Client Identification and Cookies

سرورهای وب ممکن است با هزاران کلاینت مختلف به طور همزمان صحبت کنند. این سرورها اغلب به جای تلقی کردن همه درخواست‌ها به‌عنوان کلاینت‌های ناشناس نیاز دارند تا متوجه شوند که با چه کسی صحبت می‌کنند. این فصل به برخی از فناوری‌هایی می‌پردازد که سرورها می‌توانند برای شناسایی افرادی که با آن‌ها صحبت می‌کنند استفاده کنند.

The Personal Touch

HTTP زندگی خود را به عنوان یک پروتکل Request/Response ناشناس، بدون وضعیت (Stateless)، آغاز کرد. درخواستی از یک کلاینت می‌آید، توسط سرور پردازش می‌شود و یک پاسخ به کلاینت ارسال می‌شود. اطلاعات کمی برای تعیین اینکه چه کاربری درخواست را ارسال کرده یا برای پیگیری دنباله ای از درخواست‌های کاربر بازدید کننده در دسترس وب سرور بود.

وب سایت‌های مدرن می‌خواهند که یک تماس شخصی (Personal Touch) ارائه دهند. آن‌ها می‌خواهند درباره کاربرانی که در انتهای دیگر اتصالات قرار دارند بیشتر بدانند و بتوانند آن کاربران را هنگام Browse ردیابی کنند. سایت‌های خرید آنلاین محبوب مانند Amazon.com سایت‌های خود را به چند روش برای شما شخصی سازی می‌کنند:

Personal greetings

پیام‌های خوش آمدگویی و محتویات صفحه به طور ویژه برای کاربر ایجاد می‌شوند تا تجربه خرید را شخصی‌تر کنند.

Targeted recommendations

فروشگاه‌ها با آگاهی از علایق مشتری می‌توانند محصولاتی را پیشنهاد دهند که مشتری از آن‌ها استقبال خواهد کرد. فروشگاه‌ها همچنین می‌توانند تخفیف‌های ویژه تولد را در نزدیکی تولد مشتریان و سایر روزهای مهم اجرا کنند.

Administrative information on file

خریداران آنلاین از پر کردن فرم‌های دست و پا گیر آدرس و کارت اعتباری بارها و بارها متنفرند. برخی از سایت‌ها این جزئیات را در یک پایگاه داده ذخیره می‌کنند. هنگامی که آن‌ها شما را

شناسایی کردند، می‌توانند از اطلاعات موجود استفاده نموده و تجربه خرید را بسیار راحت‌تر کنند.

Session tracking

تراکنش‌های HTTP بدون وضعیت یا Stateless هستند. هر Request/Response به صورت مجزا انجام می‌شود. بسیاری از وب‌سایت‌ها می‌خواهند هنگام تعامل با سایت، حالت افزایشی ایجاد کنند (به عنوان مثال، پر کردن یک سبد خرید آنلاین). برای انجام این کار، وب‌سایت‌ها به راهی برای تشخیص تراکنش‌های HTTP از کاربران مختلف نیاز دارند.

این فصل تعدادی از تکنیک‌های مورد استفاده برای شناسایی کاربران در HTTP را خلاصه می‌کند. خود HTTP با مجموعه‌ای غنی از ویژگی‌های شناسایی متولد نشده است. طراحان اولیه وب سایت، فناوری‌های خود را برای شناسایی کاربران ساختند. هر تکنیکی نقاط قوت و ضعف خود را دارد. در این فصل، مکانیسم‌های زیر را برای شناسایی کاربران مورد بحث قرار خواهیم داد:

هدرهای HTTP که حاوی اطلاعاتی در مورد هویت کاربر هستند.

- ردیابی آدرس IP کلاینت، برای شناسایی کاربران با آدرس IP آن‌ها
- ورود کاربر، با استفاده از احراز هویت برای شناسایی کاربران
- Fat URL ها، تکنیکی برای جاسازی هویت در URL ها
- کوکی‌ها، یک تکنیک قدرتمند اما کارآمد برای حفظ هویت پایدار

HTTP Headers

جدول زیر هفت هدر درخواست HTTP را نشان می‌دهد که معمولاً اطلاعات مربوط به کاربر را حمل می‌کنند. اکنون سه مورد اول را مورد بحث قرار خواهیم داد. چهار هدر آخر برای تکنیک‌های شناسایی پیشرفته‌تر استفاده می‌شوند که بعداً در مورد آن‌ها صحبت خواهیم کرد.

Header name	Header type	Description
From	Request	User's email address
User-Agent	Request	User's browser software
Referer	Request	Page user came from by following link
Authorization	Request	Username and password (discussed later)
Client-ip	Extension (Request)	Client's IP address (discussed later)
X-Forwarded-For	Extension (Request)	Client's IP address (discussed later)
Cookie	Extension (Request)	Server-generated ID label (discussed later)

هدر From حاوی آدرس ایمیل کاربر است. در حالت ایده آل، این یک منبع قابل دوام برای شناسایی کاربر خواهد بود، زیرا هر کاربر یک آدرس ایمیل متفاوت خواهد داشت. با این حال، تعداد کمی از مرورگرها به دلیل نگرانی در مورد سرورهای مخرب که آدرس‌های ایمیل را جمع‌آوری می‌کنند و از آن‌ها برای توزیع نامه‌های ناخواسته استفاده می‌کنند، هدرهای From ارسال می‌کنند. در عمل، هدرهای From توسط ربات‌های خودکار یا Spider ها ارسال می‌شوند، به طوری که اگر چیزی اشتباه شود، یک وب‌مستر جایی برای ارسال شکایات ایمیلی را خواهد داشت.

هدر User-Agent اطلاعات مربوط به مرورگری که کاربر از آن استفاده می‌کند، شامل نام و نسخه برنامه و اغلب اطلاعات مربوط به سیستم عامل را به سرور ارسال می‌کند. این موضوع گاهی اوقات برای سفارشی کردن محتوا برای تعامل خوب با مرورگرهای خاص و ویژگی‌های آن‌ها مفید است، اما این کار کمک زیادی به شناسایی کاربر خاص به هیچ وجه معنی دار نمی‌کند. در اینجا دو عنوان User-Agent وجود دارد که یکی توسط Netscape Navigator و دیگری توسط Microsoft Internet Explorer ارسال شده است:

Navigator 6.2

```
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:0.9.4) Gecko/20011128
  Netscape6/6.2.1
```

Internet Explorer 6.01

```
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
```

هدر Referer URL صفحه‌ای را که کاربر از آن آمده است را ارائه می‌دهد. هدر Referer به تنهایی و مستقیماً کاربر را شناسایی نمی‌کند، اما نشان می‌دهد که کاربر قبلاً از چه صفحه‌ای بازدید کرده است. می‌توانید از این برای درک بهتر رفتار مرورگر کاربر و علایق کاربر استفاده کنید. برای مثال، اگر به وب سروری رسیدید که از یک سایت بیسبال می‌آید، سرور ممکن است استنباط کند که شما طرفدار بیسبال هستید.

هدرهای From، User-Agent و Referer برای اهداف شناسایی قابل اعتماد، کافی نیستند. بخش‌های باقی‌مانده طرح‌های دقیق‌تری را برای شناسایی کاربران خاص مورد بحث قرار می‌دهند.

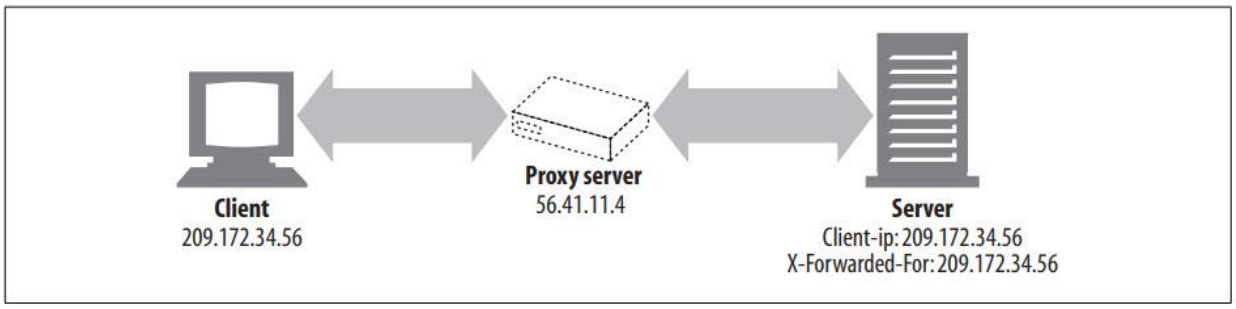
Client IP Address

پیشگامان اولیه وب سعی کردند از آدرس IP کلاینت به عنوان نوعی شناسایی استفاده کنند. این طرح در صورتی کار می‌کند که هر کاربر یک آدرس IP مجزا داشته باشد، اگر آدرس IP به ندرت (اگر همیشه) تغییر کند و اگر وب سرور بتواند آدرس IP کلاینت را برای هر درخواست تعیین کند. در حالی که آدرس IP کلاینت معمولاً در هدرهای HTTP وجود ندارد، وب سرورها می‌توانند آدرس IP طرف دیگر اتصال TCP را که درخواست HTTP را حمل می‌کند، پیدا کنند. به عنوان مثال، در سیستم‌های یونیکس، فراخوانی تابع getpeername آدرس IP کلاینت دستگاه فرستنده را برمی‌گرداند:

```
status = getpeername(tcp_connection_socket,...);
```

متأسفانه، استفاده از آدرس IP کلاینت برای شناسایی کاربر دارای نقاط ضعف متعددی است که اثربخشی آن را به عنوان یک فناوری شناسایی کاربر محدود می‌کند:

- آدرس‌های IP کلاینت فقط کامپیوتر مورد استفاده را توصیف می‌کند نه کاربر را. در این صورت اگر چندین کاربر از یک کامپیوتر مشترک استفاده کنند، قابل تشخیص نیستند.
- بسیاری از ارائه دهندگان خدمات اینترنتی به صورت پویا آدرس‌های IP را به کاربران هنگام ورود به سیستم اختصاص می‌دهند. هر بار که وارد می‌شوند، آدرس متفاوتی دریافت می‌کنند، بنابراین سرورهای وب نمی‌توانند فرض کنند که آدرس‌های IP، کاربر را در جلسات ورود شناسایی می‌کند.
- برای افزایش امنیت و مدیریت آدرس‌های کمیاب، بسیاری از کاربران، به اینترنت از طریق فایروال‌های ترجمه آدرس شبکه (NAT) دسترسی پیدا می‌کنند. این دستگاه‌های NAT آدرس‌های IP کلاینت‌های واقعی پشت فایروال را پنهان می‌کنند و آدرس IP کلاینت واقعی را به یک آدرس IP مشترک فایروال (و شماره پورت‌های مختلف) تبدیل می‌کنند.
- پروکسی‌ها و Gatway های HTTP معمولاً اتصالات TCP جدید را به سرور مبدا باز می‌کنند. وب سرور به جای آدرس سرویس گیرنده، آدرس IP سرور پروکسی را می‌بیند. برخی از پراکسی‌ها با افزودن هدرهای Client-ip یا X-Forwarded-For HTTP برای حفظ آدرس IP اصلی سعی در حل این مشکل دارند (شکل زیر). اما همه پروکسی‌ها از این رفتار پشتیبانی نمی‌کنند.



برخی از وب سایت‌ها هنوز از آدرس‌های IP کلاینت برای پیگیری کاربران در بین جلسات استفاده می‌کنند. مکان‌های زیادی وجود دارد که هدف گذاری آدرس IP به خوبی کار نمی‌کند.

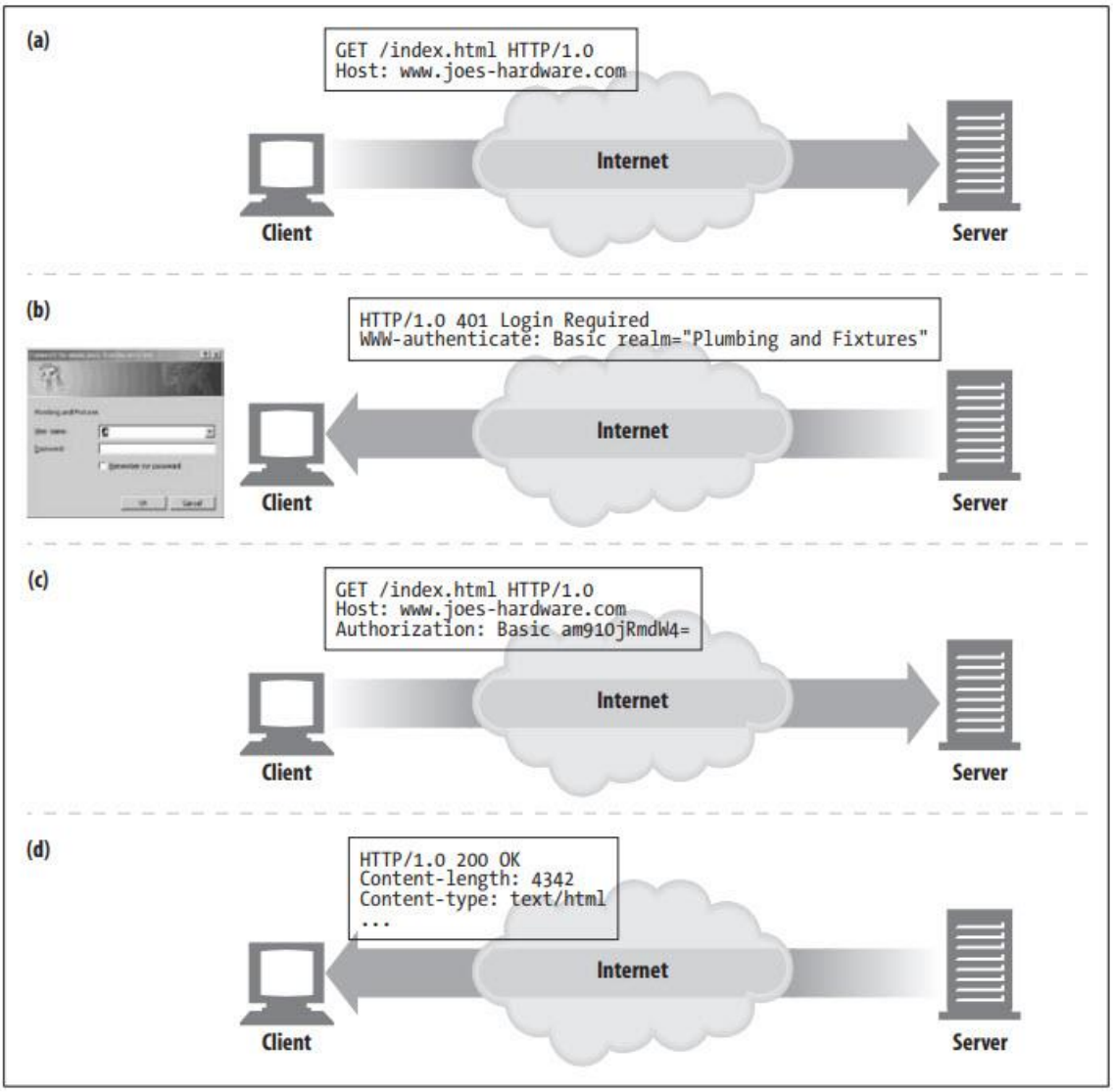
برخی از سایت‌ها حتی از آدرس‌های IP کلاینت به عنوان یک ویژگی امنیتی استفاده می‌کنند و اسناد را فقط به کاربران از یک آدرس IP خاص ارائه می‌کنند. در حالی که این ممکن است در محدوده یک اینترانت کافی باشد، اما در اینترنت خراب می‌شود (به دلیل سهولت جعل آدرس‌های IP). وجود پروکسی‌های رهگیری در مسیر نیز این طرح را شکسته است. فصل ۱۴ طرح‌های بسیار قوی‌تری را برای کنترل دسترسی به اسناد ممتاز مورد بحث قرار می‌دهد.

User Login

به جای تلاش منفعلانه برای حدس زدن هویت یک کاربر از آدرس IP او، یک وب سرور می‌تواند به طور صریح از کاربر بپرسد که او کیست و از او بخواهد با نام کاربری و رمز عبور، احراز هویت (ورود به سیستم) شود.

برای کمک به آسان‌تر کردن ورود به وب‌سایت، HTTP یک مکانیسم داخلی برای ارسال اطلاعات نام کاربری به وب‌سایت‌ها، با استفاده از هدرهای WWW-Authenticate و Authorization دارد. پس از ورود، مرورگرها به طور مداوم این اطلاعات ورود را با هر درخواست به سایت ارسال می‌کنند، بنابراین اطلاعات همیشه در دسترس هستند. ما این احراز هویت HTTP را با جزئیات بیشتری در فصل ۱۲ مورد بحث قرار خواهیم داد، اما اجازه دهید اکنون نگاهی گذرا به آن بیندازیم.

اگر سروری بخواهد کاربر قبل از دسترسی به سایت ثبت نام کند، می‌تواند کد پاسخ مورد نیاز ورود به سیستم HTTP 401 را به مرورگر ارسال کند. سپس مرورگر یک کادر محاوره‌ای ورود به سیستم را نمایش می‌دهد و اطلاعات را در درخواست بعدی با استفاده از هدر Authorization به مرورگر ارائه می‌کند. این موضوع در شکل زیر نشان داده شده است.



فرآیندی که در این شکل اتفاق می‌افتد عبارتست از:

- در بخش a شکل بالا، یک مرورگر از سایت www.joes-hardware.com درخواست می‌کند.
- سایت هویت کاربر را نمی‌داند، بنابراین در بخش b شکل بالا، سرور با بازگرداندن کد پاسخ 401 Login Required HTTP درخواست ورود می‌کند و هدر WWW-Authenticate را اضافه می‌کند. این باعث می‌شود که در مرورگر یک کادر محاوره‌ای برای ورود به سیستم ظاهر شود.
- هنگامی که کاربر یک نام کاربری و یک رمز عبور را وارد می‌کند (برای بررسی هویت خود)، مرورگر درخواست اصلی را تکرار می‌کند. این بار یک هدر Authorization اضافه می‌کند که نام کاربری و رمز عبور را مشخص می‌کند. نام کاربری و رمز عبور درهم می‌شوند تا از دید ناظران تصادفی یا تصادفی شبکه پنهان شوند.

- اکنون سرور از هویت کاربر آگاه است.
- برای درخواست‌های بعدی، مرورگر به‌طور خودکار نام کاربری و رمز عبور ذخیره‌شده را در صورت درخواست صادر می‌کند و اغلب حتی در صورت عدم درخواست، آن را به سایت ارسال می‌کند. این موضوع سبب می‌شود که با ارسال هدر مجوز (به عنوان نشانه هویت شما) در هر درخواست به سرور، یک بار به یک سایت وارد شده و هویت شما در طول جلسه حفظ شود.

با این حال، ورود به وب سایت‌ها خسته کننده است. همانطور که فرد از سایتی به سایت دیگر مرور می‌کند، باید برای هر سایتی اقدام به لاگین نماید. بدتر از همه، این احتمال وجود دارد که فرد باید نام‌های کاربری و رمزهای عبور مختلف را برای سایت‌های مختلف به خاطر بسپارد. نام کاربری مورد علاقه او، "fred" از قبل توسط شخص دیگری در زمان بازدید از بسیاری از سایت‌ها انتخاب شده است و برخی از سایت‌ها قوانین متفاوتی در مورد طول و ترکیب نام‌های کاربری و رمز عبور خواهند داشت. خیلی زود، fred اینترنت را رها می‌کند و به تماشای اپرا باز می‌گردد. بخش بعدی راه حل این مشکل را مورد بحث قرار می‌دهد.

Fat URLs

برخی از وب‌سایت‌ها با ایجاد نسخه‌های ویژه از هر URL برای هر کاربر، هویت کاربر را ردیابی می‌کنند. به طور معمول، یک URL واقعی با افزودن برخی از اطلاعات وضعیت به ابتدا یا انتهای مسیر URL گسترش می‌یابد. همانطور که کاربر سایت را مرور می‌کند، وب سرور به صورت پویا لینک‌هایی ایجاد می‌کند که همچنان اطلاعات وضعیت موجود در URL ها را حفظ می‌کند.

نشانی‌های اینترنتی که برای گنجاندن اطلاعات وضعیت کاربر اصلاح می‌شوند، Fat URLs نامیده می‌شوند. در زیر چند نمونه از این نوع URL های استفاده شده در وب سایت تجارت الکترونیک Amazon.com آورده شده است. هر URL با یک شماره شناسایی منحصر به فرد کاربر (002-1145265-8016838 ، در این مورد) مشخص می‌شود که به ردیابی کاربر در هنگام مرور فروشگاه کمک می‌کند.

```
...
<a href="/exec/obidos/tg/browse/-/229220/ref=gr_gifts/002-1145265-8016838">All
  Gifts</a><br>
<a href="/exec/obidos/wishlist/ref=gr_pl1_/002-1145265-8016838">Wish List</a><br>
...
<a href="http://s1.amazon.com/exec/varzea/tg/armed-forces/-//ref=gr_af_/002-1145265-
  8016838">Salute Our Troops</a><br>
<a href="/exec/obidos/tg/browse/-/749188/ref=gr_p4_/002-1145265-8016838">Free
  Shipping</a><br>


 <a href="/exec/obidos/tg/browse/-/468532/ref=gr_returns/002-1145265-8016838">Easy
   Returns</a>
 ...
```

شما می‌توانید از Fat URL ها استفاده کنید تا تراکنش‌های HTTP مستقل با یک وب سرور را به یک "Session" یا "Visit" متصل کنید. اولین باری که کاربر از وب سایت بازدید می‌کند، یک شناسه منحصر به فرد به روشی که برای سرور قابل تشخیص باشد ایجاد می‌گردد. این مقدار به URL اضافه می‌شود و سرور، کلاینت را به این Fat URL هدایت می‌کند. هر زمان که سرور درخواستی برای Fat URL دریافت می‌کند، می‌تواند هر حالت افزایشی مرتبط با آن شناسه کاربری (سبدهای خرید، پروفایل‌ها و غیره) را جستجو کند و تمام لینک‌های خروجی را بازنویسی می‌کند تا آن‌ها را Fat نموده تا شناسه کاربری را حفظ کند.

از Fat URL ها می‌توان به منظور شناسایی کاربران هنگام مرور یک سایت استفاده کرد. اما این فناوری چندین مشکل جدی دارد. برخی از این مشکلات عبارتند از:

Ugly URLs

Fat URL های نمایش داده شده در مرورگر برای کاربران جدید گیج کننده است.

Can’t share URLs

Fat URL ها حاوی اطلاعات وضعیتی در مورد یک کاربر و جلسه خاص هستند. اگر آن URL را برای شخص دیگری پست کنید، ممکن است ناخواسته اطلاعات شخصی انباشته شده خود را به اشتراک بگذارید.

Breaks caching

تولید نسخه‌های خاص کاربر از هر URL به این معنی است که دیگر URLهایی که معمولاً به حافظه کش دسترسی دارند وجود ندارد.

Extra server load

سرور باید صفحات HTML را بازنویسی کند تا URL ها را Fat کند.

Escape hatches

برای کاربر بسیار آسان است که به طور تصادفی از جلسه Fat URL با پرش به سایت دیگری یا با درخواست یک URL خاص "escape" کند. Fat URL ها تنها در صورتی کار می‌کنند که کاربر به شدت لینک‌های از پیش اصلاح شده را دنبال کند. اگر کاربر escape کند، ممکن است پیشرفت خود را از دست بدهد (شاید یک سبد خرید پر شده) و باید دوباره شروع کند.

Not persistent across sessions

وقتی کاربر از سیستم خارج می‌شود، همه اطلاعات از بین می‌رود، مگر اینکه Fat URL خاص را نشانه‌گذاری کند.

Cookies

کوکی‌ها بهترین راه فعلی برای شناسایی کاربران و اجازه دادن به جلسات دائمی هستند. آن‌ها از بسیاری از مشکلات تکنیک‌های قبلی رنج نمی‌برند، اما اغلب در ارتباط با آن تکنیک‌ها برای ارزش بیشتر استفاده می‌شوند. کوکی‌ها ابتدا توسط نت اسکیپ توسعه داده شدند اما اکنون توسط همه مرورگرهای اصلی پشتیبانی می‌شوند.

از آنجایی که کوکی‌ها مهم هستند و هدرهای HTTP جدیدی را تعریف می‌کنند، ما آن‌ها را با جزئیات بیشتری نسبت به تکنیک‌های قبلی بررسی می‌کنیم. وجود کوکی‌ها بر روی Cache نیز تأثیر می‌گذارد و اکثر Cache ها و مرورگرها ذخیره هر محتوای کوکی شده را ممنوع می‌کنند. بخش‌های بعدی جزئیات بیشتری را ارائه می‌دهند.

Types of Cookies

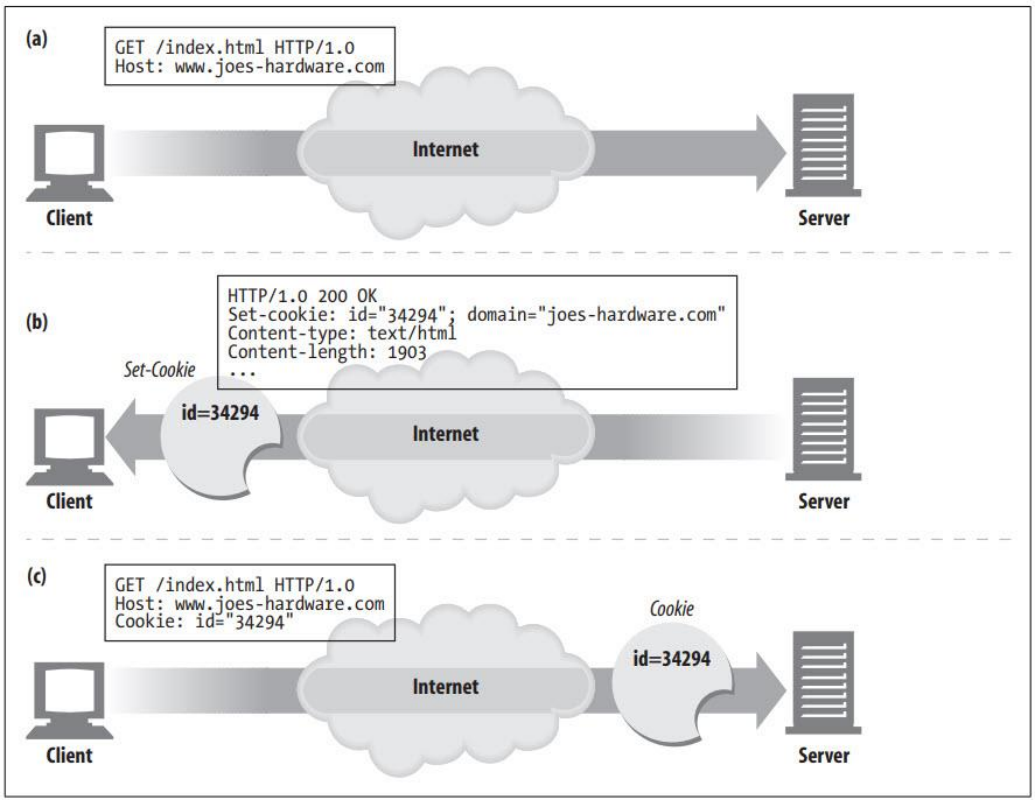
می‌توانید کوکی‌ها را به طور کلی به دو نوع دسته‌بندی کنید: کوکی‌های Session و کوکی‌های Persistent. کوکی Session یک کوکی موقت است که تنظیمات و Preference ها را هنگام حرکت کاربر در یک سایت پیگیری می‌کند. هنگامی که کاربر از مرورگر خارج می‌شود، یک کوکی Session حذف می‌شود. کوکی‌های Persistent می‌توانند بیشتر عمر کنند. آن‌ها بر روی دیسک ذخیره می‌شوند و پس از خروج مرورگر و راه اندازی مجدد کامپیوتر نیز جان سالم به در می‌برند. کوکی‌های Persistent اغلب برای حفظ پیکربندی مربوط به پروفایل یا نام ورود به سایتی که کاربر به طور دوره‌ای از آن بازدید می‌کند استفاده می‌شود.

تنها تفاوت بین کوکی‌های Session و کوکی‌های Persistent، زمانی است که منقضی می‌شوند. همانطور که بعدا خواهیم دید، اگر در یک کوکی پارامتر Discard آن تنظیم شده باشد یا اگر پارامتر Expires یا Max-Age وجود نداشته باشد که زمان انقضای طولانی‌تری را نشان دهد، ما یک کوکی Session داریم.

How Cookies Work

کوکی‌ها مانند برچسب‌های "Hello, My Name Is" هستند که توسط سرورها بر روی کاربران چسبانده شده‌اند. هنگامی که یک کاربر از یک وب سایت بازدید می‌کند، وب سایت می‌تواند تمام برچسب‌های متصل شده توسط آن سرور به کاربر را بخواند.

اولین باری که کاربر از یک وب سایت بازدید می‌کند، وب سرور چیزی در مورد کاربر نمی‌داند (بخش a شکل زیر). سرور وب انتظار دارد که همین کاربر دوباره بازگردد، بنابراین می‌خواهد یک کوکی منحصربه‌فرد را بر روی کاربر بگذارد تا بتواند این کاربر را در آینده شناسایی کند. کوکی حاوی یک لیست دلخواه از اطلاعات name=value است و با استفاده از هدرهای پاسخ HTTP مانند Set-Cookie یا Set-Cookie2 به کاربر پیوست می‌شود.



کوکی‌ها می‌توانند حاوی هر اطلاعاتی باشند، اما اغلب حاوی یک شماره شناسایی منحصر به فرد هستند که توسط سرور برای اهداف ردیابی ایجاد می‌شود. به عنوان مثال، در بخش b شکل بالا، سرور یک کوکی بر روی کاربر قرار می‌دهد که به مقدار id=34294 اشاره می‌کند. سرور می‌تواند از این شماره برای جستجوی اطلاعات پایگاه داده‌ای که سرور برای بازدیدکنندگان خود جمع آوری می‌کند (سابقه خرید، اطلاعات آدرس و غیره) استفاده کند.

با این حال، کوکی‌ها فقط به شماره شناسه محدود نمی‌شوند. بسیاری از سرورهای وب انتخاب می‌کنند که اطلاعات را مستقیماً در کوکی‌ها نگه دارند. مثلا:

Cookie: name="Brian Totty"; phone="555-1212"

مرورگر محتویات کوکی ارسال شده از سرور را در هدرهای Set-Cookie یا Set-Cookie2 به خاطر می‌آورد و مجموعه کوکی‌ها را در پایگاه داده کوکی مرورگر ذخیره می‌کند (آن را مانند یک چمدان با برچسب‌هایی از کشورهای مختلف در نظر بگیرید). هنگامی که کاربر در آینده به همان سایت باز می‌گردد (بخش c شکل بالا)، مرورگر کوکی‌هایی را که توسط آن سرور بر روی کاربر قرار داده است انتخاب می‌کند و آن‌ها را در هدر درخواست کوکی ارسال می‌کند.

Cookie Jar: Client-Side State

ایده اصلی کوکی‌ها این است که به مرورگر اجازه دهند مجموعه‌ای از اطلاعات خاص سرور را جمع آوری کند و هر بار که کاربر آن را بازدید می‌کند این اطلاعات را به سرور ارائه دهد.

از آنجایی که مرورگر وظیفه ذخیره اطلاعات کوکی را بر عهده دارد، به این سیستم Client-Side State می‌گویند. نام رسمی مشخصات کوکی HTTP State Management Mechanism است.

Netscape Navigator cookies

مرورگرهای مختلف، کوکی‌ها را به روش‌های مختلف ذخیره می‌کنند. Netscape Navigator کوکی‌ها را در یک فایل متنی به نام cookies.txt ذخیره می‌کند. مثلا:

```
# Netscape HTTP Cookie File
# http://www.netscape.com/newsref/std/cookie_spec.html
# This is a generated file!  Do not edit.
#
# domain                 allh   path     secure expires     name       value

www.fedex.com            FALSE  /        FALSE  1136109676  cc         /us/
.bankofamericaonline.com TRUE   /        FALSE  1009789256  state      CA
.cnn.com                 TRUE   /        FALSE  1035069235  SelEdition www
secure.eepulse.net       FALSE  /eePulse FALSE  1007162968  cid        %FE%FF%002
www.reformamt.org        TRUE   /forum   FALSE  1033761379  LastVisit  1003520952
www.reformamt.org        TRUE   /forum   FALSE  1033761379  UserName   Guest
  ...
```

هر خط از فایل متنی نشان دهنده یک کوکی است. هفت فیلد جدا شده با تب وجود دارد:

domain

نشان دهنده Domain مربوط به کوکی است.

allh

نشان دهنده این است که همه میزبان‌های یک دامنه، کوکی را دریافت نموده، یا فقط میزبان خاص نام‌گذاری شده آن‌ها را دریافت نمایند.

path

نشان دهنده پیشوند مسیر در دامنه مرتبط با کوکی است.

secure

نشان دهنده این است که آیا ما باید این کوکی را فقط در صورت داشتن اتصال SSL ارسال کنیم یا خیر.

expiration

نشان دهنده تاریخ انقضای کوکی به شکل ثانیه و از ۱ ژانویه 1970 00:00:00 GMT است.

name

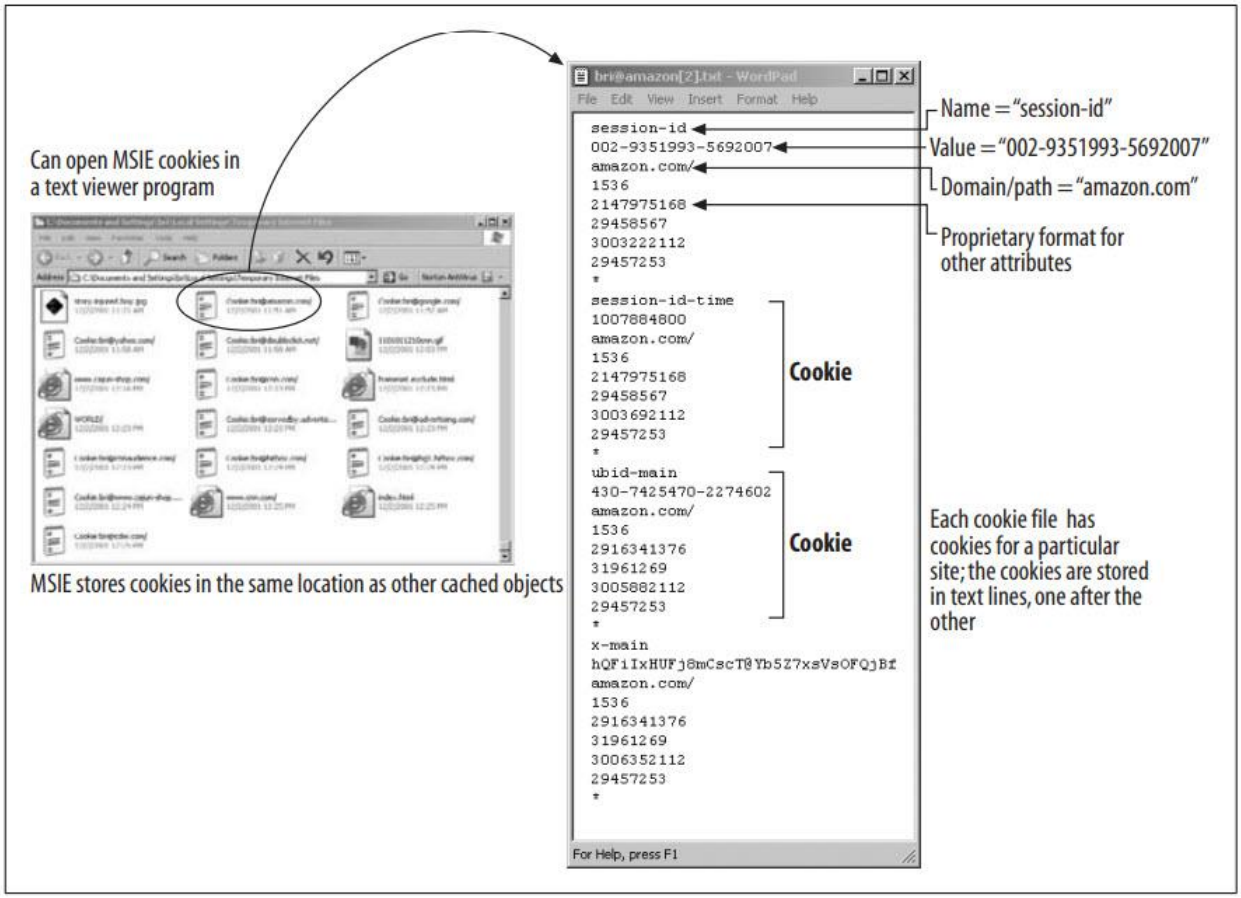
نشان دهند نام متغیر کوکی است.

value

نشان دهنده مقدار متغیر کوکی است.

Microsoft Internet Explorer cookies

مایکروسافت اینترنت اکسپلورر، کوکی‌ها را در فایل‌های متنی جداگانه در دایرکتوری Cache ذخیره می‌کند. همانطور که در شکل زیر نشان داده شده است، می‌توانید این دایرکتوری را برای مشاهده کوکی‌ها مرور کنید. فرمت فایل‌های کوکی اینترنت اکسپلورر اختصاصی است، اما بسیاری از زمینه‌ها به راحتی قابل درک هستند. هر کوکی یکی پس از دیگری در فایل ذخیره می‌شود و هر کوکی از چندین خط تشکیل شده است.



خط اول هر کوکی در فایل حاوی نام متغیر کوکی است. خط بعدی مقدار متغیر است. خط سوم شامل دامنه و مسیر است. خطوط باقی‌مانده داده‌های اختصاصی (احتمالاً شامل تاریخ و سایر فلگ‌ها) هستند.

Different Cookies for Different Sites

یک مرورگر می‌تواند صدها یا هزاران کوکی در کوکی داخلی خود داشته باشد، اما مرورگرها همه کوکی‌ها را به هر سایتی ارسال نمی‌کنند. در واقع، آن‌ها معمولاً فقط دو یا سه کوکی به هر سایت ارسال می‌کنند. در اینجا دلیل آن است:

- جابجایی تمام آن بایت‌های کوکی عملکرد را به طور چشمگیری کاهش می‌دهد. مرورگرها در واقع بایت‌های کوکی بیشتری نسبت به بایت‌های محتوای واقعی جابجا می‌کنند!
- بیشتر این کوکی‌ها برای اکثر سایت‌ها به دلیل داشتن جفت‌های name/value خاص سرور، بیهودگی غیرقابل تشخیص هستند.
- ارسال همه کوکی‌ها به همه سایت‌ها باعث نگرانی بالقوه حفظ حریم خصوصی می‌شود، زیرا سایت‌هایی که به آن‌ها اعتماد ندارید اطلاعاتی را که فقط برای سایت دیگری در نظر گرفته‌اید دریافت می‌کنند.

به طور کلی، یک مرورگر تنها کوکی‌هایی را که سرور ایجاد کرده است به سرور ارسال می‌کند. کوکی‌های تولید شده توسط joes-hardware.com به joes-hardware.com ارسال می‌شوند و نه به bobs-books.com یا marys-movies.com.

بسیاری از وب‌سایت‌ها برای مدیریت تبلیغات با فروشندگان شخص ثالث قرارداد می‌بندند. این تبلیغات به گونه‌ای ساخته می‌شوند که به نظر می‌رسند بخش جدایی‌ناپذیر وب‌سایت هستند و کوکی‌های Persistent را Push می‌دهند. هنگامی که کاربر به وب‌سایت دیگری که توسط همان شرکت تبلیغاتی سرویس دهی می‌شود، می‌رود، کوکی‌های Persistent که قبلاً تنظیم شده‌اند دوباره توسط مرورگر ارسال می‌شود (زیرا دامنه‌ها مطابقت دارند). یک شرکت بازاریابی می‌تواند از این تکنیک، همراه با هدر Referer، برای ایجاد یک مجموعه داده جامع از پروفایل‌های کاربر و عادت‌های مرور وی استفاده کند. مرورگرهای مدرن به شما امکان می‌دهند تنظیمات حریم خصوصی را برای محدود کردن کوکی‌های شخص ثالث پیکربندی کنید.

Cookie Domain attribute

سروری که یک کوکی ایجاد می‌کند، می‌تواند با افزودن یک ویژگی Domain به هدر پاسخ Set-Cookie، کنترل کند که کدام سایت‌ها آن کوکی را ببینند. به عنوان مثال، هدر پاسخ HTTP زیر به مرورگر می‌گوید که کوکی user=mary17 را به هر سایتی در دامنه airtravelbargains.com. ارسال کند:

Set-cookie: user="mary17"; domain="airtravelbargains.com"

اگر کاربر از www.airtravelbargains.com، specials.airtravelbargains.com یا هر سایتی که به airtravelbargains.com. ختم می‌شود بازدید کند، هدر کوکی زیر صادر می‌شود:

Cookie: user="mary17"

Cookie Path attribute

مشخصات کوکی حتی به شما امکان می‌دهد کوکی‌ها را با بخش‌هایی از وب‌سایت‌ها مرتبط کنید. این کار با استفاده از ویژگی Path انجام می شود که نشان دهنده پیشوند مسیر URL است که در آن هر کوکی معتبر است.

برای مثال، یک وب سرور ممکن است بین دو سازمان به اشتراک گذاشته شود که هر کدام کوکی‌های جداگانه دارند. سایت www.airtravelbargains.com ممکن است بخشی از وب سایت خود را با استفاده از یک کوکی جداگانه برای پیگیری اندازه ماشین ترجیحی کاربر به اجاره خودرو اختصاص دهد - مثلاً http://www.airtravelbargains.com/autos/ -. یک کوکی مخصوص اجاره خودکار ممکن است به این صورت ایجاد شود:

Set-cookie: pref=compact; domain="airtravelbargains.com"; path=/autos/

اگر کاربر به http://www.airtravelbargains.com/specials.html مراجعه کند، فقط این کوکی را دریافت خواهد کرد:

Cookie: user="mary17"

اما اگر او به http://www.airtravelbargains.com/autos/cheapo/index.html برود، هر دوی این کوکی‌ها را دریافت می‌کند:

Cookie: user="mary17"

Cookie: pref=compact

بنابراین، کوکی‌ها تکه‌هایی از حالت هستند که توسط سرورها بر روی کلاینت قرار می‌گیرند، توسط کلاینت‌ها نگهداری می‌شوند و تنها به آن سایت‌هایی که مناسب هستند بازگردانده می‌شوند. بیایید با جزئیات بیشتری به فناوری و استانداردهای کوکی نگاه کنیم.

Cookie Ingredients

دو نسخه مختلف از مشخصات کوکی در حال استفاده وجود دارد: کوکی‌های نسخه 0 (که گاهی اوقات «کوکی‌های Netscape» نامیده می‌شوند)، و کوکی‌های نسخه 1 («RFC 2965»). کوکی‌های نسخه 1 یک برنامه افزودنی کم استفاده از کوکی‌های نسخه 0 هستند.

مشخصات کوکی نسخه 0 یا نسخه 1 به عنوان بخشی از مشخصات HTTP/1.1 ثبت نشده است. دو سند کمکی اولیه وجود دارد که به بهترین شکل استفاده از کوکی‌ها را توصیف می‌کند که در جدول زیر خلاصه شده است.

Title	Description	Location
Persistent Client State: HTTP Cookies	Original Netscape cookie standard	*http://home.netscape.com/newsref/std/cookie_spec.html*
RFC 2965: HTTP State Management Mechanism	October 2000 cookie standard, obsoletes RFC 2109	*http://www.ietf.org/rfc/rfc2965.txt*

Version 0 (Netscape) Cookies

مشخصات اولیه کوکی توسط Netscape تعریف شده است. این کوکی‌های «نسخه 0» هدر پاسخ Set-Cookie، هدر درخواست کوکی و فیلدهای موجود برای کنترل کوکی‌ها را تعریف می‌کنند. کوکی‌های نسخه 0 به شکل زیر هستند:

```
Set-Cookie: name=value [; expires=date] [; path=path] [; domain=domain] [; secure]

Cookie: name1=value1 [; name2=value2] ...
```

Version 0 Set-Cookie header

هدر Set-Cookie یک نام کوکی و مقدار کوکی اجباری دارد. می‌توان آن را با ویژگی‌های کوکی اختیاری، که با نقطه ویرگول از هم جدا می‌شوند، دنبال کرد. فیلدهای Set-Cookie در ادامه توضیح داده شده است.

NAME=VALUE

اجباری. هر دو NAME و VALUE دنباله ای از کاراکترها هستند، به استثنای نقطه ویرگول، کاما، علامت مساوی و فضای خالی، مگر اینکه در گیومه‌های دوگانه نقل قول شوند. وب سرور می‌تواند هر ارتباط NAME=VALUE را ایجاد کند که در بازدیدهای بعدی از سایت به وب سرور بازگردانده می‌شود.

Set-Cookie: customer=Mary

Expires

اختیاری. این ویژگی یک رشته تاریخ را مشخص نموده که طول عمر معتبر آن کوکی می‌باشد. پس از رسیدن به تاریخ انقضا، کوکی دیگر ذخیره یا ارائه نخواهد شد. تاریخ به صورت زیر تنظیم شده است:

Weekday, DD-Mon-YY HH:MM:SS GMT

تنها منطقه زمانی قانونی GMT است و جداکننده‌های بین عناصر تاریخ باید خط تیره باشند. اگر Expires مشخص نشده باشد، کوکی پس از پایان جلسه کاربر منقضی می‌شود.

Set-Cookie: foo=bar; expires=Wednesday, 09-Nov-99 23:12:40 GMT

Domain

اختیاری. یک مرورگر کوکی را فقط به نام میزبان سرور در دامنه مشخص شده ارسال می‌کند. این به سرورها اجازه می‌دهد کوکی‌ها را فقط به دامنه‌های خاصی محدود کنند. دامنه "acme.com" با نام میزبان anvil.acme.com و shipping.crate.acme.com مطابقت داشته اما با www.cnn.com مطابقت ندارد.

فقط میزبان‌های درون دامنه مشخص‌شده می‌توانند یک کوکی برای یک دامنه تنظیم کنند و دامنه‌ها باید حداقل دو یا سه نقطه در خود داشته باشند تا از دامنه‌هایی به شکل .com،.edu و «va.us» جلوگیری شود. هر دامنه‌ای که در مجموعه ثابت Top-Level Domain ویژه فهرست شده در اینجا قرار می‌گیرد، تنها به دو دوره نیاز دارد. هر دامنه دیگری به حداقل سه دامنه نیاز دارد. Top-Level Domain های ویژه عبارتند از: .com، .edu، .net، .org، .gov، .mil، .int، .biz، .info، .name، .museum، .coop، .aero، و .pro .

اگر دامنه مشخص نشده باشد، نام میزبان سروری که پاسخ Set-Cookie را ایجاد کرده است، پیش‌فرض است.

Set-Cookie: SHIPPING=FEDEX; domain="joes-hardware.com"

Path

اختیاری. این ویژگی به شما امکان می‌دهد کوکی‌ها را به اسناد خاصی در یک سرور اختصاص دهید. اگر ویژگی Path پیشوند یک مسیر URL باشد، یک کوکی می‌تواند پیوست شود. مسیر "/foo" با "/foobar" و "/foo/bar.html" مطابقت دارد. مسیر "/" با همه چیز در دامنه مطابقت دارد.

اگر مسیر مشخص نشده باشد، روی مسیر URL که پاسخ Set-Cookie را ایجاد کرده تنظیم می‌شود.

Set-Cookie: lastorder=00183; path=/orders

Secure

اختیاری. اگر این ویژگی گنجانده شود، کوکی تنها در صورتی ارسال می‌شود که HTTP از اتصال امن SSL استفاده کند.

Set-Cookie: private_id=519; secure

Version 0 Cookie header

هنگامی که یک کلاینت درخواستی را ارسال می‌کند، این درخواست شامل تمام کوکی‌های منقضی نشده است که با Domain، Path و Secure Filters مطابقت دارند. همه کوکی‌ها در یک هدر کوکی ترکیب می‌شوند:

Cookie: session-id=002-1145265-8016838; session-id-time=1007884800

Version 1 (RFC 2965) Cookies

یک نسخه توسعه یافته از کوکی‌ها در RFC 2965 (قبلاً RFC 2109) تعریف شده است. این استاندارد نسخه 1 هدرهای Set-Cookie2 و Cookie2 را معرفی می‌کند، اما با سیستم نسخه 0 نیز تعامل دارد.

استاندارد کوکی RFC 2965 کمی پیچیده‌تر از استاندارد اصلی Netscape است و هنوز به طور کامل پشتیبانی نمی‌شود. تغییرات عمده کوکی‌های RFC 2965 عبارتند از:

- متن توصیفی را با هر کوکی مرتبط کنید تا هدف آن را توضیح دهید.
- پشتیبانی از تخریب اجباری کوکی‌ها در هنگام خروج از مرورگر، بدون در نظر گرفتن انقضا
- حداکثر سن پیری (Max-Age) کوکی‌ها در ثانیه به جای تاریخ‌های مطلق
- امکان کنترل کوکی‌ها با شماره پورت URL، نه فقط Domain و Path
- هدر کوکی Domain، Port و Path Filters را (در صورت وجود) حمل می‌کند.
- شماره نسخه برای قابلیت همکاری
- perfix $ در هدر کوکی برای تشخیص کلمات کلیدی اضافی از نام‌های کاربری

ساختار کوکی نسخه 1 به شرح زیر است:

```
set-cookie      =       "Set-Cookie2:" cookies
cookies         =       1#cookie
cookie          =       NAME "=" VALUE *(";" set-cookie-av)
NAME            =       attr
VALUE           =       value
set-cookie-av   =       "Comment" "=" value
                |       "CommentURL" "=" <"> http_URL <">
                |       "Discard"
                |       "Domain" "=" value
                |       "Max-Age" "=" value
                |       "Path" "=" value
                |       "Port" [ "=" <"> portlist <"> ]
                |       "Secure"
                |       "Version" "=" 1*DIGIT
portlist        =       1#portnum
portnum         =       1*DIGIT

cookie          =  "Cookie:" cookie-version 1*((";" | ",") cookie-value)
cookie-value    =  NAME "=" VALUE [";" path] [";" domain] [";" port]
cookie-version  =  "$Version" "=" value
NAME            =  attr
VALUE           =  value
path            =  "$Path" "=" value
domain          =  "$Domain" "=" value
port            =  "$Port" [ "=" <"> value <"> ]

cookie2 =       "Cookie2:" cookie-version
```

Version 1 Set-Cookie2 header

ویژگی‌های بیشتری در استاندارد کوکی نسخه 1 نسبت به استاندارد Netscape موجود است. در ادامه خلاصه‌ای سریع از ویژگی‌ها را ارائه داده می‌شود. برای توضیح بیشتر به RFC 2965 مراجعه کنید.

NAME=VALUE

اجباری. وب سرور می‌تواند هر ارتباط NAME=VALUE را ایجاد کند که در بازدیدهای بعدی از سایت به وب سرور بازگردانده می‌شود. نام نباید با "$" شروع شود، زیرا این کاراکتر رزرو شده است.

Version

اجباری. مقدار این ویژگی یک عدد صحیح است که مطابق با نسخه مشخصات کوکی است. RFC 2965 مربوط به نسخه 1 است.

Set-Cookie2: Part="Rocket_Launcher_0001"; Version="1"

Comment

اختیاری. این ویژگی نشان می‌دهد که چگونه یک سرور قصد استفاده از کوکی را دارد. کاربر می‌تواند این خط‌مشی را بررسی کند تا تصمیم بگیرد که آیا یک جلسه با این کوکی مجاز است یا خیر. مقدار باید در ساختار کدگذاری UTF-8 باشد.

CommentURL

اختیاری. این ویژگی یک نشانگر URL به مستندات دقیق در مورد هدف و خط مشی یک کوکی را ارائه می‌دهد. کاربر می‌تواند این خط‌مشی را بررسی کند تا تصمیم بگیرد که آیا یک جلسه با این کوکی مجاز است یا خیر.

Discard

اختیاری. اگر این ویژگی وجود داشته باشد، به کلاینت دستور می‌دهد که کوکی را پس از پایان برنامه کلاینت کنار بگذارد.

Domain

اختیاری. یک مرورگر کوکی را فقط به نام میزبان سرور در دامنه مشخص شده ارسال می‌کند. این به سرورها اجازه می‌دهد کوکی‌ها را فقط به دامنه‌های خاصی محدود کنند. دامنه "acme.com" با نام میزبان "anvil.acme.com" و "shipping.crate.acme.com" مطابقت داشته، اما با "www.cnn.com" مطابقت ندارد. قوانین تطبیق دامنه اساساً مانند کوکی‌های Netscape است، اما چند قانون اضافی وجود دارد. برای جزئیات بیشتر به RFC 2965 مراجعه کنید.

Max-Age

اختیاری. مقدار این ویژگی یک عدد صحیح است که طول عمر کوکی را بر حسب ثانیه تنظیم می‌کند. کلاینت‌ها باید سن کوکی را طبق قوانین محاسبه سن HTTP/1.1 محاسبه کنند. وقتی سن یک کوکی از حداکثر سن بیشتر شود، کلاینت باید کوکی را دور بیندازد. مقدار صفر به این معنی است که کوکی با آن نام باید فوراً کنار گذاشته شود.

Path

اختیاری. این ویژگی به شما امکان می‌دهد کوکی‌ها را به اسناد خاصی در یک سرور اختصاص دهید. اگر ویژگی Path پیشوند یک مسیر URL باشد، یک کوکی می‌تواند پیوست شود. مسیر "/foo" با

"/foobar" و "/foo/bar.html" مطابقت دارد. مسیر "/" با همه چیز در دامنه مطابقت دارد. اگر مسیر مشخص نشده باشد، روی مسیر URL که پاسخ Set-Cookie را ایجاد کرده تنظیم می‌شود.

Port

اختیاری. این ویژگی می‌تواند به‌عنوان یک کلمه کلیدی به تنهایی باشد، یا می‌تواند شامل فهرستی از پورت‌های جدا شده با کاما باشد که ممکن است یک کوکی روی آن اعمال شود. اگر لیست پورت وجود داشته باشد، کوکی را می‌توان فقط برای سرورهایی که پورت‌های آن‌ها با یک پورت در لیست مطابقت دارد ارائه کرد. اگر کلمه کلیدی Port به صورت مجزا ارائه شود، کوکی را می‌توان فقط به شماره پورت سرور پاسخ دهنده فعلی ارائه کرد.

Set-Cookie2: foo="bar"; Version="1"; Port="80,81,8080"

Set-Cookie2: foo="bar"; Version="1"; Port

Secure

اختیاری. اگر این ویژگی گنجانده شود، کوکی تنها در صورتی ارسال می‌شود که HTTP از اتصال امن SSL استفاده کند.

Version 1 Cookie header

کوکی‌های نسخه 1 اطلاعات بیشتری را در مورد هر کوکی تحویل داده شده با خود همراه می‌کنند و فیلترهایی را که هر کوکی ارسال می‌شود، توصیف می‌کند. هر کوکی منطبق باید شامل هر ویژگی Domain، Port یا Path از هدرهای Set-Cookie2 مربوطه باشد.

به عنوان مثال، فرض کنید کلاینت این پنج پاسخ Set-Cookie2 را در گذشته از وب سایت www.joes-hardware.com دریافت کرده است:

Set-Cookie2: ID="29046"; Domain=".joes-hardware.com"

Set-Cookie2: color=blue

Set-Cookie2: support-pref="L2"; Domain="customer-care.joes-hardware.com"

Set-Cookie2: Coupon="hammer027"; Version="1"; Path="/tools"

Set-Cookie2: Coupon="handvac103"; Version="1"; Path="/tools/cordless"

اگر کلاینت درخواست دیگری برای مسیر /tools/cordless/specials.html بدهد، از هدر طولانی Cookie2 مانند زیر عبور می‌کند:

Cookie: $Version="1";

ID="29046"; $Domain=".joes-hardware.com";

color="blue";

Coupon="hammer027"; $Path="/tools";

Coupon="handvac103"; $Path="/tools/cordless"

توجه داشته باشید که تمام کوکی‌های منطبق با فیلترهای Set-Cookie2 تحویل داده می‌شوند و کلمات کلیدی رزرو شده با علامت دلار ($) شروع می‌شوند.

Version 1 Cookie2 header and version negotiation

هدر Cookie2request برای مذاکره در مورد قابلیت همکاری بین کلاینت‌ها و سرورهایی استفاده می‌شود که نسخه‌های مختلف مشخصات کوکی را درک می‌کنند. هدر Cookie2 به سرور توصیه می‌کند که User-Agent کوکی‌های سبک جدید را فهمیده و نسخه استاندارد کوکی پشتیبانی شده را ارائه کند (بهتر است آن را Cookie-Version بنامیم):

Cookie2: $Version="1"

اگر سرور کوکی‌های سبک جدید را بفهمد، Cookie2header را می‌شناسد و باید هدرهای پاسخ Set-Cookie2 (به جای Set-Cookie) را ارسال کند. اگر یک کلاینت هم یک Set-Cookie و یک Set-Cookie2 برای یک کوکی دریافت کند، هدر Set-Cookie قدیمی را نادیده می‌گیرد.

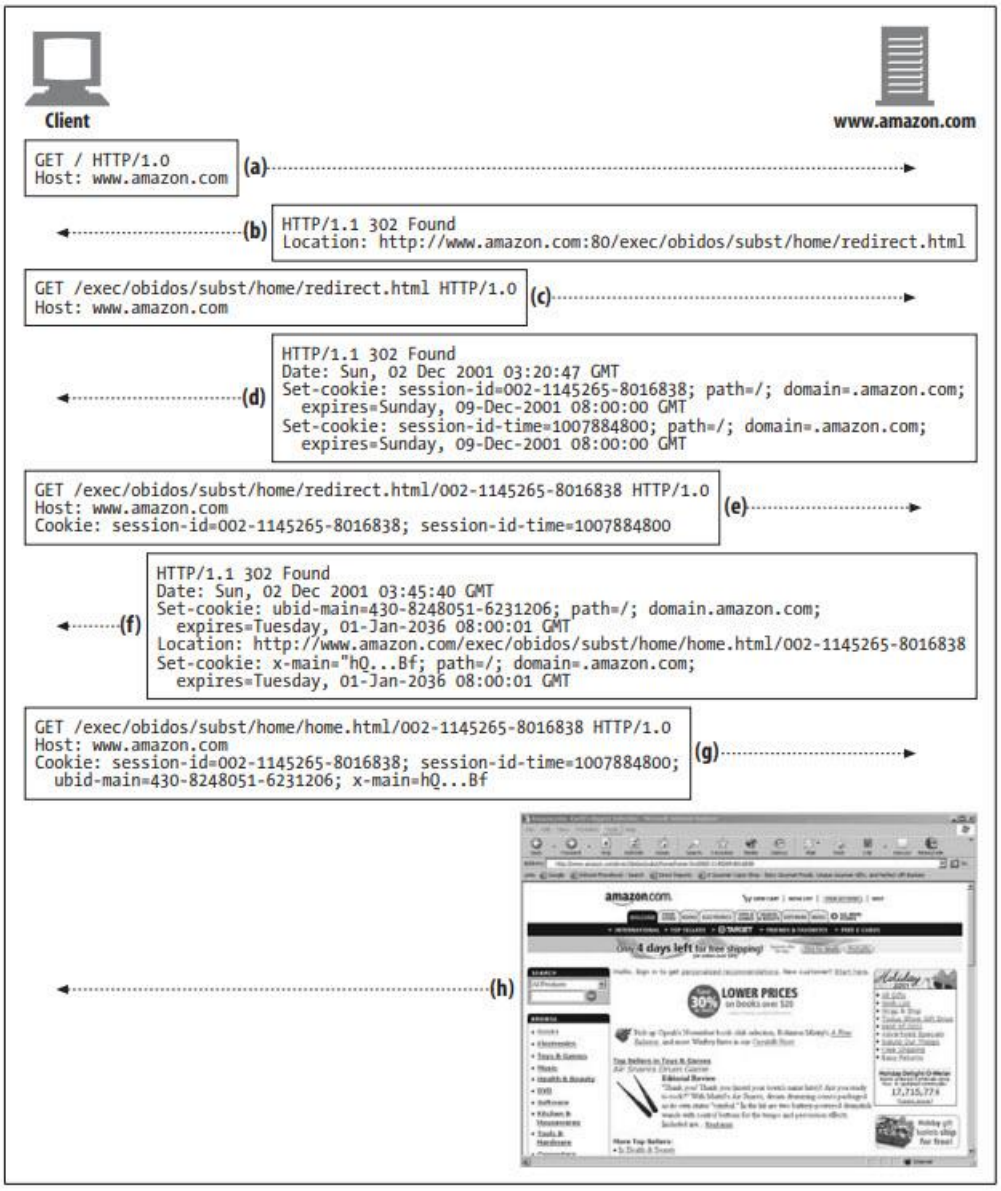
اگر یک کلاینت از کوکی‌های نسخه 0 و نسخه 1 پشتیبانی می‌کند اما هدر SetCookie نسخه 0 را از سرور دریافت می‌کند، باید کوکی‌ها را با هدر کوکی نسخه 0 ارسال کند. با این حال، کلاینت همچنین باید Cookie2: $Version="1" را ارسال کند تا به سرور نشان دهد که می‌تواند ارتقا یابد.

Cookies and Session Tracking

از کوکی‌ها می‌توان برای ردیابی کاربران در حالی که چندین تراکنش با یک وب سایت انجام می‌دهند استفاده کرد. وب‌سایت‌های تجارت الکترونیک از کوکی‌های جلسه برای پیگیری سبد خرید کاربران در هنگام مرور استفاده می‌کنند. بیایید سایت خرید محبوب Amazon.com را مثال بزنیم.

هنگامی که http://www.amazon.com را در مرورگر خود تایپ می‌کنید، زنجیره‌ای از تراکنش‌ها را شروع می‌کنید که در آن سرور وب اطلاعات شناسایی را از طریق یک سری تغییر مسیر، بازنویسی URL و تنظیمات کوکی به آن متصل می‌کند.

شکل زیر یک توالی تراکنش گرفته شده از بازدید Amazon.com را نشان می‌دهد:



بخش a شکل: مرورگر برای اولین بار درخواستی را برای صفحه اصلی Amazon.com ارسال می‌کند.

بخش b شکل: سرور کلاینت را به آدرس اینترنتی نرم افزار تجارت الکترونیک هدایت می‌کند.

بخش c شکل: کلاینت درخواستی را به URL هدایت شده ارسال می‌کند.

بخش d شکل: سرور دو کوکی Session را روی پاسخ قرار می‌دهد و کاربر را به URL دیگری هدایت می‌کند، بنابراین کلاینت با پیوست این کوکی‌ها دوباره درخواست را ارسال می‌کند. این URL جدید یکFat URL است، به این معنی که برخی از حالت‌ها در URL تعبیه شده است. اگر کلاینت کوکی‌ها را غیرفعال کرده باشد، تا زمانی که کاربر لینک‌های Fat URL ایجاد شده توسط Amazon.com را دنبال کند و سایت را ترک نکند، هنوز می‌توان برخی از شناسایی‌های اولیه را انجام داد.

بخش e شکل: کلاینت URL جدید را درخواست می‌کند، اما اکنون دو کوکی پیوست شده را ارسال می‌کند.

بخش f شکل: سرور به صفحه home.html هدایت می‌شود و دو کوکی دیگر را پیوست می‌کند.

بخش g شکل: کلاینت صفحه home.html را واکشی می‌کند و هر چهار کوکی را ارسال می‌کند.

بخش h شکل: سرور محتوا را بازگردانی می‌کند.

Cookies and Caching

هنگام ذخیره اسنادی که با تراکنش‌های کوکی مرتبط هستند، باید مراقب باشید. شما نمی‌خواهید کوکی‌های کاربر قبلی را به یک کاربر اختصاص دهید یا بدتر از آن، محتوای سند شخصی‌شده شخص دیگری را به یک کاربر نشان دهید.

قوانین مربوط به کوکی‌ها و Cache به خوبی ایجاد نشده است. در اینجا چند اصل راهنمایی برای برخورد با Cache آورده شده است:

Mark documents uncacheable if they are

مالک سند بهتر می‌داند که آیا یک سند غیرقابل ذخیره (Uncacheable) است. اگر اسناد غیرقابل ذخیره هستند، صریحاً علامت‌گذاری کنید—به‌ویژه، از Cache-Control: nocache="Set-Cookie" استفاده کنید. روش عمومی‌تر دیگر استفاده از Cache-Control: public برای اسنادی که قابل ذخیره‌سازی هستند که منجر به صرفه‌جویی در پهنای باند وب خواهند شد.

Be cautious about caching Set-Cookie headers

اگر پاسخی دارای هدر Set-Cookie باشد، می‌توانید بدنه را در Cache ذخیره کنید (مگر اینکه خلاف آن گفته شود)، اما باید در مورد ذخیره کردن هدر Set-Cookie بسیار محتاط

باشید. اگر یک هدر Set-Cookie را برای چندین کاربر ارسال کنید، ممکن است هدف گذاری کاربر را شکست دهید.

برخی از Cache ها، هدر Set-Cookie را قبل از ذخیره پاسخ در Cache حذف می‌کنند، اما این نیز می‌تواند مشکلاتی ایجاد کند، زیرا کلاینت‌هایی که از Cache ارائه می‌شوند، دیگر کوکی‌هایی را دریافت نمی‌کنند که معمولاً بدون Cache انجام می‌شوند. این وضعیت را می‌توان با وادار کردن Cache برای تأیید مجدد هر درخواست با سرور مبدا و ادغام هر هدر Set-Cookie بازگشتی با پاسخ کلاینت، بهبود بخشید. سرور مبدا می‌تواند با افزودن این هدر به کپی Cache شده، چنین اعتبار سنجی مجددی را دیکته کند:

Cache-Control: must-revalidate, max-age=0

Cache های محافظه‌کارتر ممکن است از Cache کردن پاسخ‌هایی که دارای هدر SetCookie هستند خودداری کنند، حتی اگر محتوا واقعاً قابل ذخیره‌سازی باشد. برخی از Cache ها زمانی که تصاویر Set-Cookied در Cache ذخیره می‌شوند، حالت‌ها را مجاز می‌کنند، اما متن را نه.

Be cautious about requests with Cookie headers

هنگامی که درخواستی با هدر کوکی وارد می‌شود، اشاره‌ای به این موضوع دارد که محتوای حاصل ممکن است شخصی شود. محتوای شخصی‌شده باید غیرقابل ذخیره‌سازی علامت‌گذاری شود، اما ممکن است برخی از سرورها به اشتباه این محتوا را غیرقابل ذخیره‌سازی علامت‌گذاری نکنند.

Cache های محافظه کار ممکن است تصمیم بگیرند که هیچ سندی را که در پاسخ به درخواستی با عنوان کوکی ارائه می‌شود، ذخیره نکنند و دوباره، برخی از Cache ها حالت‌هایی را در زمانی که تصاویر کوکی‌شده در Cache ذخیره می‌شوند، اجازه می‌دهند، اما متن را نه. سیاست پذیرفته‌شده‌تر این است که تصاویر با هدرهای کوکی را در Cache با زمان انقضای صفر، نگه دارید، بدین ترتیب هر بار مجبور به تأیید مجدد می‌شوید.

Cookies, Security, and Privacy

اعتقاد بر این است که کوکی‌ها به خودی خود یک خطر امنیتی فوق‌العاده نیستند، زیرا می‌توان آن‌ها را غیرفعال کرد و بسیاری از ردیابی‌ها را می‌توان از طریق تجزیه و تحلیل گزارش یا ابزارهای دیگر انجام داد. در واقع، با ارائه یک روش استاندارد و موشکافانه برای نگهداری اطلاعات شخصی در پایگاه‌های داده راه دور

و استفاده از کوکی‌های ناشناس به عنوان کلید، می‌توان فرکانس ارتباط داده‌های حساس از کاربر به سرور را کاهش داد.

با این حال، خوب است در هنگام برخورد با حریم خصوصی و ردیابی کاربران محتاط باشید، زیرا همیشه امکان سوء استفاده وجود دارد. بزرگترین سوء استفاده از وب سایت‌های شخص ثالث است که از کوکی‌های Persistent برای ردیابی کاربران استفاده می‌کنند. این عمل، همراه با آدرس‌های IP و اطلاعات هدر Referer، این شرکت‌های بازاریابی را قادر می‌سازد تا پروفایل‌های کاربر و الگوهای مرور نسبتاً دقیقی بسازند.

با وجود همه تبلیغات منفی، عقل مرسوم این است که رسیدگی به جلسه و راحتی تراکنش‌های کوکی‌ها بر بیشتر خطرات برتری دارد، البته باید در مورد افرادی که اطلاعات شخصی را در اختیار آن‌ها قرار می‌دهید و خط‌مشی‌های رازداری سایت‌ها را بررسی می‌کنید احتیاط کنید.

Computer Incident Advisory Capability (بخشی از وزارت انرژی ایالات متحده) در سال ۱۹۹۸ ارزیابی ای از خطرات بیش از حد ارائه شده کوکی‌ها نوشت. در اینجا گزیده ای از آن گزارش آمده است:

CIAC I-034: Internet Cookies

(http://www.ciac.org/ciac/bulletins/i-034.shtml)

کوکی‌ها قطعات کوتاهی از داده‌ها هستند که توسط سرورهای وب برای کمک به شناسایی کاربران وب مورد استفاده قرار می‌گیرند. مفاهیم و شایعات رایج در مورد کارهایی که یک کوکی می‌تواند انجام دهد تقریباً به ابعاد عرفانی رسیده است و کاربران را ترسانده و مدیران آن‌ها را نگران کرده است.

ارزیابی آسیب‌پذیری:

آسیب‌پذیری سیستم‌ها در برابر آسیب یا جاسوسی با استفاده از کوکی‌های مرورگر وب اساساً وجود ندارد. کوکی‌ها فقط می‌توانند به سرور وب اطلاع دهند که قبلاً در آنجا بوده‌اید یا نه و می‌توانند دفعه بعد که بازدید می‌کنید، اطلاعات کوتاهی (مانند شماره کاربر) را از سرور وب به خود ارسال کنند. بیشتر کوکی‌ها فقط تا زمانی دوام می‌آورند که مرورگر خود را باز نگه داشته اید و در صورت ترک آن از بین می‌روند. نوع دوم کوکی معروف به کوکی Persistent دارای تاریخ انقضا است و تا آن تاریخ روی دیسک شما ذخیره می‌شود. یک کوکی Persistent می‌تواند برای ردیابی عادات مرور کاربر با شناسایی او در زمان بازگشت به سایت مورد استفاده قرار گیرد. اطلاعاتی در مورد اینکه از کجا آمده‌اید و از چه صفحات وب بازدید می‌کنید، قبلاً در فایل‌های گزارش وب سرور وجود دارد و همچنین می‌تواند برای ردیابی عادات مرور کاربران استفاده شود، کوکی‌ها فقط کار را آسان‌تر می‌کنند.

For More Information

Cookies: Simon St.Laurent, McGraw-Hill

http://www.ietf.org/rfc/rfc2965.txt

http://www.ietf.org/rfc/rfc2964.txt

http://home.netscape.com/newsref/std/cookie_spec.html