ICS 351: Today's plan

- web scripting languages
- HTTPS: SSL and TLS
- certificates
- cookies
web scripting languages

- web content described by HTML was originally static, corresponding to files on the server
- since the server is a program, it can generate content that is generated dynamically, e.g. put the user's name (or bank balance) within the web page
- however, this requires the server administrator to modify the code of the server, which is error-prone
- so instead, the server program can execute a server-side script to generate new content to be served
- this script can be written in any language supported by the system on which the server is running
client-side scripts

- even with a server-side script, each change in the web page requires an HTTP request and reply, and requires that the page be rendered again
- and usually requires an explicit user action such as a mouse click
- to have more interactivity, many browsers have been designed to execute client-side scripts that can modify the displayed page and exchange data over the internet
- while these client-side scripts do much to improve the appearance of pages, there can be concerns about security and reliability
- in an attempt to address these concerns, browsers limit what scripts are allowed to do
- client-side scripts are usually in Java or Javascript
- not all browsers execute client-side scripts
secure HTTP

- HTTP by itself is very insecure: any man-in-the-middle attacker can observe all the content sent and received
- some people wish to use HTTP to send sensitive data, e.g. credit card numbers, personal email
- instead of layering HTTP over TCP, HTTP can be layered over a secure protocol that runs over TCP
- the choice of secure protocols for HTTPS (secure HTTP) is SSL (older) or TLS (newer)
- both SSL and TLS are considered secure, but
  - SSL and TLS can only authentication if they have a public key for the server
- how should one connect to a server that has not been visited before?
certificates

- * a certificate is a digital signature by entity CA verifying that the enclosed public key authenticates server S
- * there are a few certificate authorities (CAs) that are widely known and recognized by many web browsers
- * when presenting its public key, a server S also presents the certificate signed by a CA as evidence that S indeed is the server the user wants to talk with
- * certificates protect against man-in-the-middle attack (including DNS attacks), but are still vulnerable to misspellings (e.g. goggle.com)
HTTP cookies

- HTTP is a stateless protocol: a server has no real way to identify a client, so a request may or may not be connected with prior requests
- instead, a server may offer a client a *cookie*, a small amount of data that is only meaningful to the server
- on subsequent related requests to the same server, the client will send back the cookie, to confirm that the requests are connected
- cookies have an expiration time -- most legitimate cookies expire quickly
- cookies can also be used to attempt to track users as they visit multiple sites, by embedding in the several sites a small image (or other content) served from the same server
- similar tracking can be done by tracking accesses based on the IP number of the connecting client