GWME-7.1.1-11 - Foundation update with Advanced LDAP

Problem

Several performance and behavior issues have reported by customers in 7.1.1. These include

- GWMON-12883 - Virtualization dashboard does not show virtual hosts if they were previously monitored by Nagios.
- GWMON-12950 - Allow for passthrough queries in Rest API. Passthrough queries are prefixed by 'pass:'. These queries will not have any query helper functions applied but just be executed as is.
- GWMON-12156 - Plugin cannot be uploaded to foundation manage plugin section
- GWMON-12856 - Required custom group name in LDAP longer than allowed string length
- GWMON-12912 - Enable caching for improved logperf performance and scalability
- GWMON-12172, GWMON-13016 License page blank issue
- GWMON-13075, GWMON-12995 LDAPAggregator SSL keystore configuration, (useful for self-signed certificates), and enable TLSv1.2 for JDK 1.7.
- GWMON-12972 Add the following
  - The need for multiple LDAP endpoints where users are not in a single catalog (so called Global Catalog issue)
  - The need to authenticate users whose accounts are located in multiple containers in the same catalog
  - The need to authenticate users whose accounts are located in nested containers
  - Corrected Exoadmin screen for Membership permissions
- GWMON-13139, GWMON-13126 - Resolve issue with a comma in the LDAP AD Distinguished Name field which cause 403 Not Authorized error on Role Search

Further we are rolling up the changes incorporated in other patches which alter war and jar files touched in this patch, these patches are no longer required and should not be applied once this patch has been deployed:

- GWME-7.1.1-1 - LDAP Caching
- GWME-7.1.1-8 - Status Viewer auto-refresh

Solution

Apply the attached update to received the performance fixes. For the Advanced LDAP support you will need to make further adjustments to the configuration.

> If you already have LDAP configured from before applying this patch but do not need to use multiple endpoints your existing configuration will be loaded correctly without additional configuration.

Installation

1. Download the patch file tar archive to, for example the /tmp directory

Name	Size	Creator	Creation Date	Comment

TB7.1.1-11.foundation_update_with_a...	60.42 MB	Hans Kriel	Sep 01, 2017 15:00	MD5: 4105ce058f236da3ab8e31143e9f02e0

2. Decompress the install script and files and run the install script. They will appear in subdirectory TB7.1.1-11.foundation_update_with_advanced_ldap_support. Go there and make sure to set ownership and permission.

```
tar xvf TB7.1.1-11.foundation_update_with_advanced_ldap_support.tgz
cd TB7.1.1-11.foundation_update_with_advanced_ldap_support
./TB7.1.1-11_install.sh
```

The patch directory will be noted **(in the production version of the updater)** with the facts of this update, along with backup files to be used in the uninstall phase if necessary.

Jar file names are correctly identified with -11 suffix. Note that the -11 suffix is important to you when executing the command line hash generator.

In the event that you had previously installed TB7.1.1-1 and/or TB7.1.1-8 this patch will detect the condition. It will handle both making a backup to restore the files involved as well as replacing the -1 and -8 patch files with the rolled up patches.

Configuration

Application of this upgrade on a system using the default authentication requires no further configuration.

For LDAP support, "josso-gateway-config.xml" must reference "josso-gateway-ldap-stores.xml" and not "josso-gateway-gatein-stores.xml".

Application on a system previously configured with Josso Ldap store will result in continued operation using the legacy credentials in the "josso-gateway-ldap-stores.xml" file.

To take advantage of the new facilities you can make changes to "foundation.properties". If ldap configurations are found in this file, configuration settings in "josso-gateway-ldap-stores.xml" are ignored.

Details for updating these files are below.

Enabling LDAP Authentication

The use of the LDAP authentication is controlled by the same setting in "josso-gateway-config.xml" which is defaulted to use the local gatein store. To facilitate use of LDAP AD or OpenLDAP you will first change the following line. If LDAP was previously enabled the change will already be present.

Edit "josso-gateway-config.xml" and replacing:

/usr/local/groundwork/foundation/container/josso-1.8.4/lib/josso-gateway-config.xml - line 109

```
<s:import resource="josso-gateway-gatein-stores.xml" />
```

with

/usr/local/groundwork/foundation/container/josso-1.8.4/lib/josso-gateway-config.xml - line 109

```
<s:import resource="josso-gateway-ldap-stores.xml" />
```

Endpoint Definitions

The endpoints are added to the foundation.properties file. Each endpoint has a section in the file.

Multiple domains are configured by replicating sets of properties for AD or OpenLDAP below. Only properties that need to be overridden need to be copied, (the rest will default as below based on the type of domain).

- If any domains are configured here, the JOSSO endpoint configuration in josso-gateway-ldap-stores.xml is ignored.
- If NO domains are configured in foundation.properties, the JOSSO configuration is loaded into the LDAP Aggregator as the default domain.

Each specified endpoint is searched separately; the credentials and OU/CN directory in one endpoint have no bearing on the others.

Enabling domain prefixes in usernames

The requirement of using a domain in the user name is controlled by a parameter in foundation.properties whose default is false, no domain required.

/usr/local/groundwork/config/foundation.properties - line 294
`core.security.ldap.domain_prefix_required = true`

If multiple endpoint domains are specified and a user name is in more than one, the possibility exists that the authentication will be on the wrong domain and role access will not be granted properly. Therefore we recommend that this be set to true and that users be required to enter the domain string "domain\user".

These are valid login principals (notice the slash can go either way in the login process). The domain name that the user enters is in the example, "demo" or "windows2012":

- `demo\user`
- `windows2012/user`

Domain property naming considerations

Note that domain names in the endpoint definitions have **no relationship to the actual DN domain**. In fact, the domain names these endpoints are known by cannot contain the '.' character. Valid names might be 'Demo' or 'Windows2012'. These generally look like Windows NetBios domain names and are used as prefixes on the principle name during login. So if the actual DN domain name is a simple string you might use it, but observe the rule.

UPN forms are not currently supported. The default domain can also be configured with no domain specified in the properties below. The default domain, if defined, will be used to look up **users that are not authenticated with a domain prefix**.

Otherwise, when a login prefix is not entered for authentication, the named domains are searched **in the order they are defined** in this file and on the first authentication success the search terminates.

Configuring each of the properties for a specific name or default domain must utilize the following forms (core.security.ldap.config.) in the properties file

1. a named domain, (no '.' allowed in domain name):
    a. core.security.ldap.config.<domain name>.<property name> = ...

domain namespaced example
`core.security.ldap.config.windows2012.provider_url =  ldap://10.0.0.15`

2. the default domain:
    a. core.security.ldap.config.<property name> = ...

default namespaced example
`core.security.ldap.config.provider_url =  ldap://10.0.0.15`

Available LDAP configuration properties

Normally, only these property names need to be specified for each domain endpoint:

- server_type
- provider_url
- security_principal
- security_credential
- users_ctx_dn
- roles_ctx_dn

This is the full list of property names that can be configured per domain:

- credential_query_string
- enable_start_tls
- initial_context_factory
- ldap_search_scope
- principal_uid_attribute_id
- principle_lookup_attribute_id
- provider_url
- role_attribute_id
- role_matching_mode

- roles_ctx_dn
- security_authentication
- security_credential
- security_principal
- security_protocol
- server_type
- trust_store
- trust_store_password
- uid_attribute_id
- updatable_credential_attribute_id
- user_certificate_attribute_id
- user_properties_query_string
- users_ctx_dn

Security credential encryption

The security credential is still required as an encrypted string. Use the following command lines to generate the string, substituting the actual password for the example PASSWORD

```
/usr/local/groundwork/java/bin/java -cp
/usr/local/groundwork/jpp/modules/org/jasypt/main/jasypt-1.9.2.jar:/usr/local/groundwork/jpp/modules/org/
org.groundwork.foundation.ws.impl.JasyptUtils --encrypt PASSWORD 2>/dev/null
```

The result will look like this:

```
***ENCRYPTED VALUE=2eH7t2u82Cc4nfeNqhQfxK3mboEMkMBmY
```

This value will be used for the security_credential property for the corresponding domain configuration.

LDAP search scope

For example, where users are in a single master Users OU the search has only a single level. Suppose that the customer has users in a nested form, buried in subdirectories. The Aggregator allows us to define an endpoint at the top of the directory tree, and the search will descend the tree until it has either exhausted the possibilities (no match) or discovered the user (first match) and attempted authentication. The attribute in the endpoint spec that controls this is

```
ldap_search_scope = SUBTREE
```

Alternatively, you may define two or more endpoints for the same LDAP, naming scope as the "BASE" (just the indicated container or OU) or "ONE LEVEL" (objects subordinate to the named base but *not the base*) instead of "SUBTREE" (the base name and all nested objects to the maximum depth). In this way you can limit the searches according to the manner by which the customer has organized users.

LDAPS connections

When connecting to an LDAP provider that is protected by SSL or TLS two changes are needed:

1. Install the certificates from the CA into the certificate store:

   ```
   /usr/local/groundwork/java/bin/keytool keytool -import -noprompt -storepass changeit -keystore
   /usr/local/groundwork/java/jre/lib/security/cacerts -alias MY-CERTIFICATE-NAME -file
   MY-CERTIFICATE-NAME.pem
   ```

   

   Certificate Encoding
   The certificates being imported MUST be PEM encoded certificates or the import will fail. If exporting from Microsoft you should select a Base64 encoded CER certificate type. Do not include the private key when you export.

2. change the protocol used in the `provider_url` from `ldap://` to `ldaps://`:

   ```
   core.security.ldap.config.provider_url = ldaps://10.0.0.35
   ```

Examples

Here are three examples of working configurations.

Take special note of the "domain" portion of the configuration in each example. Both "windows2012" and "demo" are arbitrary. The actual domain names are "corp" and "demo". We suggest avoiding using the actual domain name so that it does not become the unconscious rule, later causing an error where the actual domain had a character like "." embedded.

Whatever you decide, the string you choose will be the one that users must enter, so for example "demo/jdoe" or "windows2012\jsmith". The slash can go either way, the form is always domain followed by slash followed by user.

Note that the port is not specified if the server you are pointing to is using default settings (389 for cert-less, 636 for LDAPs). In the third example you can see the form used for specified port.

```
# 'windows2012' AD endpoint:
#
core.security.ldap.config.windows2012.server_type = AD
core.security.ldap.config.windows2012.provider_url = ldap://10.0.0.15
core.security.ldap.config.windows2012.security_principal = cn=ldapauth,cn=Users,dc=corp,dc=localdomain
core.security.ldap.config.windows2012.security_credential = XcuJVdPmzFo9egZ4a24XFsoTzoeZafKM
core.security.ldap.config.windows2012.users_ctx_dn = cn=Users,dc=corp,dc=localdomain
core.security.ldap.config.windows2012.roles_ctx_dn = ou=GWRoles,dc=corp,dc=localdomain
#
# 'demo' AD endpoint:
#
core.security.ldap.config.demo.server_type = AD
core.security.ldap.config.demo.provider_url = ldaps://10.0.0.25
core.security.ldap.config.demo.security_principal = cn=ldapauth,cn=Users,dc=demo,dc=com
core.security.ldap.config.demo.security_credential = 2eH7t2u82Cc4nfeNqhQfxK3mboEMkMBmY
core.security.ldap.config.demo.users_ctx_dn = cn=Users,dc=demo,dc=com
core.security.ldap.config.demo.roles_ctx_dn = ou=GWRoles,dc=demo,dc=com
#
# 'default' AD endpoint:
#
core.security.ldap.config.server_type = AD
core.security.ldap.config.provider_url = ldaps://10.0.0.35:636
core.security.ldap.config.security_principal = cn=ldapauth,cn=Users,dc=demo,dc=com
core.security.ldap.config.security_credential = 3eH7t2u82Cc4nfeNqW7fxK3mboEMkMBmY
core.security.ldap.config.users_ctx_dn = cn=Users,dc=demo,dc=com
core.security.ldap.config.roles_ctx_dn = ou=GWRoles,dc=demo,dc=com
```

Uninstallation

1. Run the uninstall script, and respond to the prompts.

```
./TB7.1.1-11_uninstall.sh
```

The patch directory will be processed to reflect the restoration of the files and uninstall steps.