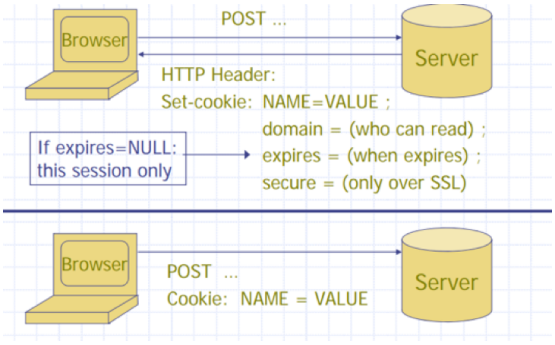
CAS CS 357

In-class Note 4

Cookies in Computer Science

1. Https vs Http (continued)
    a. Https is an encrypted version of http
    b. Without cookie, communication between client and server is stateless (server forgets the client between queries
    c. In the old days (90s), server was stateless since they did not require information from the client when posting an image of a cat
    d. In addition, there were not enough memory space for the server to keep track of everybody who visited the website
2. Cookie: client state
    a. Cookies are used to store state on user's machine
    b. 

    c. Without cookie, the site doesn't know who you are, making it hard to send information back to the user since they cannot recognize each different user
    d. Nowadays, every site wants to keep track of information, leading them to have cookies

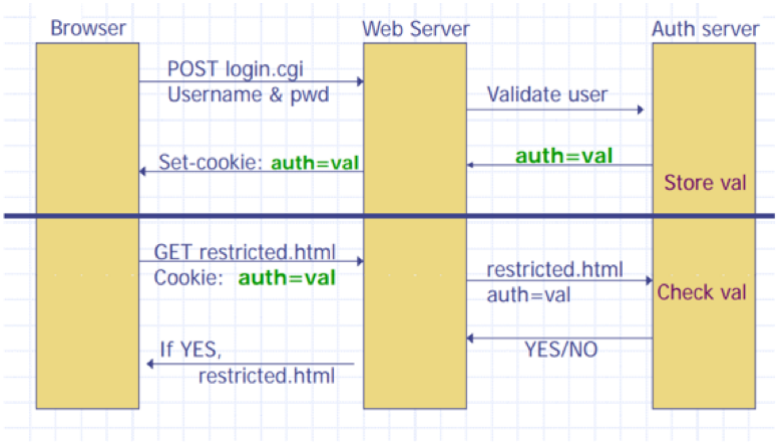
e. One example is login cookie (Google, Bank of America, etc.) where user enters password and account name to identify each user. It prevents people from logging in in every single request. For example, user does not need to reenter the account and password in Gmail every single time they open their computer or when they move to a different page in Gmail (google doc, etc.)
f. Cookie is simply a piece of information. In other words, it is a state that client has
g. They are unique random numbers per user that the server uses to identify who the user is.
h. Cookies can be expired in certain minutes, hours, and etc. and it depends on the 'expires' variable from the figure above. For example, user can be logged into Google forever while BOA forgets the cookie of the user in a few minutes (most banks do this for security issue)
i. If the cookie expires, users need to identify themselves to the server by logging in again
j. Shown in figure above, cookies have name, domain, expires, and secure as variables
k. While doing POST action by the user to the server, cookie is also sent to the server for the server to know who I am
l. Every cookie is tied to domain.

3. Security

A. Cookies should be sent to the server over https for security
B. If cookies are sent through http, the man in the middle can interfere and once they receive the unique cookie from the user, they can behave like the user

C. The man in the middle can rewrite secure cookies or can log user into attacker's account after knowing the information of the cookie

4. Cookie authentication

a. 

b. The figure above shows how cookies are created. When user initially logs in, the web server validates the user to authorization server and when it is valid, web server sets the cookie to that specific user

c. When doing other actions such as GET, the cookie is also sent to the web server and when authentication is valid, the web server does the command that the user asked for (in this case, it is GET restricted.html so the server sends the particular file to that user who requested for it)

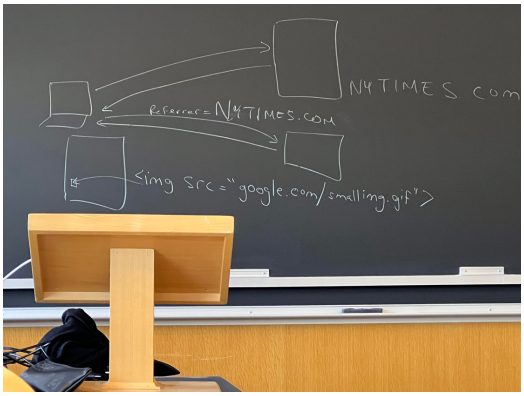
5. Cookie Security Policy

a. Uses: user authentication, personalization, user tracking

b. Browser will store: at most 20 cookies per site, and maximum 3KB per cookie

c. Origin is the tuple <domain, path>: can set cookies valid across domain suffix

6. Sharing cookies

a. 

b. Cookies can be sent from one site to another site

c. When visiting one site such as nytimes.com, there can be an image or advertisement of google.com. Here, you are actually visiting two sites or the two sites are communicating with one another

d. This is possible between the two sites by using code in HTML:

img src = “google.com/smalling.gif”

e. Therefore, this allows google to receive the cookie that the user used to go into nytimes.com and google can store information regarding the user without the user going directly to google.com

f. Another problem can rise here in between connecting websites. If at least one website is using http instead of https, the man in the middle can steal information from that browser and since the man in the middle now can gain information of the user and its cookie, it can act like the user in other https websites

g. Important idea is that all websites connected to each other has to be secure in the form of https

7. Same-origin cookie

a. When people log in at Google, the cookie remains there until they log out from that account

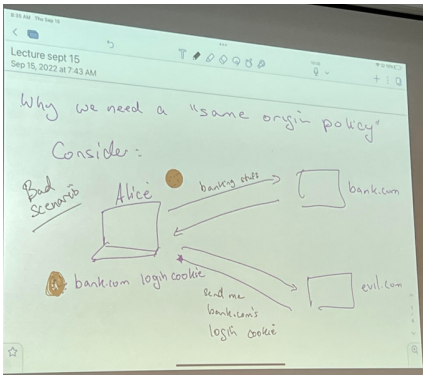
b. Same-origin policy is a concept where web browser permits scripts contained in first web page to access data in second web page

c. This happens only when both websites have the same origin (defined as combination of URI scheme, host name, and port number)

d.

Compared URL	Outcome	Reason
http://www.example.com/dir/page2.html	Success	Same scheme, host and port
http://www.example.com/dir2/other.html	Success	Same scheme, host and port
http://username:password@**www.example.com**/dir2/other.html	Success	Same scheme, host and port
http://www.example.com:**81**/dir/other.html	Failure	Same scheme and host but different port
https://www.example.com/dir/other.html	Failure	Different scheme
http://**en.example.com**/dir/other.html	Failure	Different host
http://**example.com**/dir/other.html	Failure	Different host (exact match required)
http://**v2.www.example.com**/dir/other.html	Failure	Different host (exact match required)
http://www.example.com:**80**/dir/other.html	Depends	Port explicit. Depends on implementation in browser.

e. Same-origin cookie is important since other websites can steal information about the cookie.

f. 

g. For example, when a user is at its gmail or bank account that contains important information regarding the user, the user can also be at a malicious website (such as evil.com for simplicity). At evil.com, without the same-origin cookie, it can ask for the cookie from the user that was received from important websites such

as gmail or bank and can interfere in between the action (acting like man in the middle) à stealing credentials

8. Cross-origin resource sharing
    a. Problem rises from same origin cookie
    b. Gmail.com and the google drive from the gmail.com are two completely different cross-origins. However, the information and cookie is sent over from gmail to google drive (user doesn't need to reenter his/her information)
    c. This is because of cross-origin resource sharing
    d. Each website has an allow list/whitelist that allows the storage and sharing of cookies to other websites, meaning that when one website allows a certain website, the information and cookie is also sent over when going to the other website such as Gmail and its drive.
9. EvilProxy Phishing Service
    a.
    b. An attacker/adversary can create a false website (misspelled or incorrect version) of a website that is popular, such as google.com as gogle.com, gooogle.com, and etc.
    c. When the user misspells the url to gogle.com or other websites by accident, the website that they accidentally went to might have the same design as the website they intended to log in (gogle.com having same design as google.com)
    d. The user, who thinks that he/she correctly went to the website they intended, can log in using their username and password to create a cookie.

e. After logging in, the fake website actually sends the data to google.com, and therefore the user finds no problem with it since he/she simply logged into google.com as intended
f. Without notifying the user, the adversary can now gain information about the user such as their login information or cookie, which can be later used to attack the user
g. Since the fake website is also created in https, the server also does not find anything that can be harmful and therefore the adversary can act like a man in the middle that can use the user's credentials