Topic: **networking fundamentals**

* Fundamental concepts
  + Encapsulation
  + Layered protocols (ISO-OSI model)
  + End-to-end communication (layer 3 and above) vs link-layer (layer 2) communication
  + …

NOTE: An ‘end’ or ‘end system’ could be either a client process or a server process.

* Client – server
* Server1 – server2
* A **network packet** has the packet header followed by the packet content (aka the payload).

|  |  |
| --- | --- |
| The packet **header** (containing meta-data):   * Source address = [yang@uhcl.edu](mailto:yang@uhcl.edu) * Destination address = [someone@uh.edu](mailto:someone@uh.edu) * Protocol id = 1234… (SMTP) * Message id = 3 * Length of the payload = 5K bytes * [Encryption algorithm=”AES256”] * [Hashing algorithm=”SHA256”] * … | The **payload**:  The actual packet content |

The two sides/ends/peers of this communication must understand each other, that is using the same language/protocol.

* Each of the packets belonging to the same message is routed independently through the network.
* Routing is typically a **dynamic** process, meaning that each of the routers will forward/route a packet based on the algorithm and the network condition at that time.
* Sample attacks?
  + The content of the payload may be stolen or modified.
  + The source address in the header may be modified, …

Q: How would we provide **integrity** to prevent unauthorized changes?

Ans: Use a mechanism that provide **data integrity**, such as message digests, message authentication codes, and digital signatures.

* **Packet Switching**:
  + A packet is delivered from one end, say A, to the other end, say B, by being processed by intermediate devices through the internetwork.
  + The network link is shared by multiple sources, each of which sends its packets over that link.

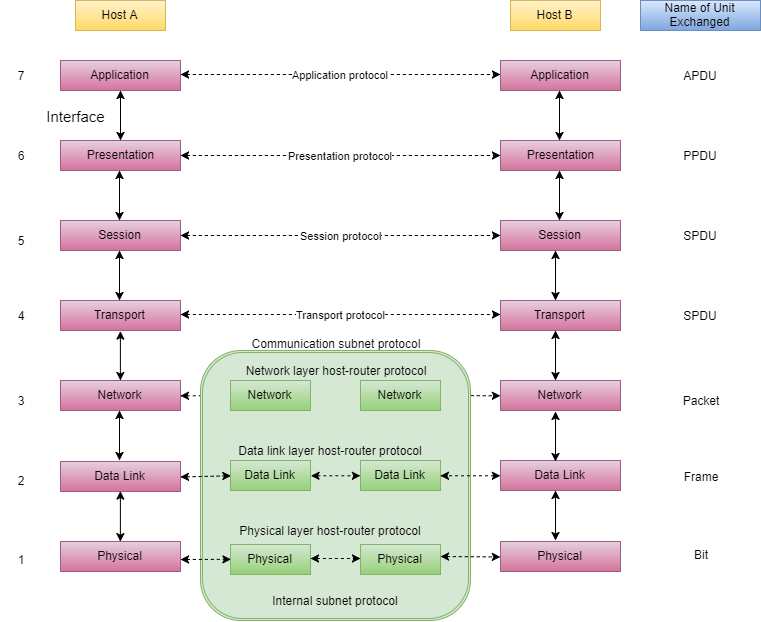
Trade-off analysis:

Pros? It’s cheaper to share the link by multiple parties.

Cons? Congestion, Packet Loss

NOTE: Two packets that belong to the same message are often delivered over different paths.

* Most networked systems are constructed as **layers**.
  + The networking functions on each of the hosts are divided into multiple layers (application layer, transport layer, network layer, data link layer, etc.)
  + Each layer implements a specific set of functions.
  + Communication at Layer 3 and above is **end-to-end**, meaning that there exists a pair of programs at both ends that communicate using the same protocol.
  + The program running at layer N uses services provided by programs running at layer N-1.
    - For example, HTTP (Layer 7) is built on top of TCP (Layer 4). That means, the HTTP program relies on the services provided by the TCP to reliably transit the application-layer packets to the other end.
    - The services provided by a program must be well-defined via the so-called APIs.



* **Encapsulation** occurs between two adjacent layers, when a layer n packet is handed down to the underlying layer, say layer n-1. The layer n packet becomes the payload of the layer n-1 packet.

This is the layer n packet.

|  |  |
| --- | --- |
| The packet **header**:   * Source address = [yang@uhcl.edu](mailto:yang@uhcl.edu) * Destination address = [someone@uh.edu](mailto:someone@uh.edu) * Protocol id = 1234… (SMTP) * Message id = 3 * Length of the payload = 5K bytes * … | The **payload**:  The actual message |

The layer n packet will be encapsulated into the layer n-1 packet.

|  |  |  |
| --- | --- | --- |
| **A new header for layer n-1:**  Source address = source IP  Destination address = destination IP  Source port# = …  Destination port# = …  Protocol id = … (TCP)  … | The packet **header**:   * Source address = [yang@uhcl.edu](mailto:yang@uhcl.edu) * Destination address = [someone@uh.edu](mailto:someone@uh.edu) * Protocol id = 1234… (SMTP) * Message id = 3 * Length of the payload = 5K bytes * … | The **payload**:  The actual message |

In comparison, **tunneling** is a process where a packet at layer n becomes the payload of another packet at layer n or above.

e.g., In VPN (Virtual Private Networks), an IP packet can become the payload of a IPsec packet.

**A** 🡪 (IP packet) 🡪 **A’s IPsec agent** 🡪 (IPsec packet) 🡪 … the Internet … 🡪 (IPsec packet) 🡪 **B’s IPsec agent** 🡪 (IP packet) 🡪 **B**

* The **OSI model** is used as a reference network model to explain how entities interact with each other in a computer network.
  + At each layer, the source and the destination are addressed differently.

A scenario: **John** is using his computer (computerJ) to send an email to Mary. **Mary** read her emails on computerM.

Assumptions:

1. John is using an email software based on the SMTP (Simple Message Transport Protocol).
2. John’s email severs is called ServerJ, and Mary’s email server is called serverM.
3. John’s default router/gateway is called rJ, and Mary’s is called rM.
4. John’s email address is [John@uhcl.edu](mailto:John@uhcl.edu), IP is jIP, and mac address is jMAC; Mary’s email address is [Mary@somewhere.org](mailto:Mary@somewhere.org), IP is mIP, and mac address is mMAC.

**Encapsulation** of network packets:

The packet (including header and payload) at certain layer becomes the payload of the packet at the layer below.

John

The message = “Hi. Mary. How are you doing?”

At Layer 7 (application layer), the packet looks like this:

|  |  |
| --- | --- |
| **SMTP** header  Sender: [John@uhcl.edu](mailto:John@uhcl.edu)  Receiver: [Mary@somewhere.org](mailto:Mary@somewhere.org)  Protocol id: SMTP id  Message id: …  Number of bytes in the payload: … | Data or payload  The message |

At layer 4 (transport layer), the layer 7 packet is encapsulated into a layer 4 packet.

|  |  |  |
| --- | --- | --- |
| **TCP** header  **Source IP: jIP**  **Source port#: …**  Destination IP: mIP  Destination port#: …  … | SMTP header  Sender: [John@uhcl.edu](mailto:John@uhcl.edu)  Receiver: [Mary@somewhere.org](mailto:Mary@somewhere.org)  Protocol id: SMTP id  Message id: …  Number of bytes in the payload: … | Data or payload  The message |

At layer 3 (network layer), the layer 4 packet becomes the payload of the layer 3 packet.

|  |  |  |  |
| --- | --- | --- | --- |
| **IP** header  Source IP: jIP  Destination IP: mIP | TCP header  Source IP: jIP  Source port#: …  Destination IP: mIP  Destination port#: …  … | SMTP header  Sender: [John@uhcl.edu](mailto:John@uhcl.edu)  Receiver: [Mary@somewhere.org](mailto:Mary@somewhere.org)  Protocol id: SMTP id  Message id: …  Number of bytes in the payload: … | Data or payload  The message |

Suppose John’s computer is connected to the Internet via wi-fi at the Data Link Layer (aka Layer 2).

NOTE: Layer 2 protocols are running between two directly-connected, wired or wireless, devices.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Wi-fi** header  Source MAC: jMAC  Destination MAC: the MAC address of **rJ**  Protocol id: 802.11 | IP header  Source IP: jIP  Destination IP: mIP | TCP header  Source IP: jIP  Source port#: …  Destination IP: mIP  Destination port#: …  … | SMTP header  Sender: [John@uhcl.edu](mailto:John@uhcl.edu)  Receiver: [Mary@somewhere.org](mailto:Mary@somewhere.org)  Protocol id: SMTP id  Message id: …  Number of bytes in the payload: … | Data or payload  The message |

**John** 🡪 packet 🡪 John’s router 🡪 … the Internet … 🡪 packet 🡪 **Mary’s router** 🡪 Mary

Understanding of the OSI model is essential.

We need to know that the attacker can not only attack the data itself (changing, stealing) but also information in those various headers (spoofing attacks).

In addition, the hacker can target any of the layers.

* **Link-layer communication** refers to communication at the Data Link Layer (part of layer 2 in the OSI model), which requires a physical connection between the two ends. Examples include Ethernet (802.3), Wi-Fi (802.11), …

**Let ========= represent a physical link between two devices.**

e.g., The **sender** ========= the sender’s router/gateway ========== the next hop/router …. ========== the receiver’s router ========= the **receiver**

* Examples of Tunneling protocols: An IP packet becomes the payload of an IPSec packet. Both IP and IPSec are at layer 3 (or the network layer).
* The processing of an outgoing packet at the sender side is top-down, and the processing of an incoming packet at the receiver end is bottom-up.
* The processing of an incoming packet at a **router** is first bottom-up (layer 1 to 2 to 3) and, after routing is done, the IP packet is re-encapsulated to become a layer 2 packet, and so on (layer 3 to 2 to 1). Thess two processes occur on every router.
* **End-to-end communication** occurs between programs at layer 3 and above, where the two ends/programs at the same layer speak the same language/protocol.

Application layer (layer 7): SMTP ------------------------------ SMTP

Transport layer (layer 4): TCP ----------------------------------------- TCP

Network layer (layer 3): IP --------------------------------------------------------- IP

* **End-to-end cryptography** means the two ends handle the cryptographic operations, whether it is encryption/decryption or hashing/validating, etc.
  + For example, if the sender encrypts its IP packets before sending them out, then the encrypted packets will not be decrypted until they reach the destination. During transmission, the packets are protected (for the sake of confidentiality).
  + Similarly, if the sender’s IP program generates a hash from the IP packet and send both to the destination, then it is the receiver’s IP program that will validates the received IP packet and the accompanying hash (for the sake of data integrity).