Cookie技术及其应用介绍

目录

什么是Cookie

- Cookie是网站存放在客户端（浏览器）、并能在网站和浏览器之间传递的一小段数据
- Cookie只是一小段文本，它不是软件，不能被编程和执行，也不会携带病毒或恶意软件
- Cookie概念来自于“magic cookie”——在计算机领域中用来描述程序接受并原样发出的一组数据
- 也被称为HTTP Cookie、Web Cookie或者Browser Cookie

为什么需要Cookie

- HTTP协议是无状态的，浏览器对网站的每一次访问都是独立的事件
- Cookie产生的目的，就是为浏览器和Web服务器之间的交互提供一种状态管理机制

- 试想，如果没有Cookie
  - 用户每次访问都像第一次访问一样
  - 网站无法判断用户是否访问过，并区分不同的客户
  - 用户必须在一次访问中完成所有交互：选购商品、填写收货信息、购买、付款……

- Cookie的引入对于互联网来说是一个划时代的转折点，它将零散的访问、无状态的请求变得有序并富有记忆，极大提升了互联网的用户体验

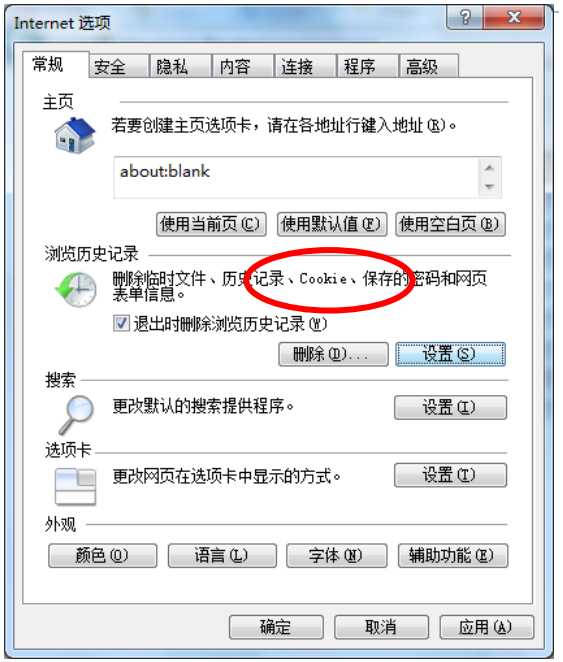
Cookie的发展历史

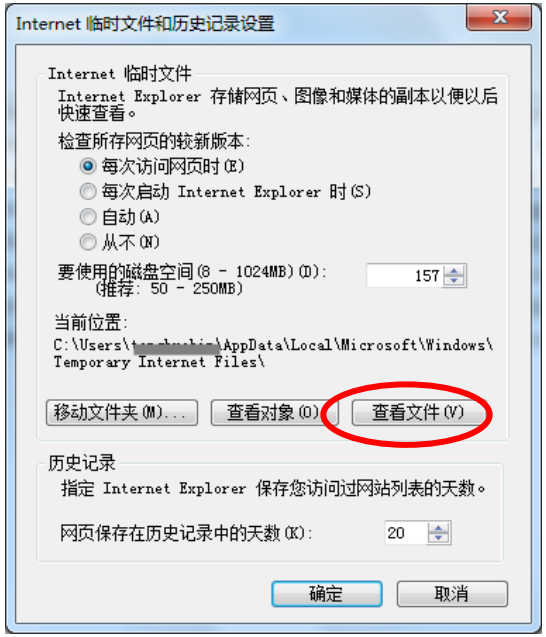
- 1994年，24岁的Netscape（网景）公司程序员Lou Montulli将Cookie概念引入Web网站
- 1994年10月，Netscape 0.9beta发布并首次支持Cookie，用于检查用户是否访问过Netscape网站
- 1995年，微软IE2开始支持Cookie
- 1995年，Netscape为Cookie申请专利（US 5774670），并于1998年获得授权
- 1996年2月，金融时报刊登了有关Cookie的文章，这一技术逐渐被公众所知
- 1995年，IETF成立了有关Cookie的特别工作组
- 1997年2月，Cookie技术规范HTTP State Management Mechanism发布（RFC 2109，后来更新为2000年RFC 2965 、2011年RFC 6265）

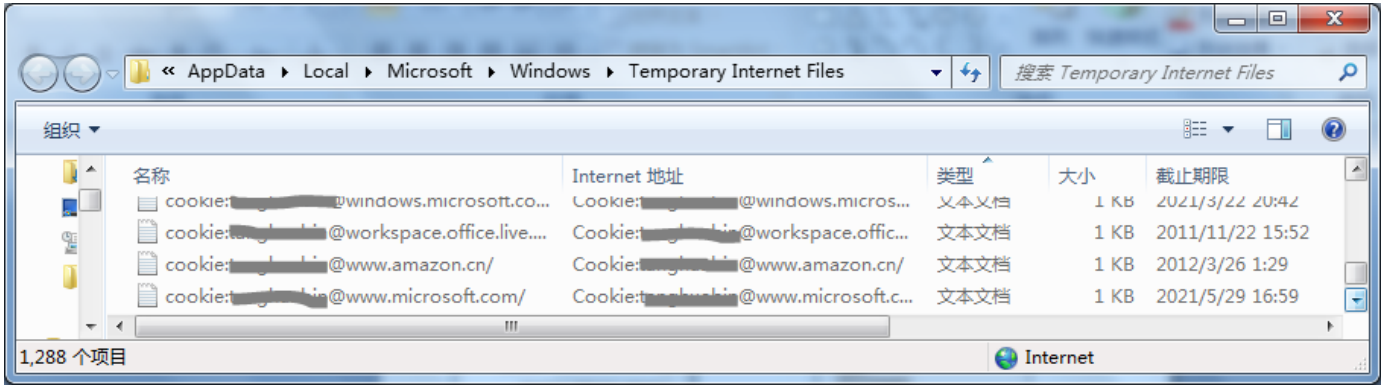
Cookie的主要用途

- 会话管理
  - 记录用户的登录状态是cookie的常见用途，免去用户多次认证和登录网站
  - 记录用户的访问状态和数据，特别是在需要多次访问的交互流程中，典型应用的是用户的“购物篮”
- 个性化
  - 用来记忆访问过网站的用户相关信息，以便下次为用户显示。例如Gmail会记忆上一次用户登录的用户名，并在下次访问时默认填写好这个用户名
  - 用来记忆用户自定义的一些功能，以便下一次访问时服务器能根据用户本地的cookie来进行配置。例如Google通过Cookie获取用户搜索设置（使用语言、每页的条数，以及打开搜索结果的方式等等）
- 记录和跟踪用户行为
  - 例如网站统计访问用户数、访问不同页面的情况等
  - 跟踪用户的网站访问习惯并投放广告

Cookie在哪里





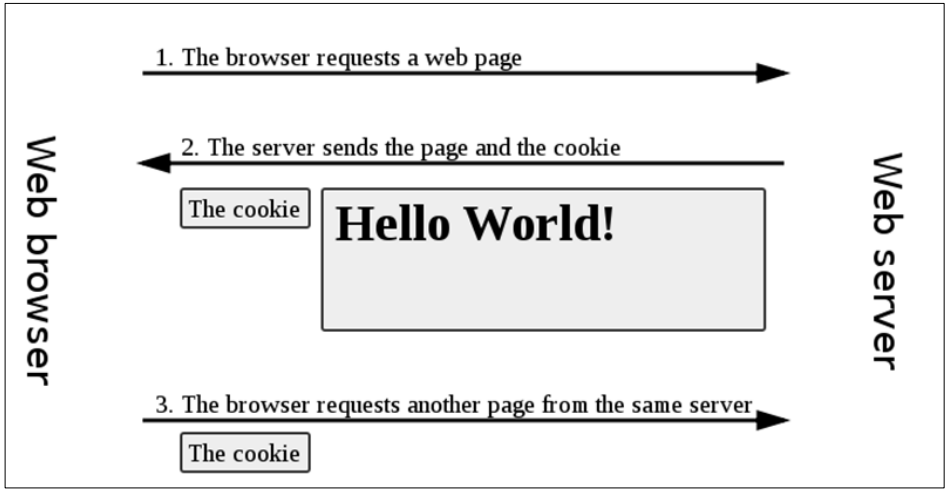


以Window7系统，IE浏览器为例

目录

Cookie的传送

- Cookie通过HTTP协议在浏览器和Web服务器之间传递



Step1.浏览器发起HTTP Request到Server

```
GET /index.html HTTP/1.1
Host: www.example.org
```

browser -------→ server

Step2.服务器返回HTTP Response，其中可以包含Cookie设置

```
HTTP/1.1 200 OK
Content-type: text/html
Set-Cookie: name=value
Set-Cookie: name2=value2; Expires=Wed, 09 Jun 2021 10:18:14 GMT

(content of page)
```

browser ←------- server

Step3.浏览器后续访问，可能携带Cookie

```
GET /spec.html HTTP/1.1
Host: www.example.org
Cookie: name=value; name2=value2
Accept: */*
```

browser -------→ server

Cookie属性

- Name-Value属性
  - 必须指定，其格式为Name=Value

- 其它属性
  - Domain和Path：设置Cookie作用的域名和路径
  - Expires或Max-Age：Cookie的过期时间或者最长寿命
  - Secure：设定Cookie只用于HTTPS安全连接
  - Httponly：设定Cookie只能用于HTTP协议，而不能通过非HTTP方法（例如JavaScript）使用

在以上属性中，只有Name-Value会由浏览器传送给Web服务器，其它属性只是告诉浏览器何时删除Cookie、是否允许或阻止Cookie发送等

Cookie属性：Domain和Path

- Domain和Path用于定义Cookie的生效作用域
- 只有当域名和路径同时满足的时候，浏览器才会将Cookie发送给Server
- 如果未设置Domain和Path属性，则默认为当前请求页面对应值
- 当指定Domain为父域名时，其作用域包括子域名
  - Domain举例：.10086.cn（顶级域名、父域名）， www.10086.cn（二级域名、子域名）
- 当指定Path为根路径（/）时，其作用域为所有路径
  - Path举例： / ， /service，/brand

第三方Cookie
如果用户访问一个网站www.a.com，但被设置了Domain为www.b.com的Cookie，则将这种Cookie称为“第三方Cookie”

Cookie属性： Expires或Max-Age

- 用于通知浏览器何时删除Cookie
- Expires的规定格式是：“Wdy, DD-Mon-YYYY HH:MM:SS GMT”
  - 如果不指定Expires时间，则此Cookie为会话型Cookie
  - 如果指定了一个过去的时间，则立即删除Cookie
- Max-Age：一个整数，表示Cookie设定多少秒后删除（RFC 6265新增的替代方案）

Session Cookie（会话型Cookie），也称为临时型Cookie，只在会话期间内有效，它保存在内存中，当浏览器关闭时即被清除
Persistent Cookie（持久型cookie），在设定的时间内长期有效，它保存在客户端计算机的磁盘文件中，即使浏览器关闭仍然存在

Cookie的创建

- 两种创建方式：
  - 在服务器端设定，并通过HTTP Response消息的Header域发送到浏览器（例如通过PHP脚本）
  - 在浏览器允许的情况下，可由客户端脚本（如JavaScript）创建
- Cookie创建示例

A. Set-Cookie: lu=1295214458; Path=/; Domain=.foo.com
B. Set-Cookie: LSID=DQAAAKEaem_vYg; Domain=docs.foo.com; Path=/accounts; Expires=Wed, 13-Jan-2021 22:23:01 GMT; Secure; HttpOnly

- 考虑以下三次URL访问，其中1会带上CookieA、B；2、3会带上CookieA

1. http://docs.foo.com/accounts/index.html
2. http://docs.foo.com/accountstest.html
3. http://www.foo.com

Cookie的限制

- Cookie技术规范建议了浏览器至少应该支持的Cookie数量
  - 总计至少 300 个 Cookie
  - 每个域至少 20 个 Cookie
    - 大部分浏览器支持更多，例如，IE8：50个，Firefox：150个，Chrome：170个
  - 每个 Cookie 4096字节
    - 不同浏览器基本与此接近，有少量差别

- 对超过限制的处理
  - 在所有浏览器中，任何cookie字节大小超过限制都会被忽略，且永远不会被设置
  - 创建的cookie数量太多会造成cookie被覆盖或清除掉，两种策略
    - 自动剔除最早的 Cookie，如IE
    - 保存最后设置的 Cookie，随机决定删除原先的 cookie ，如Firefox

目录

Cookie的缺陷

- 可能识别不准确
  - 多人共用一台电脑时
  - 一人使用多台电脑时
  - 一人使用多种浏览器时
  - Cookie被删除时
- 可能造成浏览器和服务器端的状态不一致
  - 浏览器后退cookie不会重置
- 过多使用导致HTTP请求流量浪费
- 隐私和安全问题

Cookie的可能替代方案

- IP地址
  - 对于某些IP地址固定的用户，可以通过IP地址进行跟踪
- URL + Query String
  - 例如：http://www.google.com/search?gcx=c&sourceid=chrome&ie=UTF-8&q=Cookie
  - 当Cookie被禁用时，Java Servlet和PHP的会话管理机制都采用这种方法
  - 但Query方法容易导致访问被篡改，导致部分安全漏洞被利用
- 隐藏的HTTP Form域
  - 功能与Query String类似
  - 通常采用POST方法提交

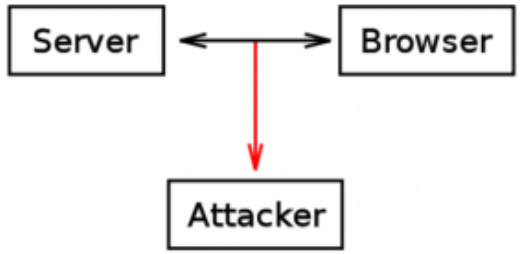
```
<form action="cgi-bin/test.cgi" method="post">
<input type="text" name="user">
<input type="hidden" name="lang" value="cn">
<input type="submit">
</form>
```

Cookie隐私问题

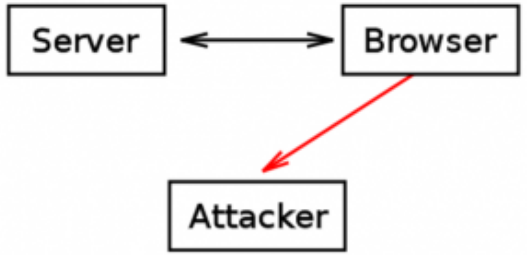
- 主要表现
  - Cookie这种在用户不知道的情况下默认设置的方式让人不满
  - 互联网广告的发展，某些网站滥用来追踪用户访问网站的行为，特别是跨站点的第三方Cookie
  - 由于网站开发者的失误，在浏览器端的Cookie中保存了用户的敏感信息
- 针对隐私的对策
  - 早在1996年-1997年，美国联邦贸易委员会听证会就专门讨论了Cookie可能带来的隐私问题
  - IETF在讨论Cookie规范时，要求至少默认不打开对第三方Cookie的支持
  - 2000年以后，美国、欧盟等都制订了关于Cookie使用的严格规定，包括明确告知用户收集的信息及其用途
  - W3C开始推进P3P等协议，用以让用户获得更多控制个人隐私权利的协议，目前大多数浏览器已经支持P3P

Cookie安全问题

- **网络监听**
  - 大多数网站都是把Cookie当作用户的唯一标识，在这种情况下，黑客可能通过窃取用户的cookie来模拟用户的请求行为
  - 应对办法：采用HTTPS连接，利用IP地址等进行辅助认证



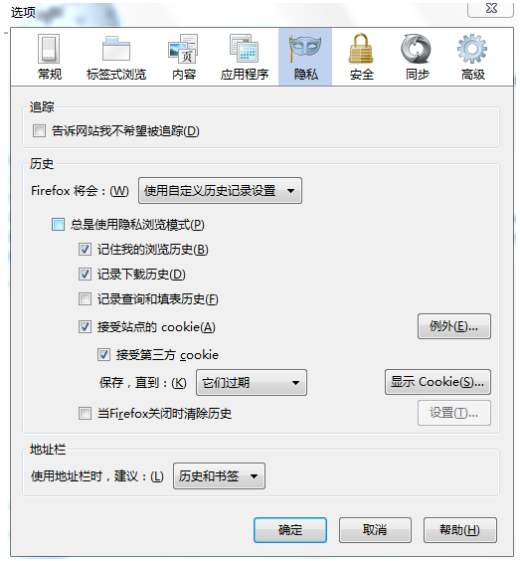
- DNS漏洞
  - 利用DNS cache、修改hosts文件或者DNS服务商的一些漏洞问题，将用户导向黑客的IP，读取并设置Cookie
  - 解决办法：1.减少dns无效配置 2.ISP服务商加强自我安全管理。3.通过HTTPS请求对请求进行加密和授权



- **跨站脚本**
  - 利用Javascript读取页面内的Cookie，并发送给黑客站点
  - 解决办法：Web开发者（特别是某些论坛）有责任去过滤掉恶意代码，或采用HTTPOnly Cookie。用户在访问某些可能有恶意脚本的站点时，可禁用JavaScript

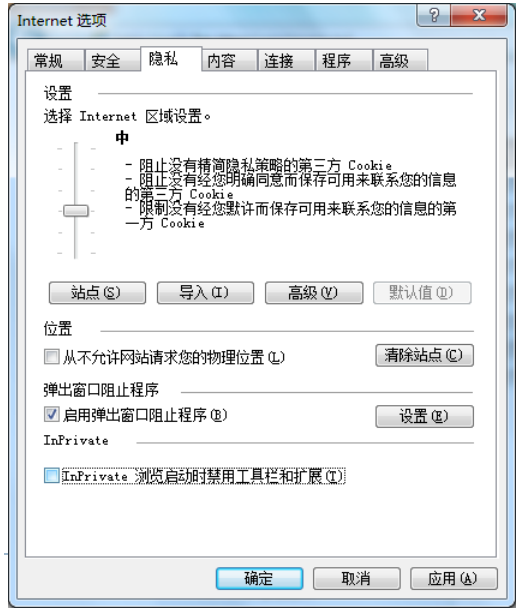
浏览器对Cookie的控制

- 控制措施
  - 禁用所有Cookie
  - 禁用第三方Cookie
  - 允许或禁止部分站点的Cookie
  - 仅允许会话Cookie
  - 关闭浏览器时删除Cookie
- 禁用Cookie可能造成
  - 某些操作，例如登录无法完成
  - 网站访问失败或者出错



Firefox

Chrome



IE