Cookies

Troy Hunt troyhunt.com @troyhunt





Outline

- Cookies 101
- Understanding HttpOnly cookies
- Understanding secure cookies
- Restricting cookie access by path
- Reducing risk with cookie expiration
- Using session cookies to further reduce risk

Cookies 101

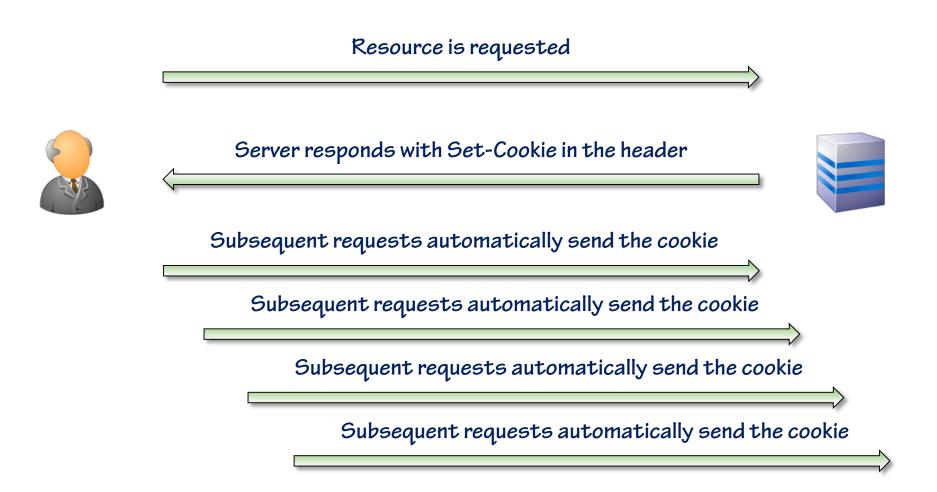
Cookies are nothing more than simple pieces of text stored in the browser:

```
Set-Cookie: name=value;
[page contents]
```

- The server may set a cookie via the HTTP response header...
- ...or it may be set (and read) via JavaScript directly in the DOM
- Cookies are automatically passed back to the website in the header of each request:

```
GET http://hackyourselffirst.troyhunt.com/ HTTP/1.1
Cookie: name=value;
```

Understanding the HTTP cookie exchange



Cookie security

- Cookies frequently contain data of a sensitive nature
 - We saw this with the auth cookie in the last two modules
- Browsers implement native defences to protect cookies to some degree
 - But this can still be exploited by attacks such as XSS
- Cookies can be further secured by understanding and tuning their attributes

Cookie attributes

- Domain
- Path
- Expiration
- HttpOnly
- Secure

```
Set-Cookie: name=value;
Domain=hackyourselffirst.troyhunt.com;
Path=/;
Expires=Sat, 10-Aug-2013 07:12:19 GMT;
HttpOnly;
Secure;
```

Summary

- Cookies are a fundamentally simple concept yet they are frequently configured in a sub-optimal fashion in terms of security
 - Default framework configurations can often exacerbate this
- HttpOnly is absolutely essential if the cookie isn't required to be accessed via client script
- Any cookie holding data of a sensitive nature should always be marked as "secure"
 - Don't allow important cookies like auth cookies to travel over HTTP connections
- Consider the path scope of a cookie; can you limit the important ones?
- Try and expire cookies as quickly as possible
 - Short expirations or session cookies are ideal...
 - ...but consider the adverse impact to usability and strike a happy balance