Технологическое описание системы   
логгирования, обнаружения и мониторинга

Система логгирования, обнаружения и мониторинга (далее “система событий”) - это набор решений

**1) позволяющий:**

* собирать значения в режимах push/pull и формировать из значений события
* формировать события из инфопотоков, проходящий через приложения
* опираться на единый формат сообщения, описывающего структуру события
* использовать единую шину обмена событиями
* отправлять события в инфраструктуру
* хранить и аггрегировать события
* описывать метрики значений в событиях
* выполнять действия (в т.ч. оповещения) по заданным условиям (триггеры)
* строить графики, комплекты графиков, сложные экраны работы с событиями
* выполненять простейшие математические операции со значениями в событиях
* получать доступ к данным посредством HTTP-интерфейса
* использовать шифрования для гарантии безопасности передачи
* создавать гибкие политики хранения по классам событий (перемещение/ удаление/ сжатие)

**2) состоящий** из трёх базовых классов элементов:

* Поставщиков событий
* Участников шины передачи событий
* Узлов обработки и хранения событий

**3) решающий** следующие типы задач :

* сбор
* приведение к единому формату
* доставка
* фильтрация / вычисления
* хранение
* визуализация
* аналитика
* предсказания

I. Поставщики событий

Поставщиком событий является любое приложение, реализующее вывод событий в форматах:

1. CEE lumberjack
2. JSON
3. syslog
4. structured string

с использованием протоколов:

1. syslog (kernel syscall)
2. syslog (UDP/TCP)
3. file
4. unix socket
5. unix kernel log
6. systemd journal

Поставщики событий по способу взаимодействаия делятся на:

1. Пассивные (pull)

* Синтетические проверки

1. Активные (push)

* Потоки
* Комманды
* Discovery-сообщения

II. Участники шины передачи событий

Общими возможностями для всех участников шины передачи событий является:

* использование сохраняемых на диск очередей
* отчётность о состоянии очередей не реже, чем раз в 5 минут
* использование протокола шины, гарантирующего отсутствие потерь при передаче и возникающих сбоях (RELP)
* использование TLS-шифрования для гарантии безопасности передачи
* использование аутентификации по сертификатам для гарантии наличия прав доступа к шине
* установка доверительных меток времени - времени получения сообщения участником шины

**Типы участников шины передачи событий:**

1. Обработчик событий на стороне клиента
2. Шлюз сбора сырых событий
3. Узлы обработки и хранения событий (только принимают сообщения)

1. Обработчик событий на стороне клиента

* Является первым звеном асинхронной шины передачи событий
* Отвечает за протокол взаимодействия с Поставщиком событий
* Преобразует формат событий Поставщика в единое стандартное событие формата CEE.
* Использует для преобразования как JSON-токены (jsonparse), так и мета-формат разбиения строк на токены (normalize)
* Устанавливает доверительные метки времени - Время события и Время получения события
* Генерирует уникальный ID события (UUID)

2. Шлюз сбора сырых событий

* Является вторым звеном асинхронной шины передачи событий
* Является точкой аггрегации всех событий от клиентских обработчиков
* Проверяет корректность заполнения поля регистрации времени события
* Устанавливает доверительные метки времени - Время получения события шлюзом
* Генерирует уникальный ID прохождения события через шлюз

III. Узлы обработки, хранения и доступа к событиям

Реализацией узлов обработки, хранения и доступа к событиям могут быть :

* серверы хранения событий в виде файлов/баз данных (Elasticsearch,PGSQL)
* приложения визуализации (Kibana/Grafana)
* системы мониторинга (Zabbix)

Узлы обработки, хранения и доступа к событиям могут реализовывать следующую функциональность:

1. Получение событий согласно спецификации шины
2. Хранение событий в виде структуры директорий/файлов
3. Хранение событий в базе Elasticsearch
4. Дополнительная обработка событий средствами rsyslog/logstash
5. Отправка событий в периметр безопасного хранения
6. Отправка событий в системы регистрации сервисов
7. Протокольные шлюзы конвертации cee-сообщений в другие форматы/протоколы (Zabbix TCP-based JSON)
8. Доступ к событиям с использованием web-приложений

4. Дополнительная обработка событий средствами rsyslog/logstash

* Получает поток событий от Шлюза сбора сырых событий
* Выполняет фильтрацию/преобразование/совмещение событий
* Устанавливает доверительные метки времени - Время получения события шлюзом