformed packets, etc.
- Can be some overlap with firewall
- Little or no view of host-base attacks
- Can have both host and network IDS

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 the warning is specific
  - Admin knows what attack is suspected
  - Admin can verify attack (or false alarm)

Signature Detection

- Suppose IDS warns whenever N or more failed logins in M seconds
- Must set N and M so that false alarms not common
- Can do this based on normal behavior
- But if attacker knows the signature, he can try N-1 logins every M seconds!
- In this case, signature detection slows the attacker, but might not stop him

Signature Detection

- Many techniques used to make signature detection more robust
- Goal is usually 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, other
  - Must take care not to increase false alarm rate

Signature Detection

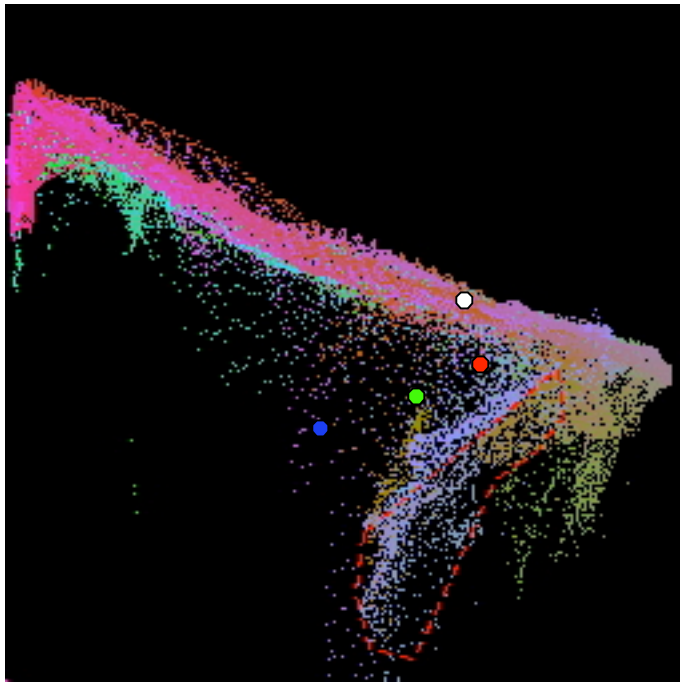
- Advantages of signature detection
  - Simple
  - Detect known attacks
  - Know which attack at time of detection
  - Efficient (if reasonable number of signatures)
- Disadvantages of signature detection
  - Signature files must be kept up to date
  - Number of signatures may become large
  - Can only 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?
- Statistics is obviously required here!
  - The **mean** defines normal
  - The **variance** indicates how far abnormal lives from normal

What is Normal?

- Consider the scatterplot below



- White dot is "normal"
- Is red dot normal?
- Is green dot normal?
- How abnormal is the blue dot?
- Stats can be tricky!

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 chance of success

How to Measure Abnormal?

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

Anomaly Detection (1)

- Spse we monitor use of three commands:

  open, read, close

- Under normal use we observe that Alice

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

- Of the six possible ordered pairs, 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 using 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 file F_n at rate A_n

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

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

Anomaly Detection (2)

- To allow "normal" to adapt to new use, we update long-term averages as

  $H_n = 0.2A_n + 0.8H_n$

- Then H_0 and H_1 are unchanged, H_2=.2*.3+.8*.4=.38 and H_3=.2*.2+.8*.1=.12
- And the long term averages are updated as

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

- New observed rates are…

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 by $H_n = 0.2A_n + 0.8H_n$

Anomaly Detection (2)

- The starting averages were

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

- After 2 iterations, the averages are

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

- The stats slowly evolve to match behavior
- This reduces false alarms and work for admin
- But also opens an avenue for attack…
- Suppose Trudy **always** wants to access F_3
- She can convince IDS this is normal for Alice!

Anomaly Detection (2)

- To make this approach more robust, must also incorporate the variance
- Can also combine N stats as, for example,

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

  to obtain a more complete view of "normal"
- Similar (but more sophisticated) approach is used in IDS known as NIDES
- NIDES includes anomaly and signature IDS

Anomaly Detection Issues

- System constantly evolves 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?
  - Only that there is possibly an attack
  - May not say anything specific about attack!
  - How to respond to such vague information?
- Signature detection tells exactly which attack

Anomaly Detection

- Advantages
  - Chance of detecting unknown attacks
  - May be more efficient (since no signatures)
- Disadvantages
  - Today, cannot be used alone
  - Must be used with a signature detection system
  - Reliability is unclear
  - May be subject to attack
  - Anomaly detection indicates something unusual
  - But lack of specific info on possible attack!

Anomaly Detection: The Bottom Line

- Anomaly-based IDS is active research topic
- Many security professionals have very high hopes for its ultimate success
- Often cited as key future security technology
- Hackers are not convinced!
  - Title of a talk at Defcon 11: "Why Anomaly-based IDS is an Attacker's Best Friend"
- Anomaly detection is difficult and tricky
- Is anomaly detection as hard as AI?

Access Control Summary

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

Access Control Summary

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

Coming Attractions…

- Security protocols
  - Generic authentication protocols
  - SSL
  - IPSec
  - Kerberos
  - GSM
- We'll see lots of crypto applications in the next chapter