CSE 477: Introduction to Computer Security

Lecture – 24

Course Teacher: Dr. Md Sadek Ferdous

Assistant Professor, CSE, SUST

E-mail: ripul.bd@gmail.com

Outline

- Introduction to HTML & HTTP
- Attacks on Web Protocol
- Mitigations Techniques

Introduction to HTML & HTTP

- World Wide Web (WWW), or web in short
  - A fundamental technology to change almost every aspect of our lives
- We use the web for banking, shopping, education, communicating, news, entertainment, collaborating, social networking and what not!
- The web is based on two major technologies: HTML and HTTP

HTML & HTTP

- Web is a collection of web sites
- Each web site consists simply of web pages of text, images and other multimedia contents
- Each web page is written in HTML (Hypertext Markup Language)
- A web browser is utilised to interpret an HTML page and visualise it to any user

HTML & HTTP

- HTML features
  - Static document description language
  - Supports linking to other pages and embedding images by reference
  - User input sent to server via forms
- HTML extensions
  - Additional media content (e.g., PDF, video) supported through plugins
  - Embedding programs in supported languages (e.g., JavaScript, Java) provides dynamic content that interacts with the user, modifies the browser interface, and can access the client computer environment

HTML & HTTP

- HTTP, Hypertext Transfer Protocol, is used for retrieving the requested web page
- How the web works:
    - The user launches the browser and types in the web address (DNS name) of the requested web page
    - The web browser first checks the local DNS cache for an entry corresponding to the domain of the web site
    - If no entry is found locally, the browser queries a DNS server to resolve the IP address of the domain name
    - The browser makes a TCP connection to a specified port on the web server, port 80 for HTTP
    - The browser submits an HTTP request to the server and the server returns an HTTP response, containing the requested HTML web page
    - The browsers renders the web page into its interface

HTML & HTTP

- HTML is structured using different types of tags which include:
- ***Text formatting***, such as <i>*text*</i>, for italics and <b>*text*</b>, for bold
- ***Itemized lists***, which list items set apart with bullets or numbers, such as <ul> <li>*first-item*</li> <li>*second-item*</li> </ul>
- ***Hyperlinks***, which provide ways to navigate to other webpages, such as in <a href="*web-page-URL*"> *Description of the other page*</a>
- ***Scripting code***, which describes various actions for the web page, such as in <script>*Computer code*</script>
- ***Embedded images***, such as in <img src="*URL-of-an-image*">

HTML

- Even though a web browser displays a web page as a single unit, the browser might actually have to make multiple HTTP requests in order to retrieve all the various elements of the page
- For example, each image embedded in a page would normally be fetched by a separate HTTP request, as would the main HTML file describing the web page itself
- Once all the responses for a page are received, the web browser interprets the delivered HTML file and displays the associated content
- In addition, most browsers provide a way for a client to directly view the source HTML file for a displayed web page, if desired

HTTP insecurity

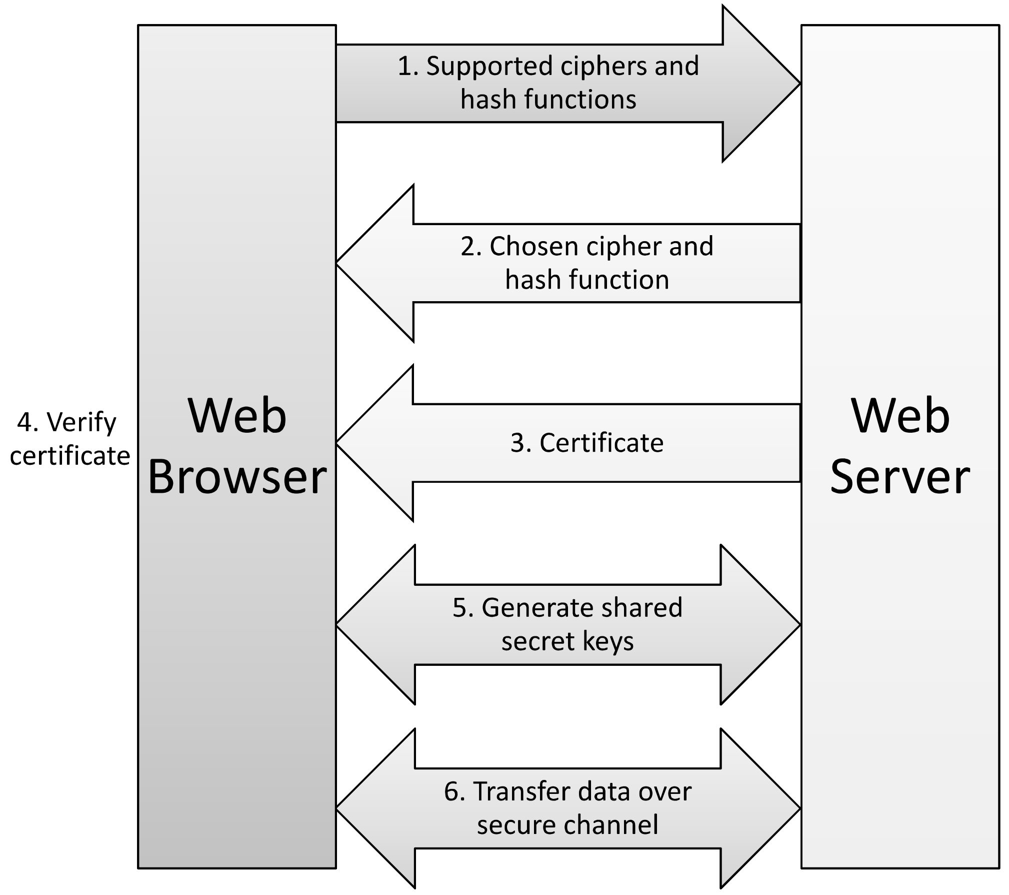
- By default, HTTP requests and responses are delivered via TCP over port 80
- There are many security and privacy concerns with this default means of communication
- The standard HTTP protocol does not provide any means of encrypting its data
    - That is, the contents are sent ***in the clear***
- Because of this lack of encryption, if an attacker could intercept the packets being sent between a web site and a web browser
    - He would gain full access to any information the user was transmitting
    - Could also modify it, as in a ***man-in-the-middle*** scenario
- This lack of confidentiality therefore makes HTTP inappropriate for the transmission of sensitive information such as
    - Passwords, credit card numbers, and Social Security numbers
- In addition, it is difficult to guarantee the identity of a web server using HTTP
    - Are you really talking to correct web server or an attacker is simulating its behaviour

HTTPS/TLS

- To solve the security problem inherent in HTTP, an alternative protocol is available called ***HTTPS*** (***Hypertext Transfer Protocol over Secure Socket Layer***)
- HTTPS is identical to HTTP syntactically, but incorporates an additional layer of security known as ***SSL*** (***Secure Socket Layer***), or a newer implementation, known as ***TLS*** (***Transport Layer Security***)
- SSL and TLS rely on the notion of a ***certificate*** to
  - Verify the identity of the server
  - Establish an encrypted communication channel between the web browser and the web server

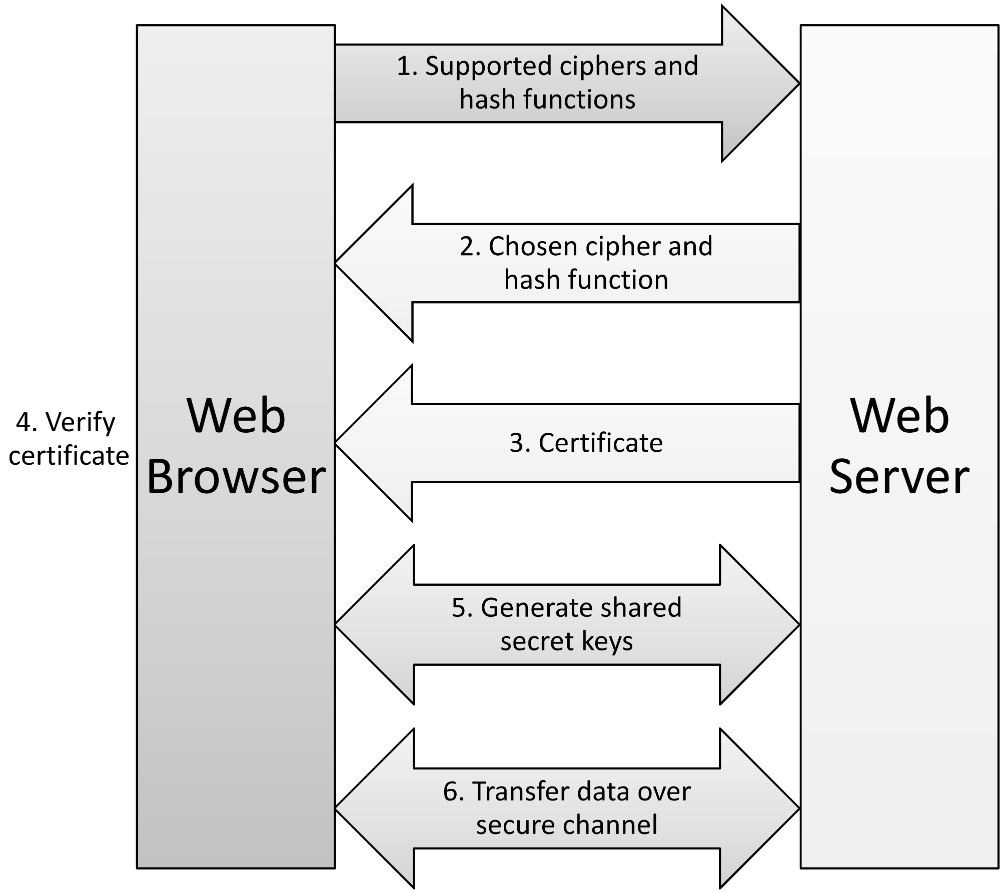
HTTPS/TLS

- The browser requests an HTTPS session with the server
  - provides a list of cryptographic ciphers and hash functions that the client supports
- The server chooses the strongest cipher and hash function that are supported by both the browser and server, informs the browser of its choice
- The server sends back its certificate, which contains the server's public encryption key
- The browser then verifies the authenticity of the certificate


Web Browser
Web Server
1. Supported ciphers and hash functions
2. Chosen cipher and hash function
3. Certificate
4. Verify certificate
5. Generate shared secret keys
6. Transfer data over secure channel

HTTPS/TLS

- To complete the session, the browser encrypts a random number using the server's public key
- The server decrypts it using its private key
- Using this random number, the client and server generate a shared secret
- The shared secret is used to
  - Encrypt each subsequent message using a symmetric encryption algorithm, satisfying confidentiality
  - Provide integrity using MAC (Message Authentication Code)
- Once all primitives are established
  - Communication can be commenced using the normal HTTP protocol
- Namely, a MAC is appended to each HTTP message and the resulting authenticated message is encrypted



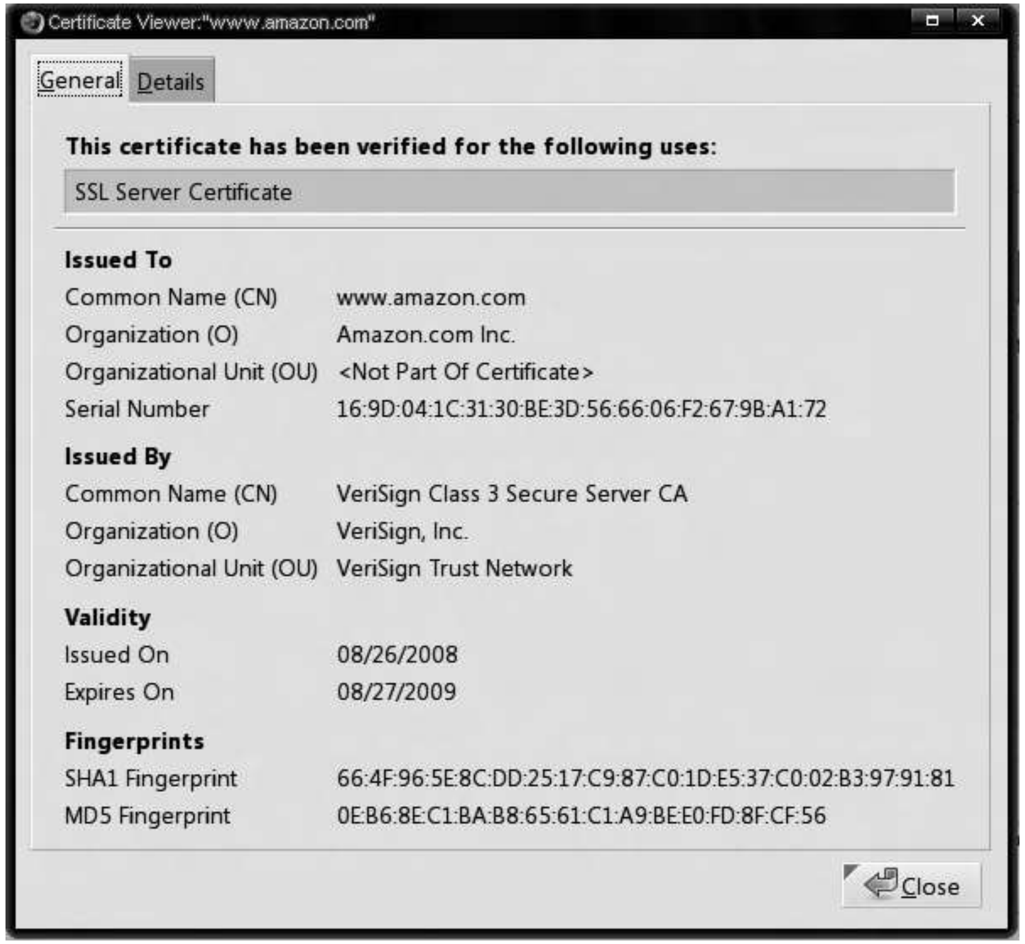
Web Browser
Web Server
1. Supported ciphers and hash functions
2. Chosen cipher and hash function
3. Certificate
4. Verify certificate
5. Generate shared secret keys
6. Transfer data over secure channel

Digital Certificates

- In addition to providing a server's public key for use in generating shared secret keys, certificates provide a means of verifying the identity of a web site to its clients
- To accomplish this goal, certificates are digitally signed using the private key of a trusted third party, known as a ***Certificate Authority*** (***CA***)
- A web site owner obtains a certificate by submitting a ***Certificate Signing Request*** to a CA and paying a fee
- After verifying the identity of the requester and ownership of the domain name for the website, the CA signs and issues the certificate
  - which the web server then sends to browsers to provide proof of its identity
- In short, a web server certificate is an attestation by the ***issuer*** (CA) of a ***subject*** consisting of the organisation owning the web site, the domain name of the web site, and the web server's public key

Digital Certificates

- A web server certificate, also called ***SSL server certificate***, contains several fields, including:
  - Name of the CA that issued the certificate
  - Serial number, unique among all certificates issued by the CA
  - Expiration date of the certificate
  - Domain name of the web site
  - Organization operating the web site and its location
  - Identifier of the public key crypto system used by the web server (e.g., 1,024-bit RSA)
  - Public key used by the web server in the HTTPS protocol
  - Identifier of the cryptographic hash function and public key cryptosystem used by the CA to sign the certificate (e.g., SHA-256 and 2,048-bit RSA)
  - Digital signature over all the other fields of the certificate



This certificate has been verified for the following uses:

SSL Server Certificate

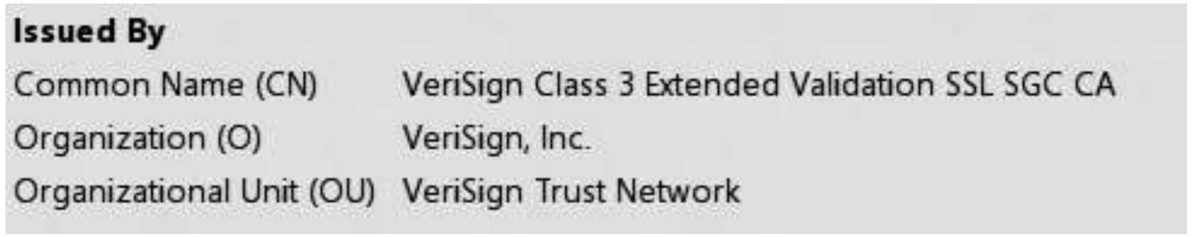
Issued To	
Common Name (CN)	www.amazon.com
Organization (O)	Amazon.com Inc.
Organizational Unit (OU)	<Not Part Of Certificate>
Serial Number	16:9D:04:1C:31:30:BE:3D:56:66:06:F2:67:9B:A1:72
Issued By	
Common Name (CN)	VeriSign Class 3 Secure Server CA
Organization (O)	VeriSign, Inc.
Organizational Unit (OU)	VeriSign Trust Network
Validity	
Issued On	08/26/2008
Expires On	08/27/2009
Fingerprints	
SHA1 Fingerprint	66:4F:96:5E:8C:DD:25:17:C9:87:C0:1D:E5:37:C0:02:B3:97:91:81
MD5 Fingerprint	0E:B6:8E:C1:BA:B8:65:61:C1:A9:BE:E0:FD:8F:CF:56

Certificate hierarchy

- How to verify the identity of the CA which signs a certificate?
  - Utilise another certificate which is signed by another higher authority
- This creates a certificate hierarchy consisting of inter-mediate CA whose certificate is signed by a CA
- In these cases, the top-level certificate is known as the ***root certificate***
- Since the root certificate clearly cannot be signed by a higher authority, the root certificate is known as a ***self-signed certificate***, where the issuer is the same as the subject
  - A self-signed certificate essentially asserts its own legitimacy
- Root certificates are referred to as ***anchor points*** in the chain of trust used to verify a certificate
- Such certificates are typically stored by the operating system or the browser in protected files, in order to be able to validate certificates lower in the hierarchy

Extended validation certificates

- Some CAs only use what is known as ***domain validation***—confirmation that the domain on the certificate being signed is in fact owned by the certificate requester
- To verify the authenticity of domains requesting certificates, ***extended validation certificates*** were introduced
- This new class of certificates can only be issued by CAs who pass an audit demonstrating that they adhere to strict criteria for how they confirm the subject's identity
- These criteria are set by the CA/Browser Forum, an organization including many high-profile CAs and vendors
- Extended validation certificates are designated in the CA field of the certificate, as shown in the figure



Issued By

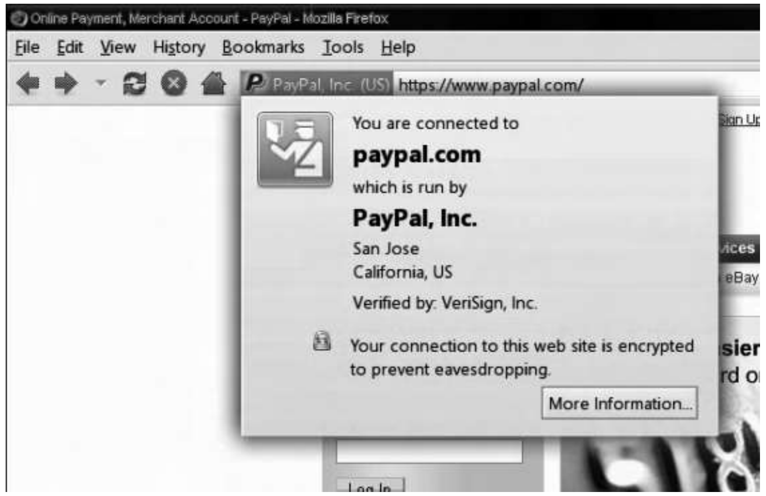
Common Name (CN)	VeriSign Class 3 Extended Validation SSL SGC CA
Organization (O)	VeriSign, Inc.
Organizational Unit (OU)	VeriSign Trust Network

Trustworthiness and usability of certificates

- The contents of a certificate specify a validity period after which it expires and is no longer considered acceptable verification of authenticity
- In addition to this built-in expiration date, a certificate includes the URL of a revocation site, from which one can download a list of certificates that have become invalid before their expiration date, called **Certificate Revocation List (CRL)**
- There are several reasons for a certificate to become invalid, including
  - private key compromise or change of organization operating the web site
- When a certificate becomes invalid, the CA revokes it by adding its serial number to the certificate revocation list, which is signed by the CA and published at the revocation site
- Checking the validity of a certificate involves not only verifying the signature on the certificate, but also
  - downloading the certificate revocation list
  - verifying the signature on this list, and
  - checking whether the serial number of the certificate appears in the list

Trustworthiness and usability of certificates

- The entire concept of certificates relies on the user understanding the information a browser displays and making informed decisions
- For example, most browsers display a visual cue when establishing a secure connection, such as
    - A padlock icon
    - The logo and name of the organisation operating the website in an area with green background of the address bar
    - Clicking on this area displays a summary of the certificate
- Additional cues are provided for extended validation certificate


Online Payment, Merchant Account - PayPal - Mozilla Firefox
File Edit View History Bookmarks Tools Help
PayPal, Inc. (US) https://www.paypal.com/
More Information...

You are connected to
paypal.com
which is run by
PayPal, Inc.
San Jose
California, US
Verified by: VeriSign, Inc.

Your connection to this web site is encrypted to prevent eavesdropping.

Semantic attack on digital certificates

- Digital certificates provide a syntactically correct technology
- Unfortunately, might struggle against semantic attack
- *Bruce Schneier on Security: “Attacks that target the way we, as humans, assign meaning to content. . . .Semantic attacks directly target the human/computer interface, the most insecure interface on the Internet”*
- Buy a domain 0ntest.com (it is an 0, not O)
- Buy a digital certificate for 0ntest.com
- Prepare a website similar to ontest.com
- Redirect a user to 0ntest.com
  - The browser will validate the certificate!
- Can a layman identify the difference?

https://www.schneier.com/crypto-gram/archives/2000/1015.html

Dynamic HTML

- If a web page provides only fixed images, it is called static
  - They are less functional, cannot facility the change of contents based on user interaction!
- In contrast, pages featuring ***dynamic content*** can change in response to user interaction or other conditions, such as the passage of time
- To provide these features, additional web languages called ***scripting languages*** were introduced
- A scripting language is a programming language that provides instructions to be executed inside an application (like a web browser), rather than being executed directly by a computer
- A program written in a scripting language is called a ***script***
- Mainly two types:
  - ***client- side scripting languages*** describe code delivered to the browser, where it is executed by a module of the browser that knows how to interpret the instructions and perform the specified actions
  - ***server-side scripting languages*** describe code that is executed on the server hosting a web site, hiding the code from the user and presenting only the output of that code
- With scripting languages, developers can make pages that change based on the user’s interaction, creating a more interactive experience

Document Object Model

- The ***Document Object Model*** (***DOM***) is a means for representing the content of a web page in an organized way
- The DOM framework takes an object-oriented approach to HTML code, conceptualizing tags and page elements as objects in parent-child relationships, which form a hierarchy called the ***DOM tree***
- The DOM facilitates the manipulation of the content of the web page by scripts, which can access objects on the web page by traversing the DOM tree

JavaScript

- One of the earliest and most popular examples of a scripting language is ***JavaScript***
    - introduced in 1995 and is now supported by every major browser
- JavaScript gives developers a whole set of tools with which to develop interactive and dynamic web applications
- To indicate to a browser that JavaScript is being used, the <script> and </script> tags are used to separate sections of JavaScript from ordinary HTML code

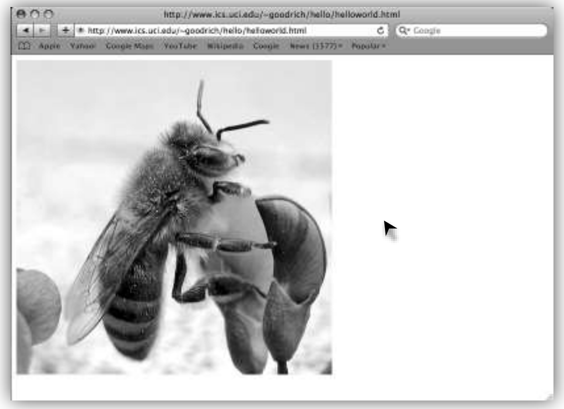
JavaScript

- JavaScript introduces the powerful feature of allowing programmers to declare functions and pass them arguments, upon which they perform some operation or return a value
- Later in the web page, if any line of JavaScript code calls the hello() function, it will result in a pop-up message box that says Hello world!
- In addition to the ability of defining functions, JavaScript also includes several standard programming constructs using the syntax of the C programming language, such as for, while, if/then/else, and switch

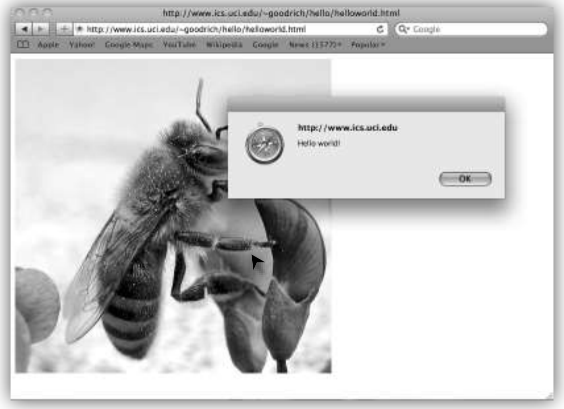
```
< script type="text/javascript">
  function hello() {
    alert("Hello world!");
  }
< /script>
```

JavaScript

- JavaScript also handles events, such as
  - a user clicking a link or
  - even simply hovering the mouse pointer over a portion of a web page, which is known as a ***mouse-over*** event
- These event handlers can be embedded in normal HTML code as below
  - <img src="picture.gif" onMouseOver="javascript:hello()">
- On hovering the mouse pointer on this image, the previously declared hello() function will be called, resulting in a pop-up message box



(a)



http://www.ics.uci.edu

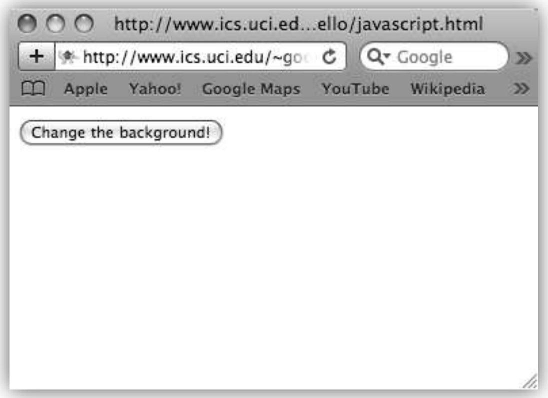
Hello world!

(b)

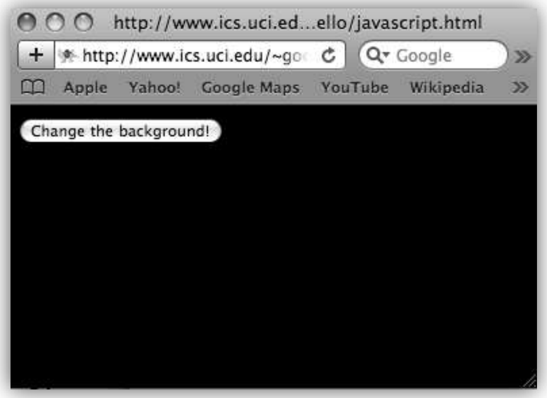
JavaScript

- JavaScript can dynamically alter the contents of a web page by accessing elements of the DOM tree

```
<html>
  <head>
    <script type="text/javascript">
      function changebackground() {
        if (document.bgColor=="#FFFFFF") {
          document.bgColor="#000000";
        }
        else {
          document.bgColor="#FFFFFF";
        }
      }
    </script>
  </head>
  <body bgcolor="#FFFFFF">
    <button type="button" onClick="javascript:changebackground()">
      Change the background!
    </button>
  </body>
</html>
```


http://www.ics.uci.ed...ello/javascript.html
http://www.ics.uci.edu/~go
Google
Apple
Yahoo!
Google Maps
YouTube
Wikipedia
Change the background!

(a)


http://www.ics.uci.ed...ello/javascript.html
http://www.ics.uci.edu/~go
Google
Apple
Yahoo!
Google Maps
YouTube
Wikipedia
Change the background!

(b)

HTTP Sessions

- It is often useful for web sites to keep track of the behaviour and properties of its users
- The HTTP protocol is stateless, however, so web sites do not automatically retain any information about previous activity from a web client
  - When a web client requests a new page to be loaded, it is viewed by default as a fresh encounter by the web server
- The notion of a ***session*** encapsulates information about a visitor that persists beyond the loading of a single page
- For example, a web site that has user accounts and a shopping cart feature would ideally keep track of its visitors so they are not forced to reauthenticate with each new page or keep track of item numbers to enter later on an order form
- There are different ways to achieve this:
  - passing session information via GET or POST variables
  - using a mechanism known as ***cookies***, and
  - implementing server-side session variables

HTTP Sessions

- Session information should be considered extremely sensitive
- This is because it is used today to allow users to maintain a consistent identity on sites that allow accessing bank accounts, credit card numbers, health records, and other confidential information
- Accompanying the concept of a session is a class of attacks known as ***session hijacking***
    - any scenario that allows an attacker to impersonate a victim's identity by gaining access to the user's session information and authenticating to a web site

HTTP Session via GET and POST

- One technique to establish user sessions is to pass session information to the web server each time the user navigates to a new page using GET or POST requests
- The session information is passed via a mechanism called ***hidden fields***
  - Hence, the information is invisible to the user
- Each time the user navigates to a new page, this code passes the user’s session information to the server allowing it to “remember” the user’s state
- The web server then performs any necessary operations using this information and generates the next page with the same hidden code to continue passing the session information
- This method is particularly susceptible to man-in- the-middle attacks, unfortunately, since HTTP requests are unencrypted
- An attacker gaining access to the GET or POST variables being submitted by a user could hijack their session and assume their identity
- In order to safely employ this method, HTTPS must be used in conjunction with sessions

The lecture slides can be found in the following location!