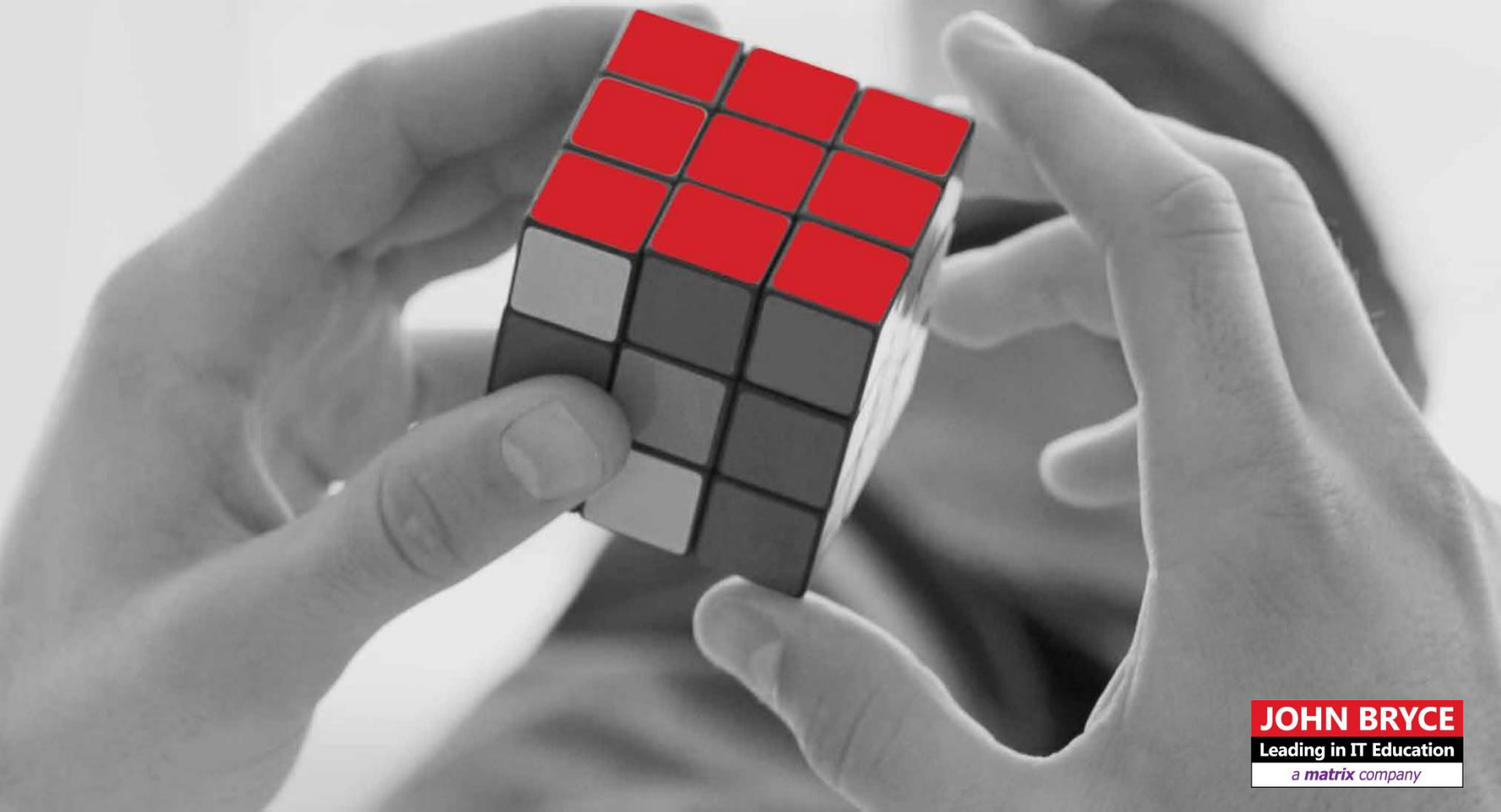


Introduction To The Internet



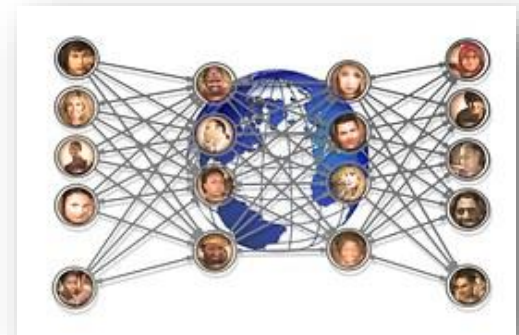
Objectives

By the end of this session

You will gain basic understanding of how Internet works

What is the Internet?

- The internet is a global system of computer networks (Servers) that serves users (clients) with information
- The computers are connected and communicating via cables, wireless and more...
- WWW - World Wide Web
- HTTP - common communication protocol



What is the Internet?

The World Wide Web Consortium (W3C) is the main international standards organization for the World Wide Web (WWW)

W3C Markup Validation Service
<http://validator.w3.org/>

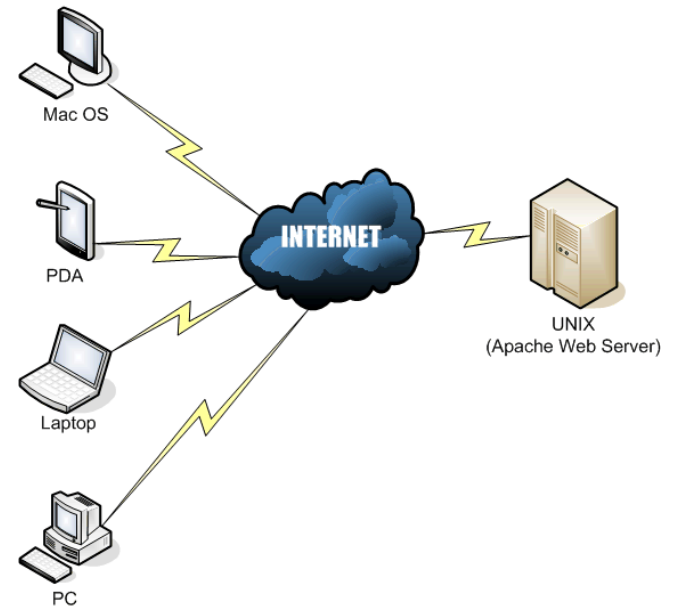
What is the Internet?

Information transferred can be any kind of resource:

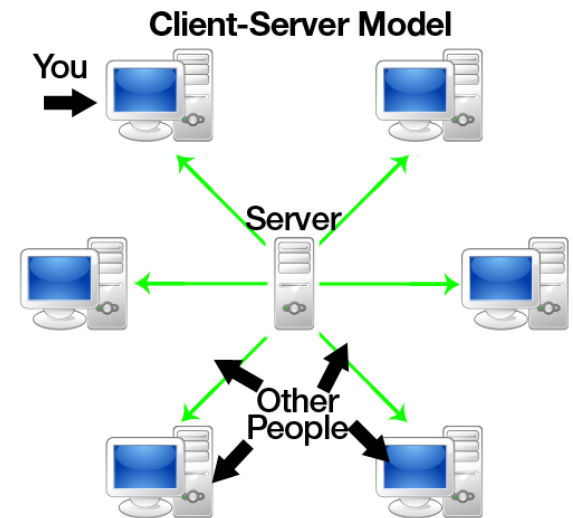
- Websites (HTML Pages)
- Images
- Video/ Audio
- More

- A software component installed on a computer (server) designed to receive a request and serve a response to the caller
- Web Server Softwares:
 - APACHE
 - IIS
 - nginx

- The main purpose of a Web Server is to send web pages (websites are built from web pages)
- All data is stored in the Web Server
- A web page usually is written in HTML language



- A Client is also a computer. This computer receives the information sent from the Web Server
- A Client must have a software that can send requests to the Web Server and receive response
- This software known as Web Browser



- A Browser is a software that gives users the ability to communicate through the internet and get information from the Web Servers
- Major Web Browsers:
 - Google Chrome
 - Mozilla Firefox
 - Internet Explorer
 - Opera
 - Safari



- Computers need to “speak the same language” in order to communicate
- Computers communicate over the internet is with a protocol called HTTP
- Server and Client both needs to understand HTTP
- This is possible with the Web Server and Client Softwares installed on each computer

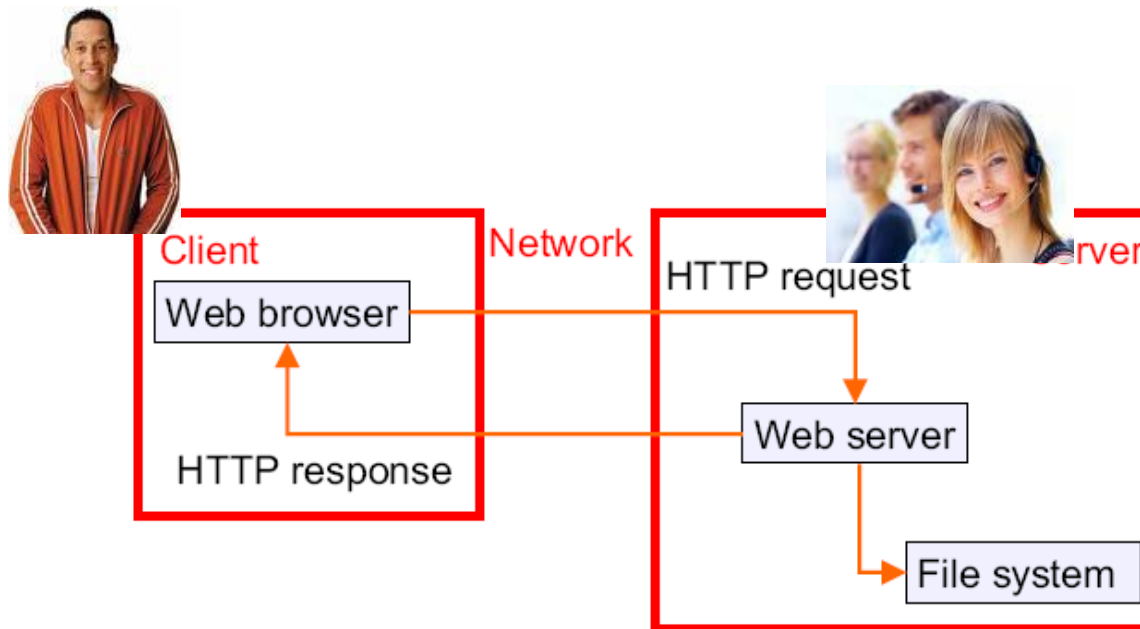
e.g: Apache Server and Chrom Browser both know to communicate with HTTP

HyperText Transfer Protocol

- HTTP is the way computers communicate in the World Wide Web (www)
- HTTP is a request-response protocol in the client-server computing model
 - Client sends a HTTP Requests.
 - Server sends a HTTP Response.

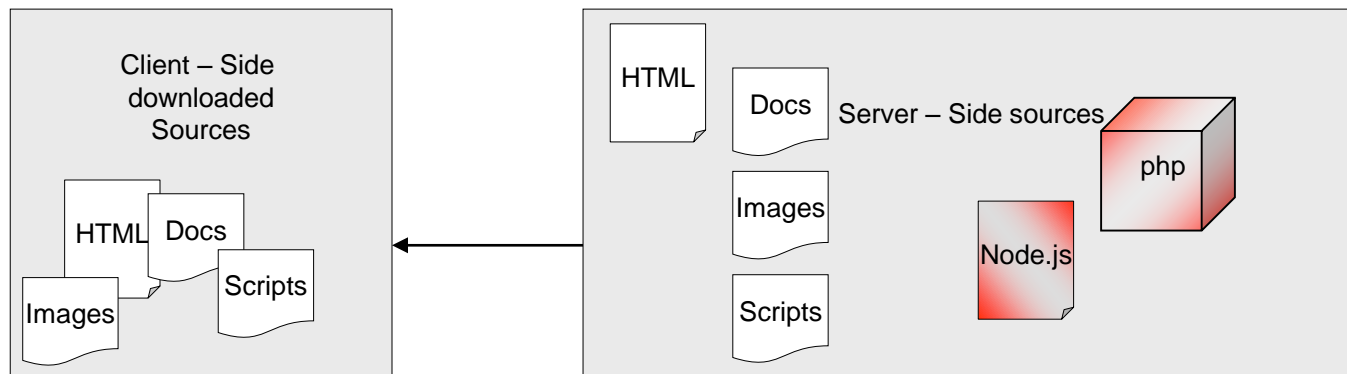


Web application cycle



Web Application includes:

- Dynamic content [PHP / Node.js]
- Static content [HTML, documents, images etc.]
- Client-side scripts [Java-scripts, flash apps.]
- Dynamic contents



Internet Network is built on layers:

UDP / TCP communication

- UDP

One to many broadcast

Small data packets

Non-indexed

- TCP

One to One / Peer to Peer

Complex and large data can be transmitted

Message can be broken to small parts

Each part has index

Internet Network is built on layers:

IP

- Internet address
- Unique address each connected device has
- IPv4 255.255.255.255
- IPv6 255.255.255.255.255.255
- Servers uses dedicated static IPs
- Clients usually obtain dynamic IP every time they connect
- Integration occurs over TCP/IP

Since

- each language has its own data structures
- each Operating Systems may host different browsers

There is no single binary platform to communicate

So, the standard was set in a TEXT format

- Every language supports texts
- Texts are used in Unicode format
- Unicode
 - a combination of 16 bits per char
 - supports most spoken languages and signs

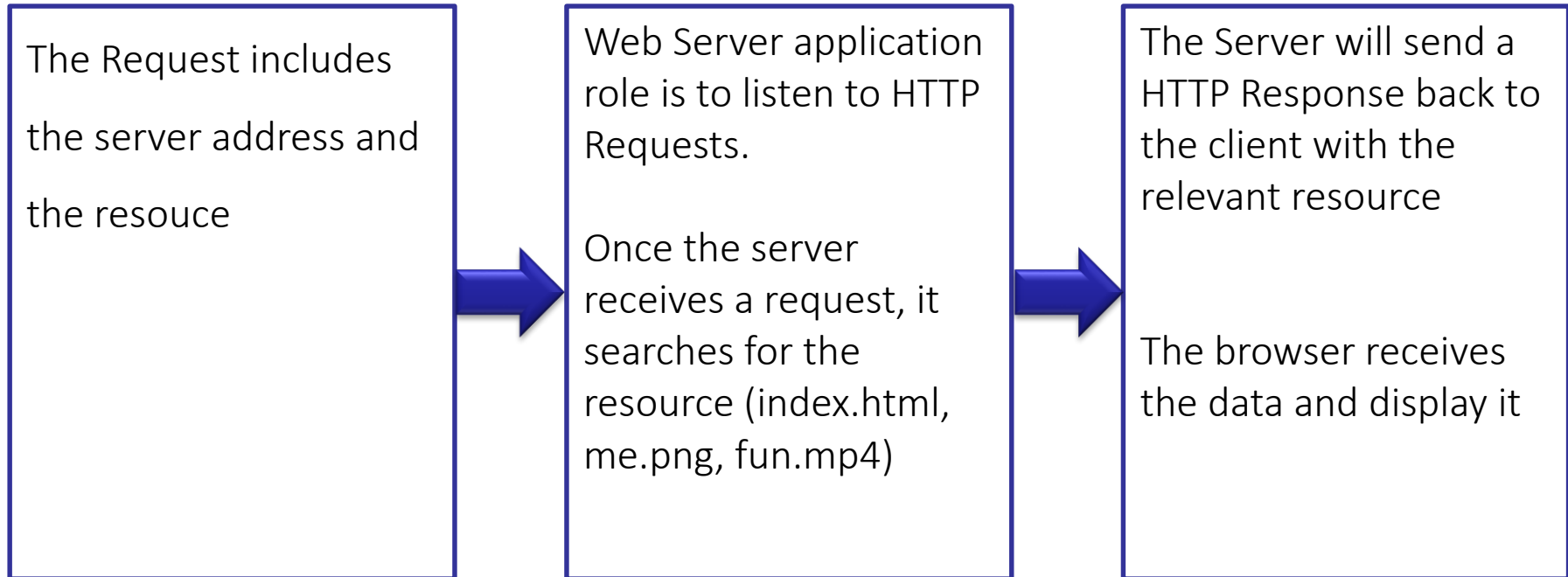
How it all works?

When you (Client) write a website address in the browser, the browser sends a HTTP Request to the server

e.g. for Google website, write in the browser the address <http://www.google.com>

The browser sends a HTTP Request to Google Web Server

How it all works?



http://www.

Internet Protocol

- Web site address has unique identifier, which allows the computers to communicate. This unique identifier is the site address

*e.g., <http://www.google.com>,
<http://www.ynet.co.il>*

- behind this address there is the actual unique identifier called “IP Address”

Internet Protocol

- IP address is a unique numerical label assigned to every device in the cyberspace
- Clients, Servers and other devices have there own IP address

<i>e.g., Google IP:</i>	<i>173.194.78.104</i>
-------------------------	-----------------------

173.194.78.104

- IP address has 4 sections separated by dot, called IPV4. Internet IPs are undergoing a change to support 6 sections to allow more addresses, or IPV6
- Each section range can represent 0-255 decimal
- The address “http://www.google.com”, is automatically converted by the browser to IP address

Every website has a Domain Name

<i>e.g.,</i>	<i>Google Domain name:</i>	<i>www.google.com</i>
	<i>Ynet Domain name:</i>	<i>www.ynet.co.il</i>

Domain name is the server address. The server contains different files of the website. Now, you should ask for the specific resource

<i>e.g.,</i>	<i>homepage.html, contact_us.html</i>
--------------	---------------------------------------

Uniform Resource Identifier

- URI is a string of characters used to identify a web resource
- Syntax: `scheme://domain:port/path?query_string`

E.g., `http://www.google.com:80/index.html?q=uri`



scheme://domain:port/path?query_string

scheme - protocol (*http, https, ftp*)

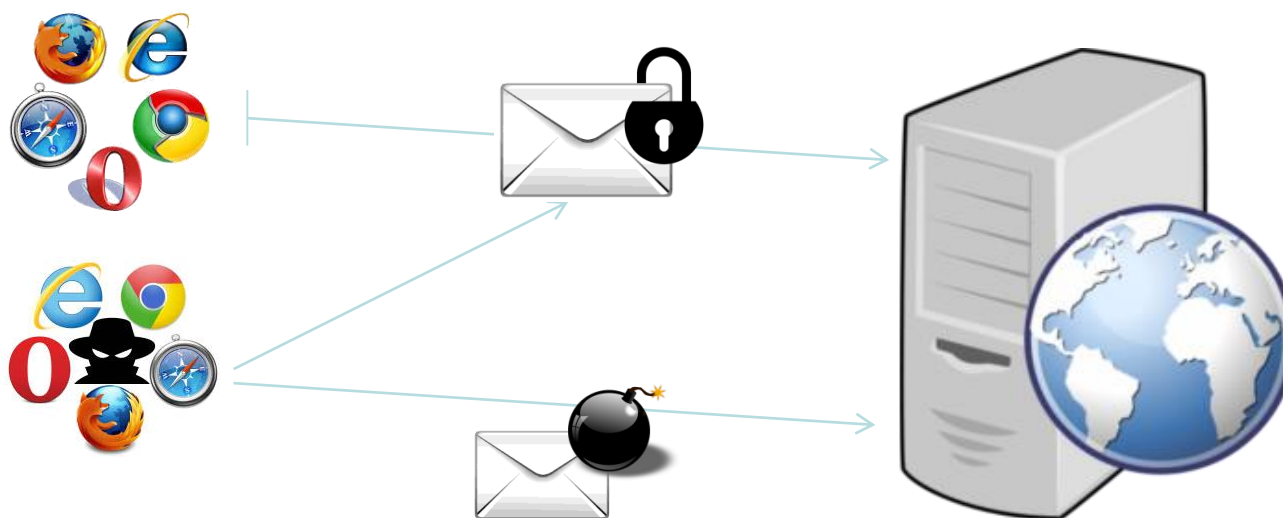
domain - server (*www.google.com*)

port - default port is 80. The browser don't show it. The server listen to HTTP requests from port 80.

path - folder path on the server and resource name
(*catalog/music/headphones.html*)

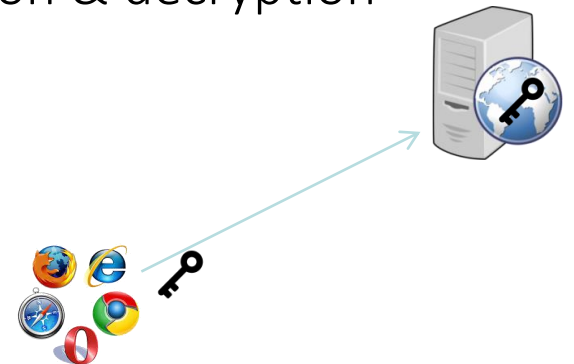
query_string - data sent to the server from the client

Insecure Communication

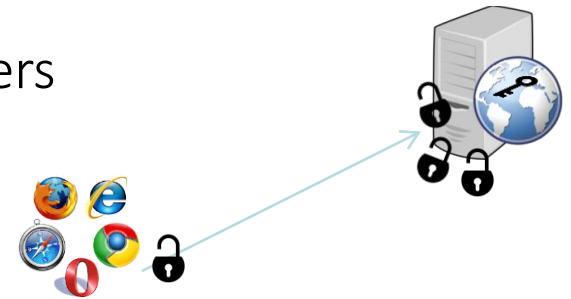


- When authorization and other sensitive data submit are done over non-encrypted protocol
- Use secured socket layer - SSL
- Use it also for intra domain communication
 - Clusters, remote management, DB & Messaging servers...

- SSL / TLS (Transport Layer Security)
- Secured HTTP is HTTPS
- How HTTPS works ? Few words on encryption:
 - Symmetric encryption
 - Both sides use the same key to encryption & decryption



- Asymmetric encryption –
 - Each side uses a different key
 - One side publishes a public key and keeps a private key
 - Client obtains a public key and use it to encrypt his message
 - Once the client encrypted the message – only owner of private key can view it
 - Server uses private key to decrypt it
 - Public key is the result of two huge primer numbers multiplied
 - Private key contains the two primer numbers



- Handshake
 - Client sends strongest supported encryption & hashing algorithms
 - Server sends back his algorithms + a public key
- Client approves public key and uses it to encrypt a symmetric key
 - The symmetric key will be used for the rest of the secured communication
- Server uses his private key to obtain the symmetric key
- Deciding on a symmetric key is done with asymmetric encryption – slower
- Messages encrypted with symmetric encryption – faster

Wait,

- When we exchange secrets and decide on a symmetric key – how do we know for sure server identity ??
- What if the page looks like what we expect but actually sends our secrets somewhere else ??

Certified Authorities (CA)

- Server digital certificate can be issued by well known authorities
- CAs are trusted organizations
- Provides digital signature for public keys
- Public keys can be signed with CA certificates

Trusted public key

Un-trusted public key

	Issuer	Usage	Market share
1	Comodo	5.3%	35.6%
2	Symantec	4.7%	31.9%
3	GoDaddy	2.1%	14.0%
4	GlobalSign	1.5%	10.0%
5	DigiCert	0.4%	2.7%

Wiki, May 2015

File Transfer Protocol

- Used for uploading and downloading files from servers
- FTP supports login
- FTP is done over TCP/IP
- Uses very long request time and timeouts
- FTP server is an internet connected application that stores files
- FTP client can be your browser

The screenshot shows an FTP client interface. A dialog box titled "Authentication Required" is open, asking for a username and password for the server ftp://ftp.johnbryce.co.il:21. The dialog has fields for "User Name:" and "Password:" and buttons for "Log In" and "Cancel".

The main window shows the directory listing for ftp://ftp.johnbryce.co.il/SANDISK/. The address bar displays the URL. The directory listing table is as follows:

Name	Size	Date Modified
[parent directory]		
sandisk_020815.zip	22.0 MB	8/3/15, 8:41:00 AM
sandisk_020815_LOCAL_SAVE.zip	22.0 MB	8/3/15, 8:41:00 AM

eXtensible Markup Language

- HTML for applications. Describes plain data
- Present and future devices will consume mostly data – not view
 - We can do much more with this
 - than we can do with that

```
<table>
  <tr>
    <td> David </td>
    <td> 20 </td>
  </tr>
  ....
</table>
```

```
<people>
  <person>
    <name> David </name>
    <age> 20 </age>
  </person>
  ....
</people>
```

eXtensible Markup Language

- XML, as any markup language may hold data in:
 - **Elements <tags>** - elements are hierarchical and may repeat
 - **Attributes** – unique in elements <tag id="...">
 - **CDATA sections** – holds text which is not parsed at all <![CDATA[.....]]>
 - **Comments** <!-- -->

XML for transferring data

- Well formed
 - Set of basic syntax rules
 - Including:
 - Processing instruction <? xml ?>*
 - Closing tags*
 - Attribute values inside quotes*
 - Case sensitive*
 - Correct element nesting...*
 - Part of W3C XML standard
 - XML parsers must not parse any non well-formed data
 - Saves checks and manipulations for small & tiny devices
 - For browsers & micro-browsers - XHTML

XML for transferring data

- Validation and types
 - XML structure is described via XSD (Schema)
 - W3C standard
 - XSD Schema defines:
 - Element name & content
 - Attributes
 - Simple and complex types
 - Since XSD defines primitives (xsd:integer, xsd:date....) – objects can be described as well

XML validation

Schema example:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="People">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Person" type="PersonType" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="PersonType">
    <xsd:sequence>
      <xsd:element name="Name" type="xsd:string"/>
      <xsd:element name="Age" type="AgeType"/>
      <xsd:element name="BirthDate" type="xsd:date" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="gender" type="GenderType" use="required"/>
  </xsd:complexType>
  <xsd:simpleType name="AgeType">
    <xsd:restriction base="xsd:nonNegativeInteger">
      <xsd:minInclusive value="0"/>
      <xsd:maxInclusive value="120"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="GenderType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="M"/>
      <xsd:enumeration value="F"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>
```

XML validation

Schema example:

```
<?xml version="1.0"?>
<People xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="PeopleSchema.xsd">
  <Person gender="M">
    <Name>Bill</Name>
    <Age>35</Age>
    <BirthDate>1984-04-13</BirthDate>
  </Person>
  <Person gender="F">
    <Name>Dana</Name>
    <Age>47</Age>
    <BirthDate>1961-11-03</BirthDate>
  </Person>
  <Person gender="F">
    <Name>Amy</Name>
    <Age>23</Age>
    <BirthDate>1991-04-15</BirthDate>
  </Person>
  <Person gender="M">
    <Name>David</Name>
    <Age>13</Age>
    <BirthDate>2000-07-02</BirthDate>
  </Person>
</People>
```

Summary

- Servers – Clients
- Browsers
- Protocols