Are you sure your access tokens are really secure?

Wesley Cabus

Story time

Based on true events

- Security is hard, so the team used a well-known library to validate JWT access tokens.
- They used a trusted third-party company's issuer service to deliver securely signed tokens.
- Everything was configured correctly.
- Only tokens issued by the third-party service were allowed.
- And tokens needed to have a valid signature.

But then, disaster struck...

But then, disaster struck…

Are you sure your access tokens are really secure?

But then, disaster struck...

Hi, I'm Wesley

A coding architect at Xebia

Coffee enthusiast

Beer aficionado

What is a JWT?

Are you sure your access tokens are really secure?

What is a JWT?

Header

Payload

Signature

What is a JWT?

Header

- Token type
- Signature algorithm type
- Key identifier

Payload

Signature

What is a JWT?

Header

- Token type
- Signature algorithm type
- Key identifier

Payload

- Claims
- Subject
- Client ID
- Scopes
- Token validation parameters

Signature

What is a JWT?

Header

- Token type
- Signature algorithm type
- Key identifier

Payload

- Claims
- Subject
- Client ID
- Scopes
- Token validation parameters

Signature

- Cryptographically calculated byte array
- crypto_alg(header+"."+payload)
- Prevents tampering with the header and the payload

This is a JWT

eyJhbGciOiJFUzI1NiIsInR5cCI6ImF0K2p3dCIsImtpZCI6IjgxODcxOTgxOTIzODcxMjk4NzEyOSJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ildlc2xleSBDYWJ1cyIsImlzcyI6Imh0dHBzOi8vc3NvLmdvdHNoYXJwLmJlIiwic2NvcGUiOiJwcmVzZW50aW5nIiwiaWF0IjoxNzM3ODEyOTg1LCJleHAiOjE3NjkzNDg5ODMsIm5iZiI6MTczNzgxMjk4NSwianRpIjoiNzkyMzIxNjU3NjEyMjg5Nzc5Mjk5NTg0NDU4NDEzNTAzMjc3NDgifQ.aZ-Xw8oomAMFUiDXSM7NnqdFrsoIiZTIZjAigajpmMO3HUZJPa7V37taSmH851CUTYFIWTQZcpIZ3ZihpwcP2Q

This is a JWT

{ "alg": "ES256", "typ": "at+jwt“, "kid": "818719819238712987129" }

.

{ "sub": "1234567890", "name": "Wesley Cabus",
  "iss": "https://sso.gotsharp.be", "scope": "presenting",
  "iat": 1737812985, "exp": 1769348983, "nbf": 1737812985,
  "jti": "792321657612289779299584584135032774 8"
}

.

aZ-Xw8oomAMFUiDXSM7NnqdFrsoIiZTIZjAigajpmMO3HUZJPa7V37taSmH851CUTYFIWTQZcpIZ3ZihpwcP2Q

Back to our story

What was happening with those tokens?

Spot The Dot!

Let’s look at three tokens…

Token A

eyJhbGciOiJFUzI1NiIsInR5cCI6ImF0K2p3dCIsImtpZCI6IjgxODcxOTgxOTIzODcxMjk4NzEyOSJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ildlc2xleSBDYWJ1cyIsImlzcyI6Imh0dHBzOi8vc3NvLmdvdHNoYXJwLmJlIiwic2NvcGUiOiJwcmVzZW50aW5nIiwiaWF0IjoxNzM3ODEyOTg1LCJleHAiOjE3NjkzNDg5ODMsIm5iZiI6MTczNzgxMjk4NSwianRpIjoiNzkyMzIxNjU3NjEyMjg5Nzc5Mjk5NTg0NDU4NDEzNTAzMjc3NDgifQ.aZ-Xw8oomAMFUiDXSM7NnqdFrsoIiZTIZjAigajpmMO3HUZJPa7V37taSmH851CUTYFIWTQZcplZ3ZihpwcP2Q

Token B

eyAiYWxnIjogIm5PbkUiLCAidHlwIjogImF0K2p3dCIsICJraWQiOiAiODE4NzE5ODE5MjM4NzEyOTg3MTI5IiB9.eyAic3ViIjogIjMzMyIsICJuYW1lIjogIkV2aWwgUGVyc29uIiwgImlzcyI6ICJodHRwczovL3Nzby5nb3RzaGFycC5iZSIsICJzY29wZSI6ICJhZG1pbiIsICJpYXQiOiAiAxNzM3ODEyOTg1LCAiZXhwIjogMTc2OTM0ODk4MywgIm5iZiI6IDE3Mzc4MTI5ODUsICJqdGkiOiAiNTg0ODgxNjUwMzkwNzM0OTczNDA5NTcyODM2NTk4MjYifQ.

Token C

eyJhbGciOiJFUzI1NiIsInR5cCI6ImF0K2p3dCIsImtpZCI6IjgxODcxOTgxOTIzODcxMjk4NzEyOSJ9.eyJzdWIiOiIxIiwibmFtZSI6IkNodWNrIE5vcnJpcyIsImlzcyI6Imh0dHBzOi8vc3NvLmdvdHNoYXJwLmJlIiwic2NvcGUiOiJiZWluZyBhd2Vzb21lIiwiaWF0IjoxNzM3ODEyOTg1LCJleHAiOjE3NjkzNDg5ODMsIm5iZiI6MTczNzgxMjk4NSwianRpIjoiNzkyMzIxNjUwMzkwNzM0OTczNDA5NTcyODM2NTk4MjYifQ.AG5NQnGrcown3tllBB4Oqwnc7YfLYTgVf9V2F8ZzAGZgxjEclvcaX-yoK-TVtR07fL19gRiX70osrUAK6I1t6w

Let’s look at three tokens…

Token A

eyJhbGciOiJFUzI1NiIsInR5cCI6ImF0K2p3dCIsImtpZCI6IjgxODcxOTgxOTIzODcxMjk4NzEyOSJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ildlc2xleSBDYWJ1cyIsImlzcyI6Imh0dHBzOi8vc3NvLmdvdHNoYXJwLmJlIiwic2NvcGUiOiJwcmVzZW50aW5nIiwiaWF0IjoxNzM3ODEyOTg1LCJleHAiOjE3NjkzNDg5ODMsIm5iZiI6MTczNzgxMjk4NSwianRpIjoiNzkyMzIxNjU3NjEyMjg5Nzc5Mjk5NTg0NDU4NDEzNTAzMjc3NDgifQ.aZ-Xw8oomAMFUiDXSM7NnqdFrsoIiZTIZjAigajpmMO3HUZJPa7V37taSmH851CUTYFIWTQZcplZ3ZihpwcP2Q

Token B

eyAiYWxnIjogIm5PbkUiLCAidHlwIjogImF0K2p3dCIsICJraWQiOiAiODE4NzE5ODE5MjM4NzEyOTg3MTI5IiB9.eyAic3ViIjogIjMzMyIsICJuYW1lIjogIkV2aWwgUGVyc29uIiwgImlzcyI6ICJodHRwczovL3Nzby5nb3RzaGFycC5iZSIsICJzY29wZSI6ICJhZG1pbiIsICJpYXQiOiAiAxNzM3ODEyOTg1LCAiZXhwIjogMTc2OTM0ODk4MywgIm5iZiI6IDE3Mzc4MTI5ODUsICJqdGkiOiAiNTg0ODgxNjUwMzkwNzM0OTczNDA5NTcyODM2NTk4MjYilH0.

Token C

eyJhbGciOiJFUzI1NiIsInR5cCI6ImF0K2p3dCIsImtpZCI6IjgxODcxOTgxOTIzODcxMjk4NzEyOSJ9.eyJzdWIiOiIxIiwibmFtZSI6IkNodWNrIE5vcnJpcyIsImlzcyI6Imh0dHBzOi8vc3NvLmdvdHNoYXJwLmJlIiwic2NvcGUiOiJiZWluZyBhd2Vzb21lIiwiaWF0IjoxNzM3ODEyOTg1LCJleHAiOjE3NjkzNDg5ODMsIm5iZiI6MTczNzgxMjk4NSwianRpIjoiNzkyMzIxNjUwMzkwNzM0OTczNDA5NTcyODM2NTk4MjYifQ.AG5NQnGrcown3tllBB4Oqwnc7YfLYTgVf9V2F8ZzAGZgxjEclvcaX-yoK-TVtR07fL19gRiX70osrUAK6I1t6w

Let’s look at token B…

{ "alg": "nOnE", "typ": "at+jwt", "kid": "818719819238712987129"}

.

{ "sub": "333", "name": "Evil Person",
  "iss": "https://sso.gotsharp.be", "scope": "admin",
  "iat": 1737812985, "exp": 1769348983, "nbf": 1737812985,
  "jti": "584881650390734973409572836598 26"
}

.

alg: none

What?

- “alg”: “none” is a valid token.
- The validation library did indeed reject tokens using “none”!
- However…
- The library allowed tokens with “alg”: “NONE”, “alg”: “nOnE”, “alg”: “nONe”, …
- And the API was wide open for attack…
- Luckily, this was very quickly fixed!

But wait, there’s more!

Of course there is more, there’s always something :/

What's wrong with this token?

eyJhbGciOiJFUzI1NiIsInR5cCI6ImF0K2p3dCIsImp3ayI6eyJjcnYiOiJQLTI1NiIsImtpZCI6IkU0MTMwRkNDRDRFNEFBNUEiLCJrdHkiOiJFQyIsIngiOiJXVlZQMjJxUkNGdE9nTzJVWkpISDRjRllxYVk2QkJ3cG5QNUpzMWJObmprIiwieSI6ImZHUFlwYk5LSTFGQnd2V0E1UC1iaHI2WTktdDI1S0Q1Y0dLM19NQzZwQU0ifSwia2lkIjoiRTQxMzBGQ0NENEU0QUE1QSJ9

.

eyJzdWIiOiIzMzMiLCJuYW1lIjoiRXZpbCBQZXJzb24iLCJzY29wZSI6ImFkbWluIiwianRpIjoiOTdhMDVlM2Q1MDFkYjFkNzQ4NDY3Mzc2OTNlZGFhMzUiLCJuYmYiOjE3Mzg1MDY2NDcsImV4cCI6MTc3MDA0MjY0Nywia WF0IjoxNzM4NTA2NjQ3LCJpc3MiOiJodHRwczovL3Nzby5nb3RzaGFycC5iZSJ9

.

ykqjOQU60IQpu674N-JKgPZqqSAU9VzQx_4pSSjCMlO75Deh6xJvveGsc2LzL3CXUm24Mz5Sl8mDTEf-sqD_OQ

What's wrong with this token?

Header from latest token

eyJhbGciOiJFUzI1NiIsInR5cCI6ImF0K2p3dCIsImp3ayI6eyJjcn
YiOiJQLTI1NiIsImtpZCI6IkU0MTMwRkNDRDRFNEFBNUEiLCJr
dHkiOiJFQyIsIngiOiJXVlZQMjJxUkNGdE9nTzJVWkpISDRjRllxYVk
2QkJ3cG5QNUpzMWJObmprIiwieSI6ImZHUFlwYk5LSTFGQnd
2V0E1UC1iaHI2WTktdDI1S0Q1Y0dLM19NQzZwQU0ifSwia
2lkIjoiRTQxMzBGQ0NENEU0QUE1QSJ9

Header from known good token

eyJhbGciOiJFUzI1NiIsInR5cCI6ImF0K2p3dCIsImtpZCI6IjgxO
DcxOTgxOTIzODcxMjk4NzEyOSJ9

What's wrong with this token?


```
{
  "alg": "ES256", "typ": "at+jwt",
  "jwk": {
    "crv": "P-256",
    "kid": "E4130FCCD4E4AA5A",
    "kty": "EC",
    "x": "WVVP22qRCFtOgO2UZJHH4cFYqaY6BBwpnP5Js1bNnjk",
    "y": "fGPYpbNKI1FBwvWA5P-bhr6Y9-t25KD5cGK3_MC6pAM"
  },
  "kid": "E4130FCCD4E4AA5A“
}
```

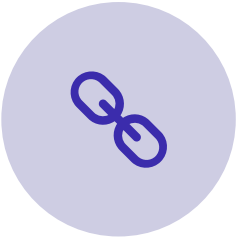
Self-signed tokens

Thank you RFC 7515

jwk: integrated JSON Web Key for verifying the signature

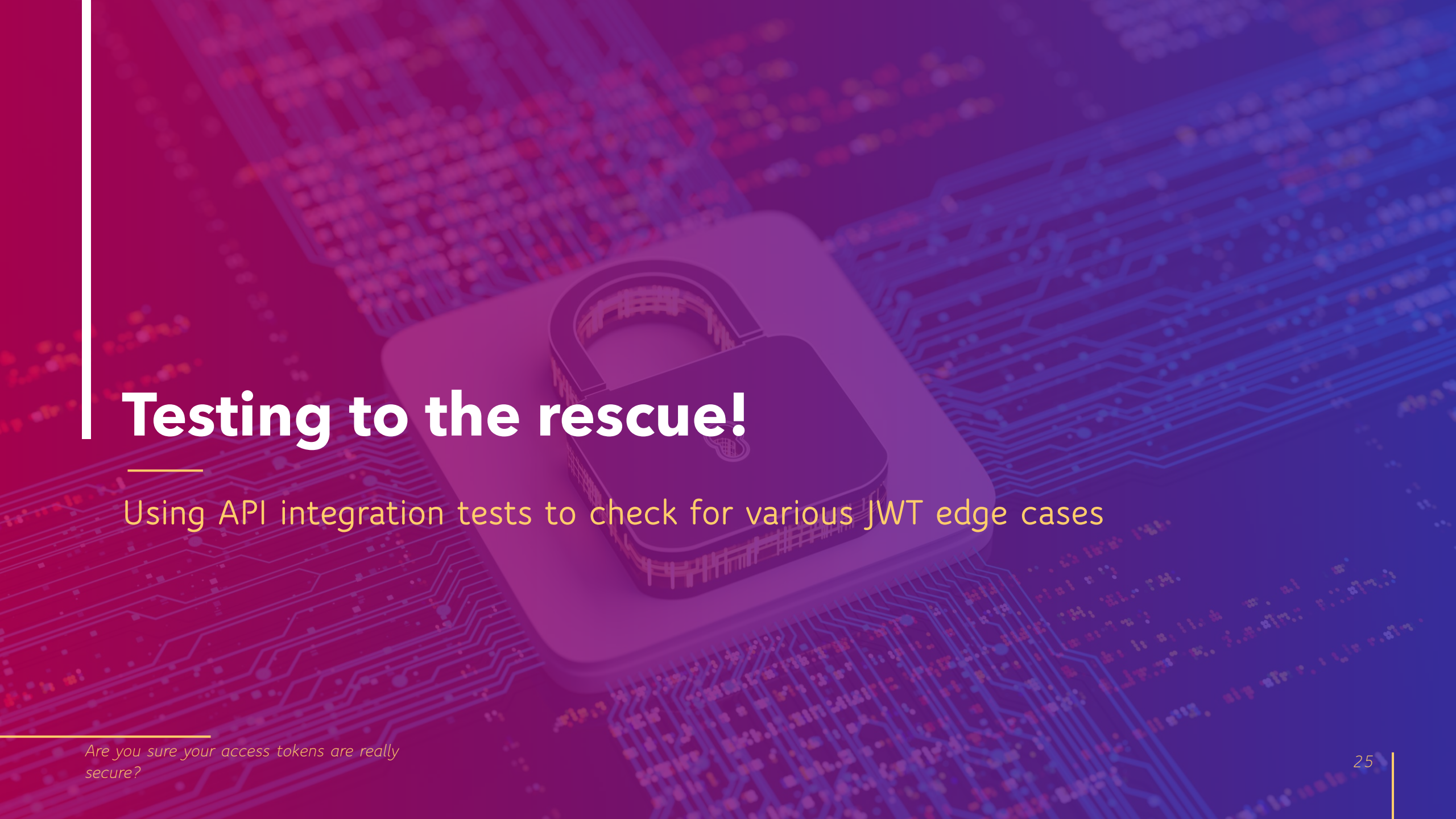
jku: URL pointing to a set of JSON Web Keys

x5c: integrated chain of certificates for verifying the signature



x5u: URL pointing to the certificate chain

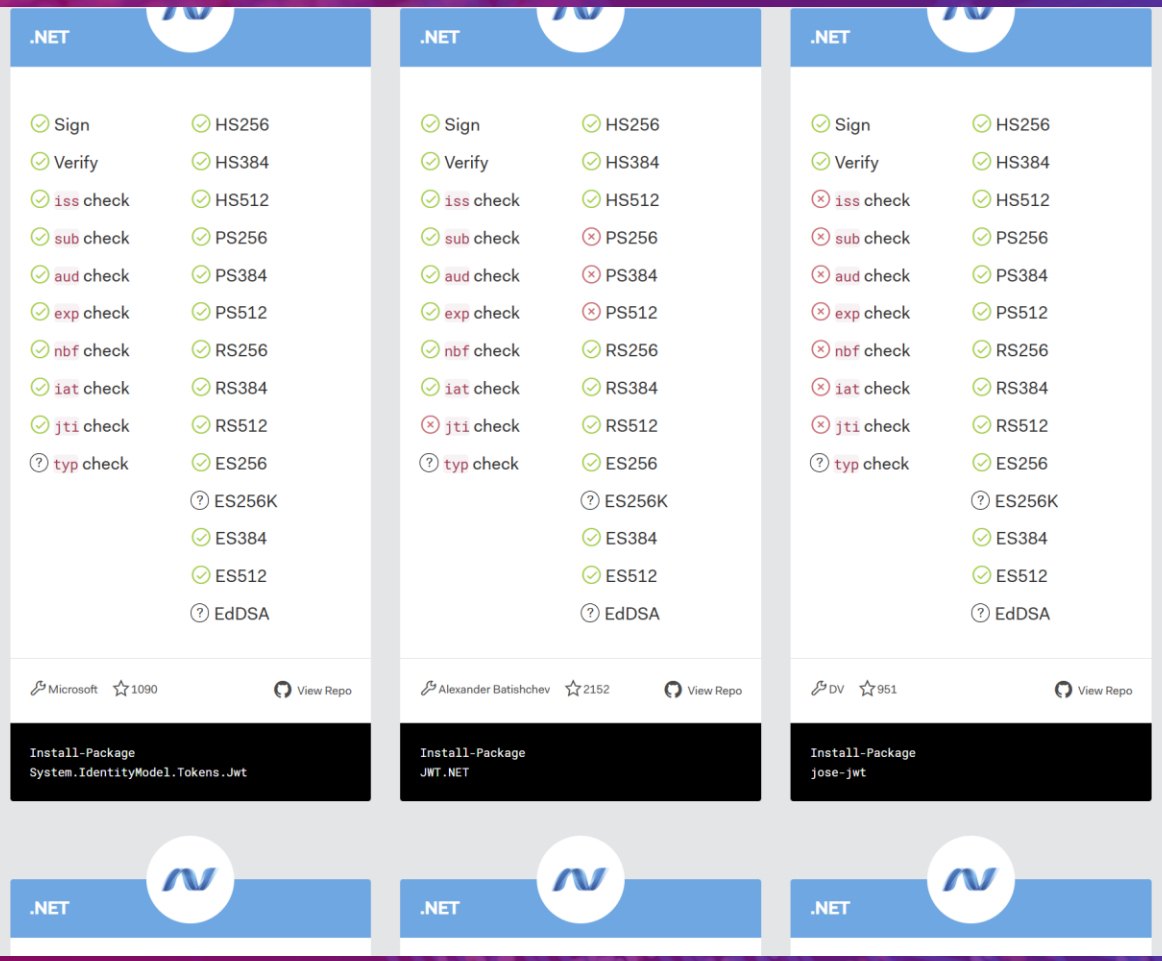
I'm wondering if anyone reads this. If you are, then: Hello you amazing person :)



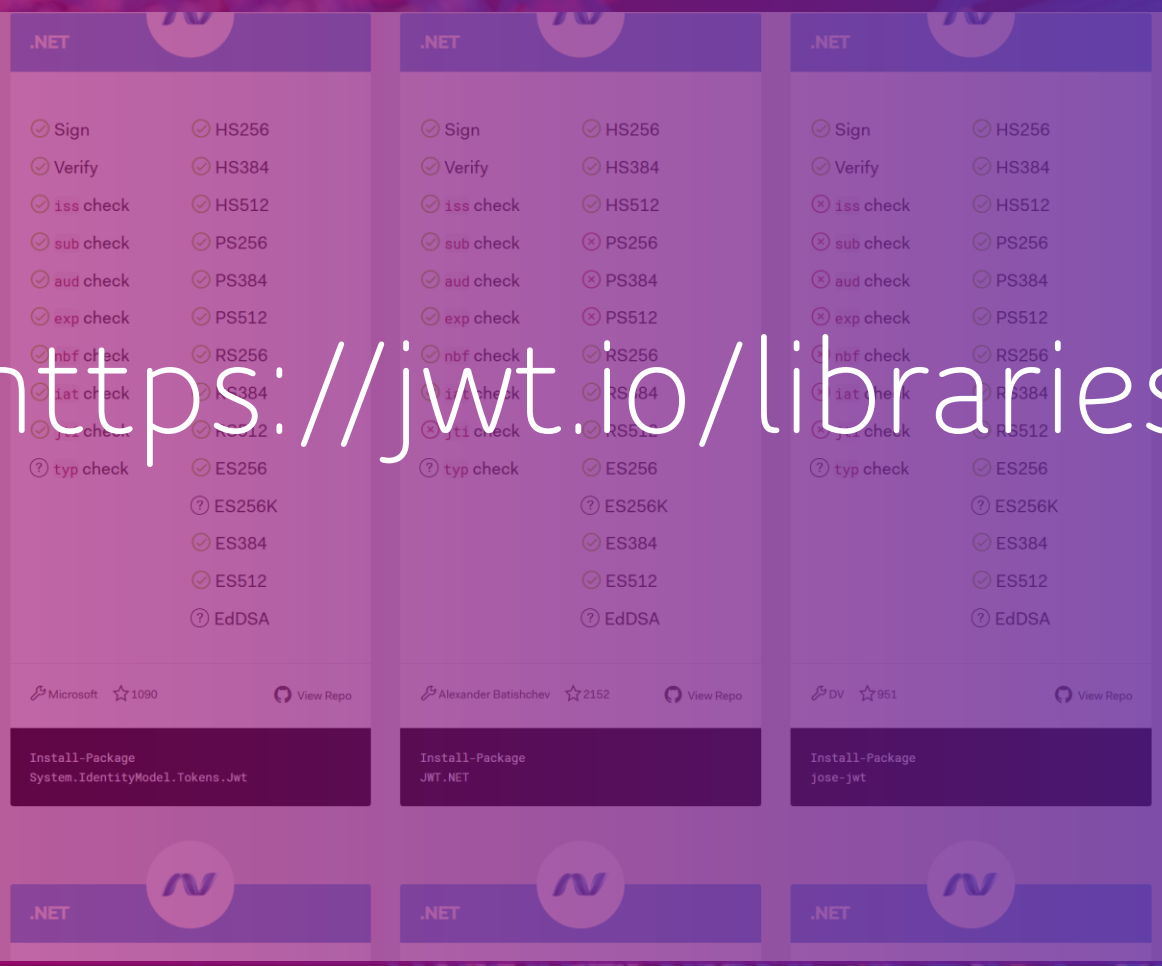
Testing to the rescue!

Using API integration tests to check for various JWT edge cases

Which JWT validation library do you use?



Which JWT validation library do you use?



https://jwt.io/libraries

Can you trust the library?

Or should you?

Open source

Vendor library

Updates

Package feed

Trust but verify

Are you sure your access tokens are really secure?

Demo time!

Adding integration tests to validate JWT access tokens

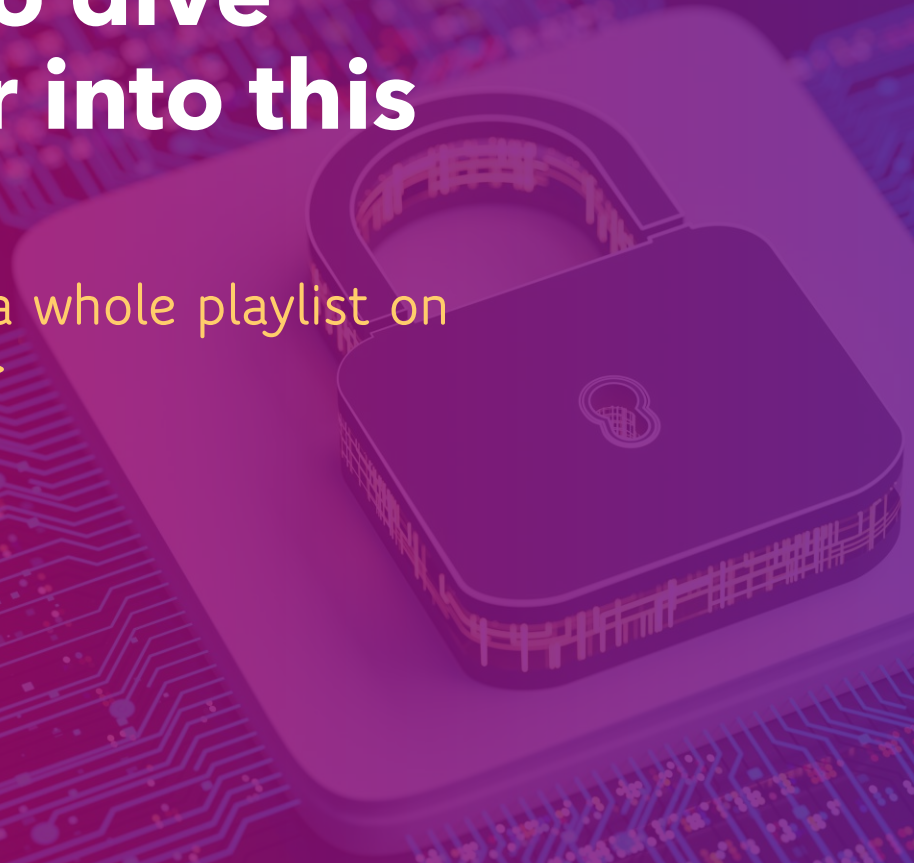
JWT Guard

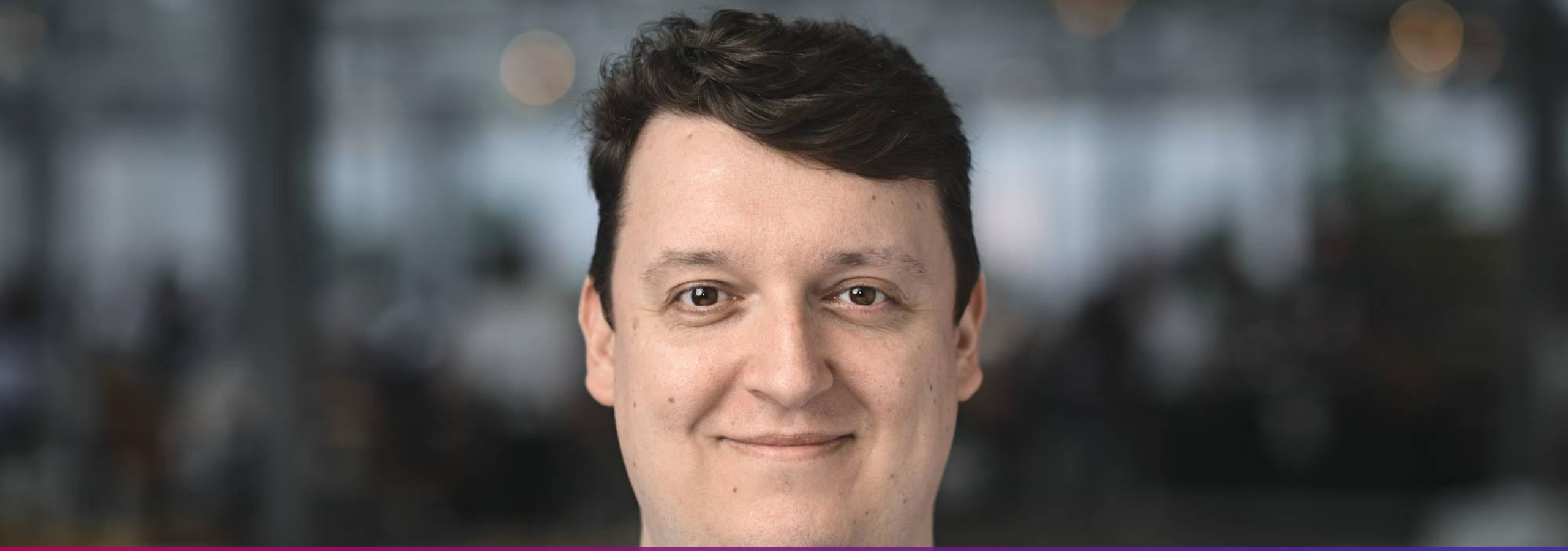
An open source integration test project for .NET APIs

https://jwtguard.net

Want to dive deeper into this stuff?

Intigriti has a whole playlist on this topic =>





THANK YOU!

Please evaluate the session ☺ =>