Part II: Access Control

Access Control

- Two parts to access control…
- **Authentication:** Are you who you say you are?
  - Determine whether access is allowed or not
  - Authenticate human to machine
  - Or, possibly, 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

Chapter 7: Authentication

Guard: Halt! Who goes there?
Arthur: It is I, Arthur, son of Uther Pendragon,
from the castle of Camelot. King of the Britons,
defeater of the Saxons, sovereign of all England!
 Monty Python and the Holy Grail

Then said they unto him, Say now Shibboleth:
and he said Sibboleth: for he could not frame to pronounce it right.
Then they took him, and slew him at the passages of Jordan:
and there fell at that time of the Ephraimites forty and two thousand.
 Judges 12:6

Are You Who You Say You Are?

- 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 sysadmin 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
  - incorrect
  - Pikachu
  - 102560
  - AustinStamp
- Good Passwords?
  - jfIej,43j-EmmL+y
  - 098643765372 63
  - P0kem0N
  - FSa7Yago
  - OnceuP0nAt1m8
  - 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 we need to verify passwords
- Solution? **Hash** passwords
  - Store $y = h(\text{password})$
  - Can verify entered password by hashing
  - If Trudy obtains the password file, she does not (directly) 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 of computing hashes in dictionary, actual attack is trivial
- Can we prevent this forward search attack? Or at least make it 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 that the salt s is not secret
  - Analogous to IV
- Still easy to verify salted password
- But lots more work for Trudy
  - Why?

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** 1/4 that password is in dictionary
- **Work** is measured by number of hashes

Password Cracking: Case I

- Attack 1 specific password ***without*** using a dictionary
  - E.g., administrator's password
  - Must try $2^{56}/2 = 2^{55}$ on average
  - Like exhaustive key search
- Does **salt** help in this case?

Password Cracking: Case II

- Attack 1 specific password ***with*** dictionary
- With **salt**
  - Expected work: $1/4\ (2^{19}) + 3/4\ (2^{55}) \approx 2^{54.6}$
  - In practice, try all pwds in dictionary…
  - …then work is at most 2^{20} and probability of success is $1/4$
- What if **no salt** is used?
  - One-time work to compute dictionary: 2^{20}
  - Expected work is of same order as above
  - But with precomputed dictionary hashes, the "in practice" attack is essentially free…

Password Cracking: Case III

- Any of 1024 pwds in file, ***without*** dictionary
  - Assume all 2^{10} passwords are distinct
  - Need 2^{55} **comparisons** before expect to find pwd
- If **no salt** is used
  - Each computed hash yields 2^{10} comparisons
  - So expected work (hashes) is $2^{55}/2^{10} = 2^{45}$
- If **salt** is used
  - Expected work is 2^{55}
  - Each comparison requires a hash computation

Password Cracking: Case IV

- Any of 1024 pwds in file, ***with*** dictionary
  - Prob. one or more pwd in dict.: $1 - (3/4)^{1024} \approx 1$
  - So, we ignore case where no pwd is in dictionary
- If **salt** is used, expected work less than 2^{22}
  - See book, or slide notes for details
  - Work ≈ size of dictionary / P(pwd in dictionary)
- What if **no salt** is used?
  - If dictionary hashes not precomputed, work is about $2^{19}/2^{10} = 2^9$

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 attacks are too easy**
  - Often, one weak password will 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 problem

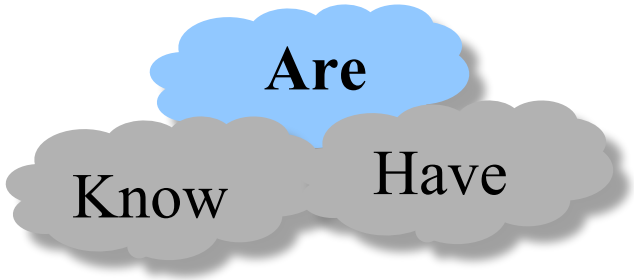
Password Cracking Tools

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

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?

- May be better than passwords
- But, 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 really that popular
  - Has not lived up to its promise/hype (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, and ???

Identification vs Authentication

- **Identification** – Who goes there?
  - Compare **one-to-many**
  - Example: 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 (mostly) 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 a weak point in real-world use
- 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 (also, terrorists in airports, etc.)
  - Often, less than ideal enrollment conditions
  - Subject will try to confuse recognition phase
- Cooperative subject makes it much easier
  - We are focused on authentication
  - So, we can assume subjects are 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 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* (murderer ID'ed by fingerprint)

Fingerprint History

- 1888 – Sir Francis Galton developed classification system
  - His system of "minutia" can be used today
  - Also verified that fingerprints do not change
- Some countries require fixed number of "points" (minutia) to match in criminal cases
  - In Britain, at least 15 points
  - In US, no fixed number of points

Fingerprint Comparison

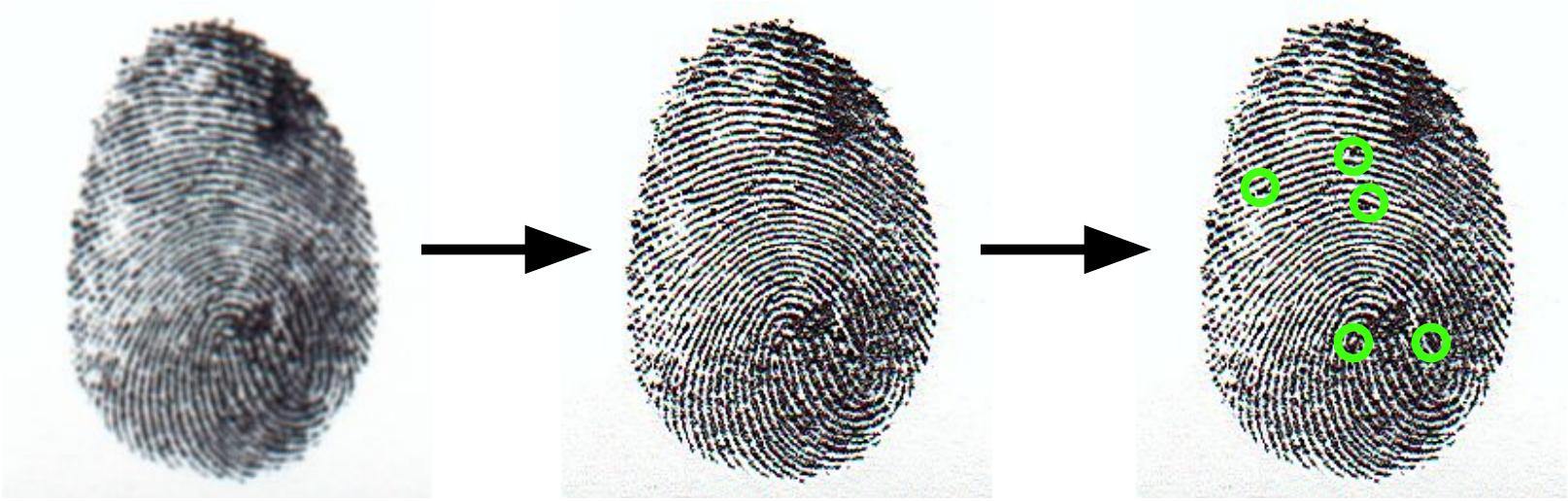
- Examples of **loops**, **whorls**, and **arches**
- Minutia extracted from these features

Loop (double)

Whorl

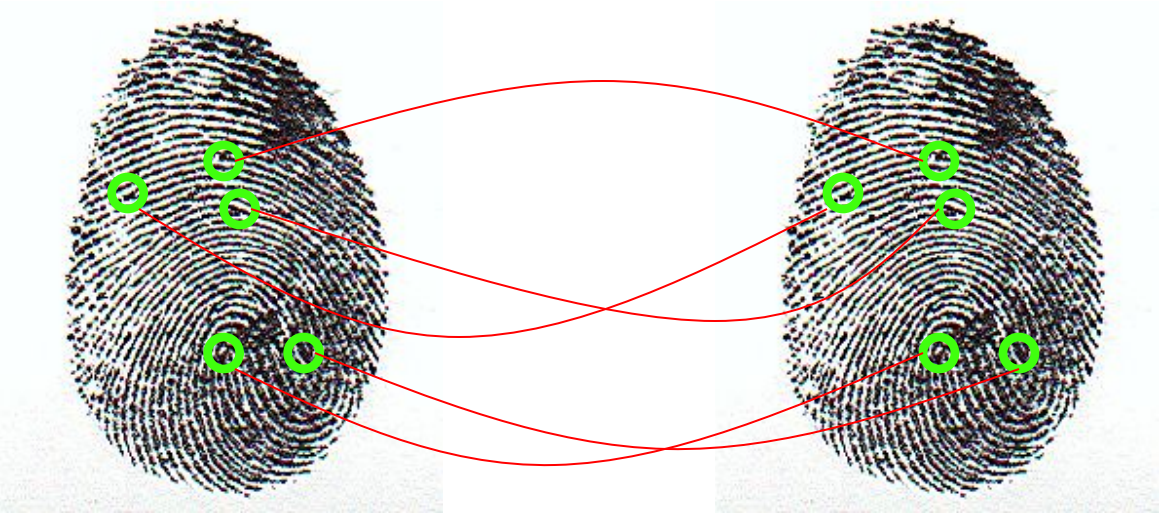
Arch

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?
- Aside: Do identical twins' fingerprints differ?

Hand Geometry

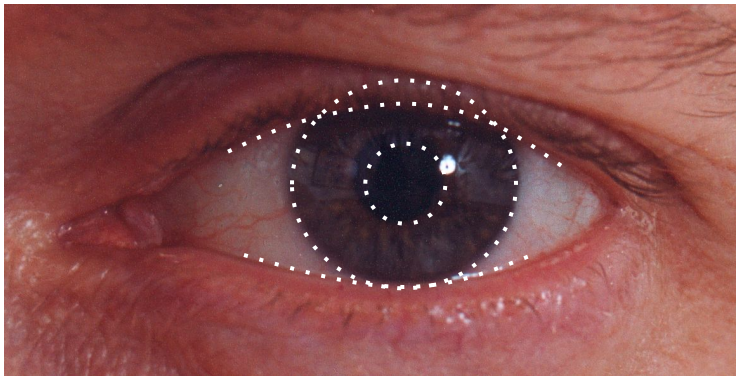
- A popular biometric
- Measures shape of hand
  - Width of hand, fingers
  - Length of fingers, etc.
- Human hands not so unique
- Hand geometry sufficient for many situations
- OK for authentication
- Not useful for ID problem

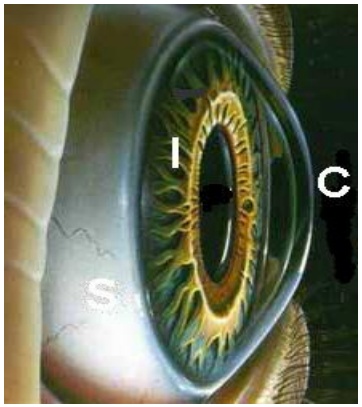
Hand Geometry

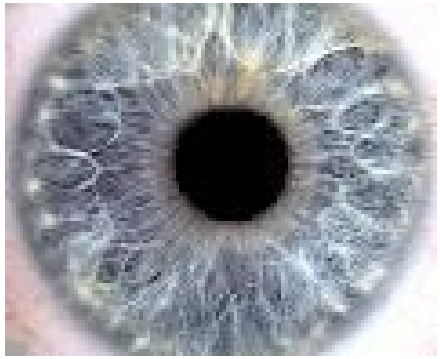
- Advantages
  - Quick 1 minute for enrollment, seconds for recognition
  - Hands are symmetric so what?
- Disadvantages
  - Cannot use on very young or very old
  - Relatively high equal error rate

5

Iris Patterns







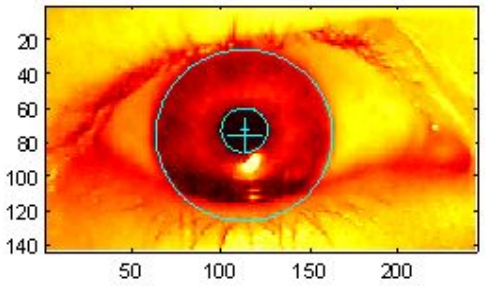
- Iris pattern development is "chaotic"
- Little or no genetic influence
- Even for identical twins, uncorrelated
- Pattern is stable through lifetime

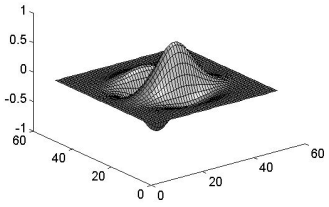
Iris Recognition: History

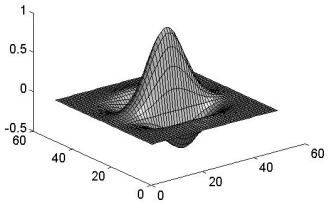
- 1936 – suggested by ophthalmologist
- 1980s – James Bond film(s)
- 1986 – first patent appeared
- 1994 – John Daugman patents new-and-improved technique
  - Patents owned by Iridian Technologies

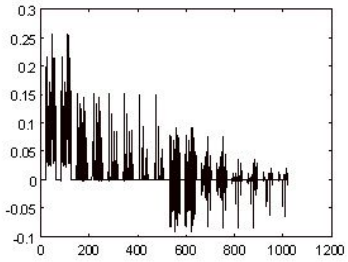
Iris Scan

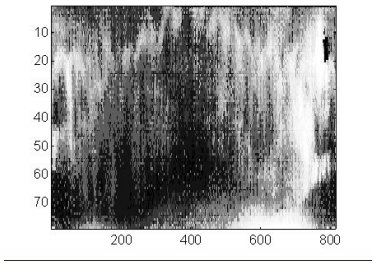
- Scanner locates iris
- Take b/w photo
- Use polar coordinates…
- 2-D wavelet transform
- Get 256 byte iris code











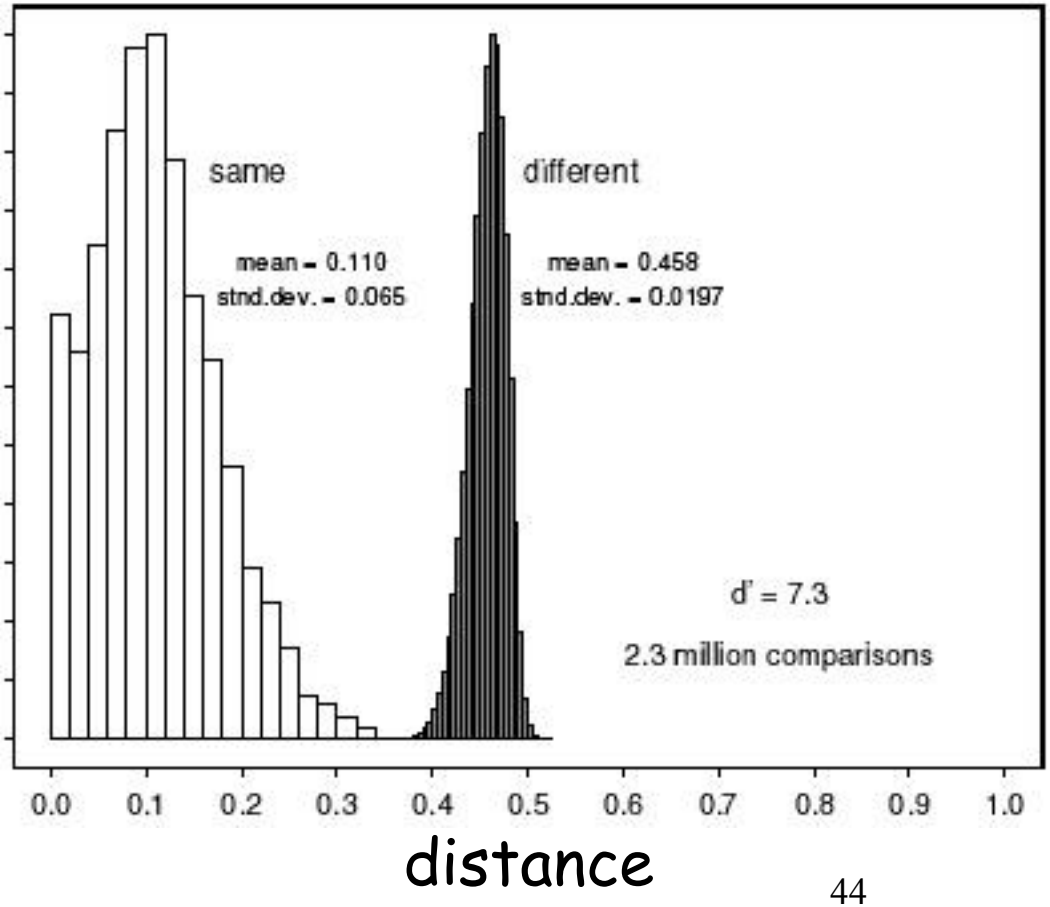
Measuring Iris Similarity

- 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 iris scan as match if distance < 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
  - Attacker could use photo of eye
- Afghan woman was authenticated by iris scan of old photo
  - Story can be found 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** biometrics used in practice have EER ranging from about 10^{-3} to as high as 5%
- **Hand geometry** has EER of about 10^{-3}
- In theory, **iris scan** has EER of about 10^{-6}
  - Enrollment phase may be critical to accuracy
- Most biometrics much worse than fingerprint!
- Biometrics useful for authentication…
  - …but for identification, not so impressive today

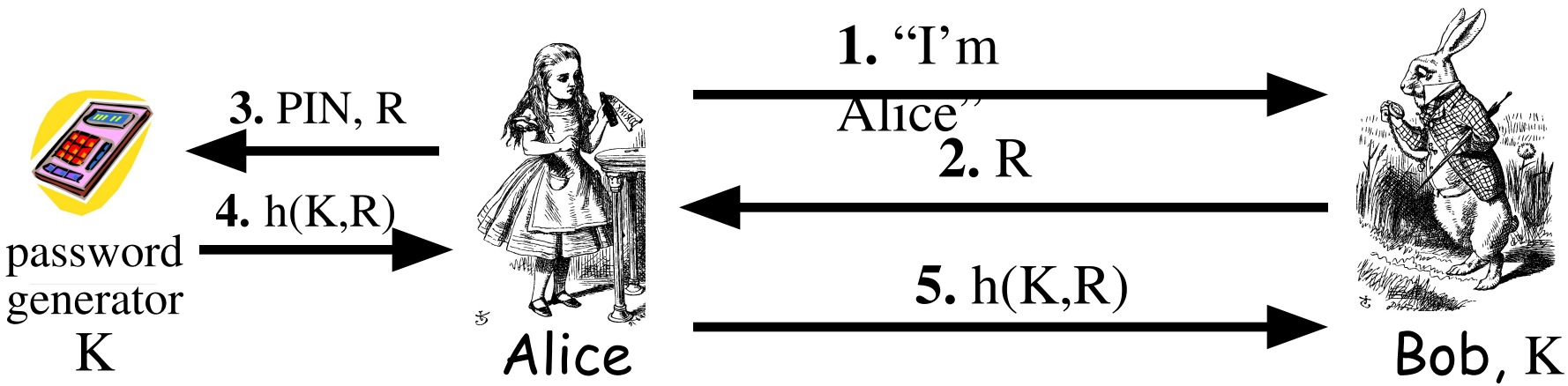
Biometrics: The Bottom Line

- 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 relatively 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 (next)
  - ATM card, smartcard, etc.

Password Generator



- Alice receives random "challenge" R from Bob
- Alice enters PIN and R in password generator
- Password generator hashes symmetric key K with R
- Alice sends "response" h(K,R) back to Bob
- Bob verifies response
- Note: Alice **has** pwd generator and **knows** PIN

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

Single Sign-on

- A hassle to enter password(s) repeatedly
  - Alice would like to authenticate only once
  - "Credentials" stay with Alice wherever she goes
  - Subsequent authentications transparent to Alice
- Kerberos a single sign-on protocol
- 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
- Sorta like a single sign-on for a website
  - But, very, very weak form of authentication
- Cookies also create privacy concerns

Authorization

Chapter 8: Authorization

It is easier to exclude harmful passions than to rule them,
and to deny them admittance
than to control them after they have been admitted.
 Seneca

You can always trust the information given to you
by people who are crazy;
they have an access to truth not available through regular channels.
 Sheila Ballantyne

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**
- But first, we look at system certification…

System Certification

- Government attempt to certify "security level" of products
- Of historical interest
  - Sorta like a history of authorization
- Still important today if you want to sell a product to the government
  - Tempting to argue it's a failure since government is so insecure, but…

Orange Book

- Trusted Computing System Evaluation Criteria (TCSEC), 1983
  - Universally known as the "orange book"
  - Name is due to color of it's cover
  - About 115 pages
  - Developed by U.S. DoD (NSA)
  - Part of the "rainbow series"
- Orange book generated a pseudo-religious fervor among some people
  - Less and less intensity as time goes by

Orange Book Outline

- Goals
  - Provide way to assess security products
  - Provide general guidance/philosophy on how to build more secure products
- Four ***divisions*** labeled D thru *A*
  - D is lowest, *A* is highest
- Divisions split into numbered ***classes***

D and C Divisions

- D minimal protection
  - Losers that can't get into higher division
- C discretionary protection, i.e., don't enforce security, just have means to detect breaches (audit)
  - C1 discretionary security protection
  - C2 controlled access protection
  - C2 slightly stronger than C1 (both vague)

B Division

- B mandatory protection
- B is a ***huge*** step up from C
  - C: break security, you might get caught
  - B: "mandatory", so you can't break it
- B1 labeled security protection
  - All data labeled, which restricts what can be done with it
  - This access control cannot be violated

B and A Divisions

- B2 – structured protection
  - Adds covert channel protection onto B1
- B3 – security domains
  - On top of B2 protection, adds that code must be ***tamperproof*** and "small"
- A – verified protection
  - Like B3, but ***proved*** using formal methods
  - Such methods still (mostly) impractical

Orange Book: Last Word

- Also a 2nd part, discusses rationale
- Not very practical or sensible, IMHO
- But some people insist we'd be better off if we'd followed it
- Others think it was a dead end
  - And resulted in lots of wasted effort
  - Aside… people who made the orange book, now set security education standards

Common Criteria

- Successor to the orange book (ca. 1998)
  - Due to inflation, more than 1000 pages
- An international government standard
  - And it reads like it…
  - Won't ever stir same passions as orange book
- CC is relevant in practice, but usually only if you want to sell to the government
- Evaluation Assurance Levels (EALs)
  - 1 thru 7, from lowest to highest security

EAL

- Note: product with high EAL may not be more secure than one with lower EAL
  - Why?
- Similarly, product with an EAL may not be any more secure than one without
  - Why?

EAL 1 thru 7

- EAL1 functionally tested
- EAL2 structurally tested
- EAL3 methodically tested, checked
- EAL4 ***designed***, tested, reviewed
- EAL5 semiformally designed, tested
- EAL6 verified, designed, tested
- EAL7 formally … (blah blah blah)

Common Criteria

- EAL4 is most commonly sought
  - Minimum needed to sell to government
- EAL7 requires formal proofs
  - Author could only find 2 EAL7 products…
- Who performs evaluations?
  - Government accredited labs, of course (for a hefty fee, like 6 figures)

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 view of authorization…
  - 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 100's of users, 10,000's of resources
  - Then matrix has 1,000,000's of entries
- How to manage such a large matrix?
- Note: We need to check this matrix before access to any resource by any user
- How to make this more efficient/practical?

Access Control Lists (ACLs)

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

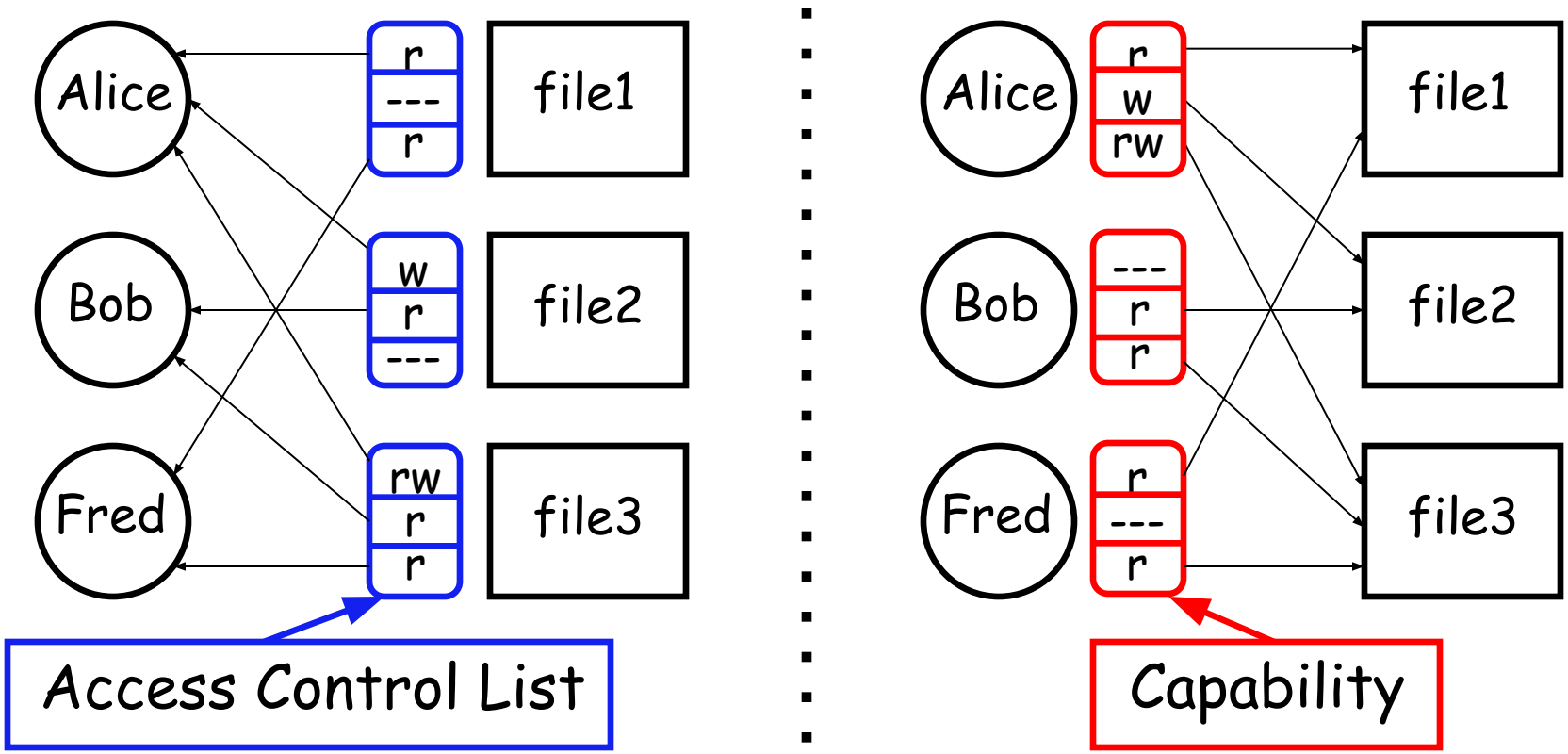
	OS	Accounting program	Accounting data	**Insurance data**	Payroll data
Bob	rx	rx	r		
Alice	rx	rx	r	rw	rw
Sam	rwx	rwx	r	rw	rw
Accounting program	rx	rx	rw	rw	rw

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

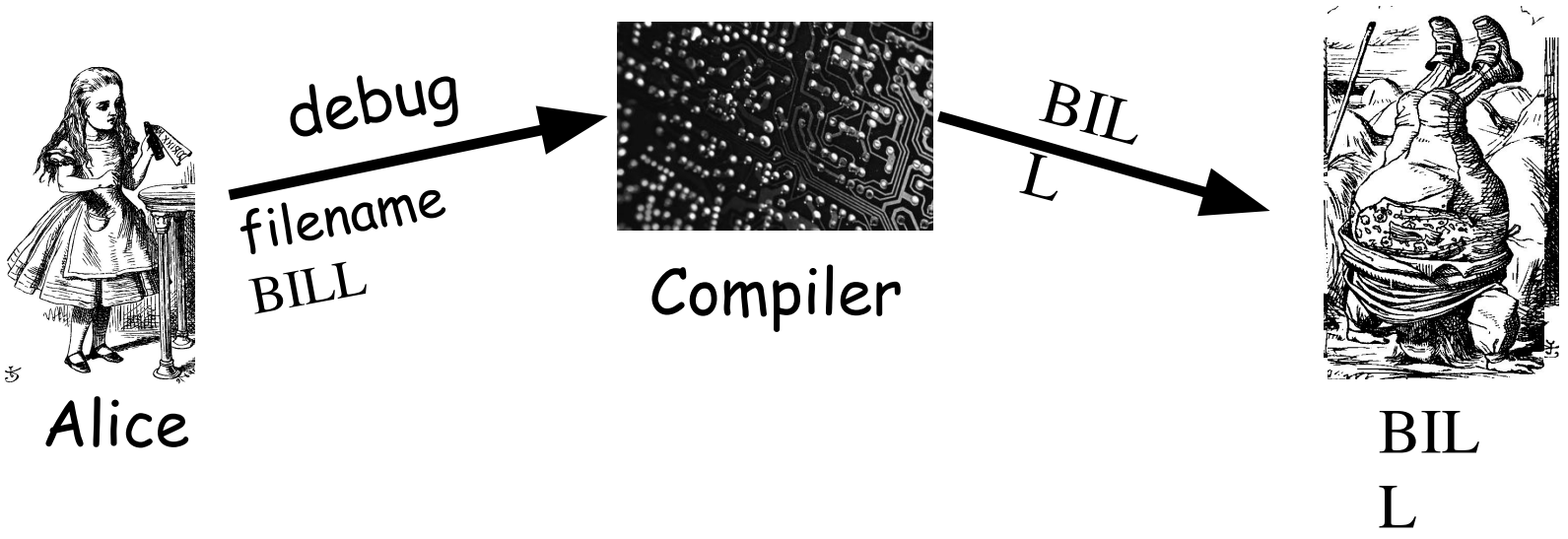
Confused Deputy

- 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, more difficult to prevent this
- With Capabilities, easier to prevent problem
  - Must maintain association between authority and intended purpose
- Capabilities 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
  - 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 (DoD) uses 4 levels:

  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 access **same system**
- MLS is a form of **Access Control**
- Military and government interest in MLS for many decades
  - Lots of research into MLS
  - Strengths and weaknesses of MLS well understood (almost entirely theoretical)
  - Many possible uses of MLS outside military

MLS Applications

- Classified government/military systems
- **Business example:** info restricted to
  - Senior management only, all management, everyone in company, or general public
- Network firewall
- Confidential medical info, databases, etc.
- Usually, MLS not really a technical system
  - More like part of a legal structure

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 $L(S) \leq L(O)$
- **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

BLP: The Bottom Line

- BLP is simple, 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
  - Models difficult to analyze, 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 $I(O) \leq I(S)$

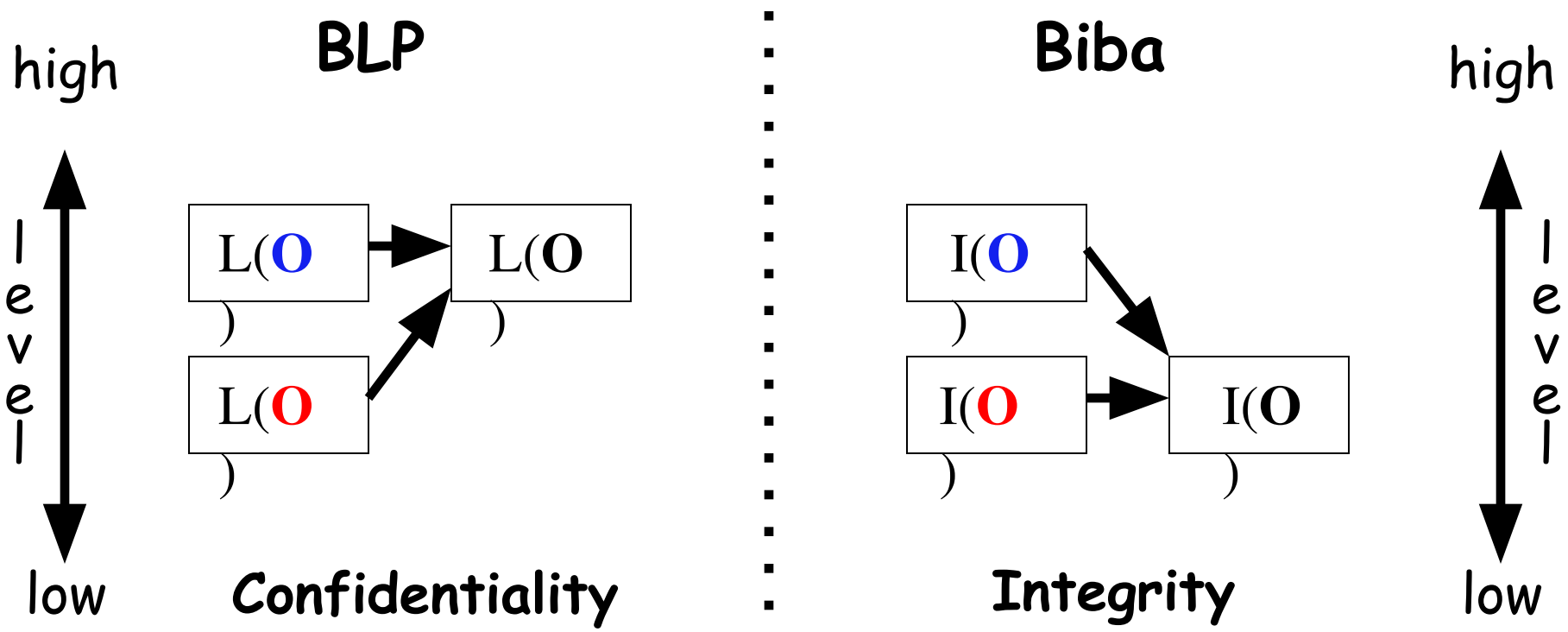
  (if S writes O, the integrity of $O \leq$ that of S)

  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


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

Compartments

Compartments

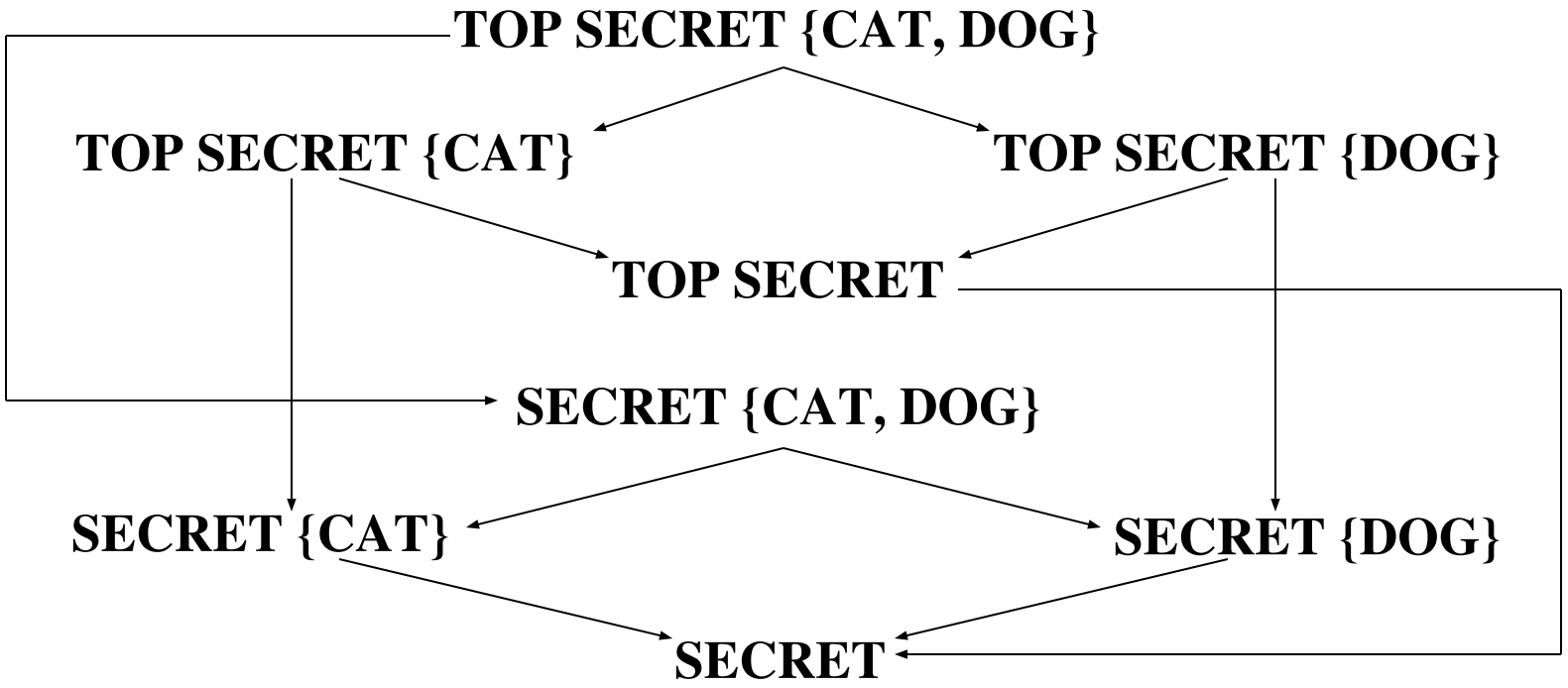
- Multilevel Security (MLS) enforces access control **up and down**
- Simple hierarchy of security labels is generally *not* flexible enough
- Compartments enforces restrictions **across**
- Suppose **TOP SECRET** divided into **TOP SECRET {CAT}** and **TOP SECRET {DOG}**
- Both are **TOP SECRET** but information flow restricted across the **TOP SECRET** level

Compartments

- 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 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

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**
  - What is the classification of an AIDS drug?
  - Everything tends toward **TOP SECRET**
  - Defeats the purpose of the system!
  - Compartments-only approach used instead

Covert Channel

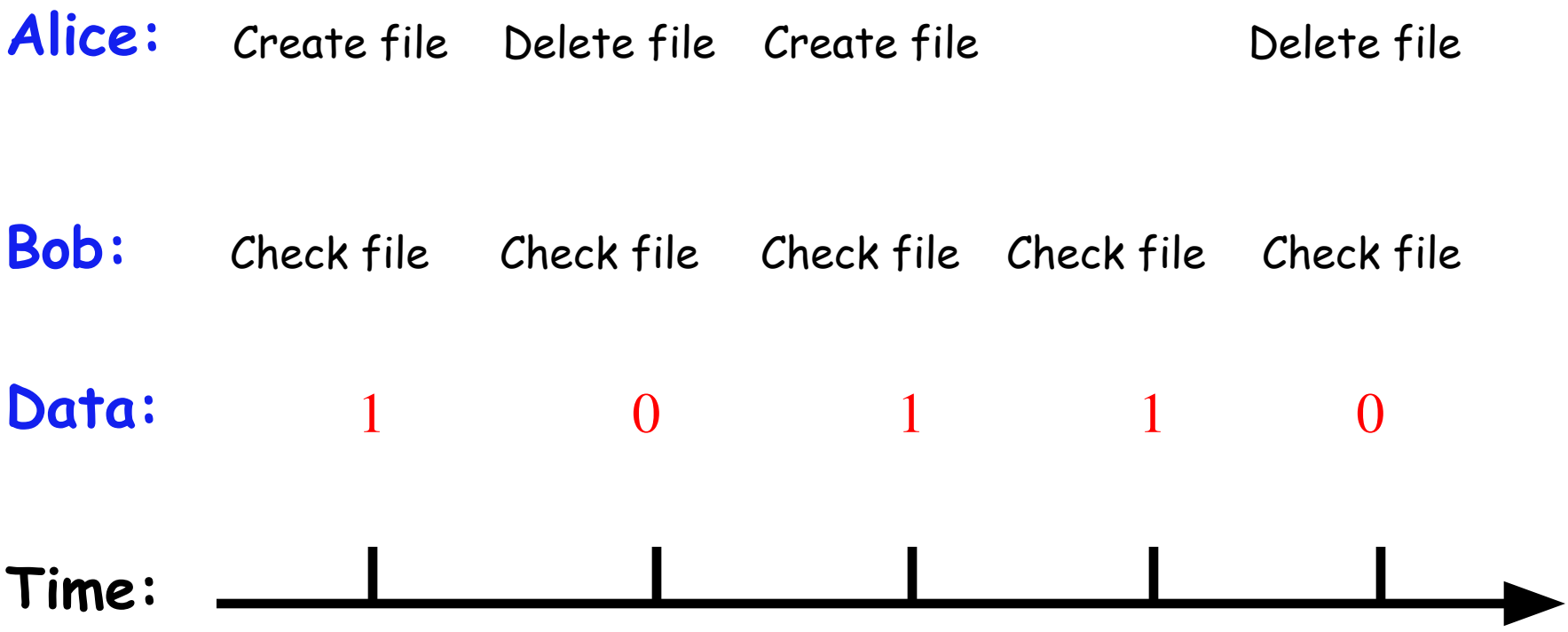
Covert Channel

- 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



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

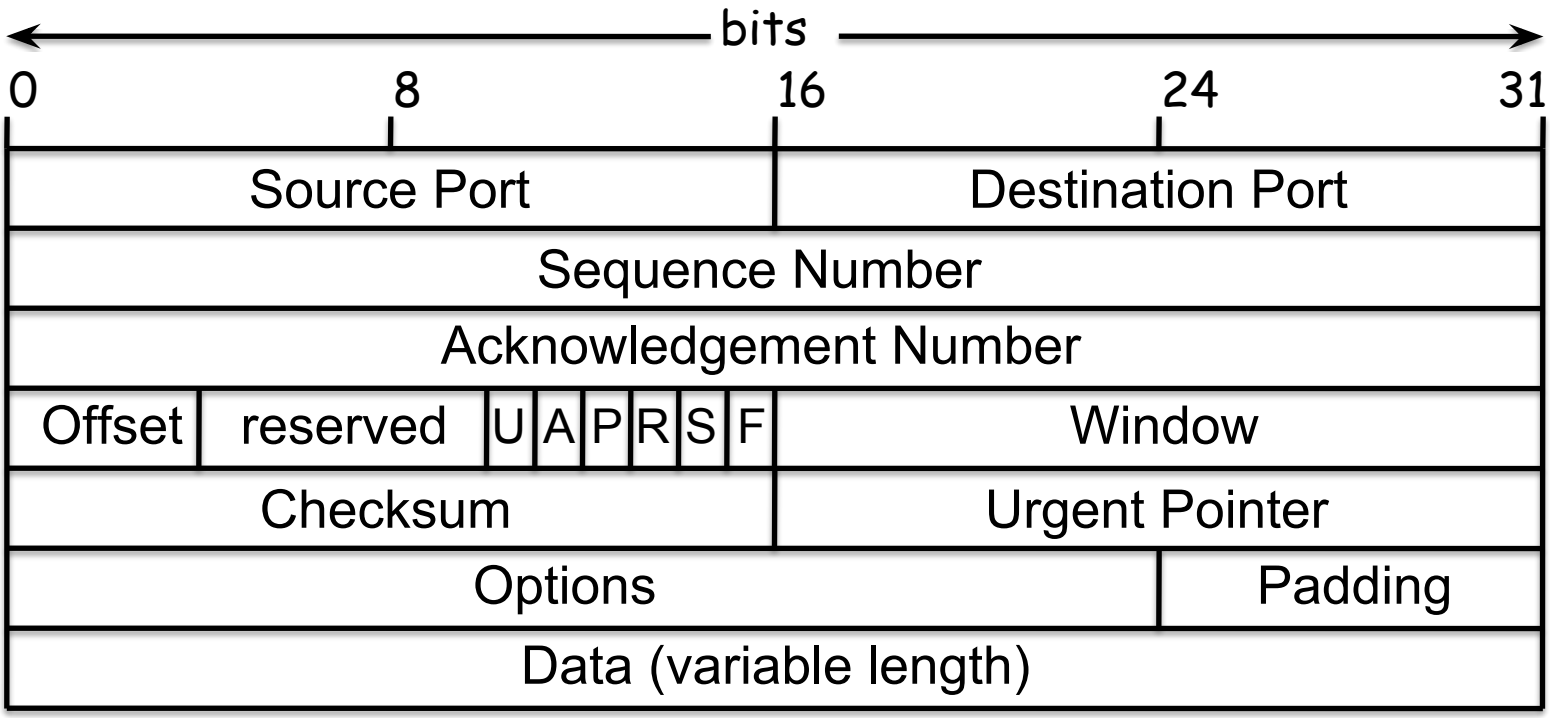
Covert Channel

- Potential covert channels are everywhere
- But, it's easy to eliminate covert channels:
  - "Just" eliminate all shared resources and all communication!
- Virtually impossible to eliminate covert channels in any **useful** information 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!

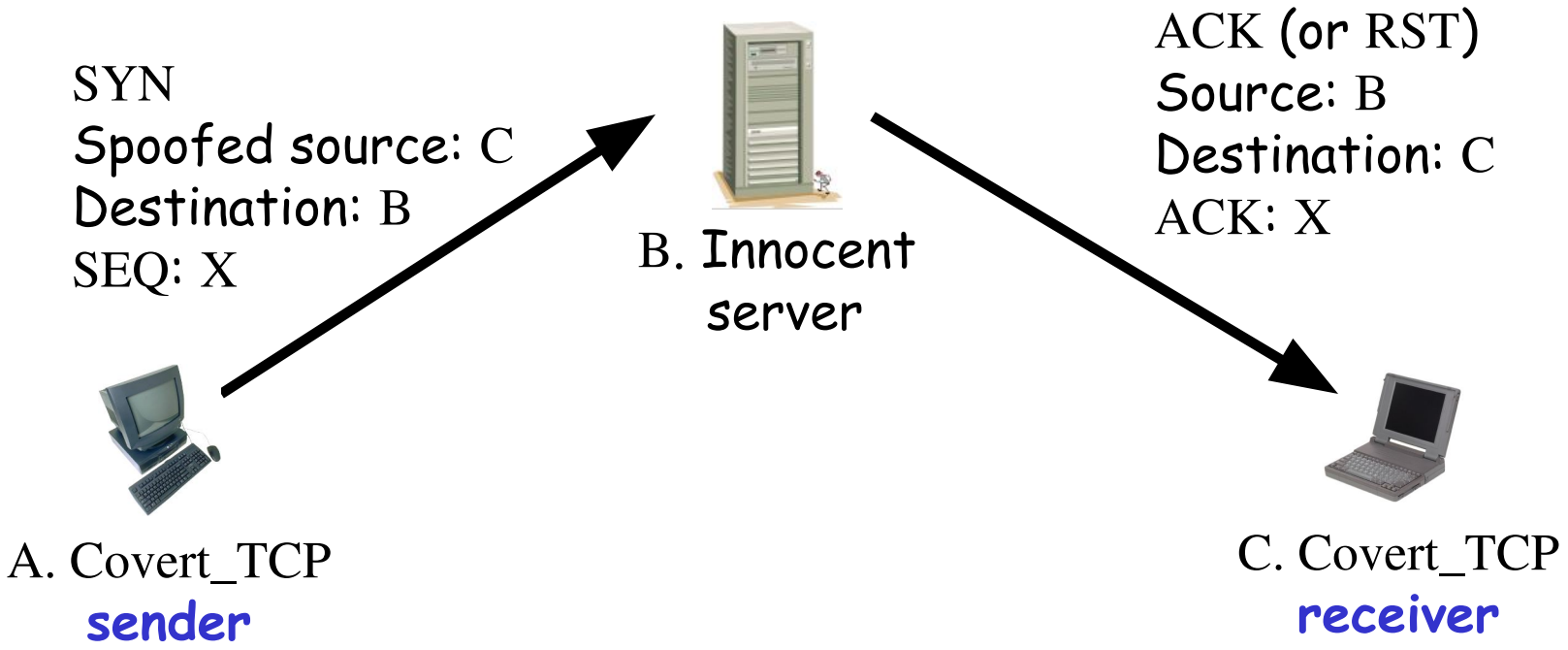
Real-World Covert Channel



- 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 & 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
  - This approach used by US Census Bureau
- Randomization
  - Add small amount of random noise to data
- Many other methods none satisfactory

Netflix Example

- Netflix prize $1M to first to improve recommendation system by 10% or more
- Netflix created dataset for contest
  - Movie preferences of real users
  - Usernames removed, some "noise" added
- Insufficient inference control
  - Researchers able to correlate IMDB reviews with those in Netflix dataset

Something Better Than Nothing?

- 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 a human and a computer, without seeing either
- If questioner cannot distinguish human from computer, computer passes
- This is the **gold standard** in AI
- No computer can pass this today
  - But some claim they are close to passing

CAPTCHA

- **CAPTCHA**
  - **C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part
- **C**ompletely **A**utomated test is generated and scored by a computer
- **P**ublic program and data are public
- **T**uring test to tell… humans can pass the test, but machines cannot
  - Also known as **HIP** == **H**uman **I**nteractive **P**roof
- 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 itself cannot pass!
- CAPTCHA purpose?
  - Only humans get access (not bots/computers)
- So, CAPTCHA is for **access control**

CAPTCHA Uses?

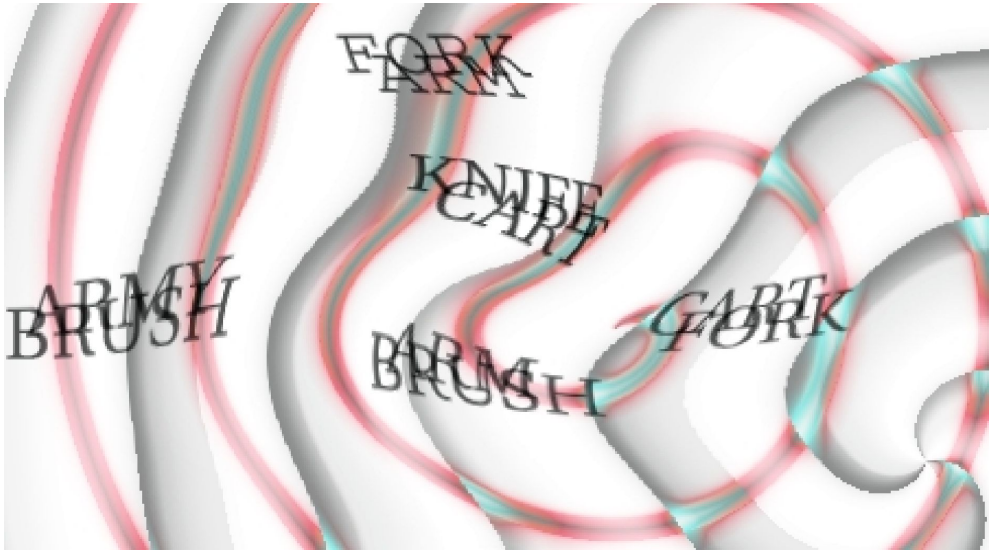
- Original motivation?
  - Automated bots stuffed ballot box in vote for best CS grad school
  - SJSU vs Stanford? No, it was MIT vs CMU
- Free email services spammers like to use bots to sign up for 1000s of email accounts
  - CAPTCHA employed so only humans get accounts
- Sites that do not want to 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
  - Similar to Kerckhoffs' Principle
- Good to have different CAPTCHAs in case someone cannot pass one type
  - E.g., blind person could not pass visual CAPTCHA

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

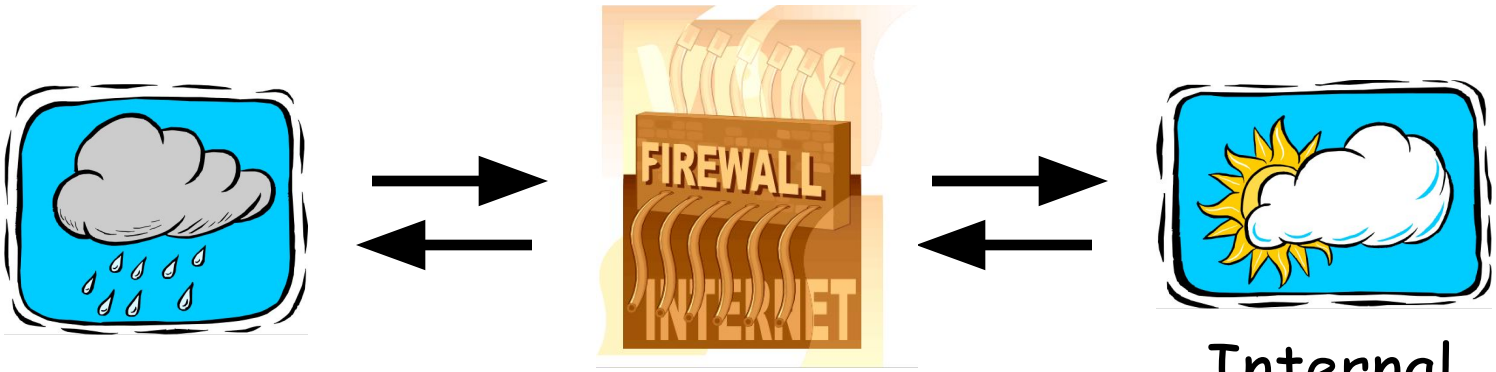
- Current types of CAPTCHAs
  - Visual like previous example
  - Audio distorted words or music
- No text-based CAPTCHAs
  - Maybe this is impossible…

CAPTCHA's and AI

- OCR is a challenging AI problem
  - Hardest part is the **segmentation problem**
  - 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 (such as OCR)
  - 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

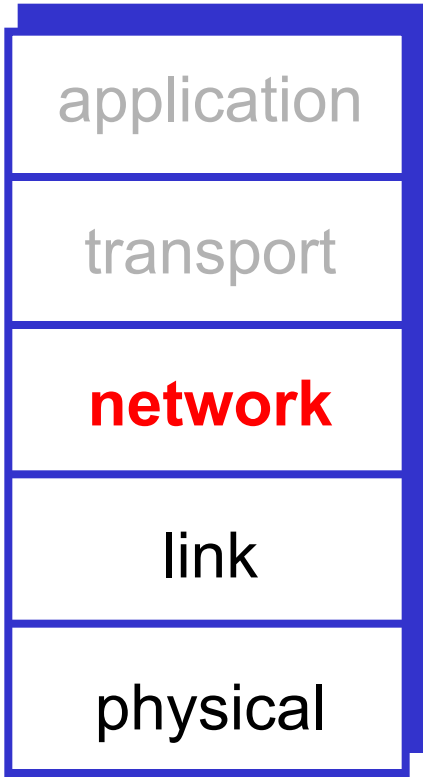
- A firewall is like a **secretary**
- To meet with an executive
  - First contact the secretary
  - Secretary decides if meeting is important
  - So, secretary filters out many requests
- You want to meet chair of CS department?
  - Secretary does some filtering
- You want to meet POTUS?
  - Secretary 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
- Lots of other terms often used
  - E.g., "deep packet inspection"

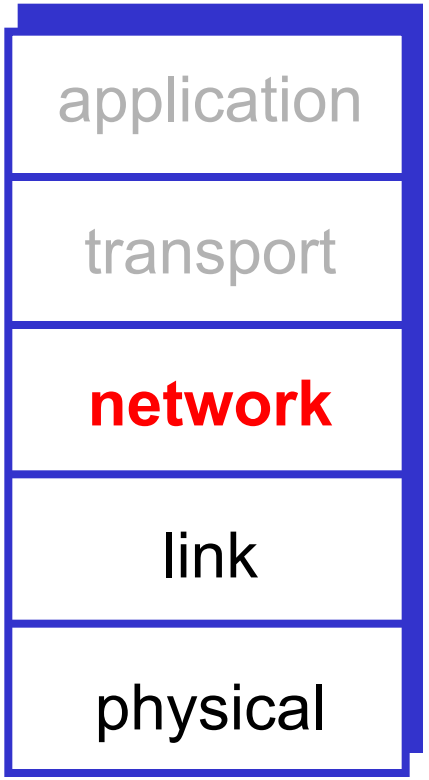
Packet Filter

- 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



Packet Filter

- Advantages?
  - Speed
- Disadvantages?
  - No concept of state
  - Cannot see TCP connections
  - Blind to application data



Packet Filter

- Configured via Access Control Lists (ACLs)
  - Different meaning than at start of Chapter 8

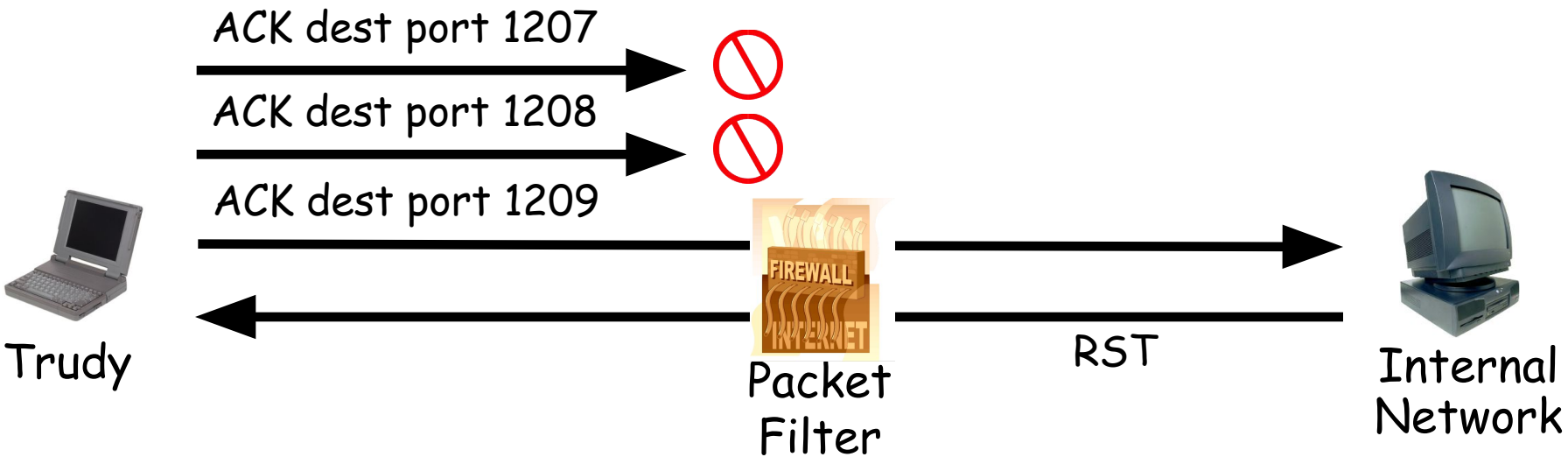
Action	Source IP	Dest IP	Source Port	Dest Port	Protocol	Flag Bits
Allow	Inside	Outside	Any	80	HTTP	Any
Allow	Outside	Inside	80	> 1023	HTTP	ACK
Deny	All	All	All	All	All	All

- **Q**: Intention?
- **A**: Restrict traffic to Web browsing

TCP ACK Scan

- Attacker scans for open ports thru firewall
  - Port scanning often *first step* in network attack
- 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

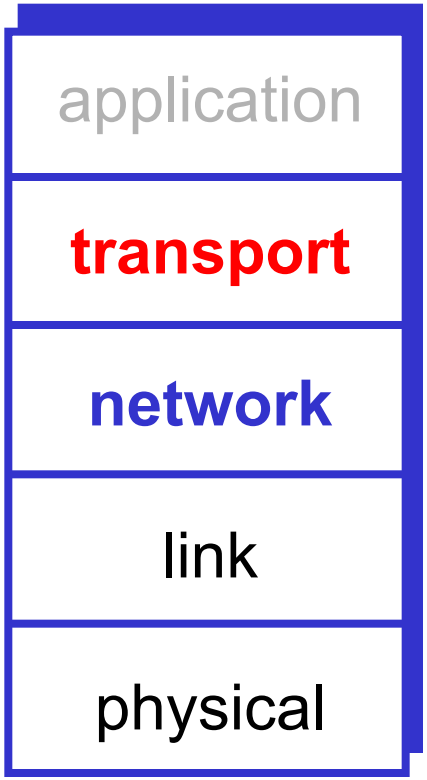
TCP ACK Scan



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

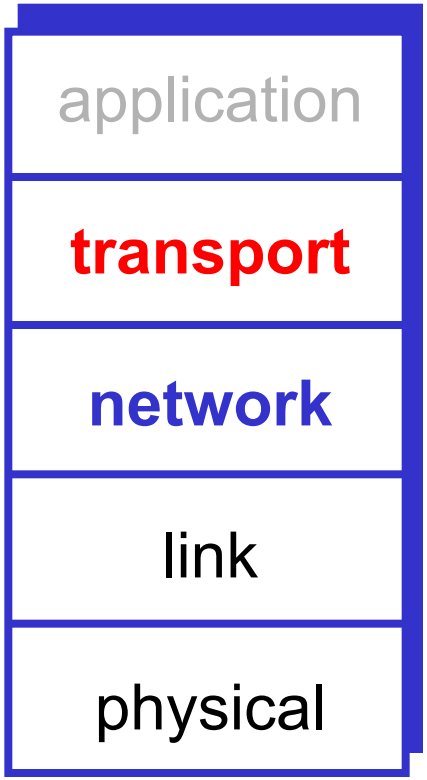
Stateful Packet Filter

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



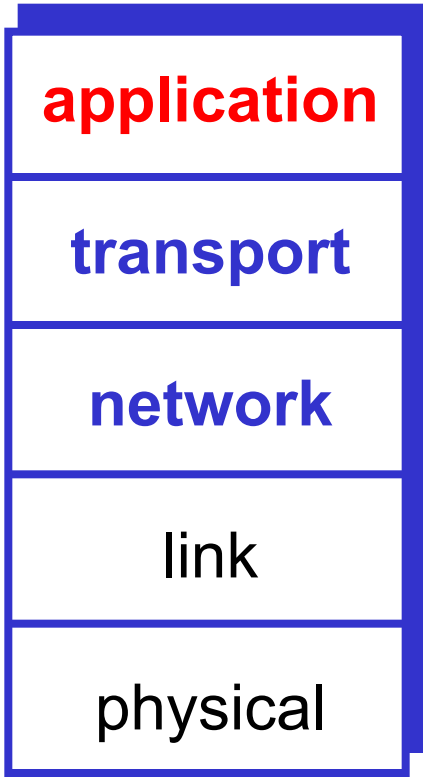
Stateful Packet Filter

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



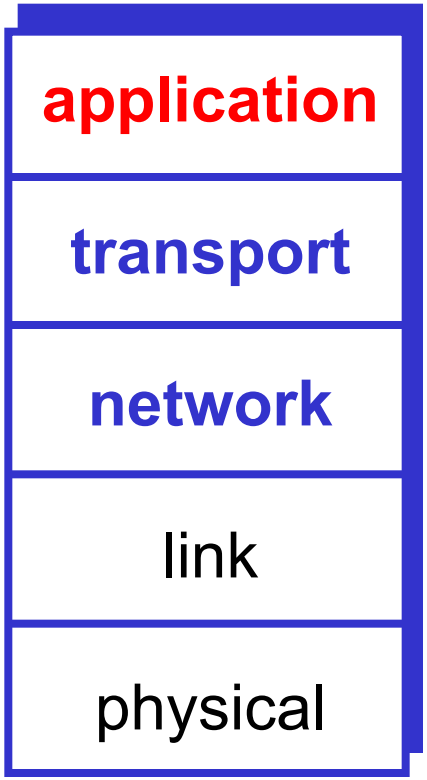
Application Proxy

- 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, Word macros)
- Disadvantages?
  - Speed



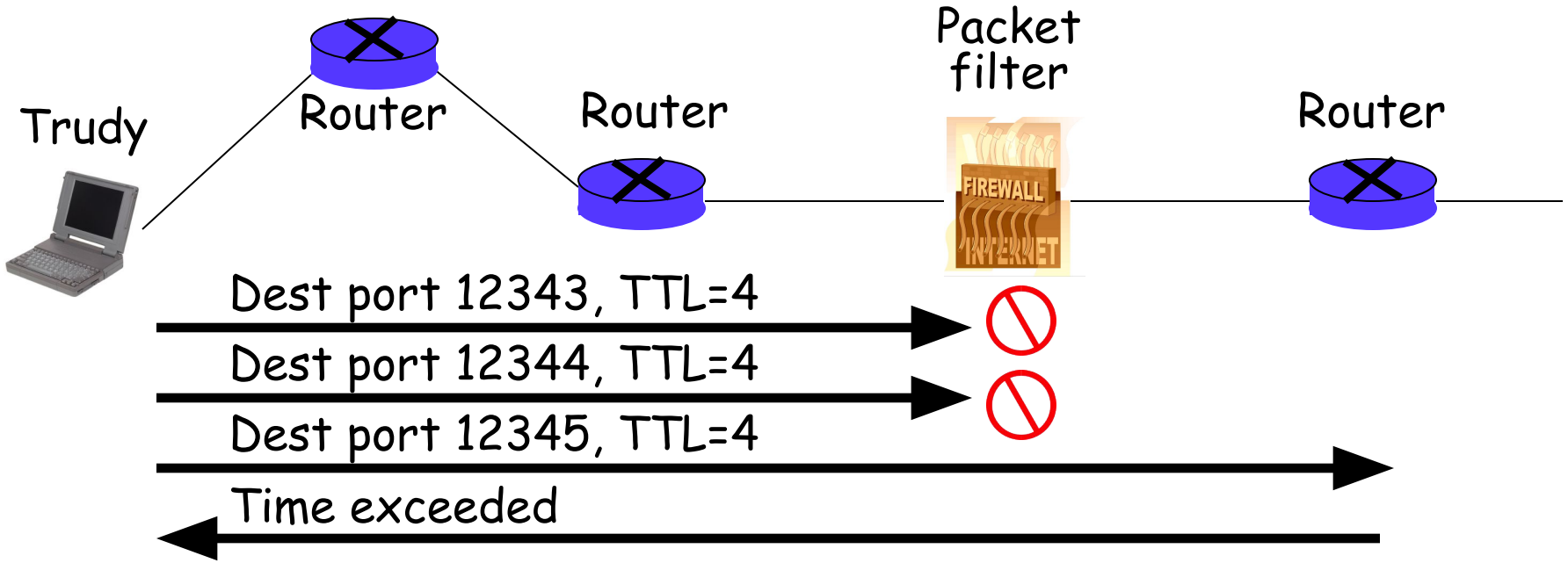
Application Proxy

- 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
- Can prevent some scans stateful packet filter cannot next slides

Firewalk

- Tool to scan for open ports thru firewall
- Attacker knows IP address of firewall and IP address of one system inside firewall
  - Set TTL to 1 more than number of hops to firewall, and set destination port to N
- If firewall allows data on port N thru firewall, get ***time exceeded*** error message
  - Otherwise, no response

Firewalk and Proxy Firewall



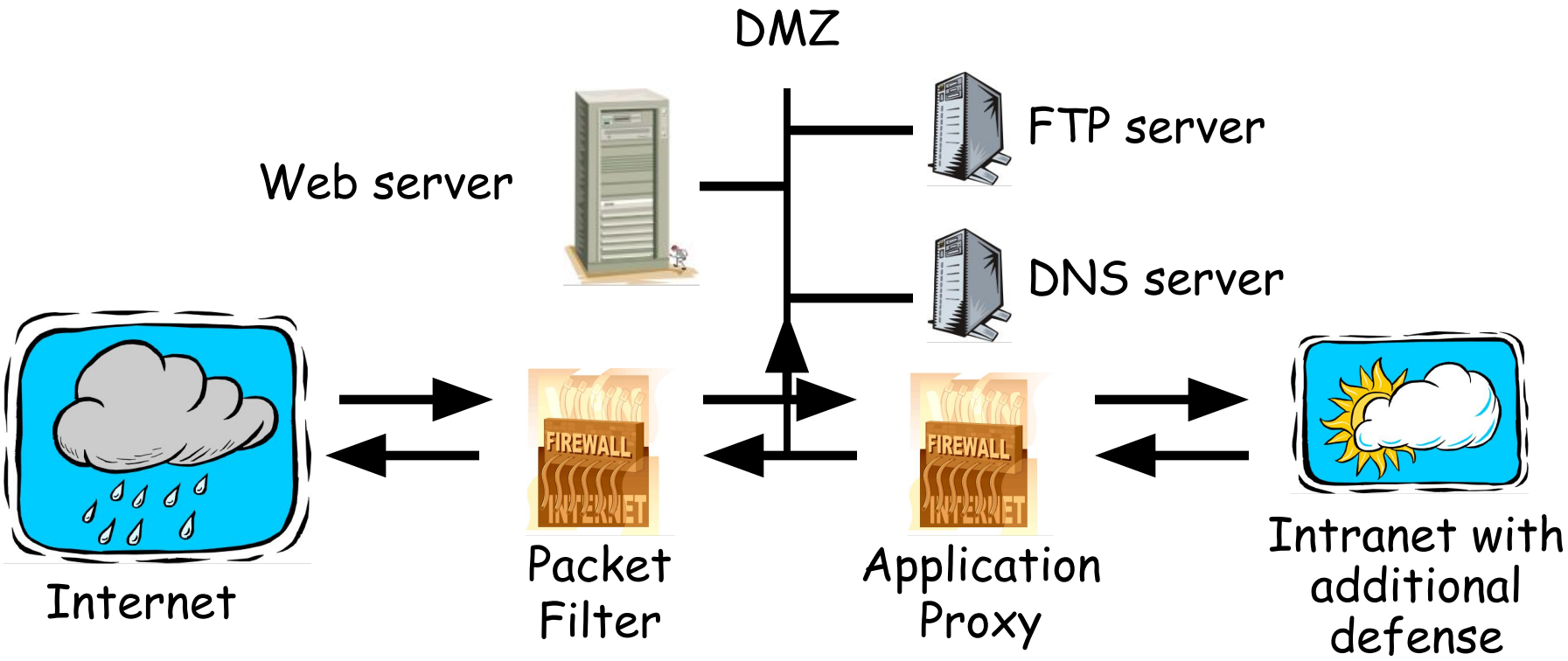
- This will **not** work thru an application proxy (why?)
- The proxy creates a new packet, destroys old TTL

Deep Packet Inspection

- Many buzzwords used for firewalls
  - One example: **deep packet inspection**
- What could this mean?
- Look into packets, but don't really "process" the packets
  - Like an application proxy, but faster

Firewalls and Defense in Depth

- Typical network 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 aimed at intrusion prevention
  - Like locking the door on your car

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
  - Network-based IDS
- Any IDS can be classified as above
  - In spite of marketing claims to the contrary!

Host-Based IDS

- Monitor activities on hosts for
  - Known attacks
  - 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.
- 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 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
  - Artificial intelligence
  - 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"
- Similar (but more sophisticated) approach is used in an IDS known as **NIDES**
- NIDES combines anomaly & signature IDS

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
  - CAPTCHA
  - Firewalls
  - IDS

Coming Attractions…

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