Getting to know your data

How effective the searches you create in Splunk Enterprise almost always depends on your particular dataset. You rely heavily on technical add-ons (TAs) to help you make sense of your data, but this dependency has many caveats. For example, sometimes

- there is no TA
- the log format changed and doesn’t match the TA anymore
- the fields aren’t CIM compliant
- you’re developing a custom use case
- you have a custom log source

To become proficient with Splunk, you need to understand your datasets.

Get to know a dataset

This example walks you through how to understand a dataset, in this case, Sysmon Operational data. For this example, the first line of SPL will always be the index and the source type. You will need to change these, and other fields, to fit the dataset you are working with.

```
index=<index name> sourcetype=XmlWinEventLog:Microsoft-Windows.Sysmon/Operational
```

1. Import your dataset and get set up.
   - use verbose mode
   - use a static time range so you can rerun your search quickly and troubleshoot without the data changing
   - use a small but representative data sample (you will expand your dataset later to validate your work)
   - have the applicable admin guide for your data source and the CIM data model documentation available for reference

2. Remove Splunk noise from the data.

   ```
   | fields - date* time*pos index source sourcetype splunk_server linecount punct tag tag::eventtype eventtype vendor_product
   ```

3. Create a table of the fields in your data.

   ```
   | fields - date* time*pos index source sourcetype splunk_server linecount punct tag tag::eventtype eventtype vendor_product
   | table action Computer direction dvc dvc_nt_host EventChannel EventCode EventDescription EventID host Keywords Level Opcode process RecordID SecurityID signature signature_id Task TimeCreated UtcTime Version
   ```

4. Review the table to determine which fields appear in 100 percent of your data and which fields are most useful to you. Use your expertise and admin guide to make this determination. Remove the remaining fields.
If you aren't sure whether you removed the right fields, run a separate search with the fields you removed and the stats command shown here. Review the results to validate your decision. Note that high cardinality fields can crash your browser.

```
index=<index name> sourcetype=XmlWinEventLog:Microsoft-Windows-Sysmon/Operational
| fields - date* time*pos index source sourcetype splunk_server linecount punct tag tag::eventtype
eventtype vendor_product
| table EventChannel EventID Keywords Level Opcode Task Version
| stats values(*) as *
```

5. Keep refining using the fields, table, stats commands. Look for fields that have duplicate, similar, or related values. In this sample Sysmon data set, Computer, dvc, dvc_nt_host, and host all contained the host name. Those fields need to be normalized.

6. Start normalizing those duplicate values. The `eval` command will be the most useful to you in order to accomplish this. Other useful examples are shown in the following sample SPL.

```
rename dvc_nt_host AS dest_nt_host
| rex field=dvc "\[\w\d\].(?<dest_nt_domain>\.*\)$"
| fields - Computer dvc host
```

7. Spot check and validate your normalization. Some useful commands for doing so are:
   - `stats values()`
   - `where isnull()`
   - `where isnotnull()`
   - `dedup`

```
rename dvc_nt_host AS dest_nt_host
| rex field=dvc "\[\w\d\].(?<dest_nt_domain>\.*\)$"
| fields - Computer dvc host
| where isnull('dest_nt_host') OR 'dest_nt_host'=="
| dedup
| where isnotnull('dest_nt_host') OR NOT 'dest_nt_host'=="
```

8. Iterate and adjust your search as needed. This requires a little bit of art and a little bit of science, depending on the dataset and your needs, but usually involves the following tasks:
   - Find, combine, extract, and normalize fields and values.
   - Keep getting rid of fields that aren’t really helpful.
   - Start to look at fields with 90% coverage, then 50%.
index=<index name> sourcetype=XmlWinEventLog:Microsoft-Windows-Sysmon/Operational
| fields - date* time*pos index source sourcetype splunk_server linecount punct tag tag::eventtype
eventtype vendor_product
| fields - EventChannel EventID Keywords Level Opcode Task Version
| rename dvc_nt_host AS dest_nt_host
| rex field=dvc "\d\d\d\d\d\.(?<dest_nt_domain>.*$"
| fields - Computer dvc host
| fields - dest_nt_host dest_nt_domain
| table action direction EventCode EventDescription process RecordID SecurityID signature signature_id
| table EventCode EventDescription signature signature_id

9. When you are finished, clean up your search so you end up with a usable base search. To do so, you will need to:
   ◦ Combine commands.
   ◦ Sort commands.
   ◦ Remove commands.
   ◦ Table remaining fields with the most important data to the left:
     ▪ timestamp
     ▪ system generating event
     ▪ type of event
     ▪ outcome of event
     ▪ remaining contextual information

| rename dvc_nt_host AS dest_nt_host Initiated AS initiated ParentCommandLine AS parent_cmdline
| CurrentDirectory AS current_dir SHA256 AS sha256 Signature AS image_signature LogonId AS logon_id
| ImageLoaded AS image_loaded
| rex field=dvc "\d\d\d\d\d\.(?<dest_nt_domain>.*$"
| eval src=mvdedup(mvsort(lower(mvappend('src','src_host','src_ip'))))
| eval dest=mvdedup(mvsort(lower(mvappend('dest','dest_host','dest_ip'))))
| eval process=if(isnull('app'),'process','app')
| eval protocol=mvdedup(mvsort(lower(mvappend('protocol','SourcePortName','DestPortName'))))
| table _time dvc signature_id signature process_id process direction user logon_id initiated src src_port
| protocol transport dest dest_port image_loaded image_signature current_dir parent_process_id parent_
| process parent_cmdline cmdline file_path sha256

10. Validate your new base search.
   ◦ Expand your time range.
   ◦ Expand your data sample.
   ◦ Do you have null or empty values?
   ◦ Do any of your fields have unexpected values?
   ◦ Is your regex still working for every event?

| table _time dvc signature_id signature process_id process direction user logon_id initiated src src_port
| protocol transport dest dest_port image_loaded image_signature current_dir parent_process_id parent_
| process parent_cmdline cmdline file_path sha256
| where isnull("field") OR 'field'=""

-------------------------------
| where isnotnull("field") OR NOT 'field'=""
11. Put the normalization into a props.conf and publish the base search for your end users, preferably using a macro.

```bash
| stats values(*) as *

index=<index name> sourcetype=XmlWinEventLog:Microsoft-Windows-Sysmon/Operational
| table _time dvc signature_id signature process_id process direction user logon_id initiated src src_port
| protocol transport dest dest_port image_loaded image_signature current_dir parent_process_id parent_process
| parent_cmdline cmdline file_path sha256
```

**Next steps**

The content in this guide comes from a [Conf21 breakout session](#), one of the thousands of Splunk resources available to help users succeed. In addition, these Splunk resources might help you understand and implement this use case:

- Blog: [This is NOT the data you are looking for (OR is it)](#)
- Resource: [Splunk security dataset project](#)
- Blog: [Boss of the SOC v3 dataset released](#)
- Docs: [Overview of the Splunk Common Information Model](#)