EXTENDING OPENLDAP PASSWORD POLICY MODULE WITH PPM

David Coutadeur

06/11/2019

PLAN

1. ABOUT ME
1. INTRODUCTION TO PASSWORD POLICIES
2. THE PASSWORD POLICY OVERLAY
3. PPM WORKING
4. PPM FEATURES
5. CONCLUSION

PART 1

ABOUT ME

ABOUT ME

DAVID COUTADEUR

- Open-source enthusiast
- ~ 10 years of experience in IT
- Specialized in identity and access management
- Contributor to LemonLDAP::NG, LTB-project
- Manager of LinID team at LINAGORA

ABOUT ME

- French IT service company, ~ 20 years
  - Integration
  - Support
  - Software editor
- Specialized in free and open-source software
- Digital sovereignty
- 

Activity stack :

ABOUT ME

LTB-PROJECT

- Community around LDAP integration:
  - **"OpenLDAP-LTB"**: Debian / Red-Hat packages for OpenLDAP
  - LDAP monitoring modules
  - **"Self Service Password"**: Web interface to reset password by mail, by sms, by answer to a question
  - **"White Pages"**: directory manager
  - Various scripts helping LDAP integration and management
  - OpenLDAP overlays:
    - **"openldap-ppolicy-check-password"**: old password policy module
    - **"ppm"**: new Password Policy Module

One URL: https://ltb-project.org

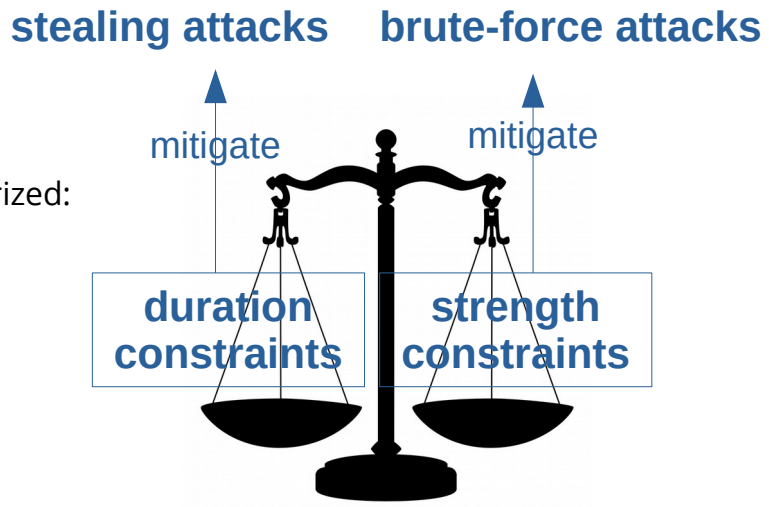
PART 2

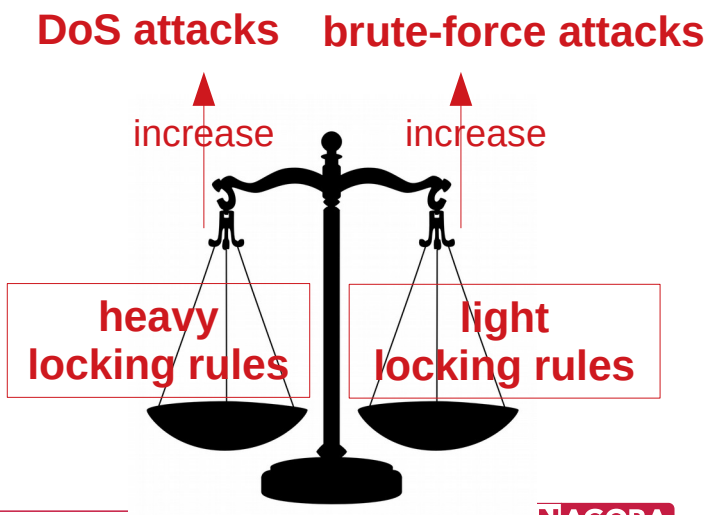
INTRODUCTION TO PASSWORD POLICIES

INTRODUCTION TO PASSWORD POLICIES

WHAT IS A PASSWORD POLICY?

- A password policy is a set of rules about the user password. They can be categorized:
  - **rules about strength**
    - length (minimum, maximum)
    - mandatory use of some character classes (upper, lower, digit, special)
    - word blacklist
  - **rules about password duration**
    - requirement to change the password after a delay (fixed or inactivity delay)
    - prohibition of password history reuse
    - prohibition to change the password before a delay
  - **locking rules**
    - locking after some bad authentications
    - locking because of unusual activity detection





INTRODUCTION TO PASSWORD POLICIES

NORMALIZATIONS

- No universal password policy!
- Recommandations / standards:
  - **NIST** (National Institute of Standards and Technology), june 2017:
    - 8 → 64 characters
    - no character classes required
    - No periodic change required
  - IETF internet draft: **"Password Policy for LDAP Directories"**
    draft-behera-ldap-password-policy (version 00: october 1999, version 10: august 2009), implemented into main directories:
    - OpenLDAP password policy overlay
    - SUN Directory Server
    - Tivoli Directory Server
    - Fedora Directory Server
    - Red-Hat Directory Server

INTRODUCTION TO PASSWORD POLICIES

DRAFT-BEHERA-LDAP-PASSWORD-POLICY

- version 10, august 2009, expired in january 2010
- Defines:
  - ✅ the functional specifications of the password policy
  - ✅ the LDAP schema for password policies and operationnal attributes
  - ✅ the extended request and response controls
  - ✅ the way the server should / must include the ppolicy into LDAP operations processing (bind, modify,...)
  - ✅ the way the clients should / must include the ppolicy into LDAP operations processing (bind, modify,...)
  - ✅ general considerations for managing policies (scope defined by SubtreeSpecification attribute, ppolicy overload)
  - ✅ replication and security considerations
- Does **not** define:
  - ❌ the enforcement of password quality criteria, which are implementation specific
  - ❌ an automatic way to force password reset after a modification is performed by an administrator

PART 3

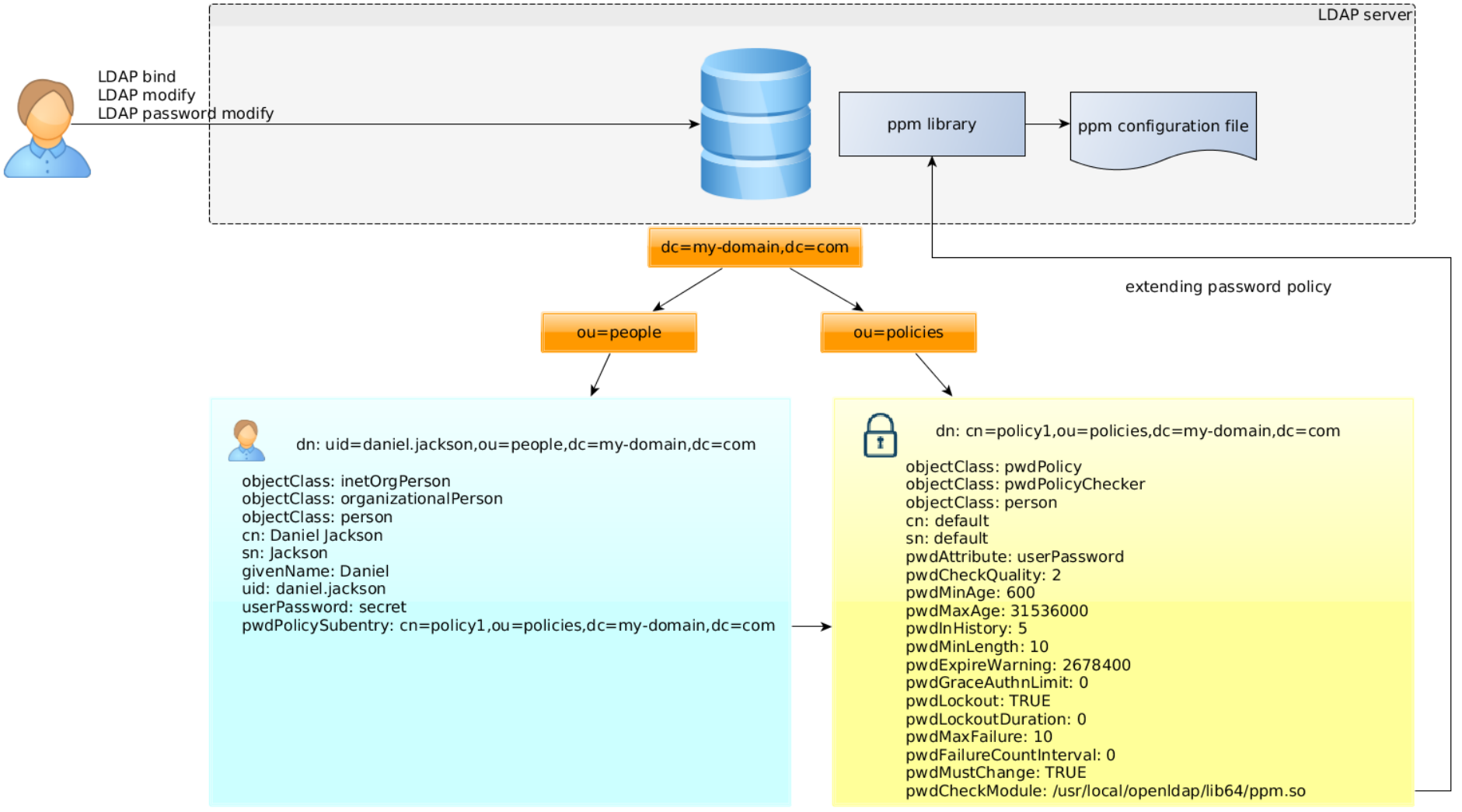
THE PASSWORD POLICY OVERLAY

THE PASSWORD POLICY OVERLAY

- OpenLDAP implements the IETF internet draft:
  - ☑ the LDAP schema for password policies and operationnal attributes
  - ☑ the extended request and response controls
  - ☑ the server implementation
  - ☑ the client implementation (ldapsearch, ldapmodify,...)

- What OpenLDAP password policy does not implement:
  - ☒ support of protection against brute-force attacks, based upon *pwdMinDelay*, *pwdMaxDelay* attributes
  - ☒ support of locking based upon *pwdLastSuccess*, *pwdMaxIdle* attributes
  - ☒ support of *pwdMaxLength* attribute
  - ☒ support of *pwdStartTime*, *pwdEndTime* attributes
  - ☒ password policies managed by scope in the DIT or by group

THE PASSWORD POLICY OVERLAY



PART 4

PPM WORKING

PPM WORKING

PPM DESIGN

- implement the minimum, most popular, basic strength rules that slapo-ppolicy misses → OK
- extensible → at some point (small piece of C code)
- efficient → not critical (only called in password modification context)
- secured → TODO (audit of code)

PPM IN A FEW WORDS

- C code, dynamic shared object, ~ 600 lines
- implement some strength rules
- 1 configuration file, read for every password change
- packaging:
  - OpenLDAP-LTB packages for Debian, Red-Hat
  - test binary for testing password strength
- current release: v1.8, 20th August 2019 (bugfix when checkRDN enabled and username contains too short parts)
- OpenLDAP Public License

PART 5

PPM FEATURES

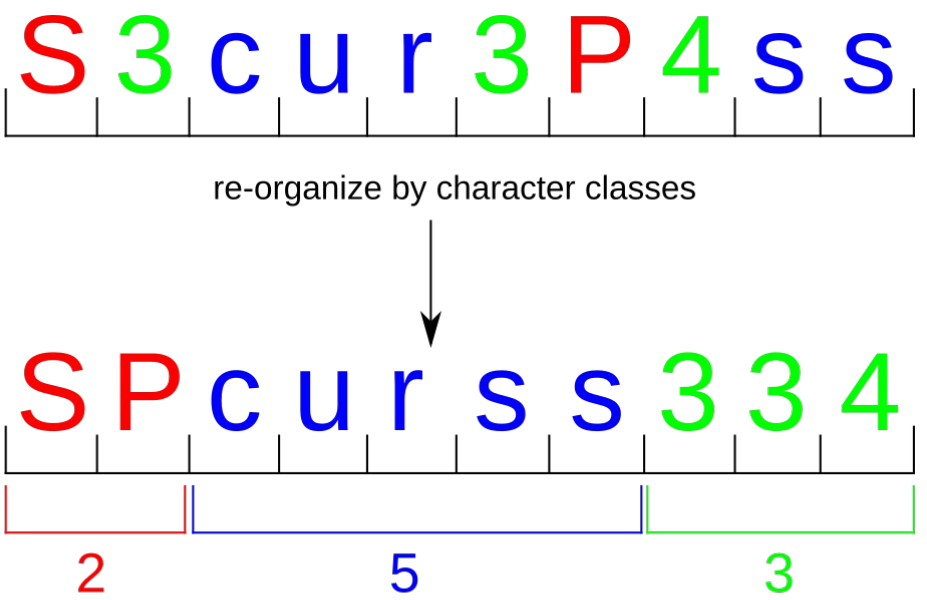
PPM FEATURES

CHARACTER CLASSES PARAMETERS

- definition of character classes
- minQuality
- min_for_point (for each character class)
- min (for each character class)

Policy example:

```
minQuality 3
class-upperCase ABCDEFGHIJKLMNOPQRSTUVWXYZ 0 1
class-lowerCase abcdefghijklmnopqrstuvwxyz 0 1
class-digit 0123456789 0 1
class-special <>,?;.:/!§ù%*µ^¨$£²&é~"#' 0 1
```



$quality_{uppercase}$ = 1 if 2 >= $min\_for\_point_{uppercase}$, else 0

$quality_{lowercase}$ = 1 if 5 >= $min\_for\_point_{lowercase}$, else 0

$quality_{digit}$ = 1 if 3 >= $min\_for\_point_{digit}$, else 0

Criteria:

$2 >= min_{uppercase}$

$5 >= min_{lowercase}$

$3 >= min_{digit}$

$quality_{uppercase} + quality_{lowercase} + quality_{digit} >= minQuality$

PPM FEATURES

OTHER STRENGTH PARAMETERS

- maxLength
- checkRDN
- forbiddenChars
- maxConsecutivePerClass
- useCracklib
- cracklibDict

Policy example:

```
maxLength 12
checkRDN 1
forbiddenChars ££
maxConsecutivePerClass 8
useCracklib 1
cracklibDict /var/cache/cracklib/cracklib_dict
```

CheckRDN criteria:

dn: uid=daniel.jackson,ou=people,dc=my-domain,dc=com

- new password: secret → OK
- new password: daniel → KO

PART 6
CONCLUSION

CONCLUSION

FOOD FOR THOUGHT

- Write a standard for strength policy rules
- Reject password changes according to context criteria:
  - attributes in LDAP entry
  - attributes linked to LDAP entry (group,...)
  - environment: client IP address range, date and time of request
- Apply password policies to group of users
  - should be done in the password policy overlay
- Implement automatic locking rules
  - attributes defined in the IETF draft
  - standardize explockout overlay? (https://github.com/davidcoutadeur/explockout)

THANK YOU FOR YOUR ATTENTION

LINAGORA
www.linagora.com
Facebook : LINAGORA - Twitter : @linagora
Tour Franklin 31ème étage
100 Terrasse Boieldieu
92042 PARIS LA DEFENSE FRANCE
Tél. : +33 (0)1 46 96 63 63 - Fax : +33 (0)1 46 96 63 64