Artigo
Vinicius Maranh... · Dez. 21, 2020 4min de leitura

Protegendo suas APIs com OAuth 2.0 no InterSystems API Management – Parte 2

Nesta série de artigos de três partes, é mostrado como você pode usar o IAM para simplesmente adicionar segurança, de acordo com os padrões do OAuth 2.0, a um serviço não autenticado anteriormente implantado no IRIS.

Na primeira parte, foram fornecidos alguns conhecimentos sobre o OAuth 2.0, juntamente com algumas definições e configurações iniciais do IRIS e IAM, para facilitar a compreensão de todo o processo de proteção dos seus serviços.

Esta parte, agora, discutirá e mostrará em detalhes as etapas necessárias para configurar o IAM para validar o token de acesso presente na solicitação de entrada e encaminhar a solicitação para o back-end se a validação for bem-sucedida.

A última parte desta série discutirá e demonstrará as configurações necessárias para o IAM gerar um token de acesso (atuando como um servidor de autorização) e validá-lo, junto com algumas considerações finais importantes.

Se você quiser testar o IAM, entre em contato com seu representante de vendas InterSystems.

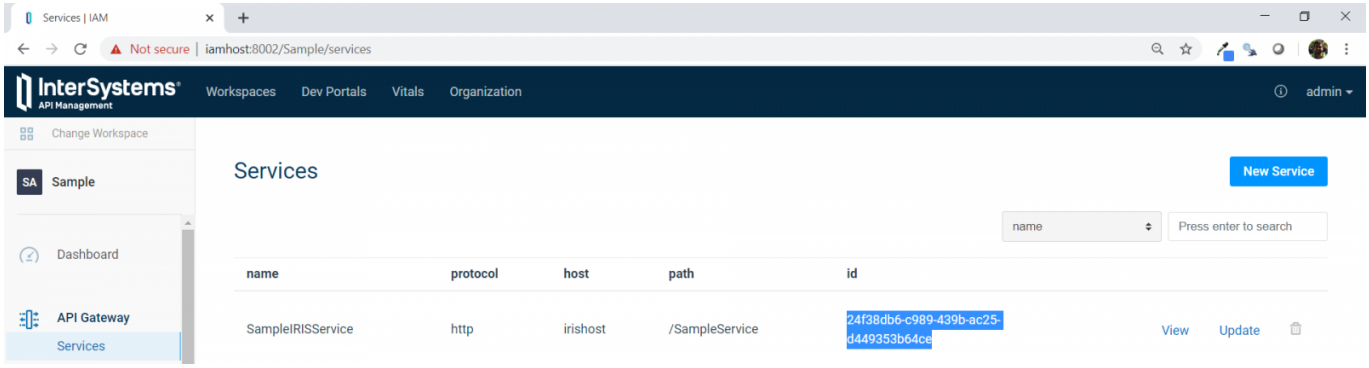
Cenário 1: IAM como um validador de token de acesso

Neste cenário, será usado um servidor de autorização externo que gera um token de acesso em formato JWT (JSON Web Token). Este JWT é assinado usando o algoritmo RS256 junto com uma chave privada. Para verificar a assinatura do JWT, a outra parte (neste caso, o IAM) precisa ter a chave pública, fornecida pelo servidor de autorização.

Este JWT gerado pelo servidor de autorização externo também inclui, em seu corpo, uma declaração chamada “ exp ” contendo o carimbo de data/hora (timestamp) de quando esse token expirará, e outra declaração chamada “ iss ” contendo o endereço do servidor de autorização.

Portanto, o IAM precisa verificar a assinatura do JWT com a chave pública do servidor de autorização e o carimbo de data/hora de expiração contido na declaração "exp" dentro do JWT antes de encaminhar a solicitação ao IRIS.

Para configurar isso no IAM, vamos começar adicionando um plugin chamado “ JWT ” ao nosso “ SampleIRISService ” no IAM. Para isso, acesse a página Services do IAM e copie o id do “ SampleIRISService ”, que usaremos posteriormente.



Depois disso, vá em Plugins, clique no botão " New Plugin ", localize o plugin " JWT " e clique em Enable.

Na página seguinte, cole o id do " SampleIRISService " no campo " serviceid " e selecione a caixa " exp " no parâmetro " config.claimstoverify ".

Observe que o valor do parâmetro " config.keyclaimname " é " iss ". Vamos usar isso mais tarde.

Em seguida, clique no botão " Create ".

Feito isso, vá até a seção " Consumers " no menu à esquerda e clique em nosso " ClientApp " criado anteriormente. Acesse a aba " Credentials " e clique no botão " New JWT Credential ".

Na página seguinte, selecione o algoritmo usado para assinar o JWT (neste caso RS256) e cole a chave pública no campo " rsapublickey " (esta é a chave pública fornecida a você pelo servidor de autorização em formato PEM).

No campo " key ", você precisa inserir o conteúdo da declaração JWT que você inseriu no campo " config.keyclaimname " ao adicionar o plugin JWT. Portanto, neste caso, preciso inserir o conteúdo da declaração iss do meu JWT, que, no meu caso, é o endereço do servidor de autorização.

Em seguida, clique no botão " Create ".
Dica: para fins de depuração, existe uma ferramenta on-line de decodificação de JWT que você pode usar para verificar as declarações e seus valores e verificar a assinatura colando a chave pública. Aqui está o link desta ferramenta on-line: https://jwt.io/#debugger

Agora, com o plugin JWT adicionado, não é mais possível enviar a solicitação sem uma autenticação. Como você pode ver abaixo, em uma simples solicitação GET, sem autenticação, para a URL

http://iamhost:8000/event/1

retorna uma mensagem não autorizada juntamente com o código de status " 401 Não autorizado ".

Para obter os resultados do IRIS, precisamos adicionar o JWT à solicitação.

Portanto, primeiro precisamos solicitar o JWT ao servidor de autorização. O servidor de autorização personalizado que estamos usando aqui retorna um JWT se uma solicitação POST for feita junto com alguns pares de valores-chave no corpo, incluindo informações de usuário e cliente, para a seguinte URL:

https://authorizationserver:5001/auth

Isto é como se parece essa solicitação e a sua resposta:

Em seguida, você pode adicionar o JWT obtido na resposta abaixo no cabeçalho de autorização como um Bearer Token e enviar uma solicitação GET para a mesma URL usada anteriormente:

http://iamhost:8000/event/1

Ou você também pode adicioná-lo como um parâmetro de querystring, com a chave de querystring sendo o valor especificado no campo " config.uriparamnames " ao adicionar o plugin JWT que, neste caso, é " jwt " :

Finalmente, existe também a opção de incluir JWT na solicitação como um cookie, se algum nome for inserido no campo " config.cookienames " .

Continue lendo até a terceira e última parte desta série para entender as configurações necessárias para o IAM gerar um token de acesso e validá-lo, junto com algumas considerações finais importantes.

#API #OAuth2 #REST API #Segurança #InterSystems IRIS

URL de origem:https://pt.community.intersystems.com/post/protegendo-suas-apis-com-oauth-20-no-intersystems-api-management-%E2%80%93-parte-2