CSE 124
Overlay networks + VPNs + IPSec

March 17, 2016
Prof. George Porter
Outline

• Today:
  – Reliable overlay networks
  – VPNs and IPSec

• Announcements:
  – Project 2, HW7/8 due today at 5
Abstractions and Overlays

Overlay Network

Physical Network
Overlays for routing: Why?

• Triangle inequality doesn’t hold in networks!
Forwarding traffic through tunneling
Overlay Networks for routing

- Underlying network
  - Internet connectivity (IP Routing)
• Potential overlay connectivity
  – SF as root
Overlay Networks

- Determine edge weights
  - E.g., bandwidth, latency
Overlay Networks

- Build overlay connectivity
  - An application-layer distribution tree
Overlay Networks

• We have had overlay networks for at least the past ~2 decades
  – Mbone, 6bone, etc.

• Orig. idea: these would be experimental networks that would help with the transition to “production” networks

• Today, overlay networks are being explored as general-purpose networks
  – Driven by content distribution networks and P2P computing
Overlay networks for fault tolerance

Resilient Overlay Networks
Figure 1: Comparison of RTT for pairs of PlanetLab hosts whose point-to-point latencies were larger than 400 ms (high-latency paths). For the “excluding top n%” graphs, we removed the top n% of one-hop alternatives for each high-latency path from consideration, then used the best remaining one-hop.
Virtual private networks

• Can we get the benefits of separate networks while maintaining the benefits of sharing a common network infrastructure?

• What are the benefits of separate networks?
  – Security?
  – Performance?
  – Addressing?
  – ?
Virtual circuits
Making tunnels secure

• Option 1: Application layer
• Option 2: Transport layer
• Option 3: Network layer
• Option 4: Link layer
• Option 5: Physical layer
IPSec overview

• Encrypted IP-level communication

• Transport mode:
  – Host-to-host secure connection
  – Encrypted, authenticated, or both

• Tunnel: an “overlay” network
  – Host-to-network or network-to-network
  – Entire IP packet tunneled in secure IPSec “envelope” to recovered at destination
Security in IPSec

• AH: Authentication header
  – Access control, message integrity, authentication, and antireplay protection
• ESP: Encapsulating Security Payload
  – Like AH, but with encryption too
• SA: Security association
  – Selection of algorithms, crypto, hashes, etc
• SPI: Security Parameters Index (SPI)
  – Per-connection index into SA database
• ISAKMP: Internet Security Association and Key Management Protocol
Standard IPv4 Datagram

<table>
<thead>
<tr>
<th>ver</th>
<th>hlen</th>
<th>TOS</th>
<th>pkt len</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ID</td>
<td>flgs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>frag offset</td>
</tr>
<tr>
<td>TTL</td>
<td>proto=TCP</td>
<td>header chksum</td>
<td></td>
</tr>
<tr>
<td>src IP address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst IP address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP Options (if present)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCP header (proto = 6)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCP payload</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Covered by header chksum

http://www.unixwiz.net/techtips/iguide-ipsec.html
# IP “next” protocols

<table>
<thead>
<tr>
<th>Protocol code</th>
<th>Protocol Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ICMP — Internet Control Message Protocol</td>
</tr>
<tr>
<td>2</td>
<td>IGMP — Internet Group Management Protocol</td>
</tr>
<tr>
<td>4</td>
<td>IP within IP (a kind of encapsulation)</td>
</tr>
<tr>
<td>6</td>
<td>TCP — Transmission Control Protocol</td>
</tr>
<tr>
<td>17</td>
<td>UDP — User Datagram Protocol</td>
</tr>
<tr>
<td>41</td>
<td>IPv6 — next-generation TCP/IP</td>
</tr>
<tr>
<td>47</td>
<td>GRE — Generic Router Encapsulation (used by PPTP)</td>
</tr>
<tr>
<td>50</td>
<td>IPsec: ESP — Encapsulating Security Payload</td>
</tr>
<tr>
<td>51</td>
<td>IPsec: AH — Authentication Header</td>
</tr>
</tbody>
</table>
IPSec in AH Transport Mode

Original IPv4 Datagram

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ver</td>
<td>Version</td>
</tr>
<tr>
<td>hlen</td>
<td>Header Length (IPv4)</td>
</tr>
<tr>
<td>TOS</td>
<td>Type of Service</td>
</tr>
<tr>
<td>pkt len</td>
<td>Protocol Data Length</td>
</tr>
<tr>
<td>ID</td>
<td>Source IP Address</td>
</tr>
<tr>
<td>flgs</td>
<td>Destination IP Address</td>
</tr>
<tr>
<td>frag offset</td>
<td>TCP header (proto = 6)</td>
</tr>
<tr>
<td>TTL</td>
<td>Proto-TCP</td>
</tr>
<tr>
<td>header checksum</td>
<td></td>
</tr>
</tbody>
</table>

New IPv4 Datagram

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ver</td>
<td>Version</td>
</tr>
<tr>
<td>hlen</td>
<td>Header Length (IPv4)</td>
</tr>
<tr>
<td>TOS</td>
<td>Type of Service</td>
</tr>
<tr>
<td>pkt len</td>
<td>Protocol Data Length</td>
</tr>
<tr>
<td>ID</td>
<td>Source IP Address</td>
</tr>
<tr>
<td>flgs</td>
<td>Destination IP Address</td>
</tr>
<tr>
<td>frag offset</td>
<td>TCP header (proto = 6)</td>
</tr>
<tr>
<td>TTL</td>
<td>Proto-AH</td>
</tr>
<tr>
<td>header checksum</td>
<td></td>
</tr>
<tr>
<td>src IP address</td>
<td></td>
</tr>
<tr>
<td>dst IP address</td>
<td></td>
</tr>
<tr>
<td>next-TCP</td>
<td>AH len</td>
</tr>
<tr>
<td>AH len</td>
<td>Reserved</td>
</tr>
<tr>
<td>SPI</td>
<td>Security Parameters Index</td>
</tr>
<tr>
<td>Sequence Number</td>
<td></td>
</tr>
<tr>
<td>Authentication Data</td>
<td>(usually MD5 or SHA-1 hash)</td>
</tr>
</tbody>
</table>

Protected by AH Auth Data

TCP payload
IPSec in AH Tunnel Mode

Original IPv4 Datagram

New IPv4 Datagram

Protected by AH Auth Data

TCP Header + payload

TCP Header (proto = 6)
Summary
Network programming fundamentals

• Network sockets API: open(), connect(), send(), recv(), etc?
• How names are resolved to addresses in DNS
• End to end protocols
  – Move from host-to-host to process-to-process communication model
  – UDP – thin layer on top of IP
  – TCP – provide abstraction of reliable in-order byte stream on top of IP protocol
• Signals and timeouts
• Concurrency, multi-tasking, multiplexing
• Locking, mutexes, sharing state between threads/processes
Protocol design and analysis

- Framing vs. parsing
- Delimiter vs. length-value
- Server-side protocol handling
- Request for comments documents (RFCs)
- Deep dive on HTTP
  - Less treatment on REST
Data centers and CDNs

• Round-robin DNS vs load balancers. advantages and disadvantages of each
• Replication vs partitioning: advantages and disadvantages
• Terms: MTTR, MTBF, availability, yield, harvest, DQ principle
• Terms: top of rack switch, PUE, SPUE
• Tail latency vs. average latency
• Energy and power
RPCs

• Explain concept of 'idempotent'
• Maybe vs at least once vs at most once semantics; how to implement each of these?
• Thrift
• Role of stub compiler, RPC runtimes
• Discuss whether the following operations are idempotent:
  – Pressing a lift (elevator) request button
  – Writing data to an offset in a file
  – Appending data to the end of a file (assuming there are no other writers in the system)
Overlay networks, P2P, Chord

• Compare and contrast aspects of flooding queries, supernodes, vs structured. Tradeoffs--which is better for joining, leaving, advertising content, querying for content.

• Assume you have a Chord system of 5 nodes with identifiers 0, 1, 2, 6, and 7.
  – Draw the identifier circle and show which nodes the following keys will be assigned to 0, 4, 5, 8. You can assume the identifier space is 0-9.

• Each chord node must maintain routing state.
  – Describe exactly what routing state must be maintained at each node to ensure correct function. Show what this state would be for node 6. What is the expected lookup time of an object?
  – Describe what routing state must be maintained at each node to ensure fast lookup times. Show what this state would be for node 6. What is the expected lookup time of an object?
A word of thanks...