Custom application and debug data

Best practices for application developers require the inclusion of debugging code in applications that can be enabled to provide minute details of application state, variables and error conditions or exceptions. Debug output is typically logged for later analysis that can expose the cause of application crashes, memory leaks, performance degradation and security holes. Furthermore, since the events that cause a security or performance problem may be spaced over time, logs—along with the problem software—can help correlate and trace temporally separated errors to show how they contribute to a larger problem. Due to the performance overhead and amount of data produced, debug logs typically are enabled only when a problem can’t be identified via test or event logs.

Application debug logs provide a record of program behavior that is necessary to identify and fix software defects, security vulnerabilities or performance bottlenecks. While test logs record the output results of application usage, debug logs provide information about an application’s internal state, including the contents of variables, memory buffers and registers; a detailed record of API calls; and even a step-by-step trace through a particular module or subroutine.

Application

When your Splunk deployment is ingesting custom application and debug data, you can use it to accomplish security and compliance, application delivery, and IT ops use cases.

- Debugging frontend errors

Sources

Guidance for onboarding data can be found in the Splunk Documentation, Getting Data In (Splunk Enterprise) or Getting Data In (Splunk Cloud). The source types depend on the custom applications in use in your organization.