Aspects of Web Security

HyperText Markup Language (HTML)

- Encodes content of web page
- Encoded content (text, picture, URL, sound etc) preceded by *beginning tag* and terminated by *ending tag*
- Tags indicate content type

HyperText Transfer Protocol (HTTP) (I)

- Protocol for web page retrieval
- Stateless request/response protocol
- HTTP transactions may go through proxy which could be on user's own machine
- Proxy acts as cache and serves to save bandwidth

HTTP (II)

- HTTP defined in RFC 2616
- Main requests are GET and POST for retrieving and sending information from/to a web server
- Response to GET is information required and status
- Status could be OK, not found, unauthorised or redirect to new URL

Information in HTTP Request Header (I)

- FROM – contains user's email address
  - browser needs to be configured by user
  - may become source of SPAM
  - limits privacy in browsing site
- AUTHORISATION – sent by browser to server on request
  - may be username and password or an HTTP Digest Authentication
  - Information may be held by browser

Information in HTTP Request Header (II)

- Cookie – data given to client by server and later returned to server in subsequent request
- Referer – URL of page from which client came

HTTP Digest Authentication (I)

- Low budget security alternative to SSL
- Challenges in providing security for HTTP
  - HTTP is stateless
  - User authentication probably password based
  - Should allow multiple requests with a single authentication
  - Should protect against compromise of server database

HTTP Digest Authentication (II)

- Client request information from URL
- Server replies with *unauthorised*, including nonce and *www-Authenticate:Digest*
- Client replies with crypto combination of password, nonce and URL
- Server checks reply against hash of password held in store and nonce
- Client increments nonce count

HTTP Digest Authentication (III)

- Server may specify Quality of Protection (QOP) for message integrity
  - *Auth* is authentication only (and protects URL)
  - *Auth_int* is authentication and integrity (of body of message)
  - *Auth, auth_int* is authentication only is acceptable

Cookie Overview

- Need to hold information accumulated during browsing but HTTP is stateless
- Cookie mechanism allows server to maintain context
- Cookie is data structure created by server and stored at client

Cookie Rules

- Cookie could contain all state information required for interaction with server but size must be less than 4K octets
- Server could hold database and use cookie to define index
- Server specifies restrictions on who should receive cookie from client when returned
- Minimum of 2 dots in DNS name to prevent sites sharing cookie information
- Cookies have lifetimes specified by server

Spoofing a site to a user

- URL may be of form
- http://www.recognisablename@inconspicuousDNSname
- Rogue site may act as man in the middle to gain passwords and/or other information
- SSL offers protection if all trusted CAs not compromised

Impersonation by Subsequent User

- Browser holds username and password requested by server for future requests
- If initial user does not logout then subsequent user may use this information
- Some cookies held in stable storage – real security problem

Poisoned Cookies

- Servers may use cookies without cryptographic protection
- Cookie may have user ID which may be modified to allow an attacker into system
- Cookie could contain price information for purchaser which could be changed to purchaser's advantage