

The Common Gateway Interface (CGI)

Pat Morin

COMP2405

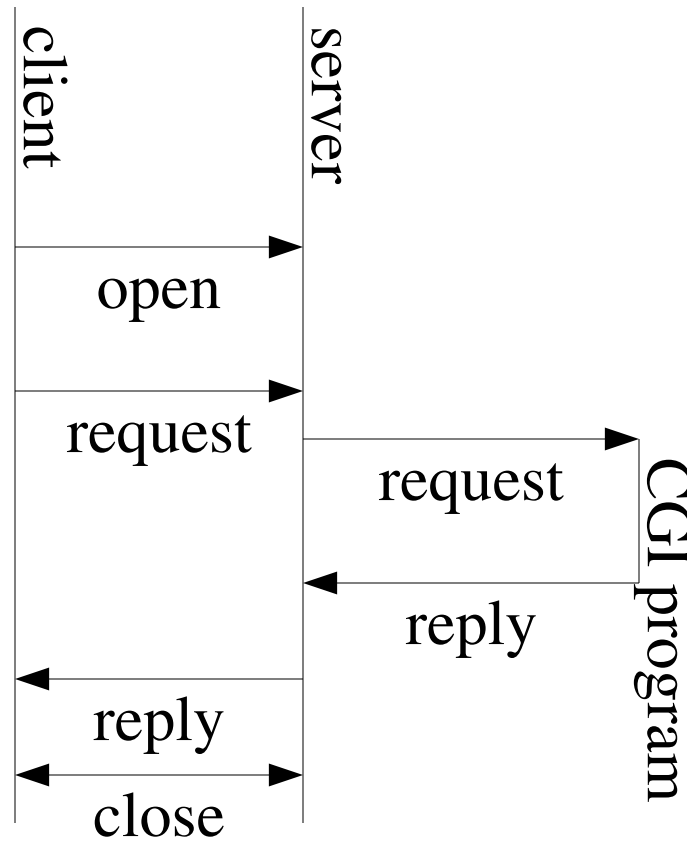
Outline

- What is CGI?
- Details of the Server/Program Interface
 - Environment variables
 - Form data
 - GET versus POST
- Security Issues
 - Common vulnerabilities

What is CGI?

- Recall the usual HTTP Transaction
 1. Client opens connection to server
 2. Client sends request to server
 3. Server responds to request
 4. Client and server close connection
- CGI is all about what happens between steps 2 and 3
- CGI is a standard interface by which the web server passes the client's request to a *program* and receives the response from that program

The CGI Process



The CGI Process

- Client open connection to server
- Client sends request to server
- Server processes request
 - Server launches CGI program
 - CGI program runs
 - CGI program outputs response
- Server sends response to client
- Client and server close connection

Sending the Request to the Program

- The web server sends information to the program using *environment variables*
- This information includes
 - HTTP headers
 - Server information
 - Client information
 - Information about the request

Receiving Form Data

- A CGI program can receive form data in two different ways
- If the form is submitted by the GET method then the query is encoded in the QUERY_STRING environment variable
- If the form is submitted by the POST method then
 - The data arrives on stdin (standard input)
 - The CONTENT_LENGTH environment indicates how much data will arrive (the server does not transmit EOF!)

Sending the Reply

- The CGI program should write its output to `stdout`
- The output consists of
 - A header, containing, as a minimum the `Content-type` but possibly also other header fields, if supported
 - A blank line
 - The content (e.g., text, html, etc.)
- Normally, we use `Content-type: text/html`

Security Concerns

- The client is sending a request that causes the server to execute a program
- The program uses data provided by the client
- Client data can not be trusted!

Security Tips

- Do not trust the client to follow rules
 - setting `maxLength` in a text field does not guarantee that you will never receive a longer string
- Never leave any opportunity to execute data provided by the client (using `eval`, or forgetting quotes)
- Be careful with file names or names passed on a command line
 - if a client sends `"../..../etc/passwd"` as a user name will this give them access to `/etc/passwd` ?

Security Tips (Cont'd)

- Don't store data where it can be accessed by HTTP clients
- Either:
 - Put data in a separate directory that is not under your `public_html` directory, OR
 - Adjust file permissions
- Always escape user-supplied data before outputting it as HTML
- Turn off server-side includes
 - if a client sends "`<!-- #include /etc/passwd -->`" as a username, will this give them access to `/etc/passwd`

Some Technical Issues

- CGI programs can be written in any programming language (C, C++, Java, Perl, bash,...)
- The web server is configured to treat executable files in certain special directories as CGI programs
 - For us, this is `~username/cgi-bin/`
- The user ID that the CGI program is run under depends on the server configuration
 - For us, it is the UID of username
- The CGI program is restricted to performing operations permitted to that user

Summary

- CGI is a simple means by which the server and CGI program exchange data
- CGI (or something like it) is required for creating web pages with dynamic content
- Form data is handled differently depending on whether the form method is GET or POST
- CGI introduces many subtle potential security problems