

# Session 2 - Bringing in Data

The process of collecting data with Splunk is enhanced, as its system makes it easy to get data from many types of computerized systems, which are responsible for much of the data produced today. Such data is frequently referred to as machine data. And since much of this is streaming data, Splunk is especially useful, as it can handle streaming data quickly and efficiently. Additionally, Splunk can collect data from many other sources.

In this session, you will learn about Splunk and its role in big data, as well as the most common methods of ingesting data into Splunk. The chapter will also introduce essential concepts such as forwarders, indexes, events, event types, fields, sources, and source types. It is paramount that you learn this early on as it will empower you to gather more data. In this session we will cover the following topics:

- Splunk and big data
- Splunk data sources
- Splunk indexes
- Inputting data into Splunk
- Splunk events and fields

## Splunk and big data

Splunk is useful for datasets of all types, and it allows you to use big data tools on datasets of all sizes. But with the recent focus on big data, its usefulness becomes even more apparent. Big data is a term used everywhere these days, but one that few people understand. In this part of the chapter, we will discuss aspects of big data and the terms that describe those aspects.

## Streaming data

Much of the data that is large and comes quickly does not need to be kept. For instance, consider a mechanical plant; there can be many sensors that collect data on all parts of the assembly line. The

significance of this data is primarily to be able to alert someone to a possible upcoming problem (through noticing a bad trend) or to a current problem (by drawing attention to a metric that has exceeded some designated level); and much of it does not need to be kept for a long period of time. Often this type of data loses its importance once its timeliness expires and its main usefulness may just be in providing a sample measurement that can be used for historical records. Fast-moving data such as this is called streaming data, and Splunk, with its ability to create alerts, allows organizations to use this data to make sure they prevent, or act quickly on, problems that can occur.

## Latency of data

The term **latency**, in regards to data, refers to the delay in how speedily it is entered into the system for analysis. Splunk is able to analyze data in real time with no latency issues when deployed on hardware that is sufficient to handle the indexing and searching workload. For example, if an alert goes off, a system can be immediately shut down if there is no latency in the data. If a denial of a service attack (a cyberattack that can dramatically hurt an e-commerce company's bottom line) is taking place, Splunk can be quickly used to figure out what is happening almost immediately.

## Sparseness of data

Splunk is also excellent for dealing with sparse data. Much data in retailing environments is considered sparse. Consider a store that has many products but where most people just buy a few of them on any given shopping trip. If the store's database has fields specifying how many items of a particular type have been purchased by each customer, most of the fields would be empty if the time interval under consideration was short. We would say then that the data is sparse. In Splunk, the sparseness of data in a search ranges from dense (meaning that a result is obtained 10 percent of the time or more) to sparse (from 0.01 to 1 percent of the time). This can also extend to super sparse, or, for a better definition, trying to find a needle in a haystack (which is less than 0.01 percent), and even to rare, which is just a handful of cases.

# Splunk data sources

Splunk was invented as a way to keep track of and analyze machine data coming from a variety of computerized systems. It is a powerful platform for doing just that. But since its invention, it has been used for a myriad of different data types, including machine data, log data (which is a type of machine data), and social media data. The various types of data that Splunk is often used for are explained in the next few sections.

## Machine data

As mentioned previously, much of Splunk's data is machine data. Machine data is data created each time a machine does something, even if it is as seemingly insignificant as a tick on a clock. Each tick has information about its exact time (down to the second) and source, and each of these becomes a field associated with the event (the tick). The term "machine data" can be used in reference to a wide variety of data coming from computerized machines, from servers to operating systems to controllers for robotic assembly arms. Almost all machine data includes the time it was created or when the actual event took place. If no timestamp is included, then Splunk will need to find a date in the source name or filename based on the file's last modification time. As a last resort, it will stamp the event with the time it was indexed into Splunk.

## Web logs

Web logs are invaluable sources of information for anyone interested in learning about how their website is used. Deep analysis of web logs can answer questions about which pages are visited most, which pages have problems (people leaving quickly, discarded shopping carts, and other aborted actions), and many others. Google, in early 2014, was registering as many as 20 billion websites each day; you can find more information about this at [http://www.roche.com/media/roche\\_stories/roche-stories-2014-01-22.htm](http://www.roche.com/media/roche_stories/roche-stories-2014-01-22.htm).

## Data files

Splunk can read in data from basically all types of files containing clear data, or as they put it, any data. Splunk can also decompress the following types of files: `tar`, `gz`, `bz2`, `tar.gz`, `tgz`, `tbz`, `tbz2`, `zip`, and `z`, along with many others.

## Social media data

An enormous amount of data is produced by social media every second. Consider the fact that 1.13 billion people (<https://zephoria.com/top-15-valuable-facebook-statistics/>) login to Facebook each day and they spend, on average, 20 minutes at a time interacting with the site. Any Facebook (or any other social media) interaction creates a significant amount of data, even those that don't include the more data-intensive acts, such as posting a picture, audio file, or a video. Other social media sources of data include popular sites such as Twitter, LinkedIn, Pinterest, and Google+ in the U.S., and QZone, WeChat, and Weibo in China. As a result of the increasing number of social media sites, the volume of social media data created continues to grow dramatically each year.

## Other data types

Almost any type of data works in Splunk. Some of these types include scripted inputs and modular inputs. Sometimes you may want to include a script that sets data up so that it is indexed the way you want. Or you may want to include data coming from a source that is unusual in some way, and you want to make sure that the fields are set up the way they should be. For these reasons, it is nice to know that you can use Python scripts, Windows batch files, shell scripts, and other utilities to make sure your inputs are formatted correctly. You will see the other data types listed when we add data to Splunk shortly.

# Creating indexes

Indexes are where Splunk Enterprise stores all the data it has processed. It is essentially a collection of databases that is, by default, located at `$SPLUNK_HOME/var/lib/splunk`. Before data can be searched, it needs to be indexed, a process we describe here.

There are two ways to create an index, through the Splunk portal or by creating an `indexes.conf` file. You will be shown here how to create an index using the Splunk portal, but you should realize that when you do that, it simply generates an `indexes.conf` file.

You will be creating an index called `wineventlogs` to store Windows Event Logs. To do this, take the following steps:

1. In the Splunk navigation bar, go to **Settings**.
2. In the **Data** section, click on **Indexes**, which will take you to the Indexes page.
3. Click on **New Index**.
4. Now fill out the information for this new index as seen in the following screenshot, carefully going through steps 1 to 4.
5. Be sure to **Save** when you are done.

New Index

Index Name \*  Step 1, create the index name.  
Set index name (e.g., INDEX\_NAME). Search using index=INDEX\_NAME.

Home Path   
Hot/warm db path. Leave blank for default (\$SPLUNK\_DB/INDEX\_NAME/db).

Cold Path   
Cold db path. Leave blank for default (\$SPLUNK\_DB/INDEX\_NAME/colddb).

Thawed Path   
Thawed/resurrected db path. Leave blank for default (\$SPLUNK\_DB/INDEX\_NAME/thaweddb). Step 3, change to MB.

Max Size of Entire Index \*  Step 2, change the max size.  Maximum target size of entire index.

Max Size of Hot/Warm/Cold Bucket \*   Maximum target size of buckets. Enter 'auto\_high\_volume' for high-volume indexes.

Frozen Path   
Frozen bucket archive path. Set this if you want Splunk to automatically archive frozen buckets.

App  Step 4, assign it to the Destinations app.

You will now see the new index in the list as shown here:

wineventlog	Edit Delete Disable	destinations	1 MB	100 MB	0
-------------	---------------------	--------------	------	--------	---

The preceding steps have created a new `indexes.conf` file. Now go ahead and inspect this file. In your administrator command prompt (assuming that you are at the root of the `c:` drive), type the following command:

```
C:\> notepad  
c:\splunk\etc\apps\destinations\local\indexes.conf
```

Every index has specific settings of its own. Here is how your index looks when automatically configured by the portal. In production environments, this is how Splunk administrators manage the indexes. Note that the max size value of 100 that you specified is also indicated in the configuration.

```
[wineventlogs]  
coldPath = $SPLUNK_DB\wineventlogs\colddb  
homePath = $SPLUNK_DB\wineventlogs\db  
maxTotalDataSizeMB = 100  
thawedPath = $SPLUNK_DB\wineventlogs\thaweddb
```

The complete `indexes.conf` documentation can be found at <http://docs.splunk.com/Documentation/Splunk/latest/admin/indexesconf>.

# Buckets

You may have noticed that there is a certain pattern in this configuration file, in which folders are broken into three locations: `coldPath`, `homePath`, and `thawedPath`. This is a very important concept in Splunk. An index contains compressed raw data and associated index files that can be spread out into age-designated directories. Each piece of this index directory is called a **bucket**.

A bucket moves through several stages as it ages. In general, as your data gets older (think colder) in the system, it is pushed to the next bucket. And, as you can see in the following list, the thawed bucket contains data that has been resurrected from an archive. Here is a breakdown of the buckets in relation to each other:

- **hot**: This is newly indexed data and open for writing (`hotPath`)
- **warm**: This is data rolled from hot with no active writing (`warmPath`)
- **cold**: This is data rolled from warm (`coldPath`)
- **frozen**: This is data rolled from cold and deleted by default but it can be archived (`frozenPath`)
- **thawed**: This is data restored from the archive (`thawedPath`)

Now going back to the `indexes.conf` file, realize that the `homePath` will contain the hot and warm buckets, the `coldPath` will contain the cold bucket, and the `thawedPath` will contain any restored data from the archive. This means you can put buckets in different locations to efficiently manage disk space.

In production environments, Splunk admins never use the portal to generate indexes. They make their changes in the `indexes.conf` file. It is best practice to always use a temporary index for new data that you are unfamiliar with. Go ahead and create a new index stanza in `indexes.conf` with the following information:

```
[temp]
coldPath = $SPLUNK_DB\temp\colddb
homePath = $SPLUNK_DB\temp\db
maxTotalDataSizeMB = 100
```

```
thawedPath = $SPLUNK_DB\temp\thaweddb
```

1. Click on **Save**.
2. **Exit** the `indexes.conf` file when you are done.
3. Restart Splunk, which you must do for this change to take effect.
4. After restarting Splunk, go to the <http://localhost:8000/en-US/manager/destinations/data/indexes> page to confirm that the new index named `temp` has been created.



# Data inputs

As you may have noticed, any configuration you make in the Splunk portal corresponds to a `*.conf` file written to the disk. The same goes for the creation of data inputs; it creates a file called `inputs.conf`. Now that you have an index to store your machine's Windows Event Logs, let us go ahead and create a data input for it, with the following steps:

1. Go to the Splunk home page.
2. Click on your Destinations app. Make sure you are in the Destinations app before you execute the next steps.
3. In the Splunk navigation bar, select **Settings**.
4. Under the **Data** section, click on **Data inputs**.
5. On the **Data inputs** page, click on the **Local event log collection** type as shown in the following screenshot:



6. In the next page select the **Application** and **System** log types.
7. Change the index to `wineventlog`. Compare your selections with the following screenshot:

Available log(s) add all

- ☐ Application
- ☒ Security
- ☒ Setup
- ☐ System
- ☒ ForwardedEvents
- ☒ Els\_Hyphenation/Analytic
- ☒ EndpointMapper
- ☒ FirstUXPerf-Analytic
- ☒ AirSpaceChannel

Select the Windows Event Logs you want to index from the list.

Selected log(s) clear all

- ☒ Application
- ☒ System

**Index**

Set the destination index for this source.

Index  
wineventlogs

Cancel

8. Click **Save**.
9. On the next screen, confirm that you have successfully created the data input, as shown in the following screenshot:

Showing 1-1 of 1 item

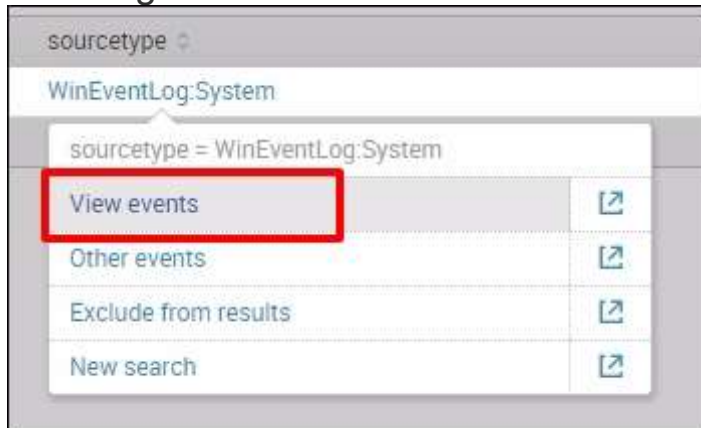
Event Log collection name	Log(s)	Host(s)
localhost	Application, System	localhost

Before we proceed further, let's make sure that the data input works. To do so, follow these steps:

1. Go to the Destinations search page: <http://localhost:8000/en-US/app/destinations/search>.
2. In the search window, run the following Splunk search query:
3. 

```
SPL> index=wineventlogs | top sourcetype
```

4. Check the search results to see that there are now two actively-indexed source types in your Splunk system, that is, **WinEventLog:System** and **WinEventLog:Application**.
5. View the events of each of these source types by clicking the source type name and selecting **View events** as shown in the following screenshot:



We have introduced a new concept here called **source types**. A source type is a type of classification of data that has been automatically made for you when you created the data input through the Splunk portal. In the same index, due to the steps we just took precedingly, Splunk internally classified the **System** and **Applications** event logs so you can access the events separately. There will be more about source types in Chapter 3, Search Processing Language.

Go ahead and inspect the `inputs.conf` file:

1. In an administrator prompt, open the file with this command:

```
C:\> notepad
c:\splunk\etc\apps\destinations\local\inputs.conf
```

2. Compare your results with this generated `inputs.conf` file:

```
[WinEventLog://Application]
checkpointInterval = 5
current_only = 0
disabled = 0
index = wineventlogs
start_from = oldest

[WinEventLog://System]
```

```
checkpointInterval = 5
current_only = 0
disabled = 0
index = wineventlogs
start_from = oldest
```

You can add directly into the `inputs.conf` file by following the same format. Use the temp index for testing. Here is an example input file based on a log file that is generated by the Splunk application. Note that this is going to be redundant to what is already being ingested through the `_internal` index and will only serve as an example.

```
[monitor://C:\Splunk\var\log\splunk\eventgen.log]
disabled = 0
sourcetype = eventgen
index = temp
```

Changes to the `inputs.conf` may require a Splunk restart. You can access that information with the following search command:

```
SPL> index=temp sourcetype=eventgen
```

The complete documentation for the `inputs.conf` file can be found at <https://docs.splunk.com/Documentation/Splunk/latest/Admin/Inputsconf>.

If you closely followed the instructions in these guides, you should now have exactly four data sources in your very own Splunk system that will be used in the remainder of the class. This is how you can query raw data from these data sources.

# Splunk events and fields

All throughout this chapter you have been running Splunk search queries that have returned data. It is important to understand what events and fields are before we go any further, for an understanding of these is essential to comprehending what happens when you run Splunk on the data.

In Splunk, a single piece of data is known as an **event** and is like a record, such as a log file or other type of input data. An event can have many different attributes or fields or just a few attributes or fields. When you run a successful search query, you will see that it brings up events from the data source. If you are looking at live streaming data, events can come in very quickly through Splunk.

Every event is given a number of default fields. For a complete listing, go to <http://docs.splunk.com/Documentation/Splunk/6.3.2/Data/Aboutdefaultfields>. We will now go through some of these default fields.

- **Timestamp:** A timestamp is applied at the exact time the event is indexed in Splunk. Usually, Splunk can figure out what timestamp to assign from the raw data it receives. For example, as a shopper clicks the final purchase button on an e-commerce website, data is collected about precisely when the sale occurred. Splunk can usually automatically detect this from the raw data.
- **Host:** The host field tells us what the hostname, IP address, or full domain name of the data is.
- **Index:** The index field describes where the event is located, giving the specific name of the index.
- **Source and sourcetype:** The source field tells us where the data came from, specifically the file, data stream, or other data input. The sourcetype is the format of the data input from which the data came. Common sourcetypes are `access_combined`, `access_custom`, and `cisco_syslog`.
- **Linecount:** The linecount is simply the number of lines contained in the event.

These default fields are name/value pairings that are added to events when Splunk indexes data. Think of fields as a quick way to categorize and group events. Fields are the primary constituents of all search queries. In later chapters you will learn more about fields and how to create custom fields from events.

## Extracting new fields

Most raw data that you will encounter will have some form of structure. Just like a CSV (comma-separated value file) or a web log file, it is assumed that each entry in the log corresponds to some sort of format. Splunk 6.3+ makes custom field extraction very easy, especially for delimited files. Let's take the case of our Eventgen data and look at the following example. If you look closely, the `_raw` data is actually delimited by white spaces:

```
2016-01-21 21:19:20:013632 130.253.37.97 GET /home - 80 -  
10.2.1.33 "Mozilla/5.0 (iPad; U; CPU OS 4_3_3 like Mac OS X;  
en-us) AppleWebKit/533.17.9 (KHTML, like Gecko)  
Version/5.0.2 Mobile/8J3 Safari/6533.18.5" 200 0 0 186 3804
```

Since there is a distinct separation of fields in this data, we can use Splunk's out-of-the-box field extraction tool to automatically classify these fields. In your Destinations app Search page, run the following search command:

```
SPL> index=main sourcetype=access_custom
```

This sourcetype `access_custom` refers to a type of file format that is generated by a server as it creates a web log file. After the data populates, click on the **Extract New Fields** link in the left column of the page as shown in the following screenshot:



In the resulting **Extract Fields** page, select one of the events that is shown in the `_raw` events area. Try to select an entry with the longest text. As soon as you do this, the text will appear highlighted at the top of the page, as per the following screenshot:

Extract Fields

Select sample

Select method

Select fields

Save

Next >

Existing fields >

### Select Sample Event

Choose a source or source type, select a sample event, and click Next to go to the next step. The field extractor will use the event to extract fields. [Learn more](#)

[I prefer to write the regular expression myself >](#)

Source type `access_custom`

```
2016-07-19 05:42:34:656523,164.218.0.0,GET,/destination/HOU/details,-,80,-,10.2.1.33,Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML; like Gecko) Chrome/30.0.1599.66 Safari/537.36,302,0,0,672,2162
```

Click on the **Next** button to proceed. In the page that appears, click on the **Delimiters** icon as indicated in the following screenshot:

(.\*?)

Regular Expression

Splunk Enterprise will extract fields using a Regular Expression.

x|y|z

Delimiters

Splunk Enterprise will extract fields using a delimiter (such as commas, spaces, or characters). Use this method for delimited data like comma-separated values (CSV files).

Click on **Next**. On the next page, click on the **Comma** delimiter as shown in the following screenshot:





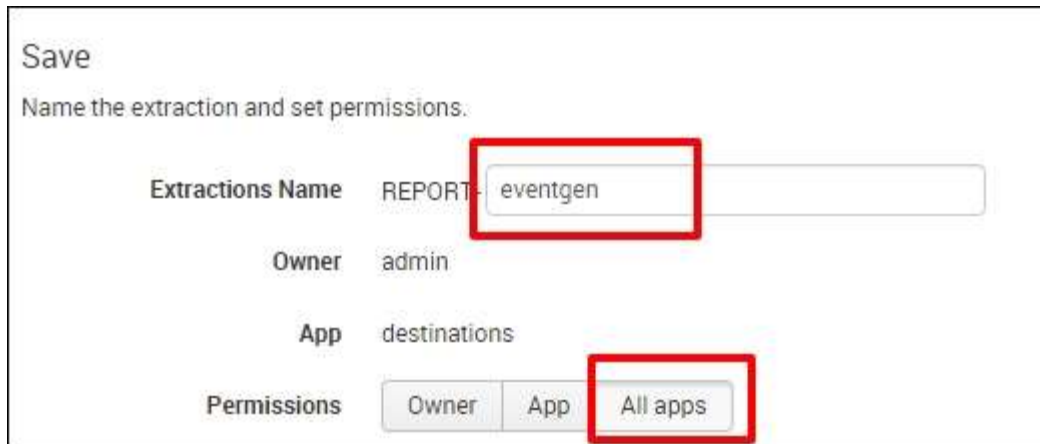
As soon as you select the **Comma** delimiter, Splunk will automatically allow you to modify each field and add a label to it. Click on the pencil icon for each field to input the label. When you're done, click on the **Rename Field** icon. For example, to edit **field3**, use the following screenshot:



Do the same for the remaining fields using the following guide. These fields will be needed in future chapters. You can skip those that are not included in the following list:

- **field1:** `datetime`
- **field2:** `client_ip`
- **field3:** `http_method`
- **field4:** `http_uri`
- **field8:** `server_ip`
- **field9:** `http_user_agent`
- **field10:** `http_status_code`
- **field14:** `http_response_time`

When you have completed the preceding task, click on **Next** to proceed. In the next window, label the **Extractions Name** as **eventgen** and select the **All apps** permission type. Refer to the following screenshot:



The screenshot shows a 'Save' dialog box with the title 'Save' and the instruction 'Name the extraction and set permissions.' Below this, there are four rows of configuration options: 'Extractions Name' with a value of 'REPORT- eventgen' (highlighted with a red box), 'Owner' with a value of 'admin', 'App' with a value of 'destinations', and 'Permissions' with three buttons: 'Owner', 'App', and 'All apps' (highlighted with a red box).

Click on **Finish** to complete the process. Now that you have extracted new fields, these will now be readily available in your search queries as exemplified below. In the next chapter, you will be shown how to use these new fields to filter search results. An example of the kind of search you can do on them is shown here:

```
SPL> index=main | top http_uri
```

If you want to go ahead and try this out now, just to prove that you have already made changes that will help you to understand the data you are bringing in, be our guest!

## Summary

In this session, we learned about some terms that need to be understood about big data, such as what the terms streaming data, data latency, and data sparseness mean. We also covered the types of data that can be brought into Splunk. Then we studied what an index is, made an index for our data, and put in data from our Destinations app. We talked about what fields and events are. And finally, we saw how to extract fields from events and name them so that they can be more useful to us. In the chapters to come, we'll learn more about these important features of Splunk.