Integrating with Active Directory, Azure Active Directory and Social Logins

Kevin Dockx
Architect

@KevinDockx https://www.kevindockx.com

Coming Up

Handling integration with 3rd party providers

Scenarios that deal with remote credentials

- Windows Authentication (Active Directory)
- Azure Active Directory
- Social login providers like Facebook, Google, …

Handling Integration with Third-party Providers

One user may have accounts with credentials in various places:

- Locally
- Windows credentials
- Azure AD credentials
- Social credentials
- …

We want to enable a user to use these credentials to authenticate

Handling Integration with Third-party Providers

Client app

One User or Many Users?

Kevin, locally
Identified by subject claim: 12345
Claim type given_name: “Kevin”
Claim type role: “FreeUser”

Kevin on Facebook
Identified by userId claim: 67890
Claim type firstname: “Kevin”
No role claim...

Handling Integration with Third-party Providers

Handling Integration with Third-party Providers

This module:

- Different integrations (AD, AAD, Facebook)

Next module:

- Account linking, claims transformation

Use Cases for Windows Authentication

Use Active Directory domain identities / Windows accounts to identify users

- Negotiate, Kerberos, NTLM
- IIS, Kestrel, HTTP.sys

Best suited for intranet environments

Windows Authentication Beneath the Covers

The web server takes care of handling this type of authentication

- Configured at level of that web server

Windows Authentication Beneath the Covers

Negotiation: client sends a login request to web server

Challenge: web server answers with a challenge (= random token)

Client generates and hashes a response, and sends it to the web server

Web server compares challenge-hashed response to expected response

Windows Authentication Beneath the Covers

NTLM is a Microsoft proprietary authentication protocol, Kerberos (developed at MIT) is an alternative

- Both can be used for Windows authentication
- NTLM is the default fallback protocol, Kerberos is the preferred protocol

Demo

Enabling Windows authentication on IIS Express

Demo

Integrating Windows authentication with IdentityServer

Federation with Third-party Identity Providers

Other credentials sets:

- Enterprise/intranet: Azure AD
- Social: Facebook, Google, Twitter, Microsoft, …

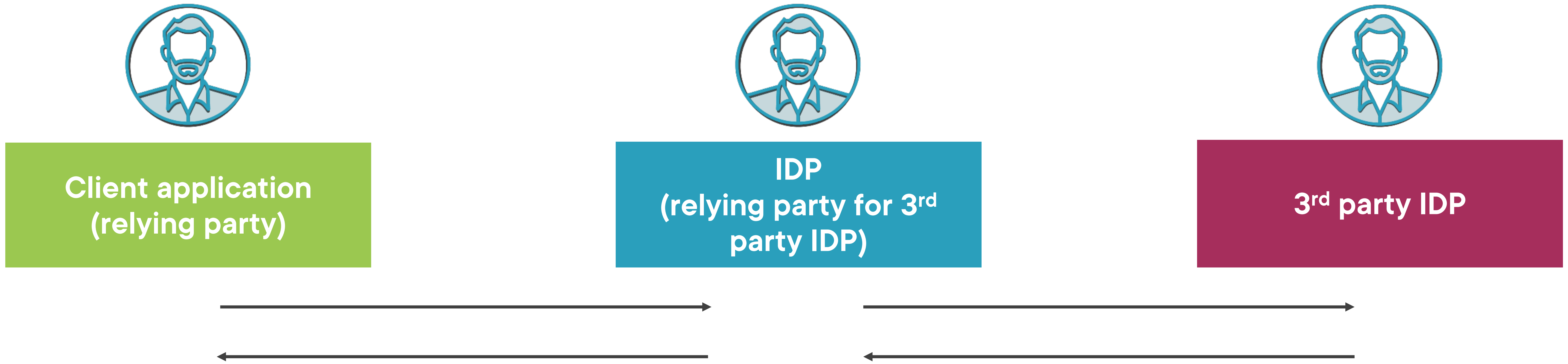
Reusing those is convenient for the user, and it shifts a lot of the IAM complexities to a third party IDP

- Federated authentication / basic form of federated identity

Federation with Third-party Identity Providers

Client application
(relying party)
IDP

Federation with Third-party Identity Providers


Client application (relying party)
IDP (relying party for 3rd party IDP)
3rd party IDP

Federation with Third-party Identity Providers

The protocol used by the third-party provider can vary

- OpenID Connect, SAML, proprietary protocol, …

Demo

Inspecting support for federating with a third-party identity provider

Demo

Registering an application on Azure AD

Demo

Integrating with Azure AD

Demo

Registering an application on Facebook

Demo

Integrating with Facebook

Challenges When Integrating with Third-party Identity Providers

You are placing a lot of trust in an identity provider that's out of your control

- Security issues at level of the 3rd party IDP are also **your** issues

Challenges When Integrating with Third-party Identity Providers

Not all IDPs are created equal

- It's up to the IDP to decide what is supported

For example: not all providers allow (federated) sign-out

- As long as users are signed in to the 3rd party provider, they can sign in to clients relying on our IDP without providing credentials

Integrating with Other Third-party Identity Providers

Integrate with any OIDC-supporting provider by using Microsoft’s default OIDC middleware

- ADFS, Azure AD, Okta/Auth0, Ping, TrustBuilder, WSO2 Identity Server, …

Summary

Most of us already have a set of credentials somewhere; reusing those

- … is convenient for the user
- … shifts a lot of IAM complexities to a third-party IDP

Keep in mind that this means you're adding the external IDP to your trust domain!

Summary

Integration comes with complexities in regards to linking users, claims transformation, …

- Handle it at level of the IDP

Summary

Windows authentication is best suited for intranet environments

- Enable it at level of the web server

Summary

When authenticated at level of a third-party provider, it can provide proof of authentication to our IDP

- That proof is used to authenticate at level of our IDP…
- … and that allows our IDP to provide proof of authentication (an identity token) to our client app

Up Next:

User Provisioning, Federation and Federated Identity