CS 492
Computer Security

Authorization

Dr. Williams
Central Connecticut State University

Authentication vs Authorization

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

Lampson's Access Control Matrix

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

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

Are You Allowed to Do That?

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

Access Control Lists (ACLs)

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

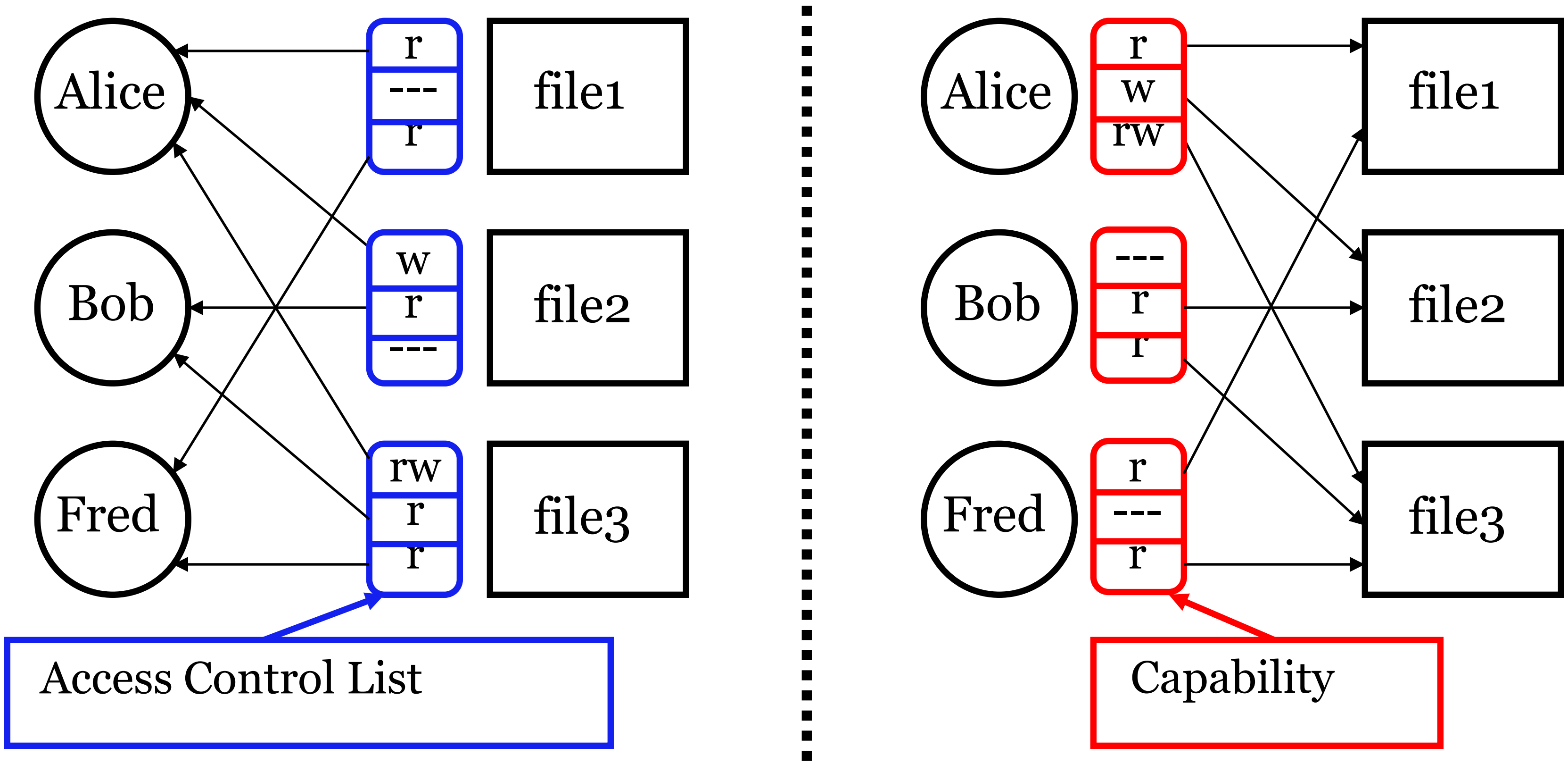
	OS	Accounting program	Accounting data	Insurance data	Payroll data
Bob	rx	rx	r	---	---
Alice	rx	rx	r	rw	rw
Sam	rwx	rwx	r	rw	rw
Accounting program	rx	rx	rw	rw	rw

Capabilities (or C-Lists)

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

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

ACLs vs Capabilities



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

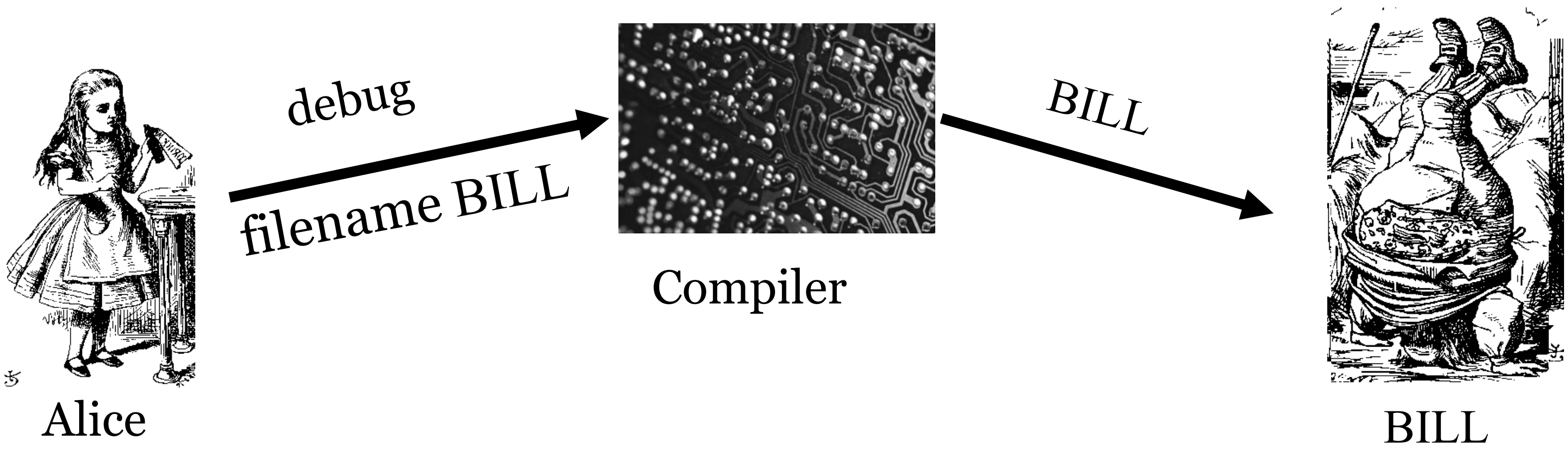
Confused Deputy

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

- Access control matrix

	Compiler	BILL
Alice	x	---
Compiler	rx	rw

ACL’s and Confused Deputy



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

Confused Deputy

- Compiler acting for Alice is confused
- There has been a separation of **authority** from the **purpose** for which it is used
- With ACLs, difficult to avoid this problem
- With Capabilities, easier to prevent problem
  - Must maintain association between authority and intended purpose
  - Capabilities also 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 use/on **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 (but, 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 a viable technical system
  - More of a legal device than technical system

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

BLP: The Bottom Line

- Criticism of BLP is "so trivial that it is hard to imagine a realistic security model for which it does not hold"
- 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
  - Suppose you trust the integrity of **O1** but not **O2**
  - If object **O3** includes **O1** and **O2** then you cannot trust the integrity of **O3**
- 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)

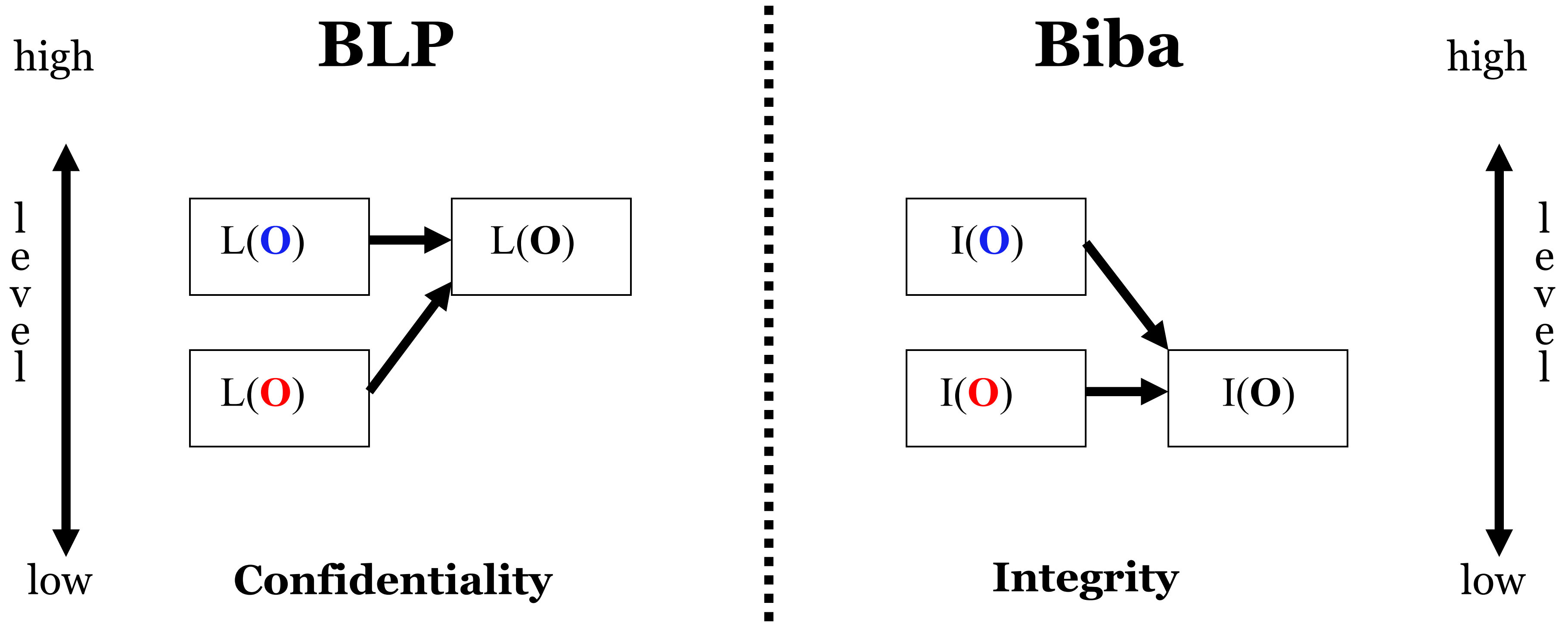
Biba's Model: S can read O if and only if $I(S) \leq I(O)$

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

What can we say?

- Using BLP, if S can read O1 and writes O2 what can we say about each of them?

- Using Biba, If S reads O1 and writes O2 what can we say about each of them?

Compartments

Compartments

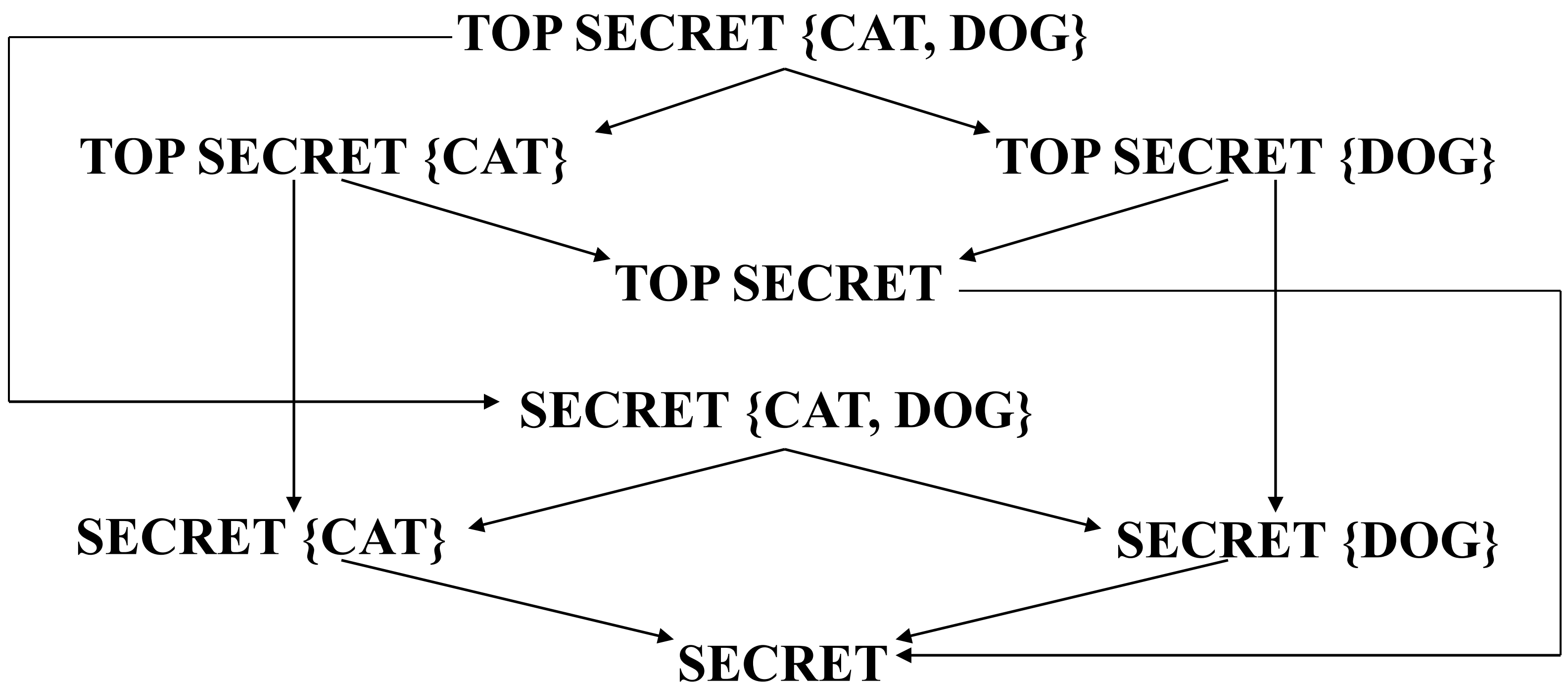
- Multilevel Security (MLS) enforces access control **up and down**
- Simple hierarchy of security labels may not be flexible enough
- Compartments enforces restrictions **across**
- Suppose **TOP SECRET** divided into **TOP SECRET {CAT}** and **TOP SECRET {DOG}**
- Both are **TOP SECRET** but information 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 your clearance, you only have access to info that you need to know

Compartments

- Arrows indicate "≥" relationship



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

TOP SECRET {CAT} VS SECRET {CAT, DOG}

MLS vs Compartments

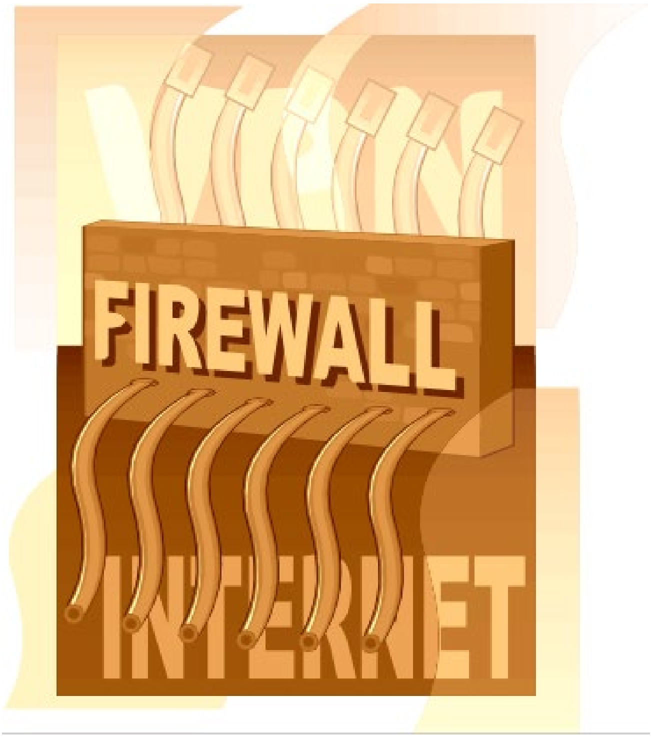
- MLS can be used without compartments
  - And vice-versa
- But, MLS almost always uses compartments

What can we say and why, draw the diagram of the information given (compartments)

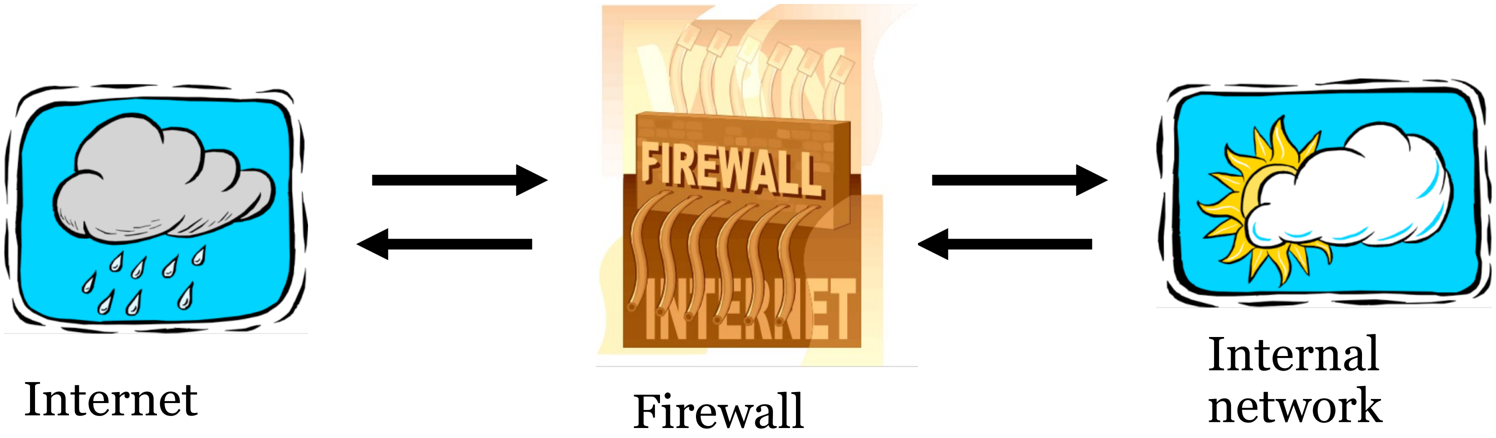
- Alice has clearance SECRET{CS 492}
  - TOPSECRET{CS 492}(O1)
  - SECRET{Bob}(O)
  - SECRET(O)
  - UNCLASSIFIED (O)

- If Bob is the overseer of all things in his realm but can only see SECRET things in Alice's realm, what would be his clearances?

Firewalls



Firewalls



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

Firewall as Secretary

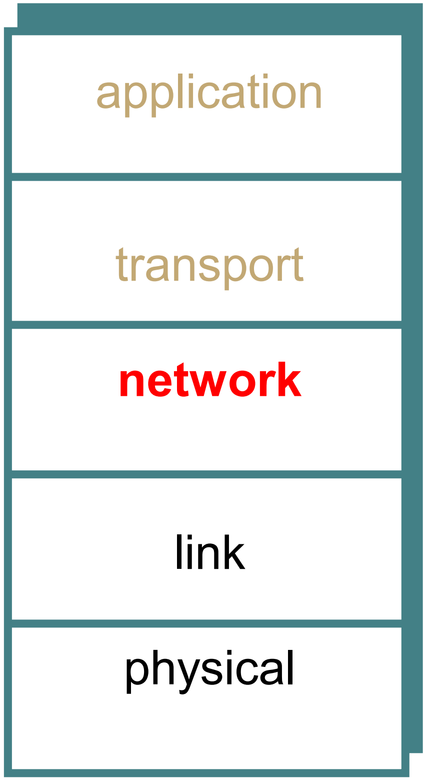
- A firewall is like a **secretary**
- To meet with an executive
  - First contact the secretary
  - Secretary decides if meeting is important
  - So, secretary filters out many requests
- You want to meet chair of CS department?
  - Secretary does some filtering
- You want to meet the POTUS?
  - Secretary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

Packet Filter

- Operates at network layer
- Can filter based on…
  - Source IP address
  - Destination IP address
  - Source Port
  - Destination Port
  - Flag bits (SYN, ACK, etc.)
  - Egress or ingress



Packet Filter

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

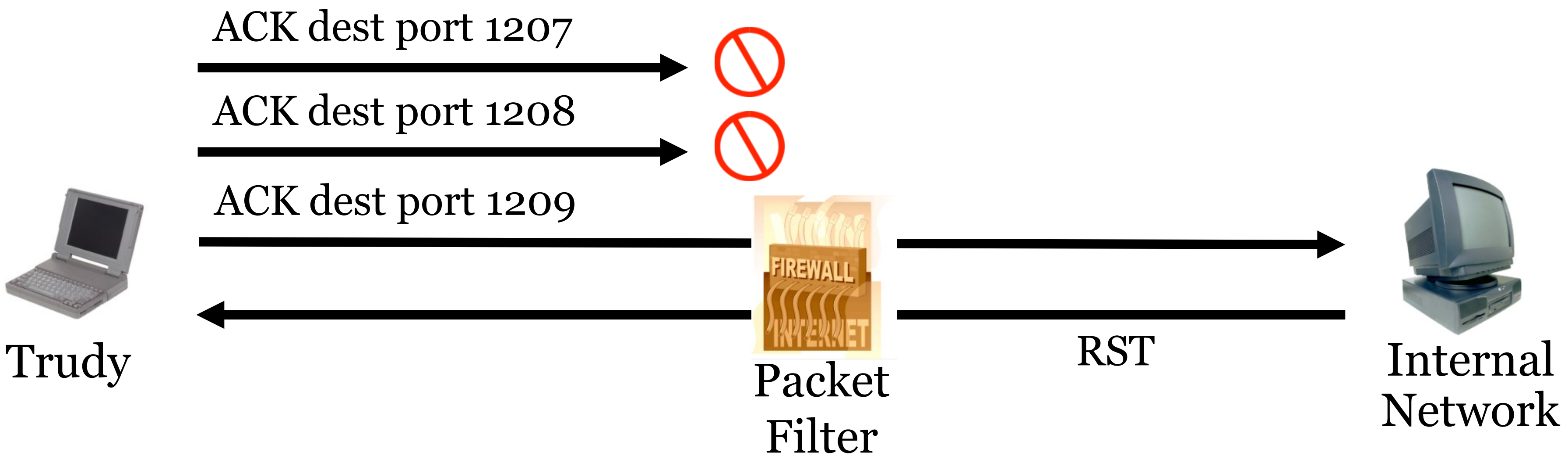
Action	Source IP	Dest IP	Source Port	Dest Port	Protocol	Flag Bits
Allow	Inside	Outside	Any	80	HTTP	Any
Allow	Outside	Inside	80	> 1023	HTTP	ACK
Deny	All	All	All	All	All	All

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

TCP ACK Scan

- Attacker scans for open ports thru firewall
  - Port scanning is *first step* in many attacks (nmap)
- 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

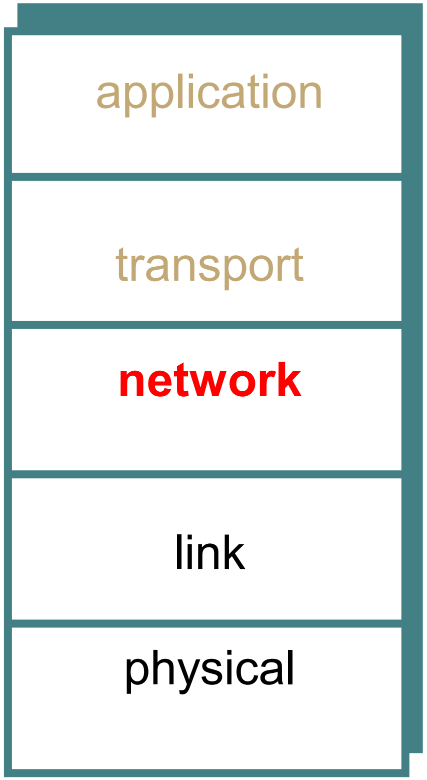
TCP ACK Scan



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

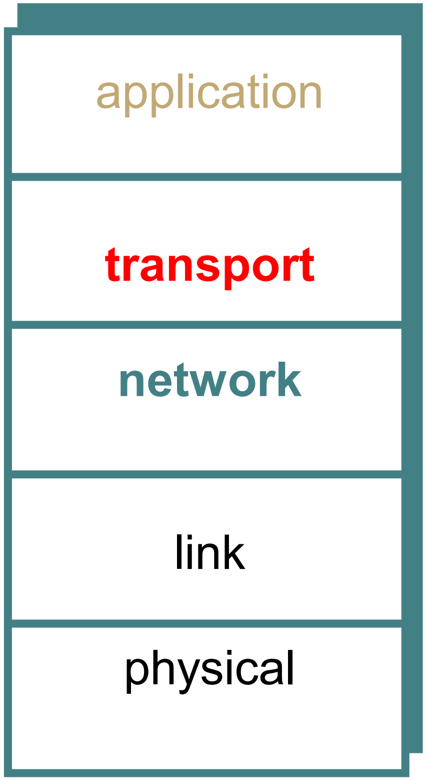
Packet Filter

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



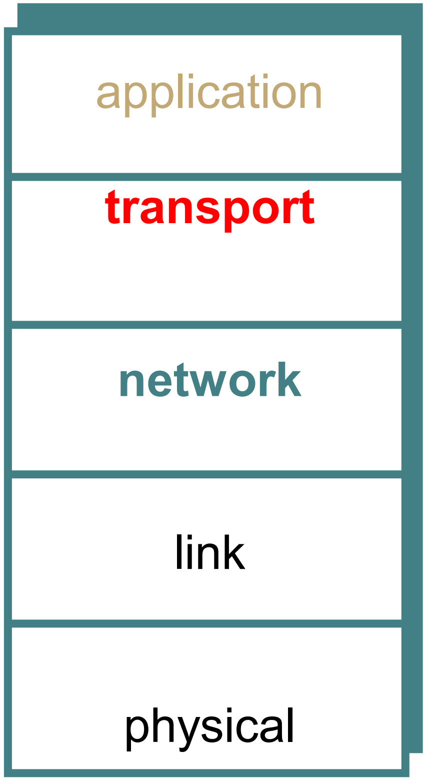
Stateful Packet Filter

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



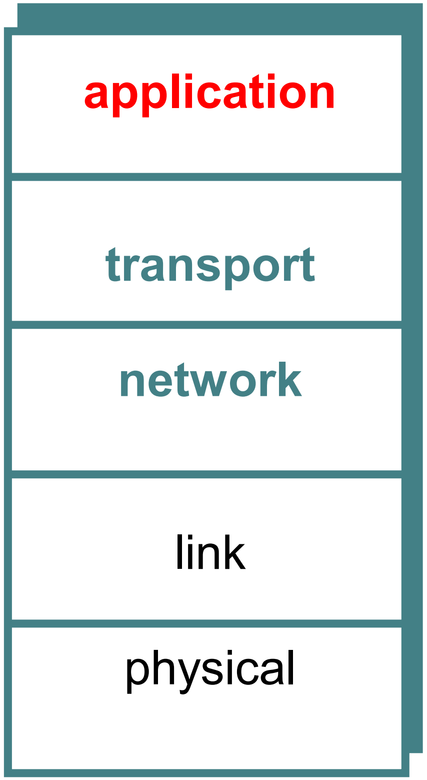
Stateful Packet Filter

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



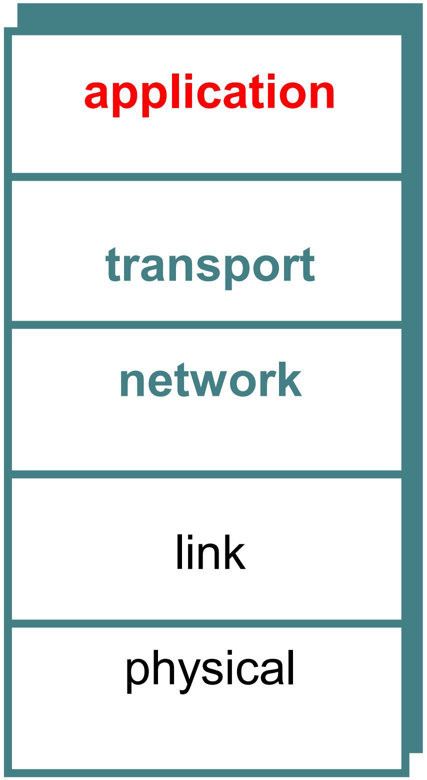
Application Proxy

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



Application Proxy

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



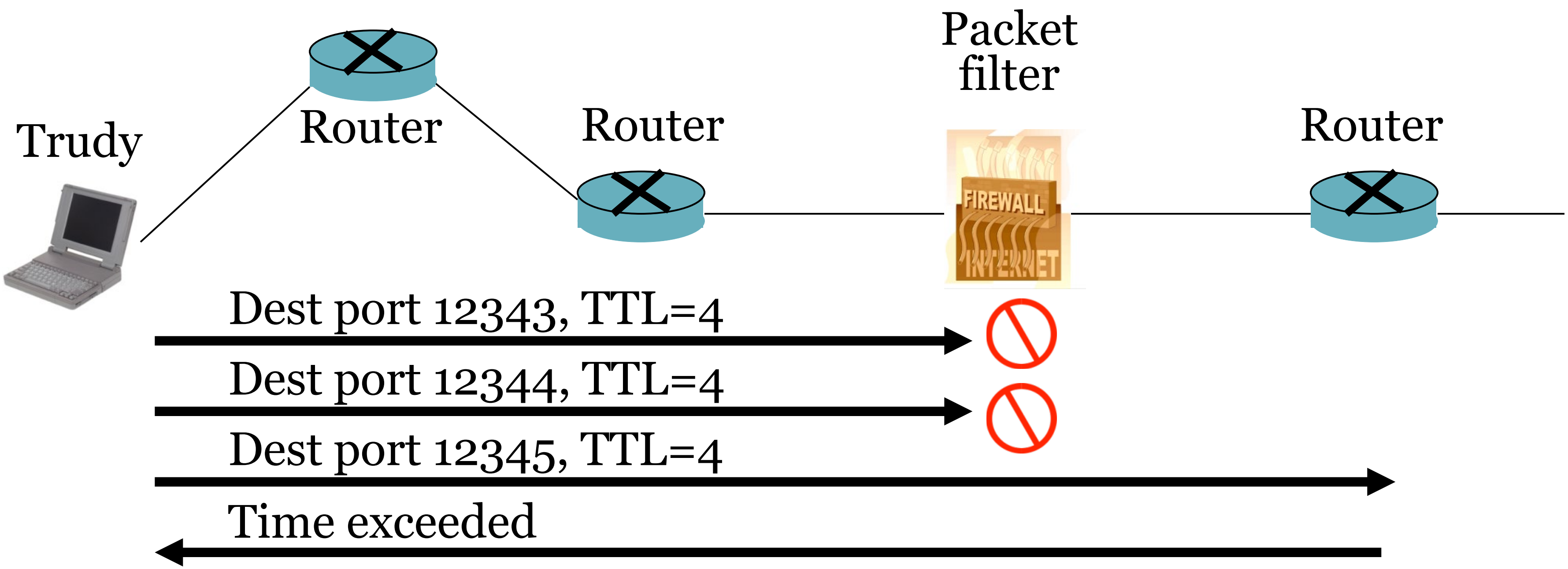
Application Proxy

- 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
- Prevents 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



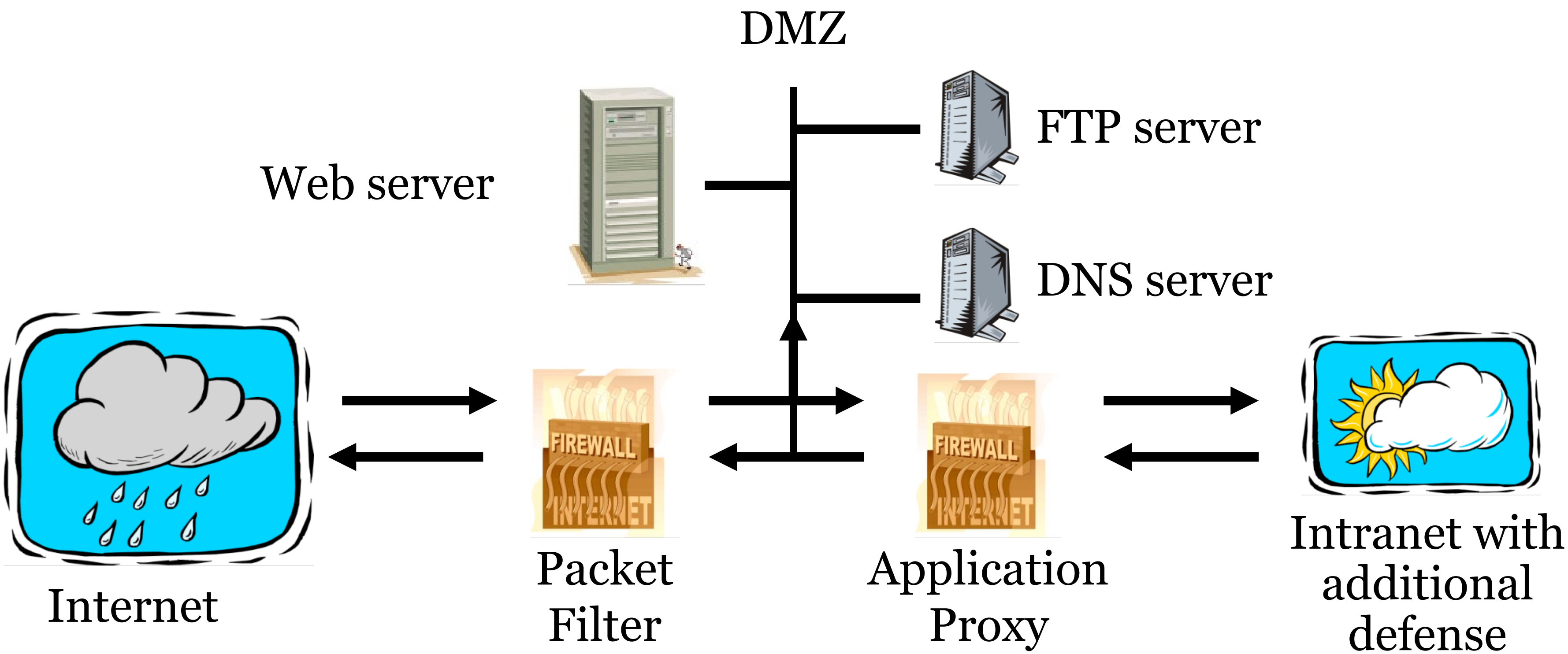
- **This will not work thru an application proxy (why?)**
- The proxy creates a new packet, destroys old TTL

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
  - Effect like 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**)

  What is it?
  - 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?**

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**
  - **How?**
  - 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
- **What would the model/signature of an attack look like?**
- 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?

Covert Channel

Covert Channel

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

Covert Channel Example

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

Covert Channel Example

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

Time:

Covert Channel

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

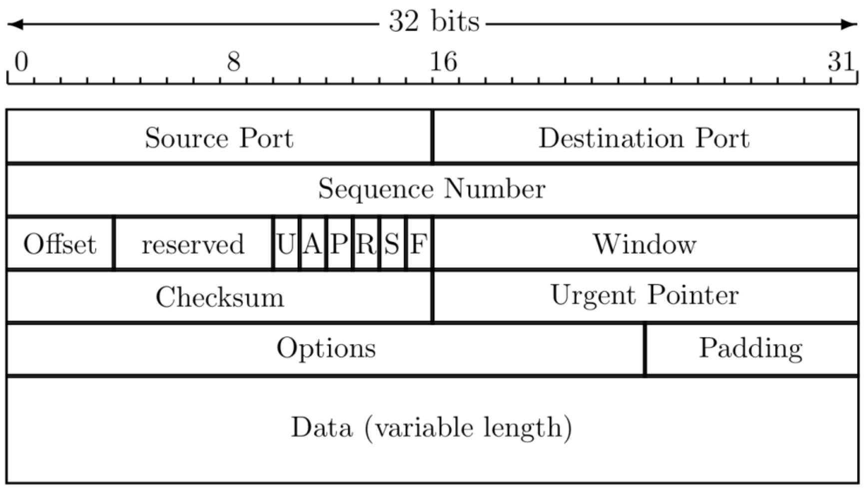
Covert Channel

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

Covert Channel

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

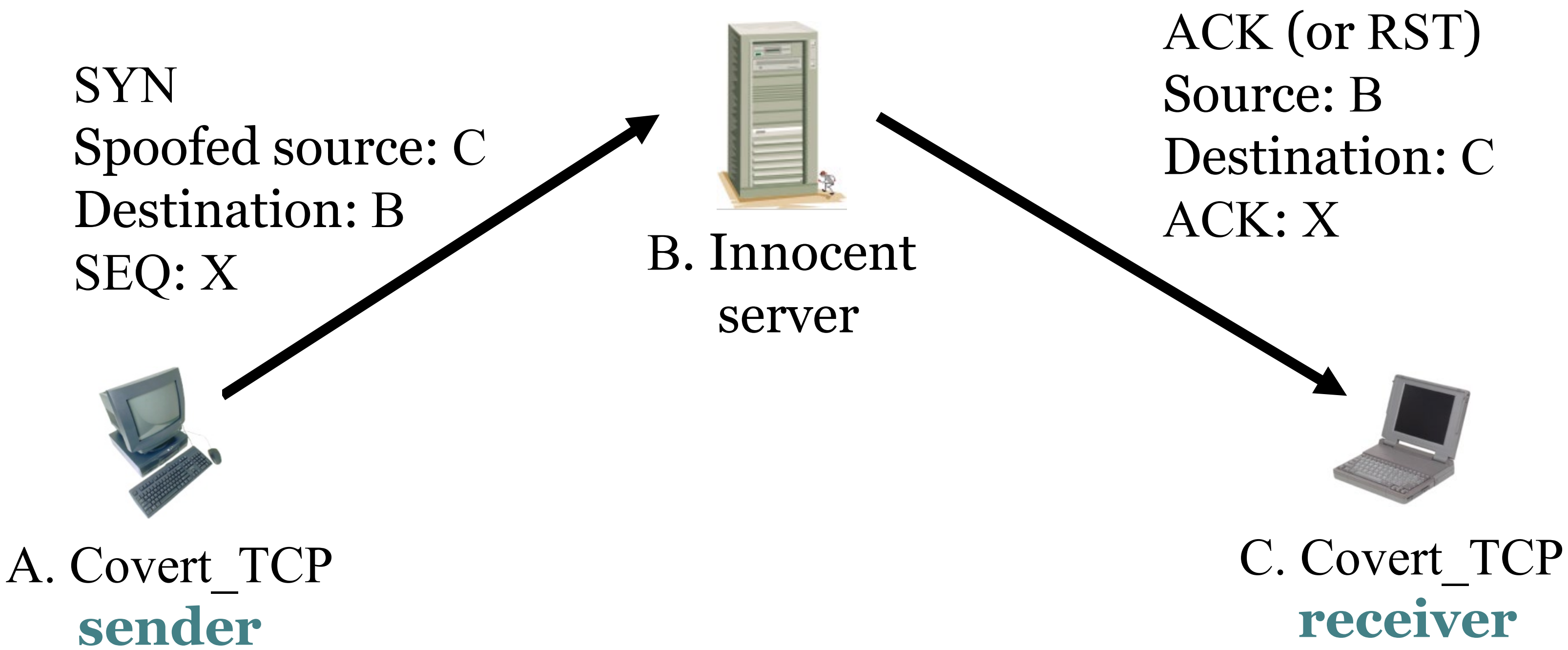
Real-World Covert Channel



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

Real-World Covert Channel

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



Inference Control

Inference Control Example

- Suppose we query a database
  - Question: What is average salary of female CS professors at SJSU?
  - Answer: $95,000
  - Question: How many female CS professors at SJSU?
  - Answer: 1
- Specific information has leaked from responses to general questions!

Inference Control and Research

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

Naïve Inference Control

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

Less-naïve Inference Control

- Query set size control
  - Don't return an answer if set size is too small
- N-respondent, k% dominance rule
  - Do not release statistic if k% or more contributed by N or fewer
  - Example: Avg salary in Bill Gates' neighborhood
  - This approach used by US Census Bureau
- Randomization
  - Add small amount of random noise to data
- Many other methods — none satisfactory

Inference Control

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

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
  - Firewalls
  - IDS
  - Covert channel
  - Inference control