



**cookie**

## INTRODUCTION

- **HTTP cookies** are small text files containing a user's settings and other data used by websites, and are stored on the user's computer. They are transmitted as parcels of text sent by a server to a Web client and then sent back unchanged by client each time it accesses that server.
- Cookies are not computer programs that run on the browsing computer



- The pieces of information are stored as **name-value pairs**.
- For example, a Web site might generate a unique ID number for each visitor and store the ID number on each user's machine using a cookie file.
- The most common place for them to reside is in a directory called **c:\windows\cookies**.



# HOW DOES COOKIE DATA MOVE?

The data moves in the following manner:

- If you type the URL of a Web site into your browser, your browser sends a request to the Web site amazon.com
- It will look on your machine for a cookie file that Amazon has set. If it finds an Amazon cookie file, your browser will send all of the name-value pairs in the file to Amazon's server along with the URL
- Web server receives the cookie data and the request for a page.
- The server creates a new ID for you in database and then sends name-value pairs to your machine in the header for the Web page it sends. Your machine stores the name-value pairs on your hard disk.



# HOW DO WEB SITES USE COOKIES?

- There are other pieces of information that the server can send with the name-value pair. One of these is an **expiration date**
- A cookie allows a site to store **state information** on your machine
- Web sites use cookies in many different ways-- Sites can **accurately determine how many people actually visit the site.**
- Using cookies, sites can determine:
  - How many visitors arrive
  - How many are new versus repeat visitors
  - How often a visitor has visited



# CREATING A COOKIE

- to access the page `http://www.example.org/index.html`, browsers connect to the server `www.example.org` sending it a request that looks like the following one:
  - `GET /index.html HTTP/1.1`  
`Host: www.example.org`

**browser**  
**server**



- The server replies by sending the requested page preceded by a similar packet of text, called 'HTTP response'. This packet may contain lines requesting the browser to store cookies:

```
HTTP/1.1 200 OK
Content-type: text/html
Set-Cookie: name=value
(content of page)
```

**Browser**



**server**



GET /spec.html HTTP/1.1  
Host: www.example.org  
Cookie: name=value  
Accept: \*/\*

**Browser  
server**



## COOKIE ATTRIBUTES

- Beside the name/value pair, a cookie may also contain an expiration date, a path, a domain name
- For example, a cookie can be created by the server by sending a line `Set-Cookie: name=newvalue; expires=date; path=/; domain=.example.org.`
- The domain and path tell the browser that the cookie has to be sent back to the server when requesting URLs of a given domain and path.





- The expiration date tells the browser when to delete the cookie. If no expiration date is provided, the cookie is deleted at the end of the user session, that is, when the user quits the browser.
- The expiration date is specified in the "Wdy, DD-Mon-YYYY HH:MM:SS GMT" format. As an example, the following is a cookie sent by a Web server (the value string has been changed):

Set-Cookie: RMID=732423sdfs73242; expires=Fri, 31-Dec-2010 23:59:59 GMT; path=/; domain=.example.net



# APPLICATIONS

## **Session management**

- Cookies may be used to maintain data related to the user during navigation, possibly across multiple visits

## **Personalization**

- Cookies may be used to remember the information about the user who has visited a website in order to show relevant content in the future.

## **Tracking**

- Tracking cookies may be used to track internet users' web browsing habits.



- HTTP cookies are used for authenticating, session tracking and maintaining specific information about users



## SCOPE OF RESEARCH

- Cookies are automatically destroyed when the user closes the Web browser, and can be tampered and thus creating a security hole. How tempering can be avoided or limited up to some extant.

