Basic Authentication

میلیون‌ها نفر از وب برای انجام تراکنش‌های خصوصی و دسترسی به داده‌های خصوصی استفاده می‌کنند. وب دسترسی به این اطلاعات را بسیار آسان می‌کند، اما آسان به اندازه کافی خوب نیست. ما نیاز به اطمینان داریم که چه کسی می‌تواند به داده‌های حساس ما نگاه کند و چه کسی می‌تواند تراکنش‌های ممتاز ما را انجام دهد. همه اطلاعات برای عموم مردم در نظر گرفته نشده است.

ما باید احساس راحتی کنیم که کاربران غیرمجاز نمی‌توانند پروفایل‌های سفر آنلاین ما را مشاهده کنند یا اسناد را بدون رضایت ما در وب سایت‌های ما منتشر کنند. ما باید مطمئن شویم که حساس‌ترین اسناد برنامه‌ریزی شرکتی ما در دسترس اعضای غیرمجاز نیست.

سرورها به راهی نیاز دارند تا هویت کاربر را تصدیق نمایند. هنگامی که یک سرور هویت کاربر را بداند، می‌تواند تصمیم بگیرد که کاربر می‌تواند به کدام تراکنش‌ها و منابع دسترسی داشته باشد. احراز هویت به معنای اثبات این موضوع است که شما چه کسی هستید. معمولاً شما با ارائه یک نام کاربری و یک رمز عبور مخفی احراز هویت می‌کنید. HTTP یک تسهیلات بومی برای احراز هویت HTTP فراهم می‌کند. در حالی که مطمئناً می‌توانید قابلیت‌های احراز هویت خود را در بالای فرم‌ها و کوکی‌های HTTP قرار دهید، برای بسیاری از موقعیت‌ها، احراز هویت بومی HTTP به خوبی مطابقت دارد.

این فصل از کتاب به مبحث احراز هویت HTTP پرداخته و رایج‌ترین شکل احراز هویت HTTP، احراز هویت اولیه یا Basic Authentication را مورد بحث قرار می‌دهد. در فصل بعدی به تکنیک قدرتمندتری به نام احراز هویت Digest خواهیم پرداخت.

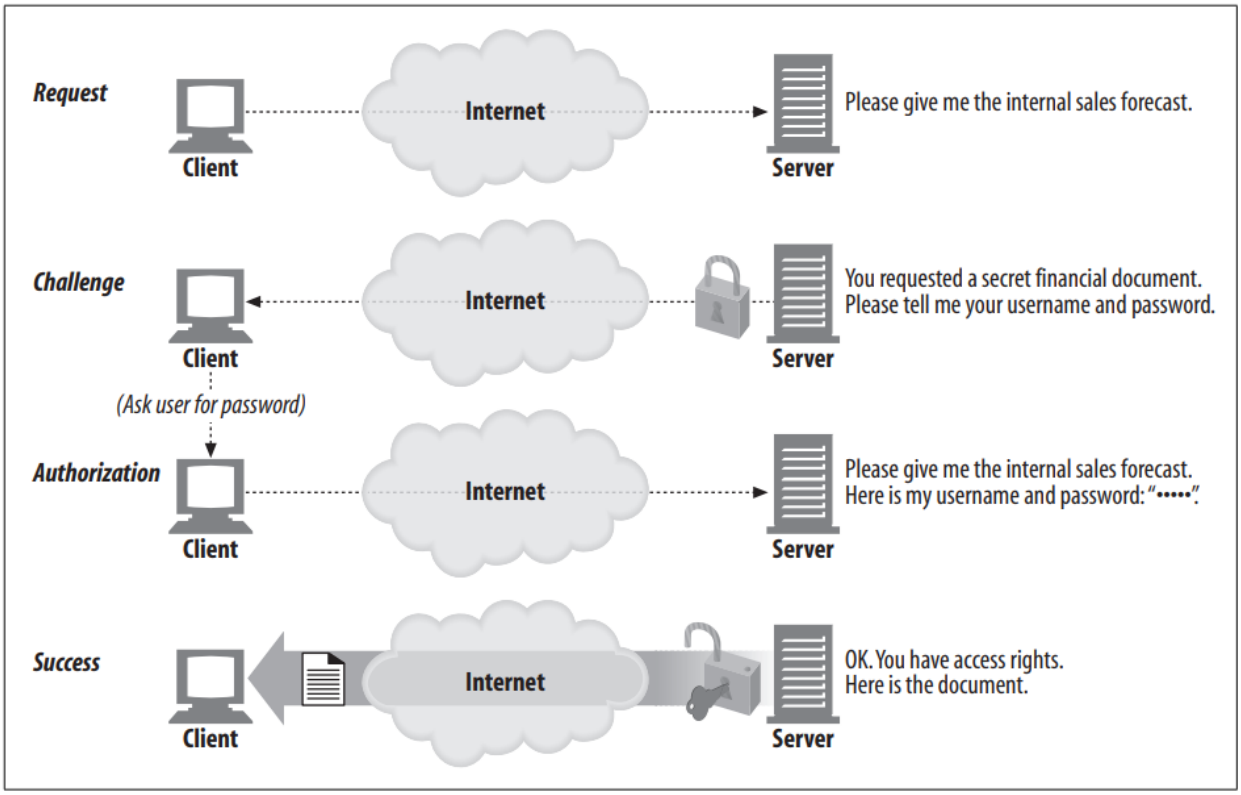
Authentication

احراز هویت به معنای نشان دادن مدارکی از هویت شماست. وقتی یک کارت شناسایی عکس دار، مانند گذرنامه یا گواهینامه رانندگی، نشان می‌دهید، شما در حال ارائه مدرکی هستید که نشان می‌دهد شما همان کسی هستید که ادعا می‌کنید. هنگامی که یک شماره پین را در دستگاه باجه خودکار تایپ می‌کنید، یا یک رمز عبور مخفی را در کادر محاوره ای رایانه تایپ می‌کنید، همچنین ثابت می‌کنید که همان چیزی هستید که ادعا می‌کنید.

در حال حاضر، هیچ یک از این طرح‌ها بدون خطا نیست. رمزهای عبور را می‌توان حدس زد یا شنید و کارت‌های شناسایی را می‌توان دزدید یا جعل کرد. اما هر یک از شواهد پشتیبان به ایجاد یک اعتماد معقول کمک می‌کند که شما همان چیزی هستید که می‌گویید.

HTTP's Challenge/Response Authentication Framework

HTTP یک چارچوب challenge/response بومی را فراهم می‌کند تا احراز هویت کاربران را آسان کند. مدل احراز هویت HTTP در شکل زیر ترسیم شده است.



هر زمان که یک برنامه وب یک پیام درخواست HTTP دریافت می‌کند، سرور به جای اقدام بر روی درخواست، می‌تواند با یک "Authentication Challenge" پاسخ دهد و کاربر را به چالش بکشد تا با ارائه برخی اطلاعات مخفی نشان دهد که او کیست.

زمانی که کاربر درخواست را تکرار می‌کند، باید Credential مخفی (نام کاربری و رمز عبور) را ضمیمه کند. اگر Credential ها مطابقت نداشته باشند، سرور می‌تواند دوباره کاربر را به چالش بکشد یا خطا ایجاد کند. اگر Credential ها مطابقت داشته باشند، درخواست به طور معمول تکمیل می‌شود.

Authentication Protocols and Headers

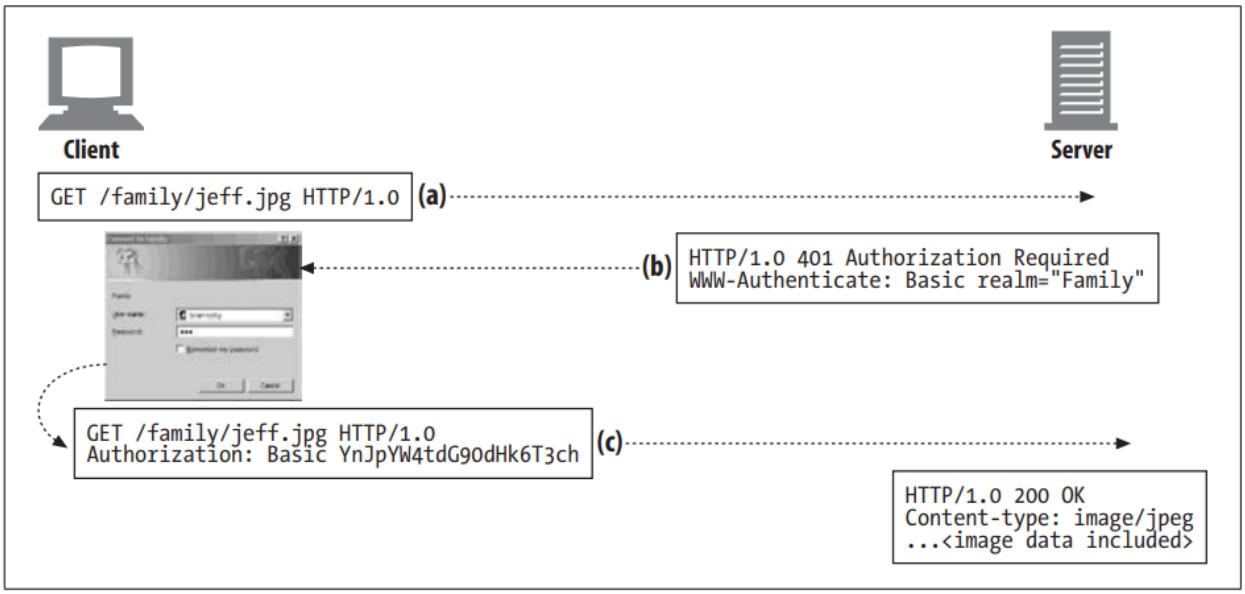
HTTP یک چارچوب قابل توسعه برای پروتکل‌های احراز هویت مختلف، از طریق مجموعه‌ای از هدرهای کنترلی قابل تنظیم، را فراهم می‌کند. قالب و محتوای هدرهای فهرست شده در جدول زیر بسته به

پروتکل احراز هویت متفاوت است. پروتکل احراز هویت نیز در هدرهای احراز هویت HTTP مشخص شده است.

Phase	Headers	Description	Method/Status
Request		The first request has no authentication.	GET
Challenge	WWW-Authenticate	The server rejects the request with a 401 status, indicating that the user needs to provide his username and password. Because the server might have different areas, each with its own password, the server describes the protection area in the WWW-Authenticate header. Also, the authentication algorithm is specified in the WWW-Authenticate header.	401 Unauthorized
Authorization	Authorization	The client retries the request, but this time attaching an Authorization header specifying the authentication algorithm, username, and password.	GET
Success	Authentication-Info	If the authorization credentials are correct, the server returns the document. Some authorization algorithms return some additional information about the authorization session in the optional Authentication-Info header.	200 OK

HTTP دو پروتکل رسمی احراز هویت را تعریف می‌کند: Basic Authentication و Digest Authentication. در آینده، افراد می‌توانند پروتکل‌های جدیدی را طراحی کنند که از چارچوب challenge/response HTTP استفاده می‌کند. بقیه این فصل Basic Authentication را توضیح می‌دهد. برای جزئیات بیشتر در مورد Digest Authentication به فصل ۱۳ مراجعه کنید.

در ادامه اجازه دهید نگاهی به شکل زیر بیندازیم.



هنگامی که یک سرور یک کاربر را به چالش می‌کشد، یک پاسخ 401 غیر مجاز برمی‌گرداند و نحوه و مکان احراز هویت را در هدر WWW-Authenticate توضیح می‌دهد (بخش b شکل).

هنگامی که یک کلاینت به سرور اجازه ادامه کار را می‌دهد، درخواست را مجددا ارسال می‌کند اما یک رمز عبور رمزگذاری شده و سایر پارامترهای احراز هویت را در یک هدر Authorization پیوست می‌کند (بخش c شکل).

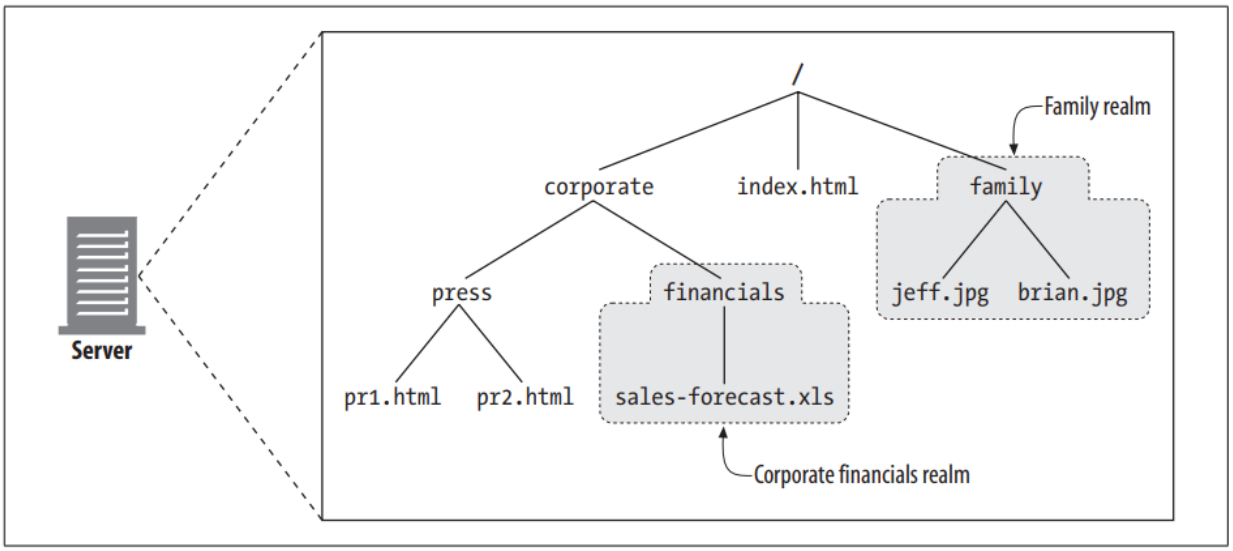
هنگامی که یک درخواست مجاز با موفقیت تکمیل شد، سرور یک کد وضعیت Normal را برمی‌گرداند (به عنوان مثال، 200 OK) و برای الگوریتم‌های احراز هویت پیشرفته، ممکن است اطلاعات اضافی را در یک هدر Authentication-Info ضمیمه کند (بخش d شکل).

Security Realms

قبل از اینکه درباره جزئیات Basic Authentication صحبت کنیم، باید توضیح دهیم که چگونه HTTP به سرورها اجازه می‌دهد تا حقوق دسترسی مختلف را به منابع مختلف مرتبط کنند. شاید متوجه شده باشید که چالش WWW-Authenticate در بخش b شکل بالا شامل یک دستورالعمل realm است. وب سرورها اسناد محافظت شده را در حوزه‌های امنیتی (Security Realms) گروه بندی می‌کنند. هر Security Realm می‌تواند مجموعه‌های مختلفی از کاربران مجاز داشته باشد.

به عنوان مثال، فرض کنید یک وب سرور دارای دو Security Realm است: یکی برای اطلاعات مالی شرکت و دیگری برای اسناد خانوادگی شخصی (شکل زیر را ببینید). کاربران مختلف دسترسی متفاوتی به

Realm ها خواهند داشت. احتمالاً مدیر عامل شرکت شما باید به پیش بینی فروش دسترسی داشته باشد، ولی ممکن است شما نخواهید به او اجازه دسترسی به عکس‌های تعطیلات خانوادگی خود را بدهید!



در اینجا یک چالش Basic Authentication فرضی، با realm مشخص شده است:

```
HTTP/1.0 401 Unauthorized
WWW-Authenticate: Basic realm="Corporate Financials"
```

یک realm باید دارای یک نام رشته توصیفی باشد، مانند "Corporate Financials"، تا به کاربر در درک نام کاربری و رمز عبور کمک کند. همچنین فهرست کردن نام میزبان سرور در نام realm ممکن است مفید باشد - برای مثال، "executive-committee@bigcompany.com".

Basic Authentication

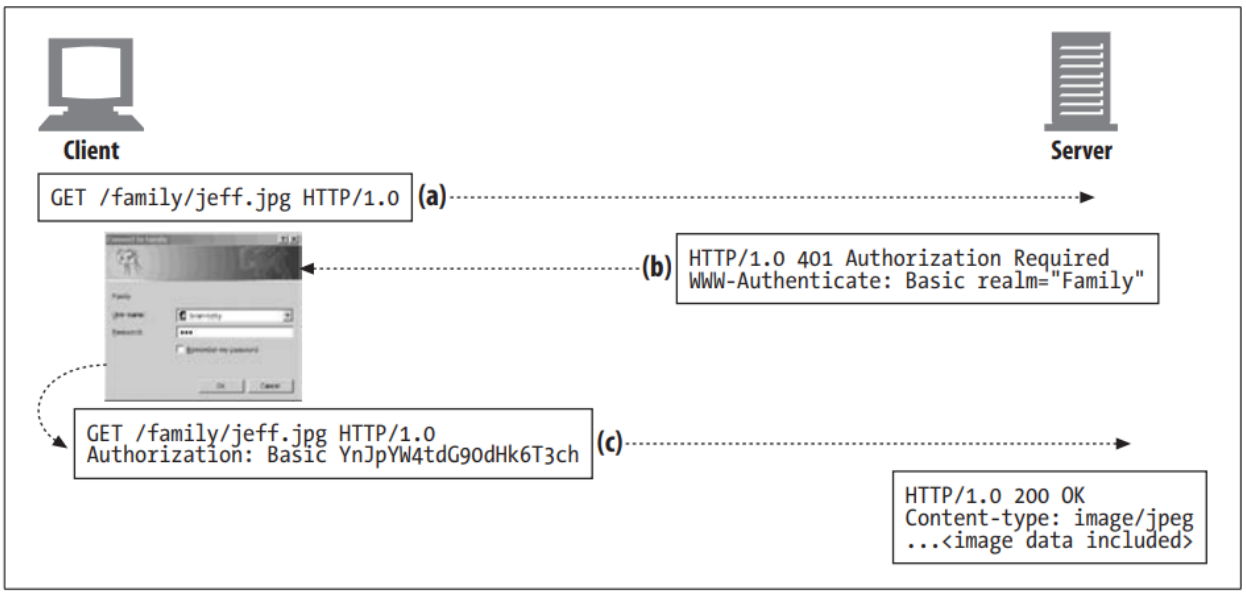
Basic Authentication رایج ترین پروتکل احراز هویت HTTP است. تقریباً هر کلاینت و سرور اصلی Basic Authentication را پیاده سازی می‌کند. Basic Authentication در ابتدا در مشخصات HTTP/1.0 توضیح داده شد، اما از آن زمان به RFC 2617 منتقل شده است، که احراز هویت HTTP را شرح می‌دهد.

در Basic Authentication، یک وب سرور می‌تواند تراکنش را رد کند و کلاینت را برای یک نام کاربری و رمز عبور معتبر به چالش بکشد. سرور چالش احراز هویت را با بازگرداندن کد وضعیت 401 به جای 200 آغاز می‌کند و Security Realm قابل دسترسی با هدر پاسخ WWW-Authenticate را مشخص می‌کند. هنگامی که مرورگر چالش را دریافت می‌کند، کادر محاوره‌ای را باز می‌کند که نام کاربری و رمز

عبور را برای این حوزه درخواست می‌کند. نام کاربری و رمز عبور با فرمت کمی درهم در داخل هدر درخواست مجوز به سرور ارسال می‌شود.

Basic Authentication Example

شکل زیر، که پیش‌تر نیز به آن اشاره شد، نمونه‌ای دقیق از Basic Authentication را نشان می‌دهد:



در بخش a از شکل، یک کاربر عکس خانوادگی شخصی /family/jeff.jpg را درخواست می‌کند.

در بخش b شکل، سرور 401 Authorization Required که مربوط به یک چالش رمز عبور است را برای عکس خانوادگی شخصی به همراه هدر WWW-Authenticate ارسال می‌کند. هدر برای realm به نام Family درخواست Basic Authentication می‌کند.

در بخش c شکل، مرورگر چالش 401 را دریافت می‌کند و کادر محاوره‌ای را باز می‌کند که نام کاربری و رمز عبور را برای realm مربوط به Family درخواست می‌کند. هنگامی که کاربر نام کاربری و رمز عبور را وارد می‌کند، مرورگر آن‌ها را با یک دونقطه به آن‌ها متصل می‌کند، آن‌ها را در به صورت Base-64 "درهم" (که در بخش بعدی بحث می‌شود) کدگذاری نموده و آن‌ها را در سربرگ Authorization ارسال می‌کند.

در بخش d شکل، سرور نام کاربری و رمز عبور را Decode نموده، صحت آن‌ها را تأیید می‌کند و سند درخواستی را در یک پیام HTTP 200 OK برمی‌گرداند.

هدرهای احراز هویت HTTP WWW-Authenticate و Authorization در جدول زیر خلاصه شده است.

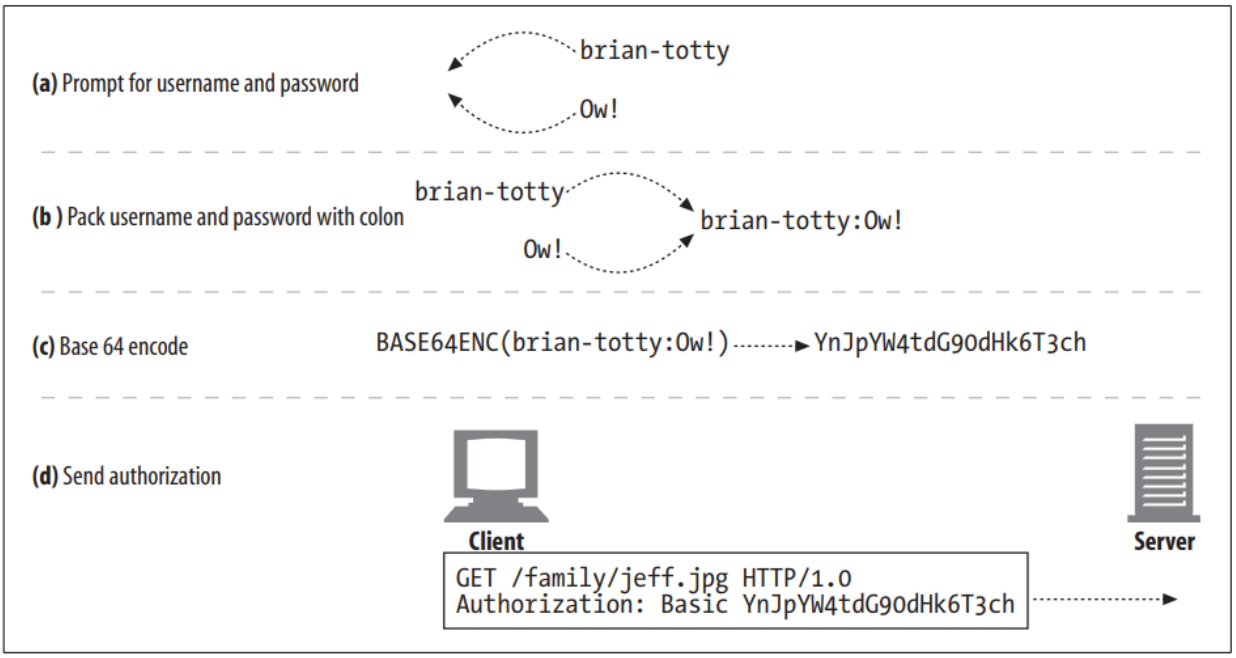
Challenge/Response	Header syntax and description
Challenge (server to client)	There may be different passwords for different parts of the site. The realm is a quoted string that names the set of documents being requested, so the user knows which password to use. `WWW-Authenticate: Basic realm=quoted-realm`
Response (client to server)	The username and password are joined together by a colon (:) and then converted to base-64 encoding, making it a bit easier to include international characters in usernames and passwords and making it less likely that a cursory examination will yield usernames and passwords while watching network traffic. `Authorization: Basic base64-username-and-password`

توجه داشته باشید که پروتکل Basic Authentication از هدر AuthenticationInfo که پیش تر اشاره شد، استفاده نمی‌کند.

Base-64 Username/Password Encoding

Basic Authentication در HTTP، نام کاربری و رمز عبور را با هم بسته بندی می‌کند (با یک دونقطه که از هم جدا شده اند) و آن‌ها را با استفاده از روش کدگذاری Base-64 کدگذاری می‌کند. اگر نمی‌دانید کدگذاری Base-64 چیست، نگران نباشید. نیازی نیست چیز زیادی در مورد آن بدانید. به طور خلاصه، کدگذاری Base-64 دنباله‌ای از بایت‌های ۸ بیتی را می‌گیرد و دنباله بیت‌ها را به تکه‌های ۶ بیتی تقسیم می‌کند. هر قطعه ۶ بیتی برای انتخاب یک کاراکتر در یک 64-character خاص استفاده می‌شود که بیشتر از حروف و اعداد تشکیل شده است.

شکل زیر نمونه‌ای از استفاده از کدگذاری Base-64 برای Basic Authentication را نشان می‌دهد. در اینجا، نام کاربری "brian-totty" و رمز عبور "Ow!" است. مرورگر نام کاربری و رمز عبور را با یک دونقطه می‌پیوندد و رشته پر شده «brian-totty:Ow!» را ایجاد می‌کند. سپس این رشته با کدگذاری Base-64 بدین شکل می‌شود: "YnJpYW4tdG90dHk6T3ch".



رمزگذاری Base-64 برای دریافت رشته‌هایی از داده‌های باینری، متن و کاراکترهای بین‌المللی (که باعث ایجاد مشکلاتی در برخی سیستم‌ها می‌شد) اختراع شد و آن‌ها را به طور موقت به یک الفبای قابل حمل برای انتقال تبدیل کرد. بدین صورت رشته‌های اصلی می‌توانستند در سیستم راه دور بدون ترس از خرابی انتقال، Decode شوند.

کدگذاری Base-64 می‌تواند برای نام‌های کاربری و رمزهای عبور حاوی کاراکترهای بین‌المللی یا سایر کاراکترهایی که در هدرهای HTTP غیرقانونی هستند (مانند علامت‌های نقل قول، دونقطه، و carriage return) مفید باشد. همچنین، از آنجایی که کدگذاری Base-64 به طور پیش پا افتاده نام کاربری و رمز عبور را به هم می‌زند، می‌تواند به جلوگیری از مشاهده تصادفی نام‌های کاربری و رمزهای عبور مدیران هنگام مدیریت سرورها و شبکه‌ها کمک کند.

Proxy Authentication

احراز هویت همچنین می‌تواند توسط سرورهای پروکسی واسطه انجام شود. برخی از سازمان‌ها از سرورهای پراکسی برای احراز هویت کاربران قبل از اینکه به آن‌ها اجازه دسترسی به سرورها، شبکه‌های محلی یا شبکه‌های بی‌سیم را بدهند، استفاده می‌کنند. سرورهای پراکسی می‌توانند راهی مناسب برای ارائه کنترل دسترسی یکپارچه در منابع سازمان باشند، زیرا خط‌مشی‌های دسترسی را می‌توان به صورت متمرکز بر روی سرور پراکسی مدیریت کرد. اولین گام در این فرآیند، ایجاد هویت از طریق احراز هویت پروکسی است.

مراحل مربوط به احراز هویت پروکسی با مراحل شناسایی وب سرور یکسان است. با این حال، هدرها و کدهای وضعیت متفاوت هستند. جدول زیر کدهای وضعیت و هدرهای مورد استفاده در وب سرور و احراز هویت پروکسی را در تضاد قرار می‌دهد.

Web server	Proxy server
Unauthorized status code: 401	Unauthorized status code: 407
WWW-Authenticate	Proxy-Authenticate
Authorization	Proxy-Authorization
Authentication-Info	Proxy-Authentication-Info

The Security Flaws of Basic Authentication

Basic Authentication ساده و راحت است، اما ایمن نیست. این فقط باید برای جلوگیری از دسترسی غیرعمدی از طرف‌های غیر مخرب استفاده شود یا در ترکیب با یک فناوری رمزگذاری مانند SSL استفاده شود.

نقایص امنیتی زیر را در نظر بگیرید:

- Basic Authentication، نام کاربری و رمز عبور را به شکلی در سراسر شبکه ارسال می‌کند که می‌توان آن را Decode کرد. در واقع، Secret Password به صورت واضح ارسال می‌شود تا هر کسی بتواند آن را بخواند و بگیرد. کدگذاری Base-64 نام کاربری و رمز عبور را مبهم می‌کند و احتمال اینکه مهمان‌های دوستانه با مشاهده تصادفی شبکه رمزهای عبور را جمع‌آوری کنند، کمتر می‌شود. با این حال، با توجه به نام کاربری و رمز عبور کدگذاری شده با Base-64، Decode را می‌توان با معکوس کردن فرآیند کدگذاری انجام داد. Decode حتی در چند ثانیه، با دست، با مداد و کاغذ قابل انجام است! گذرواژه‌های کدگذاری شده با Base-64 به طور موثر در حالت شفاف ارسال می‌شوند. فرض کنید که اشخاص ثالث با انگیزه مخرب، نام کاربری و رمز عبور ارسال شده توسط Basic Authentication را رهگیری می‌کنند. اگر این یک نگرانی است، تمام تراکنش‌های HTTP خود را از طریق کانال‌های رمزگذاری شده SSL ارسال کنید یا از پروتکل احراز هویت امن‌تر مانند Digest Authentication استفاده کنید.
- حتی اگر Secret Password در طرحی کدگذاری شده باشد که Decode نمودن آن پیچیده‌تر باشد، شخص ثالث همچنان می‌تواند نام کاربری و رمز عبور مخدوش را ضبط کند و اطلاعات مخدوش را بارها و بارها برای دسترسی به سرورهای اصلی پخش کند. هیچ تلاشی برای جلوگیری از این حملات تکراری انجام نمی‌شود.

- حتی اگر Basic Authentication برای برنامه‌های غیر بحرانی مانند کنترل دسترسی اینترانت شرکتی یا محتوای شخصی‌شده استفاده شود، رفتار اجتماعی این امر را خطرناک می‌کند. بسیاری از کاربران، غرق در سرویس‌های محافظت شده با رمز عبور، نام‌های کاربری و رمز عبور را به اشتراک می‌گذارند. یک شخص باهوش و مخرب ممکن است نام کاربری و رمز عبور را به طور واضح از یک سایت ایمیل اینترنتی رایگان دریافت کند و متوجه شود که همان نام کاربری و رمز عبور اجازه دسترسی به سایت‌های مهم بانکداری آنلاین را می‌دهد!
- احراز هویت اولیه هیچ محافظتی در برابر پروکسی‌ها یا واسطه‌هایی که به عنوان واسطه عمل می‌کنند، ارائه نمی‌دهد، هدرهای احراز هویت دست نخورده باقی می‌ماند، اما بقیه پیام را تغییر می‌دهد تا ماهیت تراکنش را به شدت تغییر دهد.
- Basic Authentication در برابر جعل توسط سرورهای تقلبی آسیبپذیر است. اگر بتوان کاربر را به این باور رساند که در حال اتصال به یک میزبان معتبر محافظت شده توسط Basic Authentication است، در واقع زمانی که در حال اتصال به یک سرور یا دروازه متخاصم است، مهاجم می‌تواند رمز عبور درخواست کند، آن را برای استفاده بعدی ذخیره کند، و تظاهر به یک خطا کند.

با همه این‌ها، Basic Authentication هنوز برای ارائه شخصی‌سازی راحت یا کنترل دسترسی به اسناد در یک محیط دوستانه، یا در مواردی که حریم خصوصی مورد نظر است اما کاملاً ضروری نیست، مفید است. به این ترتیب، Basic Authentication برای جلوگیری از دسترسی تصادفی توسط کاربران کنجکاو استفاده می‌شود.

به عنوان مثال، در داخل یک شرکت، مدیریت محصول ممکن است با رمز عبور از برنامه‌های محصول آینده محافظت کند تا توزیع زودرس را محدود کند. Basic Authentication دسترسی به این داده‌ها را برای طرف‌های دوستانه، به اندازه کافی ناخوشایند می‌کند. به همین ترتیب، ممکن است از عکس‌های شخصی یا وب‌سایت‌های خصوصی که کاملاً محرمانه نیستند یا حاوی اطلاعات ارزشمندی نیستند، با رمز عبور محافظت کنید، اما واقعاً به کسب و کار شخص دیگری هم مربوط نمی‌شود.

Basic Authentication را می‌توان با ترکیب آن با انتقال داده‌های رمزگذاری شده (مانند SSL) ایمن کرد تا نام کاربری و رمز عبور از افراد مخرب پنهان شود. این یک تکنیک رایج است.

ما در مورد رمزگذاری ایمن در فصل ۱۴ بحث می‌کنیم. فصل بعدی یک پروتکل احراز هویت HTTP پیچیده‌تر، Digest Authentication را توضیح می‌دهد که دارای ویژگی‌های امنیتی قوی‌تری نسبت به Basic Authentication است.

For More Information

http://www.ietf.org/rfc/rfc2617.txt

http://www.ietf.org/rfc/rfc2616.txt