Engineer

Protokollierung

fortlaufendes Aufzeichnen von Ereignissen beim Eintreten

Was es ist

Protokollierung, auch Ereignisprotokollierung oder Logging genannt, bedeutet, dass ein Programm fortlaufend über sein eigenes Verhalten Buch führt: ein Eintrag je Ereignis, etwa eine beantwortete Anfrage, ein abgewiesenes Passwort, ein erneuter Versuch oder ein Absturz. Jeder Eintrag, Logzeile oder Logdatensatz genannt, enthält mindestens eine Zeitangabe und eine Meldung. So lässt sich später rekonstruieren, was in welcher Reihenfolge geschah. Das OpenTelemetry-Projekt definiert einen Logeintrag als strukturierten oder unstrukturierten Textdatensatz mit Zeitstempel und optionalen Metadaten. Es merkt an, dass Logs unter allen Telemetriesignalen den größten historisch gewachsenen Bestand haben.

Für diese Vorgeschichte gibt es einen Standard. Syslog, das seit Langem verwendete Protokoll zur Weitergabe von Ereignismeldungen an eine Sammelstelle, wurde im März 2009 als RFC 5424 standardisiert. Dieser ersetzt den älteren RFC 3164 und weist jeder Meldung einen von acht Schweregraden zu: von 0 für Emergency bis 7 für Debug. Die meisten Protokollierungsbibliotheken haben eine kürzere Abstufung derselben Idee: debug, info, warning, error. OpenTelemetry bildet sie in seinem Logdatenmodell alle auf eine Zahl von 1 bis 24 ab, in sechs Gruppen von TRACE bis FATAL.

Eine unstrukturierte Zeile ist freier Text: für Menschen leicht lesbar, für eine Maschine nur mit einem regulären Ausdruck zerlegbar, der genau auf diesen Wortlaut zugeschnitten ist. Eine strukturierte Zeile folgt einem festen Aufbau, sodass jeder Eintrag dieselben benannten Felder mit denselben Typen und Bedeutungen enthält. OpenTelemetry bevorzugt strukturierte Logs im Produktionsbetrieb und ergänzt eine leicht übersehene Warnung: Eine als JSON codierte Zeile ist nicht schon deshalb strukturiert, weil sie sich parsen lässt. Entscheidend ist ein stabiler Aufbau. Der hier über mehrere Zeilen gezeigte strukturierte Eintrag steht auf der Platte in einer einzigen Zeile:

```
{
    "time": "2026-09-14T03:12:07.412Z",
    "severity": "WARN",
    "service": "checkout",
    "event": "payment.retry",
    "attempt": 3,
    "trace_id": "4bf92f3577b34da6a3ce929d0e0e4736"
}
```

Metriken und Traces sind die gewöhnlich daneben erfassten Signale. Metriken sind über die Zeit gemessene Zahlen, günstig zu speichern und geeignet, Häufigkeiten auszudrücken. Traces verfolgen eine Anfrage durch jeden von ihr berührten Dienst. Ein Log hält Einzelheiten einzelner Ereignisse fest. Deshalb liest man es nach einem Alarm, um die Ursache zu finden. OpenTelemetry verbindet Logs mit Traces, indem es jedem Logdatensatz die IDs des aktiven Traces und Spans anhängt.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist Protokollierung?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Monitoring und Alarme melden sich, solange ein Fehler noch klein ist; Kennzahlen zeigen, wie oft das gelingt.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. Das erste Observability-Dashboard aufgebaut — der erste Einblick des Unternehmens in die eigenen Systeme

Das erste Observability-Dashboard aufgebaut Kontakt aufnehmen →

Warum es wichtig ist

Ein Log ist die einzige Aufzeichnung der Vergangenheit, die die meisten Systeme hinterlassen. Eine Metrik kann sagen, dass die Fehler um 03:12 zunahmen. Das Log sagt, welche Anfrage aus welchem Dienst mit welchem Wert betroffen war und was das Programm danach tat. Geht nachts etwas kaputt und hat sich bis zum Morgen erholt, bleibt das Log als Zeuge zurück.

Diese Aufzeichnung hat ihren Preis in drei Formen. Jede Zeile kostet Plattenplatz, Ein- und Ausgabe sowie Prozessorzeit, und zusätzlich Ressourcen, wenn sie an einen zentralen Speicher gesendet wird. Ein gehosteter Dienst rechnet dort gewöhnlich nach aufgenommener Datenmenge ab. Jede Zeile ist außerdem eine gespeicherte Information über jemanden: Das Zugriffslog eines Webservers enthält die Adressen und Browser aller Aufrufenden. Und jede Zeile muss geschützt werden. Das OWASP Logging Cheat Sheet verlangt, dass jedes

Ereignis Wann, Wo, Wer und Was festhält und die Logs selbst gegen Manipulation sowie unbefugten Zugriff, Änderung und Löschung geschützt werden, weil sie personenbezogene und andere sensible Informationen enthalten können.

Auf Organisationsebene behandelt NIST Special Publication 800-92, der im September 2006 veröffentlichte Guide to Computer Security Log Management von Karen Kent und Murugiah Souppaya, die Verwaltung von Logs als Infrastruktur und Prozesse im gesamten Unternehmen statt als Einstellung eines einzelnen Programms.

Logs sind außerdem am nützlichsten, wenn sie durch Zahlen ergänzt werden. Der Instrumentierungsleitfaden des Prometheus-Projekts empfiehlt einen Zähler für jede Zeile protokollierenden Codes. Taucht eine interessante Meldung auf, lässt sich dann feststellen, wie oft und über welchen Zeitraum sie aufgetreten ist. Er empfiehlt außerdem, die Gesamtzahlen der info-, warning- und error-Zeilen zwischen Veröffentlichungen zu vergleichen.

Wie es in der Praxis aussieht

Das eigene Zugriffslog der Website wird auf der Datenschutzseite Zeile für Zeile beschrieben. Jede Anfrage zeichnet Seite und Zeitpunkt, Antwortstatus, Selbstbeschreibung des Browsers, Namen der verweisenden Website und eine vergröberte Netzwerkadresse auf: Ihr letzter Teil wird beim Schreiben der Zeile durch Nullen ersetzt, bevor etwas die Platte erreicht. Eine Zeile bleibt etwa dreißig Tage erhalten. Die Serversoftware schließt jede Datei einmal täglich oder früher bei etwa 10 MB und löscht eine geschlossene Datei achtundzwanzig Tage später. Jede Nacht liest eine Prüfung sämtliche Dateien erneut und schlägt bei jeder über dreißig Tage alten Zeile fehl. Zweimal ging das schief; die Seite dokumentiert beides. Im September 2026 füllten sich die Dateien der wenig genutzten Adressen so langsam bis zur Grenze von 10 MB, dass sie nie geschlossen worden waren und noch jede Zeile seit Beginn der Protokollierung enthielten. Deshalb kam ein täglicher Auftrag hinzu, der sie schloss. Am 26. Oktober 2026 schrieben sieben der dreizehn Adressen mit Log weiterhin in Dateien, die dieser Auftrag als geschlossen markiert hatte. Für eine Adresse blieb eine siebenundvierzig Tage alte geschlossene Datei erhalten. Der Auftrag ist abgeschafft; die Serversoftware schließt und löscht die Dateien jetzt selbst.

Ein vollständiger Testlauf einer macOS-Anwendung erfasste 610.996 Logzeilen des Anwendungskerns. Ein Schub von Logereignissen des Anwendungskerns wurde in einem unbegrenzten Hauptstrom gehalten und in Abonnentenströme kopiert, deren Verbraucher sie nicht nutzten. Der Vorfallbericht dokumentiert getrennt davon Diagnoseberichte über 111 GB komprimierter Speicherseiten auf einem Rechner mit 36 GB und eine Stichprobe des Speicherbedarfs, die während einer Schleife erneuter Verbindungsversuche ein Wachstum von etwa 41 MB pro Sekunde zeigte. Begrenzte Puffer, nach Ereignistyp gefilterte Abonnements und ein Budget für die Protokollierungsrate setzten an diesem Ereignispfad an. Der dokumentierte Vergleich vollständiger Testläufe senkte die Logzeilen des Anwendungskerns auf 1.411. Das ist weder ein Produktionsbenchmark noch eine gemessene Speichergröße nach der Korrektur. Das Budget verwirft absichtlich Informationen: Wiederholte Diagnoseereignisse dürfen entfallen, und betroffener Zustand kann stattdessen neu eingelesen werden, anstatt unbegrenzt Ereignisse aufzubewahren. Separate Zeitüberschreitungen beim Beenden brauchten eine eigene Korrektur.

Auf einem Produktionshost fand eine Härtung von dreizehn systemd-Units eine seit Bestehen der betreffenden Rollen in drei Unit-Templates falsch geschriebene Direktive. Der Dienstmanager reagiert auf einen unbekannten Schlüssel, indem er ihn protokolliert und die Unit trotzdem startet. Die einzige Spur des Fehlers war deshalb eine Warnung im Log; entdeckt wurde er beim Umschreiben der Units. Jetzt durchläuft jede Unit bei jedem Konvergenzlauf den eigenen Prüfer des Dienstmanagers, sodass derselbe Tippfehler den Lauf scheitern lässt.

Wo es schiefgeht

Das Log wächst am schnellsten, wenn das System schon ausfällt. Eine Schleife, die jeden erneuten Versuch protokolliert, schreibt an einem guten Tag eine Handvoll Zeilen und an einem schlechten Tausende pro Sekunde. Sie entzieht einem System ohne Reserven Plattenplatz, Speicher und Prozessorzeit. OWASP verlangt die Prüfung, dass Protokollierung nicht zum Erschöpfen von Systemressourcen missbraucht werden kann, etwa durch eine volle Platte. Geben Sie jedem häufig durchlaufenen Pfad ein Budget, fassen Sie wiederholte Meldungen in einer Zeile mit Anzahl zusammen und führen Sie neben der Meldung einen Zähler, damit die Rate erhalten bleibt, wenn Zeilen entfallen.

Geheimnisse und personenbezogene Daten werden aufgeschrieben. Eine vollständig protokollierte Anfrage enthält alles, was darin stand. OWASPs Liste der vor dem Log gewöhnlich zu entfernenden, zu maskierenden, zu hashenden

oder zu verschlüsselnden Daten umfasst Sitzungskennungen, Zugriffstokens, Passwörter, Datenbank-Verbindungszeichenfolgen, Verschlüsselungsschlüssel und Zahlungskartendaten. Tun Sie das beim Schreiben der Zeile, wie das Zugriffslog oben mit Adressen. Bis jemand aufräumt, kann die Zeile bereits in einer Sicherung oder einem Archiv liegen.

Eingaben von außen schreiben eine gefälschte Zeile. Ein vom Nutzer gelieferter Wert mit Wagenrücklauf oder Zeilenumbruch kann, unmaskiert in ein Textlog geschrieben, eine neue Zeile beginnen, die genau wie eine Zeile der Anwendung aussieht. OWASP nennt das Log Injection und verlangt, solche Zeichen zu bereinigen und Daten für das verwendete Ausgabeformat zu codieren. Ein strukturierter Logger, der jeden Wert als maskierte JSON-Zeichenfolge schreibt, tut dies durch seine Konstruktion.

Ein verstummtes Log gilt als gesundes System. Eine volle Platte, eine rotierte und vom Programm nie erneut geöffnete Datei oder eine ausgefallene Sammelstelle sehen von außen gleich aus: Die Zeilen bleiben aus. OWASP verlangt eine Möglichkeit zu erkennen, dass die Protokollierung aufgehört hat. Zählen Sie für jede Quelle die Zeilen pro Minute und alarmieren Sie, wenn der Zähler auf null fällt. Eine Lücke im Log wird so zu einem eigenen Alarm.

Weiterführende Lektüre

- Logs — OpenTelemetry darüber, was ein Log ist, warum strukturierte Logs bevorzugt werden und wie sie mit Traces verbunden sind.
- Logdatenmodell — die Felder eines Logdatensatzes, seine beiden Zeitstempel und die Schweregradskala von 1 bis 24.
- RFC 5424: Das Syslog-Protokoll — das Syslog-Meldungsformat und seine acht Schweregrade.
- NIST SP 800-92 — der Guide to Computer Security Log Management.
- Logging Cheat Sheet — OWASP dazu, welche Ereignisse zu erfassen, welche Daten fernzuhalten und wie die Ergebnisse zu schützen sind.
- Instrumentierung — die Verfahren des Prometheus-Projekts zur Instrumentierung von Code einschließlich Protokollierung.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Bereitstellungsweg, die Betriebsrisiken und die Bereitschaftsverantwortlichen. Wir können eine Änderung mit Prüfungen, einem Wiederherstellungsweg und klaren Zuständigkeiten abgrenzen.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist Protokollierung?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/logging/