

# Open Source Zero Trust

OSS @ Siemens China | 30.11.2025

Felix Moessbauer (Siemens AG, Foundation Technologies)

## About me



Felix Moessbauer

- Senior key expert @Siemens FT
- Embedded Linux consultant & developer
- Contributes to major OSS projects for Siemens
- Tooling developer (static / dyn. analysis, build tools, ...)



@fmoessbauer



mastodon.social /  
@fmoessbauer



felix.moessbauer@siemens.com

# The Basics

## Zero Trust and OpenID Connect

## What is Zero Trust?

- Security Framework based on the principle of “Never trust, always verify”
- Not a product you can buy!

### Core Principles

- Do not trust anyone / anything
- Minimize the granted privileges
- Continuously evaluate the permissions
- Authenticate each resource individually
  
- Isolate the network into zones (*not in scope of this talk*)

## Zero Trust, OAuth2 and OpenID Connect (OIDC)

### Login Provider

- List of applications
  - Claims
  - Allowed groups
- List of users
- List of devices
- List of groups

The login provider usually provides a self-service portal to setup new applications

### Application

- Linked to login provider
- Does not need to know about access permissions (simplified view)
- Does not need to know users

### User

That's you.



**Policy Enforcement Point (PEP):** Decide if a user is granted access, possibly by requiring further authentication factors (like MFA). PEP is done in the Login Provider

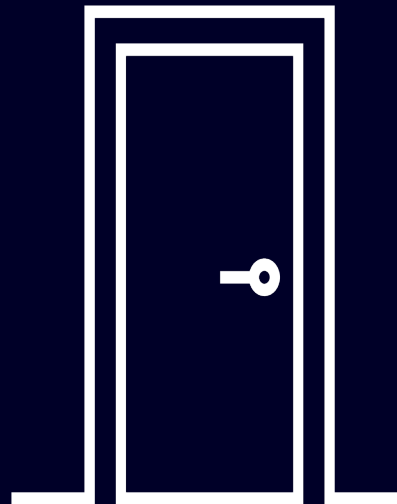
## Conditional Access Policies (1)

### Vanilla OIDC

- I'm Felix. Here is my ID card (e.g. PKI certificate or username / password)
- I want to access service "foo", with the following claims (read DB, write logs)



Login Provider




- Is Felix ID valid?
- Is Felix allowed to access "foo" with these claims?



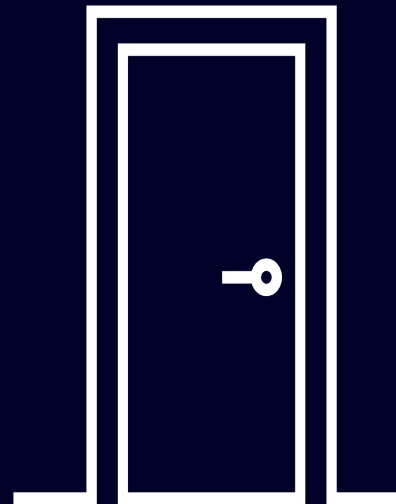
## Conditional Access Policies (2)

### Extended OIDC

- I'm Felix. Here is my ID card (e.g. PKI certificate or username / password)
- I want to access service "foo", with the following claims (read DB, write logs)
- **My device is this one:** 



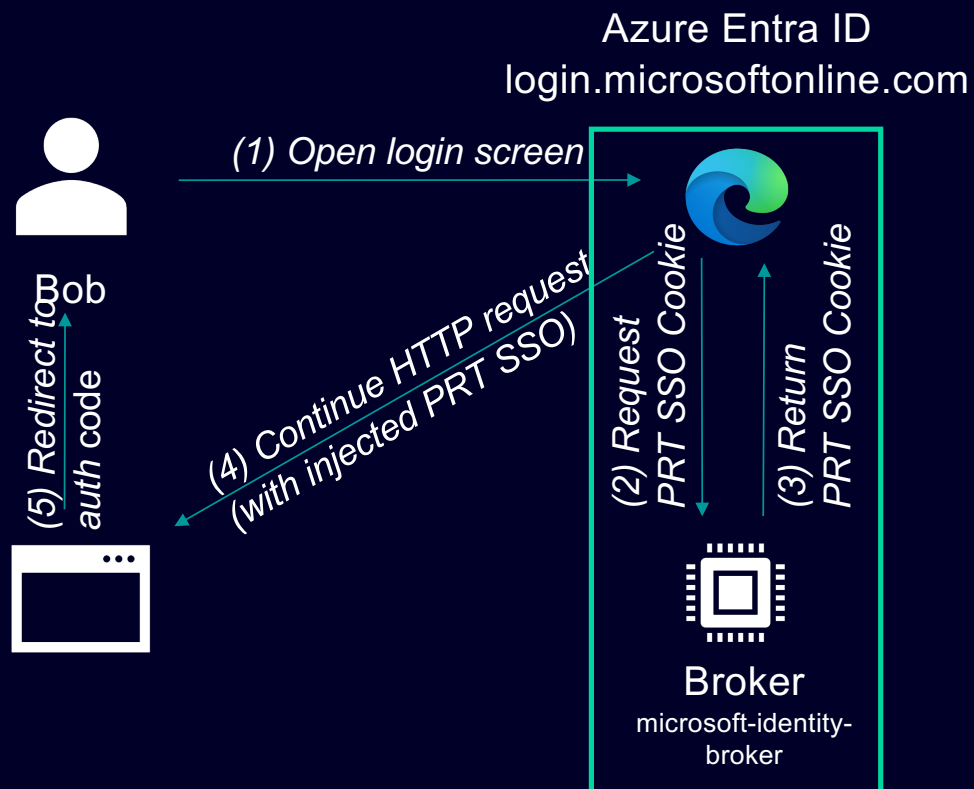
Login Provider



- Is Felix ID valid?
- Is Felix allowed to access "foo" with these claims?
- **Is Felix accessing from a listed device?**
- **Is that device compliant? Check against device DB.**



## [MS-OAPXBC]: OAuth 2.0 Protocol Extensions for Broker Clients



*"Specifies the OAuth 2.0 Protocol Extensions for Broker Clients, extensions to [RFC6749] (The OAuth 2.0 Authorization Framework) that allow a broker client to obtain access tokens on behalf of calling clients."*

*On Windows: MSAL library to communicate with broker (but not for PRT SSO Cookie).*

Spec: [https://learn.microsoft.com/en-us/openspecs/windows\\_protocols/ms-oapxbc/2f7d8875-0383-4058-956d-2fb216b44706](https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-oapxbc/2f7d8875-0383-4058-956d-2fb216b44706)  
MSAL <https://github.com/AzureAD/microsoft-authentication-library-for-python>



## Benefits of On-Device Broker for Credentials

### Single Sign On and Compliance Checks

#### Single Sign On (SSO)

- The broker is bound to a system session and a user (login) session
- As the user is authenticated on the device, issued tokens encode this information

#### Device Identifier

- The broker encodes a cryptographically secured device identifier into the issued tokens
- The backend (e.g. EntraID) checks from which device the login request is made

#### Compliance Checks

- The results of local device compliance checks are send to the backend and stored in a DB
- On login, the device state is queried from the DB and “Conditional Access Policies (CAP)” check it

# Enhancing OSS Applications

Add support for OIDC + MS-OAPXBC

## Web Applications

### Browser Extension

#### linux-entra-sso

- Entra ID SSO from browser (including CAP)
- Full support on Firefox
- Sufficient support on Chrome, Chromium, Vivaldi (due to missing APIs)
- Used by Linux team @MS



<https://github.com/siemens/linux-entra-sso/>

#### OIDC in Web Applications

- Web applications usually run in a stock browser (like Firefox, Chromium, ...)
- Browsers run in sandboxed environments, hence cannot directly talk to the broker
- Web Extension takes care of injecting the broker tokens into requests towards the login provider
- Allows to select account that should be used for SSO

One to rule them all:  
problem solved for all web apps

## Native Mail Clients

### Evolution Mail Client

- Standard mLinux mail client
- In EWS backend, add support for CAP (via MS-OAPXBC)
- In Data Server, add needed interfaces to pass tokens



### Thunderbird

- Supports Web Extensions
- Perform authentication with Firefox version of “linux-entra-ssso”



### What is different to browsers?

- Native applications usually use built-in browsers to implement OIDC logins
- Internally perform OIDC token update using a provided refresh token
- Broker support needs to be implemented into each client
- Async behavior of broker communication makes implementation challenging

**It's getting more and more (1)**  
Time to develop a library

Add support to each client,  
one at a time does not scale!



Unify common parts,  
put into a library

## It's getting more (2)

Time to develop a library

### Considerations when developing a library

- Which parts are generic, what is use-case specific
- Async vs. non async interfaces
- Which programming language

### Tradeoffs

- Makes implementations (and fixes) much easier
- Needs to get the library into the Linux distributions (takes time!)

### SSO-MIB C Library

- Tiny C library to acquire and convert PRT SSO tokens
- Abstracts away dbus communication and (most) JSON data handling
- CLI tool as debugging frontend



<https://github.com/siemens/sso-mib>

# FreeRDP: A Remote Desktop Protocol Implementation

The first sso-mib users

## Linux users sometimes need to access a Windows system

- Unfortunate, but that's how it is ☺
- Needed: RDP access to Azure Virtual Desktop
- Authentication uses OAUTH2 (without built-in browser), requires more complex Proof-of-Possession (PoP) tokens

## Implementation

- With sso-mib PoP tokens are acquired from local broker (only initial user-interaction needed)
- True SSO experience

## A quick upstream integration

- Release of sso-mib on 19.05.25
- Opened MR at FreeRDP project one the same day (MR 11600)
- MR was merged on 21.05.25
- Follow up MRs by maintainer one the same day
- Release of version 3.16.0 on 16.06.25

## SMTP against Office365 Endpoint

Using the sso-mib library

**OSS Contributors often need to work with mailing lists (e.g., to contribute to Linux)**

- git send-email tool to send patches to mailing lists
- often tricky to use in corporate environments
- authentication against Office365 SMTP endpoint with device trust avoids workarounds

```
80     app = mib_public_client_app_new(client_id, authority, NULL, NULL);
81     if (!app)
82         goto cleanup;
83
84     mib_client_app_set_redirect_uri(app, APP_REDIRECT_URI);
85     scopes = g_slist_append(scopes, "offline_access");
86     scopes = g_slist_append(scopes, "https://outlook.office365.com/SMTP.Send");
87
88     token = mib_client_app_acquire_token_interactive(
89         app, scopes, MIB_PROMPT_NONE, input.username, NULL, NULL, NULL);
90     if (!token) {
91         g_printerr("could not get token\n");
92         goto cleanup;
93     }
94     account = mib_prt_get_account(token);
95     g_print("username=%s\n", mib_account_get_username(account));
96     g_print("password=%s\n", mib_prt_get_access_token(token));
97     g_print("password_expiry_utc=%jd\n", mib_prt_get_expires_on(token));
98     g_print("authtype=bearer\n");
```

<https://github.com/siemens/sso-mib/blob/a96b95297cf9d1edd83c7f34769435cad4a75269/examples/smtplib/main.c>



# Works for me is not enough

Collaborate instead!

## Upstreaming and Backporting

### Upstreaming

- Do it rather **sooner than later**
- Ensure **high-quality of first submission** (then you're treated seriously and get help more likely)
- Implement for latest version
- Adhere to contribution guideline
- Adhere to code style
- Work in **baby-steps** – one feature per commit

### Backporting

- Usually, your users do not run the latest version of the tool
- Backport you patches locally for the needed versions, distribute internally

### Plan Ahead

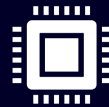
- Know the release schedule of Linux distributions
- Talk to the distro community, discuss how to get it packaged
- Reserve budget to incorporate review comments

The more you do today, the less you must do tomorrow

## Future Work: A true OSS integration

### Scratching at the Surface

Azure Entra ID  
Device  
Onboarding  
(broker  
initialization)



Himmelblau Broker



Token Acquisition  
(e.g. get PRT SSO  
Cookie)



Project Himmelblau (SUSE)  
OSS implementation of on-  
device components. WIP

<https://github.com/himmelblau-idm/himmelblau>

SSO-MIB C Library  
Browser Intune Plugin  
Evolution, ...

## Summary

- **OSS tools can support CAP:** Interfacing with the MS broker on Linux is easy
- **Per-tool effort:** Supports needs to be implemented on a per-tool basis.
- **No impact on security:** No attack / breach of the authentication protocols. Just integration.
- **Upstream first:** Contribute your changes upstream to avoid the cost of downstream maintenance

### Siemens AG

Felix Moessbauer

FT RPD CED OES-DE

[felix.moessbauer@siemens.com](mailto:felix.moessbauer@siemens.com)



siemens/sso-mib



siemens/linux-entra-sso



[opensource.siemens.com](https://opensource.siemens.com)