The Essential Guide to

User and Application Data

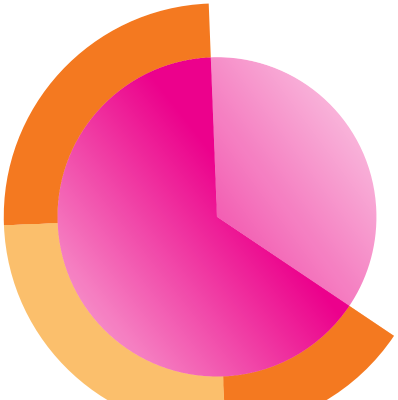
splunk>

turn data into doing™

Time-Series Data. Streaming Data. Dark Data.

It's no secret that data remains underused and undervalued in most organizations all over the world. Despite the constant talk of data-driven decisions, organizations of all sizes are still missing the mark on how to effectively capture and use the troves of data being generated every day, whether it comes from users, outside industry resources, or their own networked devices. In fact, most business and IT decision makers estimate that **55% of their data is dark data**, information you don't know you have, or can't fully tap.

This is a big missed opportunity. Important insights across IT, security and your organization lie hidden in this data. Data holds the definitive record of all activity and behavior of your customers and users, transactions, applications, servers, networks, mobile devices and more. Critical information on everything from configurations, APIs, message queues, diagnostic outputs, sensor data of industrial systems and more is all there — you just have to tap into it the right way.



With the right approach, data makes it simple to:

- Make better informed decisions about every part of your business.
- Run your operations more efficiently.
- Optimize user and customer experiences.
- Detect the fingerprints of fraud — or prevent it altogether.
- Uncover potential disasters before they happen.
- Find hidden trends that help your company leapfrog the competition.
- Make everyone who uses it look like a hero.
- … and so much more.

The challenge with leveraging the vast quantity of data that most companies collect is that it comes in a dizzying range of formats that traditional data monitoring and analysis tools aren't designed to handle. Many tools can't keep up with the varying data structures, sources or time scales. And it goes well beyond just machine data as well. But the upside to tapping into your data is tremendous, and this is where Splunk comes in.

With Splunk, you can bring data to every question, decision and action in your organization to create meaningful outcomes. Unlike any other platform, Splunk is truly able to take any data from any source and drive real action to benefit the business — from IT infrastructure and security monitoring to DevOps and application performance monitoring and management.

Data-to-Everything in Practice

Use data to:

Investigate

Monitor

Analyze

Act

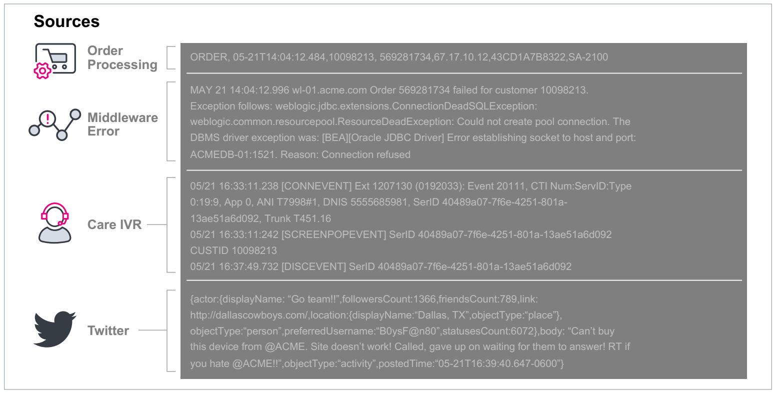
The organizations that get the most value out of their data are those able to take disparate data types, enrich them and extract answers. But not knowing what data to ingest can stop businesses before they start.

Familiarizing yourself with general use cases in security, IT operations, business analytics, DevOps, the Internet of Things (IoT) and more — including the data types and sources involved — can get you on track right away.

Here's an example:

1. A customer's order didn't go through
2. The customer called support to resolve the issue
3. After too much time on hold, the customer gave up and tweeted a complaint about the company

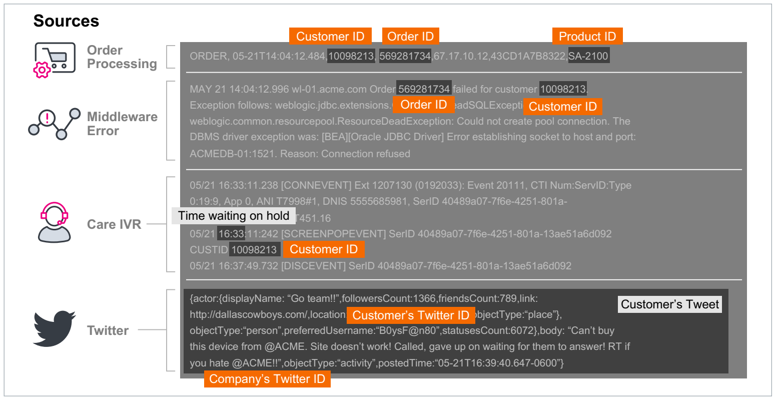
What Does Machine Data Look Like?



```
ORDER, 05-21T14:04:12.484,10098213, 569281734,67.17.10.12,43CD1A7B8322,SA-2100

MAY 21 14:04:12.996 wl-01.acme.com Order 569281734 failed for customer 10098213.
Exception follows: weblogic.jdbc.extensions.ConnectionDeadSQLException:
weblogic.common.resourcepool.ResourceDeadException: Could not create pool connection. The
DBMS driver exception was: [BEA][Oracle JDBC Driver] Error establishing socket to host and port:
ACMEDB-01:1521. Reason: Connection refused

05/21 16:33:11.238 [CONNEVENT] Ext 1207130 (0192033): Event 20111, CTI Num:ServID:Type
0:19:9, App 0, ANI T7998#1, DNIS 5555685981, SerID 40489a07-7f6e-4251-801a-
13ae51a6d092, Trunk T451.16
05/21 16:33:11:242 [SCREENPOPEVENT] SerID 40489a07-7f6e-4251-801a-13ae51a6d092
CUSTID 10098213
05/21 16:37:49.732 [DISCEVENT] SerID 40489a07-7f6e-4251-801a-13ae51a6d092

{actor:{displayName: "Go team!!",followersCount:1366,friendsCount:789,link:
http://dallascowboys.com/,location:{displayName:"Dallas, TX",objectType:"place"},
objectType:"person",preferredUsername:"B0ysF@n80",statusesCount:6072},body: "Can't buy
this device from @ACME. Site doesn't work! Called, gave up on waiting for them to answer! RT if
you hate @ACME!!",objectType:"activity",postedTime:"05-21T16:39:40.647-0600"}
```

Figure 1: Data can come from any number of sources, and at first glance, can look like random text.

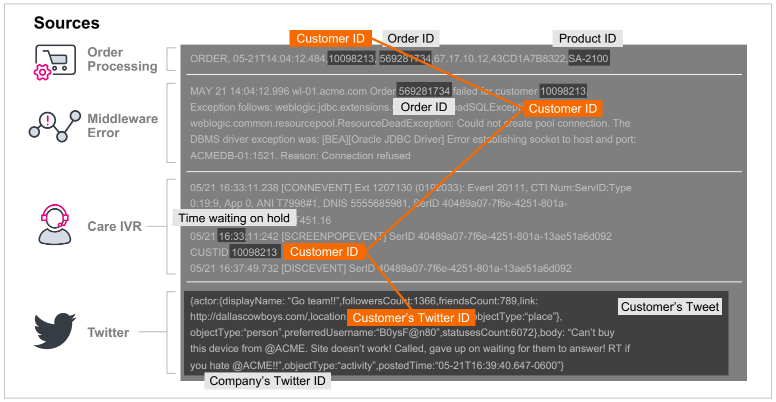
Machine Data Contains Critical Insights



```
ORDER, 05-21T14:04:12.484,10098213, 569281734,67.17.10.12,43CD1A7B8322,SA-2100

MAY 21 14:04:12.996 wl-01.acme.com Order 569281734 failed for customer 10098213.
Exception follows: weblogic.jdbc.extensions.      eadSQLExcepti
weblogic.common.resourcepool.ResourceDeadException: Could not create pool connection. The
DBMS driver exception was: [BEA][Oracle JDBC Driver] Error establishing socket to host and port:
ACMEDB-01:1521. Reason: Connection refused

05/21 16:33:11.238 [CONNEVENT] Ext 1207130 (0192033): Event 20111, CTI Num:ServID:Type
0:19:9, App 0, ANI T7998#1, DNIS 5555685981, SerID 40489a07-7f6e-4251-801a-
T451.16
05/21 16:33:11:242 [SCREENPOPEVENT] SerID 40489a07-7f6e-4251-801a-13ae51a6d092
CUSTID 10098213
05/21 16:37:49.732 [DISCEVENT] SerID 40489a07-7f6e-4251-801a-13ae51a6d092

{actor:{displayName: "Go team!!",followersCount:1366,friendsCount:789,link:
http://dallascowboys.com/,location      bjectType:"place"},
objectType:"person",preferredUsername:"B0ysF@n80",statusesCount:6072},body: "Can't buy
this device from @ACME. Site doesn't work! Called, gave up on waiting for them to answer! RT if
you hate @ACME!!",objectType:"activity",postedTime:"05-21T16:39:40.647-0600"}
```

Figure 2: The value of data is hidden in this seemingly random text.

Machine Data Contains Critical Insights



```
ORDER, 05-21T14:04:12.484,10098213, 569281734,67.17.10.12,43CD1A7B8322,SA-2100

MAY 21 14:04:12.996 wl-01.acme.com Order 569281734 failed for customer 10098213.
Exception follows: weblogic.jdbc.extensions.             eadSQLExcep
weblogic.common.resourcepool.ResourceDeadException: Could not create pool connection. The
DBMS driver exception was: [BEA][Oracle JDBC Driver] Error establishing socket to host and port:
ACMEDB-01:1521. Reason: Connection refused

05/21 16:33:11.238 [CONNEVENT] Ext 1207130 (0192033): Event 20111, CTI Num:ServID:Type
0:19:9, App 0, ANI T7998#1, DNIS 5555685981, SerID 40489a07-7f6e-4251-801a-
                    T451.16
05/21 16:33:11:242 [SCREENPOPEVENT] SerID 40489a07-7f6e-4251-801a-13ae51a6d092
CUSTID 10098213
05/21 16:37:49.732 [DISCEVENT] SerID 40489a07-7f6e-4251-801a-13ae51a6d092

{actor:{displayName: "Go team!!",followersCount:1366,friendsCount:789,link:
http://dallascowboys.com/,location                     objectType:"place"},
objectType:"person",preferredUsername:"B0ysF@n80",statusesCount:6072},body: "Can't buy
this device from @ACME. Site doesn't work! Called, gave up on waiting for them to answer! RT if
you hate @ACME!!",objectType:"activity",postedTime:"05-21T16:39:40.647-0600"}
```

Figure 3: By correlating different types of data together, you can start to gain real insight into what's going on in your infrastructure, see security threats or even use the insights to drive better business decisions.

By taking all the data involved in the process — i.e., pulling information from order processing, middleware, interactive voice response systems and Twitter — an organization can get a full view of the customer experience problem.

User and Application Data

This book provides a high-level overview into data created by applications and users as they operate. This data can support a variety of use cases, including troubleshooting login issues, identifying anomalous login behavior and DevOps.

While each organization's needs and data sources will vary by vendor, product and infrastructure, this book details where you should look for this type of machine data and the value it can provide to IT, security, IoT and business analytics use cases.

Many of the data sources listed in this book can support multiple use cases — this is a major part of what drives data's tremendous value.

Security and Compliance

IT Ops, App Delivery and DevOps

Internet of Things

Business Analytics

Table of Contents

User Data **6**

Virtual Private Networks (VPN) 6

Authentication Data 7

Application Data **8**

Antivirus 8

Application Performance Management (APM) Tool Data 9

Automation, Configuration, Deployment Tools (Platforms) 9

Binary Repositories 10

Build Systems (Platforms) 10

Code Management 11

Container Logs and Metrics 11

Container Orchestration Metrics 12

CRM, ERP and Other Business Applications 13

Custom Application and Debug Logs 14

Distributed Tracing Tools 15

Mail Server 16

Test Coverage Tools 16

Serverless Monitoring 17

Vulnerability Scanning 18

User Data

Virtual Private Networks (VPN)

Use Cases: Security and Compliance

Examples: Citrix NetScaler Nitro, Citrix NetScaler IPFIX, Cisco

Virtual private networks (VPNs) are a way of building a secure extension of a private network over an insecure, public one. VPNs can be established either between networks, routing all traffic between two sites, or between a client device and a network. Network-to-network VPNs typically are created using strong credentials such as certificates on each end of the connection. Client-to-network VPNs rely on user authentication, which can be as simple as a username and password. VPNs use network tunneling protocols such as IPSec, OpenVPN plus SSL or L2TP with cryptographically strong algorithms to scramble information in transit and ensure end-to-end data integrity.

Use Cases

Security and Compliance: VPN logs help analyze users coming onto the network. This information can be used in a number of ways, including situational awareness, monitoring foreign IP subnets, and compliance monitoring of browsers and applications of connected hosts. VPN data can also help identify:

- Activities from different locations, such as changes in location within a given amount of time.
- Access from risky countries or locations.
- User sessions at odd times, such as late evenings or weekends.
- User land speed violations.
- Abnormal frequency of sessions based on each user profile.

Authentication Data

Use Cases: Security and Compliance, IT Operations, Application Delivery

Examples and Data Sources: Active Directory, LDAP, Identity Management, Single-Sign On

Authentication data provides insight into users and identity activity. Common authentication data sources include:

- **Active Directory:** A distributed directory in which organizations define user and group identities, security policies and content controls.
- **LDAP:** An open standard defined by the Internet Engineering Task Force (IETF) and is typically used to provide user authentication (name and password). It has a flexible directory structure that can be used for a variety of information such as full name, phone numbers, email and physical addresses, organizational units, workgroup and manager.
- **Identity Management:** Identity management is the method of linking the users of digital resources — whether people, IoT devices, systems or applications — to a verifiable online ID.
- **Single Sign-On (SSO):** A process of using federated identity management to provide verifiable, attestable identities from a single source to multiple systems. SSO significantly increases security by tying user credentials to a single source, allowing changes to user rights and account status to be made once, and reflected in every application or service to which the user has access. SSO is particularly important for users with elevated security rights such as system or network administrators that have access to a large number of systems.

Use Cases

Security and Compliance: For security, authentication data provides a wealth of information about user activity, such as multiple login failures or successes to multiple hosts in a given time window, activities from different locations within a given amount of time, and brute force activities. Specifically:

- Active Directory domain controller logs contain information regarding user accounts, such as privileged account activity, as well as the details on remote access, new account creation and expired account activity.
- LDAP logs include a record of who, when and where users log in to a system and how information is accessed.
- Identity Management data shows access rights by user, group and job title (e.g., CEO, supervisor or regular user). This data can be used to identify access anomalies that could be potential threats — for example, the CEO accessing a low-level networking device or a network admin accessing the CEO's account.

IT Ops and Application Delivery: Authentication data supports IT operations teams as they troubleshoot issues related to authentication. For example, application support can be tied to logins, enabling IT operations to see whether users are struggling to log in to applications. For IT operations teams that support Active Directory, logs can be used to troubleshoot and understand the health of Active Directory.

Application Data

Antivirus

Use Cases: Security and Compliance

Examples: Kaspersky, McAfee, Norton Security, F-Secure, Avira, Panda, Trend Micro

The weakest link in corporate security are individuals, and antivirus is one way to protect them from performing inadvertently harmful actions. Whether it's clicking on an untrustworthy web link, downloading malicious software or opening a booby-trapped document (often one sent to them by an unsuspecting colleague), antivirus can often prevent, mitigate or reverse the damage.

So-called advanced persistent threats (APTs) often enter through a single compromised machine attached to a trusted network. While not perfect, antivirus software can recognize and thwart common attack methods before they can spread.

Use Cases

Security and Compliance: Antivirus logs support the analysis of malware and vulnerabilities of hosts, laptops and servers; and can be used to monitor for suspicious file paths. It can help identify:

- Newly detected binaries, file hash, files in the filesystem and registries.
- When binaries, hash, or registries match threat intelligence.
- Unpatched operating systems.
- Known malware signatures.

Application Performance Management (APM) Tool Data

Use Cases: Security and Compliance, IT Operations, Application Delivery

Examples: Dynatrace, New Relic, AppDynamics, MMSoft Pulseway, LogicMonitor, Stackify, Idera, Ipswitch

Application performance management software provides end-to-end measurement of complex, multitier applications to provide performance metrics from an end user's perspective. APM logs also provide event traces and diagnostic data that can assist developers in identifying performance bottlenecks or error conditions. The data from APM software provides both a baseline of typical application performance and record of anomalous behavior or performance degradation. Carefully monitoring APM logs can provide an early warning to application problems and allow IT and developers to remediate issues before users experience significant degradation or disruption. APM logs also are required to perform post-hoc forensic analysis of complex application problems that may involve subtle interactions between multiple machines, network devices or both.

Use Cases

Security and Compliance: Security teams can use APM logs to perform post-hoc forensic analysis of incidents that span multiple systems and exploit vulnerabilities. The data can be used to correlate security indications between the system and application activities. It also helps to identify SQL/API calls/CMD made in relation to suspicious activity, or abnormal amounts of sessions or CPU load in relation to security activity.

IT Ops and Application Delivery: By providing end-to-end measurement of complex, multi tier applications, APM logs can show infrastructure problems and bottlenecks that aren't visible when looking at each system individually, such as slow DNS resolution causing a complex web app to bog down as it tries to access content and modules on many different systems.

Automation, Configuration, Deployment Tools (Platforms)

Use Case: Application Delivery and DevOps

Examples: Puppet Enterprise, Ansible Tower, Chef, SaltStack, Rundeck machine data ingested through APIs, webhooks or run logs

Automated configuration and deployment tools, also known as infrastructure as code, allow IT and DevOps practitioners to practice continuous application delivery in the cloud or on premises. When infrastructure is treated as code, it's easy to share, collaborate, manage version control, perform peer unit testing, automate deployments, check the status of deployment and more.

Tools like Rundeck are platforms that take automation frameworks like Salt Stack and enable teams to automate states or playbooks to make sure the code is released and reported back to a central reporting tool.

Use Cases

Application Delivery and DevOps: Automation and configuration machine data monitoring helps application delivery teams deliver applications faster without sacrificing stability or security.

Binary Repositories

Use Case: Application Delivery and DevOps

Examples: Data from Nexus, Artifactory, delivered through APIs, webhooks; Yum, Pacman and Aptly data delivered through logs

A binary repository is a tool for downloading and storing binary files used and created in software development. It's used to store software binary packages, artifacts and their corresponding metadata. They're different from source code repositories, as binary repositories do not store source files. Searching through these repositories is possible by analyzing associated metadata.

Use Cases

Application Delivery and DevOps: Analyzing binary repository data helps application delivery teams and release managers to ensure that the final deployment of code to production is successful.

Build Systems (Platforms)

Use Case: Application Delivery and DevOps

Examples: Jenkins, Bamboo, TravicCI, TeamCity machine data ingested through APIs, logs, webhooks

Build platforms, like Jenkins and Bamboo, enable a continuous integration practice that allows application delivery teams — including developers, DevOps practitioners, QA and release engineering — to build artifacts, trigger new builds and environments, automate tests and more.

Use Cases

Application Delivery and DevOps: Build systems monitoring helps release managers, test and QA teams understand the health of their build environment, the status of tests, get insights into stack traces and build queues. This visibility helps remediate build or test bottlenecks and increase the application delivery velocity and quality.

Code Management

Use Case: Application Delivery

Examples: Github, GitLab

For all but the most trivial implementations, application source code consists of dozens if not hundreds of interrelated files. The complexity and volatility of code — particularly when using agile development methodologies and changes are made daily — makes keeping track of it virtually impossible without a structured, automated source code management and revision control system.

Originally built as client-server applications where developers checked in code to a central repository, today's systems (such as Git) are often distributed, with each developer working from a local copy of the full repository and changes synchronized across all subscribers to a particular project. Code management systems provide revision control (the ability to back out changes to an earlier version), software build automation, configuration status records and reporting, and the ability to branch or fork all or part of a source-code tree into a separate subproject with its own versioning.

Use Cases

Application Delivery: The version records of code management can help IT operations teams identify application changes that are causing system problems, such as excessive resource consumption or interference with other applications.

Container Logs and Metrics

Use Case: Application Delivery and DevOps

Examples: Docker

Container logs are an efficient way to acquire logs generated by applications running inside a container. By utilizing logging drivers, output that is usually logged is redirected to another target. Since logging drivers start and stop when containers start and stop, this is the most effective way of capturing machine data, given the often limited lifespan of a container.

Container metrics contain details related to CPU, memory, I/O and network metrics generated by a container. By capturing this data, you have the opportunity to spot specific containers that appear to consume more resources than others — enabling faster, more precise troubleshooting.

Use Cases

Application Delivery and DevOps: Acquiring container log files gives developers and operations teams insight on errors, issues and availability of applications running inside containers. Logs and metrics at the container level also call attention to containers whose performance is outside of expected parameters. As a result, admins can "kill" or "stop" a container instance, and "run" a new container in its place.

Container Orchestration Metrics

Use Case: Application Delivery and DevOps

Examples: Kubernetes, Amazon ECS2, Azure Container Services, Docker Swarm, Google Container Engine

Container orchestration tools provide an enterprise-level framework for automating container deployments and integrating and managing containerized applications at scale. Container orchestration tools like Kubernetes are important for ensuring the speed, availability, scaling and networking of containerized environments. Like container metrics, it's important to collect container orchestration metrics at high-resolutions due to their self-healing, ephemeral nature.

The most popular container orchestration platform is Kubernetes. Kubernetes metrics contain details related to the inventory, health and performance of container resources (cluster maps, node state, pod status, container status, namespace status, workload deployments details, etc.) along with aggregated system metrics (CPU, disk, memory, network) across nodes. By visualizing and correlating this data, you have the opportunity to keep track of infrastructure inventory, capacity, and cost and investigate underlying issues across your Kubernetes environment leading to failures — expediting troubleshooting.

Use Cases

Application Delivery and DevOps: Acquiring Kubernetes metrics gives developers and operations teams insights across all layers of their Kubernetes environment and the underlying infrastructure. This broad view helps operators monitor and manage the health of containerized environments, oversee services migrating to Kubernetes, and quickly diagnose any issues with the infrastructure, the orchestration platform itself, or the container.

For example, operators can look into an under-performing pod then to the metrics for the workload running in that pod and view its neighbors allowing for more context than just container level metrics and logs. Since particular problems in container environments can often be hard to find, this context is critical for teams to correlate patterns — reducing mean time to clue and expediting root cause analysis. This is particularly helpful during troubleshooting when DevOps teams need to quickly pinpoint which service is causing a sudden spike in latency or error rate and why. This comprehensive view also assists with resource optimization and capacity planning.

CRM, ERP and Other Business Applications

Use Cases: Security and Compliance, Application Delivery, Business Analytics

Examples: SAP, SFDC, SugarCRM, Oracle, Microsoft Dynamics

Business Applications can create a wealth of data as part of normal operations. Two examples are CRM and ERP applications:

Customer relationship management (CRM) systems have become an essential part of every organization, providing a central database of all customer contact information, communications and transaction details. CRM systems have evolved from simple contact management systems to platforms for customer support and engagement by providing personalized sales and support information. The same customer support data repository can be used to develop customized marketing messages and sales promotions. CRM systems are also useful for application support and enhancement by recording details about customer problems with a particular system or application along with their eventual solution — details that can inform future application or service updates.

Enterprise resource planning (ERP) applications are a critical back-office IT service that provides systematic, automated collection and analysis of a variety of product, supply chain and logistics data. ERP is used in product planning, tracking purchases of components and supplies, inventory management, monitoring and regulating manufacturing processes, managing logistics, warehouse inventory and shipping, and to monitor and measure the effectiveness of sales and marketing campaigns.

ERP software also integrates with CRM, HR, finance/accounting/payroll and asset management systems, with bidirectional data flows that provide consistent information across back-end digital business processes. ERP systems are typically built on a relational database management system with a variety of modules and customizations for specific functions such as supplier relationship management or supply chain management. Due to their complexity, ERP systems often are installed and managed by product specialists.

Use Cases

Security and Compliance: CRM records can help security teams unravel incidents that involve multiple customers and problem episodes over a long time span. They can also provide evidence of a breach, should records be modified outside normal business processes. In addition, the data can be used to audit access records of customer or internal user information.

Application Delivery: CRM, ERP, and other business applications are often mission-critical systems that facilitate a variety of front and back office processes. The performance of these applications can impact internal operations. Business application logs can be used to determine the health of those operations.

Business Analytics: CRM, ERP, and other business applications facilitate a variety of front and back office processes that span other systems as well. As part of an end-to-end view of those complex business processes, business application data can help provide insights into the health of business operations.

Custom Application and Debug Logs

Use Cases: Security and Compliance, IT Operations, Application Delivery

Examples: Custom applications

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 causing 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.

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. 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.

Use Cases

Security and Compliance: Security breaches are often the result of improper handling of unexpected inputs, such as buffer overflow exploits or data injection used in cross-site scripting attacks. This type of low-level vulnerability is almost impossible to detect without logging the internal state of various application variables and buffers.

Similar to APM logs, custom application and debug logs can be used to correlate security indications between the system and application activities. It also helps to identify SQL/API calls/CMD made in relation to suspicious activity, or abnormal amounts of sessions or CPU load in relation to security activity.

IT Ops and Application Delivery: Debug output can expose application behavior that causes inefficient use of system resources or application failures that can be addressed by developers and operations teams. Debug output is useful for unraveling the internal state of an application that exhibits performance problems or has been shown to have security vulnerabilities, and the data can be helpful in identifying root cause.

Distributed Tracing Tools

Use Case: IT Operations, Application Delivery and DevOps

Examples: SignalFx, OpenTelemetry, Zipkin, Jaeger, fluentd

Distributed tracing is a method used to monitor how requests flow through your microservices applications by mapping transaction paths and duration as they propagate across services through trace and span data.

Popular open source distributed tracing instrumentation tools like OpenTelemetry record and publish operation data useful for finding sources of latency and errors within a distributed system — illuminating the relationship between user-visible behavior and the complex mechanics of the microservices underneath. APM software tools metricize information collected through these instrumentation tools to provide actionable insights on performance problems drilling down into specific service-level details.

Traces contain a lot of information about the method, operation, or block of code that it captures like the operation name, the start time of the operation, how long the operation took to execute, the logical name of the service on which the operation took place, the IP address of the service instance on which the operation took place, and trace context propagation. These are often represented as RED (Rate, Errors, Duration) metrics for monitoring purposes.

Distributed tracing along with APM Tools provide a context rich, complete view of service transactions that exist in complex distributed systems so IT and developers can understand user-visible latency, SLAs, and perform root-cause analysis with preserved traces that serve as anomaly benchmarks.

Use Cases

IT Operations, Application Delivery and DevOps: By providing end-to-end measurement of complex, multi-tier applications, tracing data can show microservices problems and bottlenecks that aren’t visible when looking at each application individually, especially through service mapping, such as slow DNS resolution causing a complex web app to bog down as it tries to access content and modules on many different systems.

Distributed tracing allows DevOps teams to see all traces and spans for an API call and fix underperforming APIs. This helps teams improve system performance in real-time, before downstream effects impact customers. APM tools can expose which transaction spans deviated from the norm while showing correlation to code and infrastructure for deeper root cause analysis and troubleshooting. Since teams can visualize tracing data in real-time, this information improves time to market by making it easy to immediately see how updates and rollouts to services impact applications.

Mail Server

Use Cases: Security and Compliance, IT Operations

Examples: Exchange, Office 365

Email remains the primary form of formal communication in most organizations. As such, mail server databases and logs are some of the most important business records. Due to their size and tendency to grow without bounds, email data management typically requires both data retention and archival policies so that only important records are held and inactive data is moved to low-cost storage.

Use Cases

Security and Compliance: Mail server data can help identify malicious attachments, malicious domain links and redirects, emails from known malicious domains, and emails from unknown domains. It can also be used to identify emails with abnormal or excessive message sizes, and abnormal email activities times.

IT Ops: Email messages and activity logs can be required to maintain compliance with an organization's information security, retention and regulatory compliance processes. Mail server transaction and error logs also are essential debugging tools for IT problem resolution and also may be used for usage-based billing.

Test Coverage Tools

Use Case: Application Delivery and DevOps

Examples: Static Analysis and Unit Testing logs (SonarQube, Tox, PyTest, RubyGem MiniTest, Bacon, Go Testing), build server logs and performance metrics

Typical test coverage includes functional, statement, branch and conditional coverage. The idea is to match what percentage of code can be exercised by a test suite of one or more coverage criteria. Coverage tests are usually defined by rule or requirements. In addition to coverage testing, software delivery teams can utilize machine data to understand the line count, code density and technical debt.

Use Cases

Application Delivery and DevOps: Test coverage data monitoring helps release managers, application owners and others understand:

- How much technical debt and issues are they resolving?
- How ready is their next release?
- From unit testing — how many tests were performed per hour and what tests are being run?

If test coverage data is combined with build data, release managers can start monitoring build and release performance and start understanding the release quality. They can understand the trends in error percentage and make decisions on if the build is ready for production. Understanding code quality can also help support teams get prepared for any additional volume of calls or any particular issues that may arise.

Serverless Monitoring

Use Case: Application Delivery and DevOps

Examples: AWS Lambda, Google Cloud Functions (GCF), Azure Functions, OpenShift Serverless

Event-driven, serverless computing platforms also known as functions-as-a-service (FaaS) allow IT and DevOps practitioners to practice continuous application delivery without the need to perform administrative tasks required to provision and manage infrastructure. With FaaS, developers write single-purpose functions that are triggered and scaled on demand by events emitted from services so teams can focus on writing and delivering business critical applications. It makes it easy to automate processes, control costs, autoscale services and APIs, and promote collaboration across teams writing specialized applications in different languages. However, the "statelessness" and ephemerality of functions make monitoring their performance almost impossible without real-time, contextual solutions.

Use Cases

Application Delivery and DevOps: Serverless monitoring helps DevOps teams, application owners and others understand:

- Availability of applications running on serverless with point in time information about current state of functions like average latency and total number of function cold starts.
- Usage on concurrency for availability and cost planning. Teams can increase the amount of concurrency during times of high demand and lower it, or completely turn it off, when demand decreases in real-time.
- Errors with visibility and insights into failed invocations so developers can remediate issues before users are impacted.
- Compute duration — time from when your function code starts executing as the result of an invocation to when it stops executing for deeper understanding into costs.
- How functions are supporting business and customer experience including user requests, checkout abandonment, revenue per location, etc.
- Trends and breakdowns of functions by account, region, etc. for deeper root cause analysis.

Data from functions can also be monitored via distributed tracing for granular visibility into the performance of serverless applications along with end-to-end transaction views into invocations of multiple functions and all services.

Vulnerability Scanning

Use Case: Security and Compliance

Examples: ncircle IP360, Nessus

An effective way to find security holes is to examine infrastructure from the attacker's point of view. Vulnerability scans probe an organization's network for known software defects that provide entry points for external agents. These scans yield data about open ports and IP addresses that can be used by malicious agents to gain entry to a particular system or entire network.

Systems often keep network services running by default, even when they aren't required for a particular server. These running, unmonitored services are a common means of external attack, as they may not be patched with the latest OS security updates. Broadscale vulnerability scans can reveal security holes that could be leveraged to access an entire enterprise network.

Use Cases

Security and Compliance: Vulnerability scans yield data about open ports and IP addresses that can be used by malicious agents to gain entry to a particular system or entire network. The data can used to identify:

- System misconfiguration causing security vulnerability.
- Outdated patches.
- Unnecessary network service ports.
- Misconfigured filesystems, users or applications.
- Changes in system configuration.
- Changes in various user, app or filesystem permissions.

About **Splunk.**

Splunk turns data into doing with the Data-to-Everything™ Platform. Splunk technology is designed to investigate, monitor, analyze and act on data at any scale. Join millions of passionate users by trying Splunk for free.

Free Trial

splunk>