slapo-rwm

Overlay rwm

Overlay rewrite/remap. Permet d'effectuer des réécritures basiques de DN données et des mapping d'attribut/classe d'objet.

Mapping

rwm-map {attribute | objectclass} <local name> {<foreign name> | *} Map les attributeTypes et les objectClasse.

rwm-normalize-mapped-attrs {yes|no} à yes, rwm tente de normaliser les valeurs des attributs mappés (utile lors que les attributs distant sont inconnus du serveur)

rwm-drop-unrequested-attrs {yes|no} à yes, rwm drop les attributs qui ne sont pas explicitement demandés par une opération de recherche.

Suffix Massaging

Permet de mapper un contexte de nommage réel et virtuel. Permet par exemple de créer des vues virtuelles.

rwm-suffixmassage [<virtual naming context>] <real naming context> Réecriture de contexte de nommage.

Rewriting

Une chaîne est réécrite en fonction d'un jeu de règles, appelé un contexte de réécriture. Les règles sont basé sur des expressions régulières étendue POSIX avec correspondance de sous-chaîne.

<rewrite context> : := <rewrite rule> [...]

<rewrite rule> : := <pattern> <action> [<flags>]

Passes

Une chaîne entrante est matchée avec un jeu de **rewriteRules**. Les règles sont faites de **regex match pattern**, un **substitution pattern** et un jeu d'**actions**, décris par un jeu de **flags** optionnels. En cas de correspondance, la réécriture de chaîne est effectuée en accord avec le motif de substitution qui permet de référer à une sous-chaîne matchée dans la chaîne. Chaque règle est exécutée récursivement Un mappage est un objet générique qui map un motif de substitution à une valeur. Les flags sont divisés en "**Pattern Matching Flags**" et en "**Action Flags**".

Pattern Matching Flags

C respecte la casse

R Utilise les expressions régulière POSIX standard.

M{n} Ne permet pas plus de n récursion pour un règle.

Action Flags

: Applique la règle une seule fois (pas de récursion)

@ Arrête d'appliquer une règle en cas de correspondance

Stop l'opération courante si la règle correspond et retourne un unwilling to perform

G{n} Saute n règles en arrière

I Ignore les erreurs dans les règles

U{n} Utilise n comme code de retour si la règle match.

Substitution Pattern Syntax

Tout ce qui commence avec '**$**' nécessite une substitution. La seule exception est '**$$**', qui est retourné en un simple '**$**'. La substitution de base est '**$<d>**', où **<d>** est un chiffre ; **0** signifie toute la chaîne, de **1** à **9** pour des submatch.

un '$' suivi d'un '{' invoque une substitution avancée :
'$' '{' [<operator>] <name> '(' <substitution> ')' '}'
où
<name> : := [a-z][a-z0-9]* (case insensitive)
<operator> : := '>' '|' '&' '&&' '*' '' '$'**

<substitution> doit être un motif de substitution valide, sans limite de niveau d'imbrication. les opérateurs sont :

> Invocation de sous-contexte

| Invocation de commande externe

& Assignement de variable

***** Déréferencement de variable

$ Déréférencement de paramètre

Rewrite Context

Un contexte de réécriture est un jeu de règles qui sont appliquées en séquence. Quand une réécriture de chaîne est requise, on invoque le contexte de réécriture appropriée. Chaque opération serveur de base est associée à un contexte de réécriture, ils sont divisés en 2 groupes principaux : **client -> serveur** et **serveur -> client** :

Client->serveur

(default) Si définis et sans contexte spécifique

bindDN bind

searchDN search

searchFilter search

searchFilterAttrDN search

compareDN compare

compareAttrDN compare AVA

addDN add

addAttrDN add AVA (DN portion of "ref" excluded)

modifyDN modify

modifyAttrDN modify AVA (DN portion of "ref" excluded)

referralAttrDN add/modify DN portion of referrals

renameDN modrdn (the old DN)

newSuperiorDN modrdn (the new parent DN, if any)

newRDN modrdn (the new relative DN)

deleteDN delete

exopPasswdDN password modify extended operation DN

Serveur->client

searchEntryDN search (seulement si défini, pas de défaut, agis sur le DN des entrées recherchées)

searchAttrDN search AVA (seulement si défini, défaut sur searchEntryDN, agis sur les attributs type DN des entrées recherchées)

matchedDN all ops (Seulement si applicable ; défaut à searchEntryDN)

referralDN all ops (Seulement si applicable ; défaut : none)

Basic Configuration Syntax

rwm-rewriteEngine { on | off } Autorise la réécriture

rwm-rewriteContext <context name> [alias <aliased context name>] <Context Name> est le nom qui identifie le contexte, par ex le nom utilisé par l'application pour référer au jeu de règles qu'il contient. Un contexte peut être un alias d'un autre. Dans ce cas il ne contient pas de règle.

rwm-rewriteRule <regex match pattern> <substitution pattern> [<flags>] Détermine comment une chaîne peut être réécrite si un motif est matché.

Additional Configuration Syntax

rwm-rewriteMap <map type> <map name> [<map attrs>] Permet de définir un map qui transforme des réécritures en quelque chose d'autre. Le map est référencé dans le motif de substitution d'une règle.

rwm-rewriteParam <param name> <param value> Définis une valeur avec un scope global, qui peut être dé-référencé par la commande ${$paramName}

rwm-rewriteMaxPasses <number of passes> [<number of passes per rule>] Définis un nombre maximum de réécritures totales en une seul opération de réécriture.

Maps

les mappages supportés sont :

LDAP <URI> [bindwhen=<when>] [version=<version>] [binddn=<DN>] [credentials=<cred>] Étend la valeur en effectuant une simple recherche LDAP. bindwhen détermine quand la connection est établie (now - créée au démarrage, later - quand nécessaire, everytime - a chaque fois).

slapd <URI> étend une valeur en effectuant une recherche LDAP interne.

Exemple

activer le rewriting
rwm-rewriteEngine on
Tous les flux de données du client vers le serveur référrant au DN
rwm-rewriteContext default
rwm-rewriteRule "(.+,)?<virtualnamingcontext>$" "$1<realnamingcontext>" " :"
Règle de filtre vide
rwm-rewriteContext searchFilter
Tous les flux de données du serveur vers le client
rwm-rewriteContext searchEntryDN
rwm-rewriteRule "(.+,)?<realnamingcontext>$" "$1<virtualnamingcontext>" " :"
rwm-rewriteContext searchAttrDN alias searchEntryDN
rwm-rewriteContext matchedDN alias searchEntryDN
Diverses règles vides
rwm-rewriteContext referralAttrDN
rwm-rewriteContext referralDN
Tout ce qui est définis ici va dans le contexte default. Change le contexte de nommage de tout ce qui est envoyé
rwm-rewriteRule "(.+,)?dc=home,[ ]?dc=net$" "$1dc=OpenLDAP, dc=org" " :"
Commence un nouveau contexte (termine l'entrée du précédent) Ajoute un blanc entre les parties du DN si non présent
rwm-rewriteContext addBlanks
rwm-rewriteRule "(.*),([^].*)" "$1, $2"
Celui-ci supprime les blanc :
rwm-rewriteContext eatBlanks
rwm-rewriteRule "(.*), (.*)" "$1,$2"
Chaque contrôle revient au context default, les règles sont ajoutée à celles existantes, et pipe dans addBlanks :
rwm-rewriteContext default
rwm-rewriteRule ".*" "${>addBlanks($0)}" " :"
Réecrit la base de recherche sur les règles default :
rwm-rewriteContext searchDN alias default
Les résultats de recherche avec un DN openldap sont réecris avec dc=home,dc=net.
rwm-rewriteContext searchEntryDN
rwm-rewriteRule "(.*[^],)?[ ]?dc=OpenLDAP,[ ]?dc=org$" "${>eatBlanks($1)}dc=home,dc=net" " :"
Bind avec le mail au lieu du DN. on fait un map ldap qui transforme les attributs en DN
rwm-rewriteMap ldap attr2dn "ldap ://host/dc=my,dc=org?dn?sub"
puis on détecte le DN fait d'un simple email.
rwm-rewriteContext bindDN
rwm-rewriteRule "^mail=[^,]+@[^,]+$" "${attr2dn($0)}" " :@I"
exemple plus sophistiqué
rwm-rewriteContext bindDN
rwm-rewriteRule ".+" "${&&binddn($0)}$0" " :"
A search filter containing 'uid=' is rewritten only if an appropriate DN is bound. To do this, in the first rule the bound DN is
dereferenced, while the filter is decomposed in a prefix, in the value of the 'uid=<arg>' AVA, and in a suffix. A tag '<>' is appended to the DN.
If the DN refers to an entry in the 'ou=admin' subtree, the filter is rewritten OR-ing the 'uid=<arg>' with 'cn=<arg>'; otherwise it is left as is. This could be
useful, for instance, to allow apache's auth_ldap-1.4 module to authenticate users with both 'uid' and 'cn', but only if the request comes from a possible
'cn=Web auth,ou=admin,dc=home,dc=net' user.
rwm-rewriteContext searchFilter
rwm-rewriteRule "(.*\\()uid=([a-z0-9_]+)(\\).*)" "${**binddn}<>${&prefix($1)}${&arg($2)}${&suffix($3)}" " :I"
rwm-rewriteRule "^[^,]+,ou=admin,dc=home,dc=net$" "${*prefix}|(uid=${*arg})(cn=${*arg})${*suffix}" " :@I"
rwm-rewriteRule ".*<>$" "${*prefix}uid=${*arg}${*suffix}" " :"
This example shows how to strip unwanted DN-valued attribute values from a search result; the first rule matches DN values below "ou=People,dc=example,dc=com";

in case of match the rewriting exits successfully. The second rule matches everything else and causes the value to be rejected.
rwm-rewriteContext searchEntryDN
rwm-rewriteRule ".+,ou=People,dc=example,dc=com$" "$0" " :@"
rwm-rewriteRule ".*" "" "#"

Exemple de mapping

map objectclass groupOfNames groupOfUniqueNames
map attribute uniqueMember member
Jeu d'attribut limité, map cn, sn, manager, description a eux-même, tous les autres sont supprimés.
map attribute cn *
map attribute sn *
map attribute manager *
map attribute description *
map attribute *