Token Family Rotation & Reuse Detection

Client
OAuth2 Client Application

Server
Authorization Server

Database
oauth2_tokens table

SCENARIO 1: Normal Token Rotation

1. Client Requests Token Refresh

POST /oauth/token

grant_type=refresh_token
refresh_token=RT1

2. Server Validates Refresh Token

BEGIN IMMEDIATE

SELECT * FROM oauth2_tokens
WHERE refresh_token=RT1 AND revoked=0
Result: token_family_id=fam123, refresh_token_used=0

3. Mark Old Token as Used (Not Revoked)

UPDATE

UPDATE oauth2_tokens SET refresh_token_used=1
WHERE refresh_token=RT1

4. Generate New Tokens (Same Family)

New access_token: AT2
New refresh_token: RT2
token_family_id: fam123 (inherited)
refresh_token_used: 0

5. Store New Tokens

INSERT

INSERT INTO oauth2_tokens
(access_token=AT2, refresh_token=RT2,
token_family_id=fam123, refresh_token_used=0)

6. Return New Tokens to Client

200 OK

{access_token: AT2, refresh_token: RT2}
Client stores RT2, discards RT1

Time passes... Client continues using RT2 normally

SCENARIO 2: Refresh Token Reuse Detection

⚠ SECURITY BREACH: Attacker obtains old RT1 from logs/network

1. Attacker Attempts to Use Old Token

POST /oauth/token

grant_type=refresh_token
refresh_token=RT1 (OLD TOKEN)

2. Server Detects Token Already Used

SELECT

SELECT * FROM oauth2_tokens
WHERE refresh_token=RT1
Result: refresh_token_used=1 ⚠ REUSE!

3. Revoke Entire Token Family

REVOKE ALL

UPDATE oauth2_tokens
SET revoked=1, revoked_at=NOW()
WHERE token_family_id=fam123
→ Revokes RT1, RT2, AT2, all family tokens

4. Log Critical Security Event

INSERT INTO security_events
(event_type=token_reuse, severity=critical)

5. Return Error to Client

400 Bad Request

error: Token reuse detected
All tokens revoked, re-auth required

Impact: Legitimate client's RT2 now invalid

User must re-authenticate through normal OAuth2 flow