DOI: https://doi.org/10.36910/6775-2524-0560-2023-51-08
УДК 004.94:656.02
Мельник Василь Михайлович, к. фіз.-мат.н., доцент
https://orcid.org/0000-0001-8282-6639
Волинський фаховий коледж НУХТ
Багнюк Наталія Володимирівна, к.т.н., доцент
https://orcid.org/0000-0002-7120-5455
Бортник Катерина Яківна, к.т.н., доцент
https://orcid.org/ 0000-0001-5282-099X
Лінчук Олександр Миколайович, асистент
Луцький національний технічний університет, м. Луцьк, Україна

СИСТЕМА ДЛЯ ВЕДЕННЯ МОНІТОРИНГУ WEB-РЕСУРСУ ЗАСОБАМИ ELASTICSEARCH

Мельник В.М., Багнюк Н.В., Бортник К.Я, Лінчук О.М. Система для ведення моніторингу web-ресурсу засобами Elasticsearch. У роботі скомпоновано інтегровану систему для виконання моніторингу web-ресурсу засобом Elasticsearch. З'ясовано основні вимоги, наведено інтерфейс та необхідні ресурси для створення інтегрованої системи моніторингу, включаючи виконання інсталяції, налаштування та перевірки роботи. В систему включені основні програмні засоби, такі як Elasticsearch, Kibana, Logstash та Beats, які працюють з підтримки операційної системі Linux Ubuntu. В цілому, створена система досить добре відслідковує інциденти та виконує необхідний моніторнг.
Ключові слова: Elastic Search, веб-ресурс, безпека функціонування, вразливість, моніторинг.

Melnyk V., Bahniuk N., Bortnyk K., Linchuk O. System for web-resource monitoring by means of Elasticsearch. The paper composes an integrated system for web-resource monitoring studies made by Elastic Search. There are clarified basic requirements, an interface and the necessary resources for the integrated monitoring system creation, including installation, configuration and verification in work, are provided. The system includes basic software tools such as ElasticSearch, Kibana, Logstash and Beats, which are supported by Linux Ubuntu operating system. In general, the established system monitors incidents quite well and performs the necessary monitoring.
Keywords: Elastic Search, web resource, operational security, vulnerability, monitoring.

Вступ

Зі швидким розвитком інформаційних технологій, рівнем життя та комфортності, комп'ютерні системи та мережі стали вагомо впливати майже на всі сфери людської діяльності. Інформація перетворилася в товар, яким цілком можливо обмінюватися, придбати чи продати. В деяких випадках вартість інформації нерідко в багато разів перевершує вартість самих комп'ютерних систем, які її зберігають чи передають.

Однак, сучасні методи обробки, зберігання та передачі інформації досить вразливі до загроз, пов'язаних з її втратою, спотворенням чи розсекреченням. Тому недостатня захищеність та широке поширення таких систем в експлуатацію обумовлюють інтерес потенційних правопорушників, прояві їх нападів та зацікавленості пошуку їх вразливих сторін з метою заволодіння інформацією та її подальшого використання в корисних власних цілях. Час реагування на інцидент є надзвичайно важливим показником. Так як після будь-якої атаки власник веб-ресурсу несе різноманітні втрати, то чим довше вирішується інцидент, тим більше збитків може бути завдано. Внаслідок вище сказаного актуальним завданням є активізація вчасного реагування на інцидент та скорочення протікання його часу, яке починається з застосування відповідних підходів та засобів моніторингу з метою запобігання зловмисних дій та забезпечення безпечної роботи веб-ресурсу в мережевому оточенні.

Огляд SIEM-системи

До складу SIEM-системи входять SIM (Security Information Management) [1,11], який забезпечує аналіз, зберігання та звітність за даними накопичення та SEM (Security Event Management), що забезпечує керування безпекою, кореляцією подій, моніторингом в реальному часі, сповіщеннями та відображенням на кінцевих пристроях. SIEM-технологія забезпечує аналіз подій безпеки, отриманих від додатків і мережевих пристроїв. SIEM репрезентується додатками, послугами чи приладами і застосовується для журналювання даних і ведення звітів у сумісності з бізнес-даними. Дана система здатна відстежувати аномальну поведінку IT-систем і користувачів та своєчасно попереджувати про це співробітників за профілем.

Використання SIEM-системи може мати місце в таких випадках:

- для виявлення вразливості нульового дня та поліморфних вірусів, так як антивірусні додатки не володіють достатнім рівнем показників для виявлення цього типу шкідливих програм;

- для виконання втоматичного синтаксичного аналізу, нормалізації та класифікації журналів, незалежно від типу комп'ютера чи мережевого пристрою, які можуть журналювати події через пристрої;
- для візуалізації з використанням подій безпеки та журналу збоїв з метою допомоги у виявленні шаблонів;
- для формування протоколу відхилень зі вказівкою на неправильну конфігурацію або проблему безпеки, яку може бути виявлено під час розпізнання шаблонів, використання сповіщень та інформаційних панелей;
- SIEM має змогу виявляти шкідливі секретні повідомлення та викривати зашифровані канали;
- за допомогою SIEM з відповідною точністю може бути виявлена кібератака з можливістю визначити нападника та жертву.

В [11] наведено типи сповіщень на відповідні події, на які може бути налаштована діюча SIEM-система.

Аналіз програмних пакетів для організації системи

Одним з найкращих варіантів для реалізації SIEM-системи може використовується ELK-стек, який може поєднувати в собі три OpenSource-проекти: Elasticsearch, Logstash і Kibana [2-5]. ELK-стек є досить доступним, поширюється безкоштовно, хоча існують і платні функції, не обов'язкові для налаштування основної моніторингової системи.

Elasticsearch – це додаток з відкритим програмним кодом мови програмування Java, призначений для реалізації повнотекстового пошукового механізму та аналізу і базується на пошуковій системі Apache Lucene [2]. Він є головним елементом ELK Stack. Його особливістю є централізоване зберігання данних, необхідних для полегшення пошуку та перевірки різних аномалій чи невідповідностей. В Elasticsearch реалізовано механізм виконання та поєднання різних видів пошуку, наприклад, пошук за IP-адресою та отриманим повідомленням [3]. До недоліків його реалізації можна віднести високі апаратні вимоги, як правило, до оперативної пам'яті.

Logstash – це механізм обробки журнальних даних на стороні сервера, що виконує приймання їх з ріних джерел [4] і реалізований засобами мови програмування Java. Він за певним шаблоном чи шаблонами виконує синтаксичний аналіз вхідних данних і розбиває їх на багато полів у відповідності до попередніх налаштувань та надсилає оброблені вже данні до Elasticsearch. Вхідні дані приходять зазвичай з різних пристроїв та систем, тобто, у різних форматах, а Logstash приводить їх до єдиного вигляду з метою полегшення роботи з ними в подальшому. Організація отримання данних з різніх джерел для Logstash [10] реалізується через багаточисленні плагіни, які дозволяють фільтрувати та форматувати їх. У нього більш високі вимоги до оперативної пам'яті, ніж у Elasticsearch, а тому його розгортання рекомендується на окремій фізичній машині.

Kibana – це прикладна графічна платформа для побудови графіків та діаграм, реалізована на мові JavaScript і призначена для пошуку та перегляду даних, що проіндексовані та збережені в базі Elasticsearch [3,5]. Вона володіє простим та інтуїтивно-зрозумілим інтерфейсом для роботи в браузері.

Beats – це додатки, призначенні для збирання даних і відправки їх в Elasticsearch або Logstash [6,10], не вибагливі до ресурсів і не впливають на роботу пристрою встановлення. Однак, деякі некоректно налаштовані можуть накладати значні навантаження на систему, що може призвести до невиконання нею своїх функцій [8].

Filebeat – призначений для зчитування логів (рис. 1) [12]. Він стартує один чи кілька input-входів для перегляду місць, вказаних для журналу даних. Для кожного з журналів, який Filebeat знаходить, запускає файл, який виконує зчитування по одному з них для нового вмісту та надсилання нових його даних до Libbeat. Libbeat агрегує події та пересилає агреговані дані на вихід, налаштований для Filebeat, і запам'ятовує рядок зупинки. Це дозволяє відновити його роботу на випадок появи помилки і запобігає відправленню повторних повідомлень.

Пакет Metricbeat призначений для періодичного збору необхідних показників з операційної системи та серверних служб [13]. Він збирає метрики та статистичні дані і доставляє їх до вихідних йому вказаних даних, таких як Logstash або Elasticsearch [10]. Також він допомагає в здійсненні контролю серверів, збираючи показники із систем та серверних служб, таких як: HAProxy, Apache, MySQL, PostgreSQL, Nginx, MongoDB, Redis, System та Zookeeper.

Програма Auditbeat призначена для аудиту діяльності процесів та користувачів у робочих системах [16]. Її можна використовувати для збору та централізації подій аудиту з Linux Audit Framework чи Auditbeat для виявлення змін у таких важливих файлах, як двійкові та конфігураційні

файли, а також для виявлення потенційних порушень політики безпеки. Дана програма досить сильно навантажує систему під час відслідковування великої кількості файлів.

Packetbeat відіграє роль аналізатора мережевих пакетів в реальному часі, використовується в поєднанні з Elasticsearch для забезпечення системи моніторингу програм та аналізу їх продуктивності [15]. Він завершує Beats-платформу, забезпечуючи міжсерверну видимість в мережі. Даний пакет працює на базі захоплення мережевого трафіку між серверами, виконуючи декодування протоколів HTTP, MySQL, Redis та інших, розташованих на рівні додатків, співвідносячи запити з відповідями та фіксуючи цікаві поля для кожної з транзакцій. Він може допомогти легко виявити проблеми з внутрішньою програмою, такі як помилки чи проблеми в продуктивності, та виконує їх усунення набагато швидше.

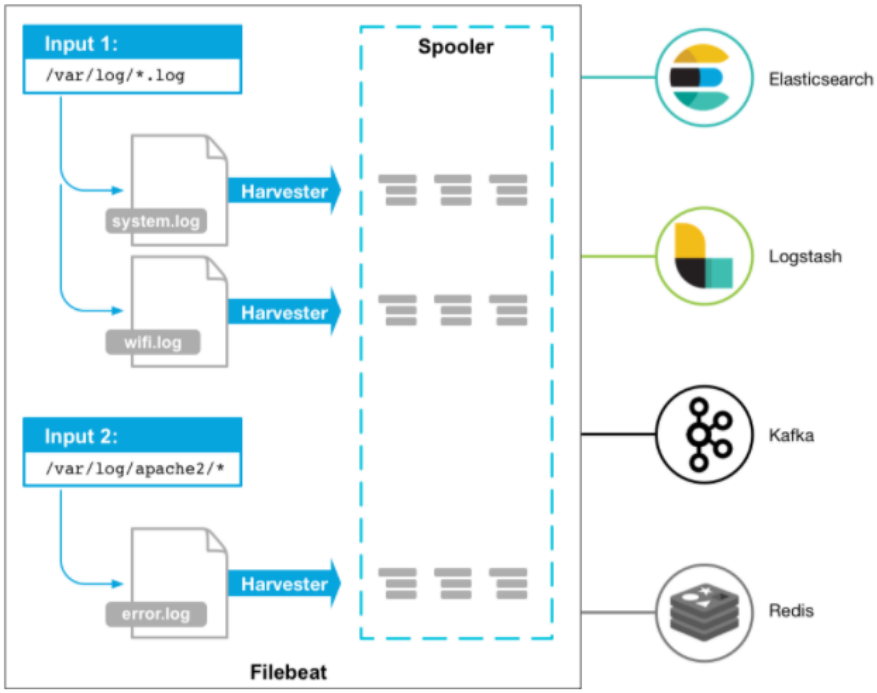

Рис.1 – Схема роботи Filebeat

Програма Packetbeat перевіряє міжсерверний трафік, аналізує протоколи на рівні прикладних програм і співвідносить повідомлення з транзакціями. Дана програма підтримує наступні мережеві протоколи: ICMP (v4 and v6), DHCP (v4), DNS, HTTP, AMQP 0.9.1, Cassandra, MySQL, PostgreSQL, Redis, Thrift-RPC, MongoDB, Memcache, NFS, TLS та SIP/SDP (beta). Packetbeat може надсилати відповідні транзакції безпосередньо в Logstash або Elasticsearch та працювати на тих же серверах, що й процеси додатків, чи на власних серверах [8,10]. В ході роботи на визначених серверах Packetbeat може отримувати трафік від прослуховуючих дзеркальних портів комутатора або від пристроїв. В процесі такого розгортання на моніторинговій програмі відсутня затримка.

Пакет Winlogbeat призначений для зчитування Windows-логів з одного чи кількох журналів подій за допомогою API, веде фільтрування події на основі критеріїв, налаштованих користувачем, і надсилає дані про події на Elasticsearch або Logstash через налаштовані виходи [10,14]. Winlogbeat веде спостереження за журналами подій з метою своєчасного надсилання нових даних про події. Позиція зчитування для кожного журналу подій зберігається на диску, щоб Winlogbeat міг легко відновити роботу після рестарту. Даний пакет може збирати дані про події з будь-яких журналів подій працюючої системи, наприклад, фіксувати такі як: події додатків, апаратні події, події безпеки та системні події.

Програма Heartbeat також входить в систему могіторингу і встановлюється на віддаленому сервері з метою періодичної перевірки статусу сервісів і визначення їх доступності [17]. На відміну від Metricbeat, який повідомляє про ефективність роботи серверів, Heartbeat визначає і повідомляє про доступність сервісів. Дана програма корисна для підтвердження виконання сервером свого необхідного рівня обслуговування на випадок безвідмовної роботи, що є також корисним для інших сценаріїв, таких як випадки використання безпеки, де необхідно переконатися в неможливості отримання доступу до послуг ззовні на приватному корпоративному сервері. Можна також

налаштувати Heartbeat на перевірку всіх DNS IP-адреси для визначеного імені хоста. Таким чином, можна перевірити всі служби, збалансовані за навантаженням, на їх доступність. В ході виконання налаштування Heartbeat вказуються монітори, що визначають імена хостів для здійснення перевірити, кожен з яких працює на базі йому вказаного розкладу. Наприклад, можна налаштувати один монітор на кожні 10 хвилин запуску, а інший – для роботи з 10:00 до 17:30. Дана програма підтримує монітори для перевірки хостів за допомогою:

- ICMP (v4 та v6) Echo Requests – для перевірки на доступність послуги використовують монітор icmp, який вимагає кореневого доступу.
- TCP – для підключення через TCP використовується tcp-монітор, який можливо додатково налаштувати для контролю кінцевої точки, надіславши та/або отримавши корисне спеціальне навантаження.
- HTTP – використовується для підключення монітор http, який можливо додатково налаштувати для перевірки повернення службою очікуваної відповіді, наприклад, коду стану, заголовка відповіді чи вмісту.

Монітори tcp і http надають підтримку для SSL/TLS та деяких налаштувань проксі.

Вимоги до SIEM-системи

SIEM-система, в основному, повинна володіти такою функціональністю:

- Функціями агрегації даних, управлінням журналами даних, збором даних з різних джерел: мереж, пристроїв та сервісів, датчиків систем безпеки, серверів, баз даних, програм із забезпеченням консолідації даних для пошуку критичних подій.
- Функції кореляції, які реалізують пошук спільних атрибутів чи зв'язування подій у кластери. Така технологія включає застосування технічних заходів для інтеграції даних з різних джерел з метою формування вихідних даних як значущої інформації. Кореляція даних представляє типові функції з підмножини Security Event Management [11].
- Функції сповіщення, які включають автоматизований аналіз подій кореляції та генерацію повідомлень про виявлені проблеми. Оповіщення можуть виводитися на панель додатка, чи направлятися в інші канали оповіщення: GSM-шлюз, e-mail чи інші.
- Функції відображення або інформаційні панелі, реалізуючі відображення діаграм для ідентифікації патернів, відмінних від стандартно прийнятої поведінки.
- Функції сумісності або трансформування, які реалізуються через спеціальні додатки, призначені для автоматичного збору даних, адаптації агрегованих даних до діючих процесів управління інформаційною безпекою, формування звітності та ведення аудиту.
- Функції збереження даних, до яких входять: довготривале зберігання даних в хронологічному часовому порядку з метою їх кореляції та забезпечення трансформування. Довготривале зберігання даних є необхідним для проведення комп'ютерно-технічних експертиз, так як розслідування мережевого інциденту все-таки відбувається з затримкою в часі з моменту його здійснення.
- Функції експертного аналізу, які забезпечують можливість пошуку по журналах, розміщених на різних вузлах і можуть приводитися до виконання в рамках програмно-технічної експертизи.

Функціонально-структурна схема роботи системи

Для організації роботи ELK-стеку встановлюються параметри Beats на цільових машинах, які направляють данні в Logstash [6] (рис. 2). Останній, в свою чергу, обробляє дані і передає їх в Elasticasearch. Logstash повинен бути встановлений на окремій машині, так як він і Elasticsearch потребують відповідних системних ресурсів.

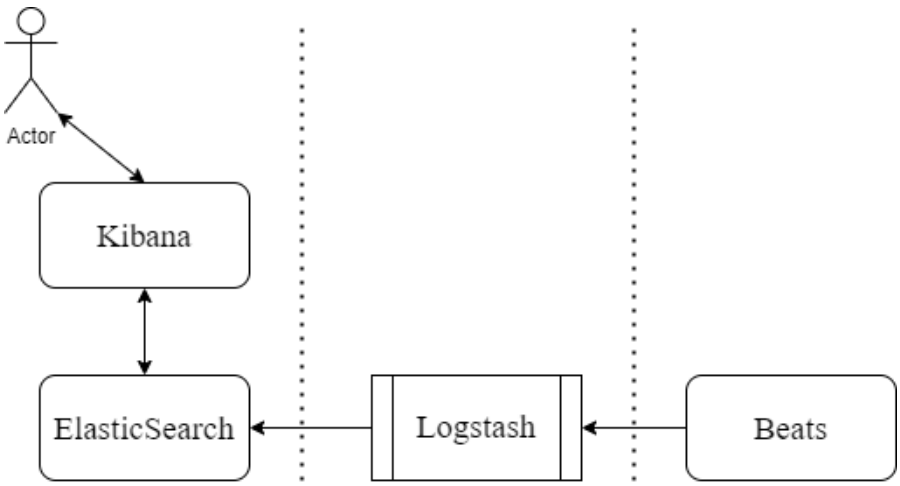

Рис. 2 – Структурна схема ELK-стеку

Elasticsearch отримує і зберігає дані та виконує їх пошук. Для отримання доступу до бази даних користувач, використовуючи Kibana, надсилає запити ElasticSearch та візуалізує отриманні дані [5]. Також в Kibana знаходяться інструменти керування різними аспектами ELK-стеку.

Мета і завдання роботи

З метою проведення дослідження моніторингу необхідно створити та скомпонувати описану базову систему, включивши в неї необхідні ресурси та засоби, згадані в попередніх розділах, з їх функціями та налаштуваннями в інтегровану автоматизовану систему реагування на інциденти та їх моніторингу.

Реалізація інтегрованої системи повинна бути здійснена на основі програмного засобу Elasticsearch із включенням всіх задіяних компонентів. Результатом є перевірка її працездатності та журналювання подій безпеки. В цілому, система повинна виконати моніторинг вибраного для дослідження веб-ресурсу і забезпечити якіску роботу спеціалістів під час її застосування.

Реалізація та налаштування системи моніторингу

Для моніторингу веб-ресурсу за допомогою даної системи необхідно спочатку виконати розгортання ELK-стеку. З цією метою використано операційну систему Linux Ubuntu з необхідними root-правами для виконання більшості необхідних для проведення моніторингу команд. Необхідністю є права суперкористувача в UNIX-подібних системах без накладання будь-яких обмежень на виконання команд.

Реалізація інтегрованої системи моніторингу виконувалася на одній фізичній машині, на якій у файлах конфігурації усі IP-адреси були встановлені нами на 127.0.0.1 чи localhost. Проте в подібних корпоративних системах встановлювати IP-адреси потрібно тільки на тих машинах, на яких встановлені необхідні для виконання даного моніторингу програмні продутки.

Формування системи почнемо зі встановлення пакета apt-transport-https, використавши команду, подану нижче:

```
sudo apt-get install apt-transport-https.
```

Для встановлення ELK-стеку необхідно під'єднати репозиторій, застосувавши команду, представлену нижче:

```
echo "deb https://artifacts.elastic.co/packages/7.x/apt stable
main" | sudo tee /etc/apt/sources.list.d/elastic-7.x.list.
```

Наступним кроком слід встановити всі програмні продукти, необхідні для функціонування ELK-стеку. Спочатку встановимо ElasticSearch, застосувавши команду:

```
sudo apt-get update && sudo apt-get install elasticsearch.
```

Далі встановлюємо програмний продукт Logstash [10], виконавши команду:

```
sudo apt-get update && sudo apt-get install logstash.
```

Наступним кроком повинен бути встановлений програмний продукт Kibana [5] командою:

```
sudo apt-get update && sudo apt-get install kibana.
```

Після встановлення Kibana необхідно встановити продукт Filebeat [12], застосувавши команди:

```
curl -L -O https://artifacts.elastic.co/downloads/
beats/filebeat/filebeat-7.13.0-amd64.deb;
```

та

```
sudo dpkg -i filebeat-7.13.0-amd64.deb.
```

Далі необхідно виконати процедуру налаштування до роботи всіх задіяних до даного моніторингу програмних продуктів. Для роботи Elasticsearch важливим є налаштування Java-машини з урахуванням виділення оперативної памяті в достатній для неї мірі. У разі, якщо кількість виділеної пам'яті для Elasticsearch буде не достатньою, то він буде сповільнювати виконання своїх функцій, або взагалі може припинити свою роботу, констатувавши помилку збою. Рекомендовано в цьому випадку йому надавати хоча б ¼ частину доступної пам'яті, і не менше 1 Ґбайта – в окремих випадках навантаження. В наведеному нами дослідженні для ElasticSearch було надано 4 Ґбайти оперативної пам'яті, а для його роботи – 1,28 Гбайт.

Файл конфігурації за замовчуваням знаходиться в директорії [9]:

```
/etc/elasticsearch/jvm.options.
##
-Xms1280m
-Xmx1380m
##.
```

Далі виконаємо налаштування програмного засобу Logstash [6]. За замовчуванням шлях до його файлу конфігурації знаходиться в директорії:

```
/etc/logstash/jvm.options.
```

Заодно одразу налаштуємо pipeline для Logstash [10], який відіграє роль спеціального "конвейєра", задіяний для обробки вхідних даних. Слід відмітити, що таких конвейєрів може бути кілька, кожен з яких може налаштовуватися для підтримки виконання різних задач. Для них необхідно виконати налаштування їх входу та виходу і, за потреби, – фільтрів.

```
- pipeline.id: main
input{
beats {
port => 5044
}
}
output {
elasticsearch{
hosts => ["http://localhost:9200"]
index           =>           "%{[@metadata][beat]}-
%{[@metadata][version]}"
}
}.
```

В даному випадку, як це видно з наведених команд, Elasticsearch виконуватиме прослуховування порту за номером 9200, а тому вивід програмного засобу Logstash слід налаштовувати на даний порт.

Для засобу Filebeat необхідно вказати шлях до файлів з метою виконання ним перевірки та виводу данних. Наведемо результат налаштування засобу Filebeat [11,12] нижче:

```
- type: filestream
# Change to true to enable
enabled: false
# Path that should be
paths:
- /var/log/*.log.
```

Представимо також налаштування виводу данних для засобу Filebeat [12]

```
output.logstash:
# The Logstash hosts
hosts: ["localhost:5044"].
```

Під час налаштування програмного продукту Kibana необхідно вказати адресу і номер порта сервера та надати доступ до програми ElasticSearch [5] (рис. 3).

```
server.port: 5601

# Specifies the address to which the Kibana serve
# The default is 'localhost', which usually means
# To allow connections from remote users, set thi
server.host: "127.0.0.1"

elasticsearch.hosts: ["http://127.0.0.1:9200"]
```

Рис. 3 – Результат налаштування засобу Kibana

Завершивши налаштування всіх програмних засобів, задіяних в моніторингу, необхідно виконати їх запуск. Для цього виконується ряд команд, наведених нжче:

```
sudo systemctl start elasticsearch;
sudo systemctl start logstash;
sudo systemctl start kibana;
sudo systemctl start filebeat.
```

Для перевірки робочого статусу всіх запущених на виконання програм, необхідно застосувати команду, наведену нижче, результат виконання якої для ElasticSearch наведений на рис. 4.

```
systemctl status elasticsearch.
```

```
kolya@kolya-VirtualBox:~$ systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2021-06-02 08:51:34 EEST; 3h 21min ago
       Docs: https://www.elastic.co
   Main PID: 669 (java)
      Tasks: 61 (limit: 2315)
     Memory: 658.7M
     CGroup: /system.slice/elasticsearch.service
             ├─ 669 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10 -XX
             └─1317 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller
```

Рис. 4 – Результат перевірки робочого статусу програми ElasticSearch

І, на завершення, потрібно запам'ятати, що для зупинки роботи всіх задіяних в даному моніторингу засобів слід виконати команду, наведену нижче:

```
sudo systemctl stop elasticsearch.
```

Обговорення системи проведення моніторингу web-ресурсу засобами ElasticSearch

Першим кроком прослідкуємо виконання програми Kibana, яка входить в систему моніторингу web-ресурсу, доступ до якої надається згідно наведених вище параметрів налаштувань. В даному випадку –дослідження localhost буде приймати значення 5601. Сюди входять також і програмні продукти Dashboard, Discover, Canvas і Maps (рис. 5).

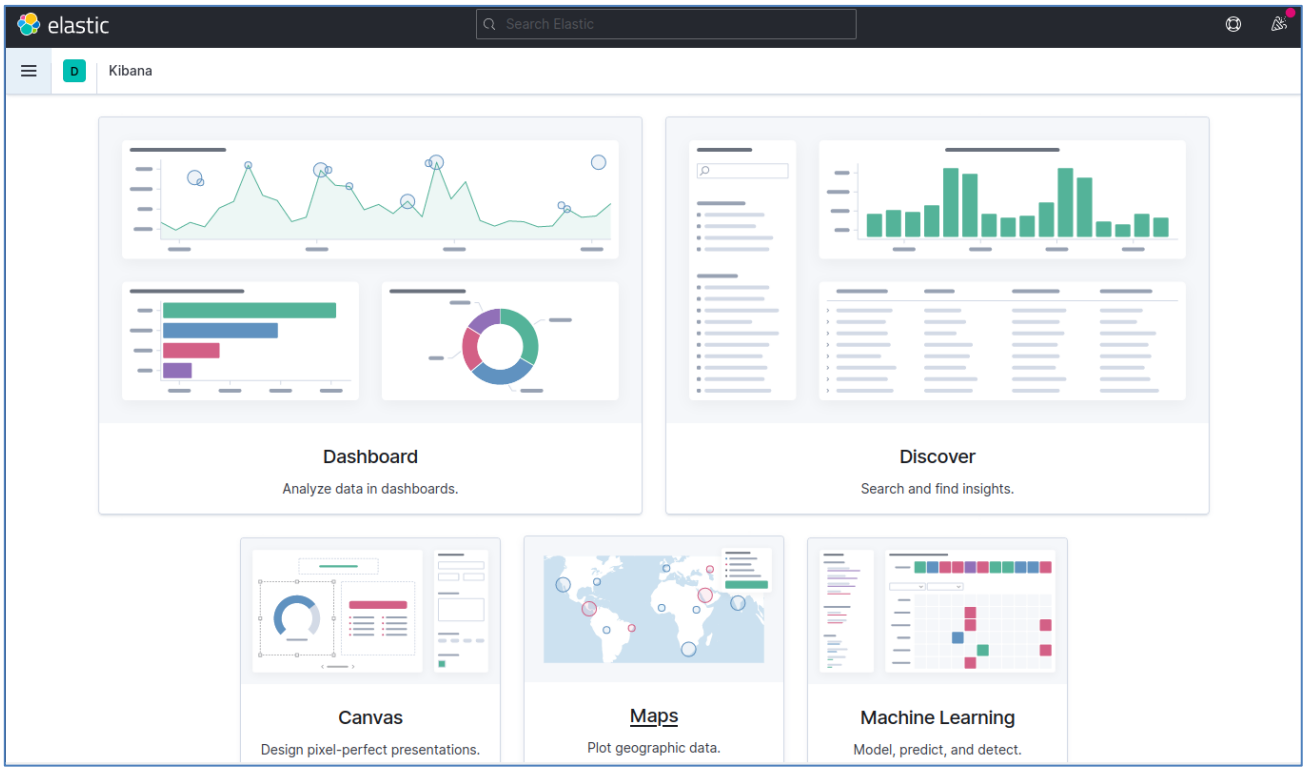

Рис. 5 – Вигляд робочого нтерфейсу програми Kibana

Наступним кроком виконаємо опис зазначених вище програмних інструментів. Почнемо першим описувати програмний інструмент Discover, який здійснює перегляд даних, а вигляд його робочого вікна наведений на рисунку 6.

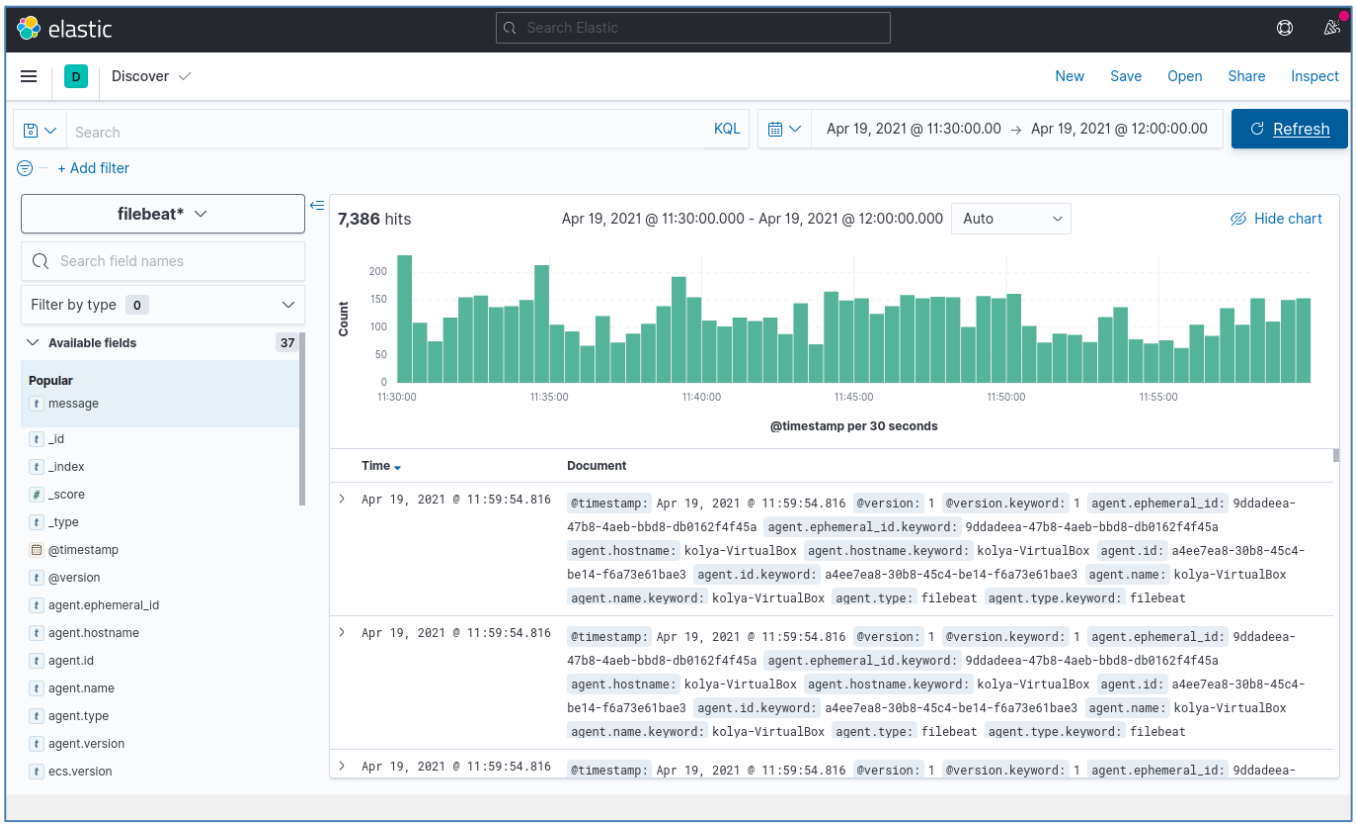

Рис. 6 – Вигляд робочого вікна програми Discover

Як видно з наведеного рисунку, на передньому плані Discover містить гістограму, призначену для виконання пошуку данних за відповідними атрибутами, такими як час їх створення та кількістю

прийнятих за визначений період часу повідомлень. Також з рисунка видно, що на гістограмі можуть бути помітними і аномалії відтворення (рис. 7).

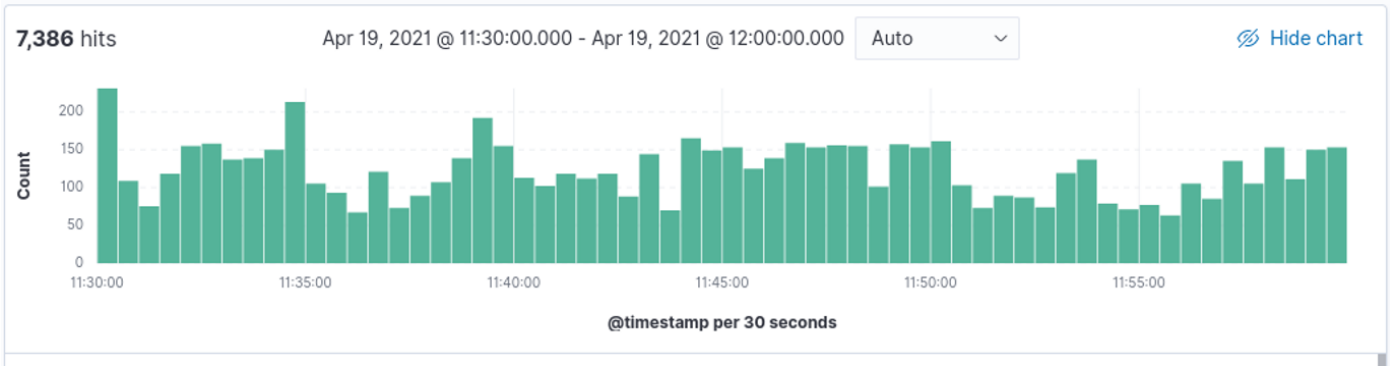

Рис. 7 – Гісторгама для пошуку даних у робочому вікні програми Discover

Справа вертикально розміщені поля, за якими реалізується можливість виконувати відповідну вибірку данних. Сукупність полів та їх значень наведено на рис. 8.

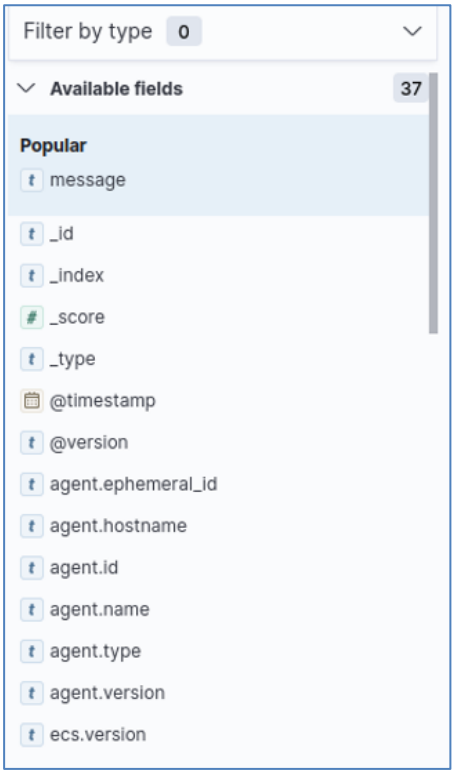

Рис. 8 – Вигляд полів програми Discover для пошуку даних

Для прикладу наведемо також перелік отриманих даних, які відображаються в зоні робочого вікна Discover нижче гістограми (рис. 9) і відповідають зазначеному часовому проміжку моніторингу під час виконання дослідження із відповідно налаштованим фільтруванням даних.

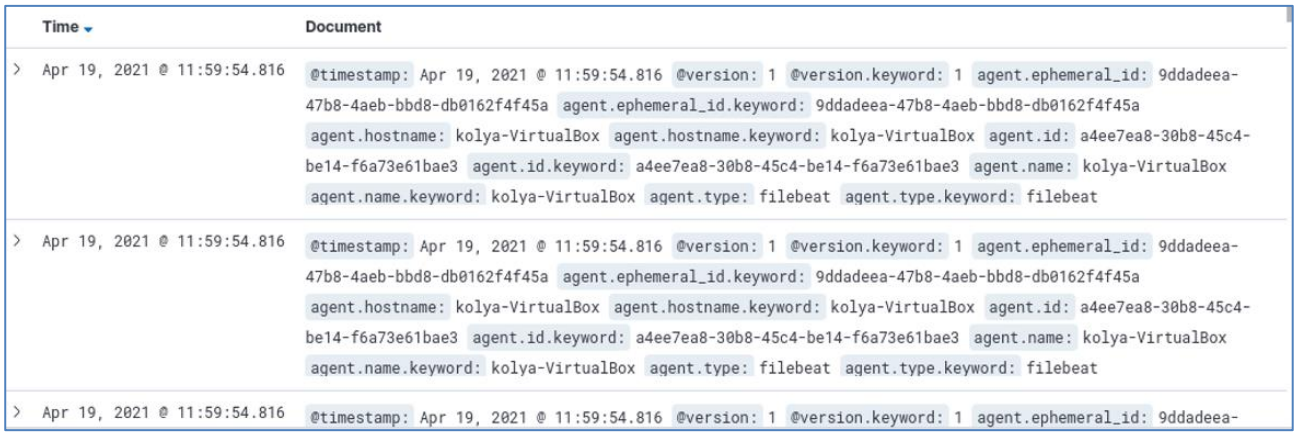

Рис. 9 – Представлення даних виконаного моніторингу в зазначений проміжок часу після фільтрування

Наступним є програмний інструмент Canvas, який призначений для візуалізації та презентації даних. Він здійснює вивантаження даних з програми моніторингу Elasticsearch та візуально поєднує їх з відповідно вибраними кольорами, зображеннями та текстом для створення динамічних мультисторінкових екранів перегляду.

Програма Canvas дозволяє також виконувати і інші операції, необхідні для проведення моніторингу, такі як створення робочого простору та його персоналізації, налаштування робочої панелі з використанням власних графічних відображень рисунків та тексту, вивантаження необхідних даних з Elasticsearch та виведення їх діаграмою, графіком чи текстовим моніторигом, фокусування отриманих та, відповідно, відфільтрованих даних з метою відображення та інші.

Програмний інструмент Dashboard надає можливість виводити інформацію у вигляді діаграм, гістограм, графіків та інших форм наочного представлення данних (рис. 10,11). Використовуючи Dashboard, можна здійснювати перетворення даних з одного або/чи кількох робочих шаблонів індексів на відповідний набір панелей, які надають можливість оптимізувати дані, а під час виконання аналізу зосередитися тільки на важливих із них.

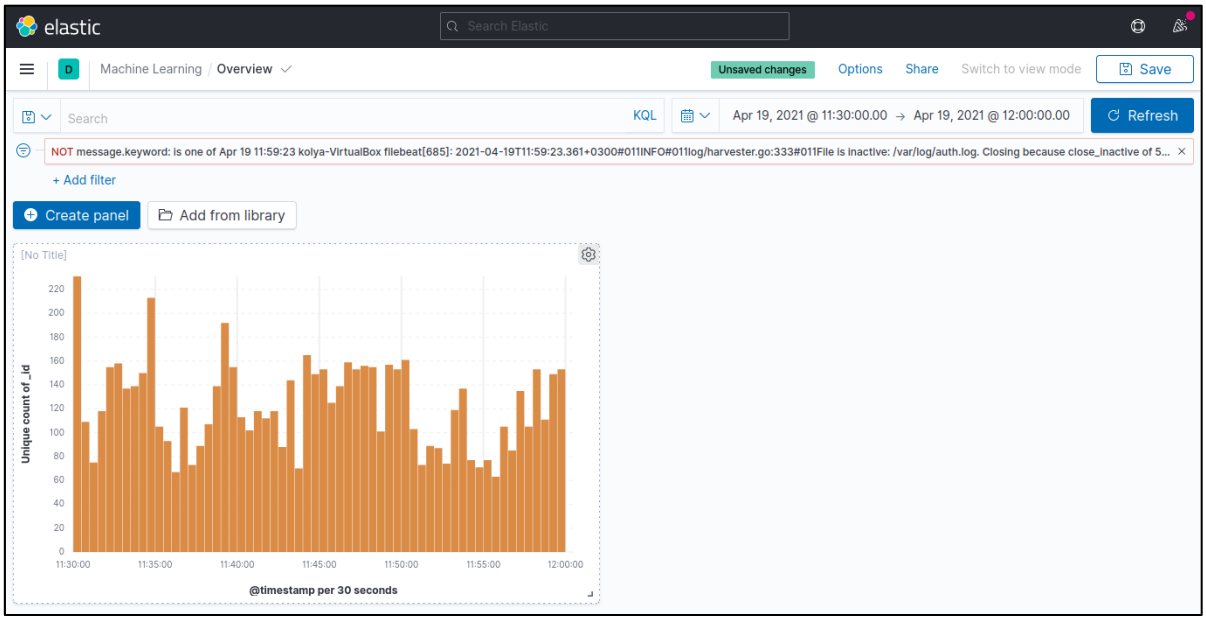

Рис. 10 – Наведена гістограма з підтримки відображення Dashboard

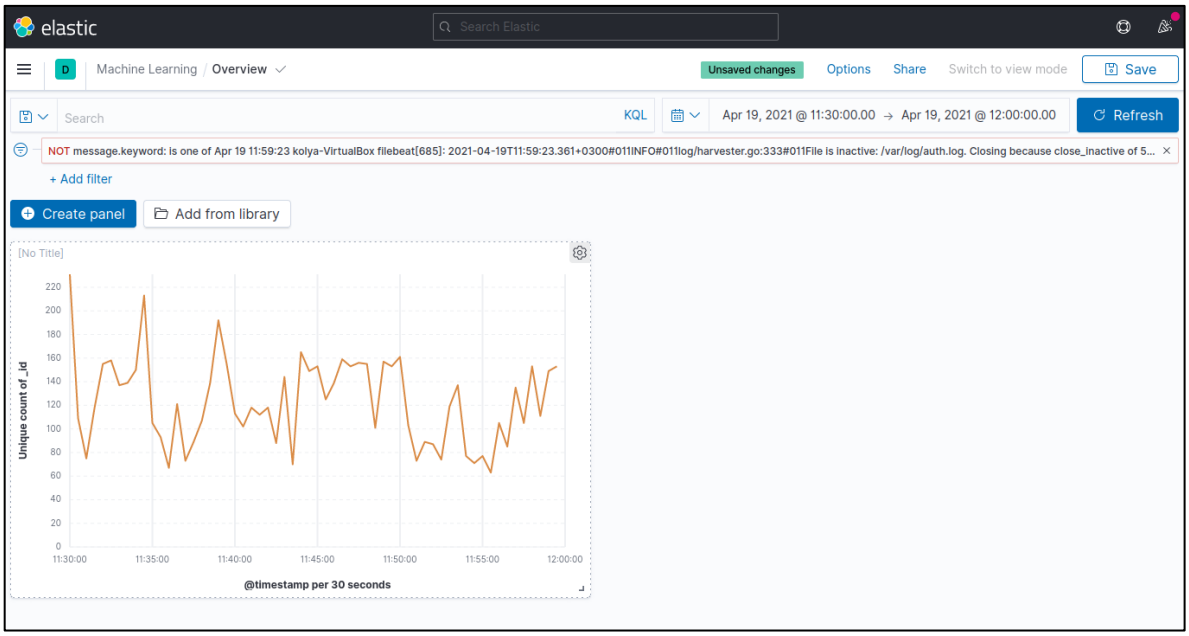

Рис. 11 – Приклад графіки з підтримки відображення Dashboard

Програмний інструмент Maps, що широко використовується в мобільних додатках [18], дає можливість відображати геодані, призначені для різного роду запитів. Для такого відображення треба інсталювати плагін Geoip для Logstash [19], за допомогою якого можна трансформувати IP-адреси в геодані, використовуючи відкриту базу даних. Ліцензоване машинне навчання дозволяє обробляти дані, отримані за допомогою нейронних мереж, які в даному випадку є потужним інструментом для знаходження аномалій в непомітних для програми чи людини місцях. Заодно, в інтерфейсі програмного засобу Kibana знаходиться Stack Management для управління такими параметрами-індексами, як ліцензія, параметризація, управління кластерними об'єднаннями, шаблонами індексації та інтерфейсом користувача.

Програмний засіб Alerting дає можливість сформувати основні положення та характеристики складних умов для їх виявлення у задіяних Kibana-додатках та застосовувати дії їх виконання. Використовуючи інтерфейс управління, засіб Alerting здатний централізовано керувати такими діями. До того ж, він надає набір сполучних ліній та правил стеку для використання і працює, активізуючи перевірки за встановленим розкладом та відповідним правилом, визначаючи їх реальні умови прояву. Якщо в ході перевірки намічена умова виконана, то правило висуває до виконання одну або/чи кілька пов'язаних дій, які взаємодіють зі службами Kibana, по можливості задіюючи сторонні інтегровані засоби.

З метою перевірки конкретних визначених умов кожне правило − це виконання на Kibana-сервері конкретного завдання в фоновому режимі, в склад якого входять три складові: умови виявлення, частота активізації перевірок та відповідні дії під час підтвердження прояву виконання зазначених умов. Для прикладу можна взяти моніторинг серверів, під час якого визначене правило має можливість перевіряти використання процесора на середнє значення, більше 0,9 на кожному, протягом визначеного інтервалу часу – останніх двох хвилин (рис. 12). Все це становить певну умову перевірки, яка стартує на виконання кожну хвилину (відповідно до розкладу), надсилаючи повідомлення у вигляді попереджень за темою "дія" через SMTP електронною поштою.

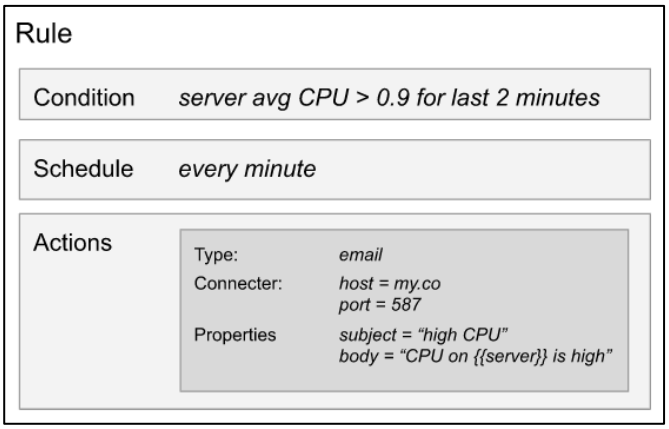

Рис. 12 – Приклад повідомлення попередження з правилом відслідковування

Правила Kibana становлять конкретні умови для запуску відповідної JavaScript-функції на Kibana-сервері. Така організація дій надає відповідної гнучкості для підтримки та перевірки широкого діапазону умов: від найпростішого – реалізації Elasticsearch-запиту, до складного – виконання громіздких обчислень над даними, що надходять з одного чи декількох джерел, якими можуть бути і зовнішні системи.

Розклади встановлених правил Kibana становлять визначені часові проміжки між активними перевірками, що можуть змінюватися від кількох секунд до кількох місяців. Дії виконання в межах правила втілюють виклики взаємодії з Kibana-сервісами або/чи сторонніми інтегрованими системами. Як уже згадувалося, вони виконуються в фоновому режимі у вигляді завдання на сервері Kibana, якщо виконуються задані умови правил. Результат виконання становить шаблон, якому надаються всі необхідні параметри для виклику відповідної служби, за винятком тих, які є відомими під час перевірки виконання умови в момент діючого правила.

Для моніторингу сервера можна використовувати адресу діючої електронної пошти з тілом повідомлення за допомогою запису:

```
.email server CPU on {{server}} is high.
```

Якщо відповідно до правила виявляється виконання певної умови (стану), воно генерує сповіщення, що в собі містить відомості про умову, шаблон з наданими даними та іменем сервера, і виконує призначену дію чи їх послідовність на сервері Kibana.

Однак в ході перевірки стану одне і те ж правило може виявляти виконання умови в кількох екземплярах серверів (рис. 13). В цьому випадку середовище Kibana відстежує кожне зі згенерованих сповіщень окремо і для виконання кожного із них вживає конкретних заходів. Таким чином, використовуючи подібний моніторинг набору серверів, кожен із них, в якого середнє завантаження процесора перевищуватиме 0,9, буде відстежуватися зі сповіщенням. Для кожного із таких серверів буде надсилатися електронною поштою окреме повідомлення, яке сповістить перевищення встановленого граничного значення навантаження процесора.

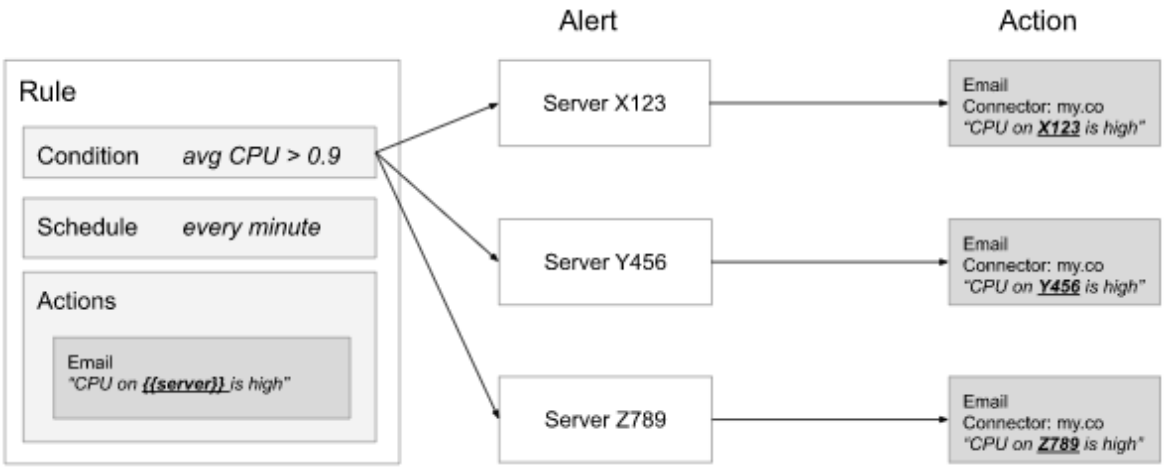

Рис. 13 – Виконання правила для кількох серверів

Висновки

Велика кількість web-ресурсів в Інтернет, їх безпечне функціонування та використання в наш час потребує реалізації досконалих підходів та технологій захисту для забезпечення їх надійності роботи. З метою запобігання вторгнень та зломів web-розробок успішно можуть бути використані SIEM-системи, які досить швидко реагують на інциденти. Важливим є і те, що сьогодні розробники SIEM-систем по різному вдосконалюють їх, в тому числі і в напрямку автоматизації. Подібні дослідження корисні і в тому сенсі, що в майбутньому подібні удосконалені системи можуть вести автоматичне відслідковування та моніторинг, швидко реагувати на прояви порушень безпеки ресурсів Інтернет будь-якого виду. Подібні системи можна удосконалювати за допомогою поєднання новітніх підходів моніторингу, параметризації та нейронних мереж.

Безпеку функціонування web-розробок в наш час підтримують і технології машинного навчання з залученням ELK-стеку. Також програмний засіб Elastic регулярно оновлюється, що спонукає до зростання надійності та зниження вірогідності компроментації наведеної системи, враховуючи і його безкошковну версію, яка в деякій мірі задовільняє потреби контролю безпеки. Його платна версія є все-таки досить потужним інструментом. Сьогодні програмний засіб Elasticseartch є досить гнучкою та перспективною системою для проведення подальших досліджень з метою реалізації моніторингу веб-ресурсів, враховуючи показники зниження часу реагування на інцинденти, простоти налаштування під задачі різного типу та мінімумом використання системних ресурсів.

Список використаних джерел

1. Огляд рішень Security Information and Event Management (SIEM). – 2020. – Електронний ресурс: Режим доступу: https://habr.com/ru/company/roi4cio/blog/528770/.
2. Installing the Elastic Stack. Upgrade to Elastic 8.6.2. – 2022. – [Електронний ресурс] – Режим доступу: https://www.elastic.co/guide/en/elastic-stack/current/installing-elastic-stack.html.
3. How To Install Elasticsearch, Logstash, and Kibana (Elastic Stack) on Ubuntu 22.04. [Електронний ресурс] – Режим доступу: https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elastic-stack-on-ubuntu-22-04.
4. Installing Logstash. Logstash Reference: Multiple Pipelines. – 2023. – [Електронний ресурс] – Режим доступу: https://www.elastic.co/guide/en/logstash/current/installing-logstash.html
5. Elastic. Kibana – your window into Elastic.– 2023. – [Електронний ресурс] – Режим доступу: https://www.elastic.co/guide/en/kibana/current/introduction.html.
6. Setting Up and Running Logstash. Logstash.yml. – 2023. – [Електронний ресурс] – Режим доступу: https://www.elastic.co/guide/en/logstash/current/logstash-settings-file.html
7. Configuring Logstash | Logstash Reference [7.17] – Elastic. Configuring Logstash. – 2023. – [Електронний ресурс]. Режим доступу: https://www.elastic.co/guide/en/logstash/current/configuration. html.
8. Multiple Pipelines. Logstash Reference [8.6] – Elastic. – 2023. – [Електронний ресурс] – Режим доступу: https://www.elastic.co/guide/en/logstash/current/multiple-pipelines.html.
9. Guide to the Most Important JVM Parameters. JVM.settings. – 2023. – [Електронний ресурс] – Режим доступу: https://www.baeldung.com/jvm-parameters.
10. Puttagunta C., Kulkarni A., Bannon S. et al. Elastic. Creating a Logstash pipeline. Structure of a Config File [Електронний ресурс] – Режим доступу: https://www.elastic.co/guide/en/logstash/current/ configuration-file-structure.html.
11. Accessing Event Data and Fields in the Configuration [Електронний ресурс] – Режим доступу до ресурсу: https://www.elastic.co/guide/en/logstash/current/event-dependent-configuration.html.
12. Configuring Logash. Filebeat overview. – 2023. – [Електронний ресурс] – Режим доступу: https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-overview.html.
13. Metricbeat Reference. Metricbeat overview [Електронний ресурс] – Режим доступу: https://www.elastic.co/guide/en/beats/metricbeat/current/metricbeat-overview.html.
14. Getting Started with Winlogbeat. Winlogbeat Overview [Електронний ресурс] – Режим доступу: https://www.elastic.co/guide/en/beats/winlogbeat/current/ winlogbeat_overview.html.
15. Getting Started with Packetbeat. Packetbeat overview. – 2023. – [Електронний ресурс] – Режим доступу: https://www.elastic.co/guide/en/beats/packetbeat/current/packetbeat-overview.html.
16. Auditbeat Reference: Auditbeat overview. – 2023. – [Електронний ресурс] – Режим доступу: https://www.elastic.co/guide/en/beats/auditbeat/current/auditbeat-overview.html.
17. Heartbeat Reference. Heartbeat overview. – 2023. – [Електронний ресурс] – Режим доступу: https://www.elastic.co/guide/en/beats/heartbeat/current/heartbeat-overview.html.
18. MAPS.ME. – 2023. – [Електронний ресурс] – Режим доступу: https://android.biblprog.org.ua/ua/ maps-me/.
19. Logstash Setup with GeoIP – Discuss the Elastic Stack. – 2023. – [Електронний ресурс] – Режим доступу: https://www.elastic.co/guide/en/logstash/current/plugins-filters-geoip.html#_description_134.

References

1. Overview of Security Information and Event Management (SIEM) solutions. – 2020. – Electronic resource: Access mode: https://habr.com/ru/company/roi4cio/blog/528770/.

2. Installing Elastic Stack. Upgrade to Elastic 8.6.2. – 2022. – [Electronic resource] – Access mode: https://www.elastic.co/guide/en/elastic-stack/current/installing-elastic-stack.html.
3. How to install Elasticsearch, Logstash and Kibana (Elastic Stack) on Ubuntu 22.04. [Electronic resource] - Access mode: https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elastic-stack-on-ubuntu-22-04.
4. Installing Logstash. Logstash link: multiple pipelines. – 2023. – [Electronic resource] – Access mode: https://www.elastic.co/guide/en/logstash/current/installing-logstash.html.
5. Elastic. Kibana – your window into Elastic.– 2023. – [Electronic resource] – Access mode: https://www.elastic.co/guide/en/kibana/current/introduction.html.
6. Configuring and running Logstash. Logstash.yml. – 2023. – [Electronic resource] – Access mode: https://www.elastic.co/guide/en/logstash/current/logstash-settings-file.html.
7. Configuring Logstash | Link to Logstash [7.17] – Elastic. Configuring Logstash. – 2023. – [Electronic resource]. Access mode: https://www.elastic.co/guide/en/logstash/current/configuration.html.
8. Several conveyors. Links to Logstash [8.6] – Elastic. – 2023. – [Electronic resource] – Access mode: https://www.elastic.co/guide/en/logstash/current/multiple-pipelines.html.
9. Guide to the most important JVM parameters. JVM.settings. – 2023. – [Electronic resource] – Access mode: https://www.baeldung.com/jvm-parameters.
10. Puttagunta C., Kulkarni A., Bannon S. and others. Elastic. Creating a Logstash pipeline. Structure of a Config File [Electronic resource] – Access mode: https://www.elastic.co/guide/en/logstash/current/ configuration-file-structure.html.
11. Accessing Event Data and Fields in the Configuration [Electronic resource] – Resource access mode: https://www.elastic.co/guide/en/logstash/current/event-dependent-configuration.html.
12. Configuring Logash. Filebeat Review. – 2023. – [Electronic resource] – Access mode: https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-overview.html.
13. Metricbeat Reference. Metricbeat overview [Electronic resource] – Access mode: https://www.elastic.co/guide/en/beats/metricbeat/current/metricbeat-overview.html.
14. Getting started with Winlogbeat. Winlogbeat Overview [Electronic resource] – Access mode: https://www.elastic.co/guide/en/beats/winlogbeat/current/ winlogbeat_overview.html.
15. Getting started with Packetbeat. Packetbeat overview.– 2023. – [Electronic resource] – Access mode: https://www.elastic.co/guide/en/beats/packetbeat/current/packetbeat-overview.html.
16. Auditbeat Reference: Auditbeat overview.– 2023. – [Electronic resource] – Access mode: https://www.elastic.co/guide/en/beats/auditbeat/current/auditbeat-overview.html.
17. Heartbeat Reference. Heartbeat overview.– 2023. – [Electronic resource] – Access mode: https://www.elastic.co/guide/en/beats/heartbeat/current/heartbeat-overview.html.
18. MAPS.ME. – 2023. – [Electronic resource] – Access mode: https://android.biblprog.org.ua/ua/maps-me/.
19. Configuring Logstash with GeoIP – Discuss Elastic Stack. – 2023. – [Electronic resource] – Access mode: https://www.elastic.co/guide/en/logstash/current/plugins-filters-geoip.html#_description_134