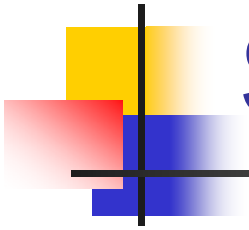
IS 2150 / TEL 2810
Introduction to Security

James Joshi
Associate Professor, SIS
Presented by
Nathalie Baracaldo

Lecture 3
Sept 18, 2013

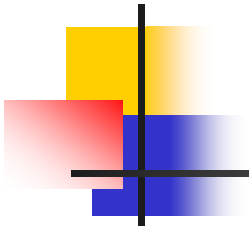
Intrusion Detection,
Firewalls & VPN
Auditing System



Some announcements

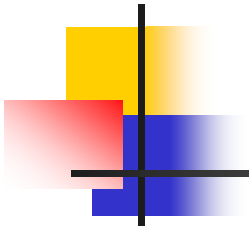
- Quiz next week
- To submit assignment 1, please print it and bring it to LERSAIS lab room 410 on the specified deadline (Sep. 20).
  - You can slide it under the door if there is nobody around or leave it on my desk.
- You should have access to Course Web!

Intrusion Detection



Intrusion Detection/Response

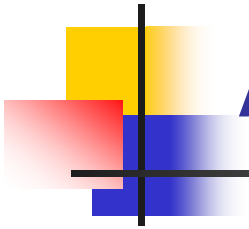
- Denning:
  - Systems under attack fail to meet one or more of the following characteristics
    1. Actions of users/processes conform to statistically **predictable patterns**
    2. Actions of users/processes **do not** include sequences of **commands to subvert security** policy
    3. Actions of processes **conform to specifications** describing allowable actions



Intrusion Detection

- Idea:
  - Attack can be discovered by one of the above being violated
- *Practical* goals of intrusion detection systems:
  - Detect a wide variety of intrusions (**known + unknown**)
  - Detect in a **timely** fashion
  - Present analysis in a useful manner
    - Need to monitor many components; proper interfaces needed
  - Be (sufficiently) accurate
    - Minimize *false positives* and *false negatives*

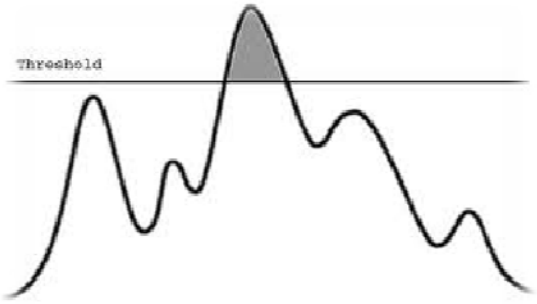
    - *False positive:* conclude there is an attack when there isn't
    - *False negative:* conclude there isn't an attack when there is one



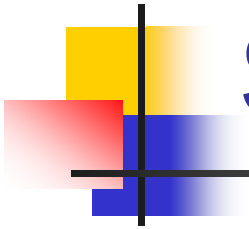
IDS Types: Anomaly Detection

- Compare system characteristics with expected values
  - Threshold metric
  - Statistical moments
  - Markov model
- All these require the establishment of *indicators*

Threshold metrics



- A minimum of ***m*** and a maximum of ***n*** events are expected to occur (for some event and some values m and n).
  - If, over a specific period of time, fewer than **m** or more than n events occur, the behavior is deemed anomalous.
  - E.g., Number of failed logins
- **Any challenge?**
  - How do you set a suitable threshold?

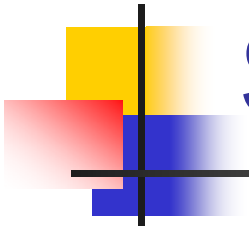


Statistical metrics

- Consider:
  - Mean/standard deviation/correlations
- Possible indicators
  - Number of user events in a system
  - Time periods of user activity
  - Resource usages profiles
- If the behavior is outside the expected measurements, it is flag as anomalous
- Any challenges here?

Markov Models

- Based on state, expected likelihood of transition to new states
  - If a low probability event occurs, then it is considered suspicious
- Any challenges?
- Other models used are neural-networks, petri nets, etc...

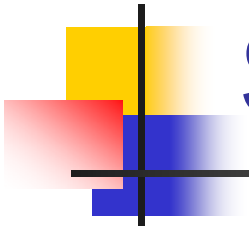


So which is better?

- Any particular advantage?
  - Threshold metric
  - Statistical moments
  - Markov model

IDS Types: Misuse Modeling

- Does sequence of instructions violate security policy?
  - Problem: How do we know all violating sequences?
- Solution: capture *known* violating sequences
  - Generate a **rule set** for an intrusion signature
- Alternate solution: State-transition approach
  - Known "bad" state transition from attack
  - Capture when transition has occurred (user → root)



Specification Modeling

- Does sequence of instructions violate system specification?
  - What is the system specification?
- Need to formally specify operations of potentially critical code
  - *trusted* code
- Verify post-conditions met

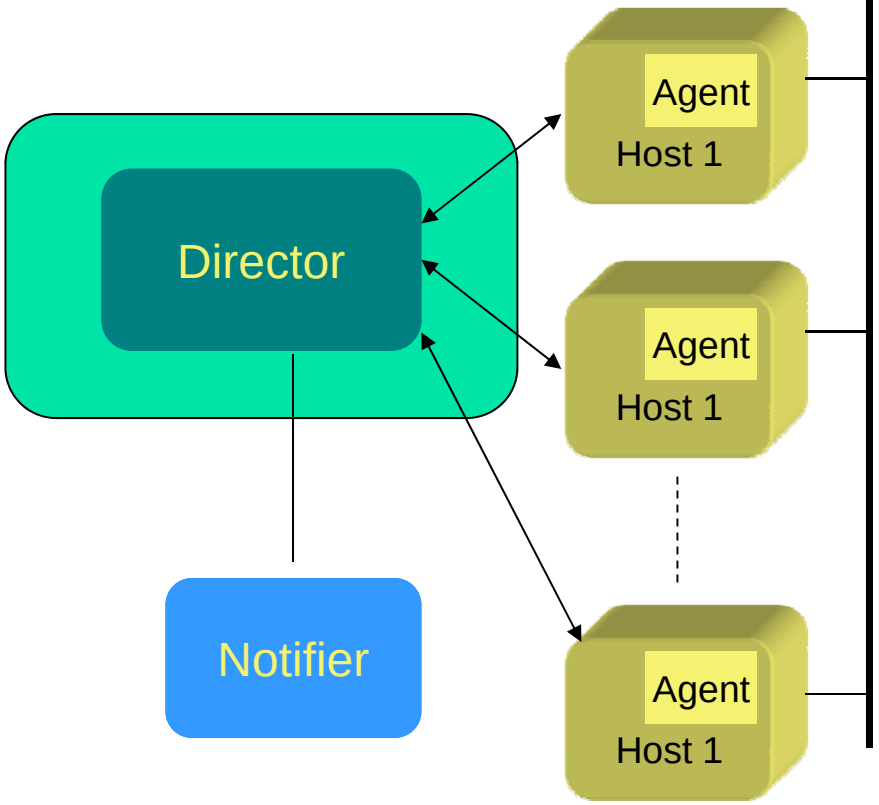
IDS Systems

- Anomaly Detection
  - Intrusion Detection Expert System (IDES) – successor is NIDES
  - Network Security MonitorNSM
- Misuse Detection
  - Intrusion Detection In Our Time- IDIOT (colored Petri-nets)
  - USTAT?
  - ASAX (Rule-based)
- Hybrid
  - NADIR (Los Alamos)
  - Haystack (Air force, adaptive)
  - Hyperview (uses neural network)
  - Distributed IDS (Haystack + NSM)

IDS Architecture

- Similar to Audit system
  - Log events
  - Analyze log
- Difference:
  - happens real-time - *timely* fashion
- (Distributed) IDS idea:
  - Agent generates log
  - Director analyzes logs
    - May be adaptive
  - Notifier decides how to handle result
    - GrIDS displays attacks in progress

Possible architecture of a distributed IDS

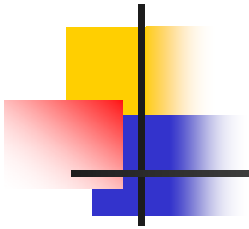


Where is the Agent?

- Host based IDS
  - Watches events on the host
  - Often uses existing audit logs
- Network-based IDS
  - Packet sniffing
  - Firewall logs

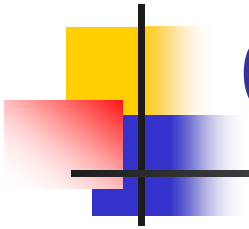
IDS Problem

- IDS useless unless accurate
  - Significant fraction of intrusions detected
  - Significant number of alarms correspond to intrusions
- Goal is
  - Reduce false positives
    - Reports an attack, but no attack underway
  - Reduce false negatives
    - An attack occurs but IDS fails to report
- Great if this alarm reduction is automatic!



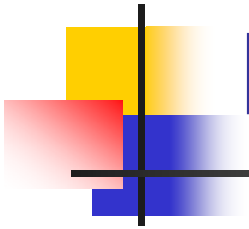
Intrusion Response

- Incident Prevention
  - Stop attack before it succeeds
  - Measures to detect attacker →a.k.a. indicators
  - Example: Jailing (also Honeypots)
- Intrusion handling
  - Preparation for detecting attacks
  - Identification of an attack
  - Contain attack
  - Eradicate attack
  - Recover to secure state
  - Follow-up to the attack - Punish attacker???



Containment

- Passive monitoring
  - Track intruder actions
  - Eases recovery and punishment
- Constraining access
  - Downgrade attacker privileges
  - Protect sensitive information
  - Why not just pull the plug?



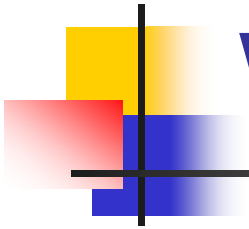
Eradication

- Terminate network connection
- Terminate processes
- Block future attacks
  - Close ports
  - Disallow specific IP addresses
  - Wrappers around attacked applications

Follow-Up

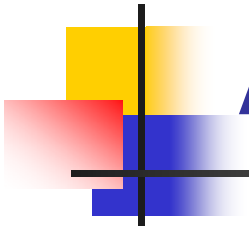
- Legal action
  - Trace through network
- Informing public?
- Cut off resources
  - Notify ISP of action
- Counterattack
  - Is this a good idea?

Auditing



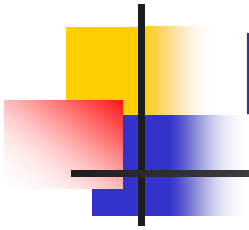
What is Auditing?

- Goals/uses
  - User accountability
  - Damage assessment
  - Determine causes of security violations
  - Describe security state for monitoring critical problems
  - Evaluate effectiveness of protection mechanisms
- Auditing systems
  - Logging
  - Audit analysis
- Key issues
  - What to log?
    - What about everything?
  - What do you audit?



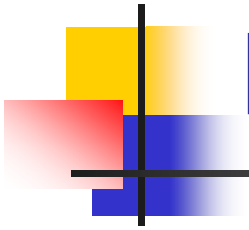
Audit System Structure

- Logger
  - Records information, usually controlled by parameters
- Analyzer
  - Logs may come from multiple systems, or a single system
  - May lead to changes in logging
  - May lead to a report of an event
- Notifier
  - Informs analyst, other entities of results of analysis
  - May reconfigure logging and/or analysis on basis of results
  - May take some action



Example: Windows NT

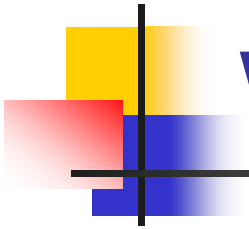
- Different logs for different types of events
  - *System event* logs record system crashes, component failures, and other system events
  - *Application event* logs record events that applications request be recorded
  - *Security event* log records security-critical events such as logging in and out, system file accesses, and other events
- Logs are binary; use *event viewer* to see them
- If log full, can have system shut down, logging disabled, or logs overwritten
- The size of the log is an important aspect!



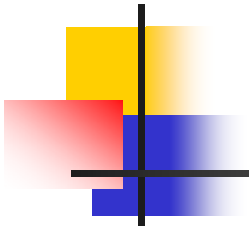
Designing an Audit System

- Goals determine what is logged
  - Idea: auditors want to detect violations of policy, which provides a set of constraints that the set of possible actions must satisfy
  - So, audit functions that may violate the constraints
- There is a policy that tells you:
  - Constraint p_i : *action* $\Rightarrow$ *condition*

An example of how this model works

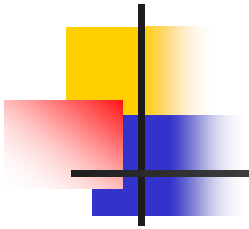


- Log this information:
  - P1: read file x→ person has enough clearance to read file x
- Things that need to be logged:
  - Clearance required to read file x
  - Clearance of the person that reads file x
- Is this enough?
  - According to this model it should be enough
  - But in reality you would also need
    - Name of user and the name of the file!



Implementation Issues

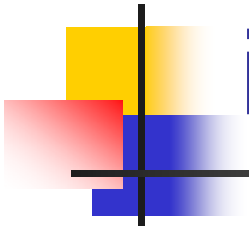
- Not all violations may be logged
- Defining violations
  - Does “write” include “append” and “create directory”?
- Multiple names for one object
  - Logging goes by *object* and not name
  - Representations can affect this
- Syntactic issues
  - Correct grammar – unambiguous semantics



Implementation Issues

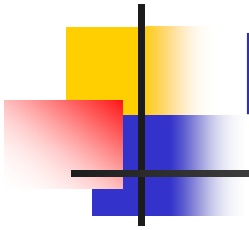
- The log shouldn’t be written or re-written by anyone in the system
  - Otherwise

Can logs leak private information?



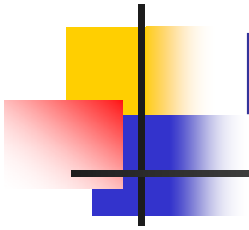
- Personal data of employees
  - Credit card numbers
  - Health related information
- Confidential data of an organization unit

- Solution: log sanitization



Example (1)

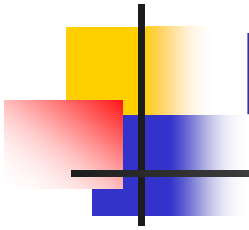
- The log may contain file names that give indications of proprietary projects or enable an industrial spy to determine the IP addresses of machines containing sensitive information
  - In this case, the unsanitized logs are available to the site administrators only



Example (2)

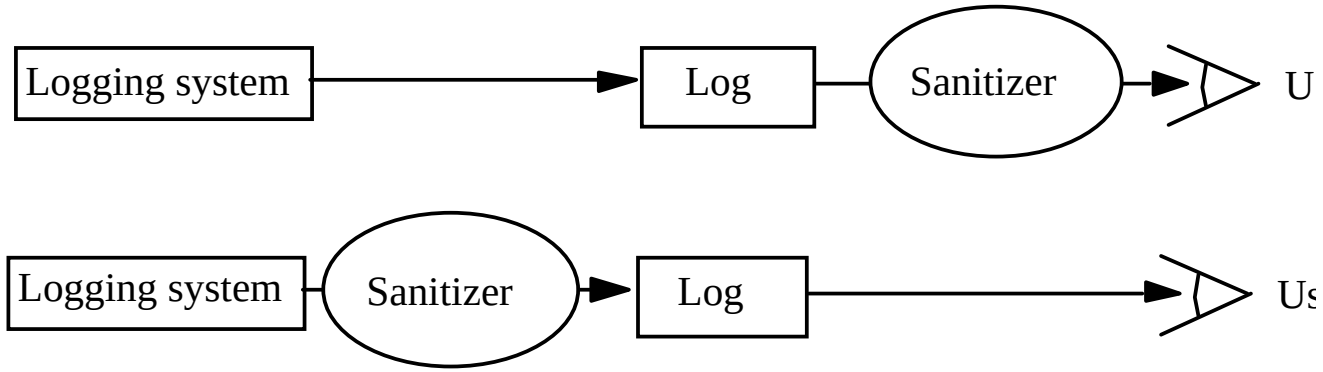
- The policy may forbid the information to leave the system
  - E.g., personal transactions of the users
- In this case, the goal is to prevent the system administration from spying on the users

Log Sanitization – the model!

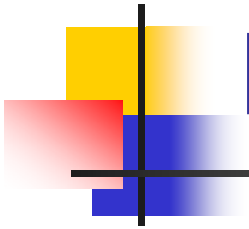


- U set of users
- P policy defining set of information $C(U)$ that members of U cannot see
- Log L is sanitized when all information in $C(U)$ deleted from L
- Two types of P
  - $C(U)$ can't leave site
    - People inside site are trusted and information not sensitive to them
  - $C(U)$ can't leave system
    - People inside site not trusted or (more commonly) information sensitive to them
    - Don't log this sensitive information

Logging Organization



- Top prevents information from leaving site
  - Users' privacy not protected from system administrators, but protected from user in *U*
- Bottom prevents information from leaving system
  - Data simply not recorded, or data scrambled before recording (Cryptography)
  - E.g., if a company uses a cloud computing. In this case, U would contain the administrators of the cloud, who have access to the log and who shouldn't see your data

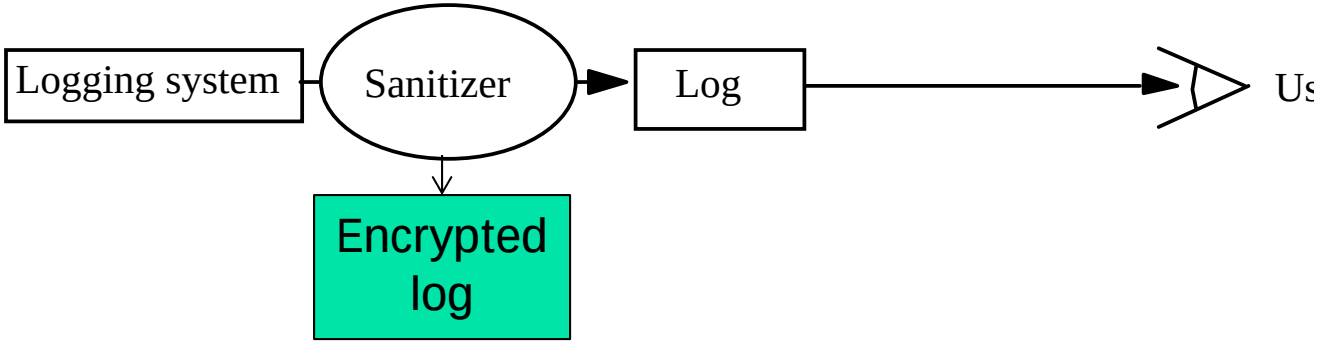


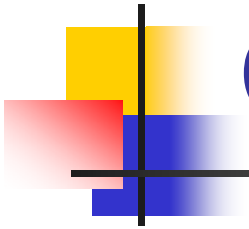
Reconstruction

- *Anonymizing sanitizer* cannot be undone
- *Pseudonymizing sanitizer* can be undone
- Importance
  - Suppose security analysis requires access to information that was sanitized?

Pseudonymizing sanitizer

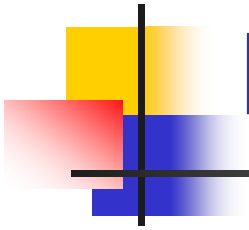
- The sanitizer may save information in a separate log that enables the reconstruction of the omitted information
- Cryptographic techniques enforce separation of privilege, so multiple administrators must agree to view the unsanitized logs





Other considerations

- Key: sanitization must preserve properties needed for security analysis
- If new properties added (because analysis changes), may have to resanitize information
  - This *requires* pseudonymous sanitization or the original log

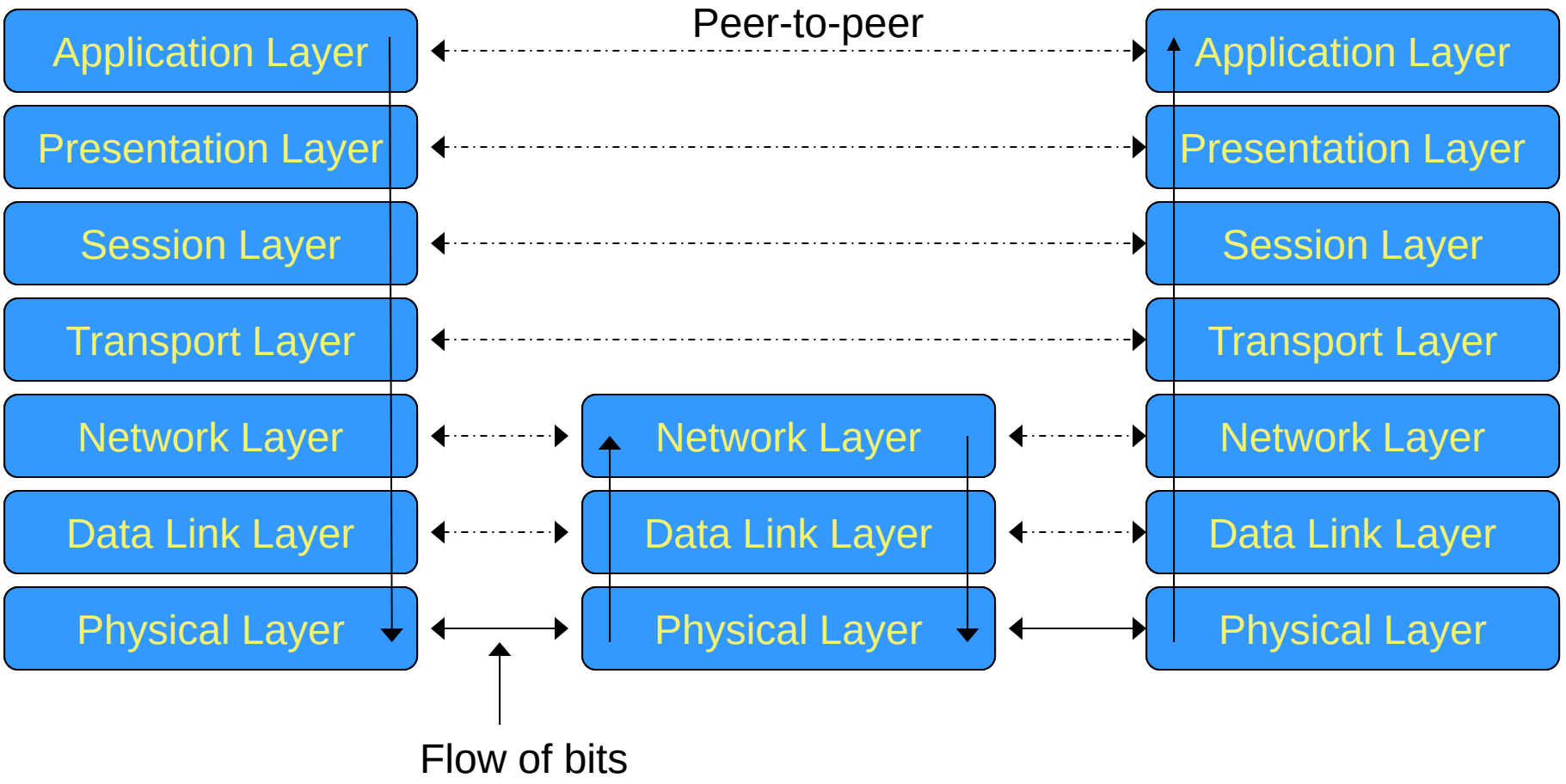


Example

- Company wants to keep its IP addresses secret, but wants a consultant to analyze logs for an address scanning attack
  - Connections to port 25 on IP addresses 10.163.5.10, 10.163.5.11, 10.163.5.12, 10.163.5.13, 10.163.5.14,
  - Sanitize with random IP addresses
    - Cannot see sweep through consecutive IP addresses
  - Sanitize with sequential IP addresses
    - Can see sweep through consecutive IP addresses

Firewalls & VPN

ISO/OSI Model

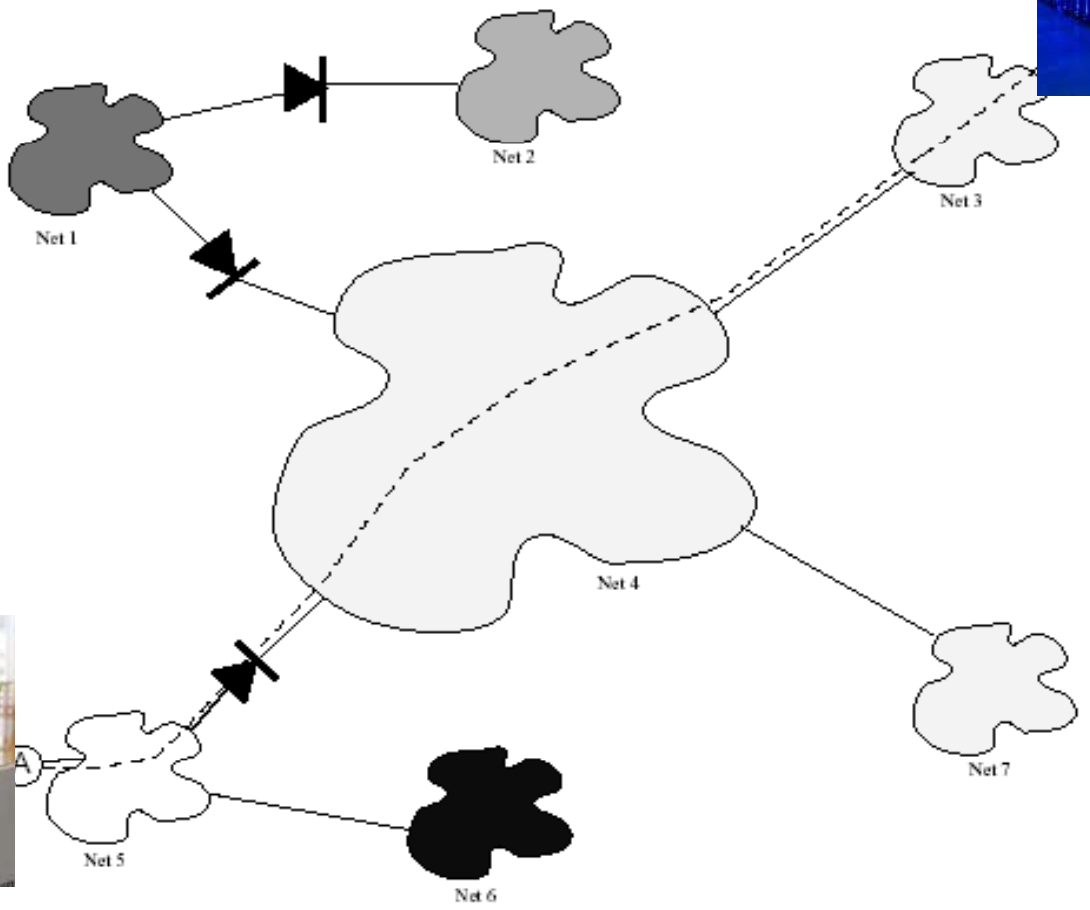


What is a VPN?

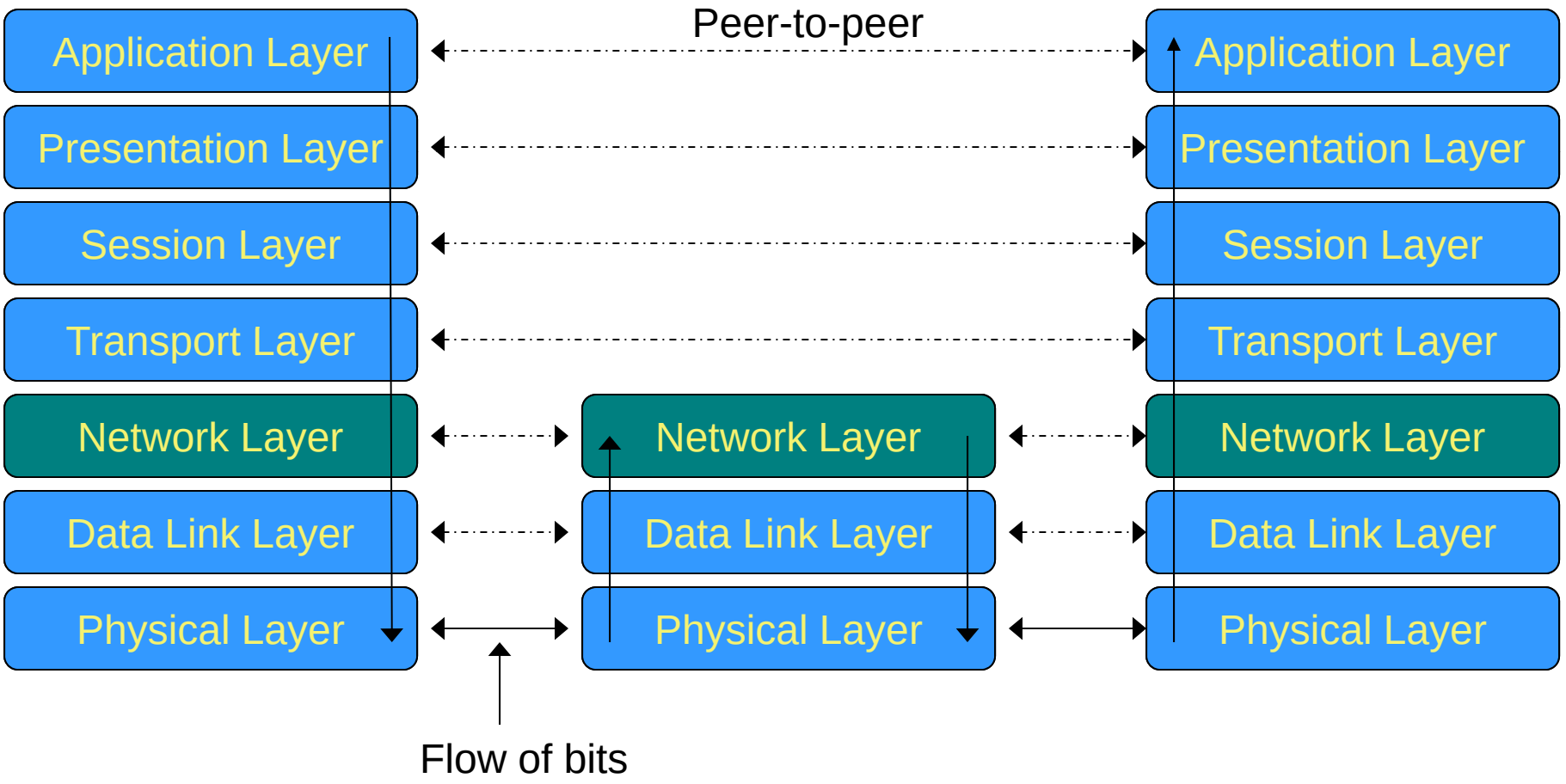
- **Virtual Private Network!**
- A network that supports a closed community of authorized users
  - Use the public Internet as part of the virtual private network
- There is traffic isolation
  - Contents, Services, Resources – secure
- Provide security!
  - Confidentiality and integrity of data
  - User authentication
  - Network access control

- **IPSec can be used**

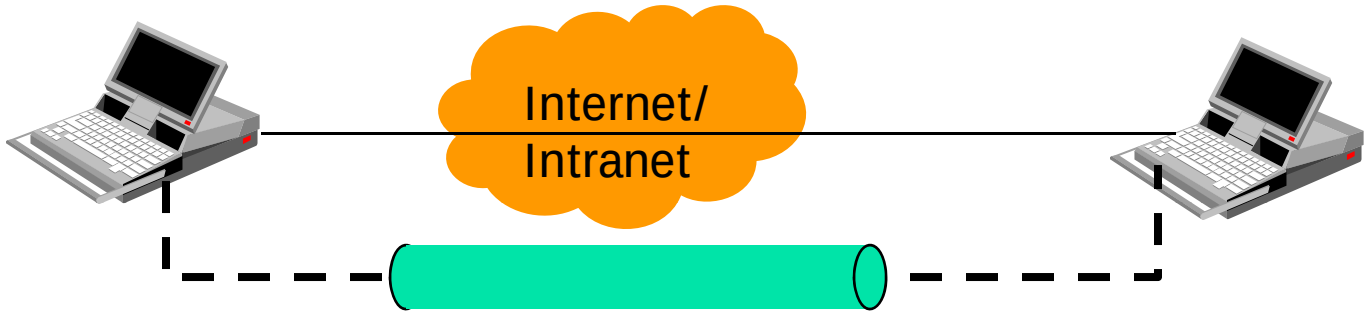
Tunneling in VPN



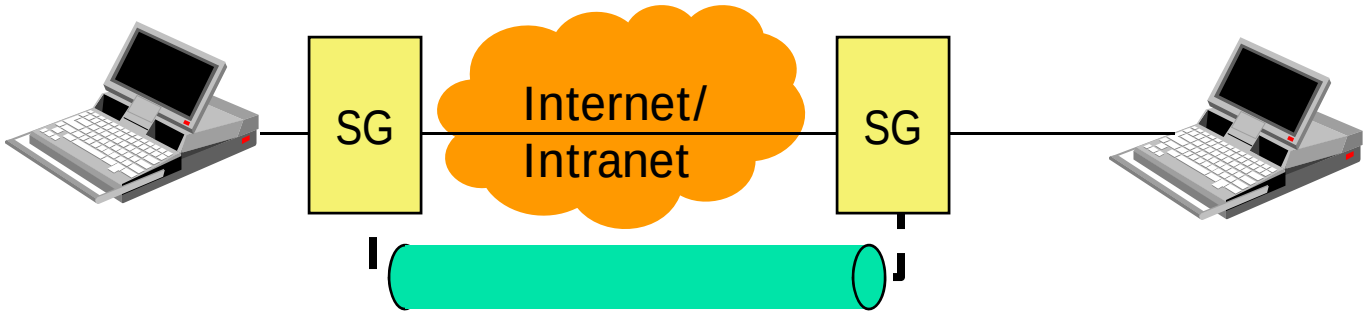
ISO/OSI Model
IPSec: Security at Network Layer



Cases where IPSec can be used

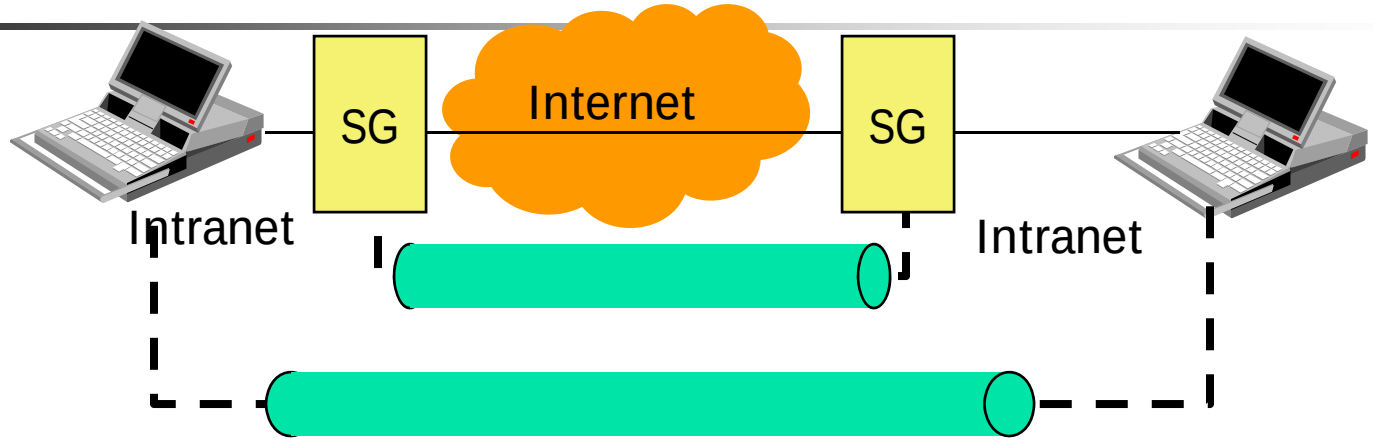


End-to-end security between two hosts

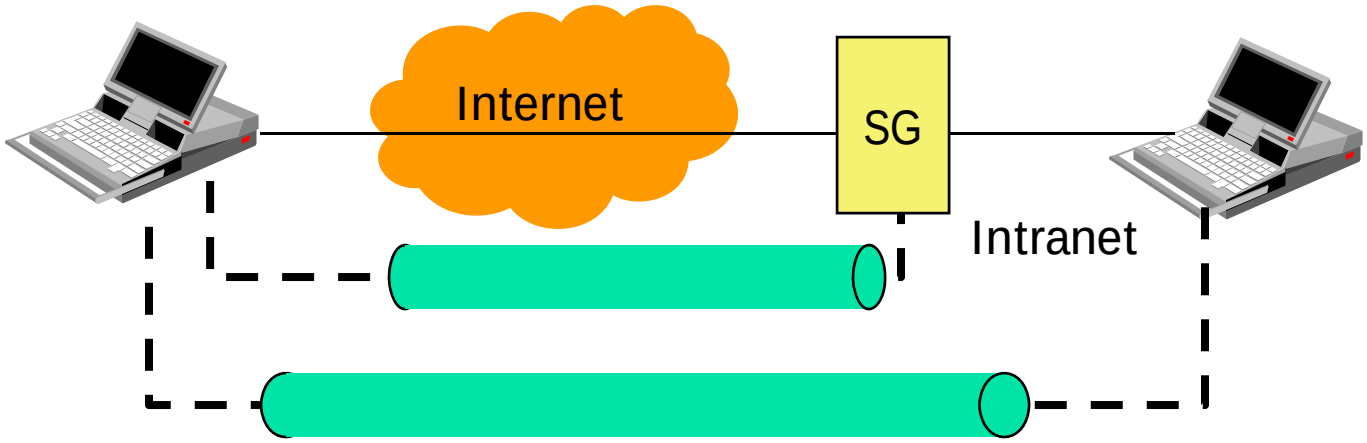


End-to-end security between two security gateways

Cases where IPSec can be used (2)



End-to-end security between two hosts + two gateways

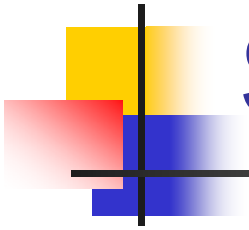


End-to-end security between two hosts during dial-up

IPSec Protocols

- Authentication header (AH) protocol
  - Message integrity
  - Origin authentication
  - Anti-replay services
- Encapsulating security payload (ESP) protocol
  - Confidentiality
  - Message integrity
  - Origin authentication
  - Anti-replay services
- Internet Key Exchange (IKE)
  - Exchanging keys between entities that need to communicate over the Internet
  - What authentication methods to use, how long to use the keys, etc.

Two types of IpSec headers

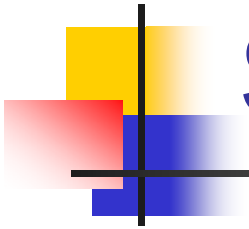


Security Association (SA)

- Cryptographic protected connection
- Unidirectional relationship between peers
- **Specifies the security services** provided to the traffic carried on the SA
  - Security enhancements to a channel along a path
- Identified by three parameters:
  - IP Destination Address
  - Security Protocol Identifier
    - Specifies whether AH or ESP is being used
  - Security Parameters Index (SPI)
    - Specifies the security parameters associated with the SA

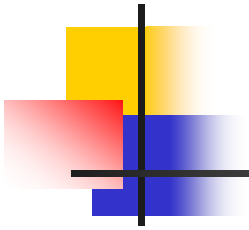
Security Association (2)

- Each SA uses AH or ESP (not both)
  - If both required two SAs are created
- Multiple security associations may be used to provide required security services
  - A sequence of security associations is called *SA bundle*
  - Example: We can have an AH protocol followed by ESP or vice versa



Security Association Databases

- IP needs to **know the SAs that exist** in order to provide security services
- Security Policy Database (SPD)
  - IPSec uses SPD to handle messages
    - For each IP packet, it decides whether an IPSec service is provided, bypassed, or if the packet is to be discarded
- Security Association Database (SAD)
    - Keeps track of the sequence number
    - AH information (keys, algorithms, lifetimes)
    - ESP information (keys, algorithms, lifetimes, etc.)
    - Lifetime of the SA
    - Protocol mode
    - MTU et.c.

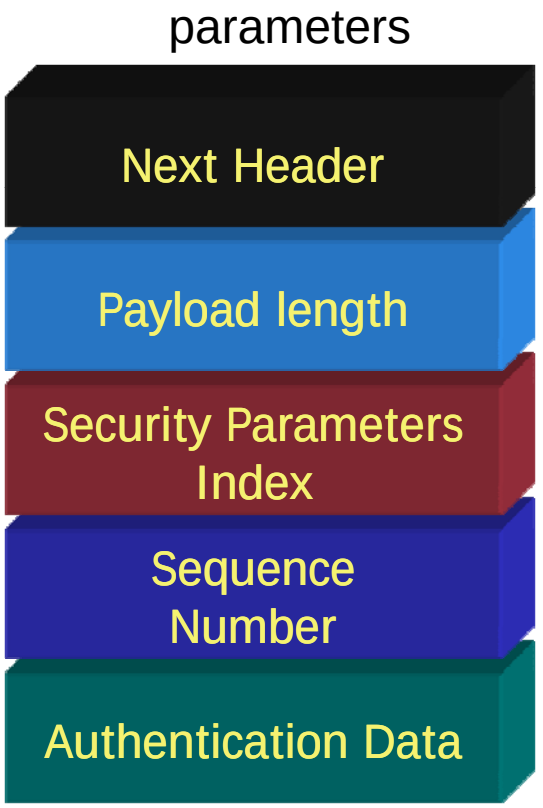


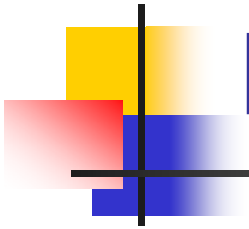
IPSec Modes

- Two modes
  - Transport mode
    - Encapsulates IP packet data area
    - IP Header is not protected
      - Protection is provided for the upper layers
      - Usually used in host-to-host communications
  - Tunnel mode
    - Encapsulates entire IP packet in an IPSec envelope
      - Helps against traffic analysis
      - The original IP packet is untouched in the Internet

Authentication Header (AH)

- Next header
  - Identifies what protocol header follows
- Payload length
  - Indicates the number of 32-bit words in the authentication header
- Security Parameters Index
  - Specifies to the receiver the algorithms, type of keys, and lifetime of the keys used
- Sequence number
  - Counter that increases with each IP packet sent from the same host to the same destination and SA
- Authentication Data
  - Crypto integrity check on the data

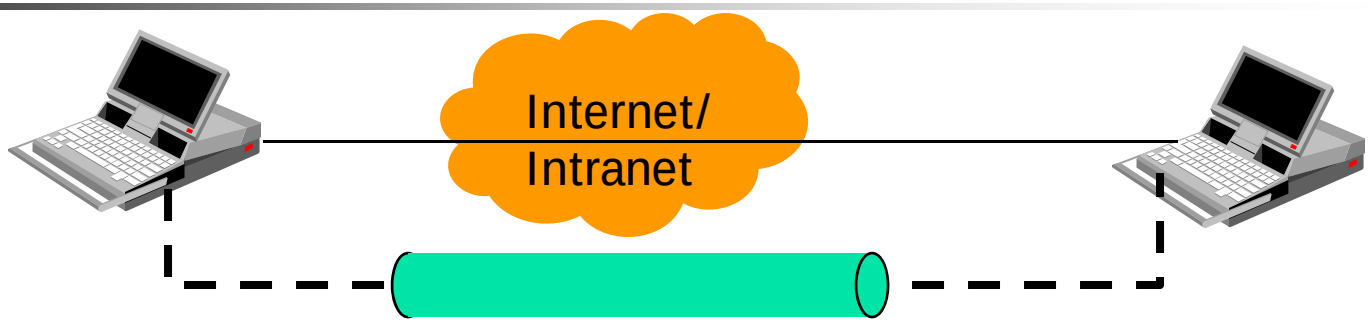


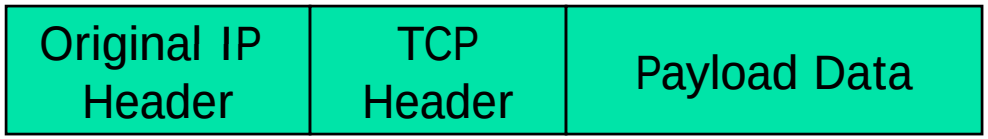


Preventing replay

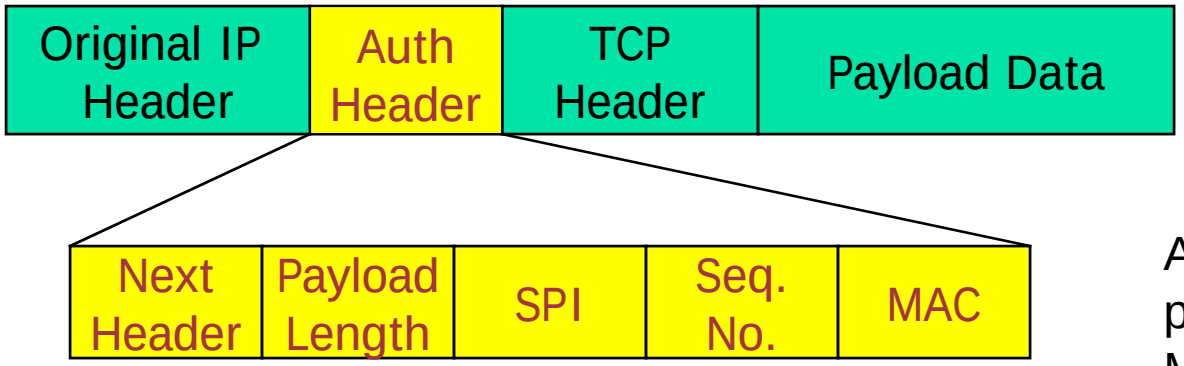
- Using 32 bit sequence numbers helps detect replay of IP packets
- The sender initializes a sequence number for every SA
- Receiver implements a window size of W to keep track of authenticated packets
- Receiver checks the MAC to see if the packet is authentic

Transport Mode AH



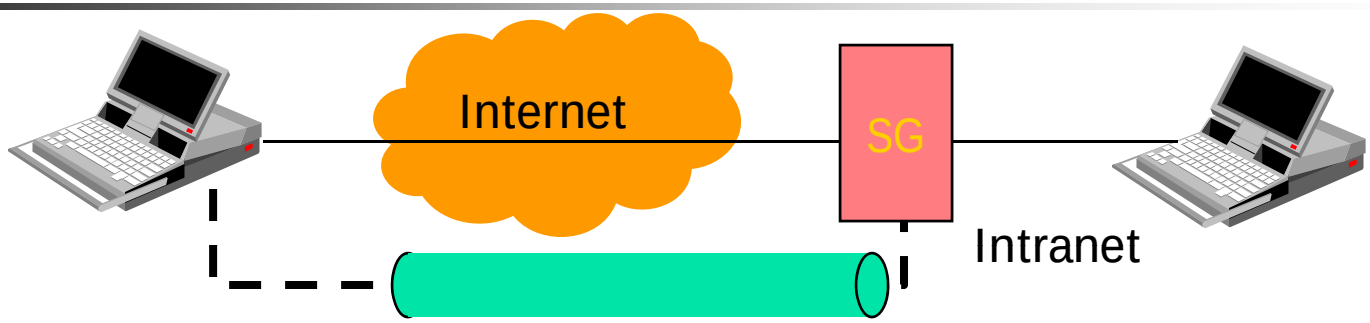


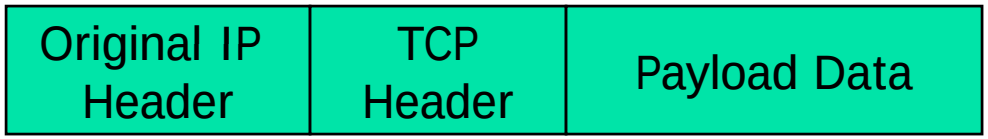
Without IPSec



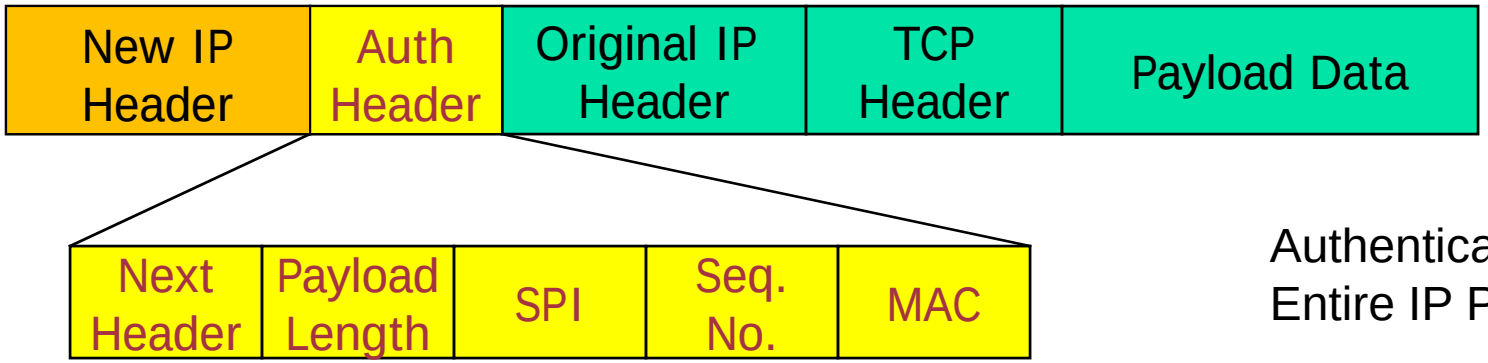
Authenticate Entire packet except for Mutable fields

Tunnel Mode AH



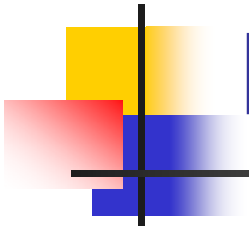


Without IPSec



Authenticate
Entire IP Packet

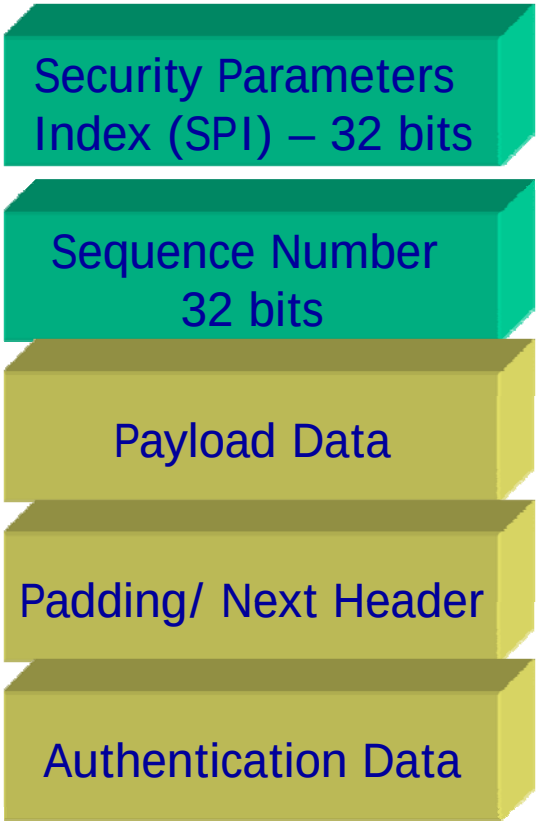
ESP – Encapsulating Security Payload



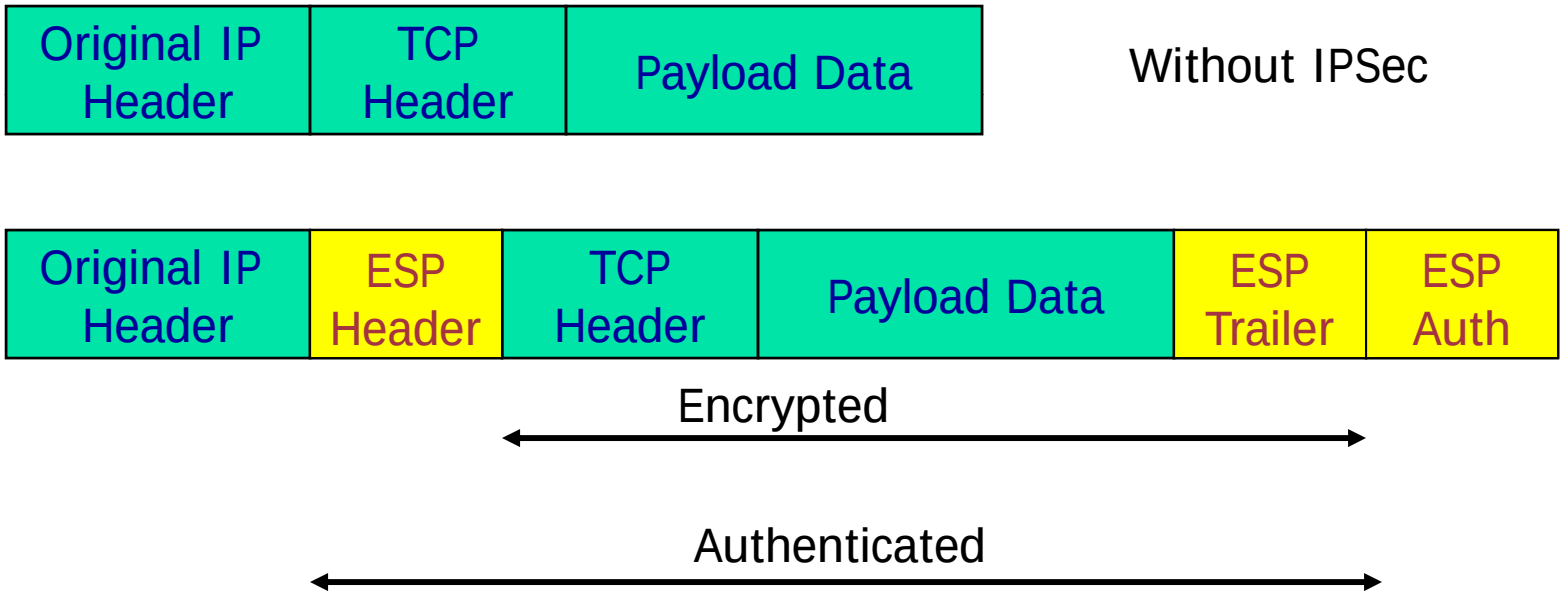
- Creates a new header in addition to the IP header
- Creates a new trailer
- Encrypts the payload data
- Authenticates
- Prevents replay

ESP – Encapsulating Security Payload

- Security Parameters Index (SPI)
  - Specifies to the receiver the algorithms, type of keys, and lifetime of the keys used
- Sequence number
  - Counter that increases with each IP packet sent from the same host to the same destination and SA
- Payload (variable)
  - TCP segment (transport mode) or IP packet (tunnel mode) - encryption
- Padding (+ Pad length, next Header)
  - 0 to 255 bytes of data to enable encryption algorithms to operate properly
- Authentication Data
  - MAC created over the packet



Transport mode ESP



Tunnel mode ESP