Engineer

Журналювання

запис подій у момент, коли вони стаються

Що це таке

Журналювання, або логування, — це безперервний облік, який програма веде про власну поведінку, по одному запису на кожну подію: обслужений запит, відхилений пароль, повторна спроба, аварійне завершення. Кожен такий запис, рядок журналу, містить щонайменше час і повідомлення, щоб згодом можна було відновити, що сталося і в якому порядку. Проєкт OpenTelemetry визначає журнал як текстовий запис із часовою міткою, структурований чи неструктурований, з необов'язковими метаданими, і зазначає, що з усіх сигналів телеметрії журнали мають найбільшу спадщину.

Ця спадщина має стандарт. Syslog, давній протокол для передавання повідомлень про події до збирача, у березні 2009 року стандартизовано як RFC 5424, що замінив старіший RFC 3164; він присвоює кожному повідомленню один із восьми рівнів серйозності, від 0 для Emergency до 7 для Debug. Більшість бібліотек журналювання мають коротшу драбину з тією самою ідеєю: debug, info, warning, error. Модель даних журналів OpenTelemetry відображає їх усі на одне число від 1 до 24, у шести смугах від TRACE до FATAL.

Неструктурований рядок — це вільний текст: людині його легко прочитати, а машина може розібрати його лише регулярним виразом, написаним саме під це формулювання. Структурований рядок дотримується схеми, тож кожен запис має ті самі іменовані поля з тими самими типами й призначенням. OpenTelemetry віддає перевагу структурованим журналам у продакшні й додає застереження, яке легко пропустити: рядок, закодований як JSON, не стає структурованим лише тому, що він розбирається; структурованим його робить стабільна схема. Тут запис показано в кількох рядках, а на диску це один рядок:

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

Метрики й трасування — сигнали, які зазвичай збирають поруч із ним. Метрики — це числа, виміряні з плином часу, дешеві для зберігання й добрі для відповіді на питання «як часто». Трасування супроводжує один запит через кожен сервіс, якого він торкається. Журнал зберігає подробиці окремих подій, тож коли спрацювала тривога, саме до нього звертаються, щоб дізнатися чому. OpenTelemetry пов’язує журнали з трасуваннями, додаючи до кожного запису ідентифікатори активного трасування й спану.

Застосуйте поняття до рішення

Застосуйте «Що таке журналювання?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Моніторинг і сповіщення, які попереджають, поки збій ще малий, і цифри, що показують, як часто так і залишається.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Запустила перший дашборд спостережуваності — перший огляд власних систем компанії

Запустила перший дашборд спостережуваності Зв'язатися з нами →

Чому це важливо

Журнал — це єдина розповідь про минуле, яку залишає по собі більшість систем. Метрика може показати, що о 03:12 кількість помилок зросла; журнал скаже, який саме запит, від якого сервісу, з яким значенням і що програма зробила далі. Коли щось ламається вночі й до ранку відновлюється, журнал — той свідок, що досі на місці.

Ця розповідь має ціну, і сплачують її трьома способами. Кожен рядок коштує місця на диску, операцій введення-виведення й процесорного часу, а ще більше — якщо його надсилають до центрального сховища, де хостингова служба зазвичай виставляє рахунок за обсягом отриманих даних. Кожен рядок — це також дані про когось: журнал доступу вебсервера містить адреси й браузери всіх, хто до нього звертався. І кожен рядок треба захищати. OWASP Logging Cheat Sheet просить, щоб кожна подія фіксувала,

коли, де, хто і що, а самі журнали були захищені від підробки та від несанкціонованого доступу, змін і видалення, бо вони можуть містити персональні та інші чутливі дані.

У масштабі організації NIST Special Publication 800-92, Guide to Computer Security Log Management, яку написали Karen Kent і Murugiah Souppaya і видали у вересні 2006 року, розглядає керування журналами як інфраструктуру й процеси в межах усього підприємства, а не як налаштування однієї програми.

Журнали також працюють найкраще поруч із числами. Посібник з інструментування проєкту Prometheus радить заводити лічильник для кожного рядка коду журналювання, щоб, коли трапиться цікаве повідомлення, можна було побачити, як часто воно з'являлося і як довго, а також порівнювати кількість рядків info, warning і error між релізами.

Як це виглядає на практиці

Власний журнал доступу сайту описано рядок за рядком на сторінці про конфіденційність. Для кожного запиту записуються сторінка й час, статус відповіді, самоопис браузера, назва сайту, з якого прийшов відвідувач, і розмита мережева адреса: останню частину замінюють нулями в момент запису рядка, ще до того, як щось потрапить на диск. Рядок живе приблизно тридцять днів. Серверна програма закриває кожен файл раз на добу, а коли той доростає приблизно до 10 МБ — то й раніше, і видаляє закритий файл через двадцять вісім днів, а щоночі окрема перевірка перечитує всі файли й завершується помилкою, якщо знаходить рядок, старший за тридцять днів. Двічі тут траплялися збої, і сторінка про конфіденційність розповідає про обидва. У вересні 2026 року малоактивні адреси набирали 10 МБ так повільно, що їхні файли жодного разу не закривалися й досі містили кожен рядок відтоді, як увімкнули журналювання, тож додали щоденне завдання, яке мало їх закривати. 26 жовтня 2026 року сім із тринадцяти адрес, що ведуть журнал, і далі писали у файли, які завдання позначило як закриті, а одна з них зберігала закритий файл, якому було сорок сім днів. Завдання прибрано, і тепер файли закриває й видаляє сама серверна програма.

Повний тестовий запуск застосунку macOS зафіксував 610 996 рядків журналу ядра програми. Сплеск подій цього журналу зберігався в необмеженому основному потоці й копіювався в потоки підписників, чиї споживачі цих подій не використовували. Запис про інцидент окремо

документує діагностичні звіти про 111 ГБ стиснутих сторінок на машині з 36 ГБ і замір обсягу зайнятої пам’яті зі зростанням приблизно на 41 МБ за секунду під час циклу повторних спроб підключення. Обмежені буфери, підписки з фільтрацією за типом події та бюджет частоти записів до журналу виправили цей шлях обробки подій. У зафіксованому порівнянні повних тестових запусків кількість рядків журналу ядра програми зменшилася до 1 411. Це не бенчмарк у виробничому середовищі й не виміряний показник використання пам’яті після виправлення. Бюджет навмисно відкидає інформацію: повторювані діагностичні події можна пропускати й оновлювати відповідний стан, замість того щоб зберігати необмежену кількість подій. Окремі тайм-аути під час завершення роботи потребували власного виправлення.

На робочому хості посилення захисту тринадцяти юнітів systemd виявило директиву, яку з помилкою писали у трьох шаблонах юнітів увесь час, скільки існували ці ролі. Менеджер служб відповідає на ключ, якого не розпізнає, записом у журнал і все одно запускає юніт, тож єдиним слідом помилки було попередження в журналі, а виявили її під час переписування юнітів. Тепер кожен юніт під час кожного застосування конфігурації проходить через власний верифікатор менеджера служб, тож така сама помилка зупиняє прогін.

Де виникають проблеми

Журнал росте найшвидше, коли система вже дає збій. Цикл, що записує кожну повторну спробу, у добрий день дає жменю рядків, а в поганий — тисячі за секунду, забираючи диск, пам’ять і процесорний час у системи, якій їх і так бракує. OWASP просить перевіряти, що журналювання не можна використати для виснаження ресурсів системи, наприклад заповнивши диск. Дайте кожному гарячому шляху бюджет, згортайте повторювані повідомлення в один рядок із кількістю повторів і тримайте лічильник поруч із повідомленням, щоб частота збереглася, коли рядки відкидаються.

Секрети й персональні дані потрапляють у запис. Запит, записаний цілком, несе все, що в ньому було. Список OWASP того, що зазвичай слід видалити, замаскувати, гешувати чи зашифрувати, перш ніж воно потрапить до журналу, містить ідентифікатори сесій, токени доступу, паролі, рядки підключення до бази даних, ключі шифрування й дані платіжних карток. Робіть це в момент запису рядка, як журнал доступу вище робить з адресами, бо поки хтось візьметься за прибирання, рядок уже може лежати в резервній копії чи архіві.

Зовнішнє введення дописує фальшивий рядок. Значення від користувача з символом повернення каретки чи переведення рядка, записане без екранування в текстовий журнал, може почати новий рядок, точнісінько схожий на той, що написав сам застосунок. OWASP називає це log injection і просить очищати такі символи та кодувати дані під формат, у якому їх записують. Структурований логер, що записує кожне значення як екранований рядок JSON, робить це самою своєю будовою.

Журнал, що замовк, виглядає як здорова система. Повний диск, ротований файл, який програма так і не відкрила знову, чи збирач, що впав, ззовні виглядають однаково: рядки перестають надходити. OWASP просить мати спосіб помітити, що журналювання зупинилося. Рахуйте рядки за хвилину для кожного джерела й ставте тривогу на падіння цього числа до нуля, щоб пропуск у журналі сам ставав тривогою.

Що почитати далі

- Logs — OpenTelemetry про те, що таке журнал, чому перевагу віддають структурованим журналам і як журнали пов’язують із трасуваннями.
- Logs Data Model — поля запису журналу, дві його часові мітки й шкала серйозності від 1 до 24.
- RFC 5424: The Syslog Protocol — формат повідомлень syslog і вісім рівнів серйозності.
- NIST SP 800-92 — Guide to Computer Security Log Management.
- Logging Cheat Sheet — OWASP про те, які події записувати, які дані тримати поза журналом і як захистити результат.
- Instrumentation — практики проєкту Prometheus з інструментування коду, зокрема журналювання.

Словник

Обирайте за фактичними умовами

Опишіть шлях розгортання, експлуатаційні ризики та чергову команду. Ми можемо визначити зміну з перевірками, шляхом відновлення та чіткою відповідальністю.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке журналювання?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

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

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/logging/