

IPv4 Overview

Cyber Security
Spring 2006

Outline

- Review Layered Network Architecture
- Network Layer protocols
- Transport Layer Protocols
- Application Layer Protocols

Reading Material

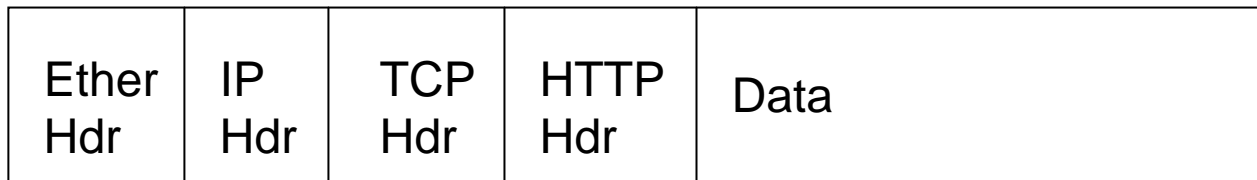
- Many texts on IP networking
 - Computer Networks, Andrew Tannenbaum
 - Data and Computer Communications, William Stallings
 - Internetworking with TCP/IP Vol 1, Douglas Comer
- Plus all the originals from the Internet Engineering Task Force (IETF)
 - <http://ietf.org/>

OSI Reference Model

- The layers
 - 7: Application, e.g., HTTP, SMTP, FTP
 - 6: Presentation
 - 5: Session
 - 4: Transport, e.g. TCP, UDP
 - 3: Network, e.g. IP, IPX
 - 2: Data link, e.g., Ethernet frames, ATM cells
 - 1: Physical, e.g., Ethernet media, ATM media
- Standard software engineering reasons for thinking about a layered design

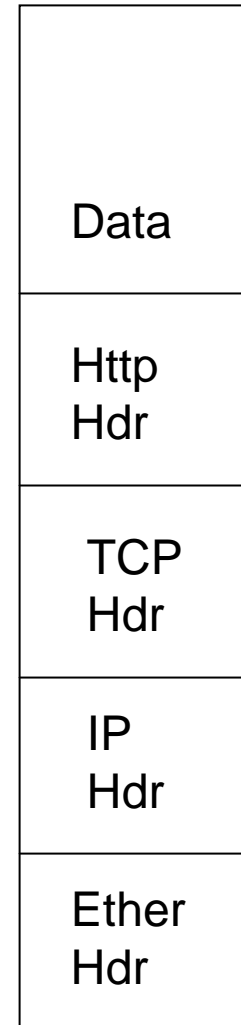
Layers Limit Need for Intelligence

- Intermediate devices only need to process the packet headers up to the level they understand



Various network devices

- Hosts and servers – Operate at Level 7 (application)
- Proxies – Operate at level 7
- Firewalls – Operate between levels 2 and 7. From the outside world make changes at levels 2 (in transparent mode) or 3 (in routing mode)
- Routers – Operate at Level 3 (network)
- Switches or Hubs – Operate at level 2 (data link)
- Gateways – Operate at level 2



IPv4

- 32 bit Addressing scheme
 - Host address, e.g., 192.168.1.1
 - Network address, e.g., 192.168.1.0/24 or 192.168.1.0 255.255.255.0
 - Host address is the first address in subnetwork, e.g. 192.168.1.0
 - Broadcast address is the last address in the subnetwork, e.g., 192.168.1.255

Version	IHL	Type of service	Total length		
Identification			DF	MF	Frag Offset
Time to live	Protocol		Header checksum		
Source address					
Destination Address					
0 or more words of options					

Address spoofing

- Sender can put any source address in packets he sends:
 - Can be used to send unwelcome return traffic to the spoofed address
 - Can be used to bypass filters to get unwelcome traffic to the destination
- Reverse Path verification can be used by routers to broadly catch some spoofers

Fragmentation

- May need to fragment an IP packet if one data link along the way cannot handle the packet size
 - Perhaps path is a mix of different HW
 - Perhaps unexpected encapsulation makes the packet larger than the source expected
 - Hosts try to understand Maximum Transmission Unit (MTU) to avoid the need for fragmentation (which causes a performance hit)
- Any device along the way can fragment
 - Identification field identifies all elements of the same fragment
 - Fragmentation stored in the MF (more fragments) and fragment offset fields
 - Devices can reassemble too
 - But generally the destination does the reassembly

Fragmentation Flaws

- Split packet to fool simple firewall and IDS
 - Intermediate content observers must do reassembly
- Overlapping fragments
 - Can be used to trick IDS by hiding, e.g. a “get /etc/password” request
 - Different clients reassemble overlapping fragments differently
 - Just drop overlapping fragments
- Bad fragment offsets exploit poor stack implementations
 - E.g. Teardrop attack, negative offsets or overlarge offsets cause buffer overflows
 - Firewalls can check for well formed packets.
- Resource attacks on re-assemblers
 - Send all but one fragment for many packets

Address Resolution Protocol (ARP)

- Used to discover mapping of neighboring ethernet MAC to IP addresses.
 - Need to find MAC for 192.168.1.3 which is in your interfaces subnetwork
 - Broadcast an ARP request on the link
 - Hopefully receive an ARP reply giving the correct MAC
 - The device stores this information in an ARP cache or ARP table

ARP cache poisoning

- Bootstrap problem with respect to security. Anyone can send an ARP reply
 - The Ingredients to ARP Poison, <http://www.governmentsecurity.org/articles/TheIngredientsToARPPoison.php>
- Classic Man-in-the-middle attack
 - Send arp reply messages to device so they think your machine is someone else
 - Better than simple sniffing because packets will get to your regardless of sniffing.
- Solutions
 - Encrypt all traffic
 - Monitoring programs like arpwatch to detect mapping changes
 - Which might be valid due to DHCP

Basic IPv4 Routing

- Static routing. Used by hosts and some firewalls and routers.
 - Routing table consists of entries of
 - Network, Next hop address, metric, interface
 - May have routing table per incoming interface
 - To route a packet, take the destination address and find the best match network in the table. In case of a tie look at the metric
 - Use the corresponding next hop address and interface to send the packet on.
 - The next hop address is on the same link as this device, so you use the next hop's data-link address, e.g. ethernet MAC address
 - Decrement “time to live” field in IP header at each hop. Drop packet when it reaches 0
 - Attempt to avoid routing loops
 - As internet got bigger, TTL fields got set bigger. 225 maximum

Routing example

- Receive a packet destined to 192.168.3.56 on inside interface
- Local routing table for inside interface
 1. 192.168.2.0/30, 127.0.0.1, 1, outside
 2. 192.168.5.0/29, 127.0.0.1, 1, dmz
 3. 192.168.3.0/24, 192.168.5.6, 1, dmz
 4. 192.168.3.0/24, 192.168.1.2, 3, outside
 5. 0.0.0.0/0, 192.168.1.2, 1, outside
- Entries 3 and 4 tie. But metric for 3 is better
- Entries 1 and 2 are for directly connected networks

Source Based Routing

- In the IP Options field, can specify a source route
 - Was conceived of as a way to ensure some traffic could be delivered even if the routing table was completely screwed up.
- Can be used by the bad guy to avoid security enforcing devices
 - Most folks configure routers to drop packets with source routes set

IP Options in General

- Originally envisioned as a means to add more features to IP later
- Most routers drop packets with IP options set
 - Stance of not passing traffic you don't understand
 - Therefore, IP Option mechanisms never really took off
- In addition source routing, there are security Options
 - Used for DNSIX, a MLS network encryption scheme

Dynamic Routing Protocols

- For scaling, discover topology and routing rather than statically constructing routing tables
 - Open Shortest Path First (OSPF): Used for routing within an administrative domain
 - RIP: not used much anymore
 - Border Gateway Protocol (BGP): Used for routing between administrative domains. Can encode non-technical transit constraints, e.g. Domain X will only carry traffic of paying customers
 - Receives full paths from neighbors, so it avoids counts to infinity.

Dynamic Routing

- Injecting unexpected routes a security concern.
 - BGP supports peer authentication
 - BGP blackholing is in fact used as a mechanism to isolate “bad” hosts
 - Filter out route traffic from unexpected (external) points
 - OSPF has MD5 authentication, and can statically configure neighbor routers, rather than discover them.

Internet Control Message Protocol (ICMP)

- Used for diagnostics
 - Destination unreachable
 - Time exceeded, TTL hit 0
 - Parameter problem, bad header field
 - Source quench, throttling mechanism rarely used
 - Redirect, feedback on potential bad route
 - Echo Request and Echo reply, ping
 - Timestamp request and Timestamp reply, performance ping
- Can use information to help map out a network
 - Some people block ICMP from outside domain

Smurf Attack

- An amplification DoS attack
 - A relatively small amount of information sent is expanded to a large amount of data
- Send ICMP echo request to IP broadcast addresses. Spoof the victim's address as the source
- The echo request receivers dutifully send echo replies to the victim overwhelming it
- Fraggle is a UDP variant of the same attack

Transport layer

- UDP and TCP
- Transport flows are defined by source and destination ports
 - A pair of devices can have numerous flows operating simultaneously by communicating between different pairs of ports
- Applications are associated with ports (generally just destination ports)
 - IANA organizes port assignments <http://www.iana.org/>
- Source ports generally dynamically selected
 - Ports under 1024 are considered well-known ports
 - Would not expect source ports to come from the well-known range
- Scanners probe for listening ports to understand the services running on various machines

Datagram Transport

- User Datagram Protocol (UDP)
 - A best-effort delivery, no guarantee, no ACK
 - Lower overhead than TCP
 - Good for best-effort traffic like periodic updates
 - No long lived connection overhead on the endpoints
- Some folks implement their own reliable protocol over UDP to get “better performance” or “less overhead” than TCP
 - Such efforts don’t generally pan out
- TFTP and DNS protocols use UDP
- Data channels of some multimedia protocols, e.g., H.323 also use UDP

UDP Header

Source Port	Destination Port
UDP Length	UDP checksum

Reliable Streams

- Transmission Control Protocol (TCP)
 - Guarantees reliable, ordered stream of traffic
 - Such guarantees impose overhead
 - A fair amount of state is required on both ends
- Most Internet protocols use TCP, e.g., HTTP, FTP, SSH, H.323 control channels

TCP Header

Source Port				Destination Port					
Sequence Number									
Acknowledgement number									
HDR Len		U R G	A C K	P S H	R S T	S Y N	F I N	Window Size	
Checksum				Urgent Pointer					
Options (0 or more words)									

Syn flood

- A resource DoS attack focused on the TCP three-way handshake
- Say A wants to set up a TCP connection to B
 1. A sends SYN with its sequence number X
 2. B replies with its own SYN and sequence number Y and an ACK of A's sequence number X
 3. A sends data with its sequence number X and ACK's B's sequence number Y
- Send many of the first message to B. Never respond to the second message.
 - This leaves B with a bunch of half open (or embryonic) connections that are filling up memory
 - Firewalls adapted by setting limits on the number of such half open connections.

Application Protocols

- Single connection protocols
 - Use a single connection, e.g. HTTP, SMTP
- Dynamic Multi-connection Protocols, e.g. FTP and H.323
 - Have a well known control channel
 - Negotiate ports and/or addresses on the control channel for subsidiary data channels
 - Dynamically open the negotiated data channels
- Protocol suites, e.g. Netbios and DNS

Spoofing Applications

- Often times ridiculously easy
- Fake Client
 - Telnet to an SMTP server and enter mail from whoever you want
 - Authenticating email servers
 - Require a password
 - Require a mail download before server takes send requests
- Fake server
 - Phishing: misdirect user to bogus server

DHCP

- Built on older BOOTP protocol (which was built on even older RARP protocol)
 - Used by diskless Suns
- Enables dynamic allocation of IP address and related information
- Runs over UDP
- No security considered in the design, obvious problems
 - Bogus DHCP servers handing out addresses of attacker's choice
 - Bogus clients grabbing addresses
- IETF attempted to add DHCP authentication but rather late in the game to do this.
- Other solutions
 - Physically secure networks
 - Use IPSec

Domain Name System (DNS)

- Hierarchical service to resolve domain names to IP addresses.
 - The name space is divided into non-overlapping zones
 - E.g., consider shinrich.cs.uiuc.edu.
 - DNS servers in the chain. One for .edu, one for .uiuc.edu, and one for .cs.uiuc.edu
- Can have primary and secondary DNS servers per zone. Use TCP based zone transfer to keep up to date
- Like DHCP, no security designed in
 - But at least the DNS server is not automatically discovered
 - Although this information can be dynamically set via DHCP
- Queries and responses use UDP.
 - Packet interception attacks
 - Name chaining attacks
 - Untrustworthy, trustworthy servers

DNSSEC

- Seeks to solve the trust issues of DNS
- Uses a key hierarchy for verification
- Has been under development for a decade and still not really deployed
- Provides authentication, not confidentiality
- DNS Threat Analysis in RFC 3833.

Summary

- IPv4 not designed with security in mind
- Complexity can be exploited
 - Poor implementations
 - Edge cases in standards
- Bootstrapping can be exploited
 - Easy of configuration vs strong trust