Splunk® Enterprise Getting Data In 8.1.0

Monitor Windows event log data

Generated: 10/23/2020 10:21 am

Monitor Windows event log data

Windows generates log data during the course of its operation. The Windows Event Log service handles nearly all of this communication. It gathers log data published by installed applications, services and system processes and places them into event log channels. Programs such as Microsoft Event Viewer subscribe to these log channels to display events that have occurred on the system.

Splunk Enterprise can monitor event log channels and files stored on the local machine, and it can collect logs from remote machines. The event log monitor runs as an input processor within the `splunkd` service. It runs once for every event log input that you define in Splunk Enterprise. If you have Splunk Cloud and want to monitor event log channels, use the Splunk Universal Forwarder to collect the data and forward it to your Splunk Cloud deployment.

New for versions 6.4.5 and later of Splunk Enterprise, the Windows Event Log monitoring input has improved performance.

Why monitor event logs?

Windows event logs are the core metric of Windows machine operations - if there is a problem with your Windows system, the Event Log service has logged it. Splunk Enterprise indexing, searching, and reporting capabilities make your logs accessible.

Prerequisites to monitoring event logs

Activity:	Required permissions:
Monitor local event logs	Splunk Enterprise must run on Windows Splunk Enterprise must run as the Local System user to read all local event logs
Monitor remote event logs	Either Splunk Enterprise or a universal forwarder must run on the Windows machine from which you want to collect event logs Splunk Enterprise or a universal forwarder must run as a domain or remote user with read access to Windows Management Instrumentation (WMI) on the target machine The user that Splunk Enterprise or the universal forwarder runs as must have read access to the event logs you want to collect

Security and remote access considerations

Splunk Enterprise collects event log data from remote machines using either WMI or a universal forwarder. Splunk best practice is to use a universal forwarder to send event log data from remote machines to an indexer. See The universal forwarder in the *Universal Forwarder* manual for information about how to install, configure and use the forwarder to collect event log data.

To install forwarders on your remote machines to collect event log data, you can install the forwarder as the Local System user on these machines. The Local System user has access to all data on the local machine, but not on remote machines.

To use WMI to get event log data from remote machines, you must ensure that your network and Splunk instances are properly configured. You cannot install the Splunk platform as the Local System user, and the user you install with determines the event logs Splunk software sees. See Security and remote access considerations in Monitor WMI-based

data for additional information on the requirements you must satisfy to collect remote data properly using WMI.

By default, Windows restricts access to some event logs depending on the version of Windows you run. For example, only members of the local Administrators or global Domain Admins groups can read the Security event logs by default.

How the Windows Event Log monitor interacts with Active Directory (AD)

When you set up an Event Log monitoring input for WMI, the input connects to an AD domain controller to authenticate and, if necessary, perform any security ID (SID) translations before it begins to monitor the data.

The Event Log monitor uses the following logic to interact with AD after you set it up:

1. If you specify a domain controller when you define the input (with the `evt_dc_name` setting in `inputs.conf`), then the input uses that domain controller for AD operations.
2. If you do not specify a domain controller, then the input does the following:
    1. The input attempts to use the local system cache to authenticate or resolve SIDs.
    2. If the monitor cannot authenticate or resolve SIDs that way, it attempts a connection to the domain controller that the machine that runs the input used to log on.
    3. If that does not work, then the input attempts to use the closest AD domain controller that has a copy of the Global Catalog.
3. If the domain controller that you specify is not valid, or a domain controller cannot be found, then the input generates an error message.

Collect event logs from a remote Windows machine

You have several choices to collect data from a remote Windows machine:

Use a universal forwarder

You can install a universal forwarder on the Windows machine and instruct it to collect event logs. You can do this manually, or use a deployment server to manage the forwarder configuration.

For specific instructions to install the universal forwarder, see Install a Windows universal forwarder from an installer in the Universal Forwarder manual.

1. On the Windows machine that you want to collect Windows Event Logs, download the universal forwarder software from Splunk.
2. Run the universal forwarder installation package to begin the installation process.
3. When the installer prompts you, configure a receiving indexer.
4. When the installer prompts you to specify inputs, enable the event log inputs by checking the "Event logs" checkbox.
5. Complete the installation procedure.
6. On the receiving indexer, use Splunk Web to search for the event log data. An example search string follows:

```
host=<name of remote Windows machine> sourcetype=Wineventlog
```

Use WMI

If you choose to collect event logs remotely using WMI, you must install Splunk Enterprise to run as an Active Directory domain user. If the selected domain user is not a member of the Administrators or Domain Admins groups, then you must configure event log security to give the domain user access to the event logs.

To change event log security for access to the event logs from remote machines, you must:

- Have administrator access to the machine from which you are collecting event logs.
- Understand how the Security Description Definition Language (SDDL) works, and how to assign permissions with it. See Security Description Definition Language (SDDL) for more information.

You can use the `wevtutil` utility to set event log security.

See Considerations for deciding how to monitor remote Windows data for information on collecting data from remote Windows machines.

1. Download Splunk Enterprise instance onto a Windows machine.
2. Double-click the installer file to begin the installation.
3. When the installer prompts you to specify a user, choose **Domain user**.
4. On the next installer pane, enter the domain user name and password that Splunk Enterprise should use when it runs.
5. Follow the prompts to complete installation of the software.
6. Once the software has installed, log into the instance.
7. Use Splunk Web to add the remote event log input, as described in Configure remote event log monitoring.

Anomalous machine names are visible in event logs on some systems

On some Windows systems, you might see some event logs with randomly-generated machine names. This is the result of those systems logging events before the user has named the system, during the OS installation process.

This anomaly occurs only when you collect logs from the above-mentioned versions of Windows remotely over WMI.

Use Splunk Web to configure event log monitoring

To get local Windows event log data, point your Splunk instance at the Event Log service.

Go to the Add New page

You can get there by two routes:

- Splunk Home
- Splunk Settings

By Splunk Settings:

1. Click **Settings** in the upper right corner of Splunk Web.
2. Click **Data Inputs**.
3. Click **Local event log collection**.
4. Click **New** to add an input.

By Splunk Home:

1. Click the **Add Data** link in Splunk Home.
2. Click **Monitor** to monitor Event Log data on the local Windows machine, or **Forward** to forward Event Log data from another Windows machine. Splunk Enterprise loads the "Add Data - Select Source" page.

3. If you selected **Forward**, choose or create the group of forwarders you want this input to apply to. See "Forward data" in this manual.
4. Click **Next**.

Select the input source

1. In the left pane, select **Local Event Logs**
2. In the **Select Event Logs** list box, choose the Event Log channels you want this input to monitor.
3. Click once on each Event Log channel you want to monitor. Splunk Enterprise moves the channel from the "Available items" window to the "Selected items" window.
4. To unselect a channel, click on its name in the "Available Items" window. Splunk Enterprise moves the channel from the "Selected items" window to the "Available items" window.
5. To select or unselect all of the event logs, click on the "add all" or "remove all" links. **Important:** Selecting all of the channels can result in the indexing of a lot of data, possibly more than your license allows.
6. Click **Next**.

Specify input settings

The **Input Settings** page lets you specify application context, default host value, and index. All of these parameters are optional.

Host only sets the **host** field in the resulting events. It does not direct Splunk Enterprise to look on a specific machine on your network.

1. Select the appropriate **Application context** for this input.
2. Set the **Host** name value. You have several choices for this setting. Learn more about setting the host value in About hosts.
3. Set the **Index** that Splunk Enterprise should send data to. Leave the value as "default", unless you have defined multiple indexes to handle different types of events. In addition to indexes for user data, Splunk Enterprise has a number of utility indexes, which also appear in this dropdown box.
4. Click **Review**.

Review your choices

After you specify all your input settings, you can review your selections. Splunk Enterprise lists all options you selected, including the type of monitor, the source, the source type, the application context, and the index.

1. Review the settings.
2. If they do not match what you want, click **<** to go back to the previous step in the wizard. Otherwise, click **Submit**.

Splunk Enterprise then loads the "Success" page and begins indexing the specified Event Log channels.

Configure remote event log monitoring

The process for configuring remote event log monitoring is nearly identical to the process for monitoring local event logs.

Selecting all of the Event Log channels can result in the indexing of a lot of data, possibly more than your Splunk license can support.

1. Follow the instructions to get to the Add New page, as described in Go to the Add New page.
2. In the left pane, locate and select **Remote Event Logs**.
3. In the **Event Log collection name** field, enter a unique name for this input that you will remember.

4. In the **Choose logs from this host** field, enter the host name or IP address of the machine that contains the Event Log channels you want to monitor.
5. Click the **Find logs** button to refresh the page with a list of available Event Log channels on the machine you entered.
6. Click once on each Event Log channel you want to monitor. Splunk Enterprise moves the channel from the "Available items" window to the "Selected items" window.
7. To unselect a channel, click on its name in the "Available Items" window. Splunk Enterprise moves the channel from the "Selected items" window to the "Available items" window.
8. To select or unselect all of the event logs, click on the "add all" or "remove all" links.
9. In the **Collect the same set of logs from additional hosts** field, enter host names or IP addresses of additional machines that contain the Event Logs you selected previously. Separate multiple machines with commas.
10. Click the green **Next** button.
11. Follow the instructions to specify input settings, as described in "Specify input settings."
12. Follow the instructions to review your choices, as described in "Review your choices."

Use inputs.conf to configure event log monitoring

Edit `inputs.conf` to configure event log monitoring.

1. Using Notepad or a similar editor, open `%SPLUNK_HOME%\etc\system\local\inputs.conf` for editing. You might need to create this file if it does not exist.
2. Enable Windows event log inputs by adding input stanzas that reference Event Log channels.
3. Save the file and close it.
4. Restart Splunk Enterprise.

For more information on configuring data inputs with `inputs.conf`, see Configure your inputs.

Specify global settings for Windows Event Log inputs

As you define Windows Event Log inputs in inputs.conf, confirm that you explicitly specify global settings in the correct place.

If you specify global settings for Windows Event Log inputs, such as `host`, `sourcetype`, and so on, you can place those settings in one of the following areas:

- Under the `[WinEventLog]` global stanza. This stanza is equal to the `[default]` stanza for other monitoring inputs. For example:

```
[default]
_meta = hf_proxy::meta_test

[WinEventLog]
_meta = hf_proxy::meta_test
host = WIN2K16_DC
index = wineventlog

[WinEventLog://Applications]
disabled = 0
```

- Under the Windows Event Log input stanza for the Event Log channel that you want to monitor. For example:

```
[default]
```

```
_meta = hf_proxy::meta_test

[WinEventLog]
host = WIN2K16_DC
index = wineventlog

[WinEventLog://Applications]
disabled = 0
_meta = hf_proxy::meta_test
```

You can always review the defaults for a configuration file by looking at the examples in `%SPLUNK_HOME%\etc\system\default` or at the spec file in the Admin Manual.

Event log monitor configuration values

Windows event log (*.evt) files are in binary format. You cannot monitor them like you do a normal text file. The `splunkd` service monitors these binary files by using the appropriate APIs to read and index the data within the files.

Splunk Enterprise uses the following stanzas in `inputs.conf` to monitor the default Windows event logs:

```
# Windows platform specific input processor.
[WinEventLog://Application]
disabled = 0
[WinEventLog://Security]
disabled = 0
[WinEventLog://System]
disabled = 0
```

Monitor non-default Windows event logs

You can also configure Splunk Enterprise to monitor non-default Windows event logs. Before you can do this, you must import them to the Windows Event Viewer. After you import the logs, you can add them to your local copy of `inputs.conf`, as follows:

```
[WinEventLog://DNS Server]
disabled = 0
[WinEventLog://Directory Service]
disabled = 0
[WinEventLog://File Replication Service]
disabled = 0
```

Use the "Full Name" log property in Event Viewer to specify complex Event Log channel names properly

You can use the "Full Name" Event Log property in Event Viewer to ensure that you specify the correct Event Log channel in an `inputs.conf` stanza.

For example, to monitor the Task Scheduler application log (`Microsoft-Windows-TaskScheduler-Operational`):

1. Launch Event Viewer.
2. Expand `Applications and Services Logs > Microsoft > Windows > TaskScheduler`.
3. Right-click `Operational` and select **Properties**.
4. In the dialog that appears, copy the text in the "Full Name" field.
5. Append this text into the `WinEventLog://` stanza:

```
[WinEventLog://Microsoft-Windows-TaskScheduler/Operational]
disabled = 0
```

Disable an event log stanza

To disable indexing for an event log, add `disabled = 1` below its listing in the stanza in `%SPLUNK_HOME%\etc\system\local\inputs.conf`.

Configuration settings for monitoring Windows Event Logs

Splunk software uses the following settings in inputs.conf to monitor Event Log files:

Attribute	Description	Default
`start_from`	How events are to be read. Acceptable values are `oldest` (meaning read logs from the oldest to the newest) and `newest` (meaning read logs from the newest to the oldest.) You cannot set this attribute to `newest` while also setting the `current_only` attribute to `1`.	`oldest`
`current_only`	How events are to be indexed. Acceptable values are 1 (where the input acquires events that arrive after the input starts for the first time, like 'tail -f' on *nix systems) or 0 (where the input gets all existing events in the log and then continues to monitor incoming events in real time.) You cannot set this attribute to `1` and also set the `start_from` attribute to `newest`.	0
`checkpointInterval`	How frequently, in seconds, the Windows Event Log input saves a checkpoint. Checkpoints store the eventID of acquired events to enable Splunk software to resume monitoring at the correct event after a shutdown or outage.	0
`evt_resolve_ad_ds`	The domain controller Splunk software uses to interact with Active Directory while indexing Windows Event Log channels. Valid only when you set the `evt_resolve_ad_obj` attribute to `1` and omit the `evt_dc_name` attribute. Valid values are `auto` (meaning choose the nearest domain controller to bind to for AD object resolution) or `PDC` (meaning bind to the primary domain controller for the AD site that the host is in.) If you also set the `evt_dc_name` attribute, Splunk software ignores this attribute.	`auto`
`evt_resolve_ad_obj`	How Splunk software interacts with Active Directory while indexing Windows Event Log channels. Valid values are 1 (meaning resolve Active Directory objects like Globally Unique IDentifier (GUID) and Security IDentifier (SID) objects to their canonical names for a specific Windows event log channel) and 0 (meaning not to attempt any resolution.) When you set this value to 1, you can optionally specify the Domain Controller name and/or DNS name of the domain to bind to, which Splunk software uses to resolve the AD objects. If you do not set this value, Splunk software attempts to resolve the AD objects.	0
`evt_dc_name`	Which Active Directory domain controller to bind to resolve AD objects. This name can be the NetBIOS name of the domain controller, the fully-qualified DNS name of the domain controller, or an environment variable name, specified as	N/A

Attribute	Description	Default
	`$Environment_variable`. If you set this attribute, then Splunk software ignores the `evt_resolve_ad_ds` attribute, which controls how the software determines the best domain controller to bind to for AD object resolution. If you specify an environment variable, you must prepend a dollar sign (`$`) to the environment variable name. Splunk software uses the specified environment variable as the domain controller to connect to for AD object resolution. For example, to use the `%LOGONSERVER%` variable, specify `evt_dc_name = $logonserver`. You can precede either format with two backslash characters. This attribute does not have a default.	
`evt_dns_name`	The fully-qualified DNS name of the domain to bind to resolve AD objects.	N/A
`evt_exclude_fields`	A list of Windows Event Log fields that the Windows Event Log input is to exclude when it ingests Windows Event Log data. When you specify this setting, the input removes both the key and value data for the fields you exclude. This setting works similar to the `suppress_*` settings, but unlike those settings, this setting is valid for all Windows Event Log fields, and excludes fields that you might have included in an allow list. When this collision happens, the instance logs an error. See "Create advanced filters with 'whitelist' and 'blacklist'" later in this topic for the list of Windows Event Log fields that you can exclude.	N/A
`suppress_text`	Whether to include the message text that comes with a security event. A value of 1 suppresses the message text, and a value of 0 preserves the text.	0
`use_old_eventlog_api`	Whether or not to read Event Log events with the Event Logging API. This is an advanced setting. Contact Splunk Support before you change it. If set to true, the input uses the Event Logging API (instead of the Windows Event Log API) to read from the Event Log on Windows Server 2008, Windows Vista, and later installations.	`false` (Use the API that is specific to the OS.)
`use_threads`	Specifies the number of threads, in addition to the default writer thread, that can be created to filter events with the blacklist/whitelist regular expression. This is an advanced setting. Contact Splunk Support before you change it. The maximum number of threads is 15.	0
`thread_wait_time_msec`	The interval, in milliseconds, between attempts to re-read Event Log files when a read error occurs. This is an advanced setting. Contact Splunk Support before you change it.	5000
`suppress_checkpoint`	Whether or not the Event Log strictly follows the 'checkpointInterval' setting when it saves a checkpoint.	false

Attribute	Description	Default
	This is an advanced setting. Contact Splunk Support before you change it. By default, the Event Log input saves a checkpoint from between zero and `checkpointInterval` seconds, depending on incoming event volume.	
`suppress_sourcename`	Whether or not to exclude the 'sourcename' field from events. This is an advanced setting. Contact Splunk Support before you change it. When set to true, the input excludes the 'sourcename' field from events and thruput performance (the number of events processed per second) improves.	false
`suppress_keywords`	Whether or not to exclude the 'keywords' field from events. This is an advanced setting. Contact Splunk Support before you change it. When set to true, the input excludes the 'keywords' field from events and thruput performance (the number of events processed per second) improves.	false
`suppress_type`	Whether or not to exclude the 'type' field from events. This is an advanced setting. Contact Splunk Support before you change it. When set to true, the input excludes the 'type' field from events and thruput performance (the number of events processed per second) improves.	false
`suppress_task`	Whether or not to exclude the 'task' field from events. This is an advanced setting. Contact Splunk Support before you change it. When set to true, the input excludes the 'task' field from events and thruput performance (the number of events processed per second) improves.	false
`suppress_opcode`	Whether or not to exclude the 'opcode' field from events. This is an advanced setting. Contact Splunk Support before you change it. When set to true, the input excludes the 'opcode' field from events and thruput performance (the number of events processed per second) improves.	false
`whitelist`	Whether to index events that match the specified text string. This attribute is optional. You can specify one of two formats: • One or more Event Log event codes or event IDs (Event Code/ID format.) • One or more sets of keys and regular expressions (Advanced filtering format.) You cannot mix formats in a single entry. You also cannot mix formats in the same stanza.	N/A

Attribute	Description	Default
	Splunk software processes whitelists first, then blacklists. If no whitelist or blacklist is present, all events are indexed. When you use the Event Code/ID format: • For multiple codes/IDs, separate the list with commas. • For ranges, use hyphens (for example "0-1000,5000-1000"). When using the advanced filtering format: • Use '=' between the key and the regular expression that represents your filter (for example "whitelist = EventCode=%^1([8-9])$%" • You can have multiple key/regular expression sets in a single advanced filtering entry. Splunk Enterprise conjuncts the sets logically. This means that the entry is valid only if all of the sets in the entry are true. • You can specify up to 10 whitelists per stanza by adding a number to the end of the `whitelist` attribute, for example `whitelist1...whitelist9`.	
`blacklist`	Do not index events that match the text string specified. This attribute is optional. You can specify one of two formats: • One or more Event Log event codes or event IDs (Event Log code/ID format.) • One or more sets of keys and regular expressions. (Advanced filtering format.) You cannot mix formats in a single entry. You also cannot mix formats in the same stanza. Splunk software processes whitelists first, then blacklists. If no whitelist or blacklist is present, all events are indexed. When using the Event Log code/ID format: • For multiple codes/IDs, separate the list with commas. • For ranges, use hyphens (for example "0-1000,5000-1000"). When using the advanced filtering format: • Use '=' between the key and the regular expression that represents your filter (for example "blacklist = EventCode=%^1([8-9])$%" • You can have multiple key/regular expression sets in a single advanced filtering entry. Splunk software conjuncts the sets logically. This means that the entry is valid only if all of the sets in the entry are true. • You can specify up to 10 blacklists per stanza by adding a number to the end of the `blacklist` attribute, for example `blacklist1...blacklist9`.	
`renderXml`	Render event data as XML supplied by the Windows Event Log subsystem. This setting is optional.	0 (false)

Attribute	Description	Default
	A value of '1' or 'true' means to render the events as XML. A value of '0' or 'false' means to render the events as plain text. If you set `renderXml` to true, if you want to also create whitelists or blacklists to filter event data, you must use the `$XmlRegex` special key in your whitelists or blacklists.	
`index`	The index that this input should send the data to.	the default index
`disabled`	Whether or not the input should run. • Valid values are 0 (meaning that the input should run) and 1 (meaning that the input should not run.	0

Use the Security event log to monitor changes to files

You can monitor changes to files on your system by enabling security auditing on a set of files and/or directories and then monitoring the Security event log channel for change events. The event log monitoring input includes three attributes which you can use in `inputs.conf`. For example:

```
[WinEventLog://Security]
disabled = 0
start_from = oldest
current_only = 0
evt_resolve_ad_obj = 1
checkpointInterval = 5
# only index events with these event IDs.
whitelist = 0-2000,3001-10000
# exclude these event IDs from being indexed.
blacklist = 2001-3000
```

To enable security auditing for a set of files or directories, read "Auditing Security Events How To" (http://technet.microsoft.com/en-us/library/cc727935%28v=ws.10%29.aspx) on MS Technet.

You can also use the `suppress_text` attribute to include or exclude the message text that comes with a security event.

When you set `suppress_text` to 1 in a Windows Event Log Security stanza, the entire message text does not get indexed. This includes any contextual information about the security event. If you need this contextual information, do not set `suppress_text` in the stanza.

```
[WinEventLog://Security]
disabled = 0
start_from = oldest
current_only = 0
evt_resolve_ad_obj = 1
checkpointInterval = 5
# suppress message text, we only want the event number.
suppress_text = 1
# only index events with these event IDs.
whitelist = 0-2000,2001-10000
# exclude these event IDs from being indexed.
blacklist = 2001-3000
```

To use a specific domain controller, set the `evt_dc_name` attribute:

```
[WinEventLog://Security]
disabled = 0
start_from = oldest
current_only = 0
evt_resolve_ad_obj = 1
evt_dc_name = boston-dc1.contoso.com
checkpointInterval = 5
# suppress message text, we only want the event number.
suppress_text = 1
# only index events with these event IDs.
whitelist = 0-2000,2001-10000
# exclude these event IDs from being indexed.
blacklist = 2001-3000
```

To use the primary domain controller to resolve AD objects, set the `evt_resolve_ad_ds` attribute to `PDC`. Otherwise, it locates the nearest domain controller:

```
[WinEventLog://Security]
disabled = 0
start_from = oldest
current_only = 0
evt_resolve_ad_obj = 1
evt_resolve_ad_ds = PDC
checkpointInterval = 5
# suppress message text, we only want the event number.
suppress_text = 1
# only index events with these event IDs.
whitelist = 0-2000,2001-10000
# exclude these event IDs from being indexed.
blacklist = 2001-3000
```

Create advanced filters with 'whitelist' and 'blacklist'

You can perform advanced filtering of incoming events with the `whitelist` and `blacklist` settings in addition to filtering based solely on event codes. To do this, specify the key/regular expression format in the setting:

```
whitelist = key=<regular expression> [key=<regular expression>] ...
```

In this format, `key` is a valid entry from the following list:

Key	Description
$TimeGenerated	The time that the computer generated the event. Splunk Enterprise only generates the time string as the event.
$Timestamp	The time that the event was received and recorded by the Event Log service. Splunk Enterprise only generates the time string as the event.
$XmlRegex	A special key that configures Splunk Enterprise to filter on XML events. To use this key, set it to the value that you want Splunk Enterprise to filter on. You must configure the input to which you want to apply a blacklist or whitelist to render events in XML. To generate XML events, specify the `renderXml = true` setting under the input stanza. Splunk Enterprise conjuncts multiple entries in a single whitelist or blacklist line. All of the filter entries must match for the filter to trigger.
Category	The category number for a specific event source.
CategoryString	A string translation of the category. The translation depends on the event source.

Key	Description
ComputerName	The name of the computer that generated the event.
EventCode	The event ID number for an event. Corresponds to "Event ID" in Event Viewer.
EventType	A numeric value that represents one of the five types of events that can be logged (Error, Warning, Information, Success Audit, and Failure Audit.) Available only on machines that run Windows Server 2003 and earlier or clients running Windows XP and earlier. See "Win32_NTLogEvent class (Windows)" (http://msdn.microsoft.com/en-us/library/aa394226(v=vs.85).aspx) on MSDN.
Keywords	An element used to classify different types of events within an event log channel. The Security Event Log channel has this element, for example.
LogName	The name of the Event Log channel that received the event. Corresponds to "Log Name" in Event Viewer.
Message	The text of the message in the event.
OpCode	The severity level of the event ("OpCode" in Event Viewer.)
RecordNumber	The Windows Event Log record number. Each event on a Windows machine gets a record number. This number starts at 0 with the first event generated on the system, and increases with each new event generated, until it reached a maximum of 4294967295. It then rolls back over to 0.
Sid	The Security Identifier (SID) of the principal (such as a user, group, computer, or other entity) that was associated with or generated the event. See "Win32_UserAccount class (http://msdn.microsoft.com/en-us/library/windows/desktop/aa394507%28v=vs.85%29.aspx) on MSDN.
SidType	A numeric value that represents the type of SID that was associated with the event. See "Win32_UserAccount class" (http://msdn.microsoft.com/en-us/library/windows/desktop/aa394507%28v=vs.85%29.aspx) on MSDN.
SourceName	The source of the entity that generated the event ("Source" in Event Viewer)
TaskCategory	The task category of the event. Event sources let you define categories so that you can filter them with Event Viewer (using the "Task Category" field. See Event Categories (Windows) (http://msdn.microsoft.com/en-us/library/aa363649%28VS.85%29.aspx) on MSDN.
Type	A numeric value that represents one of the five types of events that can be logged ("Error", "Warning", "Information", "Success Audit", and "Failure Audit".) Only available on machines that run Windows Server 2008 or later, or Windows Vista or later. See "Win32_NTLogEvent class (Windows)" (http://msdn.microsoft.com/en-us/library/aa394226(v=vs.85).aspx) on MSDN.
User	The user associated with the event. Correlates to "User" in Event Viewer.

and `<regular expression>` is any valid regular expression that represents the filters that you want to include (when used with the `whitelist` attribute) or exclude (when used with the `blacklist` attribute).

You can specify more than one key/regular expression set on a single entry line. When you do this, Splunk Enterprise logically conjuncts the sets. This means that only events that satisfy all of the sets on the line are valid for inclusion or exclusion. For example, this entry:

```
whitelist = EventCode="^1([0-5])$" Message="^Error"
```

means to include events that have an `EventCode` ranging from 10 to 15 and contain a `Message` that begins with the word `Error`.

You can specify up to 10 separate whitelist or blacklist entries in each stanza. To do so, add a number at the end of the `whitelist` or `blacklist` entry on a separate line:

```
whitelist = key=<regular expression>
whitelist1 = key=<regular expression> key2=<regular expression 2>
whitelist2 = key=<regular expression>
```

You cannot specify an entry that has more than one key/regular expression set that references the same key. If, for example, you specify:

```
whitelist = EventCode="^1([0-5])$" EventCode="^2([0-5])$"
```

Splunk Enterprise ignores the first set and only attempts to include events that match the second set. In this case, only events that contain an `EventCode` between 20 and 25 match. Events that contain an `EventCode` between 10 and 15 do not match. Only the last set in the entry ever matches. To resolve this problem, specify two separate entries in the stanza:

```
whitelist = EventCode="^1([0-5])$"
whitelist1 = EventCode="^2([0-5])$"
```

Suppress fields from Windows Event Log events

There are two options to limit the ingestion of data by removing Windows Event Log fields from events that a Splunk Platform instance ingests:

- Use the `suppress_*` settings in inputs.conf to remove certain Windows Event Log fields from ingested events.
- Use the `evt_exclude_fields` setting, which lets you remove any Windows Event Log field from a Windows Event Log event. This setting removes both the excluded key and value from the event, and excludes events even if the field exists in an allow list.

You define both of these settings in the inputs.conf configuration file, under a Windows Event Log monitoring input, for example:

`suppress_*` example (suppresses the message text)

```
[WinEventLog://System]
disabled=0
suppress_text=1
```

`evt_exclude_fields` example

```
[WinEventLog://System]
disabled=0
evt_exclude_fields=EventCode,RecordNumber
```

See the list of fields in "Create advanced filters with 'whitelist' and 'blacklist'" earlier in this topic. See "Configuration settings for monitoring Windows Event Logs", also earlier in this topic, for more information about the settings.

Resolve Active Directory objects in event log files

To specify whether Active Directory objects like globally unique identifiers (GUIDs) and security identifiers (SIDs) are resolved for a given Windows event log channel, use the `evt_resolve_ad_obj` attribute (1=enabled, 0=disabled) for that channel's stanza in your local copy of `inputs.conf`. The `evt_resolve_ad_obj` attribute is on by default for the Security channel.

For example:

```
[WinEventLog://Security]
disabled = 0
start_from = oldest
current_only = 0
evt_resolve_ad_obj = 1
checkpointInterval = 5
```

To specify a domain controller for the domain that Splunk should bind to in order to resolve AD objects, use the `evt_dc_name` attribute.

The string specified in the `evt_dc_name` attribute can represent either the domain controller NetBIOS name, or its fully-qualified domain name (FQDN). Either name type can, optionally, be preceded by two backslash characters.

The following examples are correctly formatted domain controller names:

- `FTW-DC-01`
- `\\FTW-DC-01`
- `FTW-DC-01.splunk.com`
- `\\FTW-DC-01.splunk.com`

To specify the FQDN of the domain to bind to, use the `evt_dns_name` attribute.

For example:

```
[WinEventLog://Security]
disabled = 0
start_from = oldest
current_only = 0
evt_resolve_ad_obj = 1
evt_dc_name = ftw-dc-01.splunk.com
evt_dns_name = splunk.com
checkpointInterval = 5
```

Constraints for using the evt_dc_name and evt_resolve_ad_obj attributes

When you use the `evt_resolve_ad_obj` and `evt_dc_name` attributes:

- Splunk software first attempts to resolve SIDs and GUIDs using the domain controller (DC) specified in the `evt_dc_name` attribute first. If it cannot resolve SIDs using this DC, it attempts to bind to the default DC to perform the translation.
- If Splunk software cannot contact a DC to translate SIDs, it attempts to use the local machine for translation.
- If none of these methods works, then Splunk prints the SID as it was captured in the event.
- Splunk software cannot translate SIDs that are not in the format `S-1-N-NN-NNNNNNNNNN-NNNNNNNNNN-NNNNNNNNNN-NNNN`.

If you discover that SIDs are not being translated properly, review `%SPLUNK_HOME%\var\log\splunkd.log` for clues on what the problem might be.

Specify whether to start index at the earliest or the most recent event

Use the `start_from` attribute to specify whether events are indexed starting at the earliest event or the most recent. By default, indexing starts with the oldest data and moves forward. Do not change this setting, because Splunk software stops indexing after it has indexed the backlog using this method.

Use the `current_only` attribute to specify whether to index all preexisting events in a given log channel. When set to 1, only events that appear from the moment the Splunk deployment was started are indexed. When set to 0, all events are indexed.

For example:

```
[WinEventLog://Application]
disabled = 0
start_from = oldest
current_only = 1
```

Display Windows Event Log events in XML

To have Splunk Enterprise generate Windows Event Log events in XML, use the `renderXml` setting in a Windows Event Log input stanza:

```
[WinEventLog://System]
 disabled = 0
 renderXml = 1
 evt_resolve_ad_obj = 1
 evt_dns_name = \"SV5DC02\"
```

This input stanza generates events like the following:

```
 <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'>
   <System>
     <Provider Name='Service Control Manager' Guid='{555908d1-a6d7-4695-8e1e-26931d2012f4}'
EventSourceName='Service Control Manager'/>
     <EventID Qualifiers='16384'>7036</EventID>
     <Version>0</Version>
     <Level>4</Level>
     <Task>0</Task>
     <Opcode>0</Opcode>
     <Keywords>0x8080000000000000</Keywords>
     <TimeCreated SystemTime='2014-04-24T18:38:37.868683300Z'/>
     <EventRecordID>412598</EventRecordID>
     <Correlation/>
     <Execution ProcessID='192' ThreadID='210980'/>
     <Channel>System</Channel>
     <Computer>SplunkDoc.splunk-docs.local</Computer>
     <Security/>
   </System>
   <EventData>
     <Data Name='param1'>Application Experience</Data>
     <Data Name='param2'>stopped</Data>
     <Binary>410065004C006F006F006B00750070005300760063002F0031000000</Binary>
   </EventData>
 </Event>
```

When you instruct Splunk Enterprise to render events in XML, event keys within the XML event render in English regardless of the machine system locale. Compare the following events generated on a French version of Windows Server:

Standard event:

```
04/29/2014 02:50:23 PM
LogName=Security
SourceName=Microsoft Windows security auditing.
EventCode=4672
EventType=0
Type=Information
ComputerName=sacreblue
TaskCategory=Ouverture de session sp<ciale
OpCode=Informations
RecordNumber=2746
Keywords=Succès de l'audit
Message=Privilèges sp<ciaux attribu<s à la nouvelle ouverture de session.

Sujet :
               ID de s<curit< :                  AUTORITE NT\Système
               Nom du compte :                           Système
               Domaine du compte :                    AUTORITE NT
               ID d'ouverture de session :                            0x3e7

Privilèges :                          SeAssignPrimaryTokenPrivilege
                                             SeTcbPrivilege
                                             SeSecurityPrivilege
                                             SeTakeOwnershipPrivilege
                                             SeLoadDriverPrivilege
                                             SeBackupPrivilege
                                             SeRestorePrivilege
                                             SeDebugPrivilege
                                             SeAuditPrivilege
                                             SeSystemEnvironmentP rivilege
                                             SeImpersonatePrivilege
```

XML event:

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'>
               <System><Provider Name='Microsoft-Windows-Security-Auditing'
Guid='{54849625-5478-4994-A5BA-3E3B0328C30D}'/>
                              <EventID>4672</EventID>
                              <Version>0</Version>
                              <Level>0</Level>
                              <Task>12548</Task>
                              <Opcode>0</Opcode>
                              <Keywords>0x8020000000000000</Keywords>
                              <TimeCreated SystemTime='2014-04-29T22:15:03.280843700Z'/>
                              <EventRecordID>2756</EventRecordID>
                              <Correlation/><Execution ProcessID='540' ThreadID='372'/>
                              <Channel>Security</Channel>
                              <Computer>sacreblue</Computer>
                              <Security/>
               </System>
               <EventData>
                              <Data Name='SubjectUserSid'>AUTORITE NT\Système</Data>
                              <Data Name='SubjectUserName'>Système</Data>
                              <Data Name='SubjectDomainName'>AUTORITE NT</Data>
                              <Data Name='SubjectLogonId'>0x3e7</Data>
```

```
                        <Data Name='PrivilegeList'>SeAssignPrimaryTokenPrivilege
                                    SeTcbPrivilege
                                    SeSecurityPrivilege
                                    SeTakeOwnershipPrivilege
                                    SeLoadDriverPrivilege
                                    SeBackupPrivilege
                                    SeRestorePrivilege
                                    SeDebugPrivilege
                                    SeAuditPrivilege
                                    SeSystemEnvironmentP rivilege
                                    SeImpersonatePrivilege< /Data>
            </EventData>
</Event>
```

The `Data Name` keys in the XML event render in English despite rendering in the system's native language in the standard event.

Use blacklists and whitelists to filter on XML-based events

If you render events in XML, and you want to use whitelists and blacklists to filter on those events, you must use the special key `$XmlRegex` when you build your whitelists or blacklists.

The whitelist or blacklist triggers when Splunk Enterprise finds the value that you specify with `$XmlRegex` anywhere in the XML-rendered event. `$XmlRegex` does not work if you do not explicitly specify the input to render events in XML with the `renderXml = true` setting.

The `$XmlRegex` setting does not search for key-value pairs. It configures Splunk Enterprise to expect that the incoming events have been rendered in XML format.

Following is an example of using whitelists on XML events. Splunk Enterprise indexes all XML events that contain the word "Error":

```
[WinEventLog://System]
 disabled = 0
 renderXml = 1
 evt_resolve_ad_obj = 1
 evt_dns_name = \"SV5DC02\"
 whitelist = $XmlRegex='Error'
```

See "Create advanced filters with 'whitelist' and 'blacklist' for additional information and syntax.

Use the CLI to configure event log monitoring

You can use the CLI to configure local event log monitoring. Before you use the CLI, create stanza entries in `inputs.conf` first. See "Use inputs.conf to configure event log monitoring" in this topic.

The CLI is not available for remote Event Log collections.

To list all configured Event Log channels on the local machine:

```
> splunk list eventlog
```

You can also list a specific channel by specifying its name:

```
> splunk list eventlog <ChannelName>
```

To enable an Event Log channel:

```
> splunk enable eventlog <ChannelName>
```

To disable a channel:

```
> splunk disable eventlog <ChannelName>
```

Index exported event log (.evt or .evtx) files

You can ingest the data contained in exported Windows Event Log (.evt) and Windows Event Log XML (.evtx) files similar to the method used to ingest text-based log files. However, reading the Windows Event log files requires API calls and dynamic link libraries (DLL) that are only available on Windows operating system (OS.) Monitoring exported event log files adds significant administrative overhead in maintaining the exporting, moving, and ingesting logistics, as well as disk space management.

Limitations when reading Windows Event log files

- Windows OS does not allow read access to an .evt or .evtx file that is being written to. The file must be finished and closed before reading. Typically this is accomplished by using a script to generate the Windows Event log file, and then moving the finished file into another folder that's monitored for reading.
- A Windows Event log file must be read on a Windows OS host of the same OS version, or newer. For example, a forwarder running on Windows Server 2008/2008 R2 cannot read an .evtx file exported from a system running Windows Server 2012 or later.
- If your .evt or .evtx file was exported from a non-standard Windows OS event log channel, you must load any DLL files required to read the custom event log file content on the forwarder host reading the files. This is common with 3rd-party software that utilizes Windows Event log integration.
- You cannot use the Splunk Web upload data feature to ingest .evt or .evtx files.

When producing .evt or .evtx files on one system and monitoring them on another, it's possible that not all of the fields in each event expand as they would on the system producing the events. This is caused by variations in DLL versions, availability and APIs. Differences in OS version, language, Service Pack level and installed third party DLLs, etc. can also have this effect.

Overview to setup a host to monitor Windows Event log files

1. Install the Windows universal forwarder on a host with the latest Windows OS.
2. Configure the forwarder host's outputs.conf, and confirm it is communicating with the indexer.
3. Choose a folder path on the forwarder host where the .evt or .evtx files will be placed.
4. Create a `[monitor://]` stanza in the inputs.conf on the forwarder host. See Instructions for monitoring files and directories.
    - ♦ Use the path to the folder where the .evt or .evtx files will be read from. Wildcards are accepted in the path.
    - ♦ Do not define the source type, source, or host. By default, those values are extracted from the .evt or .evtx file.
    - ♦ Do not use the `[monitor://]` stanza blacklist or whitelist options with .evt or .evtx files. To filter by file name, configure the `[monitor://]` stanza with a regex that matches the files you want.
5. Place a .evt or .evtx file into the selected path and verify the file is read by the forwarder and sent to the indexer.
    - ♦ On your search head, search for the name of the host where the .evt or .evtx file was generated. The source type will be the Windows Event log channel name. Example: A .evtx file exported from a host's

Security events channel will have the sourcetype and source set to WinEventLog:Security.