CAN201: Introduction to Networking

Lecture 12 - Network Security 3

Lecturer: Dr. Gordon Boateng & Dr. Fei Cheng

Important Information

- **Contact:**
  - Email: Gordon.Boateng@xjtlu.edu.cn
  - Office No.: SC 554A

- **Office Hours (Strictly via appointment)**
  - **Tuesday: 14:00-15:00**
  - **Wednesday: 14:00-15:00**

Network Security 3: roadmap

- Network layer security: IPSec
- Operational security: firewalls and IDS

What is network-layer confidentiality ?

TLS secures individual connections; IP layer security protects all IP packets

Between two network entities:

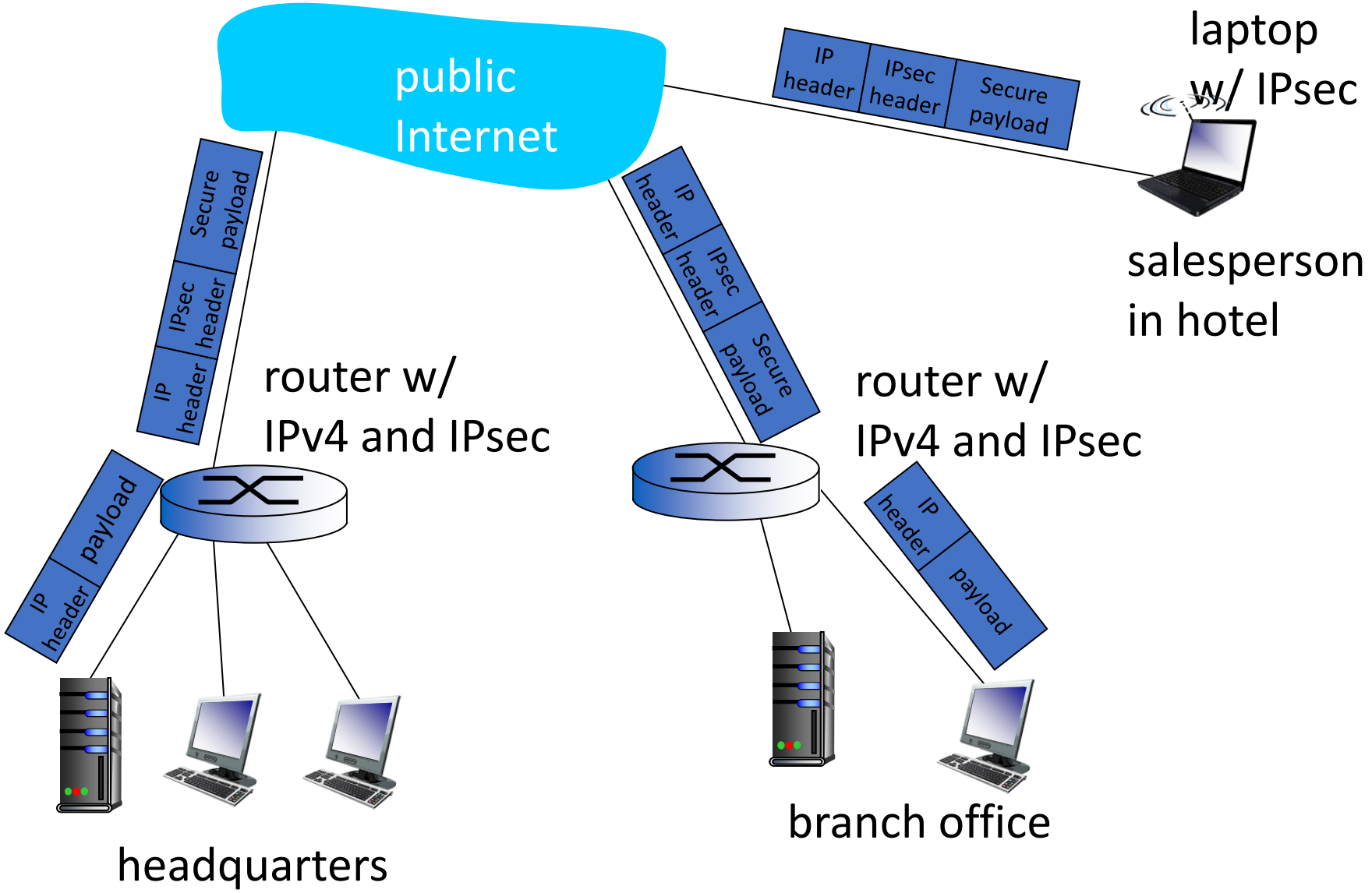
- sending entity encrypts datagram payload, payload could be:
  - TCP or UDP segment, ICMP message, OSPF message ….
- all data sent from one entity to other would be hidden from any third party (that presumably is sniffing the network):
  - web pages, e-mail, P2P file transfers, TCP SYN packets …
- "blanket coverage across apps"

Virtual Private Networks (VPNs)

Motivation:

- **Institutions often want private networks for security.**
  - An institution could actually deploy a stand-alone physical network that is completely separate from the public Internet.
  - costly: separate routers, links, DNS infrastructure.
- **VPN: institution's inter-office traffic is sent over public Internet instead**
  - encrypted before entering public Internet
  - logically separate from other traffic

Virtual Private Networks (VPNs)

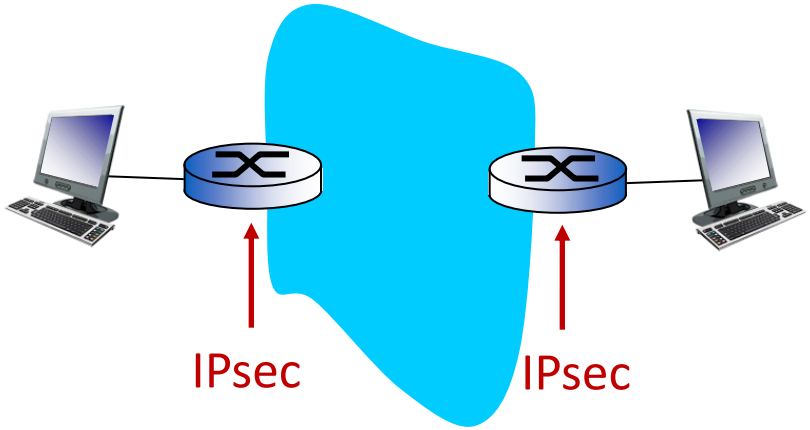


IPsec services

- **data integrity – detect modification**
- **origin authentication – confirm sender identity**
- **replay attack prevention – stop packet re-injection**
- **confidentiality – encrypt payload**

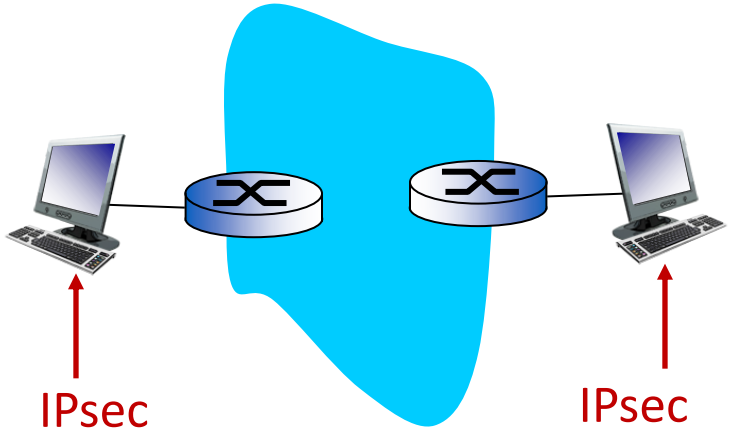
- **two protocols providing different service models:**
  - **Authentication Header (AH)**
    - provides source authentication and data integrity but *no* confidentiality
  - **Encapsulation Security Payload (ESP)**
    - provides source authentication, data integrity, and confidentiality
    - more widely used than AH

IPsec – tunneling mode & host mode



- edge routers IPsec-aware

  (between routers & firewalls)



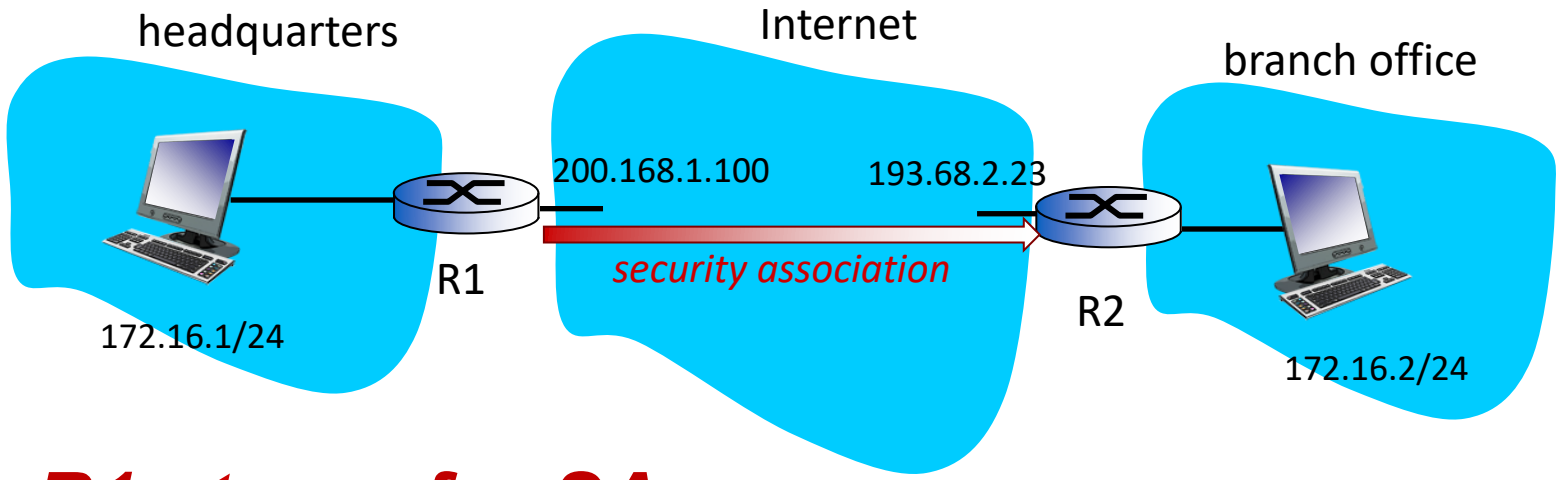
- hosts IPsec-aware

  (E2E encryption)

Security Associations (SAs)

- **before sending data, "security association (SA)" established from sending to receiving entity**
  - SAs are simplex: logical connection for only one direction
- **ending, receiving entitles maintain *state information* about SA**
- **how many SAs in VPN w/ one headquarters office, one branch office, and n traveling salesperson?**

Example SA from R1 to R2



R1 stores for SA:

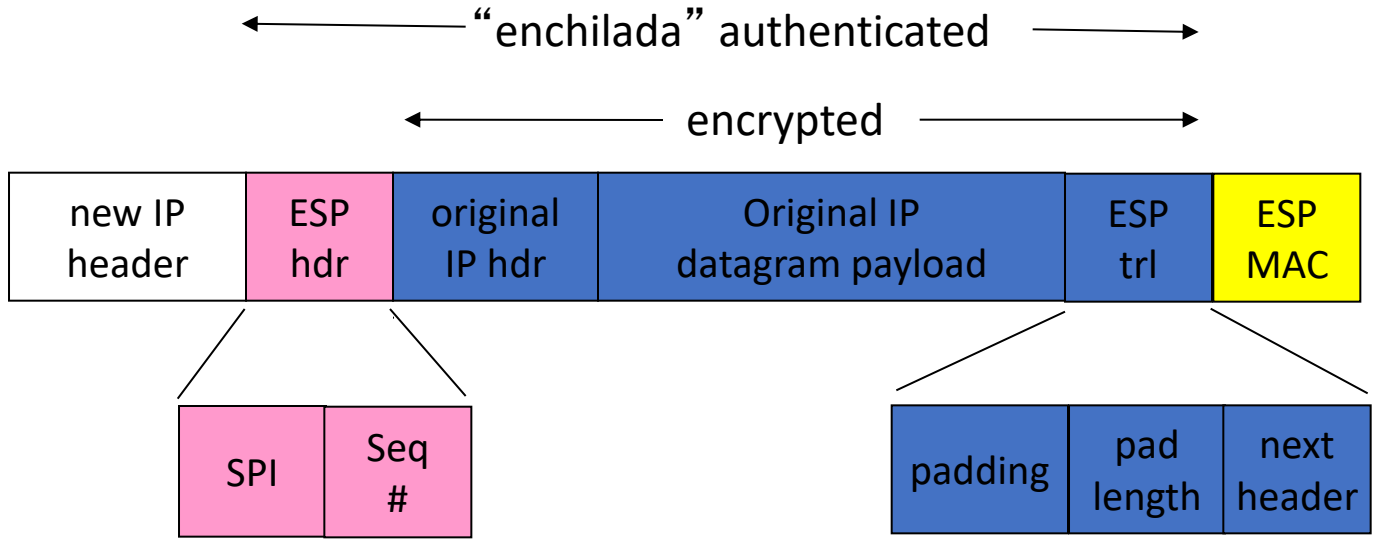
- **32-bit SA identifier: *Security Parameter Index (SPI)***
- **origin SA interface (200.168.1.100)**
- **destination SA interface (193.68.2.23)**
- **type of encryption used (e.g., 3DES with Cipher Block Chaining (CBC))**
- **encryption key**
- **type of integrity check used (e.g., HMAC with MD5)**
- **authentication key**

Security Association Database (SAD)

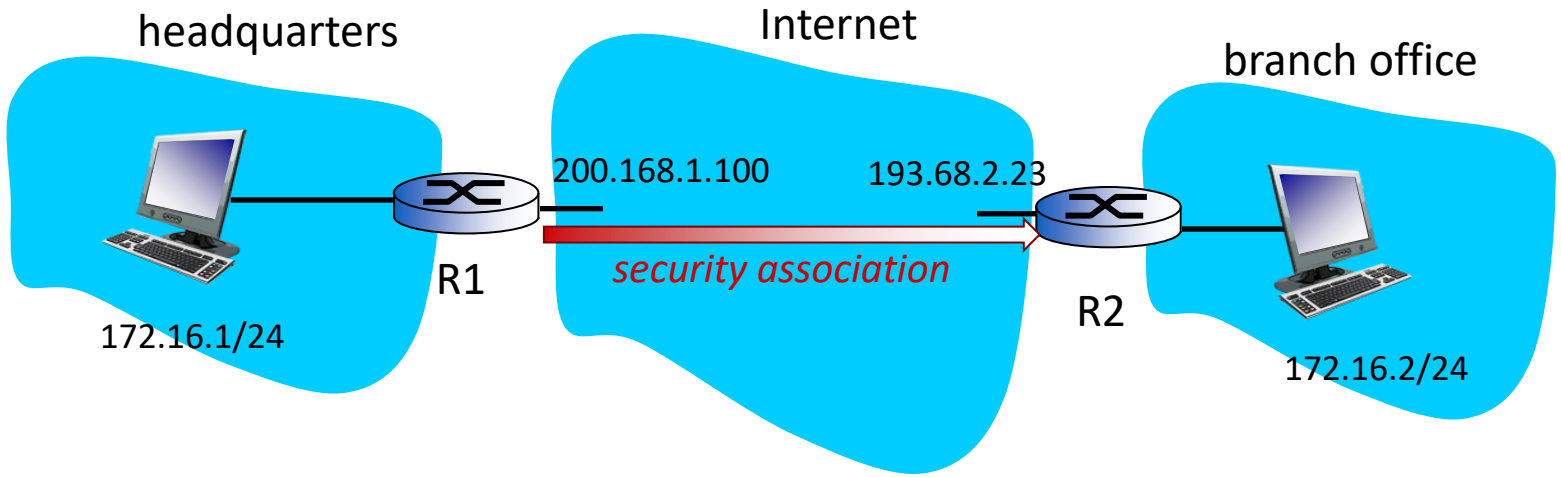
- Each endpoint holds SA state in *security association database (SAD)*, where it can locate them during processing.
- with n salespersons, 2 + 2n SAs in R1's SAD
- when sending IPsec datagram, R1 accesses SAD to determine how to process datagram.
- when IPsec datagram arrives to R2, R2 examines Security parameter index (SPI) in IPsec datagram, indexes SAD with SPI, and processes datagram accordingly.

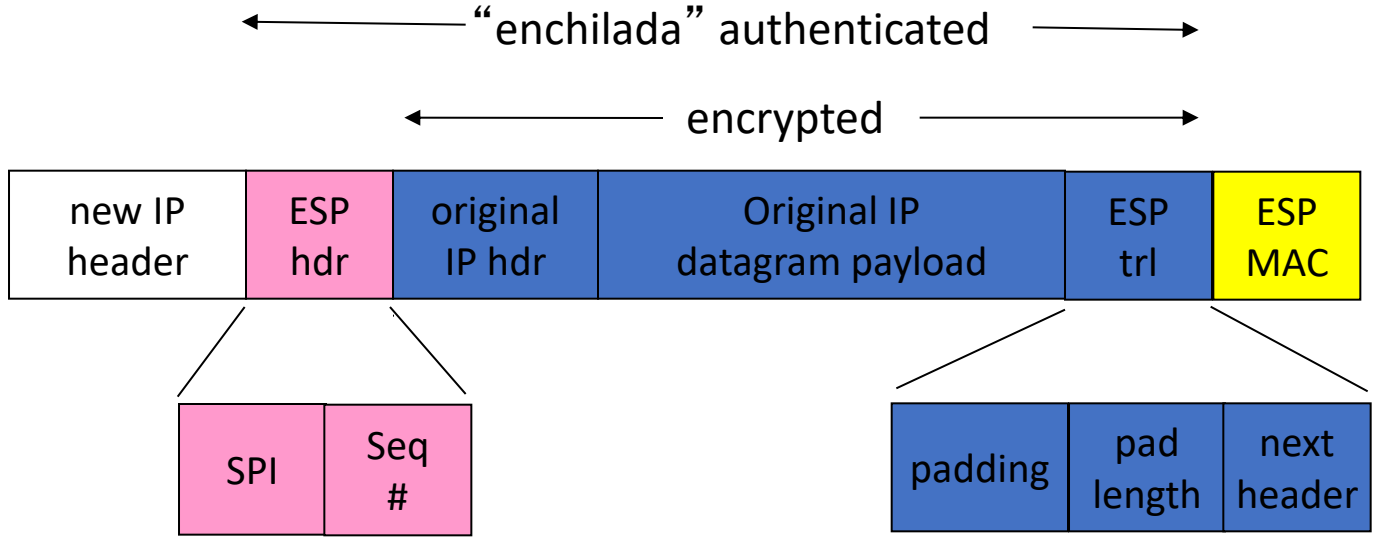
IPsec datagram

focus for now on tunnel mode with ESP



What happens?

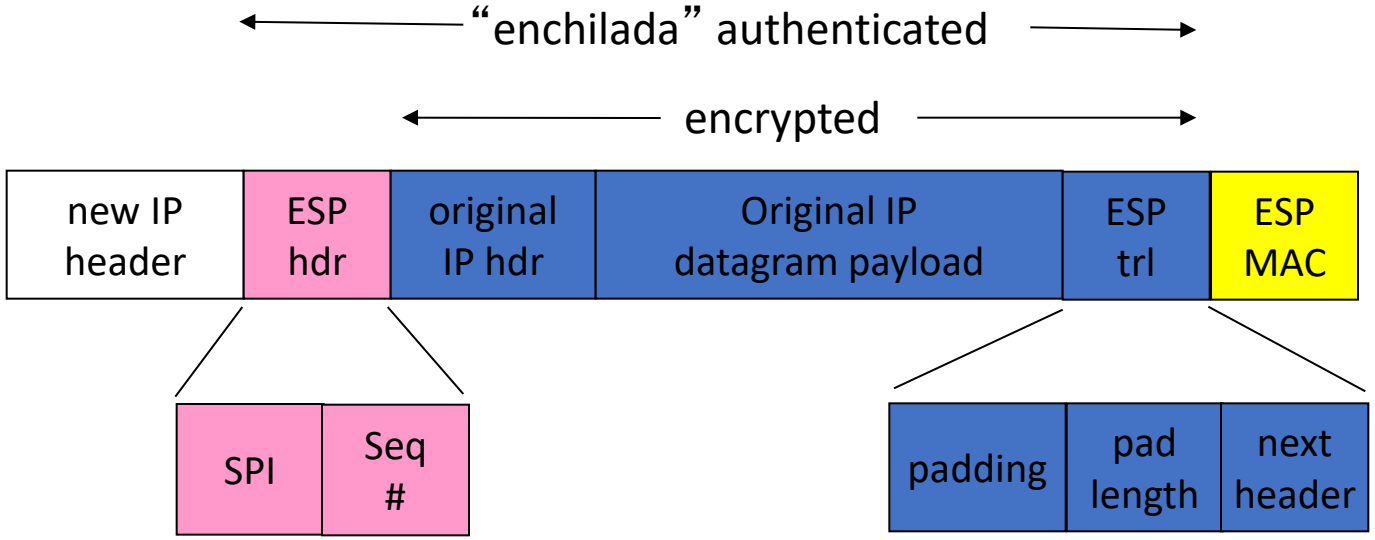




R1: convert original datagram to IPsec datagram

- appends to back of original datagram (that includes original header fields!) an "ESP trailer" field.
- encrypts result using algorithm & key specified by SA.
- appends to front of this encrypted quantity the "ESP header, creating "enchilada".
- creates authentication MAC over the *whole enchilada*, using algorithm and key specified in SA.
- appends MAC to back of enchilada, forming *payload*.
- creates brand new IP header, with all the classic IPv4 header fields, which it appends before payload.

Inside the enchilada:



- **ESP trailer: Padding for block ciphers**
- **ESP header:**
  - SPI, so receiving entity knows what to do
  - Sequence number, to thwart replay attacks
- **MAC in ESP auth field is created with shared secret key**

Security Policy Database (SPD)

- **policy: For a given datagram, sending entity needs to know if it should use IPsec or vanilla IP**
- **needs also to know which SA to use**
  - may use: source and destination IP address; protocol number
- **info in SPD indicates "what" to do with arriving datagram**
- **info in SAD indicates "how" to do it**

Summary: IPsec services

- **suppose Trudy sits somewhere between R1 and R2. she doesn't know the keys.**
  - will Trudy be able to see original contents of datagram? How about source, dest IP address, transport protocol, application port?
  - flip bits without detection?
  - masquerade as R1 using R1's IP address?
  - replay a datagram?

IKE: Internet Key Exchange

- ***previous examples:*** **manual establishment of IPsec SAs in IPsec endpoints:**

  Example SA

  SPI: 12345
  Source IP: 200.168.1.100
  Dest IP: 193.68.2.23
  Protocol: ESP
  Encryption algorithm: 3DES-cbc
  HMAC algorithm: MD5
  Encryption key: 0x7aeaca…
  HMAC key:0xc0291f…

- **manual keying is impractical for VPN with 100s of endpoints**
- **instead use *IPsec IKE (Internet Key Exchange)*** protocol, specified in RFC 5996.

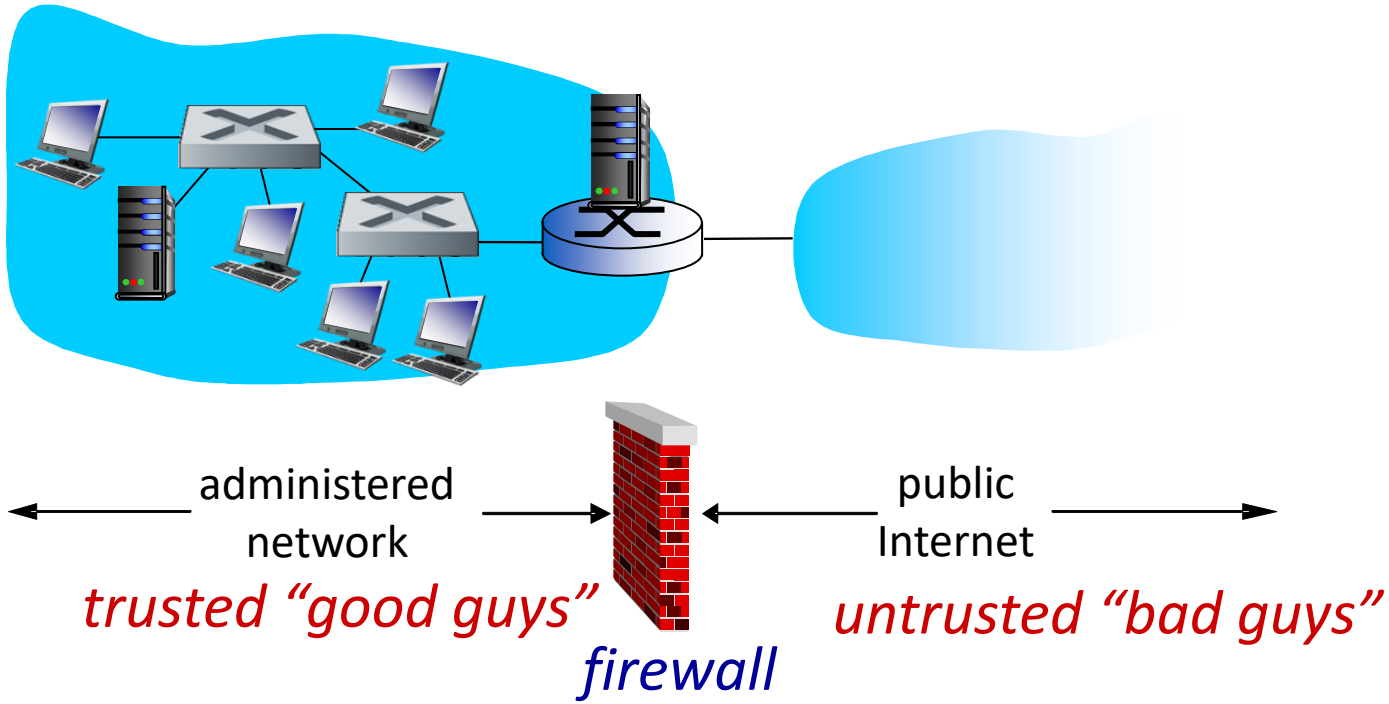
Network Security 3: roadmap

- Network layer security: IPSec
- Operational security: firewalls and IDS

Firewalls

firewall

isolates organization's internal net from larger Internet, allowing some packets to pass, blocking others



Firewalls: why

prevent denial of service attacks:

- SYN flooding: attacker establishes many bogus TCP connections, no resources left for “real” connections

prevent illegal modification/access of internal data

- e.g., attacker replaces CIA’s homepage with something else

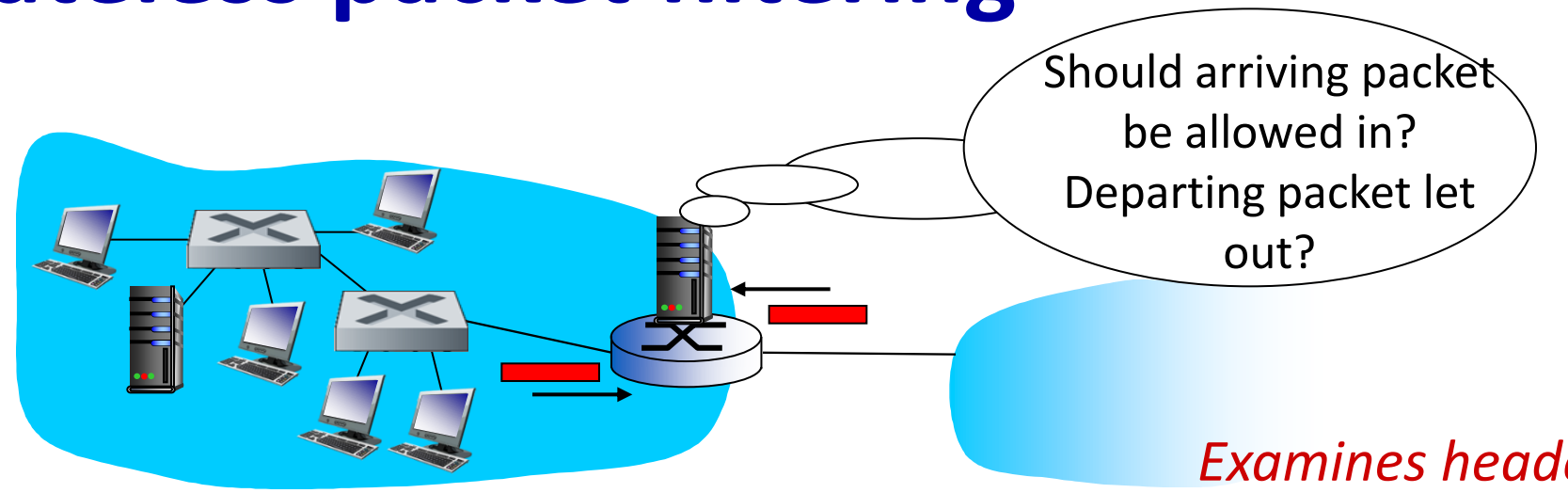
allow only authorized access to inside network

- set of authenticated users/hosts

three types of firewalls:

- stateless packet filters (network layer)
- stateful packet filters (transport layer)
- application gateways (application layer)

Stateless packet filtering



- internal network connected to Internet via *router firewall*
- router *filters packet-by-packet,* decision to forward/drop packet based on:
  - source IP address, destination IP address
  - TCP/UDP source and destination port numbers
  - ICMP message type
  - TCP SYN and ACK bits

Stateless packet filtering: example

- *example 1:* block incoming and outgoing datagrams with IP protocol field = 17 and with either source or dest port = 23
  - *result:* all incoming, outgoing UDP flows and telnet connections are blocked

- *example 2:* block inbound TCP segments with ACK=0.
  - *result:* prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside.

Stateful packet filtering

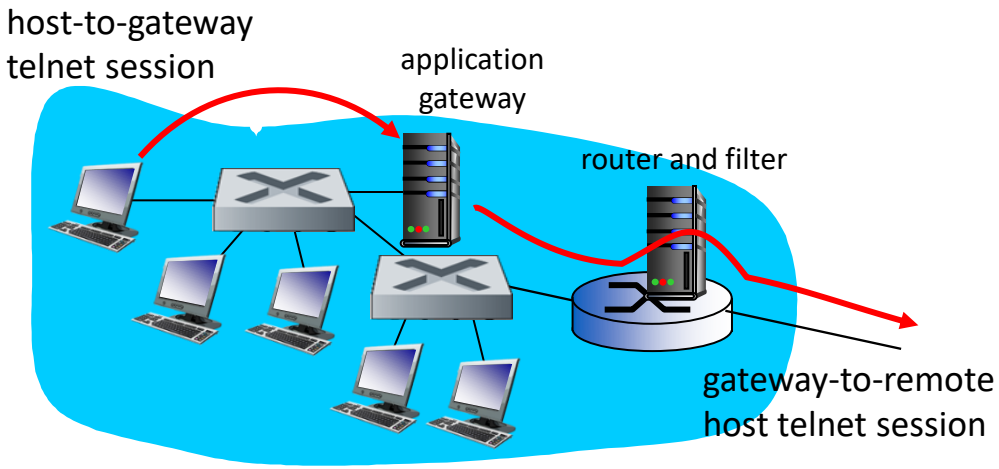
- *stateful packet filter:* track status of every TCP connection
  - track connection setup (SYN), teardown (FIN): determine whether incoming, outgoing packets "makes sense"
  - timeout inactive connections at firewall: no longer admit packets

Tracks connection state

Application gateways

Inspects application data

- filter packets on application data as well as on IP/TCP/UDP fields.
- *example:* allow select internal users to telnet outside



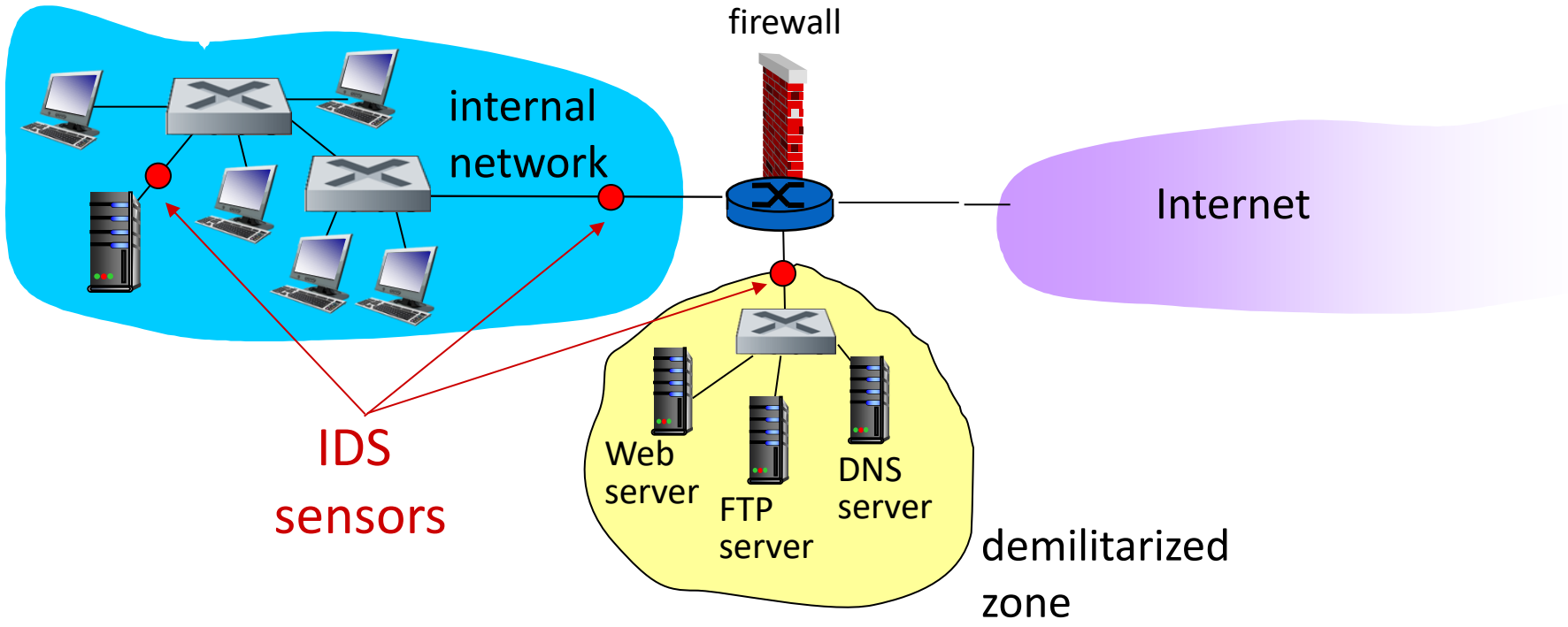
1. require all telnet users to telnet through gateway.
2. for authorized users, gateway sets up telnet connection to dest host. Gateway relays data between 2 connections.
3. router filter blocks all telnet connections not originating from gateway.

Intrusion detection systems

- **packet filtering:**
  - operates on TCP/IP headers only
  - no correlation check among sessions
- ***IDS: intrusion detection system***
  - *deep packet inspection:* look at packet contents (e.g., check character strings in packet against database of known virus, attack strings)
  - examine correlation among multiple packets
    - port scanning
    - network mapping
    - DoS attack

Intrusion Detection Systems (IDS)

multiple IDSs: different types of checking at different locations



Thank You & Good Luck!