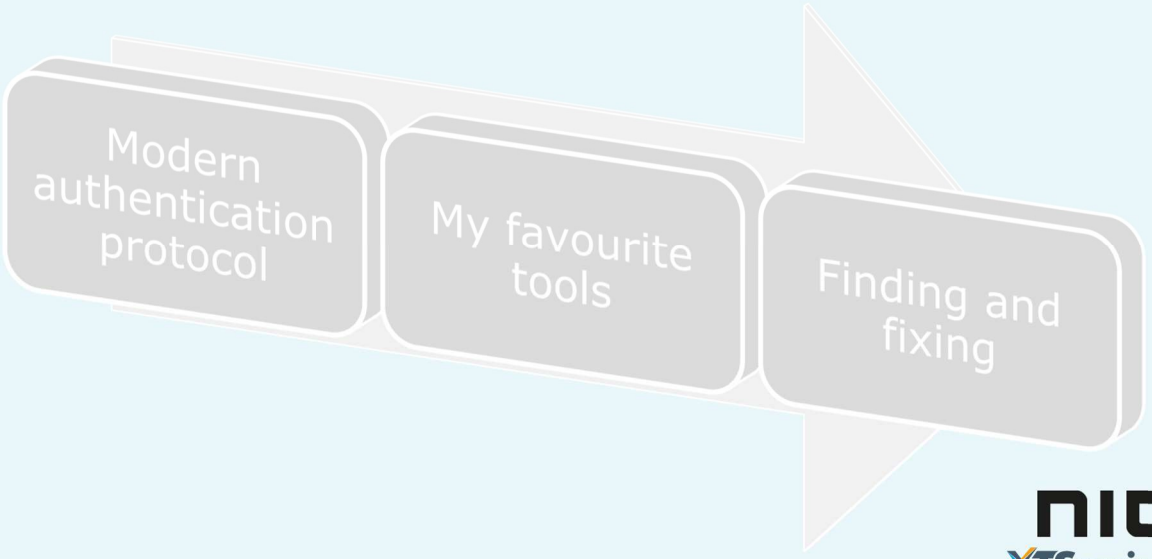
NIC Future Edition

Troubleshooting Modern Authentication Protocols OAuth 2.0 & OpenID Connect

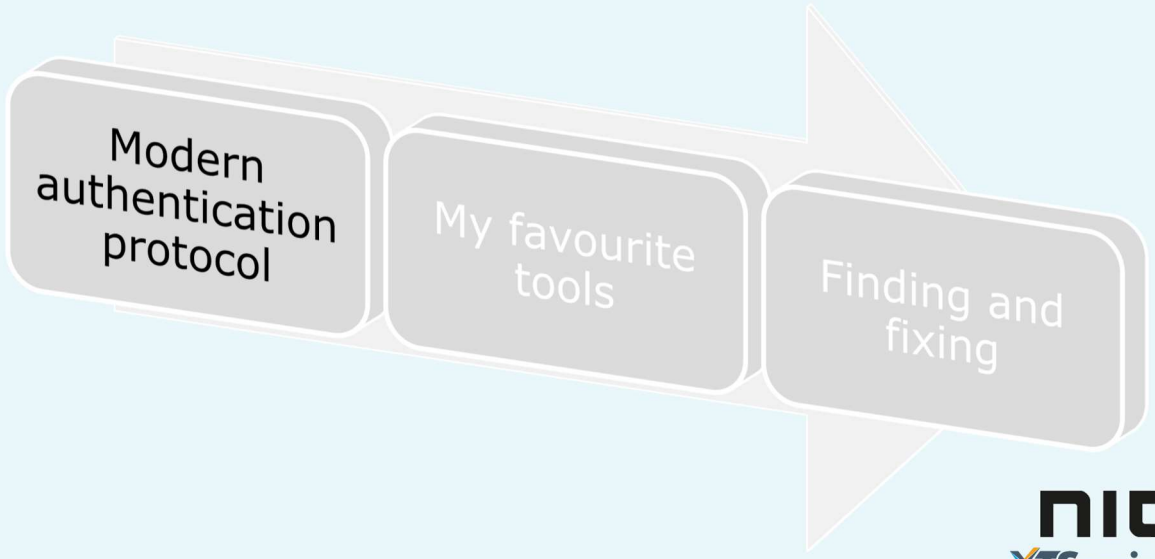
John Craddock
Identity and security architect, XTSeminars Ltd
johncra@xtseminars.co.uk @john_Craddock
XTSeminars

What's in this session



NIC XTSeminars

What's in this session



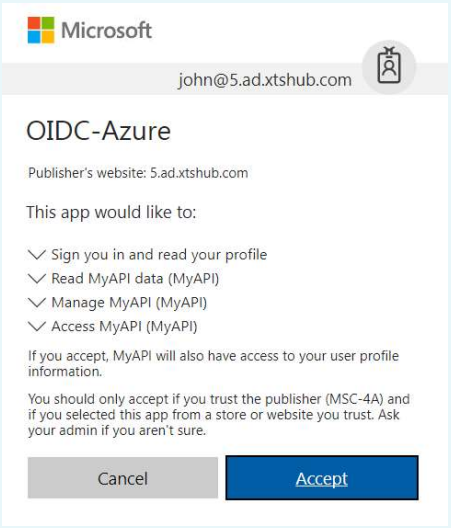
XTSeminars

Modern authentication protocols

- OpenID Connect is an authentication protocol
  - Authenticates users to an application or service
  - The identity of the user is transported in an id_token
- Oauth 2.0 is an authorization protocol
  - Provides a mechanism to authorize one application to access protected resources either with it's own identity or the identity of a user
    - If delegated access is configured, using the identity of a user, the user must "consent" to the access requested
      - Alternatively, consent can be given for all users by an administrator
  - The authorization data and identity are transported in an access_token
- OpenID Connect and OAuth2.0 supported by Azure AD and AD FS on Server 2016

XTSeminars

Asking for consent



- Azure AD allows for user consent via the displaying the details of the request to the user
- A new Azure AD pattern is being developed for incremental consent
  - Look out for the V2 endpoints
- AD FS does not support a consent framework and administrator consent is assumed

JSON Web Token (JWT)

Header

Body

Signature

eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Ik5HVEZ2ZEstZnl0aEV1THdqcHdBSk9NOW4tQSJ9.eyJhdWQiOiJodHRwczovL3h0c2VtaW5hcnMuY29tL0FQSS9NeXNhbXBsZSIsImlzcyI6Imh0dHBzOi8vc3RzLndpbmRvd3MubmV0L2FiMGE0NWM3LWMwODUtNGYzZi1hODY4LTYyZjM5Mjc5MjZlZi8iLCJpYXQiOjEzOTQ3MTE1MDAsIm5iZiI6MTM5NDcxMTUwMCwiZXhwIjoxMzk0NzE1NDAwLCJ2ZXIiOiIxLjAiLCJ0aWQiOiJhYjBhNDVjNy1jMDg1LTRmM2YtYTg2OC02MmYzOTI3OTI2ZWYiLCJvaWQiOiI1MGNlMDBmMC1lYjFhLTQ5MWQtODhiNi0zNjQ3NDZjMzg1NmQiLCJ1cG4iOiJqb2huQHh0c2VtaW5hcnMuY28udWsiLCJ1bmlxdWVfbmFtZSI6ImpvaG5AeHRzZW1pbmFyc3VjbyS1ayIsInN1YiI6IjF1SEp1OGtVZEVFcXZVaGloakl1QXNLRnV3OTlNTVpDREtISmtJVjZYdEkiLCJmYW1pbHlfbmFtZSI6IkNyYWRkb2NrIiwiZ2l2ZW5fbmFtZSI6Ikpvaa4iLCJhcHBpZCI6IjhlNzJjZmY4LWM1ZjMtNDE3Ni05ZjA0LTI0ZDBhNWIwNzE0YyIsImFwcGlkYWNyIjoiMCIsInNjcCI6InVzZXJfaW1wZXJzb25hdGlvbiIsImFjciI6IjEifQ.ZG2Kaag2tnqYWBPnoPggNgmVyh18zs1IXW-hGFMrQv0TOsF2Em5Qva0Bz2WUkeEWiw16XRo47nqgWgTtiaD0yzPDXD0_AZhnj5HbT6UmrMBEOd3tJ5Si3z6E5nbWB4KQZ4K7LZ_UhcMM3onD2uS0tyUkIzg-zj4OjGgZPzYUfIDa72AH6esy-adF7_HutB0zS6m35U97aUk6KOYs6F1-qTn63gAZ_G9_MNpm9yICKBU1vl2fktOwZMd7pQST8YDz9QYm9p5BF-dHP1_nK2KCX-1HWIbbtpSBEYInFYjjb_q310iG7BXI2HAbouf_YXDA1J5HDC1JcJIcA0xWVwkDQ

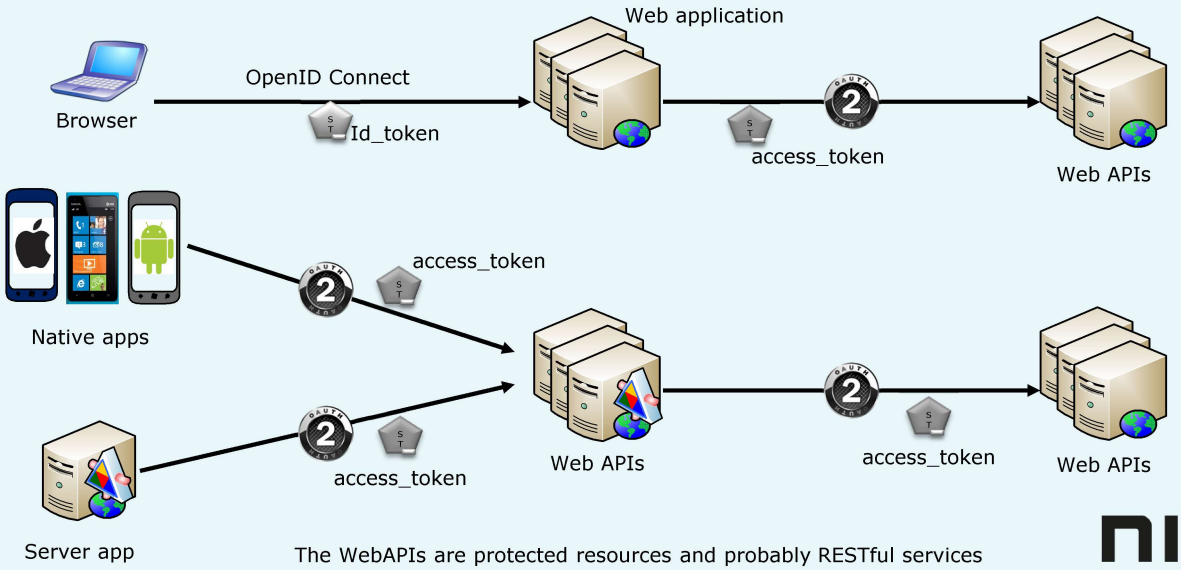

```
{
  "aud": "https://graph.windows.net",
  "iss": "https://sts.windows.net/ab0a45c7-c085-4f3f-a868-62f3927926ef/",
  "iat": 1394710750,
  "nbf": 1394710750,
  "exp": 1394714650,
  "ver": "1.0",
  "tid": "ab0a45c7-c085-4f3f-a868-62f3927926ef",
  "oid": "50ce00f0-eb1a-491d-88b6-364746c3856d",
  "upn": "john@xtseminars.co.uk",
  "unique_name": "john@xtseminars.co.uk",
  "sub": "i9Q7DOmfn6x2R6RNQu1U1KV12jg9rKm9a9JNp1jQSpo",
  "puid": "10033FFF855525C2",
  "family_name": "Craddock",
  "given_name": "John",
  "appid": "92aaae6f-cbd0-4935-93a0-03fdbd540a7c",
  "appidacr": "1",
  "scp": "62e90394-69f5-4237-9190-012177145e10",
  "acr": "1"
}
```


- JWT mandated for the id_token, often used for the access_token

Validating the token

- The website/API must validate the token before authorizing access
  - Audience (aud) claim
  - Trusted issuer (iss)
  - Time validity: nbf, iat & exp
  - Trusted signature
  - Required authorization claims

Modern authentication scenarios



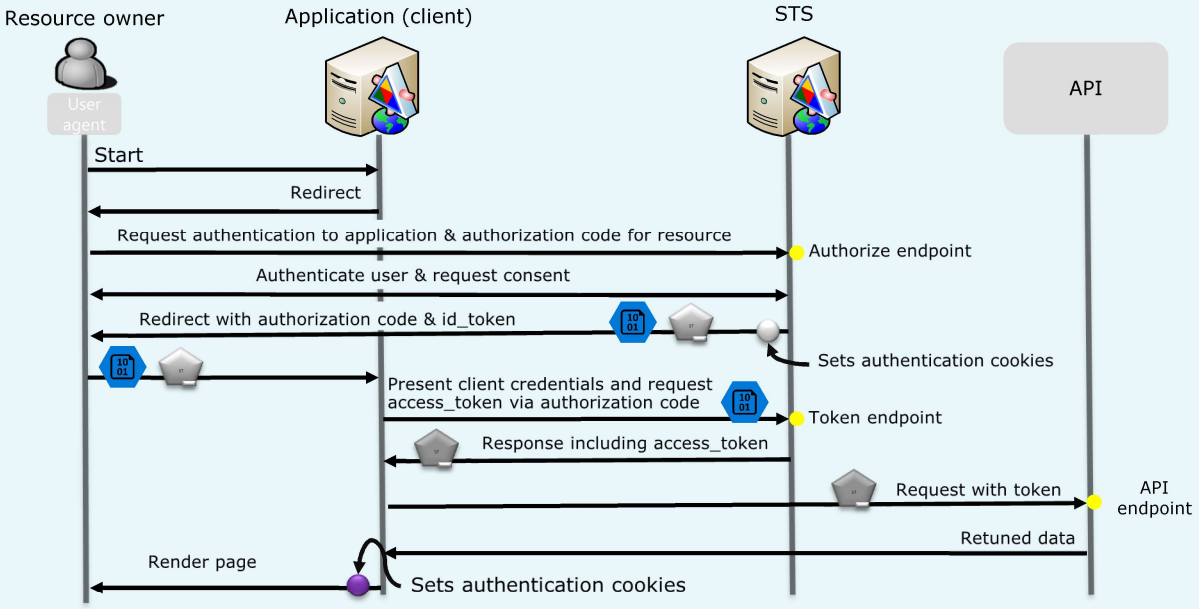
A word on terminology

- In the OAuth world the application (program) that requests the token(s) is called the client
- Microsoft refers to the “client” as “application” in Azure AD
  - AD FS uses “client”
- When tracing the protocol you will see client_id
- Different authentication flows can be configured to meet the needs of the different supported client types
  - In OAuth 2.0 terminology these are referred to as Grants

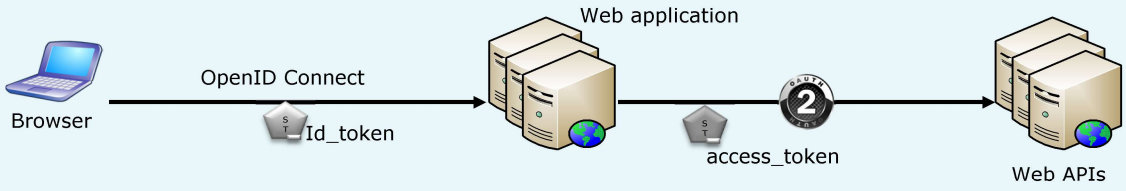
The OAuth 2.0 Grants (flows)

- Web application client (Authorization Code Grant)
  - Server side web application
- Native / mobile client (Implicit Grant)
  - Code runs locally
  - Can also use Authorization Code Grant
- Trusted client (Resource Owner Password Credential Grant)
  - Code runs locally on “trusted software”
- Client to server communication authenticating with the client credentials (client credential Grant)

Hybrid-flow - one of many flows

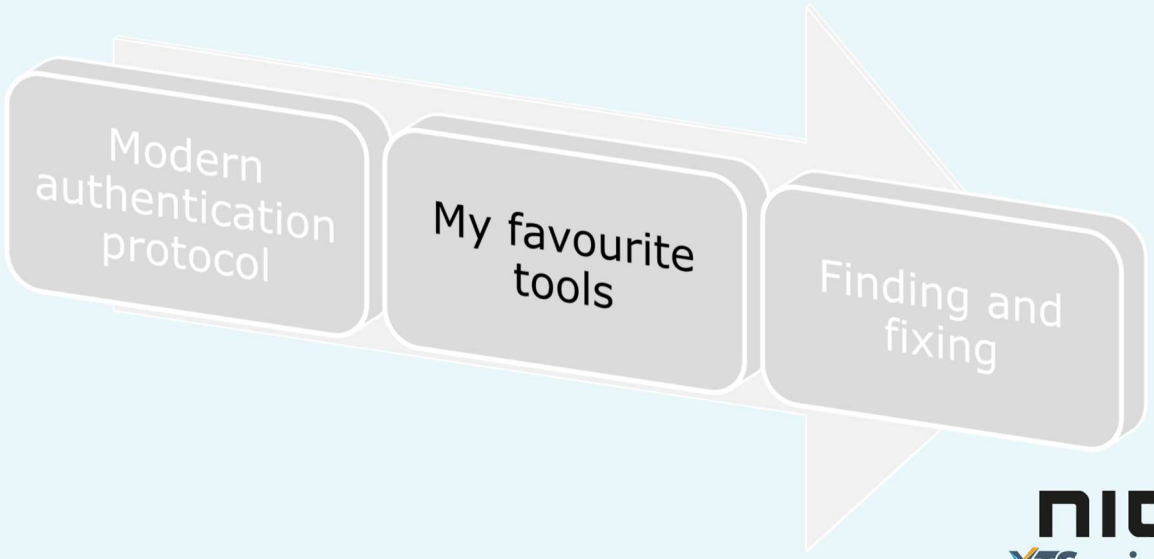


Demo environment



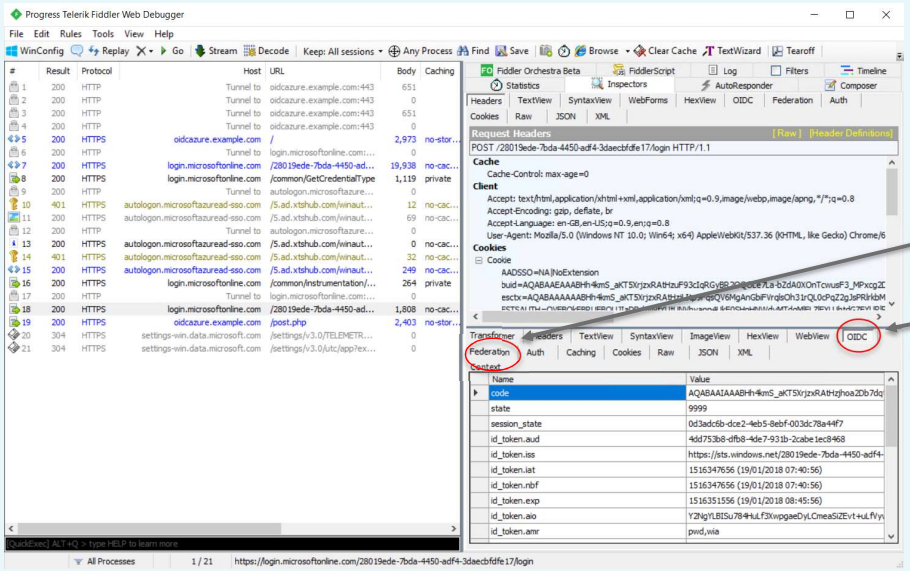
Security token service (STS)
AD FS or Azure AD

What's in this session



XTSeminars

Fiddler



Adding Inspectors

http://identitymodel.codeplex.com/releases/view/52187

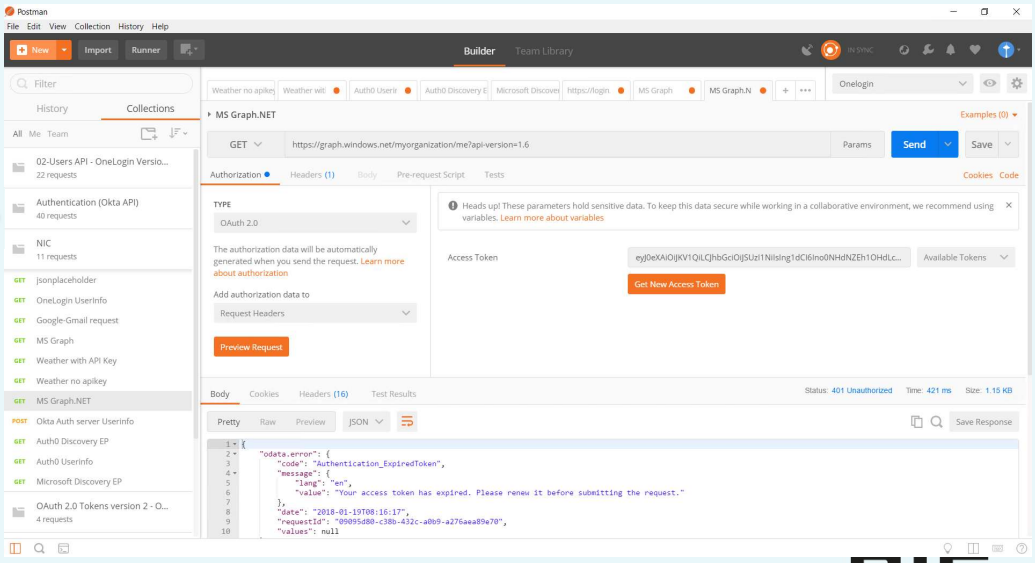
https://github.com/vibronet/OInspector/tree/dev

- www.telerik.com

XTSeminars

Postman

Many vendors supply Postman collections for testing their API

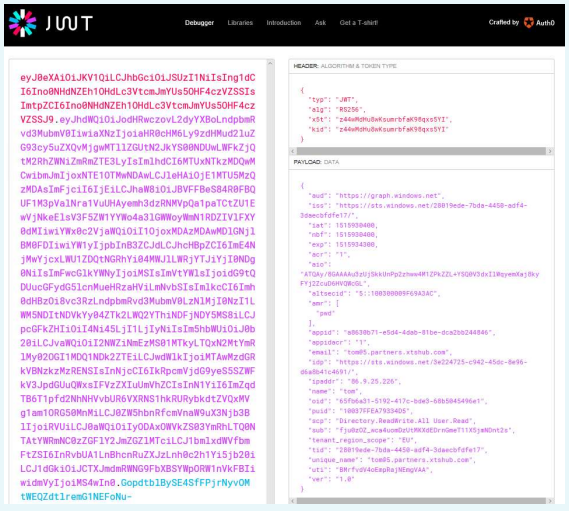


- www.getpostman.com

Examining tokens

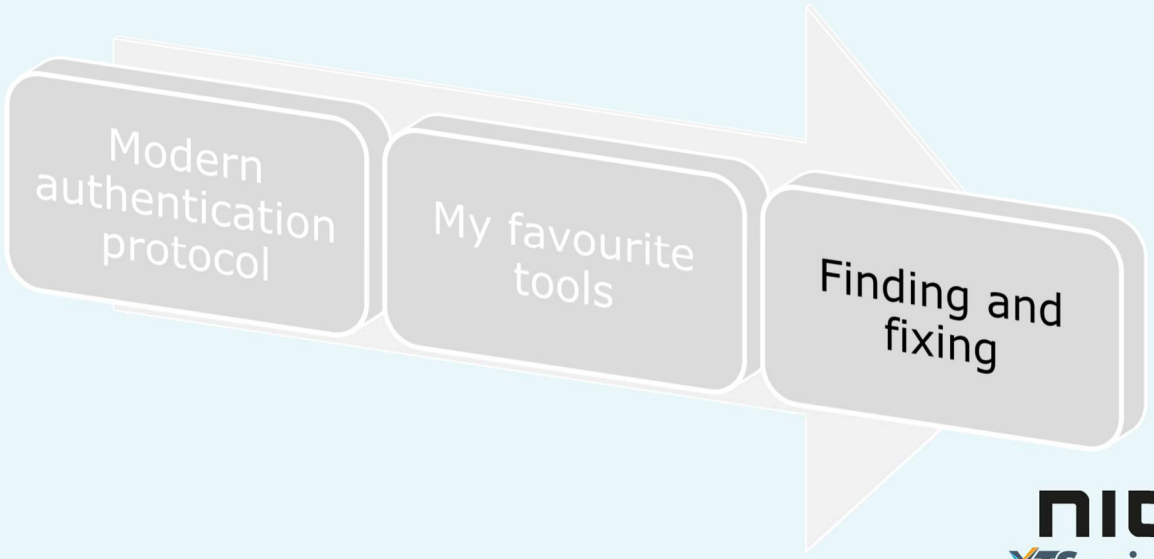
jwt.io

https://gallery.technet.microsoft.com/JWT-Token-Decode-637cf001

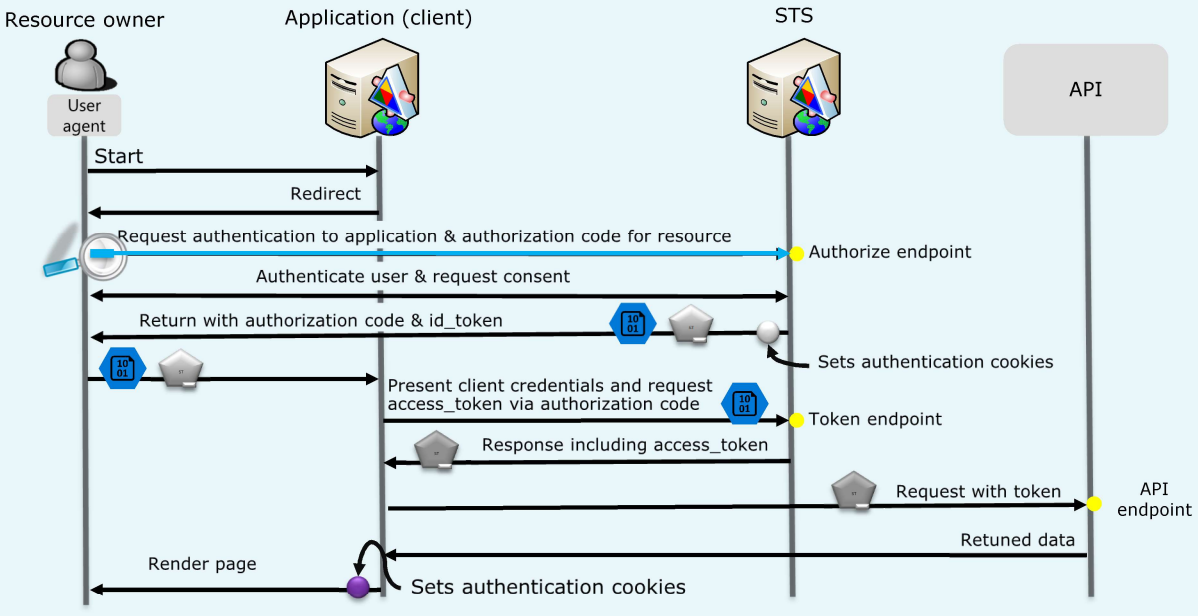


- Security sensitive tokens should not be decoded online

What's in this session



1st Request to the STS



First redirect

Desired grant type

How the response should be returned

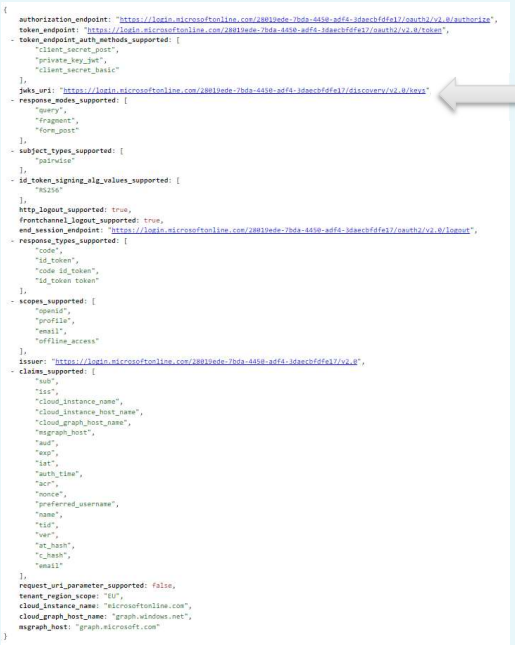
https://login.windows.net/ab0a45c7-c085-4f3f-a868-62f3927926ef/oauth2/authorize?
client_id=d4c6ff7a-cf78-4a42-b617-c576d6f3f53b&
redirect_uri=https://openidconnect.xts.local/&
response_mode=form_post&
response_type=code id_token&
scope=openid profile&
state=OpenIdConnect.AuthenticationProperties=ji5hRvSmJWoYKZITm2FL345fhovBXNHQBkHwIOKhOpkaKoIG1q7Wh2aOPhdH10gSdnrFrg_clJJdu8GKnrSKJcGGNq_BknH0aFgapuuuKp1vQG7mOe69M1kbDwKLiDkNp8qRmnj9MwgJS6STTg1jTwPLreXY3eSyhOlyBngfk8qh_O_sbXpF4RksaKap0SAW&
nonce=635953791254138969.NDc3Yzc5YmUtNDhmNi00YzIxLWE2OGEtYWVlOGIzMjk4NDJmNTNhNTkyOTQtOTk3ZC00ZDRkLTlkMGUtZTdlZjM4Yzc5NWY5

Access privileges being requested

Replay & cross-site forgery prevention. Returned in id_token

Returned unchanged

Endpoints and more…



Location of public key(s) for token signature validation

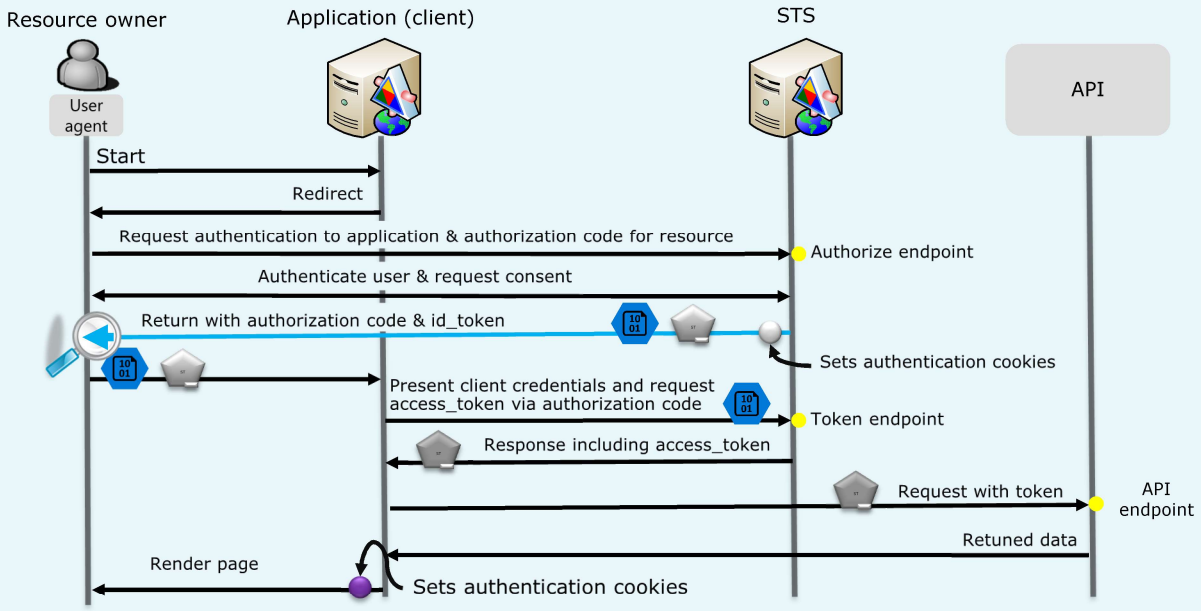
Definitive source
Azure:
https://login.microsoftonline.com/<tenant name>/v2.0/.well-known/openid-configuration

AD FS
https://<adfs service name>/adfs/.well-known/openid-configuration

Demo…

1st Redirect

1st response after authentication

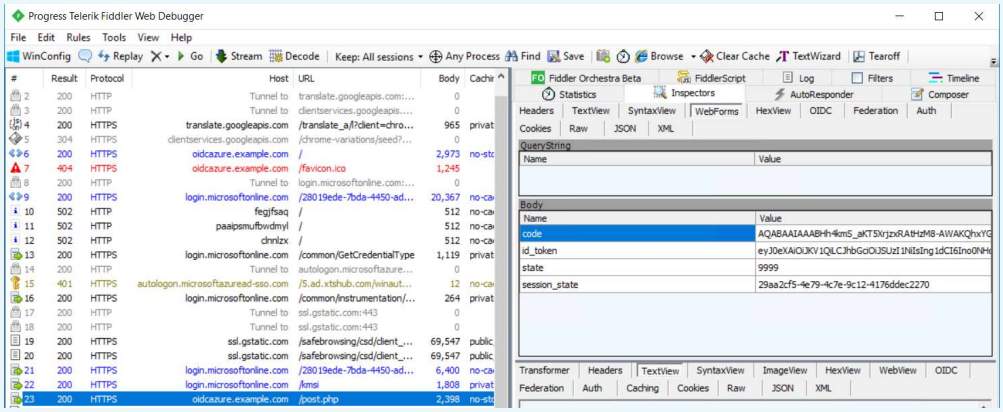


200 OK

```
<html><head><title>Working...</title></head><body>
<form method="POST" name="hiddenform" action="https://oidcazure.example.com/post.php">
<input type="hidden" name="code" value="AQAB…" />
<input type="hidden" name="id_token" value="eyJ0eX…" />
<input type="hidden" name="state" value="9999" />
<input type="hidden" name="session_state"
      value="5151341e-f9bb-4f70-b561-278794a0c761"/>
<noscript><p>Script is disabled. Click Submit to continue.</p>
<input type="submit" value="Submit" /></noscript></form>
<script language="javascript">document.forms[0].submit();</script></body></html>
```

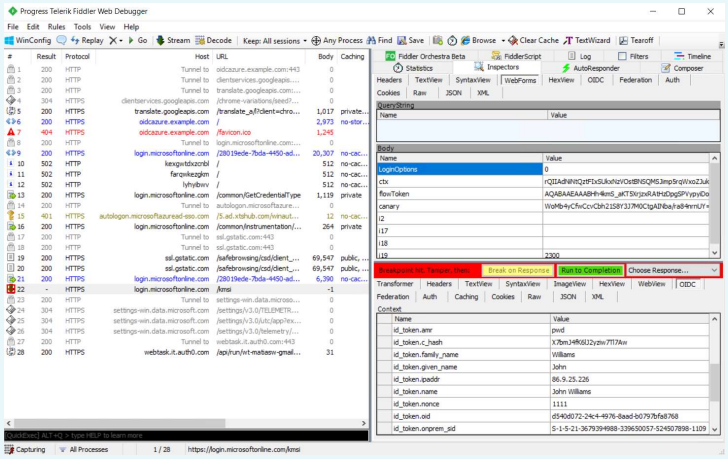
- The 1st request to the STS defines the response_mode
  - In this example it was set to form_post
  - Alternatives query and fragment

Subsequent POST



- If you are not getting back what you expect, check the response_type and scope values in the 1st redirect to the STS

200 OK



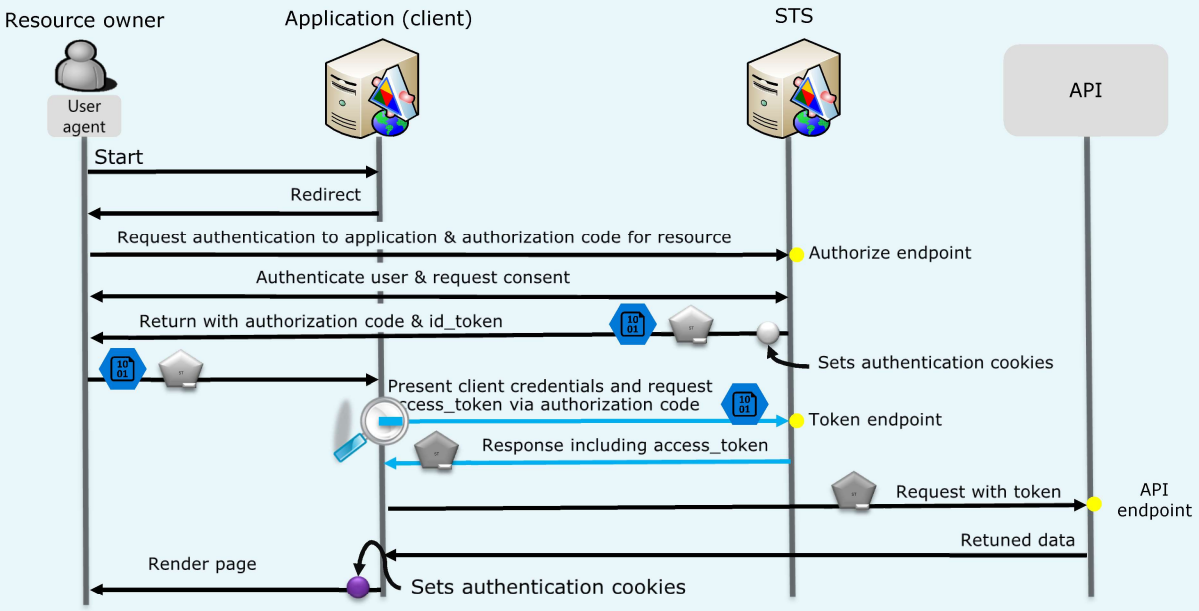
- Using Fiddler you can set a breakpoint so that you can fiddle with the response before sending it back to the application
  - Allows testing that token is being properly validated by the app
- Fiddler can be used to replay tokens and evaluate if the app is correctly handling replay detection

XTSeminars

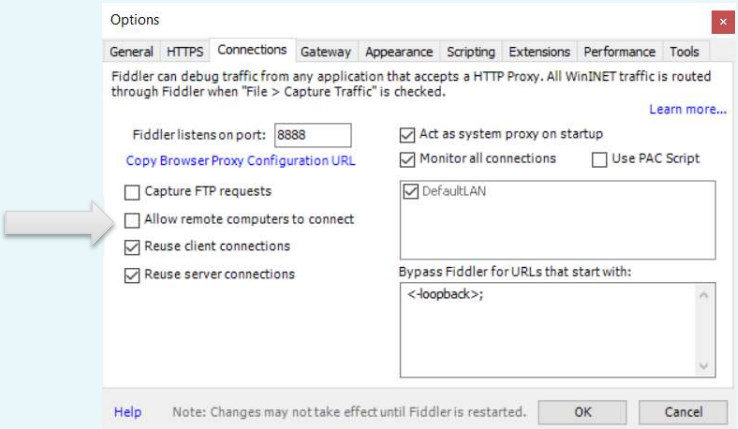
Demo…

Breakpoints

Code to token exchange

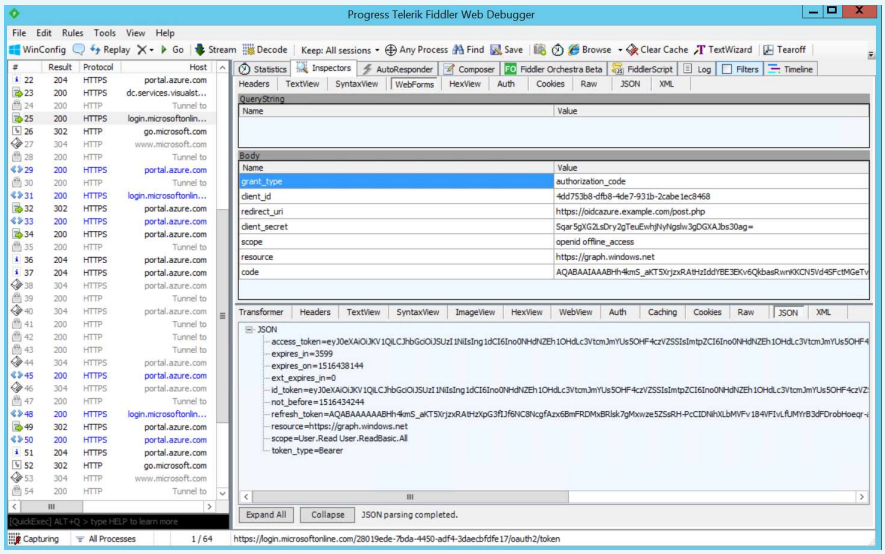


Top tip!



- Set up the application to use Fiddler as a proxy and capture the code to token exchange
- If you can't run Fiddler on the app server, enable Fiddler to support the connection of remote computers
  - Also allows tracing a browser interaction from a mobile device

Code to token exchange

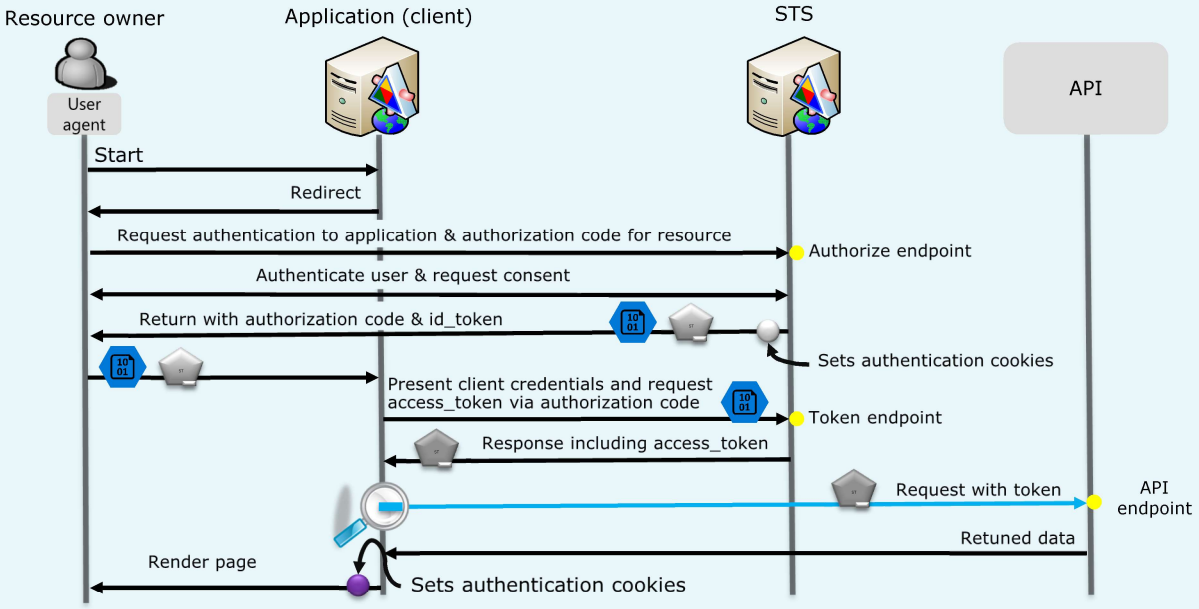


- Capture from web application

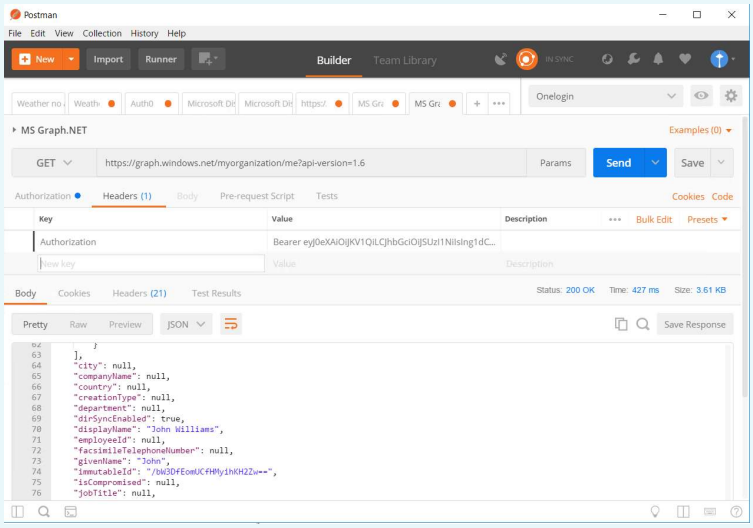
Demo…

Code to token exchange

Accessing the API



Use Postman to test interaction with the API



- Use the access_token returned to interact with the API

Demo…

Postman

The documentation nightmare!

- OAuth 2.0 is defined as a framework and there are many different interpretations as to exactly how it should be implemented
- Always look to the documentation from the appropriate IDaaS vendor when working with OAuth 2.0
- Always check that you a reading OAuth 2.0 documentation
  - OAuth is completely different
- It's easy for developers to leave security holes through inappropriate implementation
  - Always check token validation, replay detection and cross-site forgery detection

So where next?

5-Day Hands-On Masterclass with John Craddock

Microsoft Identity solutions with Azure Active Directory, on-premises AD FS and AD

38 hands-on labs

10% Discount if you book by this Friday for:
Oslo : April 23 – 27 or
Oslo : August 27 - 31

Talk to the NIC/Crayon team or email info@niccomf.com

http://www.xtseminars.co.uk/masterclass

NIC XTSeminars

Consulting services on request

John Craddock
Infrastructure and security Architect
XTSeminars Ltd

Johncra@xtseminars.co.uk
@john_craddock

John has designed and implemented computing systems ranging from high-speed industrial controllers through to distributed IT systems with a focus on security and high-availability. A key player in many IT projects for industry leaders including Microsoft, the UK Government and multi-nationals that require optimized IT systems. Developed technical training courses that have been published worldwide, co-authored a highly successful book on Microsoft Active Directory Internals, presents regularly at major international conferences including TechEd, IT Forum and European summits. John can be engaged as a consultant or booked for speaking engagements through XTSeminars. www.xtseminars.co.uk

NIC XTSeminars

Resources

Slides and demos from the conference will be available at github.com/nordicinfrastructureconference/2018 (bit.ly/2y7JhA3)