DFSC 1316: digital forensic and information assurance fundamentals I

8. Browser Forensics

HTTP Protocol

- Hypertext Transfer Protocol (HTTP) is an application layer protocol.
  - Based on TCP, uses port 80.
  - Used for Hybermedia exchange.

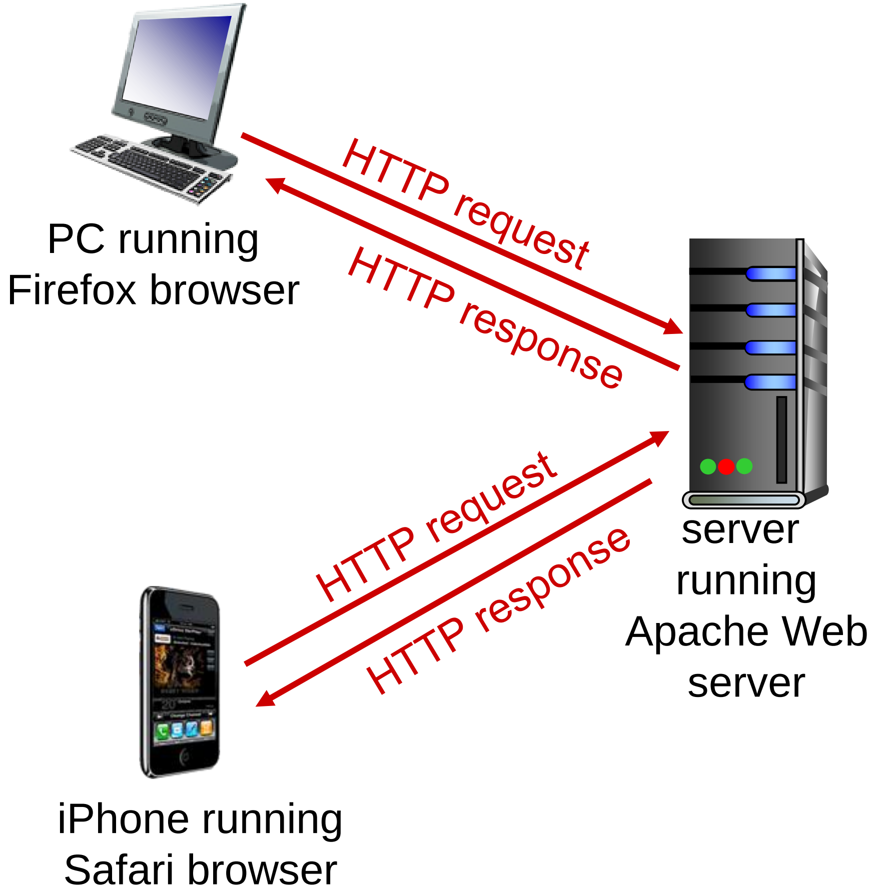
Web and HTTP

- *web page* consists of *objects*
- object can be HTML file, JPEG image, Java applet, audio file,…
- web page consists of *base HTML-file* which includes *several referenced objects*
- each object is addressable by a *URL,* e.g.,

$$\underbrace{\texttt{www.someschool.edu}}_{\text{host name}}\texttt{/}\underbrace{\texttt{someDept/pic.gif}}_{\text{path name}}$$

HTTP overview

- client/server model
  - *client:* browser that requests, receives, (using HTTP protocol) and "displays" Web objects
  - *server:* Web server sends (using HTTP protocol) objects in response to requests



PC running Firefox browser
HTTP request
HTTP response
server running Apache Web server
HTTP request
HTTP response
iPhone running Safari browser

HTTP overview (continued)

Based on TCP:

- client initiates TCP connection (creates socket) to server, port 80
- server accepts TCP connection from client
- HTTP messages (application-layer protocol messages) exchanged between browser (HTTP client) and Web server (HTTP server)
- TCP connection closed

HTTP connections

non-persistent HTTP

- at most one object sent over TCP connection
  - connection then closed
- downloading multiple objects required multiple connections

persistent HTTP

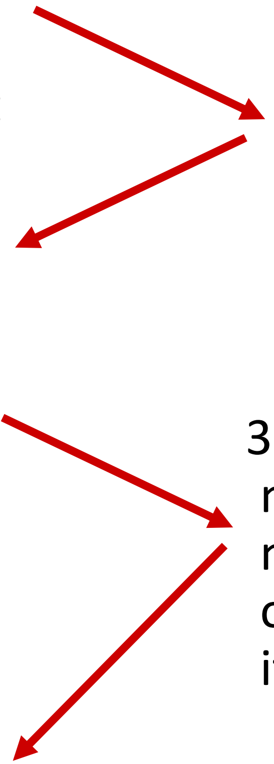
- multiple objects can be sent over single TCP connection between client, server

Non-persistent HTTP

suppose user enters URL:

`www.someSchool.edu/someDepartment/home.index`

(contains text, references to 10 jpeg images)



1a. HTTP client initiates TCP connection to HTTP server (process) at www.someSchool.edu on port 80

1b. HTTP server at host www.someSchool.edu waiting for TCP connection at port 80. “accepts” connection, notifying client

2. HTTP client sends HTTP request message (containing URL) into TCP connection socket. Message indicates that client wants object someDepartment/home.index

3. HTTP server receives request message, forms response message containing requested object, and sends message into its socket

Non-persistent HTTP (cont.)

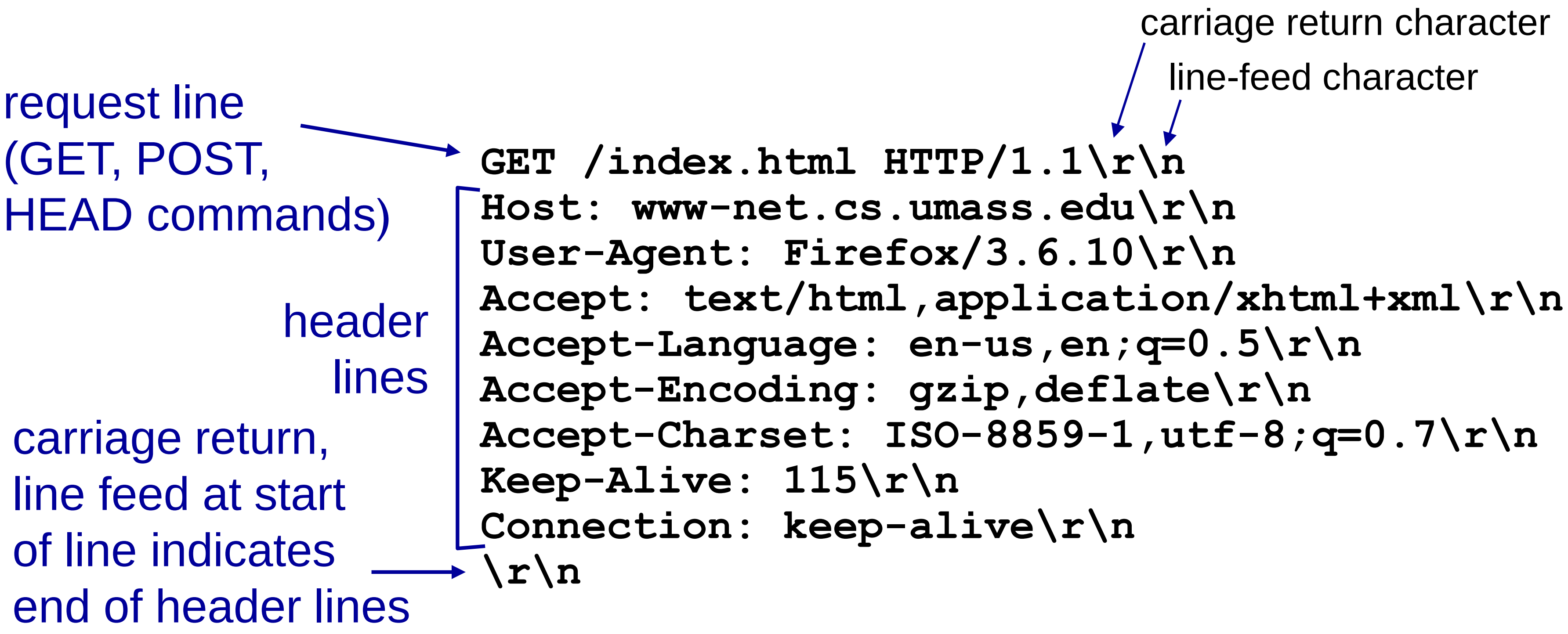
4. HTTP server closes TCP connection.

5. HTTP client receives response message containing html file, displays html. Parsing html file, finds 10 referenced jpeg objects

6. Steps 1-5 repeated for each of 10 jpeg objects

HTTP request message

- two types of HTTP messages: *request, response*



carriage return character

line-feed character

request line
(GET, POST,
HEAD commands)

header
lines

carriage return,
line feed at start
of line indicates
end of header lines

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

HTTP response message

status line
(protocol
status code
status phrase)

header
lines

data, e.g.,
requested
HTML file

```
HTTP/1.1 200 OK\r\n
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-
   1\r\n
\r\n
data data data data data ...
```

An Example of Trying out HTTP

1. Telnet to your favorite Web server:

```
telnet gaia.cs.umass.edu 80
```

opens TCP connection to port 80 (default HTTP server port) at gaia.cs.umass. edu.

anything typed in will be sent to port 80 at gaia.cs.umass.edu

2. type in a GET HTTP request:

```
GET /kurose_ross/interactive/index.php HTTP/1.1
Host: gaia.cs.umass.edu
```

by typing this in (hit carriage return twice), you send this minimal (but complete) GET request to HTTP server

3. look at response message sent by HTTP server!

(or use Wireshark to look at captured HTTP request/response)

Method types

HTTP Command	**Description**
GET	Retrieves the document specified in the URL property
HEAD	Gets the header information
POST	Sends data to the server
PUT	Replaces the page specified in the URL property with the specified data

HTTP response status codes

`200 OK`

- request succeeded, requested object later in this msg

`301 Moved Permanently`

- requested object moved, new location specified later in this msg (Location:)

`400 Bad Request`

- request msg not understood by server

`404 Not Found`

- requested document not found on this server

`505 HTTP Version Not Supported`

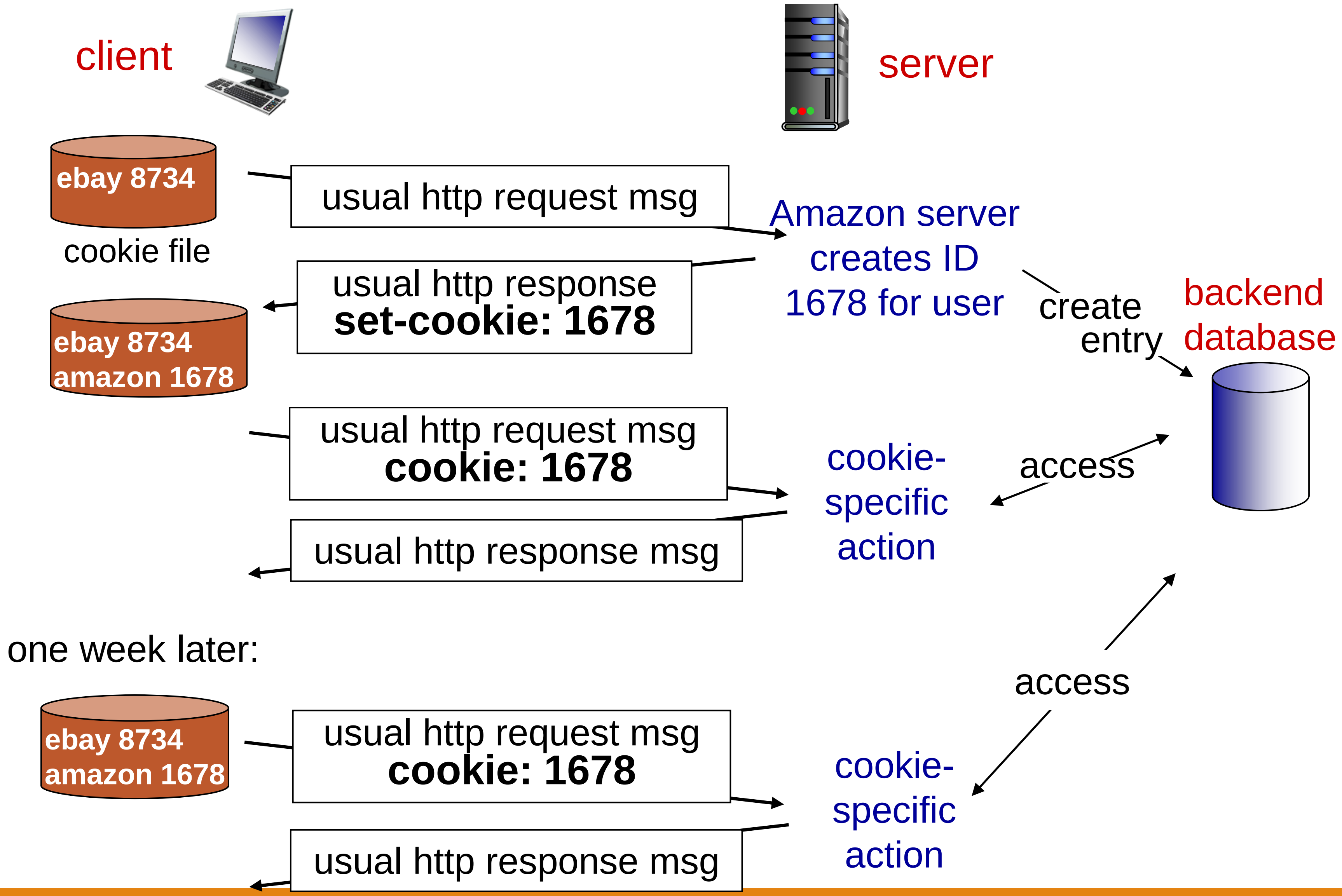
Forensic Evidence: cookies

- HTTP protocol is a stateless protocol.
  - Every time a request is sent to a server, the server behaves as if it is the first time the request is made by the client.
  - Why?
  - How to solve problems such as:
    - The items you put in the shopping cart will remain the next time you visit Amazon.
    - You are using a new computer to access your bank account, and the bank want to verify it is you.

Forensic Evidence: cookies

- The using of cookie is a common way of performing session management between HTTP server and client.
  - Cookie is a small piece of data (a file) the HTTP server puts on the client.
  - Cookie will be sent by the client to the server when a session starts, so the server “recognize” the client.

Cookies: keeping “state”



client

server

ebay 8734

cookie file

usual http request msg

Amazon server creates ID 1678 for user

create entry

backend database

usual http response
set-cookie: 1678

ebay 8734
amazon 1678

usual http request msg
cookie: 1678

cookie-specific action

access

usual http response msg

one week later:

ebay 8734
amazon 1678

usual http request msg
cookie: 1678

access

cookie-specific action

usual http response msg

Forensic Evidence: cookies

Common Web Browsers	Format	Location
Internet Explorer	Stores all cookies for a site in a text (*.txt*) file named for the site, and tracks them in an *index.dat*file.	C:\Users\<User>\AppData\Roaming\Microsoft\Windows\Cookies
Safari	Stores all cookies in an XML formatted file, *Cookies.plist.*	~/Library/Cookies (Not verified)
Firefox	Stores all cookies as records in a SQLite database, *cookies.sqlite.*	C:\Users\<User>\AppData\Roaming\Mozilla\Firefox\Profiles
Chrome	Stores all cookies as records in a SQLite database, *Cookies.*	C:\Users\<User>\AppData\Local\Google\Chrome\User Data\Default

Forensic Evidence: cookies

what cookies can be used for:

- authorization
- shopping carts
- recommendations
- user session state (Web e-mail)

cookies and privacy:

- Cookies permit sites to learn a lot about you
- You may supply name and email to sites

Forensic Evidence: flash cookies

- It is also know as Local Shared Objects.
- LSO is a cookie-like data that can be placed by a web site that is running Adobe Flash.
- What is the usage of a flash cookie?

Forensic Evidence: flash cookies

- Like regular cookie, a flash cookie also contains user information, such as the time when the site is visited.
- However, it is even more stealthier, and won’t be cleaned when you clean cookies via browser.
- Setting can be changed at Adobe Flash Player setting manager

Forensic Evidences: history

- Index.dat
  - The index.dat files are a data base file that is generated by MS Internet Explorer.
  - It contains browsing information such as visited URL, search queries and recently opened files.
  - It is to enable quick access to data used by the internet explorer.
  - On Win7, they are stored at
    - \Users\<Username>\AppData\Roaming\Microsoft\Windows\Cookies\index.dat
    - \Users\<Username>\AppData\Roaming\Microsoft\Windows\Cookies\low\index.dat

Forensic Evidences: history

- Registry
  - Registry is a centralized hierarchical database in Windows operation system.
  - It stores critical system information such as system configuration, hardware spec, etc.
  - Registry is a data that contains multiple keys, subkeys, and values. A group of keys is called a hive.
  - Keys that relates to IE:
    - *HKEY_CURRENT_USER\Software\Microsoft*\Internet *Explorer\TypedURLs* contains all URLs that were typed by the user.
    - *HEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces* contains network interface conf information.

Forensic Evidences: history

- History in IE
  - C:\Users\<User>\AppData\Local\Microsoft\Windows\History
- User Profile in Fire Fox
  - C:\Users\<User>\AppData\Roaming\Mozilla\Firefox\Profiles
- User Data in Chrome
  - C:\Users\<User>\AppData\Local\Google\Chrome\User Data\Default

Forensic Evidences: Cache

- Cache in IE
  - C:\Users\<User>\AppData\Local\Microsoft\Windows\Tempr oy Internet Files
- User Profile in Fire Fox
  - C:\Users\<User>\AppData\Roaming\Mozilla\Firefox\Profiles
- User Data in Chrome
  - C:\Users\<User>\AppData\Local\Google\Chrome\User Data\Default

Forensic Evidences: Automated Tools

- Many freeware tools are available.
  - Such as: http://www.nirsoft.net/
  - Usually developed by individuals for specific purposes.
- Better to use multiple and cross-validate the result.

Practice: private browsing

- Most browsers now has an option for “private browsing”, such as the “incognito mode” for Google Chrome.
    - Chose one browser, visit a website that uses cookies, such as Amazon, in both normal mode, and private mode.
    - Compare the results, e.g., will private mode store cookies, caches, etc.