

CGI Sessions

It's all an illusion (at the HTTP layer)

Netprog: CGI Sessions 1

Sessions

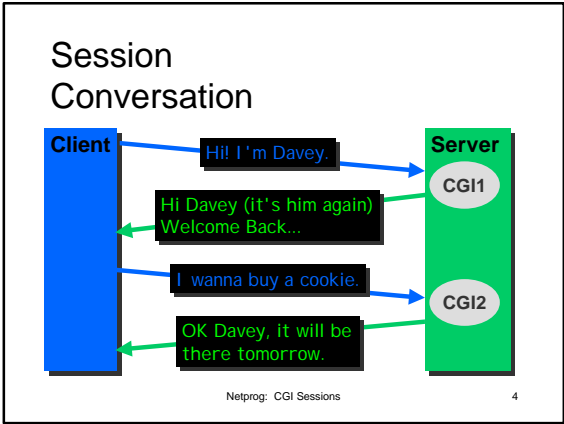
- Many web sites allow you to establish a session.
 - you identify yourself to the *system*.
 - now you can visit lots of pages, add stuff to shopping cart, establish preferences, etc.

Netprog: CGI Sessions 2

State Information

- Remember that each HTTP request is unrelated to any other (as far as the Web server is concerned).
- Each new request to a CGI program starts up a brand new copy of the CGI program.
- Providing *sessions* requires keeping state information.

Netprog: CGI Sessions 3



Hidden Field Usage

- One way to propagate state information is to use hidden fields.
- User identifies themselves to a CGI program (fills out a form).
- CGI sends back a form that contains hidden fields that identify the user or session.

Netprog: CGI Sessions 5

Revised Conversation

Initial form has field for user name.
GET /cgi1?name=davey HTTP/1.0

CGI1 creates order form with hidden field.
GET /cgi2?name=davey&order=cookie HTTP/1.0

Netprog: CGI Sessions 6

Complete Example

On the web is a complete example of a *system* that uses hidden fields to propagate state information.

CGI sample program: [pizza server](#)

Netprog: CGI Sessions

7

Session Keys

- Many Web based systems use hidden fields that identify a *session*.
- When the first request arrives, the system generates a unique *session key* and stores it in a database.
- The session key can be included in all forms/links generated by the system (as a hidden field or embedded in a link).

Netprog: CGI Sessions

8

Session Key Properties

- Must be unique.
- Should *expire* after a while.
- Should be difficult to predict.
 - typically use a pseudo-random number generator seeded carefully.

Netprog: CGI Sessions

9

Pizza Server Session Keys

- We could change the pizza server system to use session keys:

```
<INPUT TYPE=HIDDEN  
NAME=sessionkey  
VALUE=HungryStudent971890237>
```

Netprog: CGI Sessions

10

Pizza Order

A request to order a pizza might now look like this (all on one line):

```
GET /pizza.cgi?sessionkey=  
HungryStudent971890237&pizza=  
cheese&size=large HTTP/1.0
```

Netprog: CGI Sessions

11

HTTP Cookies

- A "cookie" is a *name,value* pair that a CGI program can ask the client to remember.
- The client sends this name,value pair along with every request to the CGI.
- We can also use "cookies" to propagate state information.

Netprog: CGI Sessions

12

Cookies are HTTP

- Cookies are HTTP headers.
- A server (CGI) can *give* the browser a cookie by sending a **Set-Cookie** header line with the response.
- A client can send back a cookie by sending a **Cookie** header line with the request.

Netprog: CGI Sessions

13

Setting a cookie

```
HTTP/1.0 200 OK
Content-Type: text/html
Set-Cookie: customerid=0192825
Content-Length: 12345
Favorite-Cookie: Choco-Chip
Nap-Time: 12-2
...
```

Netprog: CGI Sessions

14

Set-Cookie Header Options

The general form of the Set-Cookie header is:

Set-Cookie: name=value; options

The options include:

expires=...

domain=...

path=...

Netprog: CGI Sessions

15

expires Option

```
expires=Friday 29-Feb-2000 00:00:00 GMT
```

- This tells the browser how long to hang on to the cookie.
- The time/date format is very specific!

Netprog: CGI Sessions

16

expires Time Format

```
Weekday, Day-Month-Year  
Hour:Minute:Second GMT
```

- This all must be on one line!
- Weekday is spelled out.
- Month is 3 letter abbreviation
- Year is 4 digits

Netprog: CGI Sessions

17

Default expiration

- If there is no expires option on the **set-Cookie** header line, the browser does not save the cookie to disk.
- In this case, when the browser is closed it will forget about the cookie.

Netprog: CGI Sessions

18

domain Option

`domain=.rpi.edu`

- The domain option tells the browser the *domain(s)* to which it should send the cookie.
- *Domains* as in DNS.
- The domain must start with "." and contain at least one additional "."

Netprog: CGI Sessions

19

domain option rules

- The server that sends the Set-Cookie header must be in the domain specified.
- If no domain option is in the header, the cookie will only be sent to the same server.

↙
Default Behavior

Netprog: CGI Sessions

20

path Option

`path=/
or
path=~hollingd/netprog`

- The path option tells the browser what URLs the cookie should be sent to.

Netprog: CGI Sessions

21

path default

- If no path is specified in the header, the cookie is sent to only those URLs that have the same *path* as the URL that set the cookie.
- A *path* is the leading part of the URL (does not include the filename).

Netprog: CGI Sessions

22

Default Path Example

If the cookie is sent from:

```
/~hollingd/netprog/pizza/pizza.cgi
```

it would also be sent to

```
/~hollingd/netprog/pizza/blah.cgi
```

but not to

```
/~hollingd/netprog/soda/pizza.cgi
```

Netprog: CGI Sessions

23

Set-Cookie Fields

- Many options can be specified.
- Things are separated by ";":

```
Set-Cookie: a=blah; path=/  
domain=.cs.rpi.edu;  
expires=Thursday, 21-Feb-2002  
12:41:07 2002
```

↖ All must be on one line!

Netprog: CGI Sessions

24

CGI cookie creation

- A CGI program can send back any number of HTTP headers.
 - can set multiple cookies
- Content-Type is required!
- Blank line ends the headers!

Netprog: CGI Sessions

25

C Example

```
printf("Content-Type: text/html\r\n");  
printf("Set-Cookie: prefs=nofrms\r\n");  
printf("Set-Cookie: Java=yes\r\n");  
printf("\r\n");
```

... now sends document content

Netprog: CGI Sessions

26

Getting Cookies

Drop by Dave's office anytime!

*If you want cookies, you might consider
bringing some with you...*

Netprog: CGI Sessions

27

Getting HTTP Cookies

- The browser sends each cookie as a header:

Cookie: prefs=nofrms

Cookie: Java=OK

- The Web server gives the cookies to the CGI program via an environment variable.

Netprog: CGI Sessions

28

Multiple Cookies

- There can be more than one cookie.
- The Web Server puts them all together like this:

prefs=nofrms; Java=OK

and puts this string in the environment variable: **HTTP_COOKIE**

Netprog: CGI Sessions

29

maybe a space, maybe not!

Cookie Limits

- Each cookie can be up to 4k bytes.
- One "site" can store up to 20 cookies on a user's machine.

Netprog: CGI Sessions

30

Cookie Usage

- Create a *session*.
- Track user browsing behavior.
- Keep track of user preferences.
- Avoid logins.

Netprog: CGI Sessions

31

Cookies and Privacy

- Cookies can't be used to:
 - send personal information to a web server without the user knowing about it.
 - be used to send viruses to a browser.
 - find out what other web sites a user has visited.*
 - access a user's hard disk

* although they can come pretty close to this one!

Netprog: CGI Sessions

32

Some Issues

- Persistent cookies take up space on user's hard disk.
- Can be used to track your behavior within a web site.
 - This information can be sold or shared.
- Cookies can be shared by cooperating sites (advertising agencies do this).

Netprog: CGI Sessions

33

Cookie Examples

- `showcookie.cgi`
 - sends back an HTML table that contains a list of the cookies sent.
 - also sends a form that tells the CGI what cookie we would like it to set.
- `pizzacookie`
 - pizza server that uses a cookie for propogating state information.

Netprog: CGI Sessions

34
