LEMONLDAP::NG 2.0: MULTI-FACTOR AUTHENTICATION, IDENTITY FEDERATION, WEBSERVICE AND API PROTECTION

Clément OUDOT – Identity Solutions Manager
clement.oudot@worteks.com
Single Sign On
LDAPCon ❤️ Single Sign On

- LDAPCon 2007: The FederID Project
- LDAPCon 2011: The LemonLDAP::NG project
- LDAPCon 2015: The OpenID Connect Protocol
- LDAPCon 2017: Understanding main SSO protocols: CAS, SAML and OpenID Connect
- LDAPCon 2019: LemonLDAP::NG workshop and conference
SSO Workflow

1. First access

2. Authentication

3. Send SSO Token

4. Validate SSO token

Authentication Portal

Trust link

Application
LemonLDAP::NG Software
Project creation

Fork – version NG

Protocols CAS, SAML and OpenID

Version 1.0

Protocol OpenID Connect

Second factors (2FA)

Version 2.0

2003

2006

2010

2016

2018
Main features

- Web Single Sign On
- Access control
- Applications portal
- Authentication modules choice and chain
- Password management, account creation
- Multi-factor authentication (MFA)
- Protection of Web applications and API/WebServices
- Graphical customisation
- Packages for Debian/Ubuntu/RHEL/CentOS
Login page
Portal with application menu
Command Line Interface

root@ader-worteks:~#/usr/share/lemonldap-ng/bin/lemonldap-ng-cli info

Num : 88
Author : clement
Author IP: localhost
Date : Tue Dec 18 09:57:58 2018
Log : Edited by lnConfigEditor

Usage: /usr/share/lemonldap-ng/bin/lemonldap-ng-cli <options> action <parameters>

Available actions:
- help : print this
- info : get current configuration info
- update-cache : force configuration cache to be updated
- get <keys> : get values of parameters
- set <key> <value> : set parameter(s) value(s)
- addKey <key> <subkey> <value> : add or set a subkey in a parameter
- delKey <key> <subkey> : delete subkey of a parameter

See Lemonldap::NG::Common::Cli(3) or Lemonldap::NG::Manager::Cli(3) for more

root@ader-worteks:~#/usr/share/lemonldap-ng/bin/lemonldap-ng-cli set ldapServer 'ldap://ldap.example.com'
Free Software

- License GPL
- OW2 project
- Forge: https://gitlab.ow2.org/lemonldap-ng/lemonldap-ng
- Site: https://lemonldap-ng.org
- OW2 Community Award in 2014 and 2018
- SSO component of FusionIAM project: https://fusioniam.org/
Component roles

Portal

Manager

- Configurations
- Sessions
- Notifications
- Second factors

Handler

- Access Control
- SSOaaS
- Web Service Token
- Custom

Application menu
CAS
SAML
OpenID Connect
Self Services
SOAP/REST server
Session management
Web application protection with Handler

- Authentication
- Session creation
- Session read
- SSO cookie
- HTTP headers

Portal
Sessions
Handler
Web Application
Multi Factor Authentication
Multi Factor Authentication

- Multi-factor authentication (MFA) is a method of confirming a user's claimed identity in which a user is granted access only after successfully presenting 2 or more pieces of evidence (or factors) to an authentication mechanism:
  - knowledge (something they and only they know)
  - possession (something they and only they have)
  - inherence (something they and only they are)
One-Time Password

• One-Time Password (OTP) is a password that is valid for only one login session or transaction

• Two standards:
  • HOTP (RFC 4226): HMAC-Based One-Time Password
  • TOTP (RFC 6238): Time-Based One-Time Password

• Rely on a secret shared between user and server
TOTP

- Shared secret key K
- T0: start time
- TI: time interval
- Time Counter TC = floor((unixtime(now) − unixtime(T0)) / TI)
- TOTP = Truncate( SHA1(K ⊕ 0x5c5c… ∥ SHA1(K ⊕ 0x3636… ∥ TC)) ) & 0x7FFFFFFF
- TOTP Value = TOTP mod 10d, where d is the desired number of digits of the one-time password
Using a TOTP

- Registration on client: shared key can be registered manually or using a QR code
- Server associates shared secret to user
- At next authentication, TOTP value is computed by client and server
Universal 2nd Factor (U2F) is an open authentication standard that strengthens and simplifies two-factor authentication using specialized USB or NFC devices.

Managed by FIDO Alliance [https://fidoalliance.org/](https://fidoalliance.org/)
Using U2F

- Registration: Token generates private/public keys and a handle and send public key and handle to server
- The server associates the public key and the handle to user
- At next authentication, server sends the handle and a crypto challenge and the U2F token signs the challenge and sends it back
U2F Registration

**FIDO Client / Browser**

**U2F Authenticator**

generate:
- key $k_{pub}$
- key $k_{priv}$
- handle $h$

- $a$: challenge, origin, channel id, etc.
- $fc$

- $k_{pub}$, $h$, attestation cert, signature($a$, $fc$, $k_{pub}$, $h$)
- $s$

**Relying Party**

- AppID, challenge
- $a$

- $fc$, $k_{pub}$, $h$, attestation cert, $s$

store:
- key $k_{pub}$
- handle $h$
U2F Authentication

U2F Authenticator

FIDO Client / Browser

check AppID

handle, AppID, challenge

h, a; challenge, origin, channel id, etc.

fc

s

cntr, signature(a, fc, cntr)

Relying Party

retrieve key $k_{pub}$ from handle h

check signature using key $k_{pub}$

set cookie

cntr, fc, s
Support in LL::NG

- LemonLDAP::NG can use the following 2FA:
  - TOTP
  - U2F
  - TOTP or U2F
  - Mail
  - External
  - REST
  - Yubikey
Identity federation
Main features

- LL::NG can act as client and as server
- Attributes sharing
- Manage authentication contexts and levels
- Autogeneration of public/private keys
- Access control per services
- Publication of configuration data (metadata)
- Multi-protocols gateway
- Single logout
CAS

First access

CAS client

Service Ticket

Service ticket validation

Access to identity

CAS server

Redirection for authentication

Service Ticket
SAML

First access
IDP choice

Authentication response

Authentication request

Signature verification
Read assertion

Service Provider (SP)

Identity Provider (IDP)
OpenID Connect

First access
OP choice

Authentication request

Relying Party (RP)

Get UserInfo

OpenID Provider (OP)

First access
OP choice

Authentication request

Signature verification
Read JWT

Get UserInfo

Signature verification
Read JWT

JWT

JWT
API / WebService protection
How to protect a WebService

• Global authentication:
  • HTTP Basic
  • SSL client certificate
• User oriented authentication?
LL::NG ServiceToken Handler

• New Handler "Service Token" installed between application and WebService

• Main Handler generates a token based on time session_id and virtual hosts: cipher(time, session_id, vhost_list)

• The token is sent by application to WebService

• The Handler "Service Token" intercepts the token, validates it and apply access rules, and sent HTTP headers to WebService
LL::NG ServiceToken Handler

Authentication

Portal

Session creation

Sessions

SSO cookie

Session read

Handler

HTTP headers

Web Service

HTTP headers

Handler Service Token

Token

Web Application
Using OAuth2

- When LL::NG acts as OIDC provider, it delivers an OAuth2 access token
- This access token can be validated with different operations:
  - Call /oauth2/userinfo, which will return user attributes
  - Call /oauth2/introspect, which will return token information (including the token owner) – see RFC 7662
  - Use LL::NG OAuth2 Handler
LL::NG OAuth2 Handler

Authentication

OIDC response

Portal

Session creation

ID Token
Access Token

Web Application

Session read

Sessions

Handler
OAuth2

HTTP headers

Web Service

Access Token
Example – UserInfo Endpoint

$ curl -k \
-H "Authorization: Bearer a74d504ec9e784785e70a1da2b95d1d2" \\nhttps://auth.openid.club/oauth2/userinfo | json_pp

{
  "family_name" : "OUDOT",
  "name" : "Clément OUDOT",
  "email" : "clement@oodo.net",
  "sub" : "coudot"
}
Example – Introspection Endpoint

```bash
$ curl -k \
   -H "Authorization: Basic bGVtb25sZGFwOnNlY3JldA==" \ 
   -X POST -d "token=a74d504ec9e784785e70a1da2b95d1d2" \ 
   https://auth.openid.club/oauth2/introspect | json_pp

{
    "client_id" : "lemonldap",
    "sub" : "coudot",
    "exp" : 1572446485,
    "active" : true,
    "scope" : "openid profile address email phone"
}
```
Example – Oauth2 Handler

```bash
$ curl -k \
  -H "Authorization: Bearer a74d504ec9e784785e70a1da2b95d1d2" \ 
  https://oauth2.openid.club/api.pl

{
  "check" : "true",
  "user" : "coudot"
}
```
THANKS FOR YOUR ATTENTION

More informations:

- info@worteks.com
- @worteks_com
- linkedin.com/company/worteks