به نام خدا

درس برنامه‌سازی پیشرفته

توضیحات احراز هویت

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال دوم ۰۳ـ۰۲

استاد:

دکتر محمدامین فضلی

فهرست

احراز هویت

مقدمه‌ای بر Authentication و Authorization

فرض کنید می‌خواهید بازارچه‌ای اینترنتی درست کنید که افراد در آن امکان خرید و فروش کالاهای خود را دارند. از مهم‌ترین نکاتی که باید در نظر بگیرید این است که هر فرد در پلتفرم شما چه هویتی دارد (مشتری است یا فروشنده؟ چه اطلاعات حقوقی یا حقیقی دارد؟ و...)
در قدم بعدی باید به این فکر باشید که هر شخص در پلتفرم چه دسترسی‌هایی با توجه به هویت خود دارد. برای مثال یک مشتری نباید قادر به تغییر قیمت یک کالا باشد! هر فروشنده تنها بتواند کالاهای مربوط به خود و زیرمجموعه‌هایش را ویرایش کند و ...
به اولین مفهوم، Authentication (احراز هویت) و به مفهوم دوم، Authorization (تایید صلاحیت) می‌گویند. بدیهی است به طور کلی فرآیند Authorization باید بعد از Authentication انجام شود.

Session

هر session یک فایل به فرمتی شبیه JSON در سمت سرور است که حاوی اطلاعاتی شامل ID یکتا برای هر کاربر، زمان آخرین لاگین و مدت انقضای آن است. این فایل در سمت سرور ساخته و ذخیره می‌شود و در ادامه با درخواست‌های بعدی، سرور می‌تواند کاربر را شناسایی کند.
بعد از ساخته شدن این فایل، بعضی از اطلاعات آن مانند ID در قالب Cookie به کاربر فرستاده می‌شود. در ادامه و در هر درخواست، cookie نیز به سرور فرستاده می‌شود و باعث authenticate شدن کاربر در صورت یکسان بودن ID موجود در cookie و ID ذخیره شده در سرور می‌شود.
ذخیره شدن اطلاعات session در سمت سرور، مزایا و معایبی به دنبال دارد. از مزایای این روش می‌توان به در دسترس بودن همه session ها برای administrator های شبکه اشاره کرد. Admin شبکه می‌تواند در صورت مشاهده هر حرکت مشکوک، session را حذف کند و فرایند احراز هویت کاربر را ملغی کند.
در سمت دیگر، این ذخیره شدن در سمت سرور، باعث افزایش لود سرور می‌شود و کارایی آن را پایین می‌آورد.
از حملات معروفی که سیستم‌های Session Based در معرض آن‌ها هستند می‌توان به Cross-site request forgery و Man-in-the-middle اشاره کرد.

Token-Based Authentication

با مفهوم Session-Based Authentication آشنا شدید و به معایب آن پی بردید. از جمله این که وقتی کاربر وارد می‌شود یک Session ID در دیتابیس ذخیره می‌شود. با هر درخواست کاربر سرور باید برای بررسی صحت Session ID به دیتابیس مراجعه کند که این زمان‌بر است. همچنین با افزایش تعداد کاربران مشکل Scale شدن سرور به وجود می‌آید و ...
در این بخش با روشی آشنا می‌شویم که می‌تواند درخواست به دیتابیس را به حداقل برساند، مشکل Scale شدن سرورها را حل کند و حتّی بدون هزینه اضافه روی سرور تاریخ انقضایی بر روی آخرین ورود کاربر تعریف کند!
فرآیند احراز هویت با توکن بدین صورت است که ابتدا کاربر نام کاربری و رمزعبورش را به سرور می‌فرستد و به عنوان پاسخ یک توکن تعریف می‌کند. سپس با این توکن می‌تواند تا مدت معینی اعمال مورد نیازش را

انجام دهد. زمانی هم که توکن منقضی شد، دوباره با فرستادن اطلاعات هویتی می‌تواند توکن جدیدی دریافت کند. بدین صورت نیاز نیست با هر درخواست، نام کاربری و رمزعبورش را هم ارسال کند.

از استانداردهای معروف مورد استفاده برای این امر JSON Web Token یا به اختصار JWT است. ابتدا بهتر است با مفهوم Hash آشنا شوید و سپس به عنوان نمونه با یکی از الگوریتم‌های رمزنگاری معروف با نام SHA-۲۵۶ کار کنید. حال که کمی با مفهوم رمزنگاری آشنا شدید، می‌توانید این ویدئو در مورد JWT را مشاهده کنید و در نهایت با یک نمونه از آن دست و پنجه نرم کنید.

Authorize and Permissions

به فرایند تخصیص دسترسی به منابع (Resources) مختلف مانند فایل‌ها، سرویس‌ها، برنامه‌هایی که روی سیستم نصب شده‌اند و داده‌ها به کاربران، Authorization گفته می‌شود. این فرایند، زیر شاخه امنیت اطلاعات (Information Security) می‌باشد. یکی از معروف‌ترین مثال‌ها در مبحث Authorization، مدیر سیستم admin) یا (superuser می‌باشد که دسترسی به اطلاعات بقیه کاربران و تغییر اطلاعات و ... را دارد.

سیستم‌های تامین امنیت امروزی ابتدا فرایند احراز هویت (authentication) و سپس دسترسی به منابع (authorization) را اجرا می‌کنند و در نهایت، اجازه کار با سیستم را به کاربر می‌دهند. هر دسترسی به یک منبع، Permission نام دارد. هر دسترسی نیر قابلیت‌هایی به کاربر می‌دهد؛ پس هر کاربر با یک Permission خاص، قابلیت انجام یک operation خاص در سیستم را دارد.

برای مثال، در یک دیتابیس می‌توان دسترسی‌های یک کاربر را تغییر داد. اگر بخواهیم در یک دیتابیس Postgres، به یک کاربر تنها دسترسی read بدهیم (تنها بتواند کوئری بزند و داده‌ها را دریافت کند) می‌توانیم از دستور زیر استفاده کنیم:

```
GRANT SELECT ON ALL TABLES IN SCHEMA <schema-name> TO <user-name>a
```