OCW: Unit 4

Week 2

Session 2: Internet and its Protocols (http, https, ftp) - Security related topics

Internet: Interface between the Process

A set of interconnected networks.

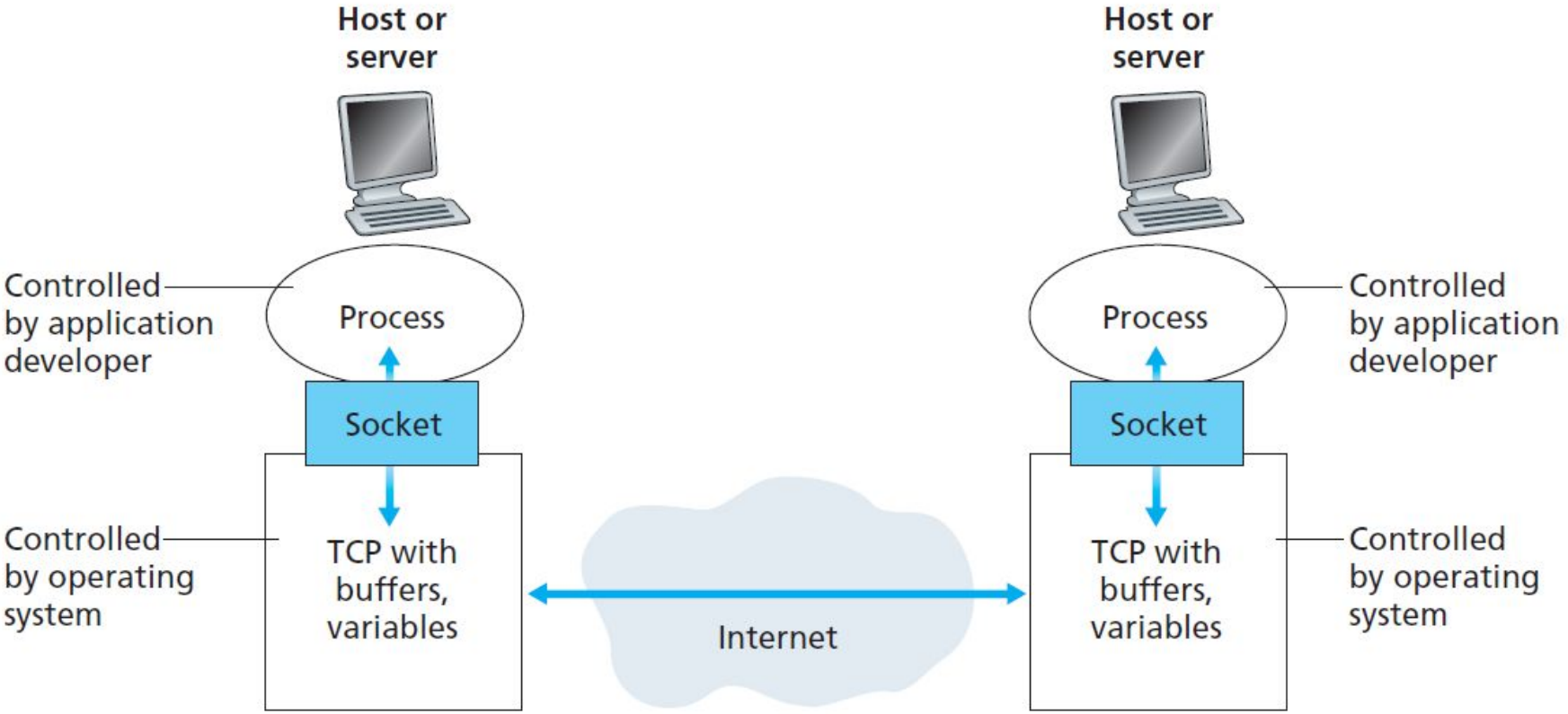
A process sends messages into, and receives messages from, the network through a software interface called a socket.

It is also referred to as the Application Programming Interface (API) between the application and the network, since the socket is the programming interface with which network applications are built.

The application developer has control of everything on the application-layer side of the socket but has little control of the transport-layer side of the socket.

The only control that the application developer has on the transport-layer side is (1) the choice of transport protocol and (2) perhaps the ability to fix a few transport-layer parameters such as maximum buffer and maximum segment sizes

Internet:Interface between the Process



Internet: Application Layer Protocols

Application	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP [RFC 5321]	TCP
Remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
File transfer	FTP [RFC 959]	TCP
Streaming multimedia	HTTP (e.g., YouTube)	TCP
Internet telephony	SIP [RFC 3261], RTP [RFC 3550], or proprietary (e.g., Skype)	UDP or TCP

Application Layer Protocols

An application-layer protocol is only one piece of a network application.

The Web is a client-server application that allows users to obtain documents from Web servers on demand.

The Web application consists of many components, including a standard for document formats (that is, HTML), Web browsers, Web servers, and an application-layer protocol.

The Web's application-layer protocol, HTTP, defines the format and sequence of messages exchanged between browser and Web server.

HTTP is only one piece of the Web application.

Application Layer Protocols

An Internet e-mail application also has many components,
including mail servers that house user mailboxes;
mail clients that allow users to read and create messages;
a standard for defining the structure of an e-mail message;
and application-layer protocols that define how messages are passed between servers,
how messages are passed between servers and mail clients,
and how the contents of message headers are to be interpreted.

The principal application-layer protocol for electronic mail is SMTP (Simple Mail Transfer Protocol) [RFC 5321].

Thus, e-mail's principal application-layer protocol, SMTP, is only one piece of the e-mail application.

HTTP:

The HyperText Transfer Protocol (HTTP), the Web's application-layer protocol, is at the heart of the Web.

It is defined in [RFC 1945] and [RFC 2616].

HTTP is implemented in two programs: a client program and a server program.

The client program and server program, executing on different end systems, talk to each other by exchanging HTTP messages.

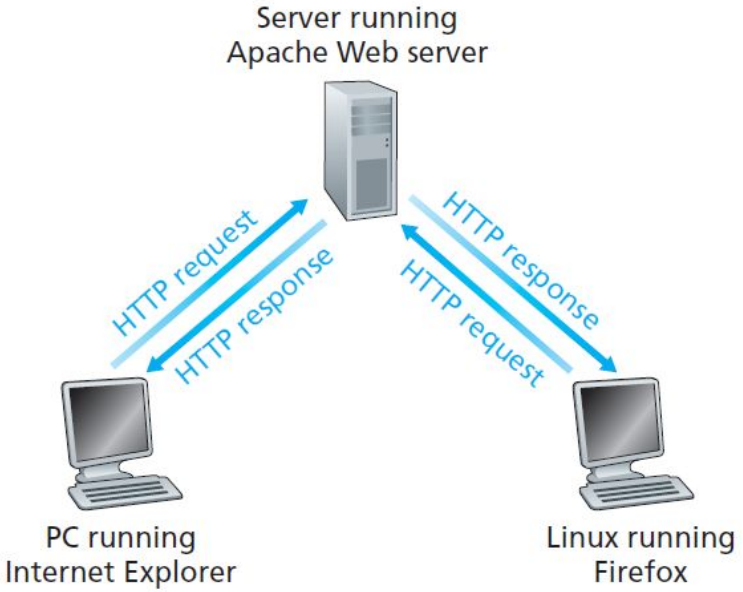
HTTP defines the structure of these messages and how the client and server exchange the messages.

HTTP

HTTP defines how Web clients request Web pages from Web servers and how servers transfer Web pages to clients.

When a user requests a Web page (for example, clicks on a hyperlink), the browser sends HTTP.

Request messages for the objects in the page to the server. The server receives the requests and responds with HTTP response messages that contain the objects.



HTTP Message Format

HTTP Request Message

Below we provide a typical HTTP request message:

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
```

HTTP Message Format

HTTP Response Message

Below we provide a typical HTTP response message. This response message could be the response to the example request message just discussed.

```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 09 Aug 2011 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 09 Aug 2011 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html
(data data data data data ...)
```

FTP: File Transfer Protocol

The user is sitting in front of one host (the local host) and wants to transfer files to or from a remote host.

In order for the user to access the remote account, the user must provide a user identification and a password.

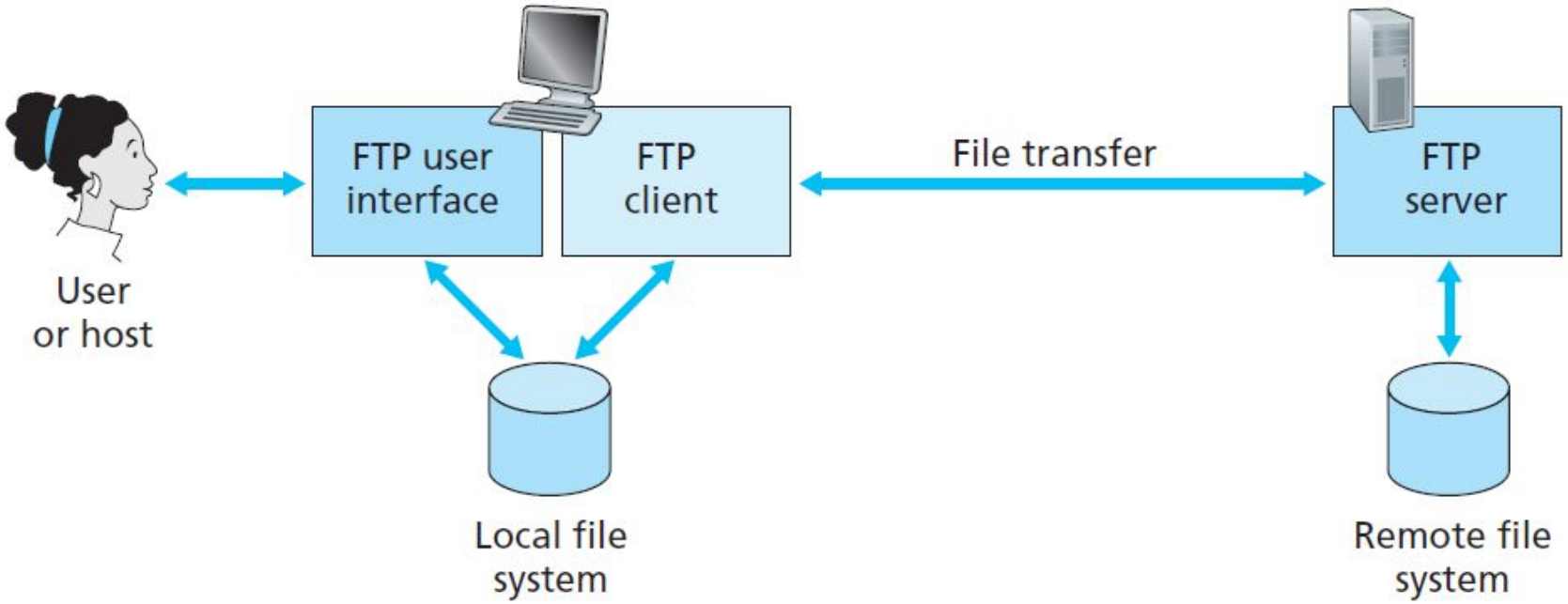
After providing this authorization information, the user can transfer files from the local file system to the remote file system and vice versa.

The user first provides the hostname of the remote host, causing the FTP client process in the local host to establish a TCP connection with the FTP server process in the remote host.

The user then provides the user identification and password, which are sent over the TCP connection as part of FTP commands.

Once the server has authorized the user, the user copies one or more files stored in the local file system into the remote file system (or vice versa).

FTP: File Transfer Protocol



HTTP Vs FTP

Common Characteristic: they both run on top of TCP.

Difference: is that FTP uses two parallel TCP connections to transfer a file, a control connection and a data connection.

The control connection is used for sending control information between the two hosts—information such as user identification, password, commands to change remote directory, and commands to "put" and "get" files.

The data connection is used to actually send a file.

Because FTP uses a separate control connection, FTP is said to send its control information out-of-band.

HTTP sends request and response header lines into the same TCP connection that carries the transferred file itself.

HTTP is said to send its control information in-band.

FTP: Commands and Replies

USER username: Used to send the user identification to the server.

PASS password: Used to send the user password to the server.

LIST: Used to ask the server to send back a list of all the files in the current remote directory. The list of files is sent over a (new and non-persistent) data connection rather than the control TCP connection.

RETR filename: Used to retrieve (that is, get) a file from the current directory of the remote host. This command causes the remote host to initiate a data connection and to send the requested file over the data connection.

STOR filename: Used to store (that is, put) a file into the current directory of the remote host.

FTP: Commands and Replies

The replies are three-digit numbers, with an optional message following the number.

Some typical replies, along with their possible messages, are as follows:

- 331 Username OK, password required
- 125 Data connection already open; transfer starting
- 425 Can't open data connection
- 452 Error writing file

HTTPS

A plain HTTP connection can be easily monitored, modified, and impersonated.

When properly configured, an HTTPS connection guarantees three things:

- **Confidentiality.** The visitor's connection is encrypted, obscuring URLs, cookies, and other sensitive metadata.
- **Authenticity.** The visitor is talking to the "real" website, and not to an impersonator or through a person-in-the-middle.
- **Integrity.** The data sent between the visitor and the website has not been tampered with or modified.

HTTPS encrypts nearly all information sent between a client and a web service.

HTTPS uses **Transport Layer Security (TLS)** protocol or its predecessor **Secure Sockets Layer (SSL)** for encryption.

Network security

- field of network security:
  - how bad guys can attack computer networks
  - how we can defend networks against attacks
  - how to design architectures that are immune to attacks
- Internet not originally designed with (much) security in mind
  - *original vision:* "a group of mutually trusting users attached to a transparent network" ☺
  - Internet protocol designers playing "catch-up"
  - security considerations in all layers!

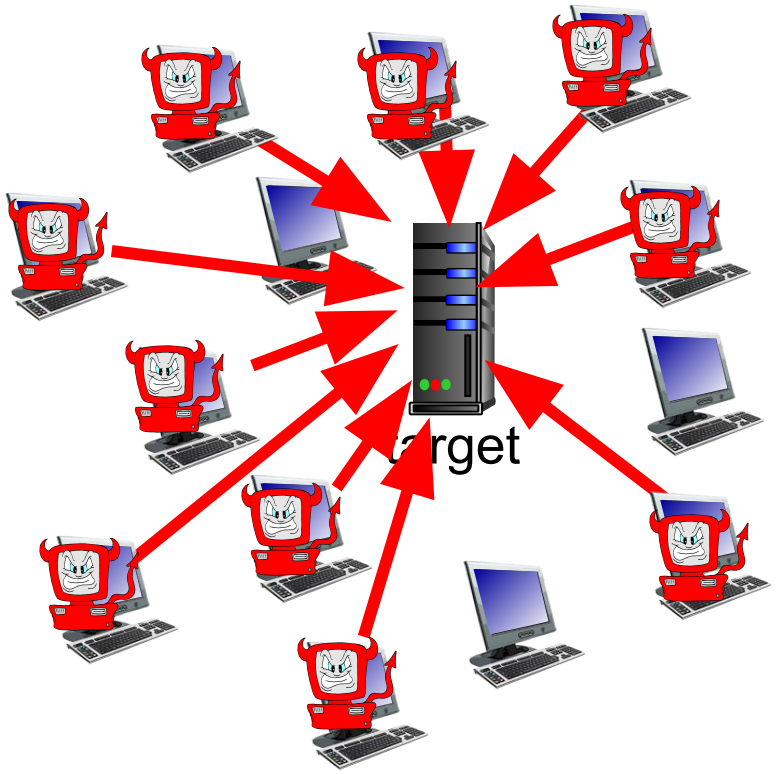
Bad guys: put malware into hosts via Internet

- malware can get in host from:
  - *virus:* self-replicating infection by receiving/executing object (e.g., e-mail attachment)
  - *worm:* self-replicating infection by passively receiving object that gets itself executed
- spyware malware can record keystrokes, web sites visited, upload info to collection site
- infected host can be enrolled in botnet, used for spam. DDoS attacks

Bad guys: attack server, network infrastructure

Denial of Service (DoS): attackers make resources (server, bandwidth) unavailable to legitimate traffic by overwhelming resource with bogus traffic

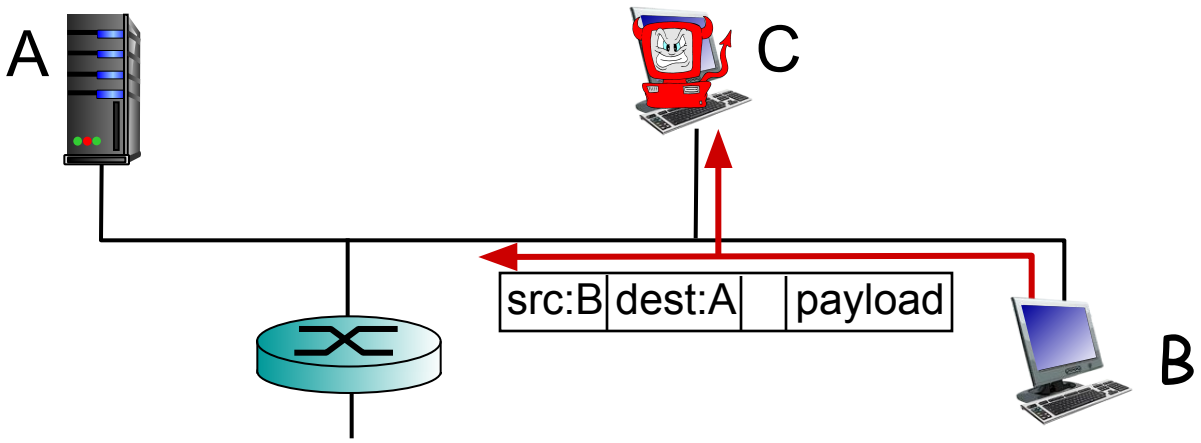
1. select target
2. break into hosts around the network (see botnet)
3. send packets to target from compromised hosts



Bad guys can sniff packets

packet "sniffing":

- broadcast media (shared ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by



- wireshark software used for end-of-chapter labs is a (free) packet-sniffer

Bad guys can use fake addresses

IP spoofing: send packet with false source address