

Prepared by Dr. Mohammed Anbar

Network Security CYB 603
Web Security (part one)

Lecture 8



Learning objectives

- Explain basic web terminologies,
- Understand HTTP and HTTPS protocols,
- Understand Web Security and web security threats,
- Explain technologies for Web Security,
- Understand Open Web Application Security Project,
- Explain attacks on web applications.

Website

Static Website

Dynamic Website

Web server

URL

HTTP

HTTPS

- Website is a collection of related web pages that may contain text, images, audio, and video. The first page of a website is called a home page.
- Each website has a specific internet address (URL) that you need to enter in your browser to access the website.
- Website is hosted on one or more servers and can be accessed by visiting its homepage using a computer/laptop or mobile phone.
- A website is managed by its owner who can be an individual, company, or organization.

Static Website

Dynamic Website

Static website

- Static website is the basic type of website that is easy to create. You don't need knowledge of web programming and database design to create a static website. Its web pages are coded in HTML.
- The codes are fixed for each page so the information contained in the page does not change and it looks like a printed page.

Dynamic website

- Dynamic website is a collection of dynamic web pages whose content changes dynamically. It accesses content from a database or Content Management System (CMS).
- Therefore, when you alter or update the content of the database, the content of the website is also altered or updated.
- Dynamic website uses client-side scripting or server-side scripting, or both to generate dynamic content.

Static Website vs Dynamic Website

Static Website	Dynamic Website
Prebuilt content is same every time the page is loaded.	Content is generated quickly and changes regularly.
It uses HTML code for developing a website.	It uses the server side languages such as PHP, SERVLET, JSP, and ASP.NET etc. for developing a website.
It sends exactly the same response for every request.	It may generate different HTML for each of the request.

Static Website vs Dynamic Website (cont.)

The content is only changed when someone publishes and updates the file (sends it to the web server).

Flexibility is the main advantage of static website.

The page contains "server-side" code which allows the server to generate the unique content when the page is loaded.

Content Management System (CMS) is the main advantage of dynamic website

MIME (Multipurpose internet Mail Extension)

The commonly used content types are given below:

- text/html
- text/plain
- application/msword
- application/vnd.ms-excel
- application/jar
- application/octet-stream
- application/x-zip
- images/jpeg
- images/png
- images/gif

- A web server is computer software and the underlying hardware that accepts requests via HTTP (the network protocol created to distribute web content) or its secure variant HTTPS.
- A user agent, commonly a web browser or web crawler, initiates communication by making a request for a web page or other resource using HTTP/HTTPS, and the server responds with the content of that resource or an error message.

- A resource sent from a web server can be a pre-existing file (static content) available to the web server, or it can be generated at the time of the request (dynamic content) by another program that communicates with the server software.
- A web server can also accept, and store resources sent from the user agent if configured to do so.
- The hardware used to run a web server can vary according to the volume of requests that it needs to handle.

Chapter

07

Web server (cont.)



Apache HTTP Server



Nginx



IIS



lighttpd



Apache Tomcat



Application server



LiteSpeed Web Server



Oracle iPlanet Web Server



NCSA HTTPd



IBM HTTP Server



Apache Software Foundation



Google Web Server

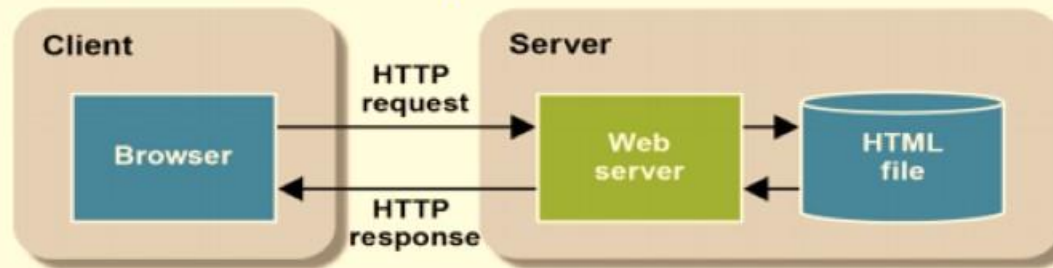


Netcraft

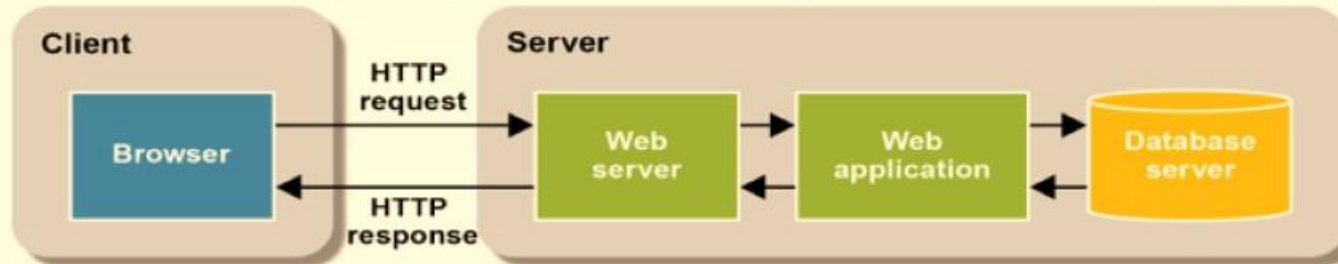


How do web server process web pages ?

How a web server processes static web pages



How a web server processes dynamic web pages



Uniform Resource Locator (URL)

- The term Uniform Resource Locator (URL) is a specific type of Uniform Resource Identifier (URI), although many people use the two terms interchangeably.
- A URL is used to uniquely identify a resource and may be denoted using a syntax such as `http://host.domain/path.html`, which indicates the protocol being used (in this case, HTTP), along with the hostname or IP address, and a path that points to a particular file stored on that server.
- URLs occur most commonly to reference web pages (HTTP) but are also used for file transfer (FTP), email (mailto), database access (JDBC), and many other applications.

http://www.example.com/index.html

- The initial label like HTTPS or HTTP is the scheme, which names the protocol by which a document can be retrieved.
- Following the colon and two slashes:// comes the hostname and optional port number.
- Finally, a path selects one document out of all the documents that might be available on a service.

HTTP (Hyper Text Transfer Protocol)

- HTTP is an application-layer protocol used to establish communication between the client and server.
- HTTP designed within the framework of the Internet protocol suite. Its definition presumes an underlying and reliable transport layer protocol, thus Transmission Control Protocol (TCP) is commonly used. However, HTTP can be adapted to use unreliable protocols such as the User Datagram Protocol (UDP).
- HTTP is used to deliver the data like image files, query results, HTML files, etc on the World Wide Web (WWW) with the default port being TCP 80.

- HTTP messages are how data is exchanged between a server and a client.
- There are two types of messages:
 - HTTP requests sent by the client to trigger an action on the server,
 - HTTP responses which are the answer from the server.

HTTP Request Message Structure

```
GET /index.html HTTP/1.1\r\n
```

← *A start-line*

```
Host: www-net.cs.umass.edu\r\n
```

```
User-Agent: Firefox/3.6.10\r\n
```

```
Accept: text/html,application/xhtml+xml\r\n
```

```
Accept-Language: en-us,en;q=0.5\r\n
```

```
Accept-Encoding: gzip,deflate\r\n
```

```
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\n
```

```
Keep-Alive: 115\r\n
```

```
Connection: keep-alive\r\n
```

← *HTTP headers*

```
data data data data data ...
```

← *Body*

Start line	HTTP method, request target, HTTP version
Headers	User-Agent, Accept, Accept-Encoding
Body	Not all requests have one: requests fetching resources, like GET, HEAD, DELETE, or OPTIONS, usually don't need one. Some requests send data to the server in order to update it: as often the case with POST requests (containing HTML form data).

HTTP/1.0:

- GET
- POST
- HEAD
 - asks server to leave requested object out of response

HTTP/1.1:

- GET, POST, HEAD
- PUT
 - uploads file in entity body to path specified in URL field
- DELETE
 - deletes file specified in the URL field

POST method:

- web page usually includes form input
- input is uploaded to the server in the entity body

Name	Value
Name	
Sex	☐ Male ☒ Female
Eye color	green
Check all that apply	☐ Over 6 feet tall ☐ Over 200 pounds
Describe your athletic ability: 	
Enter my information	

GET method:

URL uses GET method

www.example.com/check.php?id=1

GET	POST
In case of Get request, only limited amount of data can be sent because data is sent in header.	In case of post request, large amount of data can be sent because data is sent in body.
Get request is not secured because data is exposed in URL bar.	Post request is secured because data is not exposed in the URL bar.
Get request can be bookmarked.	Post request cannot be bookmarked.
Get request is more efficient and used more than Post.	Post request is less efficient and used less than get.

HTTP Response Message Structure

```
HTTP/1.1 200 OK\r\n
```

← A start-line

```
Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n
```

```
Server: Apache/2.0.52 (CentOS)\r\n
```

```
Last-Modified: Tue, 30 Oct 2007 17:00:02 GMT\r\n
```

```
ETag: "17dc6-a5c-bf716880"\r\n
```

```
Accept-Ranges: bytes\r\n
```

```
Content-Length: 2652\r\n
```

```
Keep-Alive: timeout=10, max=100\r\n
```

```
Connection: Keep-Alive\r\n
```

```
Content-Type: text/html; charset=ISO-8859-
```

← HTTP headers

```
data data data data data ...
```

← Body

Status line	protocol version, status code, status text
Headers	Content-Type , Content-Encoding
Body	Not all responses have one: responses with a status code that sufficiently answers the request without the need for a corresponding payload.

❖ status code appears in 1st line in server-to-client response message.

❖ some sample codes:

200 OK

- request succeeded.

301 Moved Permanently

- requested object moved to a new location

400 Bad Request

- Request msg not understood by server

404 Not Found

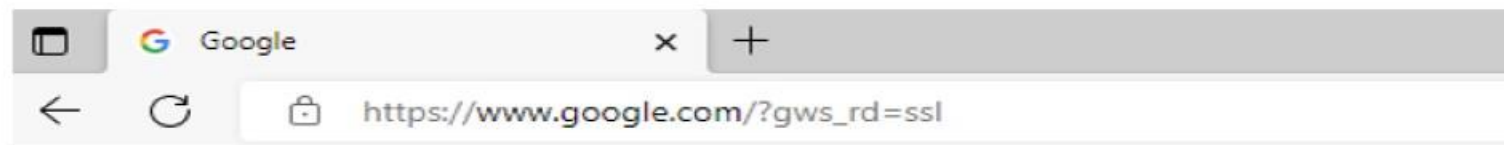
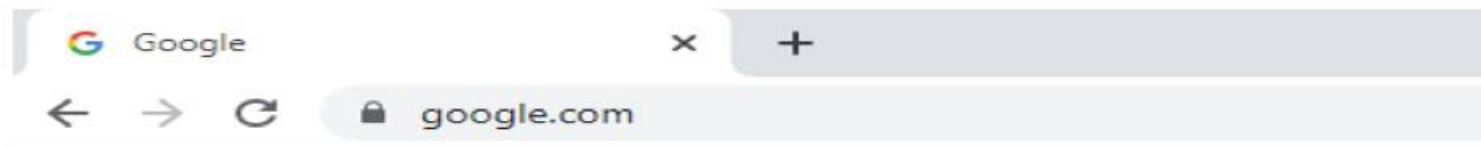
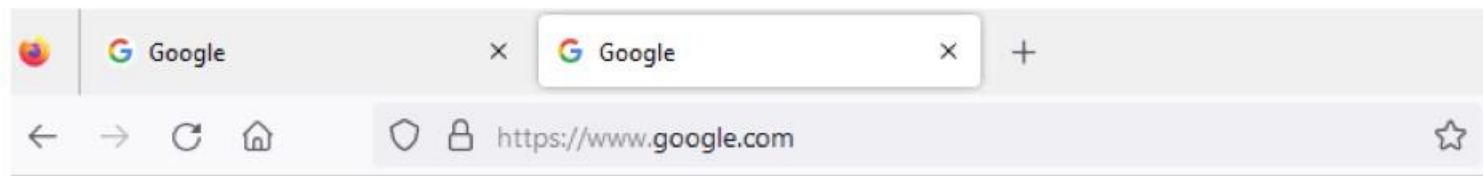
- Requested document is not found on this server

505 HTTP Version Not Supported

Hypertext transfer protocol secure (HTTPS)

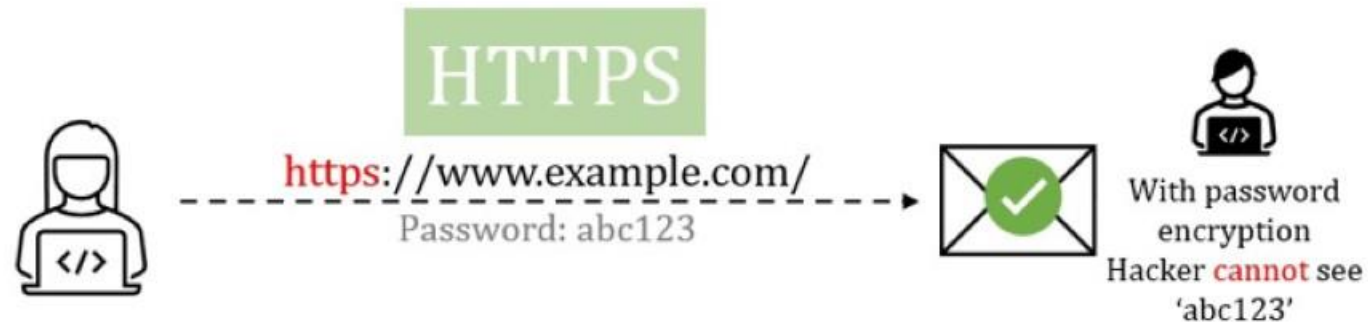
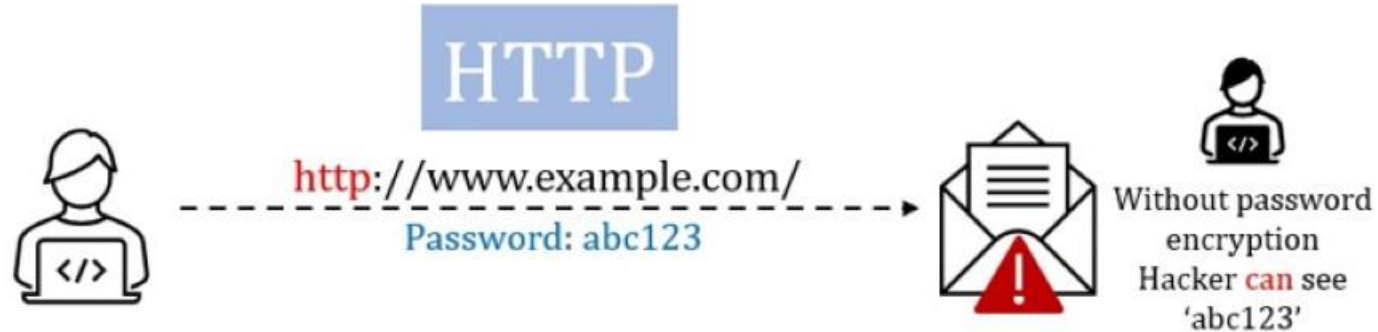
- HTTPS is the secure version of HTTP, which is the primary protocol used to send data between a web browser and a website.
- HTTPS is encrypted in order to increase the security of data transfer. This is particularly important when users transmit sensitive data, such as by logging into a bank account, email service, or health insurance provider.
- Any website, especially those that require login credentials, should use HTTPS

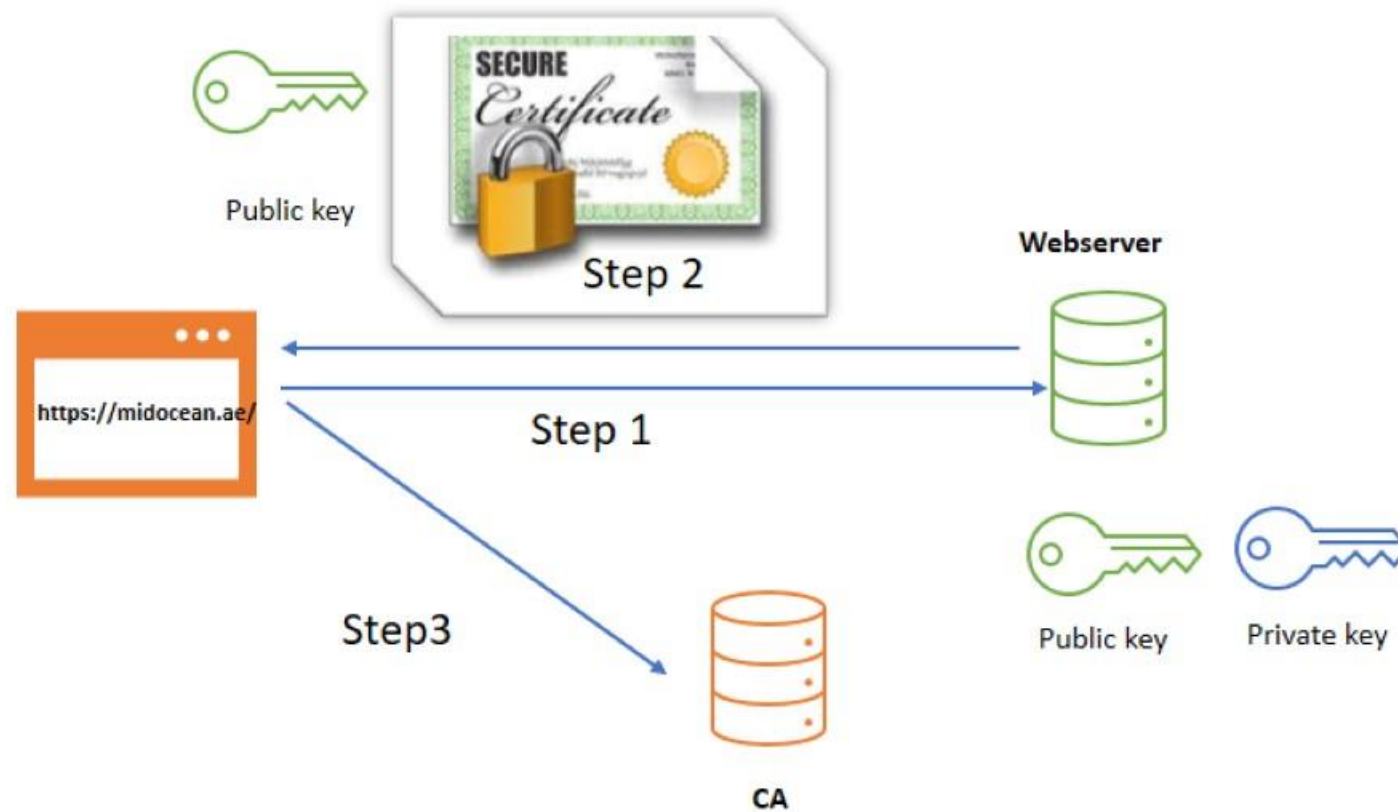
- Using web browsers like Chrome, websites that do not employ HTTPS are marked differently from those that do.
- If you want to know whether a website is secure, look for a padlock in the URL bar. Google Chrome and other browsers mark all non-HTTPS websites as insecure, demonstrating how seriously web browsers take HTTPS.



- HTTPS uses an encryption protocol to encrypt communications. The protocol is called Transport Layer Security (TLS), although formerly it was known as Secure Sockets Layer (SSL).
- This protocol secures communications by using what's known as an asymmetric public key infrastructure.
- This type of security system uses two different keys to encrypt communications between two parties

- The private key - this key is controlled by the owner of a website and it's kept. This key lives on a web server and is used to decrypt information encrypted by the public key.
- The public key - this key is available to everyone who wants to interact with the server in a way that's secure. Information that's encrypted by the public key can only be decrypted by the private key.





1. The browser requests secure pages HTTPS from the google web server
2. The server sends its certificate which is digitally signed by a third party (CA).
3. Once the web browser gets the web server certificate, it will check the issuer's digital signature to make sure the certificate is valid.
4. Once the certificate signature is verified, the padlock icon appears in the address bar which indicates that the web servers' public key really belongs to the web server.

5. One shared secret or symmetric key is created by the browser, which also keeps one and gives the web server a copy. But the browser shouldn't send the shared secret as plaintext. As a result, before sending the secret to the web server, it encrypts it using the web server's public key.

6. The web server obtains the shared key from the browser by utilizing its private key to decrypt messages. Then, using the same key as the symmetric key, all communication between a client and a web server will be encrypted and decrypted.

HTTPs (cont.)

HTTPs vs HTTP

HTTP	HTTPS
The full form of HTTP is the Hypertext Transfer Protocol.	The full form of HTTPS is Hypertext Transfer Protocol Secure.
It is written in the address bar as http://.	It is written in the address bar as https://.
The HTTP transmits the data over port number 80.	The HTTPS transmits the data over port number 443.
It is unsecured as the plain text is sent, which can be accessible by the hackers.	It is secure as it sends the encrypted data which hackers cannot understand.
It is mainly used for those websites that provide information like blog writing.	It is a secure protocol, so it is used for those websites that require to transmit the bank account details or credit card numbers.
It is an application layer protocol.	It is a transport layer protocol.
It does not use SSL.	It uses SSL that provides the encryption of the data.
Google does not give the preference to the HTTP websites.	Google gives preferences to the HTTPS as HTTPS websites are secure websites.
The page loading speed is fast.	The page loading speed is slow as compared to HTTP because of the additional feature that it supports, i.e., security.

Understanding Public Key Infrastructure (PKI) An RSA Data Security White Paper. https://paper.bobyline.com/System/understanding_pki.pdf

Stallings, W. (2006). Cryptography and network security principles and practices 4th edition



Thank You