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

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

How do you prove to someone that you are who you claim to be?

Any system with access control must solve this problem

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

Good and Bad Passwords

- Bad passwords
  - frank
  - Fido
  - Password
  - incorrect
  - Pikachu
  - 102560
  - AustinStamp
- Good Passwords?
  - jfIej,43j-EmmL+y
  - 09864376537263
  - 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
  - **Group B:** Password based on passphrase → winner
  - **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 $y = h(x)$

Passwords

Windows Passwords

- Set or change password → Windows generates a LM hash and a NT hash.
- Two hashing functions used to encrypt passwords
  - LAN Manager hash (LM hash)
    - Password is padded with zeros until there are 14 characters.
    - It is then converted to uppercase and split into two 7-character pieces
    - Each half is encrypted using an 8-byte DES (data encryption standard) key
    - Result is combined into a 16-byte, one way hash value
  - NT hash (NT hash)
    - Converts password to Unicode and uses MD4 hash algorithm to obtain a 16-byte value
- Hashes are stored in the Security Accounts Manager database
  - Commonly known as " SAM" or "the SAM file"
- SAM is locked by system kernel when system is running.
  - File location: C:\WINNT\SYSTEM32\CONFIG
- SYSKEY

Passwords

Unix Passwords

- Uses modified DES as if it were a hash function
  - Encrypt NULL string using password as the key
    - Truncates passwords to 8 characters!
  - Artificial slowdown: run DES 25 times
  - Can instruct modern UNIXes to use MD5 hash function

- Problem: passwords are not truly random
  - With 52 upper- and lower-case letters, 10 digits and 32 punctuation symbols, there are 948 ≈ 6 quadrillion possible 8-character passwords
  - Humans like to use dictionary words, human and pet names ≈ 1 million common passwords
  - On average each person has 8-12 passwords:
  - Different systems impose different requirements on passwords.
  - Passwords need to be changed often.
  - Some passwords are used occasionally (once a year).

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?

Password
Impact on Security

What we found on Al Qaeda computers were two things:

1) Simple hacking tools are available to anyone who looks for them on the Internet.

2) Tools such as LOphtCrack allow admittance into almost anyone's account if a simple eight-digit password is used. People are frightened when they learn that using only an eight-digit password with standard numbers and letters will allow anyone to figure out their passwords in less than two minutes when one downloads a publicly available tool like LOphtCrack from the Internet. This was the kind of tool which we found, nothing terribly sophisticated.

-- Richard Clark, Presidents Advisor on Cyber Security (2001-2003)

Passwords

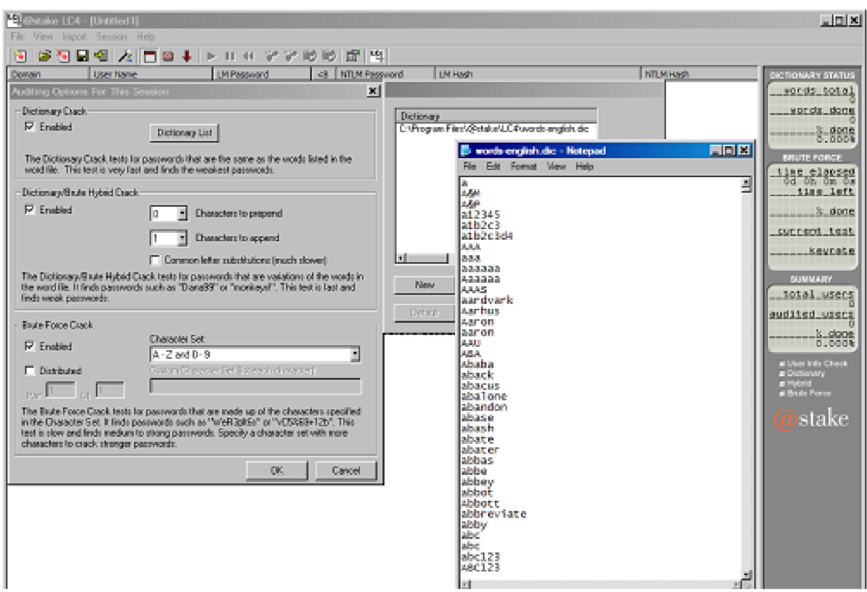
Methods of Attack

- Dictionary Attack
  - Quick technique that tries every word in a specific dictionary
- Hybrid Attack
  - Adds numbers or symbols to the end of a word
- Brute Force Attack
  - Tries all combinations of letters, numbers & symbols
- Popular programs for Windows password cracking
  - LC4
  - Sam Inside
  - Crack
  - John the Ripper (JTR)

Passwords

Dictionary Attack

- Password file /etc/passwd is world-readable
  - Contains user IDs and group IDs which are used by many system programs



- Dictionary attack is possible because many passwords come from a small dictionary
  - Attacker can compute H(word) for every word in the dictionary and see if the result is in the password file
  - With 1,000,000-word dictionary and assuming 10 guesses per second, brute-force online attack takes 50,000 seconds (14 hours) on average
    - This is very conservative. Offline attack is much faster!

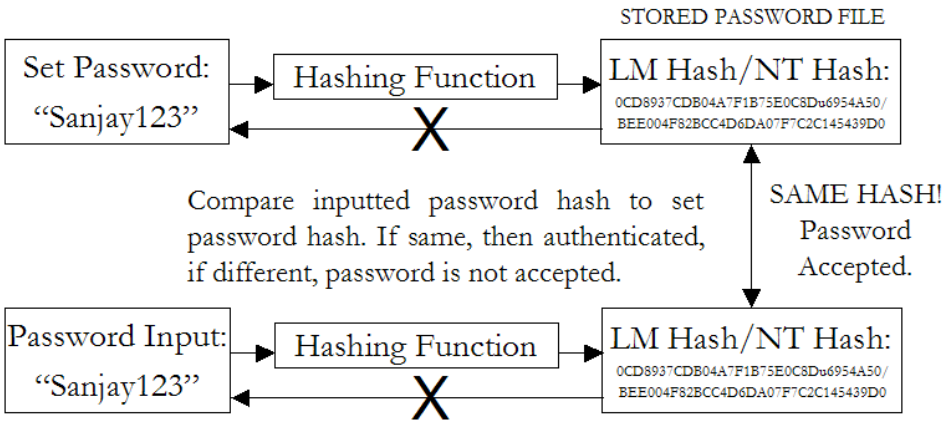
Passwords
Security Levels

- Filing System
  - Clear text
- Dedicated Authentication Server
  - Clear text
- Encrypted
  - Password + Encryption = bf4ee8HjaQkbw
- Hashed
  - Password + Hash function = aad3b435b51404eeaad3b435b51404ee
- Salted Hash
  - (Username + Salt + Password) + Hash function = e3ed2cb1f5e0162199be16b12419c012

Passwords

Hashing

- Instead of user password, store hash of password
- When user enters password, compute its hash and compare with entry in password file
  - System does not store actual passwords!

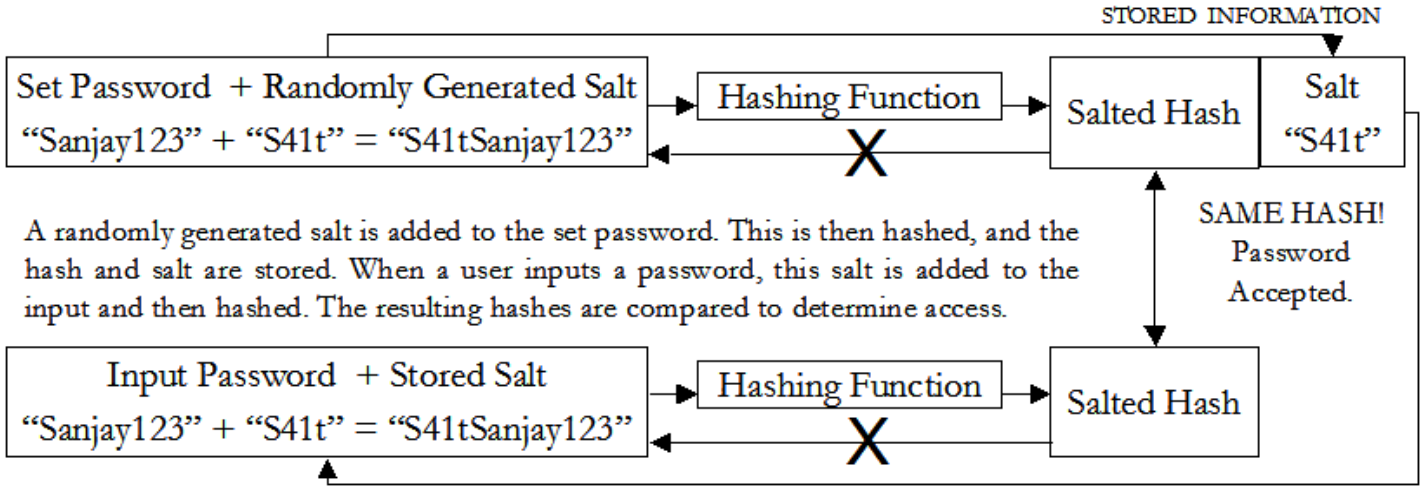


- Hash function H must have some properties
  - One-way: given H(password), hard to find password
    - No known algorithm better than trial and error
  - Collision-resistant: given H(password1), hard to find password2 such that H(password1)=H(password2)
    - It should even be hard to find any pair p1,p2 s.t. H(p1)=H(p2)

Passwords

Salting

- Salting requires adding a random piece of data and to the password before hashing it.
  - This means that the same string will hash to different values at different times
  - Users with the same password have different entries in the password file
  - Salt is stored with the data that is encrypted
- Hacker has to get the salt add it to each possible word and then rehash the data prior to comparing with the stored password.



Passwords

Salting Advantages

- Without salt, attacker can pre-compute hashes of all dictionary words once for all password entries
  - Same hash function on all UNIX machines
  - Identical passwords hash to identical values; one table of hash values can be used for all password files
- With salt, attacker must compute hashes of all dictionary words once for each password entry
  - With 12-bit random salt, same password can hash to 212 different hash values
  - Attacker must try all dictionary words for each salt value in the password file

Salt

- Hash password with **salt**
- Choose random salt s and compute

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

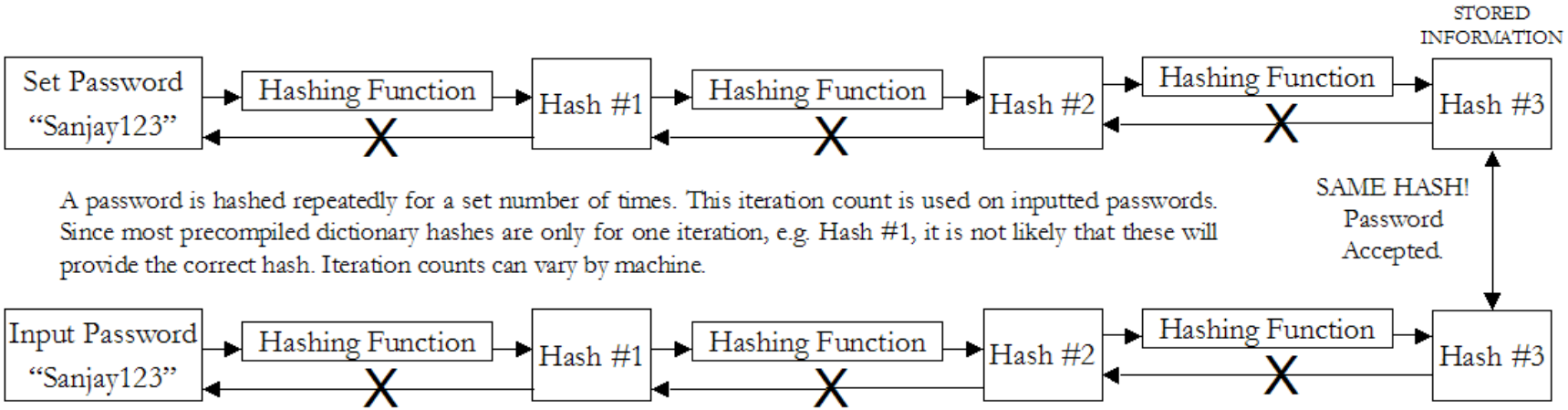
and store (s,y) in the password file

- Note that the salt s is not secret
  - Analogous to IV
- Still easy to verify salted password
- But lots more work for Trudy
  - Why?

Passwords

Iteration Count

- The same password can be rehashed many times over to make it more difficult for the hacker to crack the password.
- This means that the precompiled dictionary hashes are not useful since the iteration count is different for different systems
  - Dictionary attack is still possible!



Passwords

Shadow

- Utilized in UNIX systems
- Store hashed passwords in /etc/shadow file which is only readable by system administrator (root)
- Add expiration dates for passwords
- Early Shadow implementations on Linux called the login program which had a buffer overflow!

Passwords

Authentication Protocols

- Set of rules that governs the communication of data related to authentication between the server and the user
- TRANSFORMED PASSWORD
  - Password transformed using one way function before transmission
  - Prevents eavesdropping but not replay
- CHALLENGE-RESPONSE
  - Server sends a random value (challenge) to the client along with the authentication request. This must be included in the response
  - Protects against replay
- TIME STAMP
  - The authentication from the client to server must have time-stamp embedded
  - Server checks if the time is reasonable
  - Protects against replay
  - Depends on synchronization of clocks on computers
- ONE-TIME PASSWORD
  - New password obtained by passing user-password through one-way function n times which keeps incrementing
  - Protects against replay as well as eavesdropping

Passwords

Challenge Response

- User and system share a secret key
- Challenge: system presents user with some string
- Response: user computes response based on secret key and challenge
  - Secrecy: difficult to recover key from response
    - One-way hashing or symmetric encryption work well
  - Freshness: if challenge is fresh and unpredictable, attacker on the network cannot replay an old response
    - For example, use a fresh random number for each challenge
- Good for systems with pre-installed secret keys
  - Car keys; military friend-or-foe identification

Passwords

Improving Security

- Add biometrics
  - For example, keystroke dynamics or voiceprint
  - Revocation is often a problem with biometrics
- Graphical passwords
  - Goal: increase the size of memorable password space
- Rely on the difficulty of computer vision
  - Face recognition is easy for humans, hard for machines
  - Present user with a sequence of faces, he must pick the right face several times in a row to log in
- Other examples
  - Click on a series of pictures in order
  - Drawing a picture
  - Clicking four correct points on a picture

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 $\approx$ 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

Passwords
Protection/Detection

Protection:

- Disable storage of LAN Manager hashes.
- Configure both Local and Domain Account Policies (Password & Account Lockout Policies).
- Audit access to important files.
- Implement SYSKEY security on all systems.
- Set BIOS to boot first from the hard drive.
- Password-protect the BIOS.
- Enforce strong passwords!
- Change your passwords frequently.
- Use two or three factor authentication.
- Use one time passwords.

Passwords

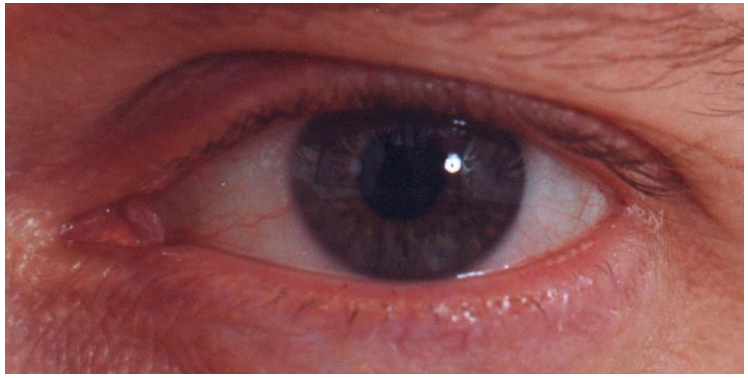
Ten Common Mistakes

1. Leaving passwords blank or unchanged from default value.
2. Using the letters p-a-s-s-w-o-r-d as the password.
3. Using a favorite movie star name as the password.
4. Using a spouse's name as the password.
5. Using the same password for everything.
6. Writing passwords on post-it notes.
7. Pasting a list of passwords under the keyboard.
8. Storing all passwords in an Excel spreadsheet on a PDA or inserting passwords into a rolodex.
9. Writing all passwords in a personal diary.
10. Giving the password to someone who claims to be the system administrator.

Password Cracking Tools

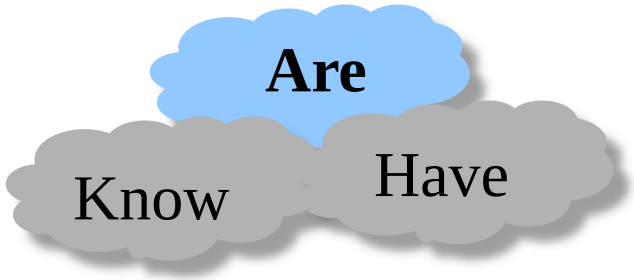
- Popular password cracking tools
  - Password Crackers
  - Password Portal
  - LOphtCrack and LC4 (Windows)
  - John the Ripper (Unix)
- Admins should use these tools to test for weak passwords since attackers will
- Good articles on password cracking
  - Passwords - Conerstone of Computer Security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

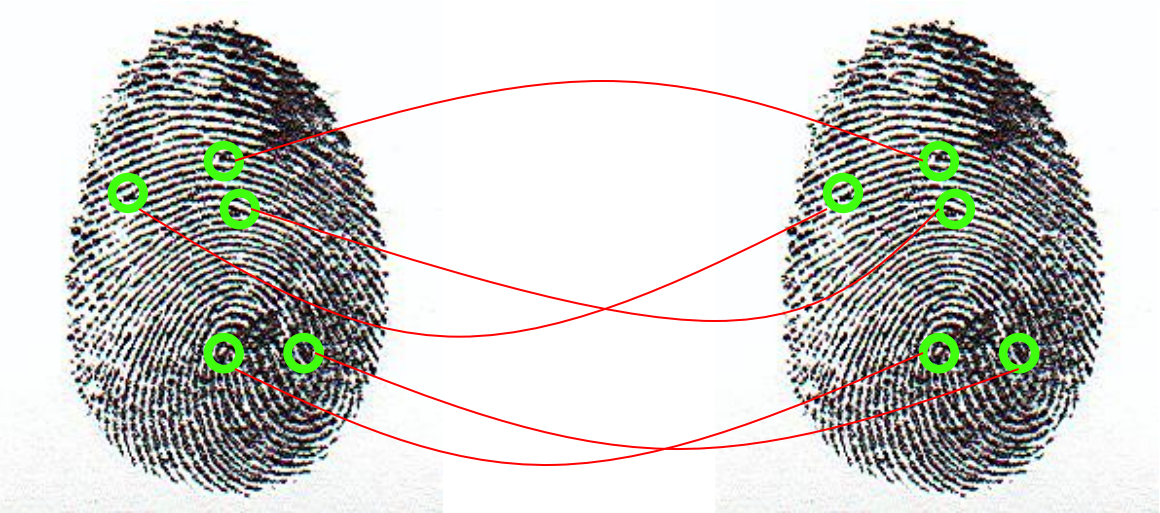
Whorl

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?

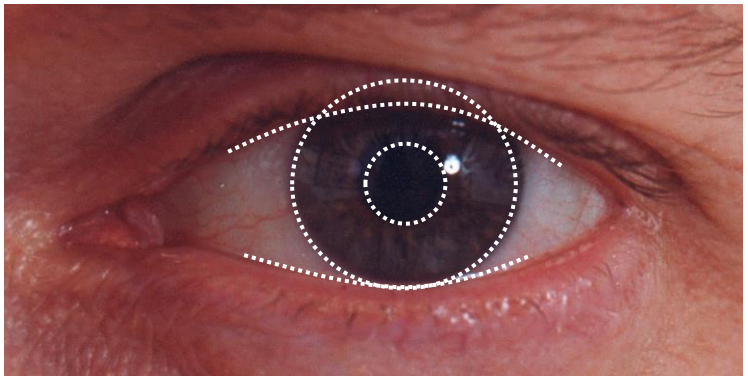
Hand Geometry

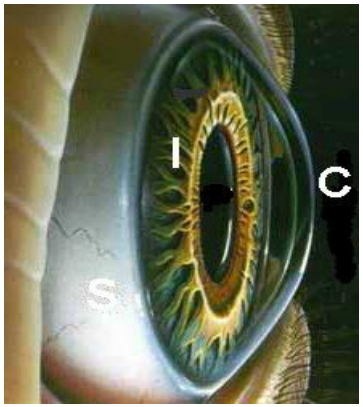
- A popular biometric
- Measures shape of hand
  - Width of hand, fingers
  - Length of fingers, etc.
- Human hands not so unique
- Hand geometry sufficient for many situations
- OK for authentication
- Not useful for ID problem

Hand Geometry

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

Iris Patterns





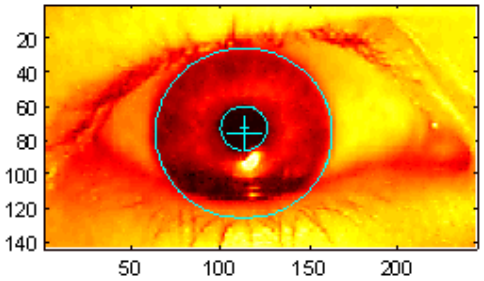
- Iris pattern development is "chaotic"
- Little or no genetic influence
- Even for identical twins, uncorrelated
- Pattern is stable through lifetime

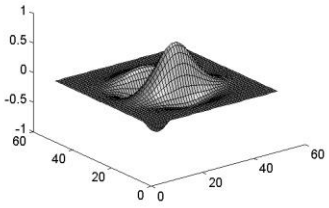
Iris Recognition: History

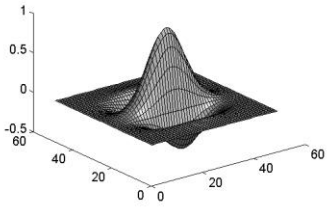
- 1936 — suggested by ophthalmologist
- 1980s — James Bond film(s)
- 1986 — first patent appeared
- 1994 — John Daugman patents new-and-improved technique
  - Patents owned by Iridian Technologies

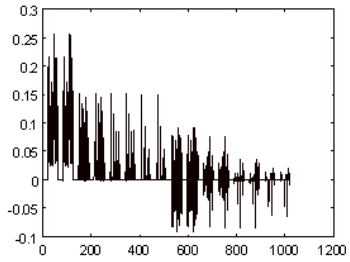
Iris Scan

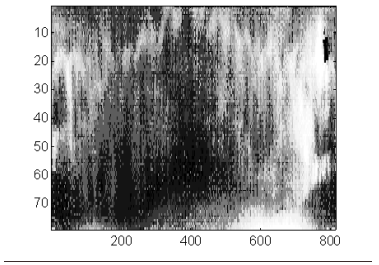
- Scanner locates iris
- Take b/w photo
- Use polar coordinates…
- 2-D wavelet transform
- Get 256 byte iris code











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

Passwords

Personal Token Authentication

- Personal Tokens are hardware devices that generate unique strings that are usually used in conjunction with passwords for authentication
- A variety of different physical forms of tokens exist
  - e.g. hand-held devices, Smart Cards, PCMCIA cards, USB tokens
- Different types of tokens exist:
  - **Storage Token:** A secret value that is stored on a token and is available after the token has been unlocked using a PIN
  - **Synchronous One-time Password Generator:** Generate a new password periodically (e.g. each minute) based on time and a secret code stored in the token
  - **Challenge-response:** Token computes a number based on a challenge value sent by the server
  - **Digital Signature Token:** Contains the digital signature private key and computes a computes a digital signature on a supplied data value

Passwords

Biometric Authentication

- Uses certain biological characteristics for authentication
  - Biometric reader measures physiological indicia and compares them to specified values
  - It is not capable of securing information over the network
- Different techniques exist
  - Fingerprint Recognition
  - Voice Recognition
  - Handwriting Recognition
  - Face Recognition
  - Retinal Scan
  - Hand Geometry Recognition

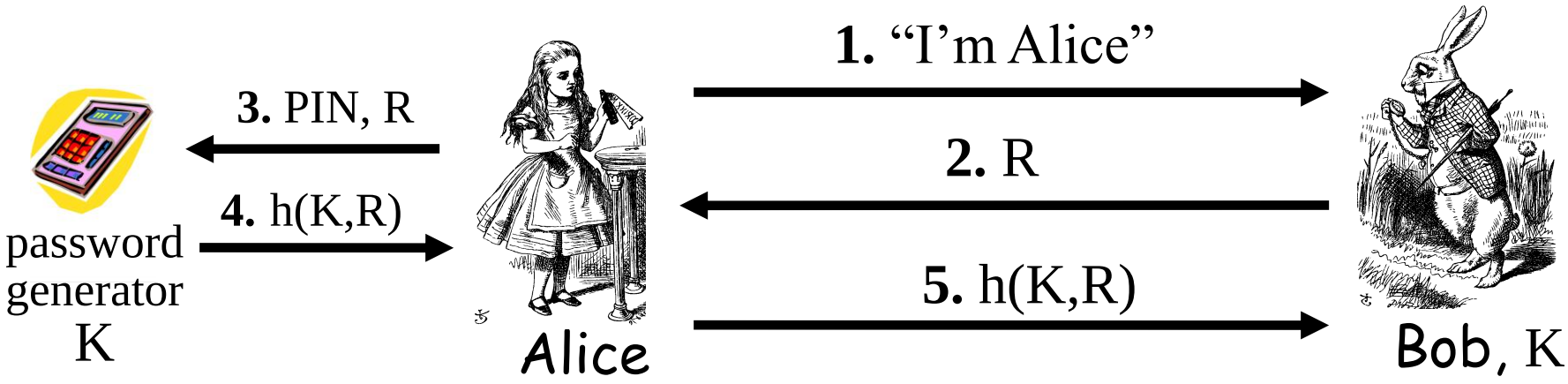
Biometrics: The Bottom Line

- 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

Covert Channel

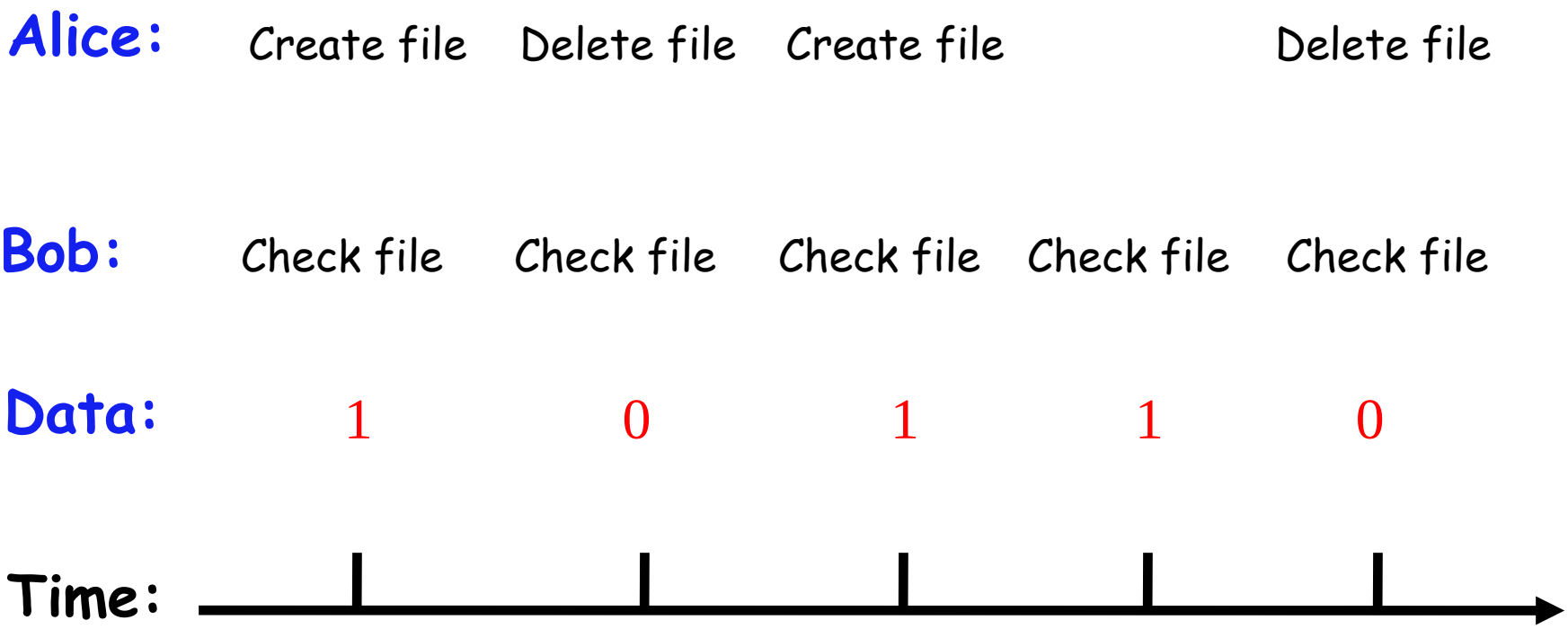
Covert Channel

- 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

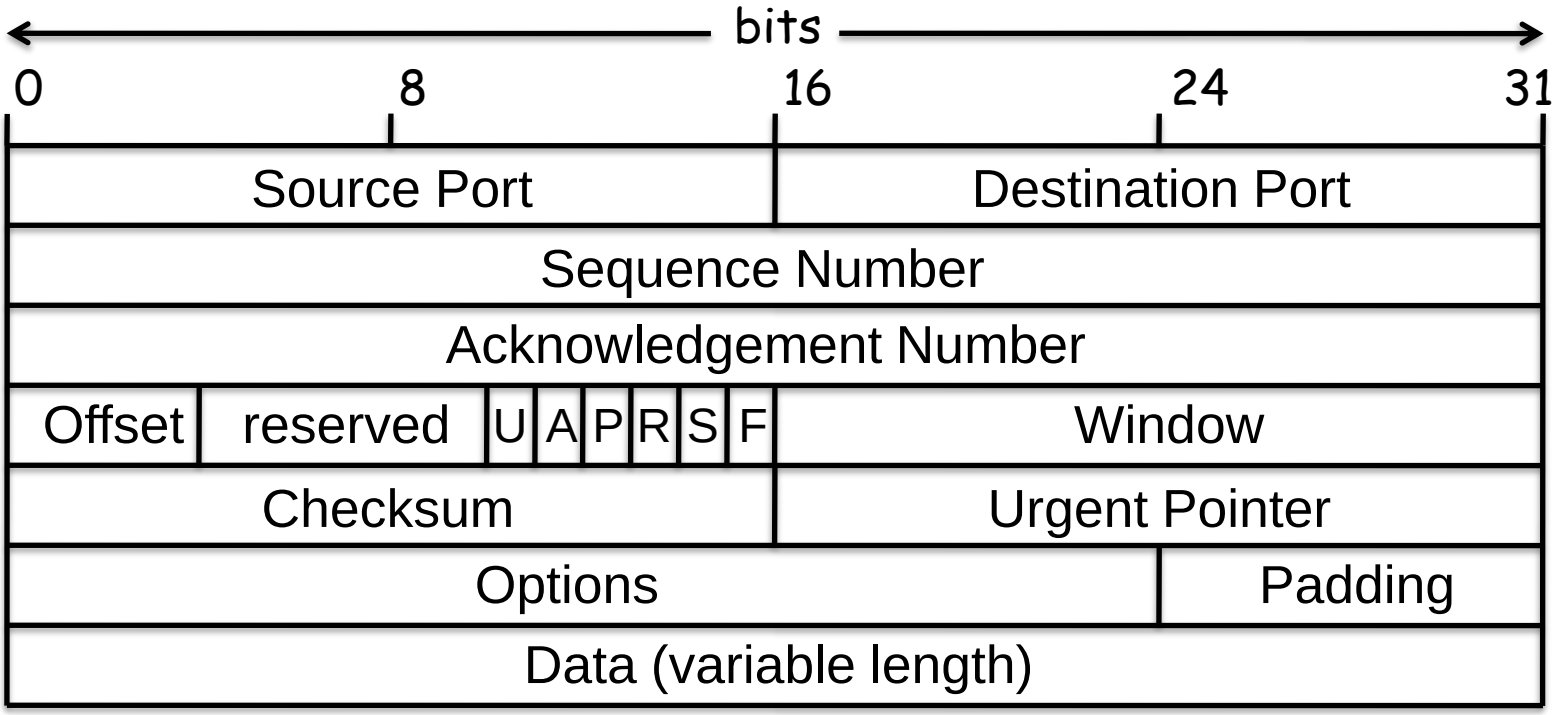
Covert Channel

- 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!

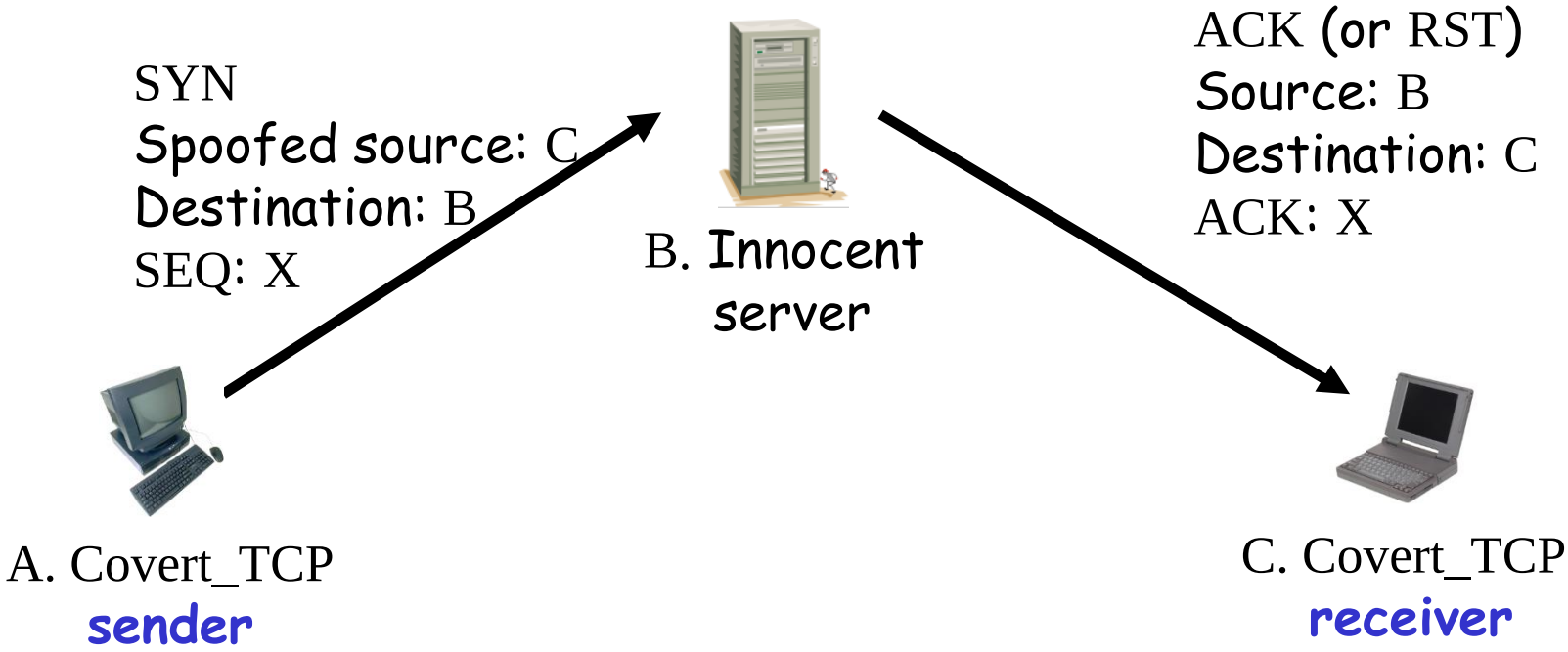
Real-World Covert Channel



- 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



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?

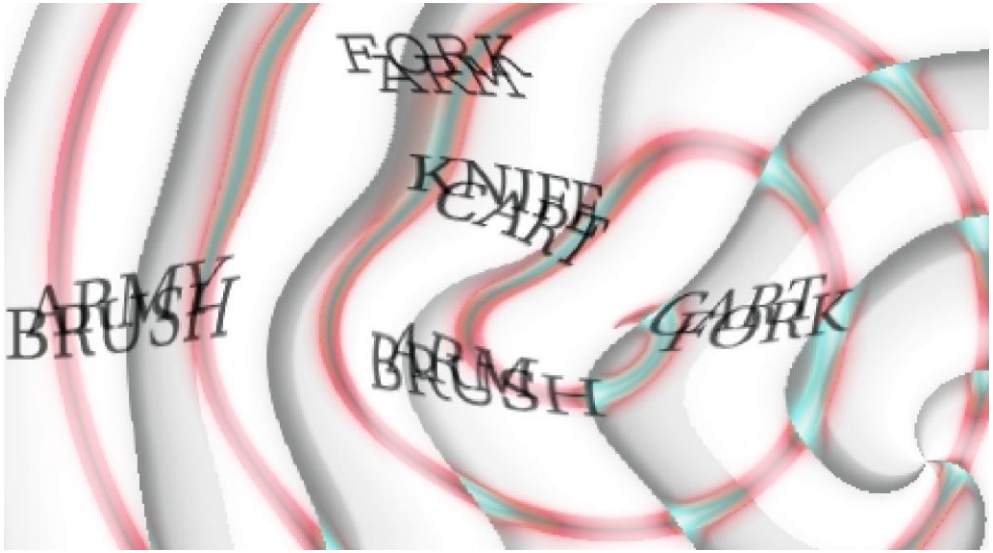
- Original motivation?
  - Automated bots stuffed ballot box in vote for best CS grad school
  - SJSU vs Stanford? No, it was MIT vs CMU
- Free email services — spammers like to use bots to sign up for 1000s of email accounts
  - CAPTCHA employed so only humans get accounts
- Sites that do not want to 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

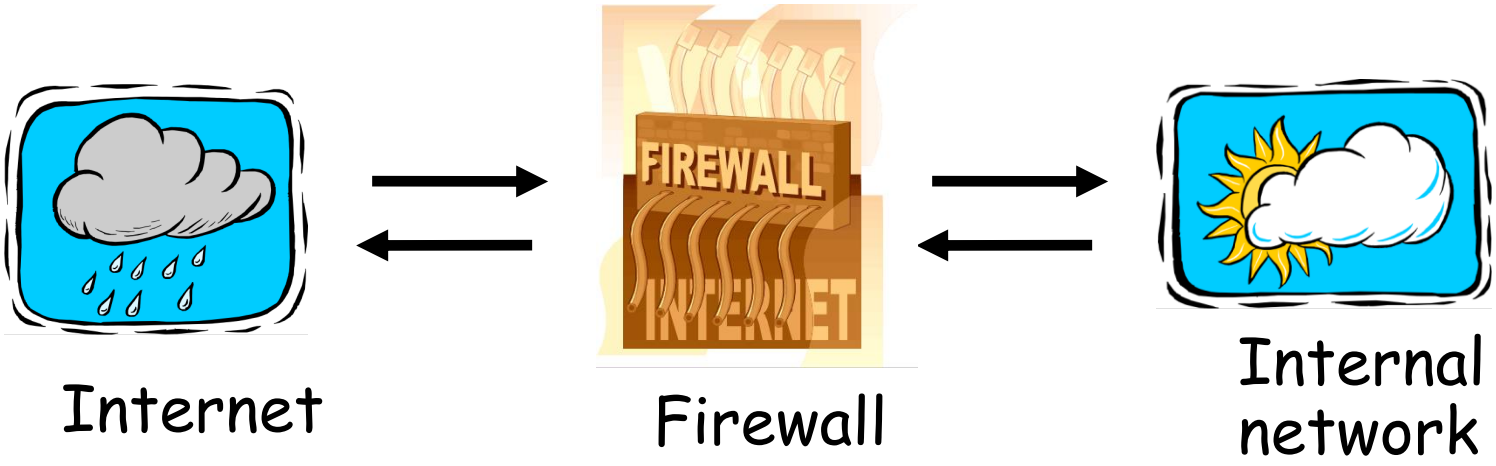
- Current types of CAPTCHAs
  - Visual — like previous example
  - Audio — distorted words or music
- No text-based CAPTCHAs
  - Maybe this is impossible…

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

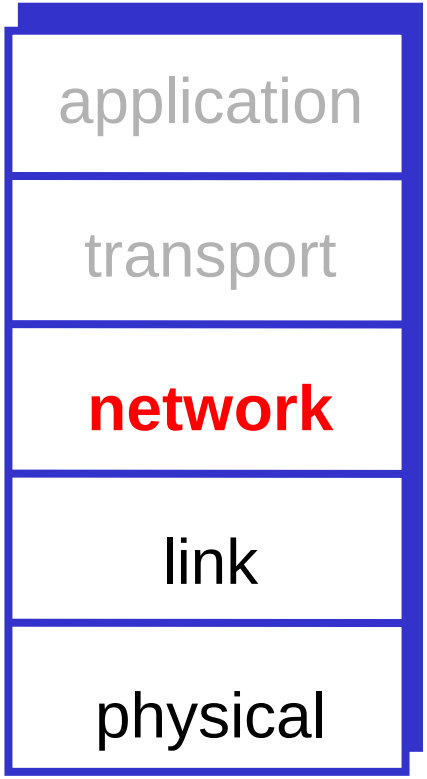
- A firewall is like a **secretary**
- To meet with an executive
  - First contact the secretary
  - Secretary decides if meeting is important
  - So, secretary filters out many requests
- You want to meet chair of CS department?
  - Secretary does some filtering
- You want to meet POTUS?
  - Secretary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

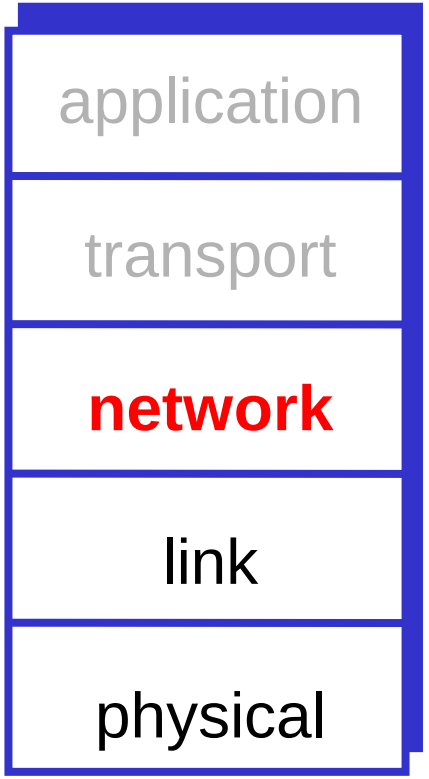
Packet Filter

- 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

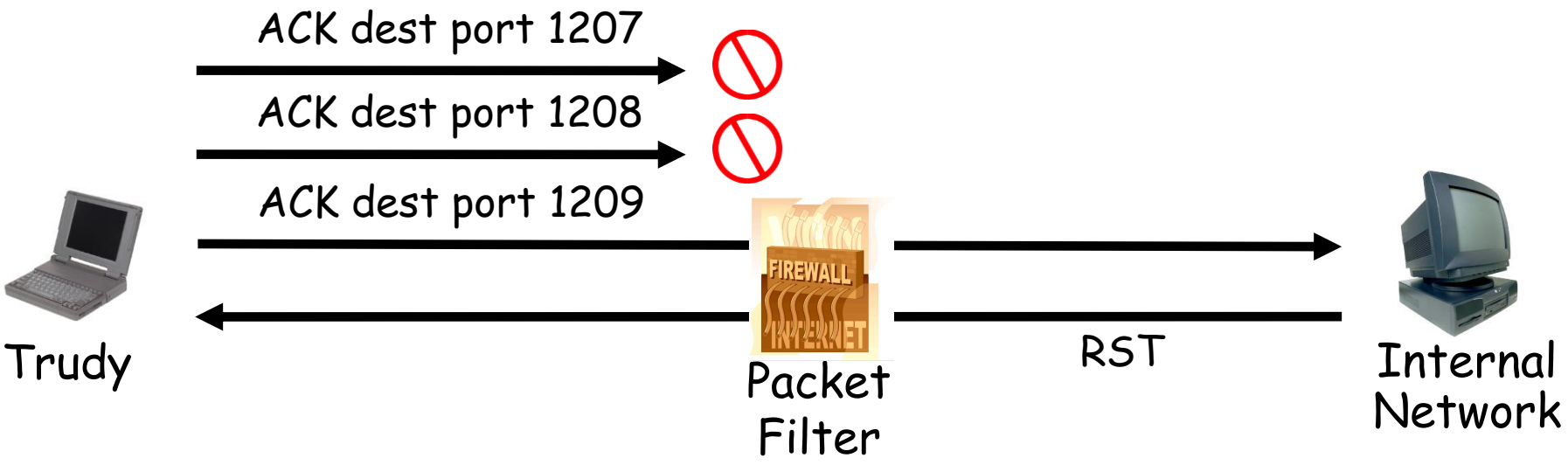
Action	Source IP	Dest IP	Source Port	Dest Port	Protocol	Flag Bits
Allow	Inside	Outside	Any	80	HTTP	Any
Allow	Outside	Inside	80	> 1023	HTTP	ACK
Deny	All	All	All	All	All	All

- **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

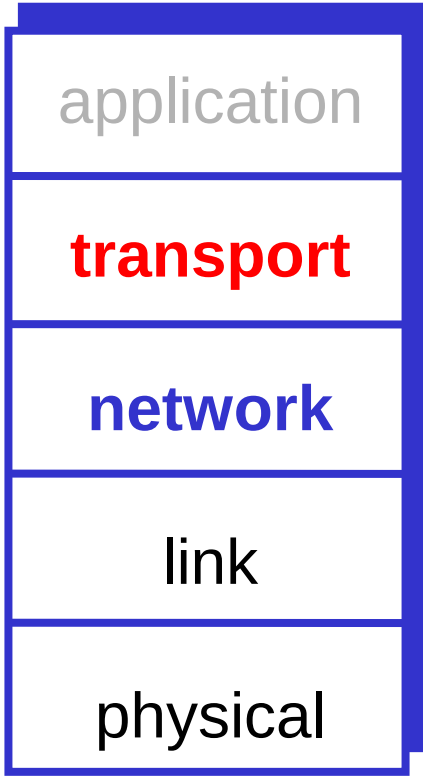
TCP ACK Scan



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

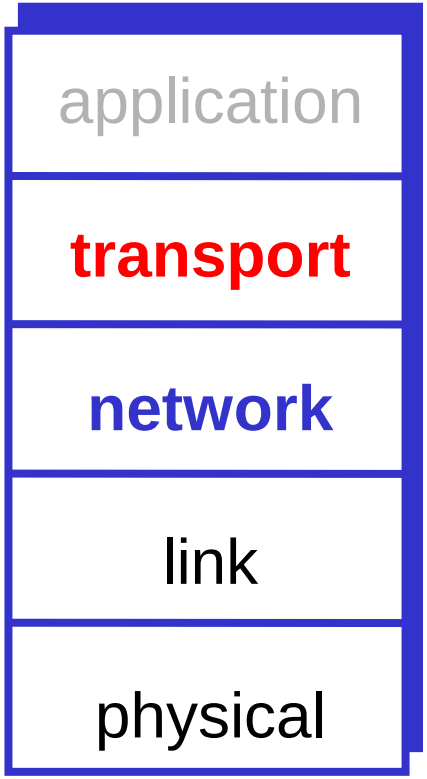
Stateful Packet Filter

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



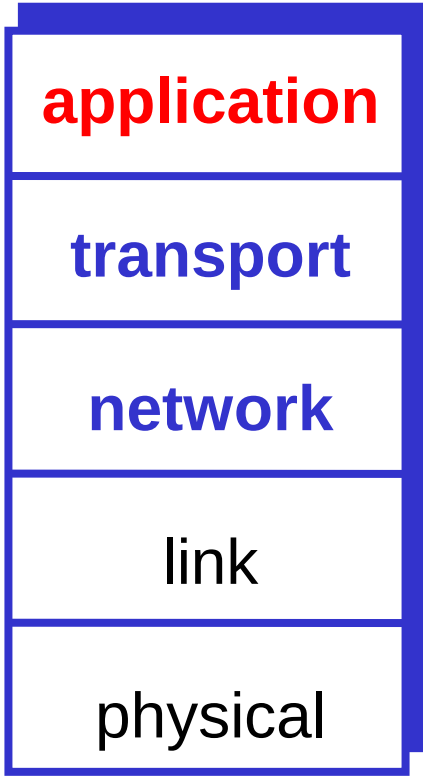
Stateful Packet Filter

- 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



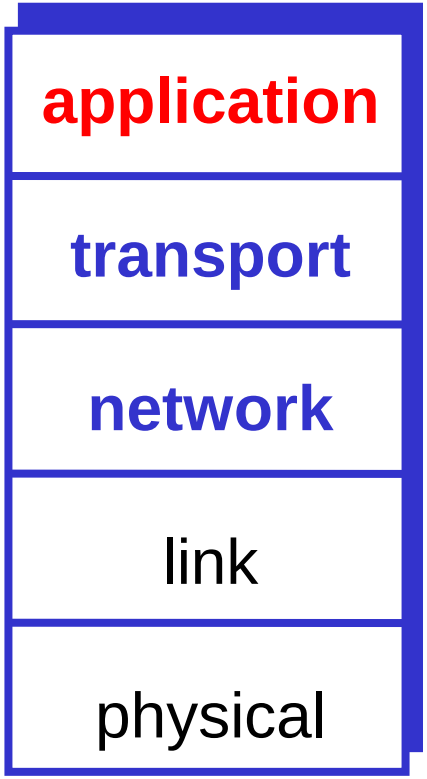
Application Proxy

- 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



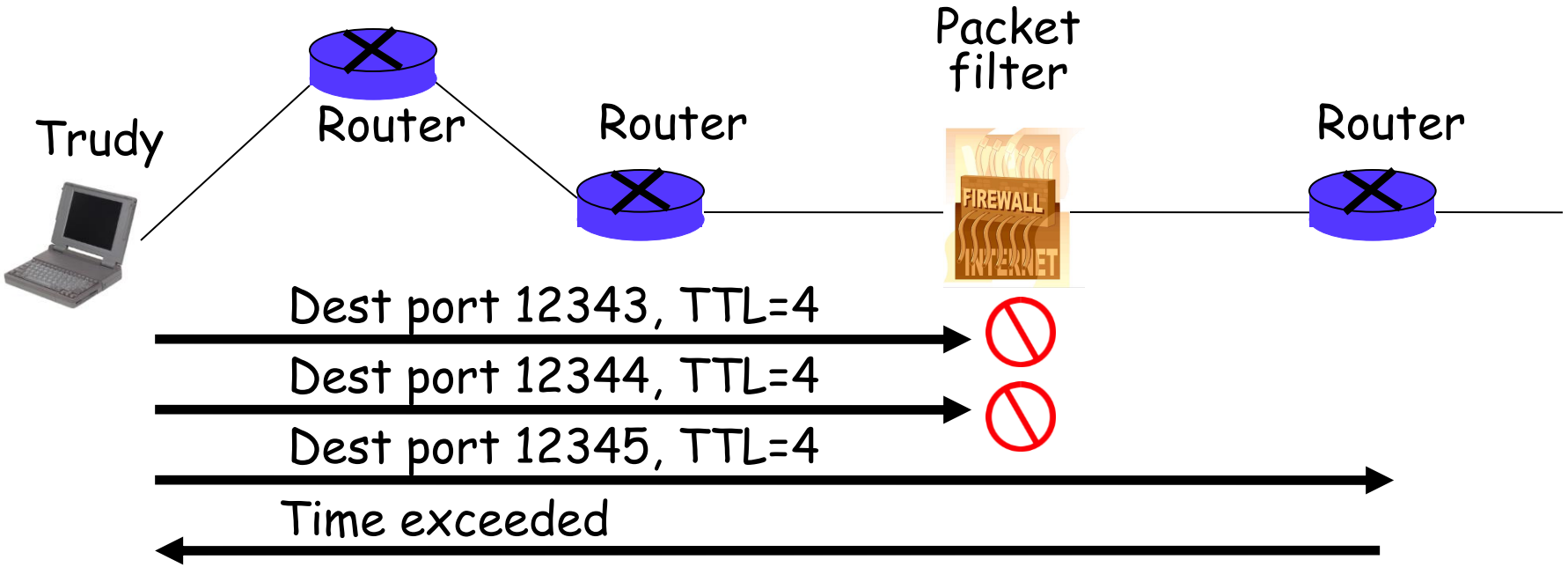
Application Proxy

- 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



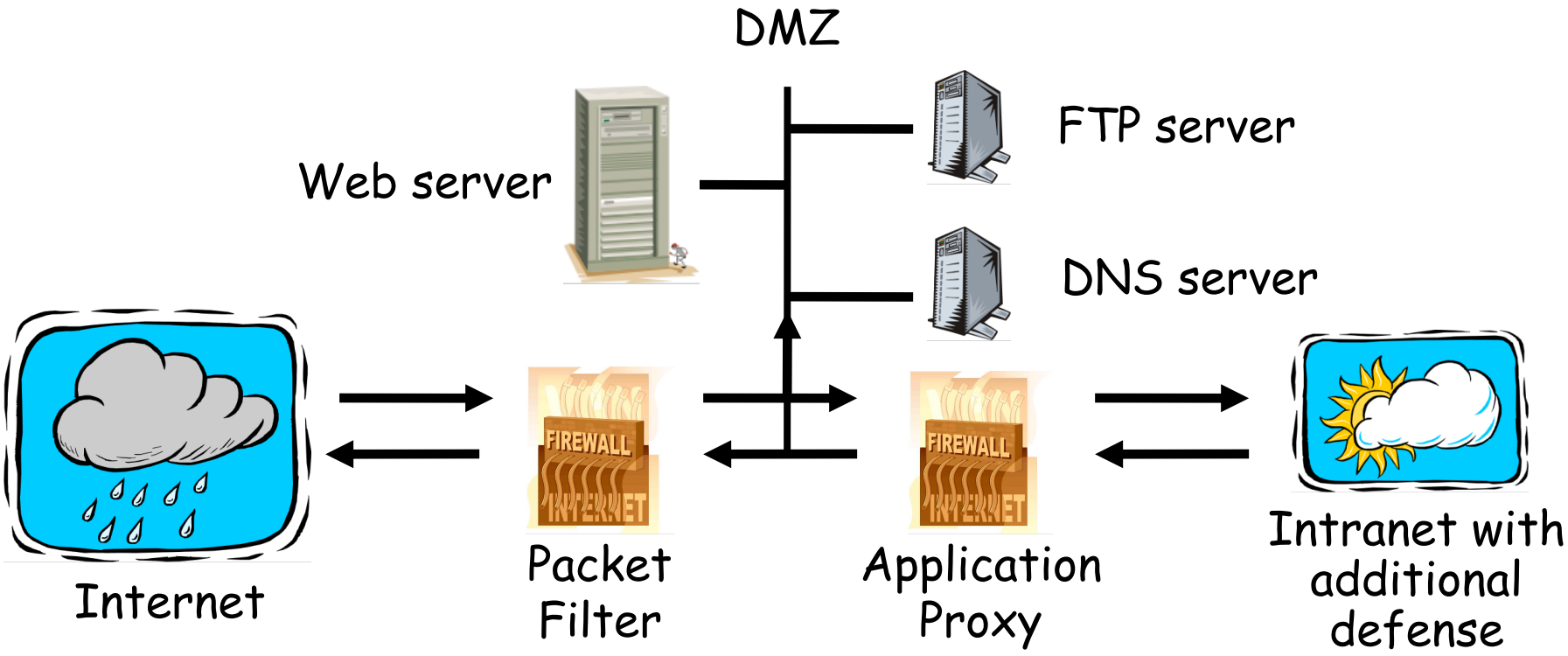
- This will **not** work thru an application proxy (why?)
- The proxy creates a new packet, destroys old TTL

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