Securing Your Minimal API

Kevin Dockx
Architect

@KevinDockx | www.kevindockx.com

Coming Up

High-level API security overview

Token-based security

- Specific to minimal APIs
- Options for token generation

Authorization and authorization policies

High-level API Security Overview

There’s no “one size fits all” approach when it comes to securing clients & APIs

High-level API Security Overview

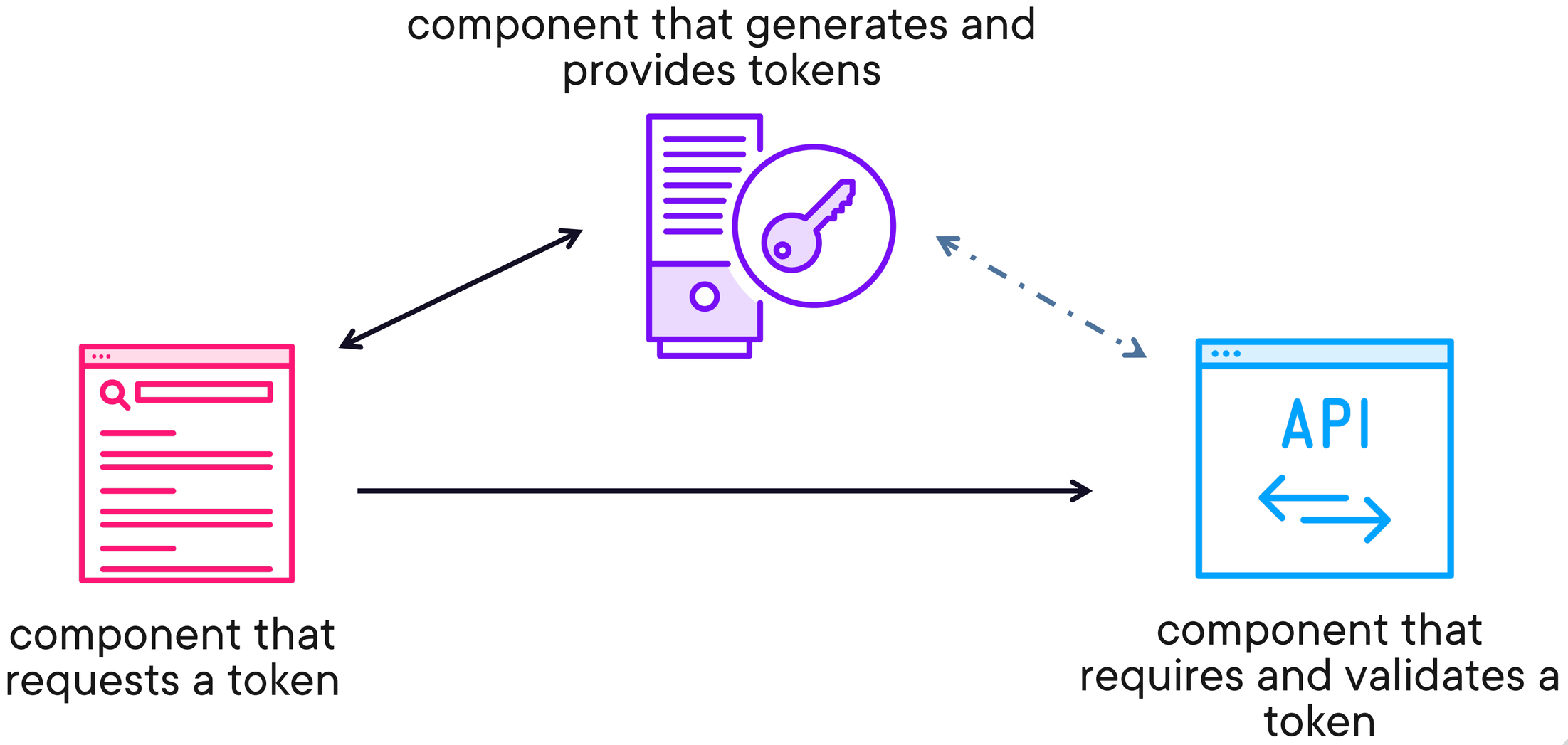
Application-level and/or infrastructure level?

On the same domain or cross-domain?

Local or centralized users & credentials?

Authentication and/or authorization?

High-level API Security Overview



```
{
  "sub": "kevins_unique_identifier",
  "given_name": "kevin",
  "role": "admin",
  "jti": "7a600c58",
  "aud": "menu-api",
  "nbf": 1678107105,
  "exp": 1686055905,
  "iat": 1678107106,
  "iss": "dotnet-user-jwts“
}
```

High-level API Security Overview

Tokens are often JWTs but don’t have to be

```
{
  "sub": "kevins_unique_identifier",
  "given_name": "kevin",
  "role": "admin",
  "jti": "7a600c58",
  "aud": "menu-api",
  "nbf": 1678107105,
  "exp": 1686055905,
  "iat": 1678107106,
  "iss": "dotnet-user-jwts“
}
```

High-level API Security Overview

Tokens are often JWTs but don’t have to be

```
{
  "sub": "kevins_unique_identifier",
  "given_name": "kevin",
  "role": "admin",
  "jti": "7a600c58",
  "aud": "menu-api",
  "nbf": 1678107105,
  "exp": 1686055905,
  "iat": 1678107106,
  "iss": "dotnet-user-jwts“
}
```

High-level API Security Overview

Tokens are often JWTs but don’t have to be

```
{
  "sub": "kevins_unique_identifier",
  "given_name": "kevin",
  "role": "admin",
  "jti": "7a600c58",
  "aud": "menu-api",
  "nbf": 1678107105,
  "exp": 1686055905,
  "iat": 1678107106,
  "iss": "dotnet-user-jwts“
}
```

High-level API Security Overview

Tokens are often JWTs but don’t have to be

```
{
  "sub": "kevins_unique_identifier",
  "given_name": "kevin",
  "role": "admin",
  "jti": "7a600c58",
  "aud": "menu-api",
  "nbf": 1678107105,
  "exp": 1686055905,
  "iat": 1678107106,
  "iss": "dotnet-user-jwts“
}
```

High-level API Security Overview

Tokens are often JWTs but don’t have to be

Our API validates a token.

Our API validates a token.
It does not generate it.

Our API validates a token.
It does not generate it.
It does not provide it.

Authentication

The process of determining whether someone or something is who or what it says it is

Token-based Security for Minimal APIs

Authentication: the token contains verifiable info on who or what is accessing it

Delegation: the token allows access on behalf of a user or application

```
builder.Services.AddAuthentication();
```

Token-based Security for Minimal APIs

Add and configure authentication services

```
builder.Services.AddAuthentication().AddJwtBearer();
```

Token-based Security for Minimal APIs

Add and configure authentication services
Tie authentication to the JWT bearer authentication handler

Authorization

The process of determining what someone or something is allowed to do

```
builder.Services.AddAuthorization();
```

Token-based Security for Minimal APIs

Add and configure authorization services

```
builder.Services.AddAuthorization();

...

var ingredientsEndpoints = endpointRouteBuilder
    .MapGroup("/dishes/{dishId:guid}/ingredients")
    .RequireAuthorization();
```

Token-based Security for Minimal APIs

Add and configure authorization services
Require authorization for a certain (group of) endpoint(s)

Demo

Requiring a bearer token

Generating a Token

Manually generate a token at level of your API

- `/login` endpoint
- Good for simple use cases, not a good approach for most scenarios

Generating a Token

OAuth2 and OpenID Connect are standardized protocols for token-based security

- "Token based security on steroids"

Centralized identity providers implement these & generate tokens

- Azure AD, IdentityServer, Auth0, ...

Our API validates a token.
It does not generate it.
It does not provide it.

Generating a Token

(ASP).NET Core includes a built-in tool to generate tokens which can be used during development

- `dotnet-user-jwts`

Demo

Generating a token with
`dotnet-user-jwts`

Demo

Creating and applying an authorization policy

Summary

An API does not generate or provide a token. Typically, it only validates the incoming token.

Summary

Authentication:

- `builder.Services`
  `.AddAuthentication()`
  `.AddJwtBearer()`

Summary

Authorization:

- `builder.Services.AddAuthorization()`
- `(endpoint(group)).RequireAuthorization()`

Up Next:

Documenting Your Minimal API