Splunk® Supporting Add-on for Active Directory Deploy and Use the Splunk Supporting Add-on for Active Directory (SA-LDAPSearch) 3.0.0

Generated: 2/02/2020 9:30 am
# Table of Contents

**Introduction**

- About the Splunk Supporting Add-on for Active Directory ........................................ 1
- New to Splunk? ........................................................................................................ 2
- How this add-on fits into the Splunk picture .......................................................... 5
- How to get support and find more information about Splunk Enterprise ............ 6

**Before you install**

- Platform and hardware requirements ...................................................................... 7

**Deploy and configure**

- Install the Splunk Supporting Add-on for Active Directory .................................... 10
- Configure the Splunk Supporting Add-on for Active Directory .............................. 13
- Upgrade the Splunk Supporting Add-on for Active Directory (SA-LDAPSearch) .... 19

**The LDAP Search commands**

- The ldapsearch command ..................................................................................... 20
- The ldapfilter command ........................................................................................ 22
- The ldapfetch command ....................................................................................... 24
- The ldapgroup command ....................................................................................... 26
- The ldaptestconnection command ........................................................................ 27

**Troubleshoot SA-ldapsearch**

- Troubleshoot the Splunk Supporting Add-on for Active Directory ....................... 29

**Reference**

- The ldap.conf configuration file ............................................................................ 35
- Data and source types for the Splunk Supporting Add-on for Active Directory .... 38

**Release Notes**

- Release Notes for Splunk Supporting Add-on for Active Directory .................... 39
- Workaround for default configuration stanza errors in distributed environments .... 40
- Third-party software attributions/credits ............................................................... 42
Introduction

About the Splunk Supporting Add-on for Active Directory

<table>
<thead>
<tr>
<th>App Version</th>
<th>3.0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor products</td>
<td>Windows Server Active Directory (LDAP) services</td>
</tr>
</tbody>
</table>

The Splunk Supporting Add-on for Active Directory lets you collect Active Directory schema and other information from Active Directory as events and filter on those events.

How does it work?

The Splunk Supporting Add-on for Active Directory has the following uses:

- To generate events based on the contents of an LDAP server such as Active Directory.
- To augment events with information from an LDAP server such as Active Directory.
- To perform Active Directory group expansions.

In order to use the Splunk Supporting Add-on for Active Directory, you must configure it. Read "Install the Splunk Supporting Add-on for Active Directory" to learn how.

How do I get it?

You can download the add-on from Splunkbase.

How do I upgrade from a previous version?

To upgrade from a previous version of the Splunk Supporting Add-on for Active Directory, rename ldap3 folder located in /apps/SA-ldapsearch/bin/packages directory to ldap_old on search head and then install this version directly on top of the previous version. You can use Splunk Web or the CLI, or you can upgrade it from the command line. In this version we have updated ldap3 package from v 0.9.5 to v2.5 to solve performance issues.
Note: If you have a previous version installed, on upgrade, the app maintains ldap.conf from the previous installation. When you add any new domains to search, the add-on stores the credentials securely, instead of in the ldap.conf file. If you edit an existing domain entry using the new Configuration page, it also stores those credentials securely. Existing entries that you do not edit continue to have their credentials stored locally.

What search commands come with it?

There are four search commands and one test command in this add-on. Once configured, the add-on uses the configuration for all the commands. You can learn more about the commands in the following topics:

- The "ldapsearch" command
- The "ldapfilter" command
- The "ldapfetch" command
- The "ldapgroup" command
- The "ldaptestconnection" command

Where can I ask questions and get help?

You can visit Splunk Answers to discuss and get help on the Splunk Supporting Add-on for Active Directory. See "How to get support and find out more information about Splunk Enterprise" for additional support options.

New to Splunk?

If this is the first time you have used Splunk, then read on...this topic introduces the most important Splunk concepts you need to understand when installing and using Splunk apps and add-ons.

Splunk and Splunk apps work together.

The key points to come away with are:

- All Splunk apps run on the Splunk platform.
- Understanding how Splunk works will greatly help you understand how Splunk apps work.
- Installing and configuring the app is only part of the experience - you might need to prepare Splunk before installing your app.
- Careful planning helps achieve a successful app deployment experience.
Splunk basics

Splunk is a software platform that accepts data from many different sources, such as files or network streams. Splunk stores a unique copy of this data in what's called an **index**. Once the data is there, you can connect to Splunk with your web browser and run searches across that data. You can even make reports or graphs on the data, right from within the browser.

You can extend Splunk's capability by installing apps. Splunk apps come with searches, reports, and graphs about specific products that are common to most IT departments. These searches, reports, and graphs reduce the amount of time it takes to glean real value from installing and running the Splunk platform.

Before you can really understand how Splunk apps work, you should understand how Splunk works. Fortunately, we've got you covered in that respect.

If you're new to Splunk, then the best place to learn more about it is in the Search Tutorial. It helps you learn what Splunk is and what it does, as well as what you need to run it and get step-by-step walk-throughs on how to set it up, get data into it, search with it, and create reports and dashboards on it.

**Licensing**

The next thing you want to learn about is Splunk's licensing model. Splunk charges you based on the amount of data you index. The licensing introduction from the Admin Manual is a great place to start learning about how licenses work. You can also find out the types of licenses that are available, how to install, remove, and manage them, and what happens when you go over your license quota.

In the context of Splunk apps, the amount of licensing capacity you need depends on how each app defines the individual data **inputs** that it uses. Splunk apps use inputs to tell Splunk what data it needs to collect for the app's purpose. Some apps, such as the Splunk App for Enterprise Security, collect a lot of data, which your license must cover in order for you to be able to search that data without interruption. When planning for your app, make sure you include enough licensing capacity.

**Configuration**

Much of Splunk's extensibility is in how configurable it is. You must configure Splunk before it can collect data and extract knowledge. All Splunk apps use configuration files to determine how to collect, transform, display, and provide
alerts for data. The Admin Manual shows you how to configure those files and includes a reference topic for each configuration file that Splunk uses. In some cases, you can also use Splunk Web or the CLI to make changes to a Splunk app's configuration.

Splunk also uses configuration files to configure itself. When Splunk initializes, it finds all of the configuration files located in the Splunk directory and merges them to build a final "master" configuration, which it then runs on. When you install a Splunk app on a Splunk instance, Splunk must determine which configuration files to use if it encounters a conflict. This is where configuration file precedence comes in.

It's important to understand how precedence works. In many cases, if there is a configuration file conflict, Splunk gives priority to an app's configuration file. In some situations, installing an app might inadvertently override a setting in a configuration file in the core platform, which might lead to undesired results in data collection. Be sure to read the previously mentioned topic thoroughly for details.

**Splunk Search**

Splunk provides the ability to look through all the data it indexes and create dashboards, reports, and even alerts. All Splunk apps rely on Splunk search, so it's a good idea to read the overview on search in the *Search Manual* to learn how powerful Splunk's search engine is (the Tutorial is also a good place to learn about Splunk search.)

You should also have an understanding of the Splunk search language. Splunk apps use the search language extensively to put together search results and knowledge objects which drive their dashboards, reports, charts, and tables.

Finally, it's a good idea to familiarize yourself with the search commands in the *Search Reference*. That manual describes the commands that both Splunk and your Splunk app can use.

**Sources and source types**

When Splunk indexes data, it does so from a source - an entity that provides data for Splunk to extract, for example, Windows event logs, or *nix syslogs. Splunk tags incoming data with a "source" field as it gets indexed. The source type is an indicator for the type of data, so that Splunk knows how to properly format and extract it as it comes in. It's also - conveniently enough - a way to categorize data, as you can use Splunk search to display all data of a certain
source type.

Splunk apps use sources and source types to extract knowledge from the data they index. Many views in an application depend on searches with specific sources and source types defined in them. Splunk apps sometimes use the source types that come with Splunk, and sometimes they define their own.

**Capacity planning and distributed Splunk**

Another important factor to consider when using a Splunk app: Do you have enough hardware to realistically support a deployment for the Splunk app you're using? Read our capacity planning documentation for a head-start on ensuring you have the machinery in place to run your Splunk app deployment at peak performance.

Learning about capacity planning is a perfect time to introduce another concept with which you should be familiar: distributed search. Nearly every Splunk app available can use distributed search, and many were developed with distributed search in mind. What this means is that you must working with multiple Splunk instances at once - with each instance playing a specific role - to use the app to its full potential. Initially, you add indexers to increase indexing performance, then you add search heads to increase search performance. The Distributed Deployment Manual provides details on how to add more Splunk instances to keep up with your app's performance demands.

**What's next?**

From this point, you are ready to plan your app deployment. Continue reading for information about how this app fits into the Splunk picture, platform and hardware requirements, and other deployment considerations.

**How this add-on fits into the Splunk picture**

The Splunk Supporting Add-on for Active Directory is one of a variety of apps and add-ons available within the Splunk ecosystem. All Splunk apps and add-ons run on top of a core Splunk Enterprise installation. You must install Splunk Enterprise (full or universal forwarders) first, and then install the Splunk Supporting Add-on for Active Directory.

- For specifics about what where to install things, read "Install the Splunk Supporting Add-on for Active Directory" in this manual.
• For details about apps and add-ons, refer to "What are apps and add-ons?" in the core Splunk Enterprise product documentation.
• To download Splunk Enterprise, visit the download page on splunk.com.
• To get more apps and add-ons, visit Splunk Apps.

How to get support and find more information about Splunk Enterprise

If you have questions about the Splunk Supporting Add-on for Active Directory and you have an enterprise support contract, log a case via the Splunk Support Portal.

If your Splunk Enterprise deployment is large or complex, you might want to engage a member of the Splunk Professional Services team to assist you.

Find more information about Splunk

If you don’t have an enterprise support contract or want to learn more about Splunk, use the following options:

• Splunk Answers
• The #splunk IRC channel on EFNET

You can also read the core Splunk documentation.
Before you install

Platform and hardware requirements

This topic discusses the underlying requirements for running the Splunk Supporting Add-on for Active Directory.

Hardware and Operating System requirements

Hardware requirements

The Splunk Supporting Add-on for Active Directory has memory, CPU, and disk requirements that meet standard hardware requirements for the core Splunk Enterprise platform. Deploy hardware that meets or exceeds these hardware requirements.

- For additional details about Splunk Enterprise system requirements, see "System requirements" in the core Splunk Enterprise documentation.
- For information about estimating hardware requirements for a Splunk deployment, see "Introduction to capacity planning for Splunk Enterprise" in the Capacity Planning Manual.

Operating system requirements

You can install the add-on on Splunk Enterprise instances that run a supported operating system. See the list of supported Windows and *nix operating systems.

What versions of Splunk does the add-on support?

All Splunk Enterprise search heads require Splunk Enterprise version 7.2 or later

What versions of Active Directory does the add-on support?

The Splunk Supporting Add-on for Active Directory supports the following versions of Active Directory:

- Microsoft Windows Server 2008 Active Directory Domain Services
- Microsoft Windows Server 2008 R2 Active Directory Domain Services
- Microsoft Windows Server 2012 Active Directory Domain Services
- Microsoft Windows Server 2012 R2 Active Directory Domain Services
- Microsoft Windows Server 2016 Active Directory Domain Services

The add-on does not support AD Lightweight Directory Services (AD LDS) or other Lightweight Directory Access Protocol (LDAP) server types.

**Distributed installation of this add-on**

This table provides a quick reference for installing this add-on onto a distributed deployment of Splunk Enterprise.

<table>
<thead>
<tr>
<th>Splunk instance type</th>
<th>Supported</th>
<th>Required</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Heads</td>
<td>Yes</td>
<td>Yes</td>
<td>The host must have access to the domain controller for the domain or forest you want to get events from. The configurations you make must be identical across the search head and all search peers.</td>
</tr>
<tr>
<td>Indexers</td>
<td>On search peers only</td>
<td>Depends</td>
<td>If the indexer acts as a search peer, then you must install it on all indexers that act as search peers. The search peers must have access to the domain controller for the domain or forest you want to get events from. Additionally, the configurations you make must be identical across the search head and all other search peers.</td>
</tr>
<tr>
<td>Heavy Forwarders</td>
<td>Yes</td>
<td>No</td>
<td>In this configuration, you can route events from the add-on to other Splunk Enterprise instances based on the target index, or filter the data to extract only the events you want.</td>
</tr>
<tr>
<td>Universal Forwarders</td>
<td>No</td>
<td>No</td>
<td>The add-on does not perform any function when you install it on this type of Splunk instance.</td>
</tr>
<tr>
<td>Light Forwarders</td>
<td>No</td>
<td>No</td>
<td>The add-on does not perform any function when you install it on this type of Splunk instance. Also, light forwarder functionality has been deprecated and</td>
</tr>
</tbody>
</table>
could be removed in a future version of the Splunk software.

**Distributed deployment compatibility**

This table provides a quick reference for the compatibility of this add-on with Splunk distributed deployment features.

<table>
<thead>
<tr>
<th>Distributed deployment feature</th>
<th>Supported</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Head Clusters</td>
<td>Yes</td>
<td>Configure your search head cluster first, then perform an installation of the add-on. The cluster replicates the configurations.</td>
</tr>
<tr>
<td>Indexer Clusters</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Deployment Server</td>
<td>Yes</td>
<td>You can deploy the add-on to search heads.</td>
</tr>
</tbody>
</table>

**What are the other prerequisites?**

*The 'admin_all_objects' Splunk account capability*

The Splunk Supporting Add-on for Active Directory requires the `admin_all_objects` capability to read storage passwords. The `admin` user has this capability by default. If you do not want to use the `admin` user, then any user you do use must have this capability added to its profile.

To learn more about Splunk users and assigning capabilities, see "About configuring role-based user access" in the core Splunk Enterprise platform documentation.
Deploy and configure

Install the Splunk Supporting Add-on for Active Directory

This topic provides instruction on how to install the Splunk Supporting Add-on for Active Directory.

Where to install it

The Splunk Supporting Add-on for Active Directory is designed to be installed across a distributed Splunk platform deployment. It can be installed on:

- Search heads
- Search peers (indexers) when you want to distribute LDAP queries across those peers. Like the search head, the search peers must have access to Active Directory for this to work. See Install SA-LDAPsearch on the search head and all search peers in this manual for details.
- Heavy forwarders. The Splunk Supporting Add-on for Active Directory does not perform any function when you install it on a universal or light forwarder.

Distributed deployment

<table>
<thead>
<tr>
<th>Search head</th>
<th>Search peers (Indexers)</th>
<th>Heavy Forwarder</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Standalone deployment

<table>
<thead>
<tr>
<th>Search head</th>
<th>Search peers (Indexers)</th>
<th>Heavy Forwarder</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In a distributed deployment, install SA-LDAPSearch on the indexer to avoid the following errors:

External search command 'ldapfilter' returned error code 1. Script output = " ERROR The default configuration stanza for ldap.conf is missing."
The default configuration stanza for ldap.conf is missing: HTTP 404 Not Found - Application does not exist: SA-ldapsearch

**How to install it**

In most situations, you can download and install the add-on by using either Splunkbase or the CLI.

Once you install it, you must then **configure it**.

**Install the add-on from the command line**

On Splunk Enterprise, you can install the add-on from the command line, using the CLI.

To install the Splunk Add-on for Windows from the command line:

1. Download the Splunk Supporting Add-on for Active Directory from Splunk Apps, if you haven't already.

   **Note:** If you have access to the Internet and have a valid link to where the app package resides, you can use the `splunk install` command to install the app directly from the Internet:

   ```
   > cd Program Files\Splunk\bin
   > .\splunk install
   http://server.com:80/files/splunk-support-for-active-directory-xxxx.tar.gz
   In this case, you can then proceed to Step 3.
   ```

2. Run the `splunk install` CLI command:

   ```
   > cd Program Files\splunk\bin
   > .\splunk install app
   <path>\splunk-support-for-active-directory-xxxx.tar.gz
   App 'sa-ldapsearch' is installed.
   **Note:** You might have to log into your Splunk Enterprise instance before it installs the app.

3. Configure the Splunk Supporting Add-on for Active Directory.
Install the add-on using Splunkbase

Install the Splunk Supporting add-on for Active Directory only on full instances of Splunk Enterprise. The most common use case for this method of installation is to provide support for another app installed on the same machine. The add-on is not available for installation on universal forwarders or light forwarders.

To install the Splunk Supporting Add-on for Active Directory:

1. Download the Splunk Add-on for Windows from Splunkbase, if you haven't already.
   The file downloads with a .tar.gz extension. Do not run this file.

2. Log into Splunk Web on the Splunk Enterprise instance on which you want to install the app.
3. Once logged in, click 'App from the menu bar.
4. Click Manage apps...
5. On the next page, click the Install app from file button.
6. On the upload screen, click Browse...
7. Select the downloaded splunk-support-for-active-directory-xxxx.tar.gz file.
8. Click Open.
9. Click Upload.
   Splunk Enterprise opens the splunk-support-for-active-directory-xxxx.tar.gz package and installs the application.
10. Click the Restart Splunk button or the link in the banner to restart Splunk.
11. A dialog box asking you if you are sure you want to restart Splunk may appear. Click OK to restart Splunk.
12. Once Splunk restarts, click OK to return to the Splunk login page.
13. Configure the Splunk Supporting Add-on for Active Directory.

Install the Splunk Supporting Add-on for Active Directory on the deployer

1. In a web browser, proceed to the Splunk Supporting Add-on for Active Directory.
2. Click the download link to begin the download process. You might need to sign in with your Splunk account before the download starts.
3. When prompted, choose an accessible location on your deployment server to save the download. Do not attempt to run the download.
4. Use an archive utility such as WinZip or tar to unarchive the file to the
%SPLUNK_HOME%/etc/apps directory on the deployer.

Configure the Splunk Supporting Add-on for Active Directory

The Splunk Supporting Add-on for Active Directory is a bundle of commands
written in Python. New for version 2.0, the add-on no longer requires an
installation of Java on the machines that run it.

Configure the Splunk Supporting Add-on for Active Directory

Use Splunk Web to configure the Splunk Supporting Add-on for Active Directory.

If you have upgraded from a previous version, you must visit the Configuration
page once to save your existing configuration in the new format. From then on,
make any changes in this page.

If you have not used the add-on, use the Configuration page to add and remove
configurations.

Configure the add-on with Splunk Web

The Splunk Supporting Add-on for Active Directory has a configuration page that
you can access from Splunk Web.
To use the configuration page, activate the add-on by selecting it from the "App" menu in the upper left corner of the screen. Then, after the add-on loads, select "Configuration" from the menu.

**Add a domain**

To add a domain using the Configuration page:

1. In the lower left corner of the domain list pane, click the "+" sign.

2. In the **Domain Name** field, type in the name of the domain that you want the add-on to get data for.

3. In the **Alternate domain name** field, type in an alternate representation of the domain in NetBIOS format. Make sure that the **Alternate domain name** is specified in UPPERCASE format.
   
   **Example:** SPLUNK

4. In the **Base DN** field, type in the same domain in LDAP notation.
   
   **Example:** DC=spl,DC=com

5. In the **LDAP Server: Hostname** field, type in the name or IP address of the host that the add-on should connect to for this domain.

6. In the **LDAP Server: Port** field, type in the port that the add-on should connect to on the LDAP server.

   **Note:** When configuring the 'default' connection, use port 3268 (for plain text) or 3269 (for SSL) connections. This tells the add-on to connect to the Global Catalog, which is faster, but might not return all attributes. For other domain connections, use port 389 (for plain text) or 636 (for SSL)

7. If you want the server to use SSL to connect, click the **SSL** checkbox.

8. In the **Credentials: Bind DN** field, enter the username that the add-on should use to connect to the LDAP server you specified previously, in LDAP notation.

   **Example:** CN=Splunk Searcher,CN=Users,DC=spl,DC=com

   **Note:** You can also enter the name in the following formats:
• User Principal Name (UPN), such as 'user@splunk.com'
• Security Account Manager (SAM) account name, such as 'splunk\username'
• Base Distinguished Name (DN).

9. In the **Credentials: Password** field, enter the password for that user.

10. Optionally, you can test whether or not the add-on can make a valid connection. To do so, click the **Test connection** button.

   A window appears while the add-on attempts to connect to the LDAP server and retrieve information. If the test succeeds, the window displays results. If no results display, then the test has failed and you must correct your settings before attempting to test the connection again.

11. Click **Save** to save your changes.

**Remove a domain**

To remove a domain:

1. In the domain list pane, click the domain that you want to remove.

2. Click the "-" button at the bottom of the pane.

3. Click the **Save** button to save the changes.

**Configure the add-on for use on a heavy forwarder**

Though the add-on is not required on the heavy forwarder, configure the add-on on the forwarding instance to run a saved search. For example, you can run the LDAP query as a saved search to get continuous data of Active Directory.

**1. Configure the receiving instance**

To configure forwarding and receiving, see Enable forwarding on a Splunk Enterprise instance and Enable a receiver in the Splunk Enterprise *Forwarding Data* manual.

1. Configure receiving on the host(s) that you want to receive the forwarded data.
2. Create an index for the forwarded data to reside in.
2. Configure SA-LDAPsearch on the forwarding instance

1. Configure forwarding on the instance that runs the add-on.
2. Configure the add-on to connect to an Active Directory domain controller or Global Catalog. See Configure the Splunk Supporting Add-on for Active Directory.
3. On the forwarder, create an index with the same name as the receiving index in step 2 of Configure the receiving instance.
4. (Optional) To forward data to a particular destination index, see Filter data by target index in the Splunk Enterprise Forwarding Data manual.

3. Run and save the search

1. On the heavy forwarder, navigate to the Search & Reporting app.
2. Type in a valid command(s) for the add-on, but do not invoke the search yet.
   See The LDAP search commands chapter for proper syntax for the LDAP search commands.
3. At the end of the search, add | collect index=<index>, where <index> is the index that you created.
4. Run the search.
5. Save the search as a report.
6. Give the report a name and a description, then click Save.
8. Confirm that you see events on the receiving indexer.

Where to find the events on the receiver

The events that you forward appear in the index that you created on the receiving indexer with the host field set to the name of the host that originated the LDAP search and a source type of stash. You can search the index that you created on the receiving instance to see all of the forwarded events.

If you do not see the events on the receiving indexer, confirm that:

- You have created an index on both the heavy forwarder and the receiving indexer, and that the index names match.
- The forwarding instance is a heavy forwarder. A universal forwarder cannot route data based on destination index.
- You have configured the forwardedindex filter to forward data destined for the index you created.
You pipe the results of your LDAP search to the `collect` command and specify the index you created.

**Configure the add-on to send data to Splunk Cloud**

When you configure the Splunk Supporting Add-on for Active Directory for use with Splunk Cloud, configure a heavy forwarder to send data to the Splunk Cloud instance. This requires installing the add-on on a heavy forwarder that is configured to forward data to that Splunk Cloud instance. The procedure is similar to configuring the add-on to use a heavy forwarder, but there are additional steps required to get the Splunk Cloud instance to accept data from the forwarder. See Overview of getting data into Splunk Cloud in the *Splunk Cloud User Manual*.

**Configure Secure Sockets Layer (SSL) settings for the add-on with ssl.conf**

You can use the `ssl.conf` file to override SSL settings on the Splunk Enterprise instance that runs the add-on.

The settings in `ssl.conf` apply only to the Splunk Supporting Add-on for Active Directory and override settings that you have defined in `server.conf` on the instance. You can specify any valid attributes within the `[sslconfig]` stanza of `server.conf`. See the `server.conf` spec file. A list of common attributes follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>caCertFile = &lt;filename&gt;</td>
<td>File name (relative to 'caPath') of the root CA (Certificate Authority) certificate store. Must refer to a PEM format file containing one or more root CA certificates concatenated together.</td>
<td>cacert.pem</td>
</tr>
<tr>
<td>sslVersions = &lt;versions_list&gt;</td>
<td>A comma-separated list of SSL versions to support for incoming connections. The versions available are &quot;ssl3&quot;, &quot;tls1.0&quot;, &quot;tls1.1&quot;, and &quot;tls1.2&quot;. The special version &quot;*&quot; selects all supported versions. The version &quot;tls&quot; selects all versions tls1.0 or newer. If you prefix a version is with &quot;.&quot; you remove it from the list.</td>
<td>*,−ssl2 (anything newer than SSLv2).</td>
</tr>
</tbody>
</table>
SSLv2 is always disabled; "-ssl2" is accepted in the version list but does nothing. When configured in FIPS mode, ssl3 is always disabled regardless of this configuration.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>sslVerifyServerCert</td>
<td>Used by distributed search: when making a search request to another server in the search cluster. Used by distributed deployment clients when polling a deployment server.</td>
<td>false</td>
</tr>
<tr>
<td></td>
<td>If you set this to true, you should make sure that the server that is being connected to is a valid one (authenticated). Both the common name and the alternate name of the server are then checked for a match if they are specified in this configuration file. A certificate is considered verified if either name matches.</td>
<td></td>
</tr>
</tbody>
</table>

1. Open a shell or command prompt.

2. Change to the `$SPLUNK_HOME/etc/apps/SA-ldapsearch/local` directory.

3. Create the `ssl.conf` file.


5. Add valid key-value pairs to the file.

6. Save the file and close it.

7. Log into Splunk Web on the instance that runs the add-on.

8. Configure the add-on to add an LDAP domain. See Configure the add-on with Splunk Web.

**Other uses**

You can also create lookup tables with the report you save, depending on the LDAP commands you use. Lookup tables cannot be forwarded, however. To work around this limitation, you can:
• Copy the lookup table to the lookup directory on a search head or indexer.
• Import the lookup table into App Key Value Store.

Upgrade the Splunk Supporting Add-on for Active Directory (SA-LDAPSearch)

Follow these instructions to upgrade the Splunk Supporting Add-on for Active Directory (SA-LDAPSearch) from v2.2.x using Splunkbase.

1. Download the updated version of the app from Splunkbase.
2. Unpack the archive.
3. Copy the Splunk Supporting Add-on for Active Directory folder to the %SPLUNK_HOME%/etc/apps folder on the search head(s) in your deployment.
   If prompted, overwrite the existing folder.
   The Splunk Supporting Add-on for Active Directory must be installed on all search heads and indexers in the deployment.
4. Restart Splunk Enterprise on the search head(s).
5. Restart the deployment server.
6. Log into Splunk Enterprise.
7. Choose Splunk Supporting Add-on for Active Directory from the list of apps.
The LDAP Search commands

The ldapsearch command

Overview

The ldapsearch command retrieves results from the specified search from the configured domains and generates events. It must be at the beginning of a search pipeline. A sample usage follows:

```
| ldapsearch domain=SPL search="(objectClass=user)"
```

There are several possible arguments for `ldapsearch`:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain=&lt;domain&gt;</td>
<td>Specifies the name of a configuration stanza in ldap.conf. If you do not specify a domain, the command uses the default stanza.</td>
</tr>
<tr>
<td>search=&lt;search-filter&gt;</td>
<td>Specifies the RFC 2254-compliant search string.</td>
</tr>
<tr>
<td>attrs=&lt;attribute-names&gt;</td>
<td>Specifies a comma-delimited list of attributes to return as fields.</td>
</tr>
<tr>
<td>debug=&lt;boolean&gt;</td>
<td>Specifies whether or not ldapsearch should write debug log data. When set to T, specifies that debug logging should occur.</td>
</tr>
<tr>
<td>limit=&lt;size-limit&gt;</td>
<td>Specifies that only a certain number of entries should be returned. This argument is optional and defaults to all entries.</td>
</tr>
<tr>
<td>basedn=&lt;search-base&gt;</td>
<td>Specifies a search base as the starting point instead of the default as set in the ldap.conf file.</td>
</tr>
<tr>
<td>scope=&lt;base</td>
<td>one</td>
</tr>
<tr>
<td>logging_level=(CRITICAL</td>
<td>ERROR</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>File:</td>
<td>Full pathname of the source file where the logging call was made.</td>
</tr>
<tr>
<td>Level:</td>
<td>Level of the logging call that was made; one of CRITICAL, ERROR, WARNING,</td>
</tr>
<tr>
<td></td>
<td>DEBUG.</td>
</tr>
<tr>
<td>Line:</td>
<td>Line number in the source file where the logging call was made.</td>
</tr>
<tr>
<td>Pid:</td>
<td>ID of the process that made the logging call.</td>
</tr>
<tr>
<td>log_source:</td>
<td>String of the form &quot;Pid=&lt;Pid&gt;, File=&lt;File&gt;, Line=&lt;Line&gt;&quot;.</td>
</tr>
<tr>
<td>message:</td>
<td>Full text of the logged message.</td>
</tr>
</tbody>
</table>

ldapsearch writes its debug logs to $SPLUNK_HOME/var/log/splunk/SA-ldapsearch.log. Splunk indexes and rotates this file by default.

On return, the raw value of each event is the LDAP Data Interchange Format (LDIF) representation of the record. In addition, the command returns a number of attributes as values. When the command returns attributes as values, it decodes those values on the fly. For instance, the command represents a globally unique ID (GUID) or security ID (SID) in human-readable form, even though the LDIF form in the event itself remains in base-64-encoded format. By default, the command returns all user attributes.

**Examples**

See Create a lookup from your current LDAP data in Splunk Enterprise Security for more details.

To get a table suitable for pushing into the identities.csv file for the Splunk App for Enterprise Security:

```bash
| ldapsearch domain=SPL search="(&(objectclass=user)(!(objectClass=computer)))" | search
userAccountControl="NORMAL_ACCOUNT" | eval suffix="" | eval
priority="medium" | eval category="normal" | eval watchlist="false" | eval endDate="" | table sAMAccountName, personalTitle, displayName, givenName, sn, suffix, mail, telephoneNumber, mobile, manager, priority, department, category, watchlist, whenCreated, endDate | rename
sAMAccountName AS identity, personalTitle AS prefix, displayName AS
nick, givenName AS first, sn AS last, mail AS email, telephoneNumber AS
phone, mobile AS phone2, manager AS managedBy, department as bunit,
whenCreated AS startDate
```
If you have multiple domains, you can run multiple `ldapsearch` commands that are joined using the `append` command. For example:

```
| ldapsearch domain=SPL search="(&(objectclass=user) (!(objectClass=computer)))" | search userAccountControl="NORMAL_ACCOUNT" | eval suffix="" | eval priority="medium" | eval category="normal" | eval watchlist="false" | eval endDate="" | eval identity = "SPL\" + sAMAccountName | table identity, personalTitle, displayName, givenName, sn, suffix, mail, telephoneNumber, mobile, manager, priority, department, category, watchlist, whenCreated, endDate | rename personalTitle AS prefix, displayName AS nick, givenName AS first, sn AS last, mail AS email, telephoneNumber AS phone, mobile AS phone2, manager AS managedBy, department AS bunit, whenCreated AS startDate | append [ ldapsearch domain=ENG search="(&(objectclass=user) (!(objectClass=computer)))" | search userAccountControl="NORMAL_ACCOUNT" | eval suffix="" | eval priority="medium" | eval category="normal" | eval watchlist="false" | eval endDate="" | eval identity = "ENG\" + sAMAccountName | table identity, personalTitle, displayName, givenName, sn, suffix, mail, telephoneNumber, mobile, manager, priority, department, category, watchlist, whenCreated, endDate | rename personalTitle AS prefix, displayName AS nick, givenName AS first, sn AS last, mail AS email, telephoneNumber AS phone, mobile AS phone2, manager AS managedBy, department AS bunit, whenCreated AS startDate ]
```

### The ldapfilter command

#### Overview

The `ldapfilter` command filters and augments events with information from Active Directory. It follows a 'search' or similar command in the pipeline so you can feed it events. A sample usage follows

```bash
eventtype=msad-successful-user-logons | ldapfilter domain=$dest_nt_domain$ search="(objectClass=$src_user$)" attrs="telephoneNumber,displayName"
```

There are several possible arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>search=&lt;search-filter&gt;</code></td>
<td>Specifies the LDAP or Active Directory search.</td>
</tr>
<tr>
<td><code>domain=&lt;domain&gt;</code></td>
<td>Specifies the LDAP or Active Directory search.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>search=&lt;search-filter&gt;</td>
<td>Specifies the RFC 2254-compliant search string.</td>
</tr>
<tr>
<td>basedn=&lt;search-base&gt;</td>
<td>Specifies a search base as the start of the default as set in the ldap configuration stanza identified by domain.</td>
</tr>
<tr>
<td>attrs=&lt;attribute-names&gt;</td>
<td>Specifies a comma-delimited list of attributes to return as fields.</td>
</tr>
<tr>
<td>debug=&lt;boolean&gt;</td>
<td>Specifies whether or not ldapfilter should write debug log data. When set to T, specifies that debug logging should occur.</td>
</tr>
<tr>
<td>logging_level=(CRITICAL</td>
<td>ERROR</td>
</tr>
</tbody>
</table>

ldapfilter writes its debug logs to $SPLUNK_HOME/var/log/splunk/SA-ldapsearch.log. Splunk indexes and rotates this file by default.

On return, ldapfilter adds the LDAP attributes specified by the attrs argument to each event based on an LDAP search. It uses the domain and search fields to determine the LDAP search to perform. You can substitute variables by surrounding field names with dollar signs. For example, $src_user$

**Examples**

To return the display name and telephone number for all successful logons:
search eventtype=msad-successful-user-logons |stats count by src_user,dest_nt_domain |ldapfilter domain=$dest_nt_domain$ search="(sAMAccountName=$src_user$)" attrs="telephoneNumber,displayName" |table src_user,dest_nt_domain,displayName,telephoneNumber,count

To print a list of all Organization Units (OU) that have linked GPOs and fetch the displayName of the GPO:

|ldapsearch domain=MYDOMAIN search="(objectclass=organizationalUnit)" attrs="ou,description,gPLink,gPOptions" |sort ou |rex field=gPLink max_match=10 "\[LDAP://(?<gpo>[^;]+);d+\]" |ldapfilter debug=T domain="{ctx}$gpo$" search="(distinguishedName=$gpo$)" attrs="cn,displayName" |table ou,cn,displayName

Important: You must specify the search, domain and attrs fields for ldapfilter to work properly.

The ldapfetch command

Overview

The 'ldapfetch' command filters and augments events with information from Active Directory. It follows a 'search' or similar command in the pipeline so you can feed it events. A sample usage follows:

|ldapsearch domain="SPL" search="(objectclass=group)" attrs="memberOf" |mvexpand memberOf |ldapfetch dn=memberOf attrs="cn,description"

There are several possible arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dn=&lt;field-name&gt;</td>
<td>Specifies the field that holds the Distinguished Name (DN) to fetch.</td>
</tr>
<tr>
<td>attrs=&lt;attribute-names&gt;</td>
<td>Specifies a comma-delimited list of attributes to return as fields.</td>
</tr>
<tr>
<td>domain=&lt;domain&gt;</td>
<td>Specifies the name of a configuration stanza in ldap.conf. If you do not specify a domain, the command uses the default stanza.</td>
</tr>
<tr>
<td>debug=&lt;boolean&gt;</td>
<td>Specifies whether or not ldapfetch should write debug log data. When set to T, specifies that debug logs should be written.</td>
</tr>
</tbody>
</table>
### Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>logging_level</td>
<td>logging should occur.</td>
</tr>
<tr>
<td></td>
<td>Specifies the logging level for the $SPLUNK_HOME/var/log/splunk/SA-ldapsearch.log file. Splunk can access this file with &quot;index=_internal sourcetype=SA-ldapsearch&quot; search and exposes the following fields:</td>
</tr>
<tr>
<td>File</td>
<td>Full pathname of the source file where the logging call was made.</td>
</tr>
<tr>
<td>Level</td>
<td>Level of the logging call that was made; one of CRITICAL, ERROR, WARNING, INFO, or DEBUG.</td>
</tr>
<tr>
<td>Line</td>
<td>Line number in the source file where the logging call was made.</td>
</tr>
<tr>
<td>Pid</td>
<td>ID of the process that made the logging call.</td>
</tr>
<tr>
<td>log_source</td>
<td>String of the form &quot;Pid=&lt;Pid&gt;, File=&lt;File&gt;, Line=&lt;Line&gt;&quot;.</td>
</tr>
<tr>
<td>message</td>
<td>Full text of the logged message.</td>
</tr>
</tbody>
</table>

ldapfetch writes its debug logs to $SPLUNK_HOME/var/log/splunk/SA-ldapsearch.log. Splunk indexes and rotates this file by default.

On return, ldapfetch adds the LDAP attributes specified by the attrs argument to each event, based on an LDAP fetch operation. It uses the dn field to determine the LDAP record to fetch.

If you do not specify the dn, then ldapfetch uses the distinguishedName field name.

### Examples

To list all Organizational Units (OU), together with any Group Policy Objects (GPOs) linked to them:

```bash
|ldapsearch domain=SPL search="(objectclass=organizationalUnit)" attrs="ou,description,gPLink,gPOptions" |sort ou |rex field=gPLink max_match=10 "\[LDAP://(?<gpo>[^;]+);\d+\]" |ldapfetch dn=gpo attrs=displayname |table ou,description,displayname |rename ou as "Name", displayname as "Linked GPO"
```
The ldapgroup command

Overview

The 'ldapgroup' command filters and augments events with information from Active Directory. It follows a 'search' or similar command in the pipeline so that you can feed it events. A sample usage follows:

```
|ldapsearch domain=SPL search="(objectClass=group)"|ldapgroup
```

There are several possible arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupdn=&lt;field-name&gt;</td>
<td>Specifies the field to use as the distinguished name of the group to expand.</td>
</tr>
<tr>
<td>domain=&lt;domain&gt;</td>
<td>Specifies the name of a configuration stanza in ldap.conf. If you do not specify a domain, the command uses the default stanza.</td>
</tr>
<tr>
<td>debug=&lt;boolean&gt;</td>
<td>Specifies whether or not ldapgroup should write debug log data. When set to T, specified logging level should occur.</td>
</tr>
<tr>
<td>logging_level=(CRITICAL</td>
<td>ERROR</td>
</tr>
<tr>
<td></td>
<td><strong>File</strong>: Full pathname of the source file where the logging call was made.</td>
</tr>
<tr>
<td></td>
<td><strong>Level</strong>: Level of the logging call that was made. One of CRITICAL, ERROR, WARNING, INFO, DEBUG.</td>
</tr>
<tr>
<td></td>
<td><strong>Line</strong>: Line number in the source file where the logging call was made.</td>
</tr>
<tr>
<td></td>
<td><strong>Pid</strong>: ID of the process that made the logging call.</td>
</tr>
<tr>
<td></td>
<td><strong>log_source</strong>: String of the form &quot;Pid=&lt;Pid&gt;, File=&lt;File&gt;, Line=&lt;Line&gt;&quot;.</td>
</tr>
<tr>
<td></td>
<td><strong>message</strong>: Full text of the logged message.</td>
</tr>
</tbody>
</table>

ldapgroup writes its debug logs to $SPLUNK_HOME/var/log/splunk/SA-ldapsearch.log. Splunk indexes and rotates
this file by default.

Once it completes execution, `ldapgroup` adds five additional fields to each event:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>member_dn</td>
<td>The list of Member Distinguished Names (DNs).</td>
</tr>
<tr>
<td>member_domain</td>
<td>The NetBIOS domain(s) for the member DN(s).</td>
</tr>
<tr>
<td>member_name</td>
<td>The sAMAccountName (SAM account name) for the member DN(s).</td>
</tr>
<tr>
<td>member_type</td>
<td>The type of membership (one of PRIMARY, DIRECT or NESTED with the group DN).</td>
</tr>
<tr>
<td>mv_combo</td>
<td>all of the above, combined into a single field separated by #.</td>
</tr>
</tbody>
</table>

**Examples**

To display a table of all groups with their members and membership type:

```bash
|ldapsearch domain=SPL search="(objectClass=group)"|table cn,distinguishedName|ldapgroup|table cn,member_dn,member_type
```

**The ldaptestconnection command**

**Overview**

The 'ldaptestconnection' command tests the connection to each of the hosts servicing the LDAP directory identified by `domain`. It must be placed at the beginning of a search pipeline. A sample usage follows:

```
| ldaptestconnection domain=default
```

There are several possible arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain=&lt;domain&gt;</td>
<td>Specifies the name of a configuration stanza in ldap.conf. If you do not specify a domain, the command uses the default stanza.</td>
</tr>
<tr>
<td>debug=&lt;boolean&gt;</td>
<td>Specifies whether or not ldapfetch should write debug log data. When set to T, specifies that debug log data is written.</td>
</tr>
</tbody>
</table>
logging should occur.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>logging_level=(CRITICAL</td>
<td>ERROR</td>
</tr>
</tbody>
</table>

File: Full pathname of the source file where the logging call was made. 
Level: Level of the logging call that was made; one of CRITICAL, ERROR, WARNING, INFO, or DEBUG. 
Line: Line number in the source file where the logging call was made. 
Pid: ID of the process that made the logging call. 
log_source: String of the form "Pid=<Pid>, File=<File>, Line=<Line>". 
message: Full text of the logged message. 

On return, ldaptestconnection generates a single event record for each host servicing the LDAP directory identified by the default domain. Each record contains the host name and the distinguished name of the domain. If the connection test fails for any reason, an error message is produced instead.

Example

To test the connection for the SPL domain:

| ldaptestconnection domain=SPL |
Troubleshoot SA-ldapsearch

Troubleshoot the Splunk Supporting Add-on for Active Directory

When an app that uses the Splunk Supporting Add-on for Active Directory cannot complete a search, it notifies you by displaying an error message in the Splunk status bar (at the top of your browser window), as follows:

External search command 'ldapsearch' returned error code 1.
ERROR "Invalid credentials for the user with binddn="<user>". Please correct and test your SA-ldapsearch credentials in the context of domain="<domain>"

It also writes a message to $SPLUNK_HOME/var/log/splunk/SA-ldapsearch.log, similar to the following:

2014-10-10 13:45:31,052, Level=ERROR, Pid=3950, File=search_command.py, Line=278, Abnormal exit: "Invalid credentials for the user with binddn="<user>". Please correct and test your SA-ldapsearch credentials in the context of domain="<domain>"

Configuration page not saving

if you get no response when saving or trying to test the connection in your domain configurations, upgrade your Splunk Enterprise deployment to version 7.0 or above.

LDAP commands exit with 'undefined domain' error

If you configure or reference an invalid domain in ldap.conf, the ldapfilter, ldapfetch, and ldapgroup commands in a subsequent search exit immediately with an error similar to the following:

External search command 'ldapfilter' returned error code 1. Script output =
" ERROR Undefined domain name: <domain>.

The commands immediately stop execution at that point and do not search
further, even if the query source has additional entries with valid domains.

To fix the problem, confirm that you have defined all domains that the add-on must connect to in ldap.conf.

**LDAP commands exit with 'No key or prefix' error**

If you do not configure the default domain in ldap.conf the ldapfilter, ldapfetch, and ldapgroup commands in a subsequent search exit immediately with an error similar to the following:

```
External search command 'ldapgroup' returned error code 1. Script output =
" ERROR "KeyError at "/Applications/Splunk/etc/apps/SA-ldapsearch/bin/packages/splunklib/data.py"", line 245 : u'No key or prefix: $text.' "
```

To prevent this error, confirm that you have configured the default domain in the add-on configuration page.

**Other Issues**

*Authentication fails despite a successful connectivity test after configuration*

If you encounter a problem where queries with SA-LDAPsearch fail despite successfully testing a connection that you set up on the configuration page, make sure that the user that you log into Splunk Enterprise as has the admin_all_objects capability. This capability must be present because the configuration page saves passwords as storage passwords, and only this capability allows users to read storage passwords.

If you cannot grant the admin_all_objects capability, as a workaround, you can use a clear-text password and obfuscate that password with base-64 encoding. In this case, however, you can not use the configuration page to save the password nor can you test the connection. This is because the configuration page moves any clear-text passwords to storage passwords when you save the configuration.

You must edit ldap.conf with a text editor and save the password(s) that way, and then use the ldaptestconnection command to test the configuration.

**In Splunk 6.4 and later**, you can set user role capabilities with a combination of rest_properties_get and list_storage_passwords to allow the various ldap
commands to retrieve and utilise the stored password value.

**Domain name is case-sensitive when user configures ldapsearch**

Windows forwarders log domain names as uppercase. Avoid configuration issues by configuring all domain names in uppercase on ldapsearch’s settings page.

**Ldapfetch, ldapfilter, and ldapgroup commands always query the default domain**

The Ldapfetch, ldapfilter, and ldapgroup commands expand the domain option based on input record values. This means that you cannot always use the domain option value to query for schema before processing records. The commands rely on the default domain when querying for schema instead. One reason for this is that they must know what column names to add to the output record stream for every domain they might query.

To fix this problem, configure the default domain so that it connects to the global catalog server for your Active Directory Domain Services (AD DS) forest. A global catalog server is a domain controller that stores a full copy of all objects in the directory for its host domain and a partial, read-only copy of all objects for all other domains in the forest. Global catalog servers respond to global catalog queries. They have a copy of the schema that applies to every domain in your forest.

For more information on Global Catalog, see "Understanding the Global Catalog (https://technet.microsoft.com/en-us/library/cc730749.aspx) on MS TechNet.

**Some attributes are not available, and associated searches do not return results**

Some attributes are constructed. This means that they get their values as a result of computation from other attributes.

Constructed attributes cannot be used in LDAP filters. Exceptions to this rule are the attributes createTimeStamp, modifyTimeStamp>, objectClass, and structuralObjectClass. Constructed attributes cannot be returned as value data in an LDAP search request unless you specify scope=base which means that the LDAP client accesses only a single object. It is not possible, for example, to search for all objects in a container and specify a constructed attribute as one of the requested attributes in that search.
Slow performance on queries to Active Directory with large number of users

To improve performance on queries against ADs with large numbers of users, select only the query attributes you need to complete your analysis. For example, if you need just two attributes, distinguishedName and sAMAccountName, say so. Use this command:

```bash
| ldapsearch search="(objectClass=user)" attrs="distinguishedName,sAMAccountName"
```

instead of:

```bash
| ldapsearch search="(objectClass=user)"
```

The former, more specific command reduces the total volume of data that it retrieves from Active Directory and passes through the search pipeline dramatically. This strategy significantly improves performance and reliability.

See the following pages for more information:

- Object Attributes of Category: Constructed (http://www.selfadsi.org/deep-inside/attribute-constructed.htm) on SelfADSI.

Unable to query Active Directory anonymously

By default, Active Directory disables anonymous binding. Until you enable it, errors like this occur:

dc-1.msapps.local: Could not access the directory service at ldaps://dc-1.msapps.local:3269: 000004DC: LdapErr: DSID-0C09072B, comment: In order to perform this operation a successful bind must be completed on the connection., data 0, v2580"

To enable anonymous binding you must change the seventh bit of the dsHeuristics attribute on the CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration,Root domain in forest directory object.
Valid values for the `dsHeuristics` attribute are 0 and 2. By default, the `dsHeuristics` attribute is not set, which is the same as if it were set to 0. If you set the seventh character to 2, anonymous clients can then perform any operation that the access control list (ACL) permits. If the attribute has been set, do not modify any bits in the `dsHeuristics` string other than the seventh. If the value has not been set, make sure that you provide the leading zeros up to the seventh bit. You can use the Active Directory Service Interface (ADSI) Editor Microsoft Management Console snap-in (adsedit.msc) to make the change to the `dsHeuristics` attribute.

After you set the attribute, if you want anonymous users to be able to query Active Directory, you can enable anonymous access to specific directory objects. Users gain anonymous access to Active Directory objects through Anonymous Logon, which is a special security identifier (SID) that AD uses to represent anonymous network callers that perform an LDAP bind with NULL credentials.

See the following pages for more information:

- How Active Directory Searches Work
- DSHeuristics Attribute (Windows)

**SA-LDAPsearch does not return custom AD attributes**

Active Directory does not replicate new attributes to the Global Catalog by default. If you run a query against a Global Catalog server and specify an attribute that has not been replicated to the GC, that attribute will not appear in a result of a query initiated by SA-LDAPsearch. To fix the problem, use the Active Directory Schema MMC snap-in to edit the attribute and enable replication to the Global Catalog. Next, restart the computer or Active Directory and attempt the query in SA-LDAPsearch again.

**SA-LDAPsearch generates "The default configuration stanza for ldap.conf is missing" errors in a distributed Splunk Enterprise or Splunk Cloud environment**

See "Workaround for default configuration stanza errors in distributed environments" in this manual.
Learn more about LDAP queries against Active Directory

Microsoft TechNet provides a number of useful documentation resources on querying Active Directory with LDAP. Here are a few recommendations:

- Active Directory Searches Technical Reference
- How Active Directory Searches Work
- Active Directory: LDAP Syntax Filters
- Global Catalog and LDAP Searches
Reference

The ldap.conf configuration file

Beginning with version 2.0.1, the Splunk Supporting Add-on for Active Directory no longer allows configuration though `ldap.conf`. Use the Configuration page to make edits to the add-on configuration.

When you upgrade from a previous version, the add-on saves your `ldap.conf` into the new configuration format (storage passwords).

The following text remains for reference only.

The ldap.conf configuration file

Within the file are a series of stanzas - one for each domain that you need to monitor. When configuring `ldap.conf`, remember to configure both the "DNS-style" and the "NetBIOS-style" names for each Active Directory domain.

There are two forms of stanza in `ldap.conf`.

**Informational stanza**

The informational stanza specifies all the information necessary to connect to the domain. Here is an example:

```
[spl.com]
server = 192.168.50.1,192.168.50.2
port = 636
ssl = true
basedn = dc=spl,dc=com
binddn = cn=Splunk Searcher,cn=Users,dc=spl,dc=com
password = {64}u9435tr8ujtgfnkjsc
alternatedomain = SPL
```

The valid attributes for the informational stanza are:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>server=&lt;server1&gt;,&lt;server2&gt;;?</td>
<td>Specifies the server or servers you want to connect to. Separate multiple servers with</td>
<td>n/a</td>
</tr>
</tbody>
</table>

35
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>port</td>
<td>Specifies the LDAP port on the servers that you want to connect.</td>
<td>636 (when ssl is true)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>389 (when ssl is false)</td>
</tr>
<tr>
<td>ssl=true/false</td>
<td>Specifies whether or not to use Secure Sockets Layer for communications.</td>
<td>false</td>
</tr>
<tr>
<td>basedn</td>
<td>Specifies the LDAP base Distinguished Name to use when connecting.</td>
<td>n/a</td>
</tr>
<tr>
<td>binddn</td>
<td>Specifies the LDAP binding Distinguished Name (the user account) to use when connecting.</td>
<td>n/a</td>
</tr>
<tr>
<td>password (deprecated)</td>
<td>Specifies the password for the user that you specified in binddn. Allows for a cleartext password or a Base-64-encoded password when prefaced with the string {64}.</td>
<td>n/a</td>
</tr>
<tr>
<td>alternatedomain</td>
<td>Specifies the NetBIOS domain that this domain represents.</td>
<td>n/a</td>
</tr>
<tr>
<td>decode</td>
<td>Specifies whether or not the add-on uses Active Directory formatting extensions. Set to true to enable formatting extensions, and false to disable them. Do not set this attribute unless you understand the ramifications of doing so.</td>
<td>true</td>
</tr>
<tr>
<td>paged_size</td>
<td>Specifies the number of entries to return in a single page of LDAP search results. Do not set this attribute unless</td>
<td>1000</td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
<td>Default</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td></td>
<td>you understand the ramifications of doing so.</td>
<td></td>
</tr>
</tbody>
</table>

Specify multiple servers

You can specify multiple servers by including a list of hosts separated by commas. In this case, SA-ldapsearch uses the fastest available connection. In this case, the server that SA-ldapsearch uses might vary from command to command. You can turn on debug mode to find out which server a particular command uses. Once a command has started on a server, it uses that server until it completes.

The port and ssl parameters are optional. If you do not specify them, SA-ldapsearch uses port 389 and no SSL by default. SA-ldapsearch uses SSL only for encryption and not for authentication. SA-ldapsearch trusts all server side SSL certificates.

The bind Distinguished Name (binddn attribute) is a user within the domain you want to monitor. It must be a user that has at least read access to all attributes and entries that you want to read with any application that uses it.

Base64-encode attributes for added security

The password attribute should be set to the password for the user specified in the binddn attribute. You can use a plain text password, or a base64-encoded one by specifying {64} before the password.

Any attribute can be encoded as Base-64, including the binddn attribute. If your binddn has a special character in it, then use Base-64 encoding to store it.

Note: If you want to base64-encode an attribute, you must use a base-64 encoder to encode the entry for that attribute, and then assign the attribute with the results, preceded by {64}. Simply placing the {64} qualifier before the plain text value will not work.

'Default' stanza

To support context lookups in the "ldapfetch" command, you will also need a "default" stanza that lists a forest-level Global Catalog server by its IP address. In this case, you must specify the port to the Global Catalog. Following is an example:
The Splunk Supporting Add-on for Active Directory has been tested to work with up to 100 domains. However, there is no built-in limit on the number of domains that the add-on can support.

Data and source types for the Splunk Supporting Add-on for Active Directory

The Splunk Supporting Add-on for Active Directory performs LDAP queries against Active Directory. As the main function of the add-on is to provide data from an external source, it does not collect data like other add-ons do.

The add-on provides the following source type:

<table>
<thead>
<tr>
<th>Log File</th>
<th>Log Format</th>
<th>Source Type</th>
<th>Default File Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA-LDAPsearch</td>
<td>Plain text</td>
<td>SA-Ldapsearch</td>
<td>$SPLUNK_HOME/var/log/splunk(sa-ldapsearch.log)</td>
</tr>
</tbody>
</table>

SA-LDAPsearch.log

The SA-LDAPsearch.log file contains debugging information from the various SA-LDAPsearch commands.
Release Notes

Release Notes for Splunk Supporting Add-on for Active Directory

This topic contains information on new features, known issues, and updates as we version the Splunk Supporting Add-on for Active Directory.

This version of the Splunk Supporting Add-on for Active Directory was released on October 22, 2019.

What's new

Version 3.0.0 of the Splunk Supporting Add-on for Active Directory provides Python 3 support in Splunk Enterprise. You can run Splunk Enterprise version 8.0.0 in either Python 2 or Python 3 mode.

See the known issues and fixed issues of these release notes for other product updates.

Known issues

This version of the Splunk Supporting Add-on for Active Directory has the following reported known issues and workarounds. If no issues appear below, no issues have yet been reported.

Fixed issues

This version of the Splunk Supporting Add-on for Active Directory fixes the following issues. If no issues appear below, no issues have yet been reported.

<table>
<thead>
<tr>
<th>Date resolved</th>
<th>Issue number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-08-22</td>
<td>TAG-12770,</td>
<td>format of attrs=&quot;whenCreated&quot; differs from the same run by ldapsearch</td>
</tr>
<tr>
<td></td>
<td>TAG-12781</td>
<td></td>
</tr>
</tbody>
</table>
Workaround for default configuration stanza errors in distributed environments

Introduction

This page discusses how to work around a problem where Splunk Supporting Add-on for Active Directory (SA-LDAPsearch) returns an error message about a missing configuration stanza when it runs in a distributed Splunk Enterprise or Splunk Cloud environment.

In a standard Splunk Enterprise environment, SA-LDAPsearch connects to Active Directory and retrieves user records on a search head. In a distributed environment or a Splunk Cloud deployment, the add-on can be configured to distribute search commands across search peers that the search head manages.

Because SA-LDAPsearch must have direct access to the Active Directory domain controllers, any distribution of the add-on requires the hosts where you distribute the app also to have access to Active Directory. As well, the add-on must have the exact same configuration on the search peers that it has on the search head.

Symptoms

When you run queries with SA-LDAPsearch in a distributed Splunk Enterprise or Splunk Cloud environment, you receive the following error message:

External search command 'ldapfilter' returned error code 1. Script output = " ERROR The default configuration stanza for ldap.conf is missing.
You might also receive a message like:

The default configuration stanza for ldap.conf is missing: HTTP 404 Not Found - Application does not exist: SA-ldapsearch
You check ldap.conf on the search head, and the [default] stanza is present.

Cause

The cause of this problem is a bug in how SA-LDAPsearch handles distributed LDAP search queries.
Workaround

There are two ways to work around this problem:

Install SA-LDAPsearch on the search head and all search peers

This option has you configure SA-LDAPsearch on the search head and any search peers. It ensures that the configuration is the same across all of the peers.

1. Install SA-LDAPsearch using Splunk Web.
2. Configure the add-on with Splunk Web by adding a domain to the SA-LDAPsearch configuration.
3. Click the Test connection button in the configuration page to confirm that the add-on can connect to the Active Directory domain you specified.
4. Once the test succeeds, click Save to save the configuration.
5. Repeat this process for all search peers in the deployment.

Modify SA-LDAPsearch to make only local queries

Modify the sa-ldapsearch add-on directly to use only local queries. When you complete the modification, the add-on performs all queries from the search head, and does not attempt to distribute the queries on any search peers. Use this option if you do not want to install the add-on into the search peers.

Caution: The following steps require that you make changes directly to the add-on. If you do not make the changes correctly, you might render the add-on unstable or unusable. Restricting LDAP queries to the search head only can result in degraded search performance. Upgrading the Splunk Supporting Add-on for Active Directory might reverse these changes. If you are either unsure or uncomfortable about making the changes, contact your Splunk support representative for assistance.

1. Use your operating system file management tools to create
   $SPLUNK_HOME/etc/apps/SA-Ldapsearch/local/commands.conf. The easiest way is to copy only the stanzas of
   $SPLUNK_HOME/etc/apps/SA-Ldapsearch/default/commands.conf that are needed for your Splunk platform deployment and add them to your
   local/commands.conf file.
2. Use a text editor to open
   $SPLUNK_HOME/etc/apps/SA-Ldapsearch/local/commands.conf for editing.
3. In each stanza within this file, change the following entry:
local = false

to

local = true

4. Save the file and close it.
5. Restart Splunk Enterprise on the instance.
6. Run a search with the add-on. You should no longer receive the error message.

Third-party software attributions/credits

Some of the components included in Splunk Supporting Add-on for Active Directory are licensed under free or open source licenses. We wish to thank the contributors to those projects.

View the license(s) associated with each component by selecting a component name on the left.

These attributions are in addition to the attributions we give for third-party vendors whose components the Splunk Enterprise software uses and redistributes. You can find those credits in the Release Notes.

We wish to thank the contributors to these projects:

- **ldap3 (v 2.5.1)** - GNU Lesser General Public License v3 (LGPLv3)

The ldap library file was modified December 2018.

https://github.com/cannatag/ldap3/

Copyright 2013 - 2018 Giovanni Cannata

This program is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public
License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see http://www.gnu.org/licenses/.

http://opensource.org/licenses/lgpl-3.0.html
http://opensource.org/licenses/gpl-3.0