SLAC 2023

23.–25. Mai 2023 | Berlin

FreeIPA and external Identity Providers

Thorsten Scherf

Red Hat

What is FreeIPA?

- Identity Management solution
  - provides centralized infrastructure to manage POSIX identities across a fleet of Linux machines
  - combines 389-ds LDAP server, MIT Kerberos, SSSD, Samba, and Python-based management tools
  - optionally also provides PKI and DNS services
  - often seen as Active Directory for Linux

FreeIPA at an operating system level

- Identity and access information
  - user and group POSIX information for Linux environments through SSSD
  - user authorization through SSSD host-based access controls
- Authentication
  - Centralized Kerberos authentication with different authentication methods
  - Single sign-on to system services
  - Centralized management of SSH public keys

FreeIPA integration to non-operating system services

FreeIPA as a backend provider

- Identity provider integration:
  - Direct identity backend to a web service with LDAP driver
  - SSSD as an identity backend to a web service
    - Apache module mod_lookup_identity
    - NGINX module nginx_http_lookup_identity_module
- Authentication integration:
  - LDAP BIND
  - SPNEGO/Kerberos
    - Apache modules mod_auth_gssapi
  - PAM authentication via SSSD PAM module
    - Apache module mod_authnz_pam
    - NGINX module nginx_http_authnz_pam_module

Disadvantages

- Applications authors haven’t really mastered LDAP and Kerberos
- Typical integration approaches struggle to scale
  - a single LDAP server in a configuration
  - lack of support for more than ‘username+password’ methods
  - Java-based frameworks have outdated Kerberos support, aren’t aware about features added since 2010
  - Java-based frameworks struggle to integrate with UNIX domain sockets

OAuth 2.0 authorization framework

- Web services moved on to OAuth 2.0 authorization framework
  - Identity Provider (IdP) handles authentication and authorization, one place to focus on instead of every single app
  - Applications rely on IdP-issued grant to operate
- Think of photo and print services
- Web services map OAuth 2.0 subjects, not system-level users, it gives a bit of flexibility to map POSIX users

RFC6749

FreeIPA as a consumer of external resources

Consume external identities & authentication

- Trust to Active Directory
  - IPA clients can talk directly to AD domain controllers
- Radius
  - exposed through a Kerberos pre-authentication method
  - user details stored in FreeIPA, authentication handled by external source
- RADIUS support has some limitations though:
  - single RADIUS server end-point per each user
  - only supports ‘PIN + token’ opaque scheme

OAuth 2.0 authorization and access to system-level resources

OAuth 2.0 moves authentication step to IdP

- Authentication is not visible to OAuth 2.0 clients, they ask IdP for a grant to access resources instead
  - IdP authenticates the user, if needed, and asks the user to authorize the request
  - All this implies use of a browser and HTTP-based redirects
  - Hard to integrate without browsers being available
  - OAuth 2.0 has few authorization flows to address different use cases; they all still need the browser to be present somewhere

We want to use OAuth 2.0 framework flows to log in over SSH

- How can we avoid running a browser on the server side?

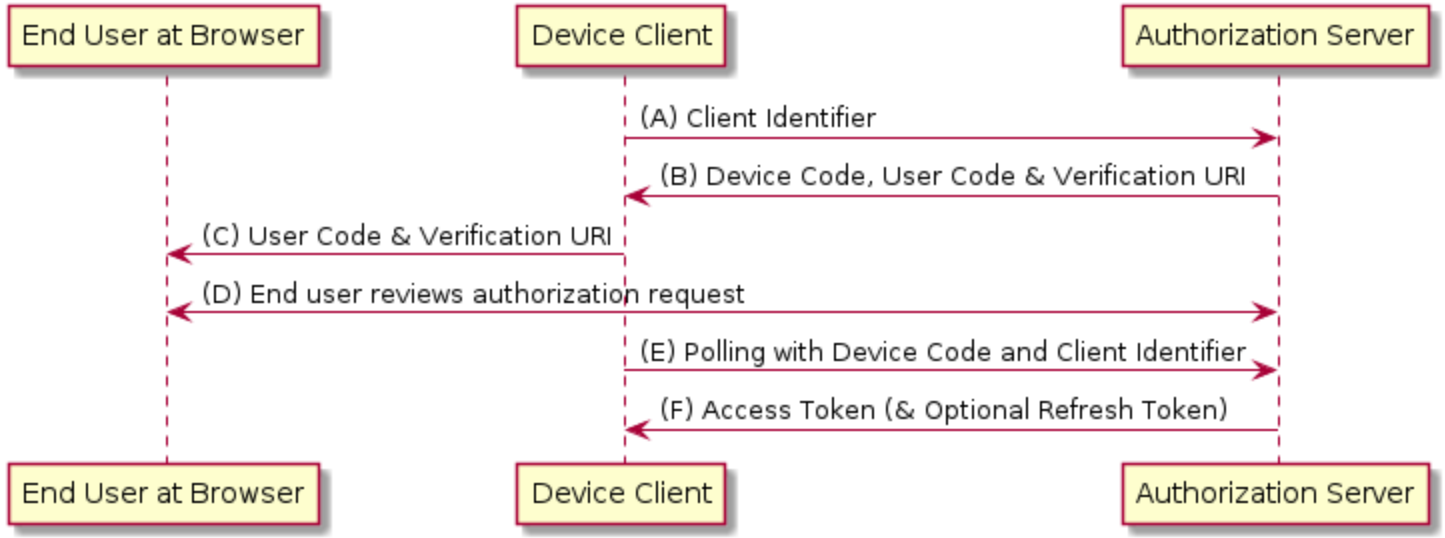
OAuth 2.0 Device Authorization Grant

OAuth 2.0 Device Authorization Grant to the rescue

- Browser to authorize the user can run anywhere
- Sounds familiar?
    - Right! Think of apps running on your smart tv-box

RFC8628

Using OAuth 2.0 device authorization grant flow



Actual flow for FreeIPA

FreeIPA client code is scoped in the MIT Kerberos pre-authentication module

- provided by SSSD project as sssd-idp subpackage
- tells KDC “I support OAuth 2.0 method, consider it”
- shows KDC response as “Authenticate at https://... and press ENTER.”

FreeIPA server side is reusing RADIUS helper ipa-otpd

- KDC side of the MIT Kerberos pre-authentication module triggers IdP support
- KDC asks RADIUS helper ipa-otpd to handle it
- ipa-otpd calls SSSD-provided oidc_child helper to talk OAuth 2.0 to user-specific IdP
- on successful authorization response from IdP, KDC issues a Kerberos ticket

User runs browser elsewhere

Kerberos ticket is issued for the user

Possible integration

- Authentication is done by an external IdP
  - FIDO2 / WebAuthn => Passwordless
- Authorization grant is turned into a Kerberos ticket by FreeIPA KDC
  - FAST required between IPA client and server
- Kerberos authentication indicator “idp” is assigned to the ticket
  - Kerberos ticket is consumed by an IPA-enrolled application
  - Application can check the authentication indicator and deny non-IdP access

OAuth 2.0 device authorization grant flow

Tested against multiple public IdPs

- Keycloak / Red Hat Single Sign-On
- Google
- Github
- Microsoft Azure
- Okta

Does not work against IdPs which do not implement OAuth 2.0 device authorization grant

- Ipsilon
- Gitlab

Roadmap

System for cross-domain identity management (SCIM v2.0)

- Automation of the user identity information exchange between identity domains
- Supported by many proprietary identity providers
- Exposes user and group information and access methods over REST API (over HTTPS)
- proof of concept SCIMv2 bridge between FreeIPA and IdPs

RFC7642
RFC7643
RFC7644
ipa-tuura

THANKS!

Credits to Alexander Bokovoy and Francisco Trivino Garcia.