МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ФРАНКА
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

ITSec-2024

Безпека інформаційних технологій

МАТЕРІАЛИ

XIII Міжнародної науково-технічної конференції

9-11 травня 2024
м. Львів (Україна)

УДК [003.26+004+519.816]:004.056:65(063)

ITSec: Безпека інформаційних технологій: матеріали XIII Міжнар. наук.-техн. конф., м. Львів, 9-11 трав. 2024 р. Л.: ЛНУ ім. І. Франка, 2024, 257 с.

Збірник містить тексти наукових матеріалів доповідей та тез учасників XIII міжнародної науково-технічної конференції «ITSec: Безпека інформаційних технологій». Основною метою конференції є ознайомлення з сучасними досягненнями та висвітлення результатів наукових досліджень з усіх аспектів кібербезпеки та захисту інформації.

Призначено вченим, інженерам, аспірантам наукових спеціальностей 05.13.21 – Системи захисту інформації, 21.05.01 – Інформаційна безпека держави, здобувачам вищої освіти за спеціальностями: 125 – Кібербезпека та захист інформації, а також всім зацікавленим.

ЗМІСТ

Дослідження log-файлів засобами платформи Elastic Stack

УДК 004.4:004.896

Ольга Смотр[1], Микита Купріков[2]

Львівський державний університет безпеки життєдіяльності,
[1]olgasmotr@gmail.com, [2]nkuprikov@ldubgd.edu.ua

Однією з найбільш поширених загроз для даних є несанкціонований доступ, що може призвести до втрати конфіденційності, цілісності та доступності інформації. Адже у сучасному світі, де взаємозв'язки набувають все більшого значення, конфіденційність інформації про продукти, процеси, клієнтів та постачальників є критично важливою для будь-якої компанії. Особливо відчутною ця необхідність стає в контексті спалаху інформаційної війни, що супроводжує повномасштабне вторгнення російських військ на територію України у 2022 році. Відповідно до прогнозів спеціалістів Cybersecurity Ventures на 2024 рік витрати світової спільноти на кіберзлочинність становитимуть близько 9,5 трильйонів доларів США. [1]. В свою чергу, кількість кібератак з метою заволодіння даними зросте більш ніж на 18%, середня вартість збитків від витоку даних становитиме понад 4,5 млн доларів США. [2].

Для розв'язання цих проблем на передньому краї технологічного розвитку стоїть аналіз log-файлів. Log-файли представляють собою записи подій, які відбуваються в системі або програмі, і є важливим інструментом для виявлення аномальних або підозрілих активностей. Аналіз log-файлів допомагає розуміти, як функціонує система та виявляти аномальну поведінку, що може свідчити про зловживання доступом або зловмисну діяльність. Це може бути корисно при розслідуванні інцидентів безпеки та при виявленні вразливостей, які можуть бути використані для атак на систему. Однак, аналіз log-файлів є трудомістким завданням, що, як правило, є складним для ручного аналізу. зважаючи на їх значний розмір та різні формати, залежно від того, яка система їх генерує. Адже log-файли можуть бути текстовими, бінарними або ж представленими у інших спеціальних форматах для зберігання даних. Для ефективного аналізу log-файлів доречно використовувати методику структурного аналізу даних, з метою відокремити корисну інформацію від зайвої та виявити можливі загрози безпеці даних та алгоритми машинного навчання для виявлення аномальної активності в системі.

На сьогодні на ринку існує ряд платформ з інструментарієм аналізу log-файлів. Це такі як Elastic Stack, Splunk, Graylog, Fluentd тощо. Для подальшої роботи з log-файлами використовуватимемо Elastic Stack. Elastic Stack (ELK) складається з таких компонентів як: Elasticsearch, Logstash, Kibana та Beats, і забезпечує комплексне рішення для збору, аналізу та візуалізації даних. Вона має ряд переваг: висока швидкість, масштабованість, а також розширені можливості візуалізації та аналізу даних [3-6]. Однак, слід зазначити, що вона також має свої недоліки, зокрема, складність налаштування та використання для не досвідчених користувачів. Зокрема продемонструємо процес розробки фільтрів для Logstash .

Розробка фільтрів для Logstash є одним із вирішальних етапів у створенні системи аналізу log-файлів. Фільтри дозволяють трансформувати та розбирати вхідні дані на більш деталізовані складові, що спрощує подальший аналіз. Одним з

основних типів фільтрів, які будуть розроблені в рамках даної роботи, є фільтр "grok". Фільтр grok дозволяє структурувати неструктуровані log-файли, шляхом використання шаблонів. Цей інструмент ідеально підходить для обробки логів syslog, Apache та інших веб-серверів, MySQL та, загалом, будь-якого формату логів, який зазвичай пишуть для людей, а не для обробки комп'ютерами.

Основа роботи з Grok полягає в комбінуванні текстових шаблонів у такий, що відповідає вашим логам. Синтаксис для шаблону grok виглядає так:

```
%{SYNTAX:SEMANTIC}
```

де SYNTAX - це назва шаблону, який буде збігатися з вашим текстом. Наприклад, число 3.44 буде збігатися з шаблоном NUMBER, а IP-адреса 55.3.244.1 - з шаблоном IP. SEMANTIC - це ідентифікатор, який ви задаєте частині тексту, яка збігається. Наприклад, число 3.44 може бути тривалістю події, тому ви можете іменувати його просто "duration". Рядок "55.3.244.1" може ідентифікувати клієнта, який робить запит. У цьому випадку наш фільтр grok міг би виглядати так:

```
%{NUMBER:duration} %{IP:client}
```

Також можемо додати до свого шаблону grok перетворення типів даних. За замовчуванням, всі семантичні складові зберігаються як рядки. Якщо ви хочете змінити тип даних семантичної складової, наприклад, змінити рядок на ціле число, додайте до нього цільовий тип даних. Наприклад, %{NUMBER:num:int}, який конвертує семантичну складову "num" з рядка в ціле число. На даний момент підтримуються тільки перетворення в int і float.

Розглянемо, як за допомогою синтаксису і семантики можна витягти корисні поля, для прикладу, з такого логу HTTP-запиту:

```
55.3.244.1 GET /index.html 15824 0.043
```

Шаблон для цього логу може бути наступним:

```
%{IP:client} %{WORD:method}
%{URIPATHPARAM:request}
%{NUMBER:bytes} %{NUMBER:duration}
```

У кінцевому результаті наш файл конфігурації виглядатиме так (рис. 1).

```
input {
  stdin { } # Вхідний логік для отримання даних з консолиfile {
    path => "/var/log/http.log" # Шлях до файлу журналу, який потрібно обробити
  }
}
filter {
  grok {
    match => { "message" => "%{IP:client} %{WORD:xethod}
%{URIPATHPARAM:request} %{NUMBER:bytes} %{NUMBER:duration}" }
  }
}
output { stdout {
    codec => rubydebug # Вивід даних у форматі Ruby-відладки
  }
  elasticsearch {
    hosts => ["http://localhost:9200"] # Адреси серверівElasticsearch
    index => "test.logstash" # Індекс, у який будуть зберігатись
дані
    user => "elastic" # Користувач Elasticsearch
    password => "3guzVh6WIwXCb*qVFQfp" # Пароль користувача
Elasticsearch
```

Рис.1. Лістинг - Конфігурація Logstash-файлу «learn.conf» для обробки журналу HTTP

Наш запит можемо здійснити через командний рядок або ж у Kibana. Результат запиту з командного рядка та Kibana відображено на рисунках 2 та 3 відповідно.

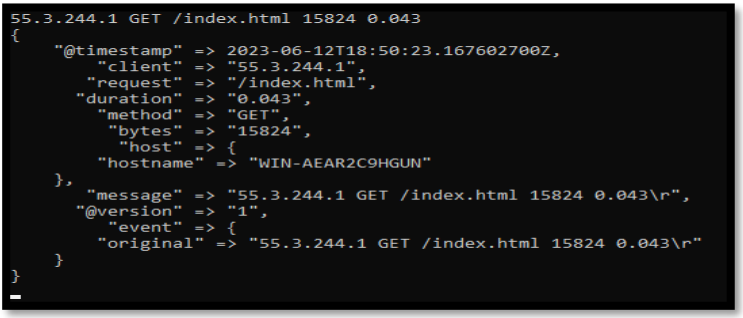

Рисунок 2 - Результат зчитування log-файлу у командному рядку

Запит "55.3.244.1 GET /index.html 15824 0.043" був оброблений Logstash і вивід результату показує, що дані були успішно розпізнані і відформатовані за допомогою grok-виразу. Кожен елемент запиту відображений відповідним полем, наприклад, IP-адреса ("client"), HTTP-метод ("method"), URL-шлях ("request"), обсяг байтів ("bytes") та тривалість ("duration"). Додаткові поля, такі як "@timestamp" та "host.hostname", також відображаються. Загальний результат містить ключі та значення для кожного поля, яке було оброблено. Ці дані можна використовувати для подальшої аналітики, візуалізації та зберігання у Elasticsearch для подальшого використання.

У Kibana, в розділі "Expanded document" ми також можемо побачити розширені дані документу, які були збережені у Elasticsearch після обробки Logstash (див. рис. 3).

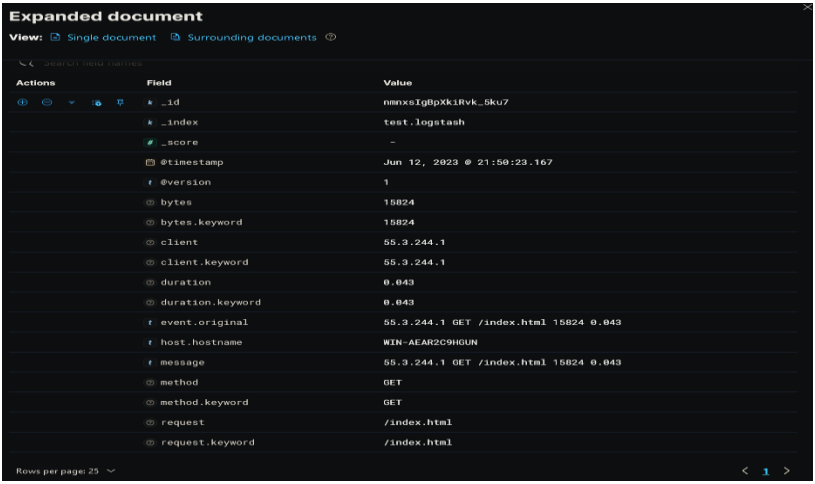

Рисунок 3 - Результати нашого запиту у Kibana

Тепер за допомогою Kibana можна встановлювати фільтри за різними полями, такими як "client", "method" або "request", і швидко отримувати підсумкові дані для конкретних сегментів інформації. Окрім того, Kibana пропонує нам розширені можливості фільтрування та пошуку. Ми можемо виконувати розширені пошукові запити, використовуючи синтаксис Elasticsearch Query DSL, щоб знайти конкретні документи або аналізувати дані за певними умовами. Як наслідок, ми можемо точно налаштувати свої запити та отримувати результати, які відповідають конкретним потребам користувача. Наприклад, можна налаштувати систему моніторингу трафіку на мережевому рівні, щоб виявляти незвичайну активність, таку як велика кількість запитів від одного IP-адреси або незвичайні шаблони поведінки.

Висновок. Застосування системи Elastic Stack для аналізу log-файлів є доволі ефективним засобом виявлення потенційних загроз безпеці інформації. Даний підхід дозволяє підвищити рівень захисту інформації та забезпечити вчасне виявлення можливих загроз. Під час аналізу загроз безпеці інформації в системі ELK, можливо виявити різноманітні типи атак, такі як діяльність шпигунів, спам-атаки, фішингові атаки, вторгнення в систему, DDOS-атаки та інші. Для виявлення таких атак можна використовувати різноманітні техніки та методи, такі як: машинне навчання, статистичний аналіз, розробка правил або комбінування цих підходів.

Аналіз log-файлів надає можливість виявляти та вирішувати проблеми безпеки, що виникають в системі, а також розгортати проактивні заходи безпеки, щоб запобігти майбутнім інцидентам. Розробка шаблонів та фільтрів для аналізу log-файлів є ключовою складовою налаштуванню Elasticsearch. Вона передбачає створення шаблонів для Elasticsearch, які визначають структуру даних та надають змогу ефективно зберігати та швидко отримувати доступ до них. Розробка фільтрів для Logstash допомагає приймати, обробляти та передавати дані в Elasticsearch для подальшого аналізу. Налаштування та впровадження системи ELK для аналізу log-файлів вимагає ретельного планування і налаштування, але при цьому дозволяє здійснювати ефективний моніторинг та виявлення потенційних загроз безпеки.

1. eSentire. Офіційний звіт про кіберзлочинність за 2023 рік. URL: https://www.esentire.com/resources/library/2023-official-cybercrime-report (дата звернення: 12.04.2024).
2. H-X Technologies. Прогнози на 2024 рік. URL: https://www.h-x.technology/ua/blog-ua/cyber-threats-forecast-2024-ua (дата звернення: 12.04.2024).
3. Elastic N.V., "Elastic Stack Documentation." - [Електронний ресурс]. - Режим доступу : https://www.elastic.co/guide/index.html (дата звернення: 01.04.2024)
4. Kuprikov M, Smotr O. Monitoring and analysis of large amounts of data using the elastic stack platform. Information security and information technologies IBIT-2023: collection of abstracts of the VI All-Ukrainian scientific and practical conference, November 30, 2023. - Lviv, LSU of Life Safety, 2023. - P.341-343.

НАУКОВЕ ВИДАННЯ

МАТЕРІАЛИ
XIII Міжнародної науково-технічної конференції
«ITSec: Безпека інформаційних технологій»

9-11 травня 2024 року

м. Львів (Україна)

Організаційний комітет конференції та редакція можуть не поділяти думки авторів і не несуть відповідальність за достовірність викладеної інформації.

За науковий зміст і викладення матеріалу, достовірність та коректність фактичних даних (у тому числі класифікаційного індексу УДК) уся відповідальність покладається на авторів та їх наукових керівників.

Неінформативний текст матеріалів доповіді міг бути скорочений або вилучений на розсуд Оргкомітету конференції.

Оригінал-макет підготовлено на кафедрі кібербезпеки
Львівського національного університету імені Івана Франка