

University of Rochester
CSC290B
Introduction to Computer Security

More Network Security

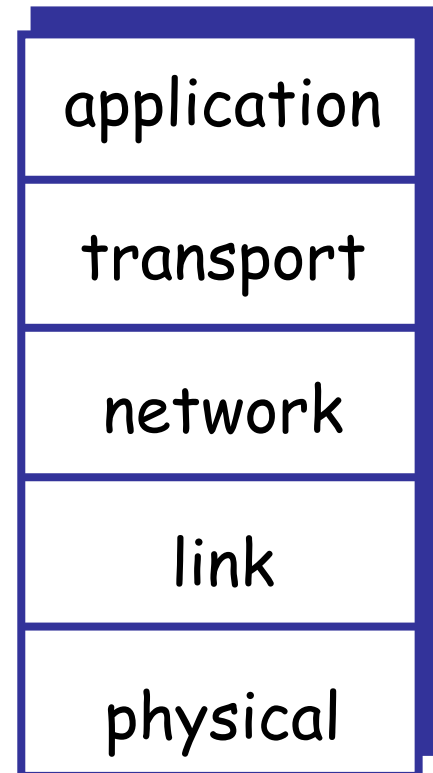
April 23, 2009

Network Architecture

The OSI Model: Network Layers

[OSI also defines **presentation** and **session** layers]

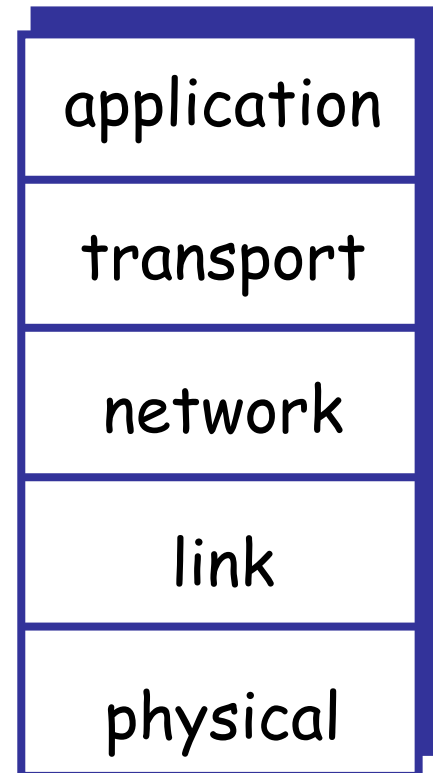
- **application**: anything you want to do on computer networks
- **transport**: host-to-host data transport
 - reliable data transport, congestion control, flow control
- **network**: host-to-host connectivity
 - routing, addressing
- **link**: data transfer between neighboring network elements
 - encoding, framing, error correction, access control for shared links
- **physical**: electromagnetic signals “on the wire”



Network Architecture

For example:

- **application:** remote sessions with SSH or telnet, email with SMTP, domain name service with DNS protocols
- **transport:** TCP for flow-controlled congestion-fair in-order byte streams, UDP for unordered datagrams
- **network:** IP for routing of packets through routers, BGP for route calculation and communication
- **link:** Ethernet among nearby wired hosts, Wi-Fi among wireless access points and hosts
- **physical:** Electrical signals on baseband media, radio signals on wireless media



Attacks on the Application Layer

- Examples
 - Exploit bugs in application implementations: browsers using HTTP, mail transfer agents using SMTP, terminal servers using SSH
 - Create new applications to use existing protocols in new ways:
 - Fast Flux DNS servers create ever-changing network of proxy hosts and location services for them (for phishing web sites)
 - Botnet implementations with intricate command-and-control mechanisms

Attacks on the Transport Layer

- Examples:
 - Exploit bugs in protocol handling: Very unusual protocol fields or bad length fields trigger kernel bugs, resulting in compromise or denial of service
 - Unusual but legal packet sequences to slip past intrusion detection systems: Retransmit TCP segments, with a mix of innocent-looking data with already-acked sequence numbers (won't be passed to applications), plus bits of hostile data in the new as-yet-unacked sequence numbers
 - Exploit protocol features that require state maintenance, cause denial of service by forcing machines to commit massive resources

Attacks on the Network Layer

- Examples:
 - Attacks on the routing protocols, exploit too much trust (and not enough route filtering) among routers to inject bad routing table information, create routes through hostile machines, to snooping, spoofing, or denial of service problems.
 - Exploit lack of checking or impossibility of checking certain protocol fields, inject spoofed packets that look like they are coming from one place, are really coming from another.

Attacks on the Link Layer

- Examples:
 - Interfere with inter-switch communication, “layer 2” packet handling among switches, with similar opportunities as network layer attacks (snooping, spoofing, MITM)
 - Spanning Tree Protocol (STP) interference
 - VLAN hopping
 - Address Resolution Protocol (ARP) Poisoning [spans network and link layer]

Attacks on the Physical Layer

- Examples:
 - Physical interception or modification or disruption of signals
 - Wire tapping
 - Fiber tapping
 - Jamming

Strategies for the Application Layer

- Examples:
 - Many

Strategies for the Network Layer

- Examples:
 - IP Security (IPsec)
 - Internet Security Association Key Management Protocol (ISAKMP) Key Exchange (IKE) establishes security associations (bundles of security state: protocols, algorithms, and keys)
 - Encapsulation Security Payload (ESP) headers to encrypt data
 - Transport mode encapsulates data only, leaves headers (e.g. addresses) intact
 - Tunnel mode encapsulates entire IP packets, so there is an “outer” address and an “inner” address (typical use: virtual private networks [VPNs])
 - Wrap data payloads with Authentication headers to provide integrity features

Strategies for the Transport and Network Layers

- Examples:
 - Network Address Translation (NAT)
 - TCP and UDP transport sessions based on (local host, local port, remote host, remote port) tuples.
 - Intermediate devices (network address translating firewalls) can modify packets to use different host addresses and port numbers
 - NAT state maintained in firewall
 - Use of private IP address space (RFC1918) gives some assurance that all sessions are completely mediated by the firewall, since those addresses are only reachable using the rules and state in the firewall

Availability

- Access over Internet must be unimpeded
 - Context: flooding attacks, in which attackers try to overwhelm system resources
- Example: SYN flood
 - Problem: server cannot distinguish legitimate handshake from one that is part of this attack
 - Only difference is whether third part of TCP handshake is sent
 - Flood can overwhelm communication medium
 - Can't do anything about this (except buy a bigger pipe)
 - Flood can overwhelm resources on our system
 - We start here

Intermediate Hosts

- Use routers to divert, eliminate illegitimate traffic
 - Goal: only legitimate traffic reaches firewall
 - Example: Cisco routers try to establish connection with source (TCP intercept mode)
 - On success, router does same with intended destination, merges the two
 - On failure, short time-out protects router resources and target never sees flood

Intermediate Hosts

- Use network monitor to track status of handshake
 - Example: synkill monitors traffic on network
 - Classifies IP addresses as not flooding (good), flooding (bad), unknown (new)
 - Checks IP address of SYN
 - If good, packet ignored
 - If bad, send RST to destination; ends handshake, releasing resources
 - If new, look for ACK or RST from same source; if seen, change to good; if not seen, change to bad
 - Periodically discard stale good addresses

Endpoint Hosts

- Control how TCP state is stored
 - When SYN received, entry in queue of pending connections created
 - Remains until an ACK received or time-out
 - In first case, entry moved to different queue
 - In second case, entry made available for next SYN
 - In SYN flood, queue is always full
 - So, assure legitimate connections space in queue to some level of probability
 - Two approaches: SYN cookies or adaptive time-outs

SYN Cookies

- Source keeps state
- Example: Linux 2.4.9 kernel
 - Embed state in sequence number
 - When SYN received, compute sequence number to be function of source, destination, counter, and random data
 - Use as reply SYN sequence number
 - When reply ACK arrives, validate it
 - Must be hard to guess

Adaptive Time-Out

- Change time-out time as space available for pending connections decreases
- Example: modified SunOS kernel
 - Time-out period shortened from 75 to 15 sec
 - Formula for queueing pending connections changed:
 - Process allows up to b pending connections on port
 - a number of completed connections but awaiting process
 - p total number of pending connections
 - c tunable parameter
 - Whenever $a + p > cb$, drop current SYN message

Acknowledgements

- Substantial portions of these slides are ©2002-2004 Matt Bishop and used with permission