HISCOX

Auth0 Private Key JWT – Client credentials Integration Guide

Step-by-Step Guide: Integrating with Us Using Private Key JWT

Instructions for Secure Client Integration

Overview

This guide provides step-by-step instructions for integrating your application with our services using Private Key JWT (JSON Web Token) authentication. Private Key JWT is a secure, standards-based approach that allows your system to authenticate using a cryptographically signed token.

Step 1: Generate a Key Pair

- Generate a public/private key pair using RSA we enforce a minimum RSA key size of 2048 bits and a maximum key size of 4096 bits (open SSL is a good tool for performing this step)

For example:
To generate a public and private key pair.
`openssl genrsa -out test_key.pem 2048`
Now to extract the public key in PEM format.
`openssl rsa -in test_key.pem -outform PEM -pubout -out test_key.pem.pub`

- Store your private key securely; do not share it with anyone.
- Provide your public key to our integration support team as directed.

Step 2: Register Your Public Key

- Send your public key, in PEM or JWK format, to our onboarding contact
- We will register your public key and associate it with your client credentials in our system.

Step 3: Obtain Client Credentials

- After registration, you will receive a client identifier `(client_id)` and any additional integration details required for authentication.

Step 4: Construct the JWT Assertion

- Create a JWT (JSON Web Token) with the following claims
- In the jwt header:
  - alg (algorithm): should be `RS256`
- In the main token:
  - iss (issuer): Your `client_id`
  - sub (subject): Your `client_id`
  - iat (issued at): The time the token is generated
  - exp (expiration): Expiry time, maximum 5 minutes from issued time
  - aud (audience): `https://ext.auth.uk.hiscox.com/`
  - jti (JWT ID): A unique identifier for the token (e.g., a UUID)

Sign the JWT using your private key and the appropriate algorithm (e.g., RS256 for RSA).

An example of the finished claims would look like this:

```
{
  "alg": "RS256"
},
{
  "iss": "y72Q5ylUWHE4fEphAnK5UZMT4ZomA2sS",
  "sub": "y72Q5ylUWHE4fEphAnK5UZMT4ZomA2sS",
  "iat": 1754557355,
  "exp": 1754557605,
  "jti": "77b45523-bdb7-4755-be3c-f321d864b157",
  "aud": "https://ext.auth.uk.hiscox.com/"
}
```

Step 5: Request an Access Token

- Send a token request to our OAuth 2.0 token endpoint
  `POST: https://ext.auth.uk.hiscox.com/oauth/token`

With the following form encoded parameters in the body:

- `client_assertion_type : urn:ietf:params:oauth:grant-type:jwt-bearer`
- `client_assertion : (The signed jwt from the previous step)`
- `grant_type : client_credentials`
- `audience : https://uk.nexus.hiscox.com/api/`

Use HTTPS to ensure the request is sent securely.

Step 6: Handle the Access Token Response

- If authentication is successful, you will receive a JSON response containing:
- access_token: The bearer token for API calls
- token_type: Should be "Bearer"
- expires_in: The token's lifetime in seconds

Store the access token securely and use it in the Authorization header for subsequent API requests to the audience specified above.

Step 7: Make Authenticated API Requests

- Include the received access token in the HTTP Authorization header as:
- Authorization: Bearer

Follow our API documentation for endpoint details, request formats, and expected responses.

Step 8: Token Renewal

- When the access token expires, repeat Steps 4-6 to request a new token using a freshly signed JWT assertion.

Key Rotation

- We can support a second credential running alongside the primary in order to allow key rotation without service interruption. A key lifecycle can be agreed with our on-boarding contact.

Tips and Best Practices

- Always protect your private key; never share it or expose it in client-side code.
- Rotate your keys periodically and notify us of new public keys as necessary.

For further assistance or questions, please contact our integration support team.