gadspwsync – Sync Google Apps passwords with eDirectory

The gadspwsync script allows for the synchronization of Google passwords with eDirectory passwords using the free Google Apps Directory Sync tool from Google. It is a one-way sync from eDirectory to Google allowing existing eDirectory password policies to be applied to Google passwords. If a user changes their Google password it will get overwritten the next time synchronization with Google occurs.

Google Apps Directory Sync can only read passwords from LDAP if they are stored as SHA1 or MD5 hashes or as Plaintext. Other password formats are not compatible. The basis of the script is to retrieve a user's password from eDirectory using Universal Password and the Getpass Cool Tool as a SHA1 hash and write it back to eDirectory to an unused attribute.

Testing revealed that the script can compare and synchronize eDirectory passwords with the SHA1 hash for Google in 32 seconds for over 600 users. The Google Apps Directory Sync synchronization runs for about 1 minute and 30 seconds in the same environment. These benchmarks will vary depending on the network, size of user base, and Internet speed.

By default eDirectory restricts [Public] from reading most user attributes. During testing of the script it was discovered that some networks have [Public] configured with read [All Attribute Rights] exposing the SHA1 password hash to anonymous LDAP binds. It is suggested to test for this with an LDAP browser to ensure that user passwords remain secure. Take the necessary steps to secure eDirectory if it is found that anonymous LDAP binds can read all user attributes.

Required Utilities

- SUSE Linux Enterprise Server
- Universal Password
- Getpass 2.1 Cool Tool by Timothy Patterson
- OpenLDAP2 Client Utilities
- OpenSSL
- Google Apps Directory Sync (GADS) – Linux version

1. Configure a Universal Password policy for the users being synced with Google Apps. More information about configuring Universal Password can be found at http://www.novell.com/documentation/password_management33/.

2. Download Getpass 2.1 from Novell's Cool Tools website. Install and configure Getpass and its prerequisites per the included documentation. Getpass 2.1 can be found at http://www.novell.com/communities/node/11696/getpass-21-universal-password-retrieval-utility-updated.

3. Install OpenLDAP2 Client Utilities and OpenSSL using YaST Software Management (if not already installed).

4. Install Google Apps Directory Sync.

5. Edit the `/etc/openldap/ldap.conf` file setting the following variables:

   ```
   HOST <FQDN or IP Address of LDAP host>
   PORT <LDAP host port number>
   ```

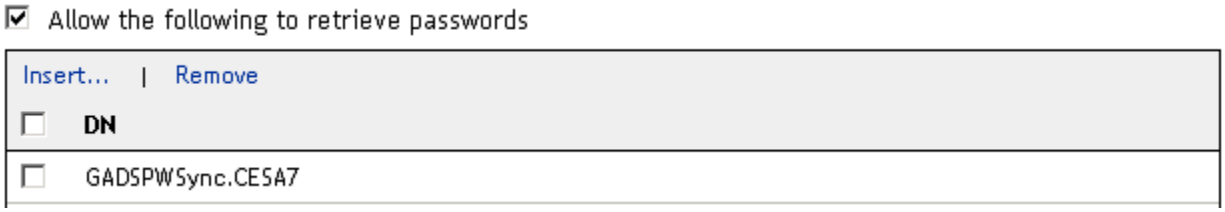
   Conditional: If the LDAP host requires secure bind (ldaps), export the appropriate certificate from eDirectory as a .b64 certificate file and place it somewhere on the GADS server. Set the following variables:

   ```
   TLS_REQCERT demand
   TLS_CACERT <Full path to .b64 certificate>
   ```

6. Create an eDirectory user (GADSPWSync in this example) and assign it a password. Assign this user the following rights at the tree level:

   NOTE: The carLicense eDirectory attribute will be used to store the users' passwords in a hashed format supported by Google Apps Directory Sync. A different unused attribute may be used if more convenient. The carLicense attribute will be used throughout the documentation.

7. Edit the Universal Password policy assigned to the users granting the GADSPWSync user the right to retrieve users' passwords:



8. Extract the gadspwsync script and its supporting files to a directory on the GADS server (`/gadspwsync` for example).

9. Edit the `/gadspwsync/contexts.txt` file. List the contexts to be searched for users listing one context per line. Contexts should be listed in LDAP format.

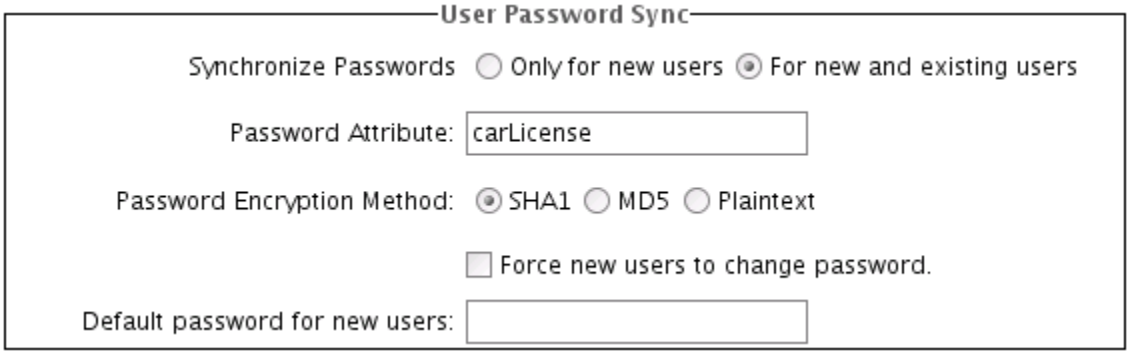
10. Edit the `/gadspwsync/gadspwsync.sh` script file. Adjust the following variables to suite the environment:

    - `SCRIPTPATH` – Path to the script
    - `CONTEXTSFILE` – File, including path, listing eDirectory contexts to search
    - `LDAPSCOPE` – Specify "one" or "sub" to search sub OUs or not
    - `LDAPHOST` – FQDN or IP address of LDAP server
    - `LDAPURI` – LDAP URI to LDAP server (ldap://LDAPserver or ldaps://LDAPserver)
    - `LDAPBINDDN` – Username, including context, for GADSPWSync user
    - `LDAPPASSWD` – GADSPWSync user password
    - `GETPASS` – Location of Getpass 2.1 Cool Tool
    - `LDAPATTRIB` – eDirectory attribute used to store hashed passwords for GADS
    - `GADSCMD` – Full path to the GADS sync-cmd
    - `GADSCONF` – Full path to GADS configuration file

11. Set the permissions on the `/gadspwsync/gadspwsync.sh` script file so that only the root user can read the file. From the terminal prompt:

```
chown root:root /gadspwsync/gadspwsync.sh
chmod 700 /gadspwsync/gadspwsync.sh
```

12. Configure Google Apps Directory Sync per Google's documentation. Set the Password Attribute field to the selected eDirectory attribute for storing hashed passwords (carLicense for example).



13. Schedule gadspwsync.sh to run on a scheduled basis to synchronize with Google. Because gadspwsync.sh calls GADS at the end of the script it is not necessary to call GADS separately. Edit the /etc/crontab file and add a similar entry (example runs daily at 3:30am):

```
30 3 * * *   root  /gadspwsync/gadspwsync.sh  >/dev/null 2>&1
```

Running Multiple GADS Configuration Files

If you have the need to run multiple GADS configuration files, locate the following lines at the end of the script:

```
# Exit script and run Google Apps Directory Sync
exit & $GADSCMD -a -c $GADSCONF
```

Replace the above lines with something similar to match your environment:

```
# Run Google Apps Directory Sync for teachers
$GADSCMD –a –c /opt/GoogleAppsDirSync/teachers.xml
sleep 30

#Exit script and run Google Apps Directory Sync for students
exit & $GADSCMD -a –o -c /opt/GoogleAppsDirSync/students.xml
```

Delete lines 35 & 36 near the start of the script:

```
# Full path to GADS configuration file
GADSCONF="/gadspwsync/DigitalAirlines.xml"
```

Thank you to Matt Schlawin for suggesting adding the sleep command to the multiple GADS configuration files tweak and providing benchmarks!

If you have any questions or problems with the script please contact Brad Rodgers at brad@rodgeville.com. Thank you to Matt Schlawin, Scott Ripley, Linda Currie, Shane Farmer, and Debbie Blakeney for testing the script! Thank you to the Northeast Wisconsin Novell User Group for reviewing the script and providing valuable feedback!