

## Lesson 2 Outline - CGI Details

- I. Details of browser-server communication
  - A. Control flow
    1. Browser displays HTML with a button or link to a CGI
    2. User clicks on button or link
    3. Browser sends request for CGI URL (and sends any input)
    4. Server executes the CGI script
    5. CGI script returns output
    6. Browser displays output for user
  - B. Input
    1. GET
      - a. input data is passed via the URL
      - b. inherently insecure since data is visible in the URL
      - c. may encounter limits on size of strings sent via GET
    2. POST
      - a. data is encoded and sent after the HTTP header
      - b. more secure
    3. Input is typically sent to the CGI script by means of a form. The METHOD attribute specifies either GET or POST.
    4. If a form is not used, GET data can be passed directly via the URL (or SSI call)
    5. Even if data is not passed via GET or POST, the CGI script has access to environment variables that can reveal some information about the user
  - C. Output
    1. The CGI script must return something to the browser to communicate the results of the action
    2. First thing returned by the CGI is a "Content-type" string to indicate the type of data being returned
      - a. HTML (most common)
      - b. Image (often used in graphical counters)
      - c. Binary data (downloading a file)
      - d. Location (redirecting to another URL)
    3. If no data is returned browser gives "No Data" error
    4. NPH scripts (Non-parsed header) are the exception; they do not include the Content-type string in their output.
- II. Prerequisites for CGI
  - A. The web server must know when a URL refers to a CGI script (and must be executed) instead of an HTML page; this is done two ways
    1. ScriptAlias
    2. File name extension mapping
  - B. The script must comply with the requirements of the web server configuration. These may include one or both of the following:
    1. Placed in appropriate directory (ex. cgi-bin)
    2. Correct file name extension (ex. \*.cgi)
  - C. Permissions on the server must be set properly for the CGI to run
    1. Execute permission on script
    2. It's important to know which user the web server will use to run the CGI scripts; typically it is not the same user you FTP files as, but rather "nobody". This means you have to explicitly grant execute permission on the script to everyone (and any files that will be written to from the script must be writeable by everyone, etc)
- III. Server errors
  - A. The server returns a status code after every HTTP transfer

- B. Error codes can indicate either a server error or a client error
  - C. The granularity of these errors is fairly large; when debugging CGI scripts you will primarily only see the 500 error (Internal Server Error)
- IV. SSI
- A. An alternative to the “click a link or button” approach to invoking CGI scripts
    1. A special comment-like tag is embedded in the HTML
    2. This instructs the server to execute a CGI script and insert the script’s output into the HTML before sending the data back to the browser
  - B. SSI must be specifically enabled on the web sever and files must conform to proper format (ex. Typically must be named as \*.shtml)
  - C. SSI can present security concerns
  - D. Common uses of SSI are hit counters, banner rotations, “today’s date” and “last updated” messages