**معرفی پروتکل های احراز هویت**

**: (Security Assertion Markup Language) SAML**

اجازه دسترسی به برنامه های مختلف را فقط با یک بار احراز هویت میدهد و به عبارت دیگر با ارائه روشی فناوری SSO را پیاده سازی می کند که پروتکل استاندارد را بر اساس XML ارائه می دهد.

SSO یا Single Sign On چیست؟

یک سرویس متمرکز تائید هویت است که تنها با یک نام کاربری کاربر میتواند به چند برنامه دسترسی داشته باشد. به عنوان مثال اگر در google حساب Gmail خود را ثبت کنیم میتوانیم به چند برنامه مانند Youtube و یا غیره دسترسی داشته باشیم.

**فرآیند کار SSO های مبتنی بر SAML**

1. کاربر درخواست آغاز SSO را به Service Provider میدهد.
2. Service Provider یک درخواست رمز نگاری شده ایجاد کرده و به Identity Provider ارسال می کند.
3. Identity Provider درخواست را تائید کرده و از کاربر میخواهد تا Login کند.
4. Identity Provider فرم XHTML را همراه با پاسخ رمز نگاری شده(Base64) به کاربر ارسال می کند.
5. کاربر پاسخ را به SAML ارسال می کند.
6. Service Provider پاسخ را دریافت کرده و کاربر را به منبع هدف Redirect می کند.

**OpenID :**

تکنولوژی احراز هویت کاربری در یک وبگاه از طریق وبگاه دیگر است .در واقع همان پیاده سازی فناوری SSO به شمار می رود. مانند SAML. به درگاهی که درخواست احراز هویت را می دهد OpenID Provider می گویند.

**: OAUTH**

پروتکلی برای دادن مجوز دسترسی از یک سرویس به سرویس دیگر بدون به اشتراک گذاشتن نام کاربری و رمز عبور است.یکی از رایج ترین روش های انتقال مجوز از یک سرویس یک پارچه احراز هویت (SSO) به سامانه های دیگر است.

**امنیت در معماری MicroService**

**چالش های پیاده سازی سیستم احراز هویت در این معماری**

**1. وجود کد تکراری در هر سرویس برای احراز هویت**

**2. نقض اساسی ترین اصل در معماری میکروسرویس به نام Single Concern**

**3. ایجاد پیچیدگی**

**3 رویکرد برای پیاده سازی سیستم احراز هویت در معماری میکروسرویس**

**1. Edge-Level Authorazation**

**در این نوع احراز هویت، به صورت ساده، سیستم احراز هویت در حاشیه اتفاق می افتد و عمدتا از** [Api Gateway](#ApiGateway) **ها برای پیاده سازی آن استفاده می کنند. که دارای معایب زیر می باشد:**

* **امنیت پایین: اگر مهاجم بتواند از** [API Gateway](#ApiGateway) **عبور کند میتواند به تمام سرویس ها دسترسی پیدا کند که این مسئله، اصل "**[Defence In Depth](#DID)**" را نقض می کند.**
* **سخت بودن مدیریت: با افزایش پیچیدگی در دسترسی های نرم افزار درخواست های احراز ارسالی به** [API Gateway](#ApiGateway) **به مرور ساخت و غیر قابل مدیریت خواهد بود.**

**2. Service-Level Authorization**

**این استراتژی باعث می شود تا همه سرویس ها مستقلا سیستم احراز هویت خود را بدست بگیرند و به طور مستقل سیاست های امنیتی خود را پیاده سازی کنند.این رویکرد موارد زیر را دربر میگیرد:**

* **نقطه مدیریت سیاست امنیتی: به مدیران مجوز ساخت، مدیریت و تست دسترسی را می دهد.**
* **نقطه تصمیم سیاست امنیتی: چک میکند که درخواست جاری چه نوع دسترسی را دربر میگیرد تا اعطای دسترسی یا رد درخواست را ارزیابی کند.**
* **نقطه اجرای سیاست امنیتی: تصمیمات دسترسی را فراهم می کند و سیاست دسترسی را برای درخواست های خاص اعمال می کند.**
* **نقطه اطلاعات سیاست امنیتی: به عناصر در سیستم اجازه بازیابی اطلاعات برای بررسی ویژگی های حساب کاربری را می دهد.**

**3. External Entity Identity Propagation:**

**الگوی Api Gateway :**

تعداد درخواست ها و پاسخ های رفت رو برگشت از کاربر به سرور و از سرور به کاربر را کنترل می کند و باعث ایجاد یکپارچگی بین سرویس ها و کاربران می شود.به عبارت دیگر با ایجاد یک نقطه ورودی به تمام درخواست های مشتریان عمل می کند.پیاده سازی این الگو باعث کاهش افزونگی در کد و ارتقای ثبات و بستری برای پیاده سازی عملیات احراز هویت و ثبت نام و محدود کردن دسترسی ها میگردد.اما سوالی در این میان پیش می آید و آن اینست که اگر هکر بتواند با هر ترفندی از این Gateway عبور کند به تمام سرویس ها دسترسی خواهد داشت.در این صورت چه باید کرد؟

**الگوی BFF :**

مخفف عبارت Backend For Frontend است

**مباحث تکمیلی**

**Defense In Depth یا DID چیست؟**

دفاع در عمق یا DID به یک رویکرد امنیت اطلاعات اشاره دارد که در آن یک سری مکانیزم ها و کنترل های امنیتی به طور مدبرانه در سراسر یک شبکه کامپیوتری لایه بندی می شوند تا از محرمانه بودن، یکپارچگی و در دسترس بودن شبکه و داده های درون آن محافظت کنند. با این رویکرد در صورتی که یکی از مکانیزم ها شکست بخورد، در لایه ی بعدی مکانیزم بعدی جلوی خطر را خواهد گرفت. این رویکرد به طور قابل توجهی امنیت شبکه را در برابر بسیاری از حمله ها تقویت می کند.