Report

No .4

By Amir Lotfi

Contents

Surrogate Key

کلیدی که از محتویات رکوردها یا از سوی یک سازنده برداشت میشود. یک دنباله عددی یا هش رکوردها، کلید نهایی را تولید و آماده بهره برداری میکند. این کلیدهای زمانی که هیچ کلید طبیعی نداشته باشیم، جایگزین خوبی خواهند بود؛ البته میتوانیم از کلیدهای کاندید دیگری نیز استفاده کنیم.

چرا کلیدهای جانشینی را ترجیح میدهیم ؟!

- این کلیدها غیر قابل تغییر هستند؛ گاهی نیاز داریم که رکوردهای جداول را بروز کنیم در صورت بروز رسانی ستونهایی که کلید هستند، امکان نقض جامعیت داده ها بوجود میاید.
- کلیدهای طبیعی، به منطق سیستم وابسته هستند، شاید این کلیدها را در آینده تغییر دهیم پس باید جداول را عوض کنیم و به تبع کدهای سیستم را نیز عوض میکنیم. ما نباید جامعیت را به پایداری سیستم گره بزنیم.

Authentication

فرایندی که شخصی یا چیزی در آن اثبات کند که خودش است، اغلب آنرا احراز هویت با نام کاربری و رمز عبور در نظر میگیریم که تقریباً درست است. یک سیستم دارای یک سری منابع است و خدماتی را به کاربران ارائه میدهند ولی نه به هر کس؛ بلکه باید در سیستم از قبل شناسه هایی برای احراز هویت خود داشته باشند.

Authorization

حال کاربری وارد سیستم شده است، آیا این کاربر اجازه دسترسی به همه منابع را دارد یا تنها باید به منابع و سرویسهای محدود دسترسی داشته باشد؟! سرویس پخش موزیک اسپاتیفای را در نظر میگیریم. همه بعد از احراز هویت وارد سیستم میشوند و موزیکها را آنلاین گوش میدهند اما کسانی که مبلغی را پرداخت کنند میتوانند موزیکها را دانلود کنند و آفلاین به آنها گوش بدهند. **فقط کسانی که مبلغی را پرداخت کرده باشند اجازه دارند.**

Web Applications Authentication Mechanism

روشهای عمده و محبوب برای احراز هویت کاربران در برنامه های تحت وب در دسته های زیر قرار میگیرند.

1 .Basic-Based

به واسطه username, password که این دو با Base-64 کدگذاری میشوند و در پروتکل HTTP در داخل هدر Authorization با ذخیره میشود. ابتدا کلمه Basic و یک فاصله سپس مقدار کدگذاری شده در آن بخش قرار میگیرد. به مجموع username, password اغلب credentials گفته میشود.

```
Authentication : Basic <Base64(username:password)>

if
{
    username : amirlotfi,
    password : 123456789
}
then
Authentication : Basic YW1pcmxvdGZpOjEyMzQ1Njc4OQ==
```

سپس Servlet این دیتاها را میخواند و در داخل شی Authentication قرار میدهد. در سوی سیستم این دادههای امنیتی در جداول ذخیره میشوند، البته نه به صورت خام. اغلب سیستمها از رمز عبور کاربران بی اطلاع هستند و مقادیر هش شده آنها را ذخیره میکنند.

2 .Digest-Based

فرایند بالا تکرار میشوند با این تفاوت، مقداری که در credentials قرار میگیرد یک مقدار هش شده است. سرور نیز در صورت تطابق اجازه ورود کاربر به سیستم را میدهد. توابعی مانند MD-5 عملیات هش را انجام میدهد. البته در این روش تنها credentials کاربر در رشته هش قرار ندارد؛ متغیرهای دیگری نیز از سوی کلاینت و سرور، هردو، در این فرایند دخیل هستند.

3 .Assertions-Based

در این روش، یک سامانه یا موجودیت مورد اعتماد به نام identity provider یا IDP یک امضا از روی credentials کاربر تولید میکند. سپس این امضا توسط یک Service Provider یا سرور باز میشود و کاربر را احراز هویت میکند. امضا بر روی بستر XML صورت میگیرد.

4 .Token Base

توکنها شبیه به کارتهای ورود عمل میکنند. یک قطعه رمز گذاری شده با ساختار معیین که مجموعه از اطلاعات کاربران برای احراز هویت و اعطای مجوز دسترسی به منابع را در خود نگهداری میکند. این قطعه ها اغلب بر روی فرمت JSON سوار شدند. خود این توکنها به چهار مجموعه تقسیم میشوند JWT, OIDC, OAuth Access Token & Refresh Token .

زمانی که کاربر احراز هویت را انجام میدهد، داده ها از سوی سرور اهراز هویت تبدیل به توکن میشوند، سپس کاربر این توکن را به سرور دیگر که منابع را دارد ارسال میکند. سرور منابع و خدمات، توکن را تجزیه و تحلیل میکند و با توجه به اطلاعات داخل آن به کاربر مجوز دسترسی را میدهد تا از منابع یا خدمات استفاده کند.

این روش شباهتهای زیاد با Assertion –Based دارد البته تفاوتها معطوف به پروتکل و محل استفاده آنها است. اغلب توکنها را در برنامه های کاربردی روزمره و طرف دیگر را در برنامه های بزرگ و صنعتی استفاده میکنند.

5 .Session-Based

کاربران اهراز هویت را انجام میدهند، سپس داده های مرتبط با آنها در بانک اطلاعاتی سرور تحت یک Session Identifier ذخیره میشود سپس این اطلاعات به کاربر در قالب کوکی ارسال میشود و مرورگرِ کاربر در درخواستهای بعدی از این کوکی برای اهراز هویت استفاده میکند.

6 .Cookie-Based

شبیه به مورد Session-Based عمل میکند اما کوکی را **فقط** در سوی سیستم کاربر ذخیره میکند و هرباری که کاربر درخواستی به سوی سرور ارسال میکند، کوکی را نیز توام با آن ارسال میکند.

این روش بسیار کم مورد استفاده قرار میگیرد؛ امنیت، مقیاس پذیری از جمله خصوصیتهای مثبت این روش هستند اما فلسفه REST را زیر سوال میبرند. زیرا وضعیت کاربران در سیستم ذخیره میشود و این برخلاف Stateless بودن این معماری است.