Collecting logs in Splunk

Learn how to optimize data at the source so that Splunk can collect logs easier, faster, and more accurately.

Splunk does not need or require a logging standard. Splunk identifies an event using a few default fields from the incoming event's raw data, then identifies and correlates common elements with other events on the fly at search time. That means there is no fixed schema, which makes searching with Splunk fast, easy, and flexible.

However, you can optimize how data is formed at the source so that Splunk can parse event fields easier, faster, and more accurately when the events do arrive.

The article Logging best practices in the Splunk developer forum provides specific guidelines for forming events and creating operational guidelines. This article provides more general considerations for your logging practices.

Audience

- Architect
- Developer
- Search expert
- Knowledge manager

For more about these roles, see Setting roles and responsibilities.

Guidelines for logging best practices

"If you can read it, you can Splunk it."

If the meaning is not codified in the log events, then it needs to be added, either when the log is created (optimize log files at the source), or on the fly (using fields in Splunk).

Semantic logging is writing event logs explicitly for gathering analytics that will be consumed and processed by software. Logs are generally written by developers to help them debug, or to form an audit trail, so they are often cryptic or lack the detail needed for data analysis.

Only optimize event logs if it is practical

Optimizing your logs at the source is not necessary or required, but it can streamline your Splunk experience. Optimizing event logs makes the most sense for systems in active use whose source code is easily accessible. It may not be practical to try rewriting the logs for a legacy application whose source code is no longer available. A better approach in that case is to use Splunk knowledge objects to add meaning to existing log information.
Capture data from a variety of sources

Think of a business scenario you might want to analyze. Think about what elements you would want to visualize, and consider what data might be needed to help answer basic questions about that scenario. For example:

- Graph transaction volume by hour, by day, by month
- How long are transactions taking during different times of the day and different days of the week?
- Are transactions taking longer than they did last month?
- What volume of transactions come from which geographical regions?
- How many transactions are failing? Graph these failures over time.
- Which specific transactions are failing?

To begin answering these questions, consider all the systems involved in a business transaction workflow. The more varied your sources are, the better your correlation will be. For example:

- Application logs
- Database logs
- Network logs
- Configuration files
- Cron jobs and other scheduled tasks
- Performance data (CPU, disk, memory, and so on)

Treat your data source as part of your development software stack

Work with your development team to establish detailed, organized, human-readable logs.

- Encourage development teams to create tags and notations in logs for easier identification
- Include creating custom reports, dashboards, and alerts in each application backlog
- Build analytics to support all code as part of its delivery criteria before releasing it

Practice good log file management

Log locally to files to create a persistent record. This avoids any data gap when there is a Splunk restart.