Client-Side HTTP Cookie Security: Attack and Defense

David Wyde
DEF CON 22
Game Plan

• Why are HTTP cookies valuable to attackers?
• How do popular web browsers store cookies?
• How can cookies be stolen?
• How can cookies be protected?
Disclaimers

• The opinions in this presentation are mine, and not my employer’s.

• The security issues I discuss are not specific to any one website, and are not vulnerabilities in the conventional sense.
What is an HTTP Cookie?

- Cookies are transmitted as HTTP headers
  - Name-value pairs

- HTTP clients store state using cookies
  - E.g., trade credentials for a session cookie
Cookies in Action

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Domain</th>
<th>Path</th>
<th>Expires / Max-Age</th>
<th>Size</th>
<th>HTTP</th>
<th>Secure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request Cookies</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Response Cookies</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>454</td>
<td></td>
<td></td>
</tr>
<tr>
<td>_twitter_sess</td>
<td>BAh7CSIKZ...</td>
<td>.twitter.com</td>
<td>/</td>
<td>Session</td>
<td>347</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>goth</td>
<td>1</td>
<td></td>
<td></td>
<td>Session</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>guest_id</td>
<td>v1%3A1402...</td>
<td>.twitter.com</td>
<td>/</td>
<td>Mon, 13 Jun 2016 19:16:27 GMT</td>
<td>100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
User-Readable Data

• Any process that runs as your user can read:
  • Your private keys
  • Some software saves passwords as plaintext
  • Web browser cookies

• Damage is done without privilege escalation
Cookies Are Valuable to Attackers

- Cookies can be more valuable than passwords
  - Gmail: bypass two-factor authentication
  - Facebook: don't warn of login from a new device

- Counterpoints
  - "Please re-enter your password"
  - Cookies expire
Gmail: Two-Factor Authentication

2-Step Verification

Enter the verification code generated by your mobile application.

Enter code

Verify

Don't ask for codes again on this computer
Hi David,

We detected a login into your account from a new device named "Chrome on Linux" on Sunday, July 6, 2014 at 8:52pm. This device has been added to your account.

Operating System: Linux
Browser: Chrome
Location: Austin, TX, US (IP=[REDACTED])

Note: Location is based on internet service provider information.

If this was you, please disregard this email.
If this wasn't you, please secure your account, as someone else may be accessing it.

Thanks,
The Facebook Security Team

Please note: Facebook will never request your login information through email.
Browser Cookie Storage
Cookie Storage: Intro

- Almost all browsers store cookies as plaintext
- The HttpOnly and Secure flags apply inside browsers
  - Malware need not respect them
Firefox

- Stores cookies in an SQLite database
  - Cookies can be read using sqlite3, Python, etc.
$ sqlite3 ~/Library/Application\ Support/Firefox/Profiles/*/cookies.sqlite
SQLite version 3.7.13 2012-07-17 17:46:21
Enter ".help" for instructions
Enter SQL statements terminated with a ";"

```
sqlite> .schema
CREATE TABLE moz_cookies (id INTEGER PRIMARY KEY, baseDomain TEXT, appId
INTEGER DEFAULT 0, inBrowserElement INTEGER DEFAULT 0, name TEXT, value
TEXT, host TEXT, path TEXT, expiry INTEGER, lastAccessed INTEGER,
creationTime INTEGER, isSecure INTEGER, isHttpOnly INTEGER, CONSTRAINT
moz_uniqueid UNIQUE (name, host, path, appId, inBrowserElement));
CREATE INDEX moz_basedomain ON moz_cookies (baseDomain, appId,
inBrowserElement);

sqlite> SELECT value FROM moz_cookies WHERE name='GX';
```

DQAAAPEAAABWYmsr2PFvwQi4XhQWYycw_5coZVfjh-efmKTNeljyLx04sHi_Ih-
xMOsSRaZ6J38QzDGYct5v6DKYkkoc6TeX8QKuaOPSaqGTEo4v2Y6kvzmzlS-SvdU4zTcuJ-
z4uCF7uiZ7ic-
H6USM5t7leqmsDhQeEoL01z5OF6iLoxUeCHU_91eWrA2bOpU8ppqVjutpi4WVhyqLV7WX6hgSnE
kWnpsN-XwcDF84V7u0Dr1KCQFupzmCfa3nt_tARY-SxbyNrmY_0rH4YF-
xBVvPFXBQpKqUZrW_zMdGmWgmPER_7mBTGXt1h9PM5nP_bW09oIQXrQb_OhHe7c3AnnIg2EIq-
g
Internet Explorer

- Stores cookies as text files
- The folder varies depending on IE version
- Filenames are random: need to read the files
Reading Internet Explorer Cookies

![Image of IE cookies]

- **utma**
  - chromium.org
  - 1600
  - 1463462656
  - 30503714
  - 1707109413
  - 30356863

- **utmb**
Opera and Safari

- Custom binary formats
  - Can be parsed by free software tools
- Safari: Cookies.binarycookies
- Opera: cookies4.dat
$ python ~/Desktop/BinaryCookieReader.py ~/Library/Cookies/Cookies.binarycookies | grep yahoo

Cookie: hpc=d=ItIKgZXDu9Pkv2_sEb7ygoVyN9bHZ2mmjnr8eBC8z9Ynw88Tayw7ixgQfT4vleMQ56bGUussxMNmYBusbq3RHgXIkea3DhM.Yzckc.y6GAQEiJoPoK1DzyvYg1cyBoMWlZccOkvv7wvPUmDHnNk1uyiJwon3_YjfMMyCXstKdmUKmePy_Wn04tFoVbui1wlLTuSpqTw-&v=2; domain=.www.yahoo.com; path=/; expires=Wed, 15 Jul 2015;

Cookie: B=2b26v3t9s955p&b=3&s=oh; domain=.yahoo.com; path=/; expires=Fri, 15 Jul 2016;

Cookie: CRZY=%7B%221048616551%22%3A%7B%22expires %22%3A1405564858541%2C%22data%22%3A%7B%22%22nv%22%22%3A1%2C%22bn%22%3A0%7D%7D%7D; domain=.yahoo.com; path=/; expires=Thu, 17 Jul 2014;
Reading Opera Cookies

$ python opera_reader.py ~/.opera/cookies4.dat

file_version_number 4096
app_version_number 8193
idtag_length 1
length_length 2
domain record
    [('0x1e', 'name of the domain part', 3, 'org')]
end of path record
domain record
    [('0x1e', 'name of the domain part', 8, 'slashdot')]
cookie record
    [('0x10', 'name of the cookie', 6, '__gads'), ('0x11', 'value of the cookie', 69, 'ID=2628549bf6c27042:T=1405392507:S=ALNI_Maix2zTTIQ4159AfUM0tHp7h_ODgQ'), ('0x12', 'expiry', 8, '2016-07-13 21:48:27'), ('0x13', 'last used', 8, '2014-07-14 21:49:28'), ('0x28', 'unknown cookie data id', 8, ''), ('0xa9', 'unknown cookie data id', 0, '')]
Chromium

- Encrypts cookies in recent versions
  -Implementation and security vary by platform

- Stores cookies in an SQLite database
  - BLOB field for encrypted cookie values
Chromium on Linux

- Linux has no single standard keyring mechanism
  - (KDE, Gnome, etc.)

- Cookies encrypted with AES (symmetric key)
  - Hard-coded key and salt

- Can be decrypted on any machine
  - Link against Chromium libs, call code to decrypt
Reading Chromium Cookies: Linux

```
david@computer /d/code/snickerdoodle/chromium-linux $ LD_LIBRARY_PATH=/d/code/lib/chromium/src/out/Release/lib ./db_reader
Opened DB.
Cookie: GX
Blob prefix: v10

Blob length: 371

DQAAAP0AAAC0gP2ud0jJGNys2yxG2IA2onom9AydcHeMcOewTg3loCkSSFUUnmRgOHe0FJv2YuFYnt6B1yL4BRuK92fZ5dY15frLwGyOFnfOGoEBt64tb471eIhHskxB8DMuXCTu2AyeyJWHbh5Hr2ixjtysqUR48k-u0m9MRTmho8PVTwhSsqZGJtXg2uwvFNhAUhPuujQkCnobJTYs2Pc5jP0QALJ2dHqCopxu5L0qYqOLng_pIm_DcSymgad1NF_waZb0GGDI3yd9ogfS6ajyhlAVRS-1k8K20JjdfmEZlSGLa8SyusXwKUuxgAKf9dcFqbYPWluYql-1oakDTlssLd0Ej6C1gDa
```
david@localhost Desktop]$ python chromium_b64_cookie_linux.py

djEwXgab42ZPnVqGRirZqEHsvEN8bC/
chT84CbmJxMSJDr6XA7mQLZdCuLwYSNA6srVf7NDn7rHdBOFJf8SX4jdCx1QhcrUGH
+0KzFz
+hUXUcgRzy6jWEZyYe4QDegh1YGtfdCGiZ2TgHkEifJ0Mojf4VpuKhFw7SVpCzCorz86JF
czNpco7LZwM/xng7UPmVEY4sIQwAG1TXoY9ThgaliP8HGviwkK0ozW9/FMUiGaxBIqDD
+FSfsGszckv9zRbK8XL2PbHVslRmG2ENQ8wESu2Czajb20BQ+L3dMRv0cVbW+gwt+H/
cBG23dnjnhFxGcvm9DSDyz87o5ssILocgMT+kddTBCG8ohvy7iNE3njT6WOFktK8Hd/
+rhSUarnCtZt9UB1EZtikWbpqn0PKrVCKn0wVpO4oyeD1e96xEesn/IM=

david@computer /d/code/snickerdoodle/chromium-linux $ 
LD_LIBRARY_PATH=/d/code/lib/chromium/src/out/Release/lib ./
base64_reader $ (cat fedora-cookie.txt)

DQAAANMAAAD55DvOAnmlugeHzwGKs0asFxYtMfX1-
Xdg7MtLYmdj5GDI3iyPh70Ds6OKgogfATna2KV9d7JqX5e7SA-
shH1oxvQFs1WsFo_9WzEfj9VamEV5C0uml6tVuzhIGzrrKM0__0SI6QANb-y-
qyM3QJSDKCB7QrXR_Ug71FzjibDW7Fsfg15SUCTmfQz9YLBp4oYSOt_pJRVf5XZgbN_2J-
KQzBqtZznZwKVE4TatBaAucT-
R9jXnjM5aMdoJvr7ubghi0p1m7yvPevqNNRItPkeB5aV_cPXHKRMjwhAAk6_2w
Chromium on Windows

- `CryptProtectData` is used to encrypt
  - A Windows cryptography API
  - Uses login credentials as part of the encryption

- `CryptUnprotectData` is used to decrypt
  - Must be called by the user that encrypted, on the same machine
Chromium on Mac

- Store an encryption key in the system keychain
  - If no key exists, a random one is generated
- AES is used to encrypt/decrypt
- Keychain prompts when accessed from unsigned apps
keychain-reader wants to use your confidential information stored in “Chrome Safe Storage” in your keychain.

The authenticity of “keychain-reader” cannot be verified. Do you want to allow access to this item?

[Buttons: Always Allow, Deny, Allow]
Browser Cookie Storage: Summary

- Chromium encrypts cookies on Windows and Mac
- Chromium obfuscates cookies on Linux
- Other popular browsers store cookies as plaintext
Attack Vectors
Physical Access

- Cookies are there for the taking with most browsers
- Chromium protects you on Windows and Mac
Social Engineering

- Excel/Word macros
- Malicious executables
- Don't need to install anything - just run once
Malware

• Drop and run an executable to extract cookies
  • Metasploit
  • Any process that runs as your user

• HTTP POST cookies to a malicious server
Proof of Concept
Proof of Concept: Login

Google Mail and Facebook login screens are displayed.
Defenses
Disk Encryption

- Protect against physical access to plaintext cookies
Application Firewalls

- Block/allow (server, port) pairs for each application
  - Chromium can access www.google.com on port 443

- Examples
  - Mac: Little Snitch
  - Windows: NetLimiter?
  - Linux: ?
Little Snitch

```
computer:~ dwyde$ curl -I reddit.com
```

**Terminal via curl**

wants to connect to **reddit.com** on port 80 (http)

- [ ] Forever
- [X] Once

- [ ] Any Connection
- [ ] Only port 80 (http)
- [ ] Only reddit.com
- [X] Only reddit.com and port 80 (http)

[Deny] [Allow]
SELinux

- Security-Enhanced Linux
- Separate from standard Unix permissions
- Can isolate a user’s applications from each other
Idea: Master Password for Cookies

- Type in a password to decrypt your cookies
- Firefox has this to protect passwords
Firefox: Master Password

Password Required
Please enter the master password for the Software Security Device.

[Input field]

[Buttons: Cancel, OK]

[Options: Remove, Remove All, Show Passwords]
Server-Side Defenses

• Tie a session cookie to the login IP
  • The cPanel web hosting tool can optionally enforce this
  • Kind of annoying in a world of mobile clients

• Warn users, rather than force them to log in again
  • “You’ve logged in from X and Y countries this month”
Conclusions

- Cookies should be handled with care
- Client-side cookie security is not a solved problem
References

- Opera reader: https://gist.github.com/gwarser/1324501#file-readcookies-py

- Safari reader: http://www.securitylearn.net/2012/10/27/cookies-binarycookies-reader/

- Firefox master password: http://kb.mozillazine.org/Master_password

- cPanel cookie IP validation: http://www.cpanelkb.net/cpanel-security-settings-checklist/