Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(освітній рівень)

на тему: *"Аналіз лог-файлів з використанням ELK для виявлення потенційних загроз безпеці"*

Виконав: студент (ка)
Спеціальності:
125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)
Чурбаков К.О.
підпис (прізвище та ініціали)

Керівник *Козак Р.О.*
підпис (прізвище та ініціали)

Нормоконтроль *Лобур Т.Б.*
підпис (прізвище та ініціали)

Завідувач кафедри *Загородна Н.В.*
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль - 2023

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет ком'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
« » 2023 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

студенту Чурбакову Константину Олексійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Аналіз лог-файлів з використанням ELK для виявлення потенційних загроз безпеці

Керівник роботи Козак Р.О., к.т.н, доц. каф. КБ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «*03*» *04* 2023 року № *4/7-349*

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи Колекція лог-файлів з різних систем, встановлені та налаштовані Компоненти стека ELK (Elasticsearch, Logstash, Kibana), документація системи ELK

4. Зміст роботи (перелік питань, які потрібно розробити)
Теоретичні аспекти аналізу лог-файлів та системи ELK
Відмінності логів у Windows та Linux, система журналювання подій Windows та Linux
Принципи збору та аналізу лог-файлів з використанням ELK
Аналіз загроз безпеці інформації, включаючи статистику та аналіз атак
Практичні приклади використання аналізу лог-файлів для виявлення загроз
Впровадження системи для аналізу лог-файлів з використанням ELK

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Пилипець М.У., д.т.н., професор каф. МТ		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення із завданням до кваліфікаційної роботи	16.01 - 19.01	Виконано
2.	Підбір та опрацювання джерел із теоретичних аспектів аналізу лог-файлів та системи ELK	20.01 - 05.02	Виконано
3.	Вивчення відмінностей логів у Windows та Linux, їхні системи журналювання	06.02 - 22.02	Виконано
4.	Вивчення та аналіз принципів збору та аналізу лог-файлів з використанням ELK.	23.02 - 20.03	Виконано
5.	Підбір та аналіз джерел щодо загроз безпеці інформації, включаючи статистику та аналіз атак, класифікацію загроз.	21.03 - 05.04	Виконано
6.	Практична робота з прикладами використання аналізу лог-файлів для виявлення загроз.	06.03 - 17.04	Виконано
7.	Встановлення та налаштування Elasticsearch, Logstash і Kibana для впровадження системи аналізу лог-файлів.	18.04 - 29.04	Виконано
8.	Розробка шаблонів та фільтрів для аналізу лог-файлів в системі ELK.	30.04 - 13.05	Виконано
9.	Тестування та налагодження розроблених шаблонів та фільтрів.	14.05 - 21.05	Виконано
10.	Розробка розділу про вимоги ергономіки до організації робочого місця оператора ПК.	22.05 - 05.06	Виконано
11.	Написання висновків та формування списку використаних джерел.	06.06 - 07.06	Виконано
12.	Оформлення додатків та фінальна верифікація всіх розділів роботи.	07.06 - 09.06	Виконано
13.	Нормоконтроль	10.06 - 11.06	Виконано
14.	Перевірка на плагіат	12.06 - 15.06	Виконано
15.	Попередній захист кваліфікаційної роботи	16.06 - 19.06	Виконано
16.	Захист кваліфікаційної роботи	21.06.2023	

Студент _____ Чурбаков К.О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Козак Р.О.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Аналіз лог-файлів з використанням ELK для виявлення потенційних загроз безпеці // Кваліфікаційна робота ОР «Бакалавр» // Чурбаков Константин Олексійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБ-41 // Тернопіль, 2023 // С. - 84, рис. -43 , додат. - 3.

Ключові слова: ELK, ЛОГ-ФАЙЛИ, БЕЗПЕКА, ЗАГРОЗИ, АНАЛІЗ ДАНИХ.

Метою дипломного проєкту є дослідження можливостей використання стеку ELK для аналізу лог-файлів з метою виявлення потенційних загроз в безпеці.

Для досягнення поставленої мети потрібно вирішити такі завдання:

– зібрати лог-файли з різних джерел та джерел зберігання даних;

– завантажити та індексувати ці лог-файли в ELK стек;

– налаштувати систему відслідковування лог-файлів та збір метрик з ELK;

– розробити та застосувати алгоритми аналізу лог-файлів для виявлення потенційних загроз безпеці;

– візуалізувати результати аналізу за допомогою інструментів візуалізації даних в ELK.

Об'єктом дослідження є лог-файли, які збираються з різних джерел.

Предметом дослідження є алгоритми аналізу даних для виявлення потенційних загроз безпеці та їх візуалізація, а також процес автоматичного сповіщення та плани дій щодо вирішення виявлених проблем.

Практична вагомість. Аналіз лог-файлів є важливою складовою безпеки комп'ютерних систем. Зловмисники можуть скористатися вразливостями

системи, щоб отримати незаконний доступ до конфіденційної інформації. Виявлення потенційних загроз безпеці та реагування на них є надзвичайно важливим для забезпечення безпеки та збереження даних. Використання ELK для аналізу лог-файлів дозволяє автоматизувати процес виявлення загроз та спрощує процес реагування на них, що робить цю технологію незамінною для практичного застосування в сфері безпеки комп'ютерних систем.

ANNOTATION

Analysis of log files using ELK to identify potential security threats // Qualification work for bachelor's degree // Churbakov Konstantyn Oleksiyovych // Ternopil National Technical University named after Ivan Pului, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, group СБ-41 // Ternopil, 2023 // P. - 84, Fig. - 43, Supplement - 3

Keywords: ELK, LOG FILES, SECURITY, THREATS, DATA ANALYSIS.

The purpose of the graduation project is to investigate the possibilities of using the ELK stack to analyze log files in order to identify potential security threats.

To achieve this goal, it is necessary to solve the following tasks:

– collect log files from various sources and data storage sources;

– upload and index these log files to the ELK stack;

– set up a system for tracking log files and collecting metrics from ELK;

– develop and apply algorithms for analyzing log files to identify potential security threats;

– visualize the results of the analysis using data visualization tools in ELK;

The object of the study is the log files collected from various sources.

The subject of the study is data analysis algorithms for identifying potential security threats and their visualization, as well as the process of automatic notification and action plans to address the identified problems.

Practical significance. Analyzing log files is an important component of computer system security. Attackers can exploit system vulnerabilities to gain illegal access to confidential information. Detecting and responding to potential security threats is crucial to ensure the security and safety of data. Using ELK to analyze log files automates the process of detecting threats and simplifies the process of responding to them, making this technology indispensable for practical use in the field of computer system security.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ELK - ElasticSearch, Logstash, Kibana

FIM (File Integrity Monitoring) - моніторинг цілісності файлів

МСБ - малий та середній бізнес

DoS (Denial of Service) - відмова у обслуговуванні

DDoS (Distributed Denial of Service) - розподілена відмова у обслуговуванні

США - Сполучені Штати Америки

ФБР - Федеральне Бюро Розслідувань

API (Application Programming Interface) - інтерфейс програмування додатків

rps - запитів за секунду

MD5 (Message Digest Algorithm 5) - алгоритм гешування повідомлень 5

SHA1 (Secure Hash Algorithm 1) - безпечний алгоритм хешування 1

SHA256 (Secure Hash Algorithm 256-bit) - безпечний алгоритм хешування 256-бітний

ПЗ - програмне забезпечення

IP (Internet Protocol) - інтернет-протокол

CBD-списки - списки центральної бази даних

TCP (Transmission Control Protocol) - протокол контролю передачі

RST - скидання

AFM/DHD (Advanced Firewall Manager / Distributed Denial of Service Hybrid Defender) - розширений менеджер брандмауера / гібридний захисник від розподілених відмов у обслуговуванні

VS/PO (Virtual Server / Protected Object) - віртуальний сервер / захищений об'єкт

IPI (Intelligent Platform Interface) - інтелектуальний платформенний інтерфейс

PPS - пакетів за секунду

DNS (Domain Name System) - система доменних імен

TLS (Transport Layer Security) - безпека транспортного рівня

HTTP (HyperText Transfer Protocol) - протокол передачі гіпертексту

HTTPS (HyperText Transfer Protocol Secure) - безпечний протокол передачі гіпертексту

ВСТУП

Розвиток інформаційних технологій та комп'ютерних мереж значно збільшив можливості обміну та обробки інформації для сучасних компаній. Завдяки цьому виникли нові методи ведення бізнесу та співпраці з клієнтами, партнерами та постачальниками. Однак, зростання залежності від інформаційних технологій призвело до збільшення числа потенційних загроз безпеки, що можуть мати катастрофічні наслідки для компаній.

У останні роки інформаційні технології відіграють все більшу роль в нашому житті, стаючи невід'ємною складовою повсякденної діяльності і розвитку. Цифрова трансформація впливає на всі аспекти нашого життя, від особистого спілкування і розваг до роботи та управління ресурсами компаній. Такий стрімкий розвиток IT став можливим завдяки неперервним інноваціям, що відбуваються в галузі, а також різноманітним можливостям, що відкриваються перед нами.

Однак, разом з тим, як інформаційні технології все більше переплітаються з нашим життям, збільшуються й потенційні ризики, пов'язані з ними. Це стосується, зокрема, зберігання та обробки персональних даних, безпеки корпоративних мереж і захисту від злому. З цієї причини компаніям дедалі важливіше стає розуміти, як забезпечити високий рівень безпеки та контролю над своїми інформаційними ресурсами.

У цьому контексті аналіз лог-файлів відіграє важливу роль у виявленні та відстеженні потенційних загроз безпеки. Стек ELK (Elasticsearch, Logstash, Kibana) є сучасним та потужним інструментом для аналізу та моніторингу лог-файлів, що дозволяє фахівцям з кібербезпеки швидко виявляти аномалії та забезпечувати захист інформаційних систем.

ELK стек, як згадувалось раніше, є одним з інструментів, що допомагає компаніям ефективно вирішувати проблеми безпеки. Але важливо зазначити, що успішна реалізація стратегії безпеки залежить не тільки від використання відповідних технічних рішень, а й від розуміння корпоративною культурою

значення інформаційної безпеки та відповідального підходу до управління даними.

Критично важливим бізнес-активом для повсякденної діяльності будь-якої компанії та її виживання є конфіденційна інформація про продукти, процеси, клієнтів та постачальників. З розвитком інформаційних технологій та збільшенням кількості даних, що обробляються, аналіз лог-файлів з використанням стеку ELK для виявлення потенційних загроз безпеці стає все більш актуальним.

Найбільш поширеною загрозою в мережній системі є несанкціонований доступ до інформаційних та обчислювальних ресурсів компанії. Це може призвести до втрати конфіденційності, цілісності та доступності інформації, яка є технологічним активом. Відповідно, аналіз лог-файлів та виявлення аномалій у поведінці системи може допомогти вчасно виявити потенційні загрози та забезпечити кібербезпеку організації.

Несанкціонований доступ до даних через компрометування комп’ютерної безпеки також відомий як злом. В ідеалі будь-яка організація повинна мати якийсь план реагування на інциденти для боротьби зі зломами локальної мережі, але дослідження показують, що цьому моменту приділяється мало уваги. Застосування аналізу лог-файлів з використанням ELK може допомогти в розробці такого плану та його впровадженні на практиці.

Метою цієї дипломної роботи є дослідження можливостей використання стеку ELK для аналізу лог-файлів з метою виявлення потенційних загроз в безпеці. Для досягнення цієї мети, будуть вивчені основні принципи роботи стеку ELK, його архітектура та можливості в контексті аналізу лог-файлів. Також будуть розглянуті методи застосування стеку ELK для виявлення аномалій та атак на інформаційні системи.

Отже, підсумовуючи, можна стверджувати, що тема "Аналіз лог-файлів з використанням ELK для виявлення потенційних загроз безпеці" є дуже актуальною в наш час, коли велика кількість інформації зберігається та

обробляється в електронному вигляді. Виявлення потенційних загроз безпеці та запобігання їх реалізації є критично важливим завданням, особливо у сфері інформаційної безпеки. Використання системи ELK для аналізу лог-файлів дозволяє швидко та ефективно виявляти потенційні загрози безпеці та приймати необхідні заходи для їх запобігання. Дослідження теми "Аналіз лог-файлів з використанням ELK для виявлення потенційних загроз безпеці" є важливим для розробки та вдосконалення систем інформаційної безпеки та може бути корисним для багатьох галузей, де забезпечення безпеки даних є критично важливим завданням.

1 ТЕОРЕТИЧНІ АСПЕКТИ АНАЛІЗУ ЛОГ-ФАЙЛІВ ТА СИСТЕМИ ELK

1.1 Огляд лог-файлів та їх значення в інформаційній безпеці

Лог-файли - це файли, які містять записи про події, які відбуваються в операційній системі або програмному забезпеченні. Вони зберігають інформацію про те, як користувачі взаємодіють з системою, що відбувається під час виконання програм, які процеси запущені та інші дії, які відбуваються в системі.

Лог-файли можуть мати різні формати та розміри, залежно від того, яка система їх генерує. Вони можуть бути текстовими, бінарними або використовувати спеціальні формати для зберігання даних. Лог-файли є важливим інструментом для аналізу безпеки та виявлення проблем в системі. Вони можуть допомогти виявити помилки, відмови, зловживання доступом або зловмисну діяльність. Аналіз лог-файлів дозволяє виявляти та вирішувати проблеми безпеки, що виникають в системі, а також розгортати проактивні заходи безпеки, щоб запобігти майбутнім інцидентам.

Огляд лог-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 лог-файлів може бути трудомістким і складним завданням, оскільки лог-файли можуть бути великими та недоступними для ручного аналізу. У таких випадках використання системи збору та аналізу лог-файлів, такої як ELK, дозволяє автоматизувати цей процес та зробити його більш ефективним.

1.1.1 Види лог-файлів

Лог-файли є важливим джерелом інформації про роботу систем та додатків. Вони зберігають різноманітну інформацію про роботу системи, таку як повідомлення про помилки, інформацію про події, що сталися, і інші дані, які можуть бути важливими для забезпечення безпеки системи.

Найпоширеніші види лог-файлів, які зустрічаються у більшості операційних систем і додатків, такі:

– системні лог-файли: містять інформацію про події, що сталися на рівні операційної системи, такі як старт/стоп служб, помилки ядра тощо;

– лог-файли додатків: містять інформацію про події, що сталися у програмах та додатках. Наприклад, лог-файли баз даних містять інформацію про запити до бази даних, а лог-файли веб-серверів містять інформацію про запити до веб-сайту;

– лог-файли мережевої активності: містять інформацію про мережеву активність, таку як інформацію про з'єднання, трафік, аутентифікацію, авторизацію тощо.

Крім того, існують різні спеціалізовані лог-файли, які зберігають інформацію про певні типи подій, такі як лог-файли антивірусного програмного забезпечення, які містять інформацію про виявлені загрози безпеки.

Для кожного типу лог-файлів існують спеціальні інструменти для їхнього збору, обробки та аналізу. ELK став дуже популярним інструментом для збору, обробки та аналізу лог-файлів, особливо в контексті забезпечення безпеки інформаці

1.1.2 Особливості зберігання та аналізу лог-файлів

Лог-файли можуть містити велику кількість інформації, що може бути корисною для виявлення загроз безпеці та дослідження подій, що сталися в системі. Однак, зберігання та аналіз великої кількості лог-файлів може бути складним завданням через обмеженість ресурсів та складність обробки

великої кількості даних. Нижче розглянемо деякі особливості зберігання та аналізу лог-файлів.

Зберігання лог-файлів

Для зберігання лог-файлів можна використовувати різні методи. Один з найпоширеніших методів - зберігання лог-файлів на локальному диску системи. Однак, цей метод може бути обмеженим через обмеженість розміру диску та неефективність резервного копіювання даних.

Інші методи зберігання лог-файлів, такі як зберігання на віддаленому сервері, можуть забезпечити більшу надійність та безпеку даних. Для цього можна використовувати системи зберігання даних, такі як Elasticsearch, який дозволяє зберігати та обробляти великі обсяги даних.

Аналіз лог-файлів

Аналіз лог-файлів може бути складним завданням через велику кількість даних, які необхідно обробляти. Щоб допомогти у цьому завданні, можна використовувати інструменти, такі як ELK, які дозволяють збирати та обробляти лог-файли, а також створювати зручні звіти та візуалізації даних.

При аналізі лог-файлів важливо визначити, яку інформацію необхідно відслідковувати та які дії можуть викликати потенційну загрозу для системи. Також важливо забезпечити регулярне оновлення програмного забезпечення та забезпечити відповідну реакцію на потенційні загрози безпеки.

Для ефективного аналізу лог-файлів також можна використовувати певні методики та алгоритми. Наприклад, методика структурного аналізу даних може допомогти відокремити корисну інформацію від зайвої та виявити можливі загрози безпеки. Також можна використовувати алгоритми машинного навчання для виявлення аномальної активності в системі та прогнозування потенційних загроз безпеці.

Отже, зберігання та аналіз лог-файлів є важливим аспектом забезпечення інформаційної безпеки. Використання спеціальних інструментів, таких як ELK, дозволяє збирати, зберігати та аналізувати великі обсяги даних

з метою виявлення потенційних загроз безпеці та підвищення ефективності реагування на них.

1.2 Відмінності логів у Windows та Linux

Операційні системи Windows та Linux використовують різні методики для журналювання подій, які стаються в системі, і це призводить до відмінностей у їхніх лог-файлах.

В системі Windows цей процес контролюється службою журналу подій, яка реєструє важливі події на рівні системи і застосунків. Журнали подій Windows зберігаються в бінарному форматі, що забезпечує більш безпечне та структуроване зберігання даних. Ці журнали можна поділити на три основні типи: системний, застосунок і безпека. Кожен із цих журналів реєструє певний тип подій, наприклад, події безпеки, пов'язані із входом та виходом користувачів, змінами політики безпеки тощо.

З іншого боку, Linux використовує більш розподілену систему журналювання, залежно від різноманітних системних процесів та демонів. Більшість подій в системі Linux реєструється в текстовому форматі, що спрощує аналіз та пошук. Централізоване журналювання в Linux здійснюється за допомогою демона syslog або, в більш сучасних системах, systemd-journald.

Виходячи з цих особливостей, обробка та аналіз лог-файлів з Windows і Linux можуть вимагати різних підходів. Однак, інструменти аналізу логів, такі як ELK Stack, допомагають уніфікувати цей процес, надаючи засоби для централізованого збирання, обробки та аналізу логів незалежно від того, з якої операційної системи вони походять. Це значно полегшує процес виявлення потенційних загроз безпеки в міжплатформеному середовищі.

1.2.1 Система журналюваня подій Windows

Система журналювання подій Windows, відома як Event Log Service (див. рис. 1.2), є вбудованою службою операційної системи, яка відповідає за зберігання подій, пов'язаних зі системою і застосунками. Ця служба використовується як системою, так і застосунками для ведення записів про важливі події та помилки, що виникають під час роботи системи.

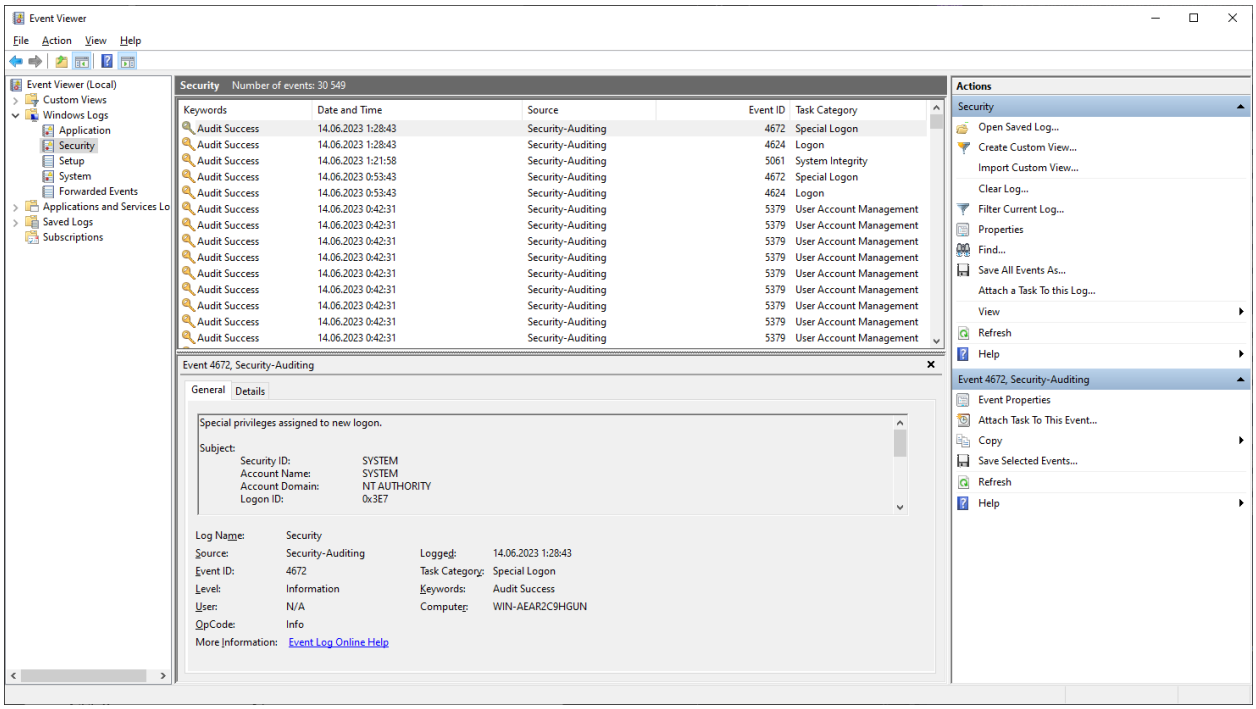


Рисунок 1.2 - Система журналювання подій Windows 10

Event Log Service реєструє події в різних журналах, у тому числі в системному журналі, журналі застосунків та журналі безпеки. Системний журнал включає події, пов'язані з операційною системою, такі як відключення, помилки драйверів або інші системні помилки. Журнал застосунків використовується застосунками, що встановлені в системі, для реєстрації важливих подій та помилок. Журнал безпеки є місцем для запису подій, пов'язаних з безпекою, таких як входи користувачів, зміни політики безпеки та інше. Кожна подія, яка реєструється в Event Log Service, має визначену структуру, що включає інформацію про джерело події, її тип, час виникнення

та іншу важливу інформацію. Всі ці дані зберігаються в бінарному форматі і можуть бути переглянуті за допомогою вбудованого додатка Windows Event Viewer.

Система журналювання подій Windows є незамінною для моніторингу стану системи і виявлення потенційних проблем або загроз. За допомогою інструментів аналізу логів, таких як ELK, можна автоматизувати процес збору, обробки та аналізу цих подій для раннього виявлення та відгуку на потенційні загрози безпеки.

1.2.2 Система журналюваня подій Linux

Система журналювання подій Linux є відкритою та гнучкою системою, яка забезпечує запис важливих системних подій. Заснована на декількох взаємопов'язаних компонентах, основний з яких - це служба syslog. Syslog є стандартним протоколом для ведення журналів, який використовується у більшості систем Linux і Unix.

Служба syslog працює на фоні і слухає системні повідомлення від різних джерел, включаючи ядро системи, демони та інші застосунки. Ці повідомлення класифікуються за їх важливістю (рівнем критичності) та джерелом (фасилітетом). Syslog може бути налаштований для маршрутизації цих повідомлень до різних місць залежно від цих класифікацій, включаючи різні файли журналів, електронну пошту, консоль або навіть віддалені сервери. Linux також використовує службу journald, частина системного демона systemd, яка реєструє події, пов'язані зі системою та застосунками, в бінарному форматі. Journald надає додаткові можливості порівняно з syslog, включаючи зберігання більше метаданих про події, зберігання журналів у бінарному форматі для кращого стиснення та перформансу, а також контроль доступу до журналів на основі політик безпеки. Аналіз лог-файлів в Linux є важливим інструментом для моніторингу стану системи та виявлення можливих проблем або загроз. За допомогою інструментів аналізу логів, таких

як ELK, можна автоматизувати процес збору, обробки та аналізу цих подій для раннього виявлення та реагування на потенційні загрози безпеки.

Система журналювання подій Linux є доволі відкритою і гнучкою, в основі якої лежать служби, як syslog та journald. Ці служби відповідають за збір та обробку системних повідомлень від різних джерел, включаючи ядро системи, демони та інші застосунки.

Файли журналів Linux зберігаються в каталозі /var/log, де розташовані журнали для всіх ключових компонентів системи, включаючи ядро, менеджери пакетів, процеси завантаження, Xorg, Apache, MySQL та інші. Важливість конкретного файлу журналу залежить від того, які аспекти системи ви досліджуєте. Один з найбільш критичних файлів журналів - це syslog, де реєструються майже всі системні події, за винятком повідомлень, пов'язаних з автентифікацією. Ви можете переглянути файли журналів за допомогою стандартних команд Linux, таких як ls, vi, tail, або dmesg. Наприклад, команда tail дозволяє вам переглядати останні рядки в файлі журналу, що може бути особливо корисним для відстеження активних проблем.

За допомогою ELK, ви можете зібрати, агрегувати та аналізувати ці логи на масштабах всієї системи, що значно полегшує процес знаходження, усунення несправностей та виявлення потенційних загроз безпеки.

1.3 Принципи збору та аналізу лог-файлів з використанням ELK

ELK є системою збору та аналізу лог-файлів, яка використовує Elasticsearch, Logstash та Kibana. На рисунку 1.3 наведена архітектура системи ELK. Вона включає лог-файли, які потрібно проаналізувати. Для збору лог-файлів і даних про події використовується Logstash, який також здійснює аналіз і трансформацію цих даних. Перетворені дані з Logstash зберігаються в ElasticSearch, де вони індексуються та стають доступними для пошуку. Kibana

використовує базу даних ElasticSearch для дослідження, візуалізації та обміну даними (див. рис 1.3).

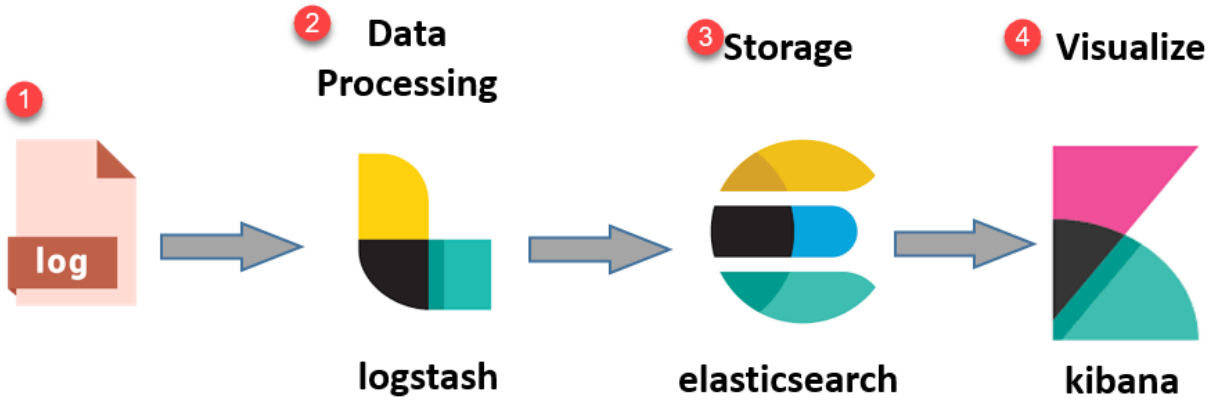


Рисунок 1.3 - Діаграма архітектури ELK Stack

ELK дозволяє збирати та аналізувати дані з різних джерел, таких як лог-файли, системні події, метрики, потокові дані та інші. Використання ELK для аналізу лог-файлів дозволяє зробити цей процес більш ефективним та забезпечити безпеку системи.

Важливість аналізу лог-файлів зростає зі зростанням обсягу даних, які генеруються системами. За допомогою аналізу лог-файлів можна виявити несправності та інциденти безпеки, які можуть негативно вплинути на діяльність організації. Крім того, згідно з деякими стандартами безпеки, такими як PCI DSS, HIPAA та інші, аналіз лог-файлів є обов'язковим елементом відповідності.

Окрім того, використання ELK дозволяє зменшити час, необхідний для аналізу та виявлення проблем безпеки в системі. ELK забезпечує автоматизацію процесу збору та аналізу даних, що дозволяє зосередитися на основних проблемах безпеки та вжити вчасних заходів для їх вирішення.

1.3.1 Elasticsearch

Elasticsearch є розподіленою системою зберігання та пошуку даних, яка була створена на основі Apache Lucene, інструменту для повнотекстового пошуку та аналізу даних. Elasticsearch забезпечує швидкий та ефективний пошук даних, що робить його популярним інструментом для збору та аналізу даних великих обсягів.

Основою Elasticsearch є так званий інвертований індекс, який дозволяє швидко знаходити документи, що містять задані слова. Elasticsearch дозволяє ефективно зберігати дані, індексувати їх та швидко знаходити необхідну інформацію. Для забезпечення високої швидкості та доступності Elasticsearch використовує розподілену архітектуру, яка дозволяє розподіляти дані та операції на кілька вузлів.

Elasticsearch також має вбудовані механізми для автоматичної реплікації та шарування даних, що дозволяє зберігати дані з високою надійністю та забезпечувати їх доступність. Крім того, Elasticsearch має велику кількість інтегрованих інструментів для аналізу та візуалізації даних, що робить його потужним інструментом для аналізу даних у сфері безпеки та моніторингу.

1.3.2 Logstash

Logstash є інструментом для збору та обробки даних, який дозволяє збирати дані з різних джерел, включаючи лог-файли, та перетворювати їх в однорідний формат для подальшого аналізу. Logstash дозволяє збирати дані у реальному часі, оброблювати їх та надсилати до Elasticsearch для зберігання та аналізу.

Logstash забезпечує можливість фільтрації та обробки даних, що дозволяє вибирати лише необхідну інформацію для аналізу. Це дозволяє знизити розмір зберіганих даних та забезпечити швидкий та ефективний аналіз даних.

Logstash має велику кількість вбудованих фільтрів та кодеків для обробки різноманітних типів даних, таких як лог-файли, JSON-файли, CSV-

файли та інші. Крім того, Logstash дозволяє розширювати свої можливості за допомогою плагінів, що дозволяє збирати та обробляти дані з різних джерел.

У цілому, Logstash є важливим інструментом для збору та обробки даних у режимі реального часу, що дозволяє ефективно аналізувати дані та виявляти потенційні проблеми безпеки. Використання Logstash разом з Elasticsearch та Kibana дозволяє створити потужну систему для аналізу даних та забезпечення безпеки системи.

1.3.3 Kibana

Kibana є інструментом для візуалізації та аналізу даних, який дозволяє створювати інтерактивні графіки, діаграми та звіти на основі даних, що зберігаються в Elasticsearch. Kibana забезпечує інтерактивний та інтуїтивно зрозумілий інтерфейс для аналізу даних, що дозволяє легко знайти необхідну інформацію та виявити потенційні проблеми безпеки.

Kibana дозволяє створювати різноманітні графіки та діаграми, такі як стовпчасті діаграми, кругові діаграми, лінійні графіки та багато інших. На рисунку 1.4 ви побачите панель навігації, яка містить вкладки та меню для переходу між різними розділами Kibana, такими як Дашборди, Візуалізації та Пошук. У розділі "Дашборди" ви можете створювати та переглядати свої власні дашборди, які складаються з різних візуалізацій та інших елементів, що відображають ваші дані у зручному форматі. В розділі "Візуалізації" ви можете створювати різні типи візуалізацій, такі як графіки, діаграми, карти та інші, для аналізу та візуалізації ваших даних. У розділі "Пошук" ви можете виконувати складні запити та фільтрувати дані для отримання специфічних результатів. Ви також можете налаштовувати різні параметри відображення, використовувати фільтри, збільшувати та зменшувати масштаб візуалізацій, а також зберігати та експортувати свої налаштування. Крім того, Kibana має велику кількість вбудованих фільтрів та панелей, що дозволяє швидко та ефективно аналізувати дані.

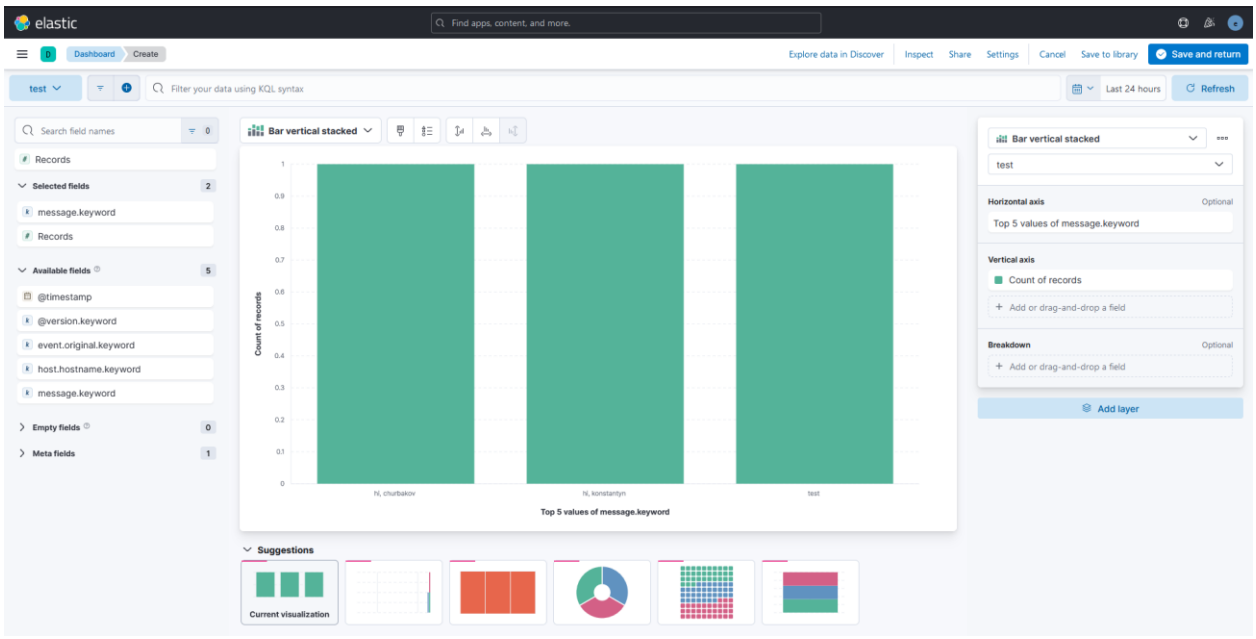


Рисунок 1.4 - Скріншот дашборду Kibana, з візуалізаціє даних з лог-файлів

Kibana також дозволяє створювати та налаштовувати сповіщення, що дозволяє оперативно реагувати на потенційні проблеми безпеки. Крім того, Kibana має вбудований механізм для імпорту та експорту дашбордів та звітів, що дозволяє легко обмінюватися інформацією між користувачами.

Загалом, Kibana є потужним інструментом для візуалізації та аналізу даних, який дозволяє легко виявляти потенційні проблеми безпеки та приймати ефективні рішення для забезпечення безпеки системи. Використання Kibana разом з Elasticsearch та Logstash дозволяє створити потужну систему для збору, обробки та аналізу даних у сфері безпеки.

2 АНАЛІЗ ЗАГРОЗ БЕЗПЕКИ ІНФОРМАЦІЇ

У цьому пункті роботи буде розглянуто аналіз загроз безпеці інформації в системі ELK. ELK є потужним інструментом для збору, аналізу та візуалізації даних, що може бути використано для виявлення різноманітних загроз безпеці інформації, таких як атаки на мережу, вторгнення в систему, фішингові атаки та інші.

Одним з переваг ELK є можливість збору лог-файлів з різних джерел, включаючи сервери, додатки та мережеві пристрої. Збір і аналіз цих лог-файлів дозволяє виявляти потенційні загрози безпеці раніше, ніж вони стають реальними проблемами.

Крім того, ELK забезпечує можливість реалізації цілком автоматизованих процесів збору та аналізу даних, які дозволяють швидко виявляти потенційні загрози безпеки та приймати рішення щодо їх подальшої обробки. Наприклад, за допомогою ELK можна налаштувати систему автоматичного виявлення несправностей та аномальних поведінок в системі, що може свідчити про потенційну загрозу безпеці.

Однак, при використанні ELK для виявлення загроз безпеці необхідно враховувати ряд технічних та організаційних аспектів, таких як правильна настройка системи, моніторинг та аналіз лог-файлів у режимі реального часу, оцінка серйозності виявлених загроз та прийняття вчасних заходів щодо їх вирішення.

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.

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. Крім того, можна застосовувати методи машинного навчання, щоб автоматично визначати аномальність поведінки користувачів та системи, яка може свідчити про потенційну загрозу безпеці.

Важливо пам'ятати, що виявлення загроз безпеки інформації - це тільки перший крок. Для ефективної захисту інформації важливо також приймати вчасні та ефективні заходи щодо їх усунення. ELK може допомогти забезпечити цей процес, надаючи зручні інструменти для візуалізації та аналізу даних, що дозволяють ефективніше виявляти загрози та приймати вчасні рішення.

Для аналізу загроз безпеки інформації в системі ELK також можна використовувати спеціальні інструменти, які дозволяють виявляти певні типи атак або відслідковувати конкретні події. Наприклад, можна налаштувати систему виявлення фішингу, яка буде відстежувати відправлення електронних листів, які містять підозрілий контент або незвичайні запити.

Для більш детального аналізу можна використовувати такі інструменти, як системи моніторингу трафіку мережі, які забезпечують поглиблений аналіз даних інтернет-трафіку та дозволяють виявляти зловмисну активність в режимі реального часу. Такі системи можуть надавати важливі дані для виявлення атак, які можуть пройти непоміченими в інших випадках.

Застосування системи ELK для аналізу лог-файлів є дуже ефективним засобом виявлення потенційних загроз безпеці інформації. Даний підхід дозволяє підвищити рівень захисту інформації та забезпечити вчасне виявлення можливих загроз.

2.1 Статистика та аналіз атак

У міру розгортання цифрових технологій, ми стикаємося зі зростанням кібератак. Навіть за оптимістичними оцінками, від кіберзлочинності світ майже відразу втратить 8 трильйонів доларів США в 2023 році, згідно з

даними Cybersecurity Ventures. Це ставить кіберзлочинність на третє місце за обсягом економіки, якщо б її можна було виміряти як окрему країну - вона знаходиться за США та Китаєм [16].

Крім того, відповідно до прогнозів, глобальні втрати від кіберзлочинності зростатимуть на 15% щороку протягом наступних трьох років, досягнувши 10,5 трильйонів доларів США щорічно до 2025 року, у порівнянні з 3 трильйонами доларів США в 2015 році.

Збитки від кібератак можуть приймати різні форми: пошкодження і втрата даних, крадіжка коштів, втрата продуктивності, порушення інтелектуальної власності, виток персональних і фінансових даних, шахрайство, негативний вплив на бізнес-процеси, судові розслідування, витрати на відновлення зламаних систем і даних, а також втрата репутації.

Офіційний звіт про кіберзлочинність за 2022 рік, опублікований Cybersecurity Ventures за підтримки eSentire, містить важливу інформацію про масштаби кіберзагроз, з якими ми стикаємося, а також статистичні дані та прогнози, які допомагають нам краще зрозуміти, як ми можемо впоратися з цими загрозами. Звіт також надає детальний огляд втрат від кіберзлочинності, які очікуються у 2023 році. Це дозволяє нам бачити витрати в контексті, що включає 8 трильйонів доларів США щорічно, 667 мільярдів доларів на місяць, 154 мільярди доларів на тиждень, 21,9 мільярда доларів на день, 913 мільйонів доларів на годину, 15,2 мільйона доларів за хвилину і 255 000 доларів в секунду [17].

Агентство Moody's вказує на особливо ризиковані галузі. Це включає критично важливу інфраструктуру, яка має дуже високий рівень кібер-ризику, включаючи електричні, газові, водопровідні компанії та лікарні. А також банки, телекомунікаційні компанії, технологічні компанії, хімічна промисловість, енергетична сфера та транспортні служби, які мають високий рівень кібер-ризику [18].

Практично кожен бізнес може стати жертвою кіберзлочинів. "Будь-яка компанія, яка хоче гарантувати безперебійність своєї роботи, захистити

репутацію та забезпечити безпеку даних своїх співробітників та клієнтів, повинна інвестувати в кібербезпеку і бути завжди на крок попереду", - зазначила Ерін Маклін, директор з маркетингу компанії eSentire [19].

Велика частина атак спрямована проти малого та середнього бізнесу (МСБ). На жаль, близько 60% таких компаній припиняють свою діяльність протягом шести місяців після того, як стають жертвами витоку даних або злому.

Програми-вимагачі

Спостерігається зростання кількості та вартості атак з використанням програм-вимагачів. В 2021 році прогнозувалась глобальна вартість таких атак на рівні 20 мільярдів доларів, що в десятки разів перевищує показник 2015 року - 325 мільйонів доларів. Очікується, що до 2031 року річні втрати від рансомвар сягнуть величезної суми в 265 мільярдів доларів [20]. Частота таких атак зростає, і, як очікується, до 2031 року вони будуть відбуватися кожні дві секунди. При цьому компанія CNA Financial з Чикаго уже встановила рекорд, виплативши 40 мільйонів доларів хакерам [21].

Криптозлочинність

Криптозлочинність, включаючи шахрайство та крадіжку криптовалюти, зростає. В 2025 році очікується, що світові втрати від таких злочинів сягнуть 30 мільярдів доларів, зростаючи на 15% щорічно. Це майже вдвічі більше, ніж 17,5 мільярдів доларів, втрачених у 2021 році. У 2022 році ФБР зареєструвало зростання шахрайства з криптовалютами з 907 мільйонів доларів у 2021 році до 2,57 мільярдів доларів [22]. Зловмисники використовують децентралізовані біржі та інші сервіси для виведення незаконних коштів, що сягає 4 мільярдів доларів. Значна кількість незаконних криптотранзакцій пов'язана з Росією, включаючи збір пожертв для російських військових. Загальний обсяг незаконних транзакцій у 2022 році сягнув рекордних 20,6 мільярда доларів.

Головні взломи:

– один з найбільш значних витоків даних, зафіксованих у 2023 році, належить компанії T-Mobile. У січні телекомунікаційний гігант розкрив

крадіжку персональної інформації, що належить 37 мільйонам поточних постоплатних та передплачених облікових записів клієнтів, яка стала можливою завдяки використанню API. Друге порушення сталося в травні;

– у лютому 2023 року Cloudflare виявив і пом'якшив наслідки найбільшої розподіленої атаки на відмову в обслуговуванні (DDoS), яка коли-небудь була зафіксована. DDoS-атака потужністю 71 мільйон запитів на секунду (rps), яку назвали "гіпер-об'ємною", на 54 відсотки перевищила попередню атаку потужністю 46 мільйонів rps, що відбулася в червні 2022 року;

– одного з найбільших витоків даних в історії зазнала компанія Yahoo. Інцидент з безпекою, що стався у 2013 році, вплинув на всі три мільярди облікових записів користувачів компанії. Лише за три місяці до розкриття інформації у 2015 році технологічний гігант виявив окремий витік, який вплинув на щонайменше 500 мільйонів акаунтів.

Загальна картина, що складається зі згаданих вище даних, показує тривожне зростання кіберзлочинності. Рансомвари та кріптозлочинність продовжують ставати все більш розповсюдженими та руйнівними, приносячи шкоду на мільярди доларів. Ці атаки впливають на бізнеси, уряди, споживачів і навіть цілі країни, а їх частота продовжує зростати.

Зазначена статистика відображає нагальну потребу в глибокому та всебічному підході до кібербезпеки. Створення надійних, витривалих інформаційних систем, ефективне розслідування та притягнення до відповідальності злочинців, а також посилення міжнародного співробітництва у цій сфері є ключовими аспектами боротьби з цим глобальним викликом.

Важливо врахувати, що ці дані - лише вершина айсберга. Багато атак не реєструються або не виявляються, а незаконні транзакції часто приховані або змасковані. Отже, реальна шкода та розмах кіберзлочинності можуть бути значно вищими, ніж це відображено в статистиці.

2.2 Класифікація загроз безпеки інформації

Загрози безпеці інформації - це події, які можуть призвести до порушення цілісності, конфіденційності та доступності інформації. Для ефективного аналізу загроз безпеці, їх можна класифікувати за різними критеріями.

Один з можливих способів класифікації загроз безпеці інформації - це відповідно до виду атаки, яку вони викликають. Атаки можуть бути активними та пасивними.

Активні атаки - це атаки, при яких зловмисники активно взаємодіють з цільовою системою з метою її порушення або отримання доступу до конфіденційної інформації. Ці атаки можуть включати напади на мережевий протокол, експлойти вразливостей програмного забезпечення, злам паролів та багато іншого. Активні атаки можуть мати на меті викрадення, внесення змін або знищення даних, завдання шкоди роботі системи або злам безпеки мережі.

Пасивні атаки - це атаки, при яких зловмисники не взаємодіють з системою, а лише збирають інформацію, що передається по мережі. Такі атаки можуть включати перехоплення мережевого трафіку, аналіз вмісту пакетів, викрадення файлів локальних систем та інші. Ці атаки можуть бути складнішими у виявленні, оскільки зловмисники не змінюють стан системи, а лише збирають інформацію.

Оскільки активні та пасивні атаки мають різні характеристики та можуть використовувати різні методи, виявлення цих атак вимагає використання різних методик та інструментів. Важливо також розуміти, що залежно від цілей та мети зловмисника, можуть використовуватись як активні, так і пасивні атаки.

2.2.1 Активні атаки та їх виявлення за допомогою ELK

Активні атаки є більш складними в реалізації порівняно з пасивними атаками, але водночас можуть бути більш ефективними. Вони спрямовані на зміну даних або заборону доступу до них.

У залежності від цілей, активні атаки можуть бути поділені на наступні типи:

– атаки на цілісність даних - ці атаки спрямовані на зміну даних, таких як вставка нових даних, видалення чи зміна існуючих даних, або навіть повне видалення даних. Ці атаки можуть бути небезпечними, оскільки вони можуть призвести до порушення цілісності даних, які є важливими для роботи системи або її користувачів. Щоб спостерігати ці атаки в ELK використовуєтсья компонент контролю цілісності файлів (File Integrity Monitoring, FIM). Він генерує сповіщення, коли він виявляє зміну в файловій системі. Метадані включають контрольні суми MD5, SHA1 та SHA256, розміри файлів (до та після зміни), права на файли, власника файлу, зміни вмісту та користувача, який зробив ці зміни (who-data). На рисунках 2.1 - 2.2, у розділі контролю цілісності інформації в панелі керування користувачі можуть бачити всі деталі спрацьованих сповіщень та знайти повний звіт про виявлені зміни (див. рис. 2.1.1 - 2.1.2). Модуль FIM проводить періодичні скани на конкретних шляхах та контролює конкретні каталоги на предмет змін в реальному часі. Ви можете встановити, які шляхи слід контролювати, в налаштуваннях агентів та менеджера. FIM зберігає контрольні суми файлів та інші атрибути в локальній базі даних FIM. Під час сканування агент повідомляє про будь-які зміни, які модуль FIM знаходить на контрольованих шляхах. Модуль FIM шукає зміни файлів, порівнюючи контрольні суми файлу зі збереженими контрольними сумами та значеннями атрибутів. Він генерує сповіщення, якщо знаходить розбіжності;

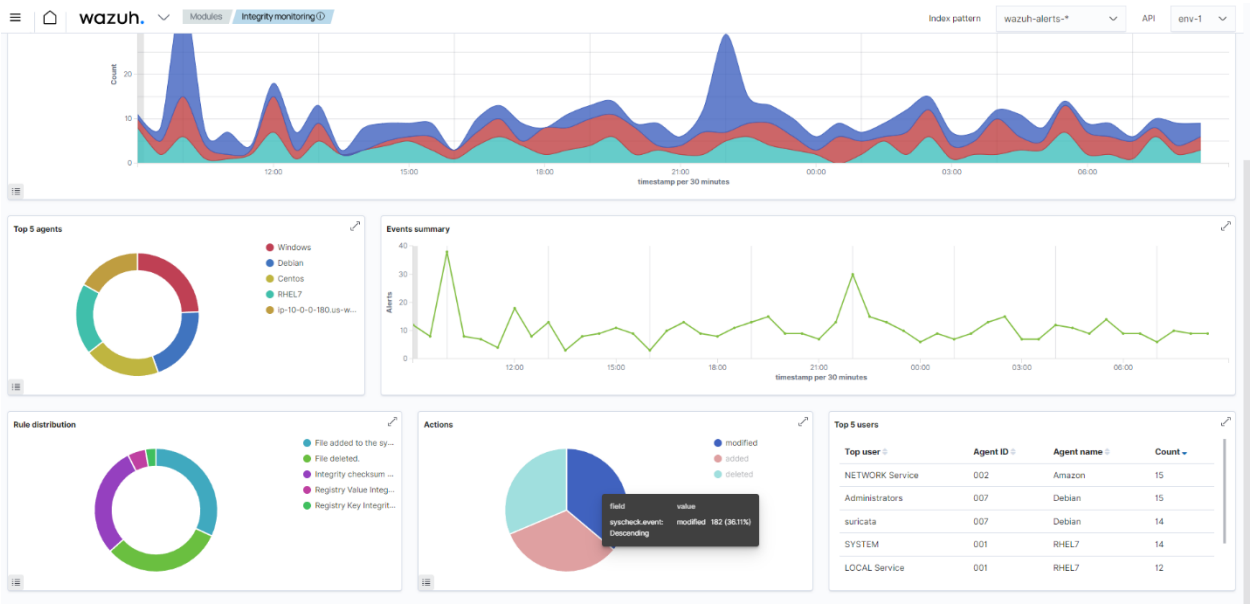


Рисунок 2.1 - Розділ інформаційної панелі, де користувачі можуть переглядати деталі сповіщень у вигляді графіків

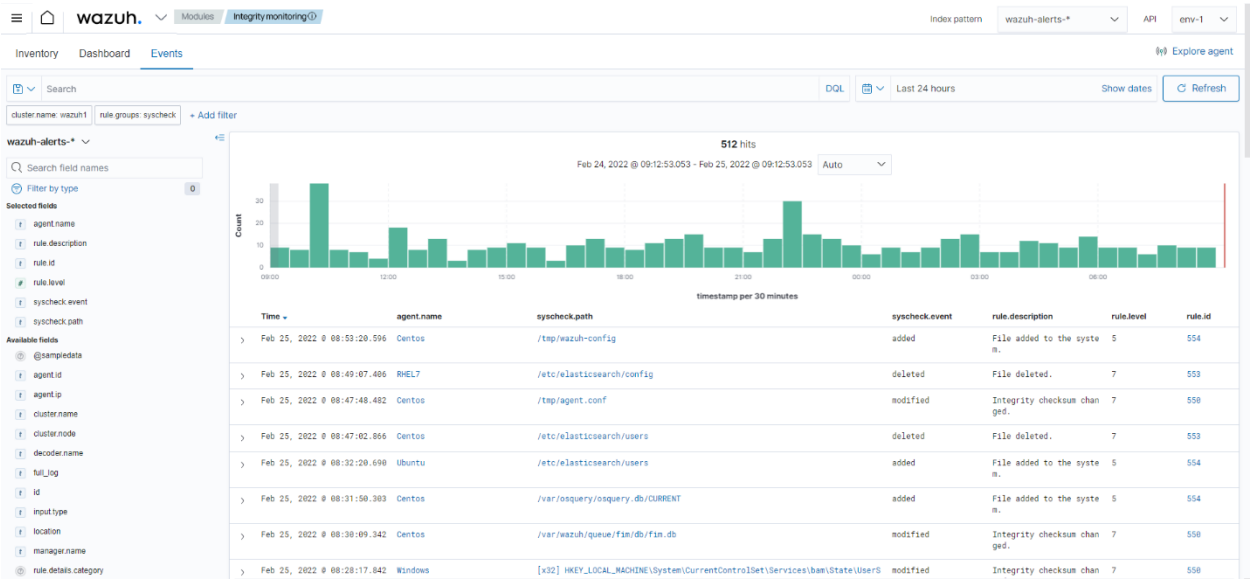
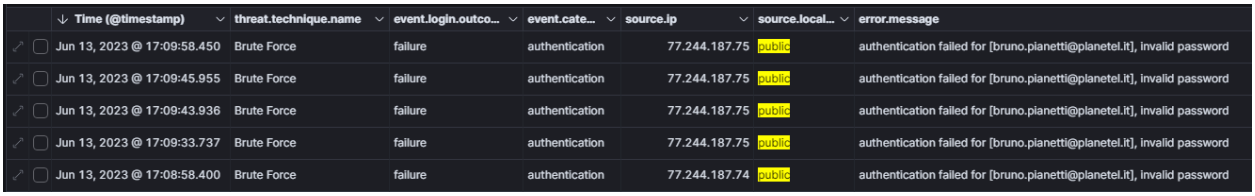


Рисунок 2.2 - Розділ інформаційної панелі, де користувачі можуть переглядати деталі сповіщень у вигляді окремих випадків

– атаки на доступність даних - ці атаки спрямовані на блокування або переривання доступу до інформації. Ці атаки можуть призвести до перерв у роботі системи або її окремих компонентів та відновлення роботи може бути досить складним та часом затримати роботу системи;

– атаки на аутентифікацію та авторизацію. Ці атаки спрямовані на отримання нелегітимного доступу до системи. Наприклад, зловмисник може використовувати методи підбору паролів або використовувати вразливості системи для зламу паролів. На рисунку 2.3, за допомогою ELK, ми можемо спостерігати несподівно та підозрілі спроби входу, наприклад, що це за техніка взлому, результат входу, категорія події, IP-адреса джерела, місцезнаходження джерела, повдіомлення про помилку та інші (див. рис 2.3);



Time (@timestamp)	threat.technique.name	event.login.outco...	event.cate...	source.ip	source.local...	error.message
Jun 13, 2023 @ 17:09:58.450	Brute Force	failure	authentication	77.244.187.75	public	authentication failed for [bruno.pianetti@planetel.it], invalid password
Jun 13, 2023 @ 17:09:45.955	Brute Force	failure	authentication	77.244.187.75	public	authentication failed for [bruno.pianetti@planetel.it], invalid password
Jun 13, 2023 @ 17:09:43.936	Brute Force	failure	authentication	77.244.187.75	public	authentication failed for [bruno.pianetti@planetel.it], invalid password
Jun 13, 2023 @ 17:09:33.737	Brute Force	failure	authentication	77.244.187.75	public	authentication failed for [bruno.pianetti@planetel.it], invalid password
Jun 13, 2023 @ 17:08:58.400	Brute Force	failure	authentication	77.244.187.74	public	authentication failed for [bruno.pianetti@planetel.it], invalid password

Рисунок 2.3 - Приклад зображення атаки на автентифікацію у ELK

– атаки на програмне забезпечення. Ці атаки спрямовані на злам програмного забезпечення з метою завдання шкоди системі. Наприклад, зловмисник може використовувати віруси, черв'яки, троянські програми або інші види шкідливого програмного забезпечення. Продукти безпеки можуть ідентифікувати шкідливе ПЗ, перевіряючи сигнатури відомого шкідливого програмного забезпечення. Інструменти безпеки також можуть виявити зловмисну активність, виявляючи підозрілу поведінку від програмної активності. Коли шкідливий код інфікує систему, він може змінити її, використовуючи різні техніки для уникнення виявлення. Модуль контролю цілісності файлів використовує багатоаспектний підхід для протидії цим технікам з метою виявлення зловмисних файлів і аномальних шаблонів, які свідчать про наявність шкідливого коду. FIM допомагає виявляти зловмисні файли на контрольованих кінцевих точках. Сам по собі модуль FIM не може виявити зловмисні файли. Однак, ви можете виявити шкідливе ПЗ, поєднуючи модуль FIM з правилами виявлення загроз і джерелами інформації про загрози. Ви можете налаштувати використання подій FIM з джерелами інформації про

загрози, такими як VirusTotal та CDB-списки, що містять хеші файлів, та скани YARA для виявлення шкідливого ПЗ;

– атаки на мережевий рівень. Ці атаки спрямовані на порушення роботи мережі з метою завдання шкоди системі. Наприклад, зловмисник може використовувати DDoS-атаки для перевантаження мережевих ресурсів або використовувати віруси, що заважають нормальній роботі мережі. Моніторинг запитів до веб-серверів є основним методом, який використовується в даний час, і в цьому контексті здатність аналізувати та візуалізувати доступ до журналів помилок у серверному аналізі журналів є ключовою для швидкої ідентифікації. Незалежно від того, Apache, IIS чи NGINX - ці журнали доступу містять багато інформації про запити, що надходять до ваших серверів, такі як помилки відповідей та вихідні IP-адреси, геолокації та пристрої. За допомогою Kibana ви можете побудувати серію візуалізацій для моніторингу ознак, які можуть вказувати на те, що відбувається атака. Моніторинг "503" відповідей від вашого серверу через повільні часи відгуку - це занадто пізно, оскільки коди відповіді вказують на те, що ваш сервіс вже не працює. Однак атаки засипають сервер неправильними запитами, тому моніторинг спалахів "404" може допомогти. На основі базової метрики ви могли б створити сповіщення, яке повідомило б вас, коли певна кількість "404" буде активована протягом певного періоду часу (див. рис. 2.4);

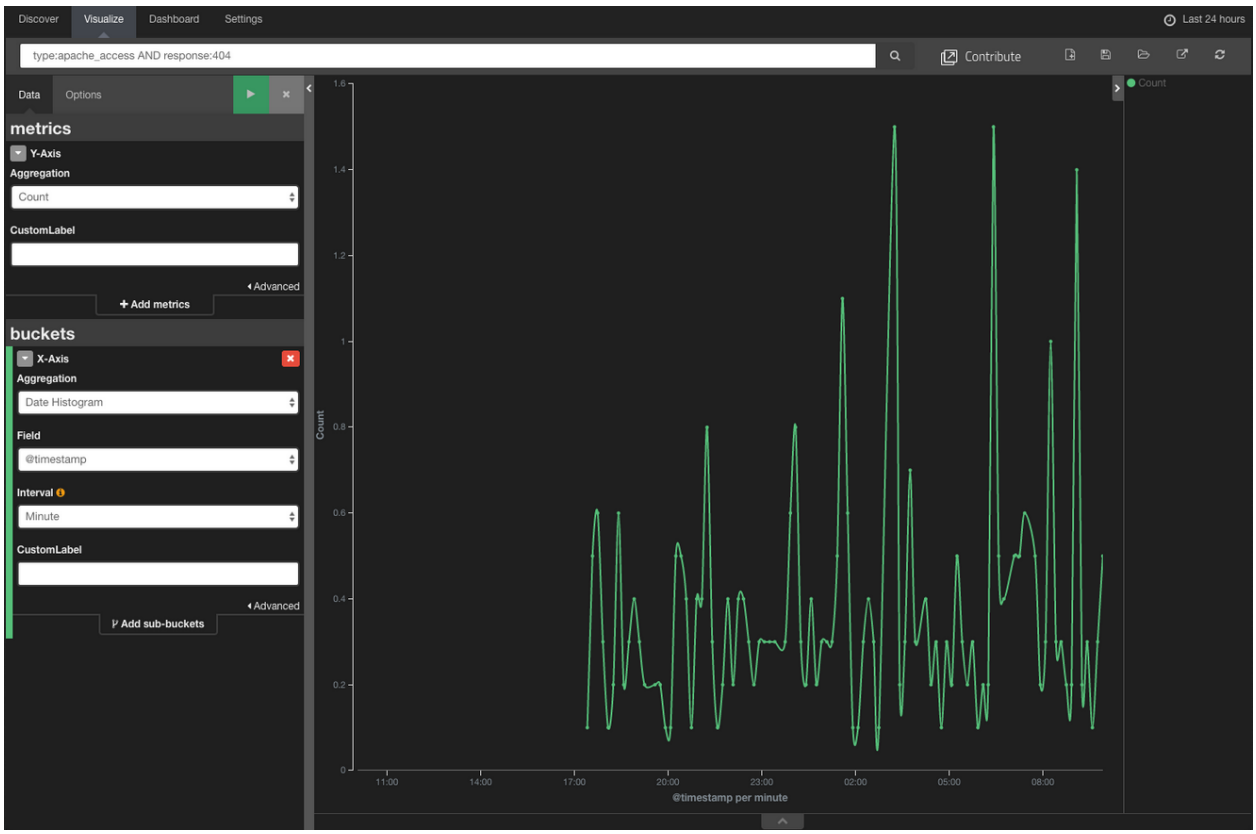


Рисунок 2.4 - Гістограма, яка відповідає відповіді «404» та типу «apache_access»

– атаки на людський фактор. Ці атаки спрямовані на використання людського фактору з метою завдання шкоди системі. Наприклад, зловмисник може використовувати соціально-інженерні методи для отримання доступу до конфіденційної інформації.

2.2.2 Пасивні атаки та їх виявлення за допомогою ELK

Пасивні атаки, хоча менш складні в реалізації, все ж можуть мати серйозні наслідки для інформаційної безпеки. Оскільки вони не змінюють інформацію, а лише перехоплюють її, то часто їх важко виявити.

У залежності від цілей, пасивні атаки можуть бути поділені на наступні типи:

– аналіз трафіку - це пасивна атака, в якій зловмисник моніторить і вивчає паттерни мережевого трафіку, який проходить до і від цільових систем.

Вони використовують статистичні методи для аналізу і тлумачення обміну комунікацій через мережу. Зазвичай це роблять навіть з зашифрованим мережевим трафіком, але ще більш поширеним є використання незашифрованого трафіку. Це створює ризик, оскільки за допомогою аналізу трафіку зловмисники можуть виявити важливу інформацію, таку як об'єм даних, частоту передачі, джерело і призначення комунікації, що може бути використано для подальшого планування атак. Elastic Stack є інструментом, що може допомогти у цьому процесі. Він використовує компонент, званий Beats, для збору даних мережевого трафіку з різних джерел. Наприклад, Packetbeat може перехоплювати та збирати інформацію про пакети, які передаються через мережу. Після збору даних, Elasticsearch може бути використаний для індексації та обробки цих даних. Він допомагає організувати та структурувати дані таким чином, що спрощує їх пошук та аналіз. Це може включати розбиття даних на окремі поля, виконання статистичних обчислень або візуалізацію даних за допомогою Kibana;

– перехоплення пакетів - ця техніка є іншою формою пасивної атаки, де зловмисник "слухає" або "перехоплює" пакети даних, що передаються через мережу. Це може включати в себе всі види даних, включаючи, але не обмежуючись, електронні повідомлення, дані веб-браузера, файлові передачі тощо. Перехоплення пакетів може надати зловмисникам доступ до конфіденційної інформації, такої як паролі, особиста інформація та деталі фінансових транзакцій. Це створює великі ризики безпеки, оскільки такі дані можуть бути використані зловмисниками для вчинення шахрайства, крадіжки ідентичності або навіть для більш складних цільових атак;

– збір інформації - ці атаки спрямовані на збір різних типів інформації з різних джерел, наприклад, баз даних або відкритих джерел. Збір інформації може відбуватися як в ручному режимі, так і за допомогою спеціального програмного забезпечення;

– соціальний інжиніринг - ці атаки спрямовані на отримання доступу до інформації шляхом маніпулювання людьми або використання

психологічних технік, щоб переконати людей розкрити конфіденційну інформацію. Хоча ELK Stack не може безпосередньо виявляти соціальний інжиніринг, він може бути весьма корисним у виявленні певних показників таких атак. Через аналіз логів і моніторинг активності користувачів ELK може виявляти несанкціонований доступ до системи, аналізувати патерни поведінки користувачів і виявляти підозрілі дії, які можуть вказувати на атаку соціального інжинірингу. Він може допомогти у виявленні зловмисних дій, таких як незвичайні спроби авторизації, шкідливий вміст у електронних листах або веб-запитах, а також незвичайні мережеві взаємодії. Завдяки цьому, ELK Stack стає важливим інструментом в протидії атакам соціального інжинірингу.

Для ефективного виявлення потенційних загроз безпеці необхідно проводити аналіз лог-файлів, що збираються системою моніторингу безпеки, та визначати надзвичайні події. При цьому потрібно звертати увагу на різноманітність атак та можливі шляхи їх реалізації, щоб мати можливість діяти вчасно та ефективно у разі виявлення загроз.

2.3 Практичні приклади використання аналізу лог-файлів для виявлення загроз

2.3.1 Застосування ElasticSearch та Kibana для моніторингу DDoS-атак через аналіг лог-файлів

Основні налаштування DDoS-атак ґрунтуються на двох ключових порогах, незалежно від вибраної стратегії (ручне керування, автоматичний режим, мультиплікатор і т.д.): виявлення та нейтралізація наслідків На рисунку 2.5 ми можемо побачити рівень виявлення та усунення атак (див. рис. 2.5).

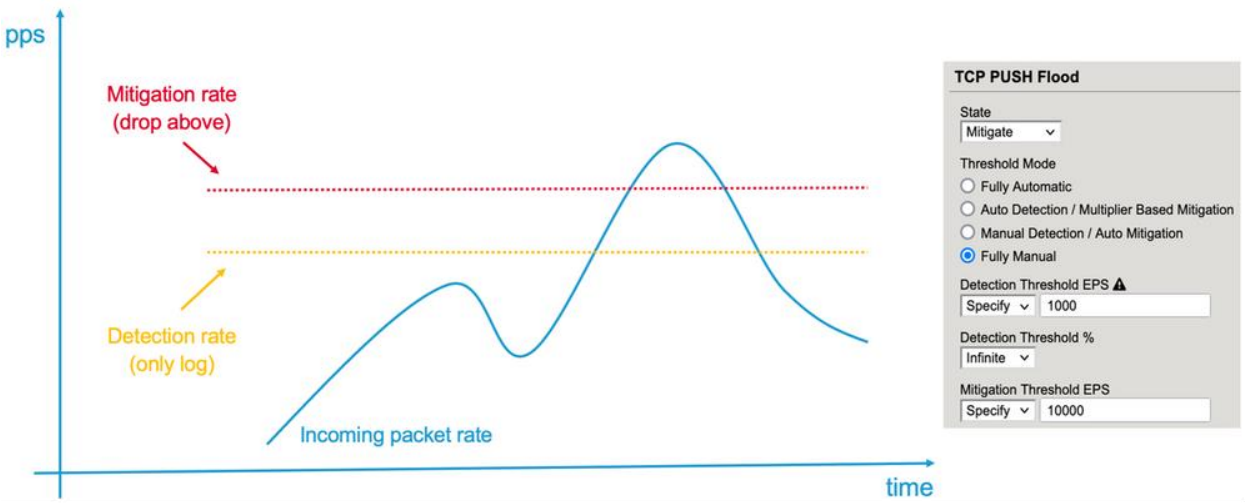


Рисунок 2.5 - Рівень виявлення та усунення атак

"Виявлення" відноситься до процесу сповіщення DDoS-оператора про перевищення розрахованої швидкості вхідного трафіку. Тут не відбувається блокування трафіку, але відправляється інформація в журнал. Порог виявлення визначається або обчислюється на основі "нормальної" швидкості. Все, що перевищує цей поріг, вважається "ненормальним" і, отже, підозрілим, але це не обов'язково означає атаку. Оператор DDoS повинен бути проінформований про цю подію.

Як приклад, розглянемо лог-повідомлення, яке генерується, коли швидкість передачі пакетів перевищує поріг виявлення (див. лістинг 2.1):

Лістинг 2.1 - Приклад повідомлення, яке відправляється, коли швидкість передачі пакетів перевищує поріг виявлення

```
Jun 16 23:08:46 172.30.107.11 action="Allow",hostname="lon-
i5800-
1.pme.itc.f5net.com",bigip_mgmt_ip="172.30.107.11",context_name=
"/Common/www_10_103_2_80_80",date_time="Jun 17 2021
22:58:12",dest_ip="10.103.2.80",dest_port="80",device_product="D
DoS Hybrid
Defender",device_vendor="F5",device_version="15.1.2.1.0.317.10",
dos_attack_event="Attack
Sampled",dos_attack_id="550542726",dos_attack_name="TCP Push
Flood",dos_packets_dropped="0",dos_packets_received="117",errdef
s_msgno="23003138",errdefs_msg_name="Network DoS
```

Продовження лістингу 2.1

```
Event",flow_id="0000000000000000",severity="4",dos_mode="Enforce
d",dos_src="Volumetric, Per-SrcIP, VS-specific attack,
metric:PPS",partition_name="Common",route_domain="0",source_ip="
10.103.6.10",source_port="39219",vlan="/Common/vlan3006_client"
```

Action = "Allow" означає, що BIG-IP не відкидає пакети (з позицій DoS), але надає оператору повідомлення про те, що за останню секунду захищений контекст (у цьому випадку: /Common/www_10_103_2_80_80) отримав 117 (dos_packets_received) push-пакетів (dos_attack_name) від IP-адреси 10.103.6.10 (source_ip).

Зазначимо, що це повідомлення журналу "Volumetric, Per-SrcIP, VS-specific attack" (dos_src) також повідомляє вам, що IP-адресу джерела визнано зловмисним актором. Отже, ця подія була спровокована налаштуваннями Bad Actor в рамках вектора TCP Push flood.

Якщо швидкість вхідних пакетів перевищує встановлений поріг пом'якшення для DoS-вектора або сигнатури атаки, BIG-IP починає відкидати або обмежувати швидкість трафіку, який перевищує це значення. Визначення такого значення позначає DDoS-атаку, оскільки на захищений контекст (сервер, службу, мережу, BIG-IP тощо) буде впливати велика кількість пакетів на секунду. В таких випадках пристрій BIG-IP DoS (AFM/DHD) має знизити кількість пакетів, які надходять в контекст, і починає відкидати пакети на відповідному векторі. Поріг пом'якшення може бути встановлений вручну або автоматично розрахований на основі історії або шляхом множення порогу виявлення.

Розглянемо приклад повідомлення в логах про відкинуті пакети (див. лістинг 2.2).

Лістинг 2.2 - Приклад повідомлення про відкинуті пакети

```
Jun 17 23:05:03 172.30.107.11 action="Drop",hostname="lon-
i5800-
```

Продовження лістингу 2.2

```
1.pme.itc.f5net.com",bigip_mgmt_ip="172.30.107.11",context_name=
"Device",date_time="Jun 17 2021 22:54:29",dest_ip="10.
103.2.80",dest_port="0",device_product="DDoS Hybrid
Defender",device_vendor="F5",device_version="15.1.2.1.0.317.
10",dos_attack_event="Attack
Sampled",dos_attack_id="3221546531",dos_attack_name="Bad TCP
flags (all
cleared)",dos_packets_dropped="152224",dos_packets_received="152
224",errdefs_msgno="23003138", errdefs_msg_name="Network DoS
Event",flow_id="0000000000000000",severity="4",dos_mode="Enforce
d",dos_src="Volumetric, Aggregated across all SrcIP's, Device-
Wide attack, metric:
PPS",partition_name="Common",route_domain="0",source_ip="10.
103.6.10",source_port="12826",vlan="/Common/vlan3006_client"
```

Це повідомлення показує, що в пристрій BIG-IP надійшла агрегована велика кількість пакетів з різних IP-адрес джерел (dos_src="Volumetric, Aggregated across all SrcIP's, Device-Wide attack") і ці пакети були відкинуті (dos_packets_dropped="152224") протягом останньої секунди. Отже, IP-адреса джерела (source_ip="10.103.6.10") - це лише один з джерел пакетів, що були відкинуті. На даний момент не виявлено "поганого актора". Це може статися, якщо не налаштовано функцію виявлення зловмисника, або якщо кожен пакет має іншу IP-адресу джерела.

Структура інформаційних панелей

Для візуалізації DDoS-атак на інформаційних панелях, ми адаптували дві ключові події реєстрації - дозвіл та падіння пакетів.

Оператор, який моніторить DDoS-атаки, повинен знати, коли в мережі виникає аномалія, які вектори атаки активовані цією аномалією, які цільові точки залучені, та які джерела викликають цю аномалію.

Але також важливо знати, коли мережа стала об'єктом атаки, які заходи були вжиті для зменшення її наслідків, в яких напрямках та з яких джерел. Наскільки велика була втрата пакетів і т.д.?

Коли ви відкриваєте "Панель моніторингу DDoS" та вибираєте "Оглядову панель", ви побачите, що вона розділена на дві частини (див. рис. 2.6). Зліва - інформація про моменти, коли пристрій DDoS відкидав пакети, справа - інформація про "підозрілі" пакети, тобто коли трафік перевищував поріг виявлення, але не був відкинутий (дія "Дозволити").

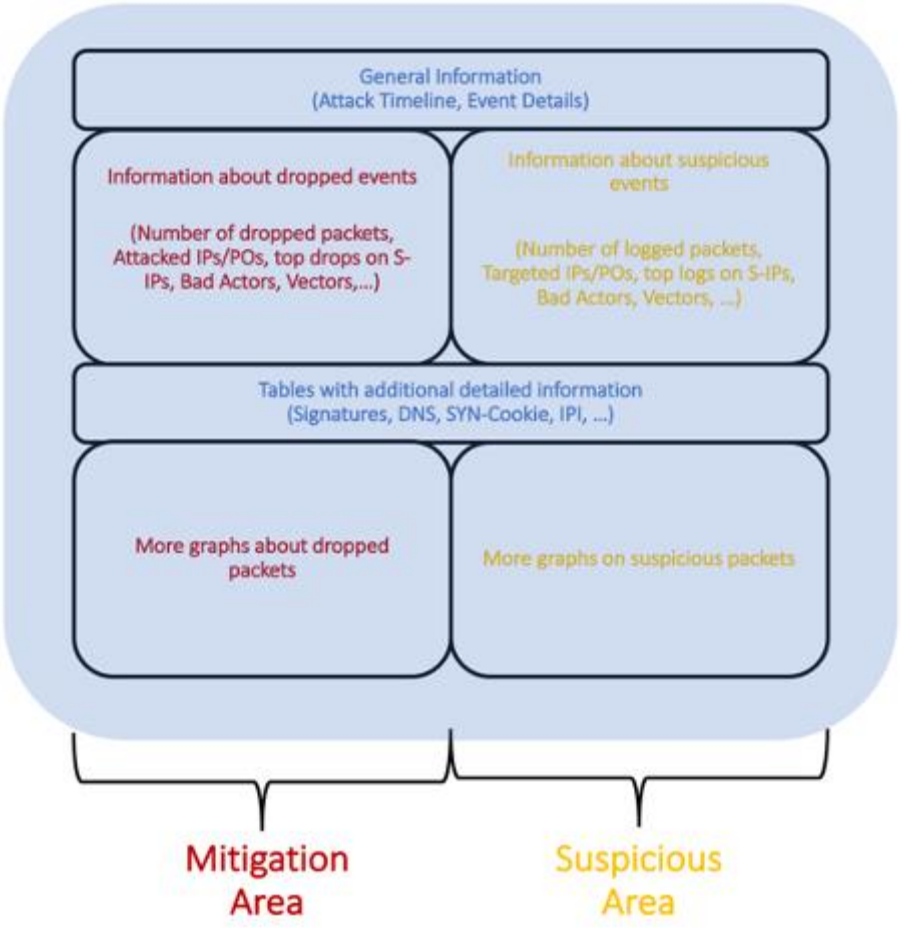


Рисунок 2.6 - Структура інформаційної панелі

На цій панелі ви також знайдете діаграми або таблиці, які не розділяють панель на дві частини. Це місце, де ви знайдете об'єднану інформацію з обох секторів/подій (запобігання та підозрілих).

Візуалізація DDoS-атак у Kibana: Гід по дашбордах

У розділі меню "Головна/Аналітика/Дашборд" ви знайдете всі дашборди (див. рис. 2.7 - 2.8).

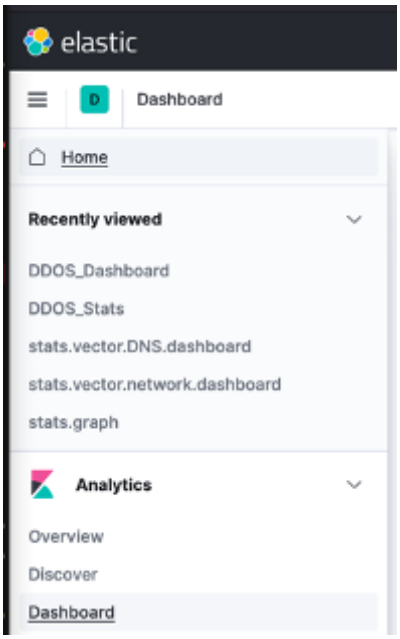


Рисунок 2.7 - Меню дашбордів

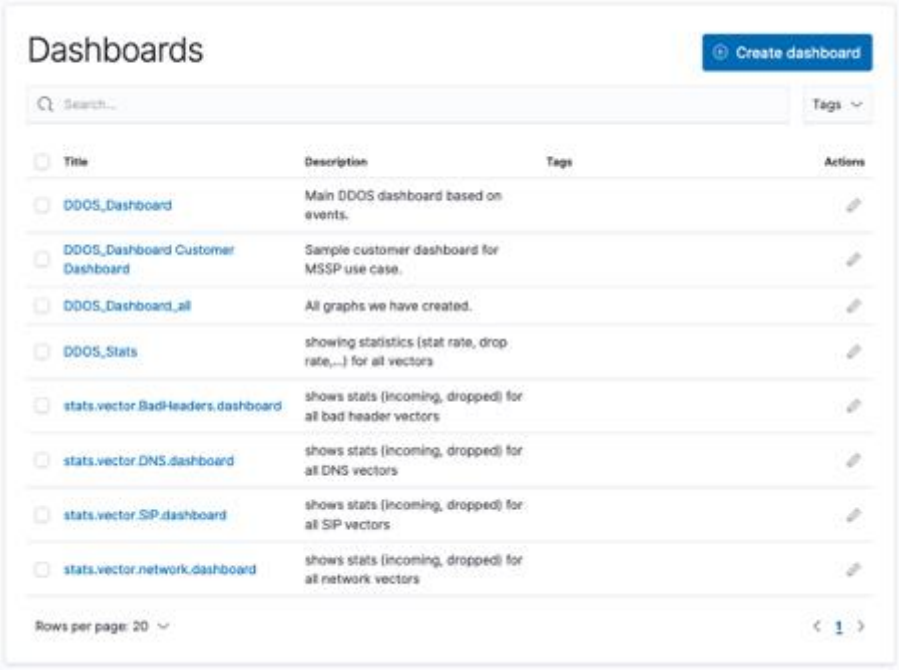


Рисунок 2.8 - Огляд дашбордів

DDoS_Dashboard - це панель, на якій ви можете бачити всі події за обраний період часу, який ви можете вибрати у верхньому правому куті цієї панелі (див. рис. 2.9).

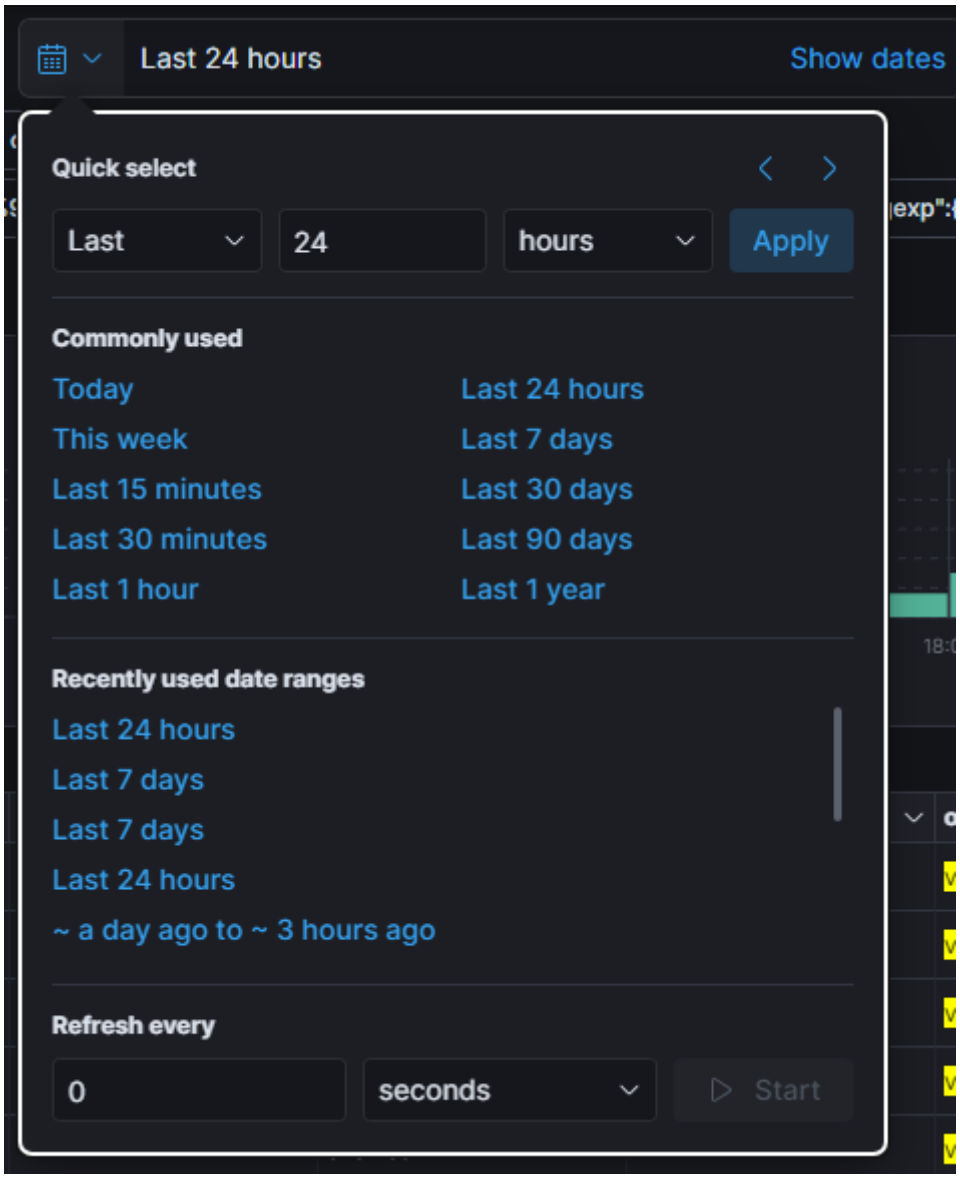


Рисунок 2.9 - Вибір періоду часу

У верхній частині сторінки ви знайдете Провідник інформаційної панелі (див. рис. 2.10). Звідси ви можете легко переміщатися між усіма відповідними дашбордами, не переходячи до розділу Аналітика в головному меню.

Рисунок 2.10 - Провідник дашбордів

DDOS STATS Dashboard: показує детальну інформацію про показники і пороги (швидкість передачі пакетів, поріг виявлення і пом'якшення, швидкість падіння) для всіх векторів, включаючи пороги зловмисника і атакованої адреси призначення. Тут вам потрібно вибрати відповідний вектор і контекст, щоб побачити деталі.

DDOS Network Vectors: показує детальну інформацію про вхідну швидкість і швидкість падіння для кожного мережевого вектора на одній сторінці.

DDOS DNS Vectors: показує детальну інформацію про кількість вхідних і вихідних повідомлень для кожного DNS-вектора на одній сторінці.

DDOS Bad Header Vectors: показує детальну інформацію про вхідний трафік і рівень падіння для кожного вектора поганих заголовків на одній сторінці.

DDOS SIP vector: показує детальну інформацію про кількість вхідних та вихідних дзвінків для кожного SIP-вектора на одній сторінці.

Далі ви побачите панель управління статистикою (див. рис. 2.11).

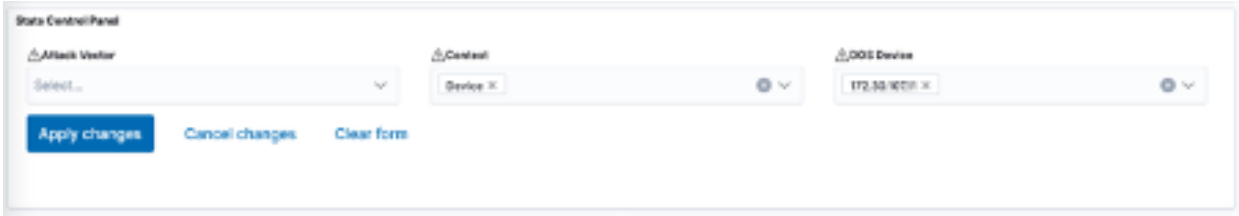


Рисунок 2.11 - Панель керування статистикою

За замовчуванням вона покаже події (drop/allow) для всіх векторів у всіх контекстах (VS/PO, Device) на всіх DoS-пристроях. Але за допомогою випадаючого меню ви можете відфільтрувати певні дані. Всі встановлені вами

фільтри також можна легко зберегти і використовувати знову. Kibana надає велику гнучкість.

Далі ви потрапляєте на часову шкалу найпопулярніших атак, яка показує вам 10 найпопулярніших векторів атак, які відкидали пакети (див. рис. 2.12).

Рисунок 2.12 - Часова шкала атак

При наведенні миші ви побачите кількість втрачених пакетів для цього вектора. Праворуч від цього графіка ви бачите деталі події атаки (див. рис. 2.13).

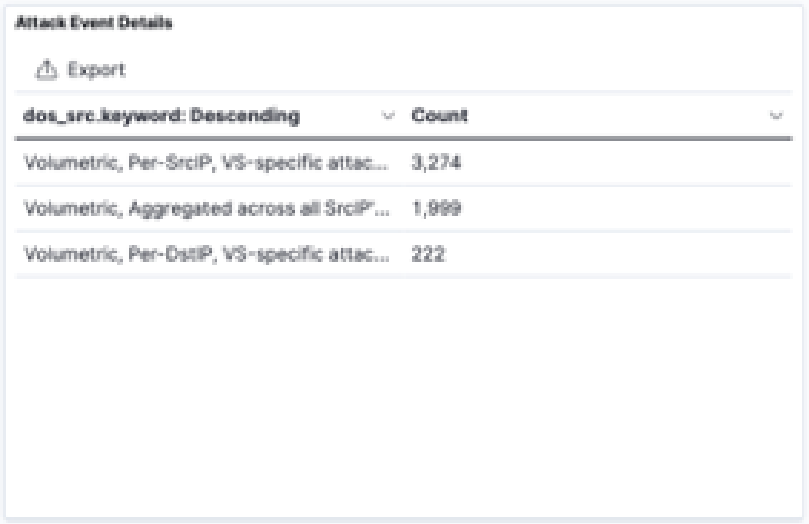


Рисунок 2.13 - Деталі події атаки

Тут показано, скільки логів ви отримали для кожної події. Пам'ятайте, що кожен механізм (наприклад, для кожної події джерела, призначення) має свої власні логи.

У наступному рядку ліворуч показано, скільки пакетів було втрачено за вибраний проміжок часу (див. рис. 2.14).

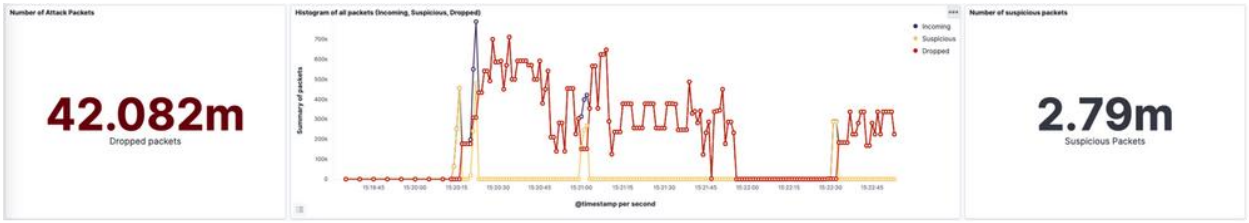


Рисунок 2.14 - Відкинуті та підозрілі пакети

Праворуч ви бачите, скільки пакетів було визначено як підозрілі, оскільки їх швидкість перевищувала поріг виявлення, але не перевищувала поріг пом'якшення. Це повідомлення про подію має дію "Дозволити".

На середньому графіку ви бачите співвідношення підозрілих пакетів, відкинутих пакетів і вхідних пакетів (вхідні пакети - це сума відкинутих і підозрілих пакетів).

Наступний графік дає вам також огляд отриманих пакетів у порівнянні з втраченими пакетами (див. рис. 2.15).

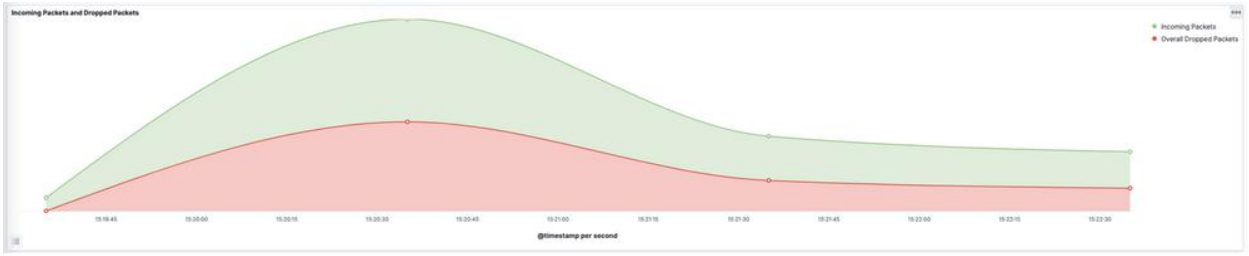


Рисунок 2.15 - Вхідні та відкинуті пакети

Основна відмінність від графіка на рисунку 2.14 полягає в тому, що ви бачитимете дані і тоді, коли події немає (дозволено/заборонено), оскільки залежно від налаштованої частоти надсилання таблиці "dos_stat", ви отримуватимете дані. Графіки, засновані на подіях журналу, звичайно, можуть з'являтися тільки тоді, коли є подія і надсилаються логи.

Цей графік показує всі вхідні пакети, підраховані за всіма увімкненими векторами, незалежно від того, чи вони підраховані на поганих акторів, атаковані адреси призначення або глобальну статистику за вектором. Те саме стосується і втрачених пакетів. Він дає загальний огляд вхідних і втрачених

пакетів. Щоб отримати більш детальну інформацію про те, який вектор або механізм (BA, AD) спричинив атаку, вам потрібно перейти до Панелі моніторингу DDOS STATS Dashboard.

Важливою інформацією для DDoS-оператора є інформація про те, які служби (IP-адреси) зазнали атаки і які контексти або захищені об'єкти були задіяні (див. рис. 2.16).

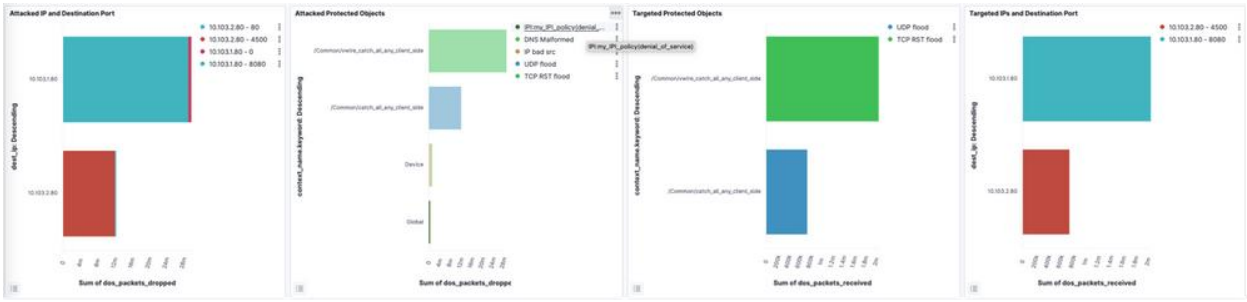


Рисунок 2.16 - Інформація про ціль

Звичайно, також, які вектори використовує зловмисник. Це показано в наступному рядку. На двох лівих графіках ви отримуєте цю інформацію для втрачених пакетів. На двох правих графіках ви бачите цю інформацію для пакетів, що перевищують поріг виявлення, але не досягають рівня усунення загрози.

Attacked IP and Destination Port показує атаковані IP-адреси, включаючи порти призначення.

Attacked Protected Objects показує контекст (VS/PO, Device, Global) по відношенню до векторів атаки. Контекст "Global" використовується для IPI (IP-розвідки). У цьому прикладі пакети були відкинуті, оскільки IP-адреси джерела були налаштовані в рамках політики IPI "my_IPI" і категорії "відмова в обслуговуванні". Усунення наслідків було виконано на глобальному рівні. Дії IPI показані як вектори атаки (див. рис. 2.17).

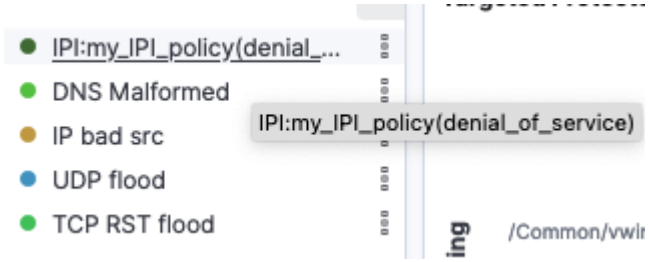


Рисунок 2.17 - Інформація IP-розвідки

При наведенні миші ви можете отримати повний рядок. Більш детальну інформацію про вектори атаки та активність IPI ви побачите нижче на сторінці Attack destination details. Там ви знайдете таблицю з інформацією про IP-адреси, які були ідентифіковані як атаковані за допомогою функції "Attack destination details", налаштованої на векторі (див. рис. 2.18).

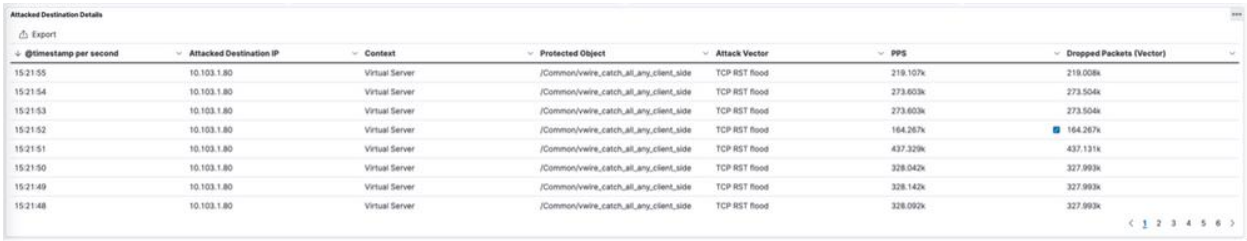


Рисунок 2.18 - Виявлення призначення атаки

Таблиця містить інформацію про різні аспекти атаки DDoS:

– Attacker Destination IP (10.103.1.80) - цей стовпець показує IP-адресу цільового системи, на яку направлена атака. У цьому випадку, цільова IP-адреса є 10.103.1.80;

– Context (Virtual Server) - в даному контексті, "Virtual Server" може означати, що атака цілиться на віртуальний сервер в мережі. Це надає інформацію про середовище, в якому цільова система розміщена;

– Protected Object (/Common/vwire_catch_all_any_client_side) - цей стовпець ідентифікує захищений об'єкт, який був атакований. У даному

випадку, це "/Common/vwire_catch_all_any_client_side", що, мабуть, є назвою віртуального сервера або конкретного ресурсу на сервері;

– Attack Vector (TCP RST flood) - цей стовпець показує використаний вектор атаки. TCP RST flood - це тип DDoS-атаки, де зловмисник шле велику кількість TCP RST (Reset) пакетів до цільової системи з метою переривання з'єднань;

– PPS (219.107k) - PPS означає пакети за секунду. Це метрика пропускної спроможності, яка вимірює кількість пакетів, переданих кожну секунду. У даному випадку, є 219.107 тисяч пакетів за секунду;

– Dropped Packets (219.008k) - цей стовпець показує кількість пакетів, які були відкинуті системою з різних причин, можливо, як частину заходів захисту від DDoS-атак. У даному випадку, відкинуто 219.008 тисяч пакетів.

Ці дані важливі для аналізу DDoS-атак та визначення відповідної стратегії захисту.

Висновок

Інформаційні панелі DDoS, базовані на стеку ELK, надають операторам DDoS можливість контролювати події DDoS.

Інформаційна панель обробляє журнали, які надсилає BIG-IP, на основі подій DDoS на рівнях L3/4/DNS і візуалізує їх у вигляді графіків. Ці графіки надають важливу інформацію про типи атак, їхні джерела та цілі. В залежності від вашої конфігурації BIG-IP DoS, ви отримаєте деталі про "поганих акторів" або деталі про "атаковані цілі". Ви також побачите, які IP-адреси були заблоковані за певними категоріями IPI та багато іншого. На додаток до іншої показаної інформації, стек ELK також здатний обробляти дані з таблиці dos_stats, яка надає вам деталі про поведінку вашої мережі на рівні векторів. Крім того, ви можете побачити, як "автоматичні пороги" розраховують пороги виявлення та зменшення загроз.

3 ВПРОВАДЖЕННЯ СИСТЕМИ ДЛЯ АНАЛІЗУ ЛОГ-ФАЙЛІВ З ВИКОРИСТАННЯМ ELK

3.1 Встановлення Elasticsearch, Logstash і Kibana

3.1.1 Встановлення Elasticsearch

Для впровадження системи аналізу лог-файлів, ключовим етапом є встановлення Elasticsearch, Logstash та Kibana, які разом утворюють ELK стек. Вибір ОС Windows в даному контексті обумовлений рядом причин. Windows надає зручний інтерфейс для встановлення та конфігурації, а також підтримує широкий спектр програмного забезпечення.

Процес встановлення Elasticsearch на Windows за допомогою архіву Windows .zip - це послідовна процедура, яка включає кілька ключових кроків. Важливо зазначити, що цей підхід до встановлення включає команду elasticsearch-service.bat, яка налаштовує Elasticsearch на роботу в якості сервісу.

Перед початком процесу встановлення необхідно переконатися, що функція машинного навчання Elasticsearch підтримується бібліотекою Microsoft Universal C Runtime, яка входить до складу Windows 10, Windows Server 2016 та більш пізніх версій Windows. У старіших версіях Windows цю бібліотеку можна встановити через Windows Update або завантажити окремо. Якщо бібліотеку Microsoft Universal C Runtime неможливо встановити, функцію машинного навчання можна відключити, щоб використовувати інші компоненти Elasticsearch.

Початковий етап процесу встановлення передбачає отримання останньої стабільної версії Elasticsearch, яку можна знайти на сторінці завантаження Elasticsearch (див. рис. 3.1). Згодом завантажується та встановлюється .zip-архів для Elasticsearch Завантажений .zip-архів потім розпаковується за допомогою бажаного інструменту, в результаті чого створюється каталог,

позначений як %ES_HOME%. Перехід до каталогу %ES_HOME% у вікні терміналу означає завершення цього етапу.

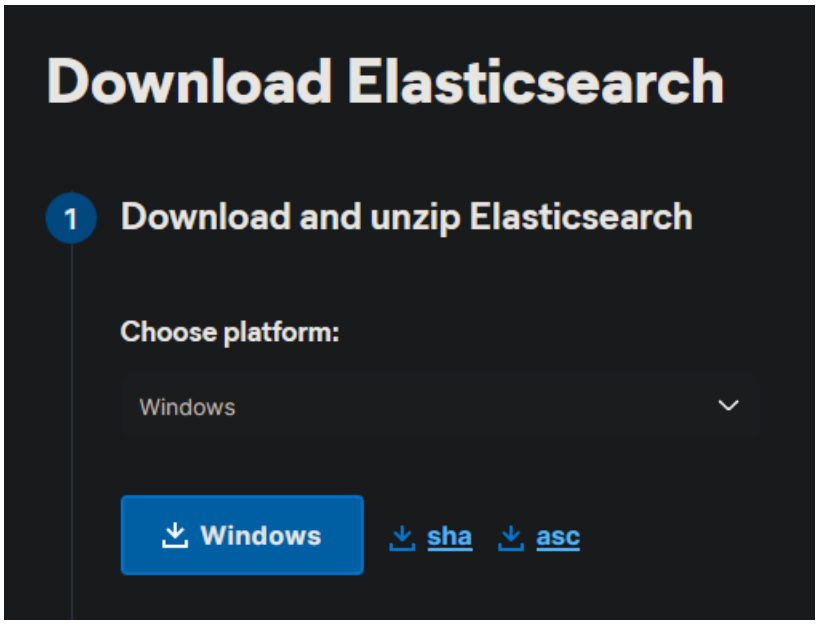


Рисунок 3.1 - Сторінка завантаження Elasticsearch

Наступним кроком процесу є запуск Elasticsearch з командного рядка, який включає певні автоматичні конфігурації безпеки. Виконавши команду «elasticsearch.bat», ви можете побачити вивід та журнали запуску Elasticsearch. На рисунку 3.2 ви можете бачити різні повідомлення, інформацію про запущені процеси Elasticsearch та будь-які помилки, що виникають під час запуску сервера (див. рис 3.2). Під час першого запуску Elasticsearch такі функції безпеки, як автентифікація та авторизація, вмикаються за замовчуванням, генерується пароль для еластичного вбудованого суперкористувача, а також сертифікати та ключі для безпеки транспортного рівня (TLS) для транспортного та HTTP-рівня. Крім того, для Kibana генерується токен реєстрації, який дійсний протягом 30 хвилин. Пароль для еластичного користувача і токен реєстрації для Kibana виводяться на термінал під час цього кроку.

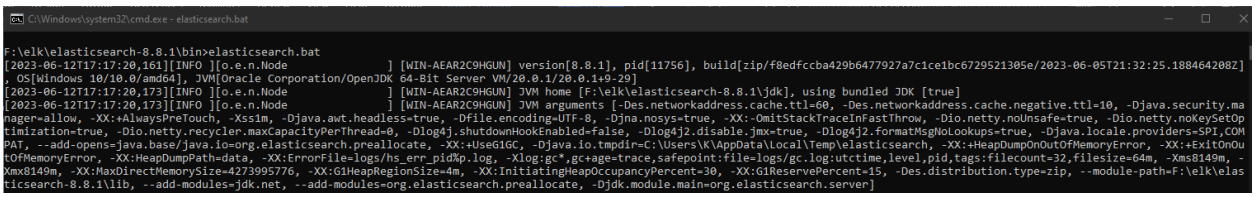


Рисунок 3.2 - Запуск Elasticsearch з командного рядка

Щоб перевірити, що Elasticsearch працює належним чином, можна надіслати HTTPS-запит на порт 9200 на localhost. Важливо використовувати https у запиті, інакше запит не пройде. Пароль користувача “elastic”, згенерований під час інсталяції, вводиться у відповідь на запит, який має повернути певну відповідь. На рисунку 3.3 ви побачите вікно введення пароля, так як для доступу до інтерфейсу Elasticsearch була встановлена аутентифікація.

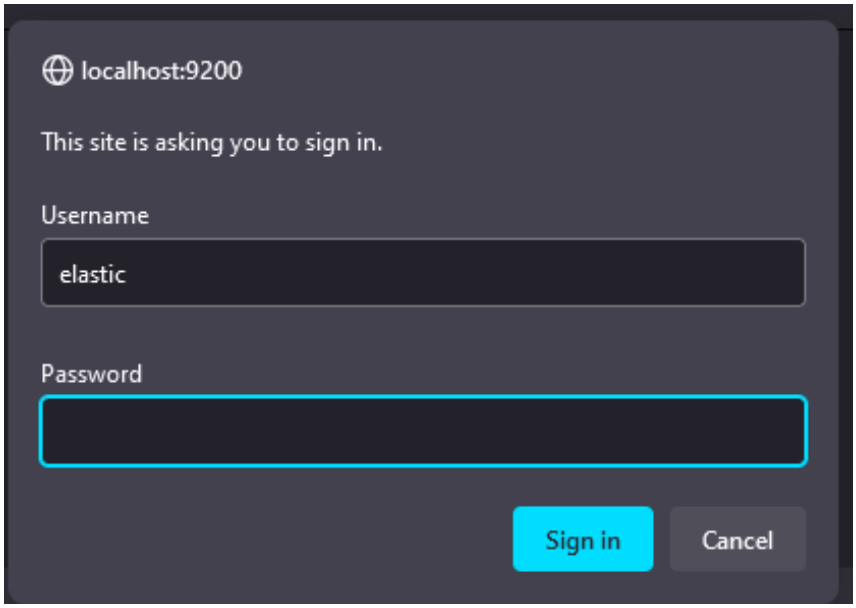


Рисунок 3.3 - Вікно введення логіну та паролю

Після введення логіну та паролю і успішної аутентифікації у веб-браузері ви отримаєте відповідь у форматі JSON. Вона буде містити інформацію про Elasticsearch, зокрема:

– "name" - назва вузла Elasticsearch;
– "cluster_name" - назва кластера Elasticsearch;

– "cluster_uuid" - унікальний ідентифікатор кластера Elasticsearch;
– "version" - інформація про версію Elasticsearch;
– "tagline" - гасло Elasticsearch, що описує його функціонал.

Цей JSON надає загальну інформацію про запущений екземпляр Elasticsearch, його версію та налаштування. Він може бути корисним для перевірки версії, контролю стану кластера та отримання основної інформації про сервер Elasticsearch. На рисунку 3.4 ви можете бачити результат (див. рис. 3.4).

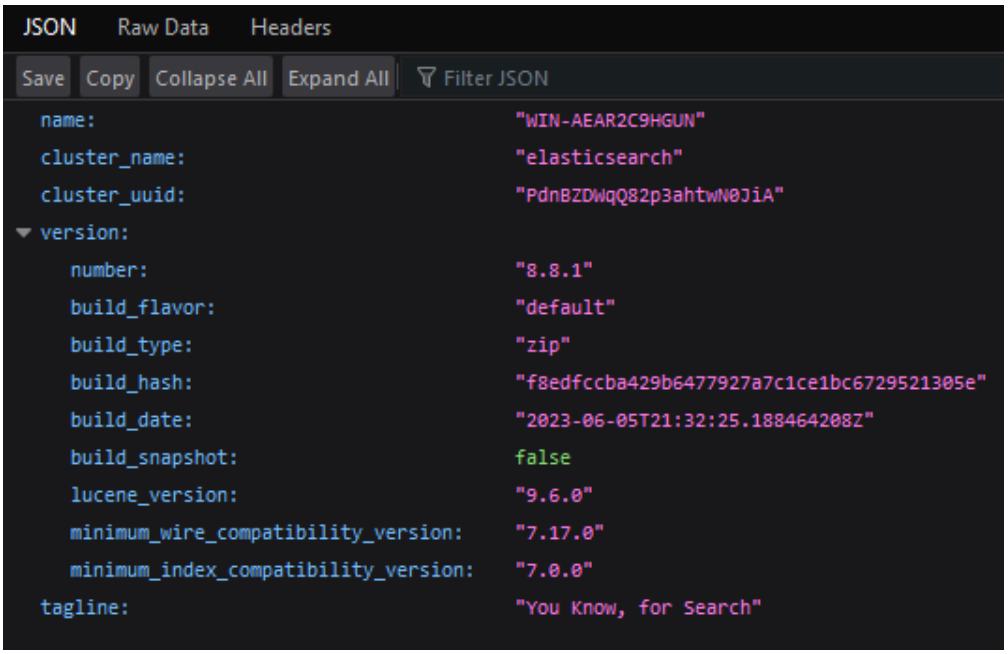


Рисунок 3.4 - Результат введення облікових даних для входу

Elasticsearch можна встановити і запустити як службу в Windows. Це означає, що він може працювати у фоновому режимі або запускатися автоматично під час завантаження без будь-якої взаємодії з користувачем. Таким чином, процес інсталяції Elasticsearch в Windows за допомогою архіву Windows .zip завершено. На рисунку 3.5 у вікні "Служби Windows" ви можете побачити список установлених служб на вашому комп'ютері. Серед цих служб можна знайти Elasticsearch, якщо він встановлений та працює. Якщо

Elasticsearch працює, ви можете побачити інформацію про нього, таку як назва служби (зазвичай "Elasticsearch"), статус (запущено), тип запуску (автоматичний або ручний), ім'я облікового запису, від імені якого він працює, та, можливо, додатковий опис. Це вікно дозволяє вам керувати станом служби Elasticsearch, зупиняти або запускати його, змінювати параметри запуску та переглядати журнали подій, пов'язані з цією службою. Таким чином, ви можете підтвердити, що Elasticsearch працює, якщо бачите, що його статус - "Запущено" (див. рис. 3.5).

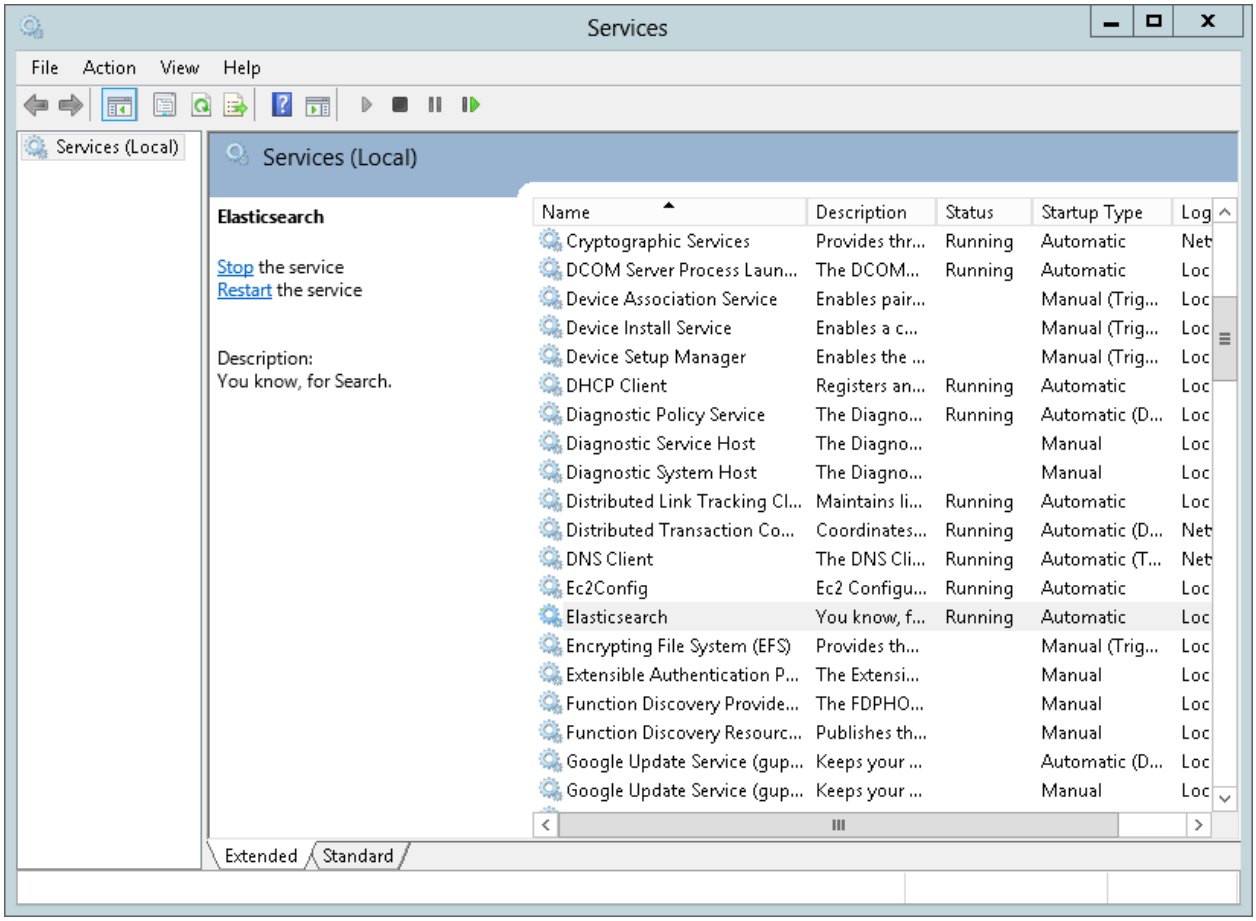


Рисунок 3.5 - Запуск Elasticsearch як служби в Windows

3.1.2 Встановлення та налаштування Logstash для обробки лог-файлів

Процес встановлення Logstash на Windows починається з завантаження актуального дистрибутива з офіційного веб-сайту (див. рис 3.6). При цьому вибір версії відіграє важливу роль і залежить від специфіки конкретної

ситуації та вимог. Завантажений дистрибутив Logstash розпаковується до обраного каталогу, що становить завершення цього етапу.

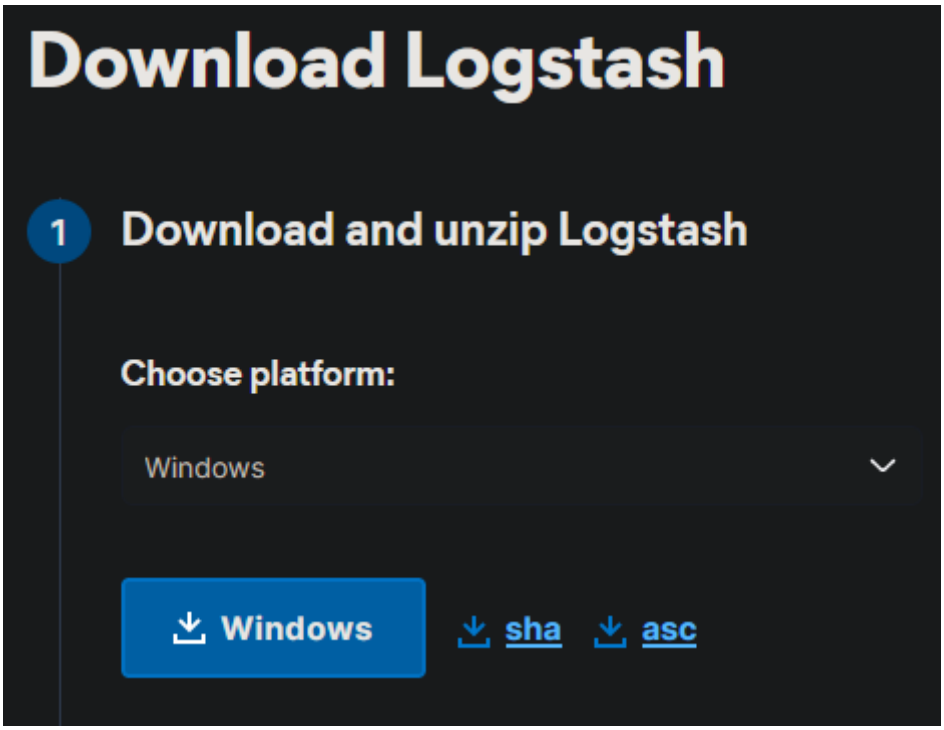


Рисунок 3.6 - Сторінка завантаження Logstash

Наступний етап включає конфігурацію Logstash. Для цього використовуються конфігураційні файли, де задаються вхідні дані, фільтри для перетворення даних та вихідні потоки, куди дані будуть відправлені після обробки. На цьому етапі визначається шлях до лог-файлу, задаються необхідні фільтри та вказується Elasticsearch як місце зберігання даних. На лістингу 3.1 ви можете бачити стандартний формат файлу конфігурації, який потім буде використовуватися для написання фільтрів (див. лістинг 3.1).

Лістинг 3.1 - Початковий файл конфігурації «learn.conf»

```
input {
  stdin { } # Вхідний потік для отримання даних з консолі
}
output {
  stdout {
    codec => rubydebug # Вивід даних у форматі Ruby-відладки
  }
```

Продовження лістингу 3.1

```
       elasticsearch {
         hosts => ["http://localhost:9200"] # Адреси серверів
Elasticsearch
         index => "test.logstash" # Індекс, у який будуть
зберігатись дані
         user => "elastic" # Користувач Elasticsearch
         password => "3guzVh6WIwXCb*qVPQfp" # Пароль користувача
Elasticsearch
       }
     }
```

Цей фрагмент конфігураційного файлу Logstash використовує плагіни stdin для отримання даних з консолі та stdout для виводу даних у форматі Ruby-відладки на консоль.

Також використовується плагін elasticsearch для відправки даних до серверу Elasticsearch. У цьому плагіні вказуються адреси серверів Elasticsearch, індекс, в який будуть зберігатись дані, та облікові дані користувача Elasticsearch для аутентифікації.

Після конфігурації Logstash запускається з створеним конфігураційним файлом. Після виконання команди "logstash -f .\config\learn.conf --config.reload.automatic" у командному рядку, відкриється вікно командного рядка, де буде відображений вивід та журнали запуску Logstash.

Ця команда дозволяє запустити Logstash з вказаною конфігурацією з файлу "learn.conf" та встановленим параметром автоматичного перезавантаження конфігурації (--config.reload.automatic). Це дозволяє автоматично оновлювати конфігурацію Logstash при зміні файлу "learn.conf" без необхідності зупиняти та запускати Logstash вручну.

На рисунку 3.8 ви зможете побачити вікно, де буде показана різноманітна інформація, включаючи статус запуску Logstash, інформацію про конфігураційний файл "learn.conf", а також відображені повідомлення про завантаження конфігурації та будь-які помилки або проблеми, що виникають під час роботи Logstash (див. рис. 3.8).

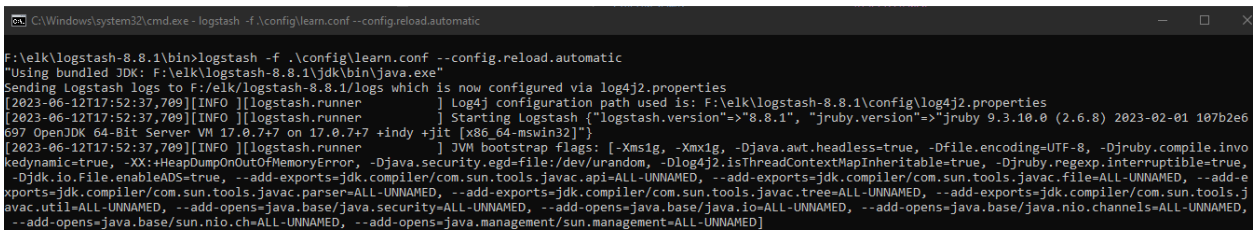


Рисунок 3.8 - Запуск Logstash з командного рядка

Коректність роботи Logstash перевіряється шляхом аналізу його логів та перегляду даних в Elasticsearch. На рисунку 3.9 ви побачите вивід, після введення команди та написання «test» у вікні командного рядка Logstash (див. рис. 3.9).

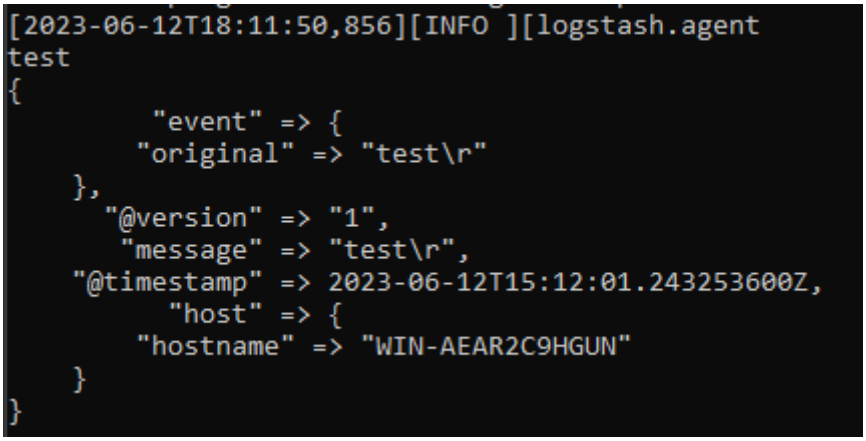


Рисунок 3.9 - Перевірка коректності відправлення лог-файлів

Цей вивід представляє оброблені дані за допомогою Logstash. Дані відображаються у форматі JSON, де кожен рядок представляє окремий ключ і значення.

У цьому конкретному виводі ви бачите наступне:

– "event" містить ключ "original" зі значенням "test\r", яке представляє оригінальну вхідну подію;

– "@version" містить значення "1", що вказує на версію обробленої події;

– "message" містить значення "test\r", що є повідомленням, яке було оброблено;

– "@timestamp" містить значення, що представляє мітку часу події;

– "host" містить ключ "hostname" зі значенням "WIN-AER2C9HGUN", що вказує на хост, з якого була отримана подія.

Цей вивід представляє оброблені дані в результаті виконання обробки події "test" за допомогою конфігураційного файлу Logstash.

3.1.3 Встановлення та налаштування Kibana для візуалізації результатів

Процес встановлення Kibana на Windows є послідовною дією, яка охоплює декілька важливих кроків. З початку потрібно завантажити актуальний дистрибутив Kibana з офіційного веб-сайту (див. рис. 3.10). Обираючи версію, потрібно враховувати специфіку вашого проекту та вимоги до системи. Після завантаження, дистрибутив Kibana розпаковується до вибраного каталогу, що означає завершення цього етапу.

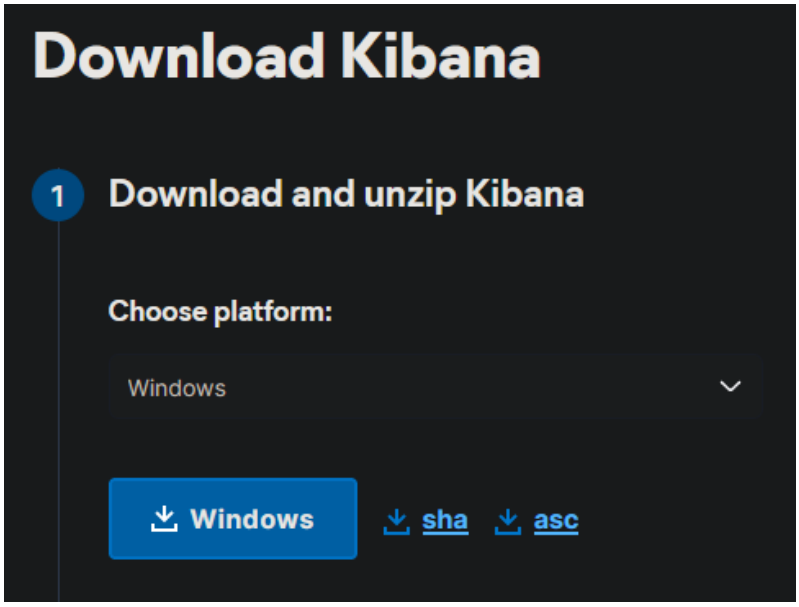


Рисунок 3.10 - Сторінка завантаження Kibana

Далі йде конфігурація Kibana. Для цього потрібно відредагувати конфігураційний файл Kibana (kibana.yml), що знаходиться в папці config. В цьому файлі вказується адреса Elasticsearch, що дозволяє Kibana взаємодіяти з Elasticsearch для отримання та візуалізації даних.

Потрібно зробити такі дії:

– розкоментуйте властивість server.port: Знайдіть рядок, який містить # server.port і змініть його на server.port. Це розкоментує властивість server.port, що дозволить вам встановити порт, на якому Kibana буде слухати запити. Замість значення за замовчуванням ви можете вказати потрібний вам порт;

– розкоментуйте властивість server.host і встановіть значення "0.0.0.0" для прослуховування всіх адрес: Знайдіть рядок, який містить # server.host і замініть його на server.host. Це розкоментує властивість server.host, дозволяючи Kibana слухати запити на всіх доступних мережевих інтерфейсах. Встановіть значення "0.0.0.0", щоб дозволити Kibana слухати на всіх IP-адресах;

– розкоментуйте властивість elasticsearch.hosts: Знайдіть рядок, який містить # elasticsearch.hosts і замініть його на elasticsearch.hosts. Це розкоментує властивість elasticsearch.hosts, дозволяючи Kibana взаємодіяти з Elasticsearch. Ви можете вказати адреси серверів Elasticsearch, з якими Kibana повинна спілкуватися;

– розкоментуйте властивість elasticsearch.username: Знайдіть рядок, який містить # elasticsearch.username і замініть його на elasticsearch.username. Це розкоментує властивість elasticsearch.username, дозволяючи вказати ім'я користувача для аутентифікації Kibana у Elasticsearch;

– видаліть коментар і встановіть властивість elasticsearch.password з паролем, встановленим на етапі "Налаштування паролів": Знайдіть рядок, який містить # elasticsearch.password і замініть його на elasticsearch.password. Видаліть коментар та встановіть властивість elasticsearch.password з паролем, який ви вказали на етапі налаштування паролів Elasticsearch.

Ці зміни у файлі "kibana.yml" дозволять налаштувати Kibana для з'єднання з Elasticsearch, використовуючи вказані порт, хости, ім'я користувача та пароль. Це дозволить Kibana взаємодіяти з Elasticsearch для отримання даних та роботи з ними. На рисунках 3.11 - 3.12 можете побачити вигляд файлу kibana.yml після виконаних дій.

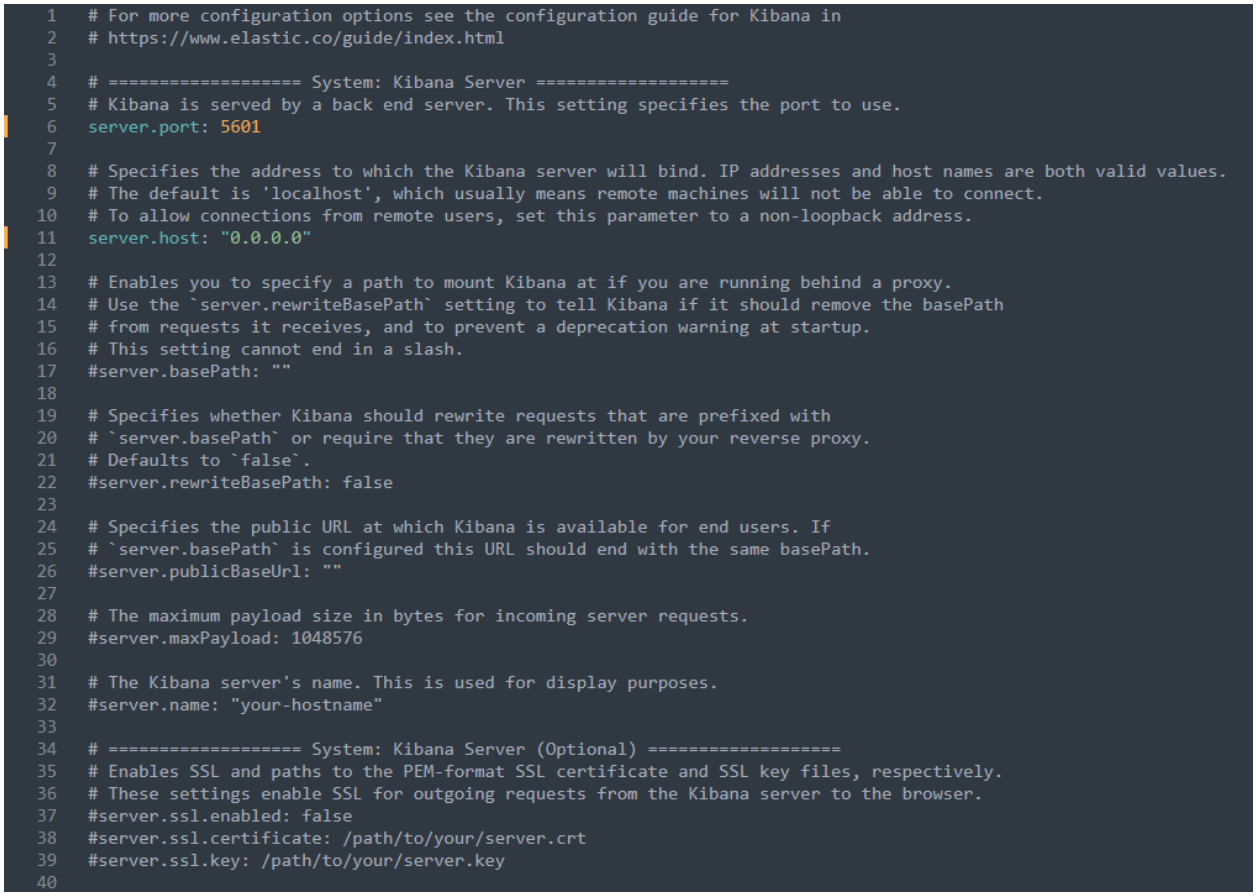


Рисунок 3.11 - Зміни до «Kibana Server» у файлі kibana.yml

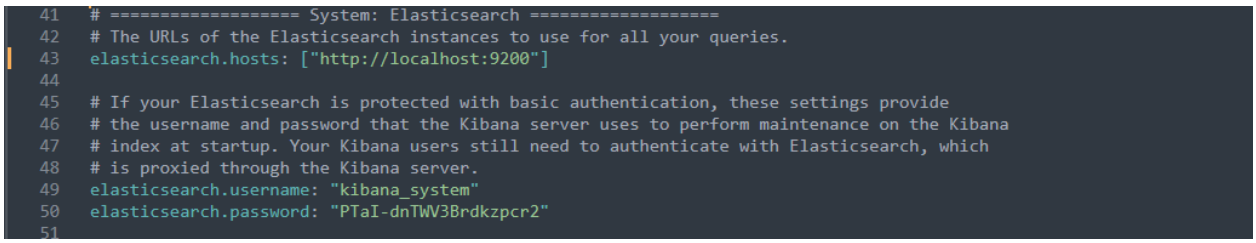


Рисунок 3.12 - Зміни до «Elasticsearch» у файлі kibana.yml

Запуск Kibana відбувається через командний рядок, введенням команди «kibana.bat» в директорії Kibana.

Запускається Kibana, і ми побачимо наступне:

– відкривається вікно командного рядка Kibana, яке відображає вивід і журнали запуску Kibana. У цьому вікні ми бачимо різну інформацію, включаючи версію Kibana, статус запуску та повідомлення про будь-які

помилки або проблеми. Ми можемо бачити процес запуску Kibana та повідомлення про успішний запуск на рисунку 3.13 (див. рис. 3.13);

– у вікні командного рядка відображається URL-адреса, за допомогою якої ми можемо отримати доступ до веб-інтерфейсу Kibana. Зазвичай це буде щось на зразок «http://localhost:5601» або «http://127.0.0.1:5601». Ми можемо скопіювати цю URL-адресу і відкрити її в браузері для доступу до Kibana;

– після успішного запуску Kibana ми можемо працювати з його веб-інтерфейсом. Цей інтерфейс надає нам різноманітні можливості, включаючи візуалізацію та аналіз даних, налаштування панелей і панелей керування, роботу з індексами та інші функції, пов'язані з обробкою та відображенням даних.

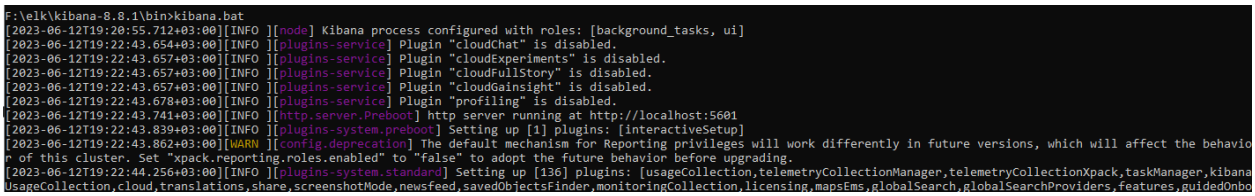


Рисунок 3.13 - Запуск Kibana через командний рядок

Після запуску Kibana, ми можемо перевірити, чи вона правильно працює, перейшовши в браузері за адресою «http://localhost:5601».

Загальні елементи сторінки автентифікації Kibana можуть включати:

– поле введення імені користувача: Користувач повинен ввести своє ім'я користувача або електронну пошту відповідно до налаштувань;

– поле введення пароля: Користувач повинен ввести свій пароль, який відповідає обліковим даним, встановленим у Kibana або іншій системі автентифікації, що використовується;

– кнопка "Увійти" або аналогічний елемент керування: Користувач натискає цю кнопку після введення своїх облікових даних для початку процесу автентифікації;

– посилання на відновлення пароля або інші опції відновлення доступу: Залежно від конкретної реалізації автентифікації, можуть бути надані посилання для відновлення пароля, відновлення облікового запису або інші опції відновлення доступу, якщо користувач забув свій пароль або має проблеми з авторизацією;

– повідомлення про помилку або підтвердження успішного входу: Після введення облікових даних, система перевіряє їх на коректність. У разі невірних облікових даних може з'явитися повідомлення про помилку, а при успішному вході може з'явитися підтвердження автентифікації та перехід до наступних сторінок Kibana.

На рисунку 3.14 ми бачимо поле введення імені користувача, поле введення пароля та кнопку увійти (див. рис. 3.14).

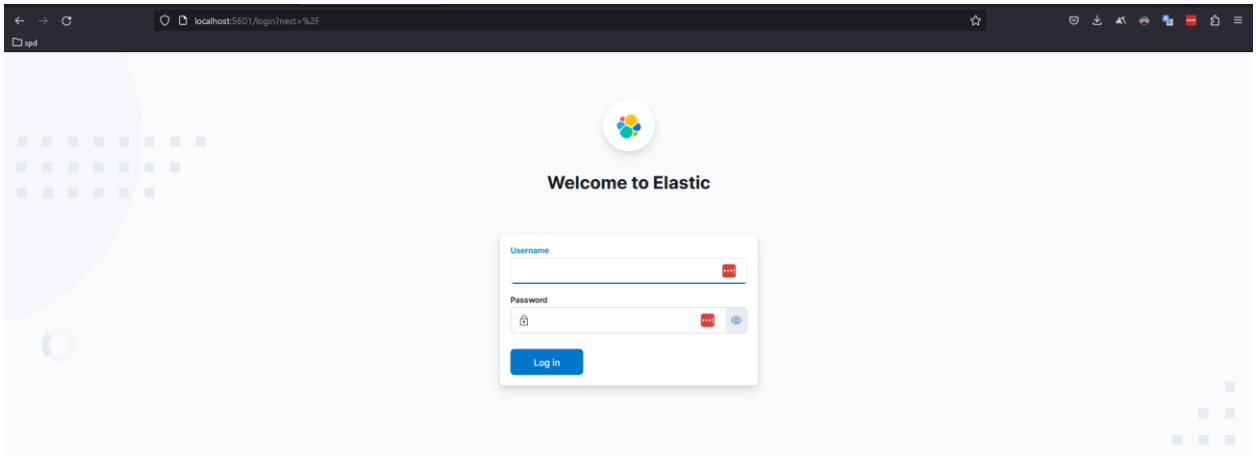


Рисунок 3.14 - Сторінка автентифікації користувача у Kibana

На веб-сторінці Kibana, у вікні «Visualize Library» ви зможете створювати візуалізації та оглядати вже існуючі дані, що надходять з Elasticsearch. Також на рисунках 3.15 - 3.16 ви зможете побачити набір доступних типів візуалізації, серед яких зазначені: "Metric" (метрика), "Bar Horizontal" (горизонтальна стовпчаста діаграма), "Bar Horizontal Percentage" (горизонтальна стовпчаста діаграма у відсотках), "Bar Vertical" (вертикальна стовпчаста діаграма), "Bar Vertical Percentage" (вертикальна стовпчаста діаграма у відсотках), "Proportion: Donut" (кругова діаграма), "Proportion: Pie"

(секторна діаграма), "Treemap" (діаграма у вигляді дерева) і "Heatmap" (теплова карта) (див. рис. 3.15 - 3.16).

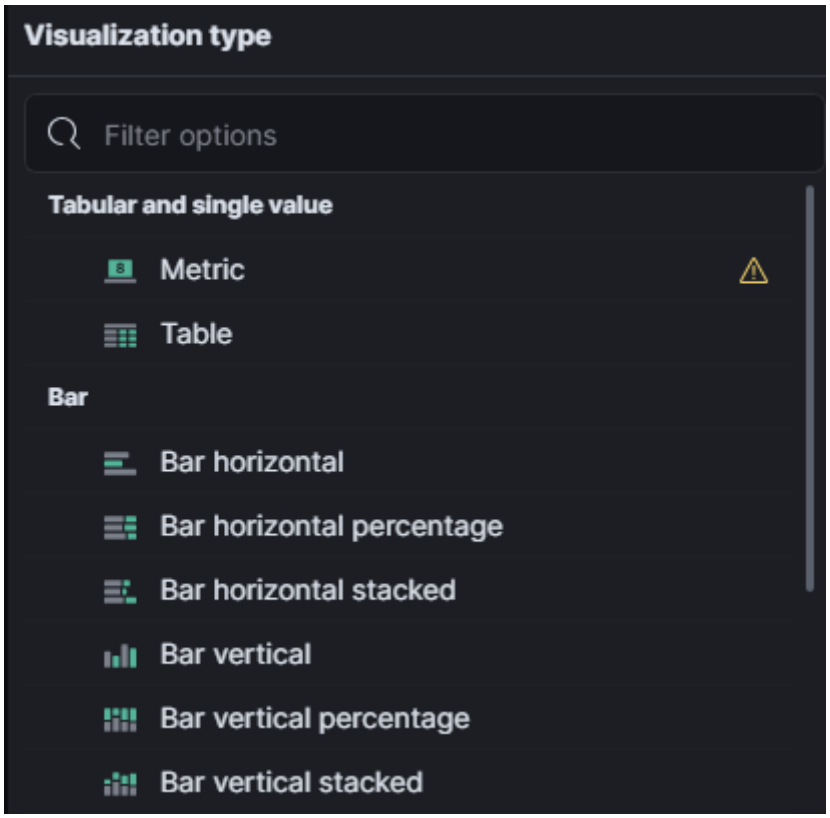


Рисунок 3.15 - Перша частина доступних типів візуалізації у Kibana

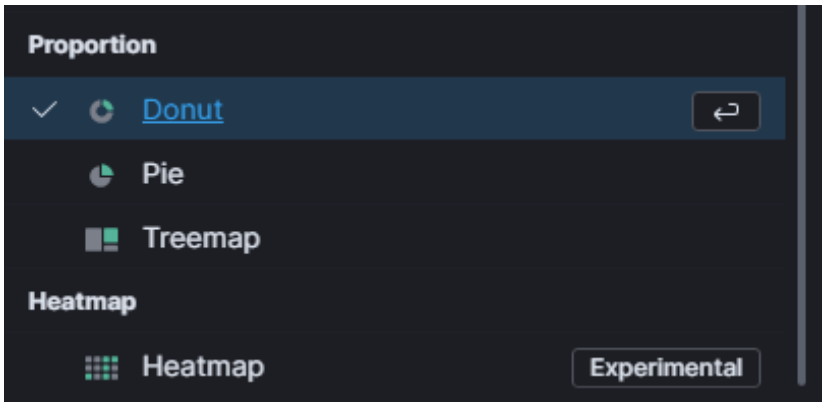


Рисунок 3.16 - Друга частина доступних типів візуалізації у Kibana

Давайте використаємо тип візуалізації у формі "Donut" для аналізу країн, які найбільше зустрічаються у наших лог-файлах протягом останнього тижня. На графіку будуть відображені країни разом з відсотковим співвідношенням.

В нашому випадку, ми спостерігаємо наступні результати (див. рис. 3.17):

– Росія: 89.65%;
– Італія: 7.45%;
– Сполучені Штати: 1.16%;
– Велика Британія: 0.37%;
– Японія: 0.08%;
– Нідерланди: 0.01%.

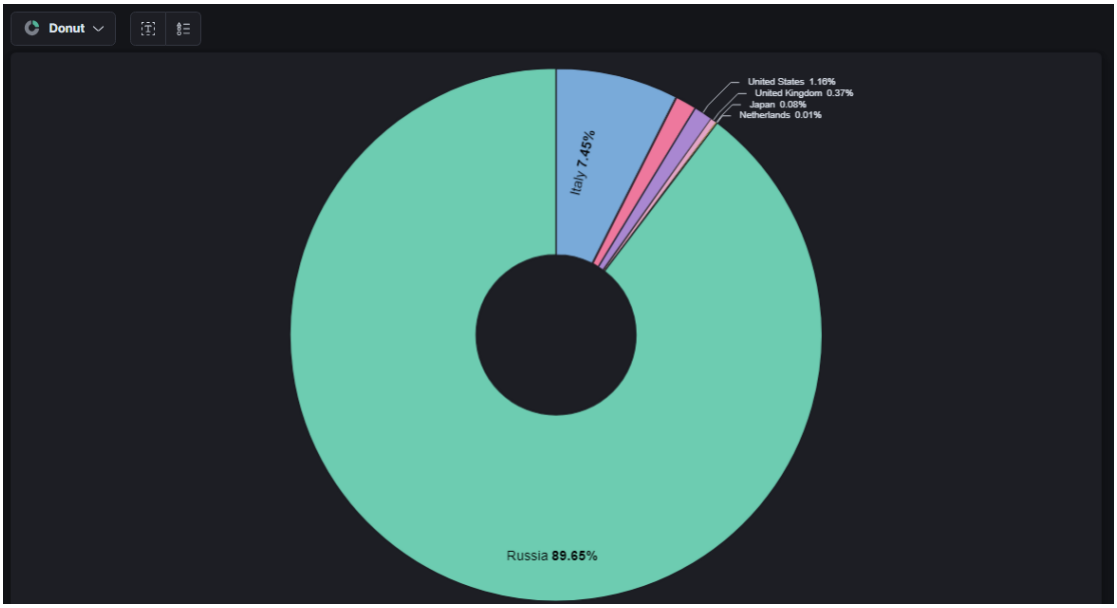


Рисунок 3.17 - Графік країн, з яких найбільше надходило брутфорс-атак

Цей Donut-графік дозволяє нам зрозуміти, які країни є основними джерелами лог-подій за останній тиждень та визначити їх відносну участь у загальному обсязі.

3.2 Розробка шаблонів та фільтрів для аналізу лог-файлів

3.2.1 Розробка шаблонів для Elasticsearch

Шаблони індексів Elasticsearch - це спосіб указати Elasticsearch, як налаштувати індекс при його створенні. Для потоків даних шаблон індексу налаштовує індекси, що є основою для потоку, в момент їх створення. Шаблони налаштовуються до створення індексу. Коли індекс створюється - або вручну, або через індексацію документа - налаштування шаблону використовуються як основа для створення індексу.

Існують два типи шаблонів: шаблони індексів та компонентні шаблони. Компонентні шаблони - це повторно використовувані елементи, які налаштовують відображення, налаштування та псевдоніми. Хоча ви можете використовувати компонентні шаблони для створення шаблонів індексів, вони не застосовуються безпосередньо до набору індексів. Шаблони індексів можуть містити колекцію компонентних шаблонів, а також безпосередньо вказувати налаштування, відображення та псевдоніми.

Приклад створення компонентного шаблону в лістингу 3.2.

Лістинг 3.2 - Запити на створення компонентних шаблонів

```
PUT _component_template/component_template1
{
  "template": {
    "mappings": {
      "properties": {
        "@timestamp": {
          "type": "date"
        }
      }
    }
  }
}


PUT _component_template/runtime_component_template
{
  "template": {
    "mappings": {
      "runtime": {
        "day_of_week": {
          "type": "keyword",
          "script": {
            "source":
"emit(doc['@timestamp'].value.dayOfWeekEnum.getDisplayName(TextS
tyle.FULL, Locale.ROOT))"
          }
        }
      }
    }
  }
}
```

Перший запит створює компонентний шаблон з назвою "component_template1". У шаблоні визначена маппінгове налаштування для поля "@timestamp", яке встановлюється як тип "date". Це дозволяє Elasticsearch коректно індексувати та обробляти дані, пов'язані з часом.

Другий запит створює компонентний шаблон з назвою "runtime_component_template". У шаблоні визначається спеціальний маппінг для полів рантайму. У цьому конкретному прикладі створюється поле "day_of_week" з типом "keyword". Для значень цього поля використовується скрипт, який витягує день тижня з поля "@timestamp" і конвертує його у повний текстовий формат за допомогою локалізованого відображення.

Виконаємо ці команди у консолі вікна Dev Tools, використовуючи Kibana. На рисунку 3.17 зображене це вікно, введений запит та отримання відповіді "{ "acknowledged": true }", це означає, що запити були успішно виконані і компонентні шаблони "component_template1" і "runtime_component_template" були створені (див. рис. 3.18).

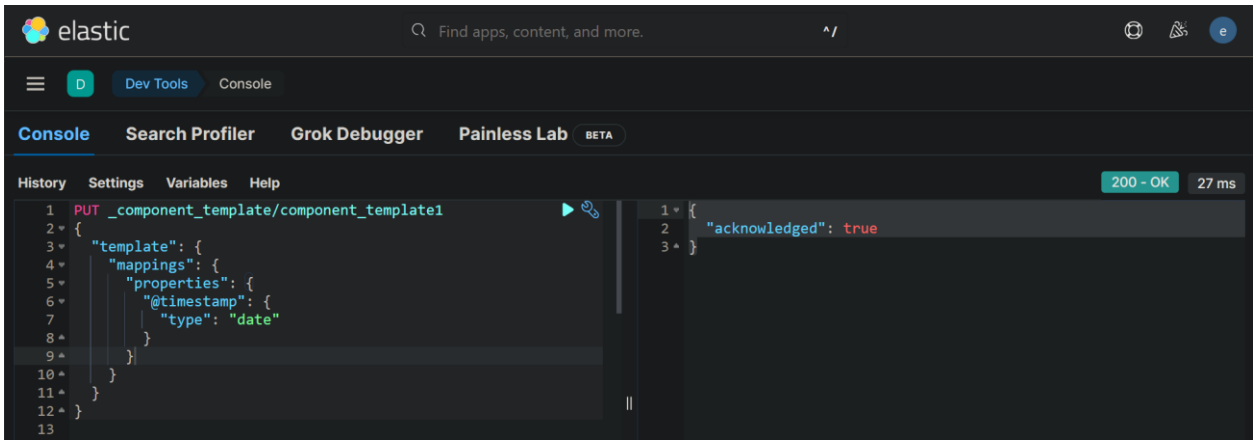


Рисунок 3.18 - Успішне виконання команд у консолі Dev Tools, Kibana

Для перевірки, чи вони працюють, виконаємо наступні запити:

GET _component_template/component_template1

GET _component_template/runtime_component_template

Ці запити повернуть інформацію про відповідні компонентні шаблони. Ми можем перевірити, чи відповідні шаблони присутні і містять визначені настройки та маппінги полів.

На рисунку 3.19 результат виконання команд (див. рис. 3.19).

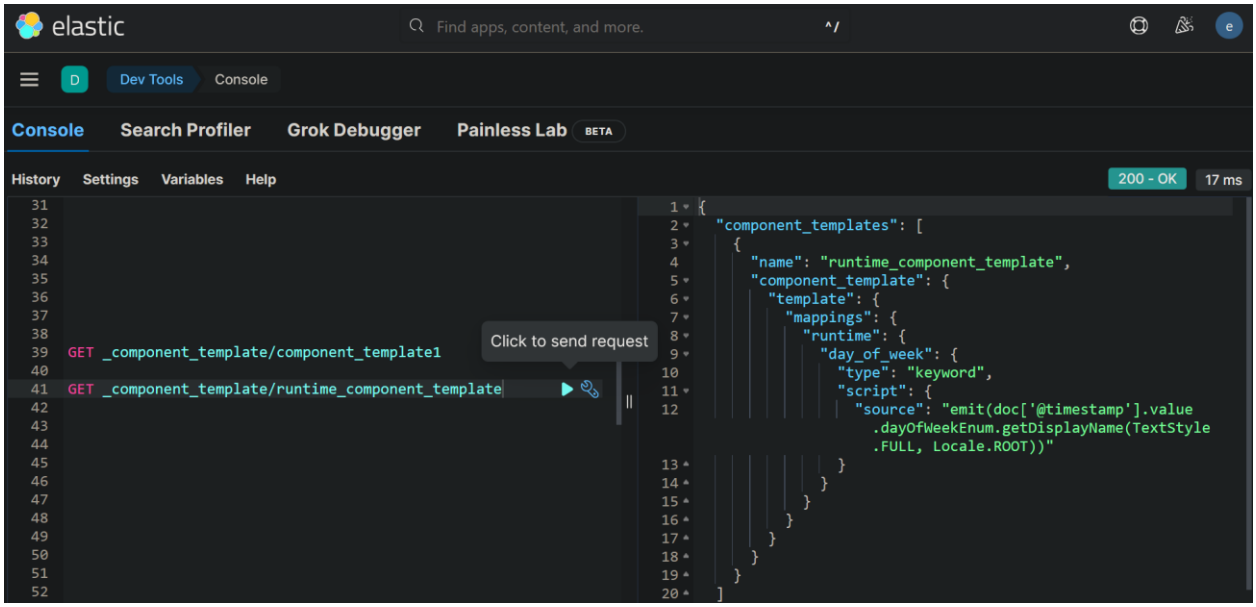


Рисунок 3.19 - Успішний результат виконання команд, який підтверджує присутність шаблонів і вміст визначених налаштувань

Після виконання цих запитів ми отримали відповіді у форматі JSON, які містять детальну інформацію про компонентні шаблони, включаючи настройки, маппінги та інші визначені елементи.

Наступний запит створює шаблон індексу, який складається з цих компонентних шаблонів (див. лістинг 3.3):

Лістинг 3.3 - Створення індексного шаблону з налаштуваннями

```
PUT _index_template/template_1
{
  "index_patterns": ["te*", "bar*"],
  "template": {
    "settings": {
      "number_of_shards": 1
    },
    "mappings": {
```

```
      "_source": {
        "enabled": true
      },
      "properties": {
        "host_name": {
          "type": "keyword"
        },
        "created_at": {
          "type": "date",
          "format": "EEE MMM dd HH:mm:ss Z yyyy"
        }
      }
    },
    "aliases": {
      "mydata": { }
    }
  },
  "priority": 500,
  "composed_of": ["component_template1",
"runtime_component_template"],
  "version": 3,
  "_meta": {
    "description": "my custom"
  }
}
```

Цей лістинг представляє запит на створення індексного шаблону у Elasticsearch. Основна мета цього шаблону полягає в налаштуванні параметрів індексу для певного набору шаблонів індексів.

Опис запиту:

– PUT _index_template/template_1: Запит на створення індексного шаблону з назвою "template_1";

– "index_patterns": ["te*", "bar*"]: Визначення шаблонів індексів, які відповідатимуть цьому шаблону. У даному випадку, цей шаблон буде застосовуватися до всіх індексів, які починаються з "te" або "bar";

– "template": { ... }: Визначення налаштувань шаблону, які включають налаштування кількості шардів, включення вихідного документу (_source), визначення маппінгу полів, включаючи поле "host_name" як тип "keyword" і поле "created_at" як тип "date" зі специфічним форматом дати;

– "aliases": { "mydata": { } }: Визначення псевдоніму "mydata" для індексу, на який застосовується шаблон;

– "priority": 500: Визначення пріоритету шаблону, де більше значення пріоритету має вищий пріоритет;

– "composed_of": ["component_template1", "runtime_component_template"]: Визначення компонентних шаблонів, які використовуються для створення цього шаблону;

– "version": 3: Версія шаблону;

– "_meta": { "description": "my custom" }: Додаткові метадані, які можуть бути додані до шаблону для вказівки додаткової інформації.

Цей запит створює шаблон, який буде використовуватися для індексів, що відповідають шаблонам "te*" або "bar*". Він також встановлює налаштування для кількості шардів, визначає типи полів, додає псевдонім та вказує компонентні шаблони, які використовуються.

Також запишемо наш запит у консоль Dev Tools і очікуємо на отримання відповіді про успіх. На рисунку 3.20 успішний результат виконання.

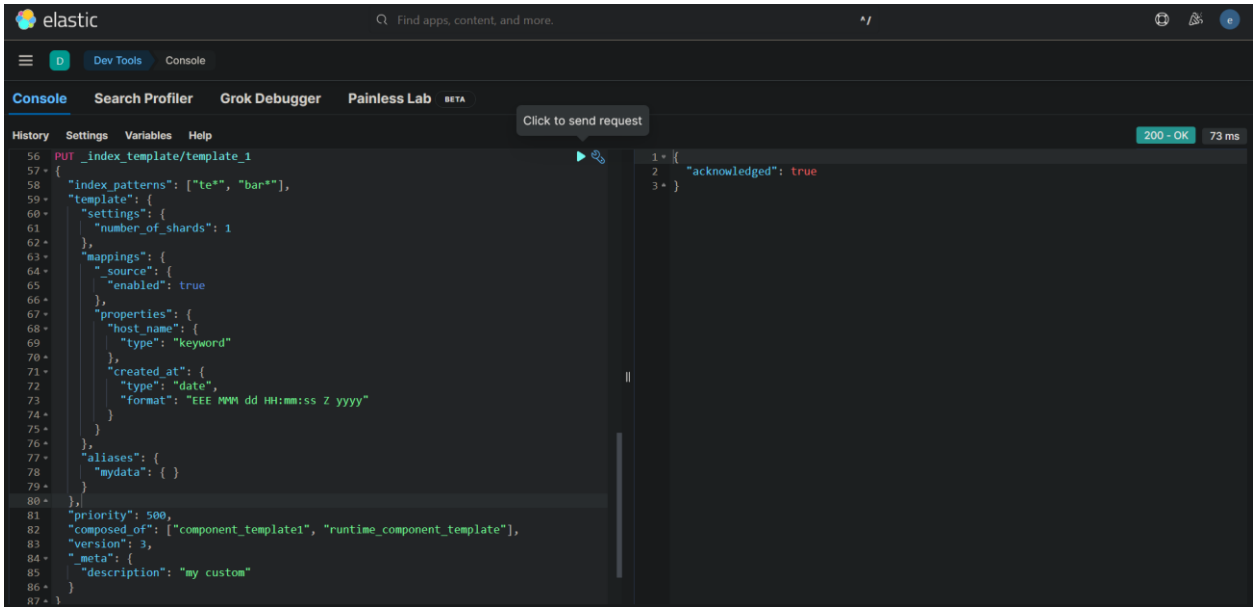


Рисунок 3.20 - Відповідь про успішне створення індексного шаблону

Після отримання відповіді "acknowledged: true" на запит PUT _index_template/template_1, ми можемо перевірити результати шаблону, виконавши наступний запит:

```
GET _index_template/template_1
```

Цей запит поверне нам інформацію про створений індексний шаблон з назвою "template_1" в форматі JSON. На рисунку 3.21 ми можемо перевірити наявність визначених параметрів, таких як шаблони індексів, налаштування, маппінги полів, псевдоніми і метадані (див. рис. 3.21).

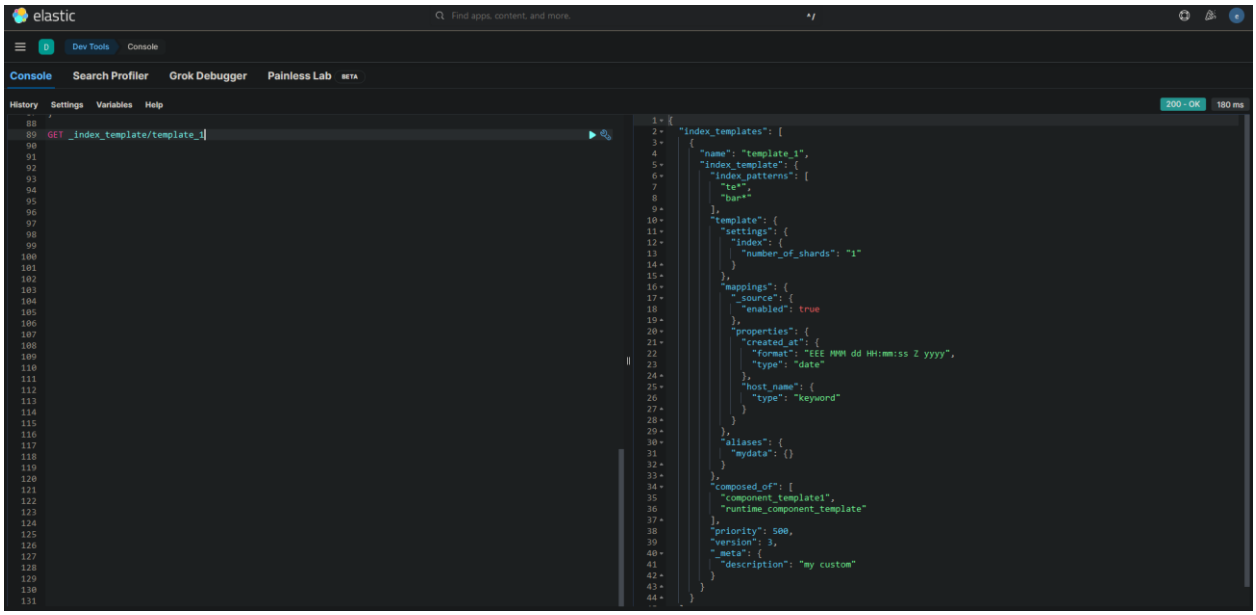


Рисунок 3.21 - Успішна перевірка результату шаблону

Наш розроблений шаблон індексу Elasticsearch по суті є декількома керуючими інструкціями, які автоматично застосовуються до нових індексів при їх створенні. Він дозволяє нам визначати налаштування та відображення для наших лог-файлів, що спрощує аналіз даних в майбутньому.

Шаблон включає універсальний тип даних "date" для поля "@timestamp", що забезпечує спільне розуміння формату дати/часу. Також він

включає універсальний тип даних "keyword" для поля "host_name", що полегшує пошук логів, пов'язаних із конкретним хостом.

Уявімо, що в нашому файлі з логами є поля з назвою "log_level". На нашу систему надходить безліч логів з різними рівнями - від "ERROR" до "INFO". Використовуючи шаблон, ми можемо заздалегідь вказати Elasticsearch, що поле "log_level" повинно бути типу "keyword". Це дозволяє нам легко фільтрувати або сортувати наші логи за рівнем.

Більше того, використання шаблону забезпечує, що весь наш журнал буде узгоджено структурований і готовий до аналізу. Завдяки шаблону, у нас є можливість виконувати складні запити, аналізувати тенденції у наших логах і визначати аномалії або важливі події.

3.2.2 Розробка фільтрів для Logstash

Розробка фільтрів для Logstash є одним із вирішальних етапів у створенні системи аналізу лог-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 структуризувати неструктуровані лог-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}
```

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 даєте частині тексту, яка збігається. Наприклад, число 3.44 може бути тривалістю події, тому ви може назвати це 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 на прикладі, як за допомогою синтаксису і семантики можна витягти корисні поля з такого вигаданого 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 (див. лістинг 3.4).

Лістинг 3.4 - Конфігурація Logstash-файлу «learn.conf» для обробки журналу HTTP

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

Запишемо наш запит у командний рядок і отримаємо результат (див. рис. 3.22).

Також можемо побачити результати у Kibana (див. рис. 3.23).

Рисунок 3.22 - Результат зчитування логу 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. 23).

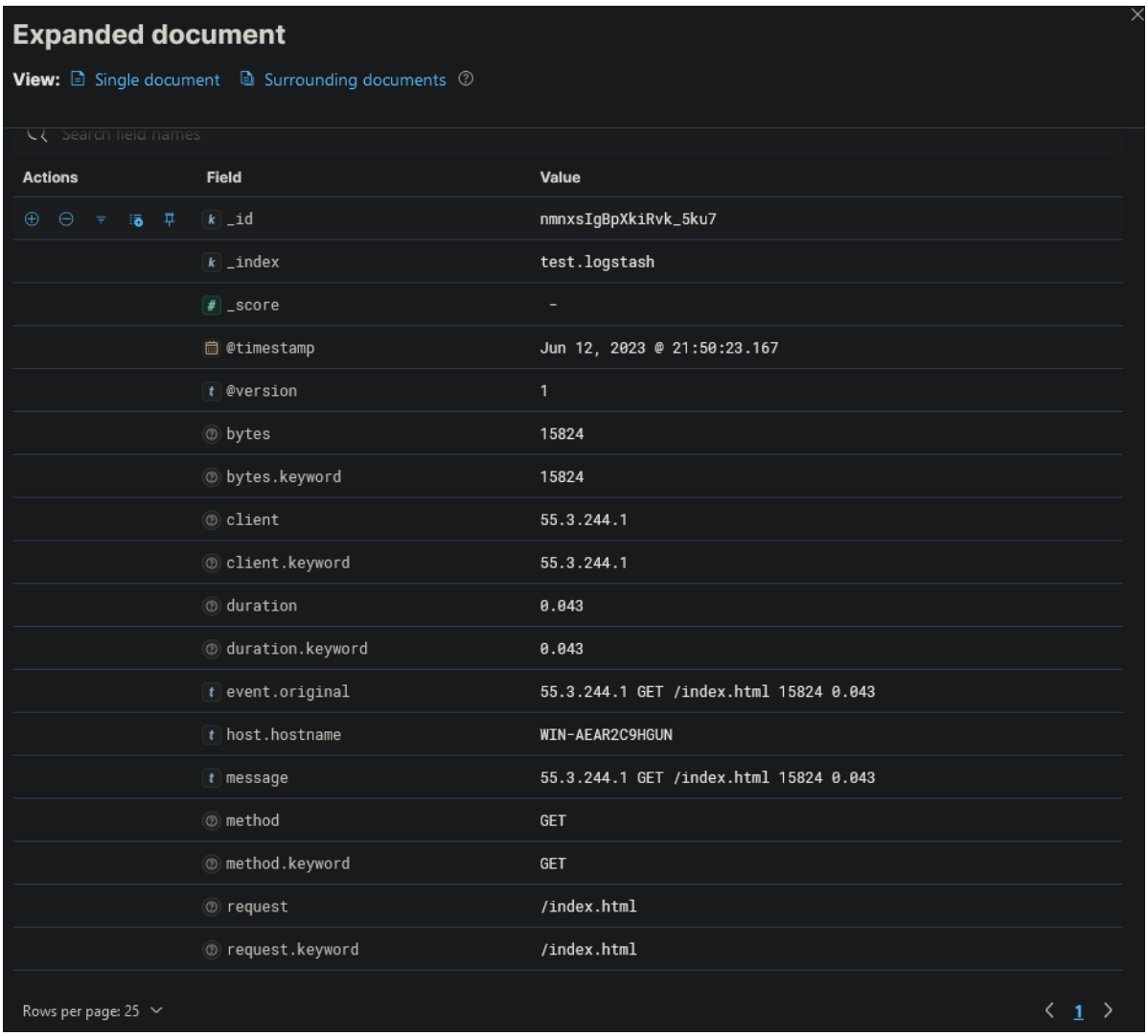


Рисунок 3.23 - Результати нашого запиту у Kibana

За допомогою Kibana ми тепер можемо встановлювати фільтри за різними полями, такими як "client", "method" або "request", і швидко отримувати підсумкові дані для конкретних сегментів вашої інформації. Це дозволяє нам швидко виділити найважливіші аспекти та глибше досліджувати ваші дані.

Крім 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. Це дозволяє нам точно налаштувати свої запити та отримувати результати, які відповідають вашим потребам.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ

4.1 Значення адаптації в трудовому процесі

Праця людини безпосередньо пов'язана із виробничим середовищем. Працівник може нормально здійснювати трудову діяльність лише тоді, колиумови зовнішнього середовища відповідають оптимальним. Якщо вони змінюються, стають несприятливими, то на протидію їм організм людини включає спеціальний механізм, який зберігає постійність внутрішнього середовища, або змінює його в межах допустимого. Такий механізм називається адаптацією. Адаптація є важливим засобом попередження травмування, виникнення нещасних випадків у трудовому процесі і відіграє значну роль в охороні праці.

Професійна адаптація - адаптація (пристосування) людини до нових для нього умов праці. Різновид професійної адаптації - виробнича адаптація (пристосування до умов, вимог, норм тощо конкретного виробництва, виробничого процесу).

Адаптація особистості до об'єктивних умов і вимог діяльності забезпечується такими методами: - вдосконалення або зміна в певних межах окремих властивостей; - формування стереотипів дій при незмінних особистісних якостях; - позитивна мотивація до праці; - вироблення індивідуального стилю діяльності. Ці методи, як правило, стосуються тих професій, які ставлять до людини відносні вимоги професійної придатності.

Суть механізму адаптації полягає у змінах меж чутливості аналізаторів, розширенні діапазону фізіологічних резервів організму та зміні в певних межах параметрів фізіологічних функцій. Завдяки фізіологічній адаптації фізичні табіохімічні параметри, які визначають життєдіяльність організму, змінюються у вузьких межах порівняно із значними змінами зовнішніх умов: підвищується стійкість організму до

холоду, тепла, недостачі кисню, змін барометричного тиску та інших факторів. Велике значення у фізіологічній адаптації має реактивність організму, його початковий функціональний стан, в залежності від якого змінюються і відповідні реакції організму на різні дії. Процес фізіологічної адаптації до незвичайних, екстремальних умов проходить декілька стадій, або фаз: спочатку переважають явища декомпенсації (порушення функцій), потім неповного пристосування (активний пошук організмом стійких станів, що відповідають новим умовам середовища) і, нарешті, фаза відносного стійкого пристосування.

Адаптація в трудовій діяльності поділяється на:

– фізіологічна адаптація - це сукупність фізіологічних реакцій, які є в основі пристосування організму до змін оточуючого середовища і направлені на збереження відносної постійності його внутрішнього середовища. Суть механізму адаптації полягає у змінах меж чутливості аналізаторів, розширенні діапазону фізіологічних резервів організму та зміні в певних межах параметрів фізіологічних функцій (підвищується стійкість організму до холоду, тепла, недостачі кисню, змін барометричного тиску);

– психічна адаптація - це процес встановлення оптимальної відповідності особистості до оточуючого середовища в процесі діяльності. Психічна адаптація в процесі праці залежить від психічних властивостей працівника, його психічного стану, психологічних реакцій на стреси, що виникають на роботі, кваліфікації та культури людини, особливостей професійної діяльності, конкурентних умов праці;

– соціальна адаптація - це пристосування працюючої людини до системи відносин у робочому колективі з його нормами, правилами, традиціями, ціннісними орієнтаціями. При несприятливому протіканні соціальної адаптації підвищується рівень стресу на роботі, наслідки якого позначаються на поведінці працівника та можуть призвести до між особових конфліктів, нещаснихвипадків;

– професійна адаптація - це адаптація до трудової діяльності з усіма її складовими і адаптація до робочого місця, знарядь та засобів праці, об'єктів та предметів праці, особливостей технологічного процесу, головних параметрів роботи. Професійна адаптація виражається у розвитку стійкого позитивного;

– ставлення працівника до своєї професії, певного рівня оволодіння ним специфічними навичками та уміннями у формуванні необхідних для якісного виконання роботи властивостей.

Кожен із розглянутих видів адаптації впливає на працездатність та здоров'япрацівника, формує у нього певний рівень чутливості та стійкості до психоемоційних перевантажень, внаслідок розвитку яких може істотно змінитися надійність професійної діяльності.

4.2 Вимоги ергономіки до організації робочого місця оператора ПК

Робоче місце - це зона простору, що оснащена необхідним устаткуванням, де відбувається трудова діяльність одного працівника чи групи працівників [13].

Раціональне планування робочого місця має забезпечувати: найкраще розміщення знарядь і предметів праці, не допускати загального дискомфорту, зменшувати втомлюваність працівника, підвищувати його продуктивність праці. Площа робочого місця має бути такою, щоб працівник не робив зайвих рухів і не відчував незручності під час виконання роботи. Важливо мати також можливість змінити робочу позу, тобто положення корпуса, рук, ніг. Проте доцільно виключати або мінімізувати всі фізіологічно неприродні і незручні положення тіла. Проведені дослідження показують, що при раціональній організації робочих місць продуктивність праці зростає на 15-25% [14].

Організація робочого місця користувача ПК має відповідати ергономічним вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце

для виконання робіт у положенні сидячи. Загальні ергономічні вимоги», ДСан ПіН 3.3.2.007-98, характеру та особливостям трудової діяльності.

Площа одного робочого місця користувача ПК повинна складати не менше 6 $м^2$, а об'єм - не менше 20 $м^3$. Конструкція робочого місця користувача ПК повинна відповідати сучасним вимогам ергономіки, характеру виконуваної роботи і забезпечити оптимальне розміщення на робочій поверхні документів та обладнання ПК (монітора, системного блоку, клавіатури, мишки та інших периферійних пристроїв. Монітор на робочому місці встановлюється так, щоб верхній край екрана знаходився на рівні очей.

Розташування монітора ПК має забезпечувати:

- безпечність роботи в цілому;

- зручність та ефективність зорової роботи з екраном в вертикальній площині під кутом 300 від лінії зору, площина екрана при цьому має бути перпендикулярною нормальній лінії зору користувача.

Клавіатура розміщується на поверхні столу або висувній полиці на відстані 100-300 мм від краю, ближчого до користувача. Кут нахилу клавіатури має бути в межах 5-150. Поверхня клавіатури повинна бути матовою з коефіцієнтом відбиття 0,4. клавіші клавіатури мають бути зручними в роботі і м'якими при натисканні (хід всіх клавіш має бути одинаковим з мінімальним опором натискання 0,25 Н та максимальним - не більше 1,5 Н) [15].

При розміщенні робочих місць з ПК слід дотримуватися вимог, зазначених в ДНАОП 0.00-1.31-99 «Про затвердження Правил охорони праці під час експлуатації електронно-обчислювальних машин»:

– робочі місця розміщуються на відстані не менше 1м від стін з світловими прорізями;

– відстань між бічними поверхнями моніторів ПК має бути не менше 1,2 м;

– відстань між тильною поверхнею монітора одного ПК та екраном монітора іншого ПК має бути не меншою 2,5 м.

Вимоги двох останніх пунктів враховуються також при розміщенні робочих місць з ПК в суміжних приміщеннях з урахуванням конструктивних особливостей стін та перегородок.

Загальні принципи організації робочого місця:

- на робочому місці не повинно бути нічого зайвого. Усі необхідні для роботи предмети мають бути поряд із працівником, але не заважати йому;

- ті предмети, якими користуються частіше, розташовуються ближче, ніж ті предмети, якими користуються рідше;

- предмети, які беруть лівою рукою, повинні бути зліва, а ті предмети, які беруть правою рукою - справа;

- якщо використовують обидві руки, то місце розташування пристосувань вибирається з урахуванням зручності захоплювання його двома руками;

- робоче місце не повинно бути захаращене.

Статичні напруження працівника в процесі праці пов'язані з підтриманням у нерухомому стані предметів і знарядь праці, а також підтриманням робочої пози.

Робоча поза - це основне положення працівника у просторі: зручна робоча поза має забезпечувати стійкість положення корпуса, ніг, рук, голови працівника під час роботи, мінімальні затрати енергії та максимальну результативність праці. Неправильна сидяча поза може викликати застій крові в ногах, а якщо виконується великий обсяг роботи для пальців рук - запалення суглобів.

Організація робочого місця користувача комп'ютера повинна забезпечувати відповідність усіх елементів робочого місця та їх взаємного розташування ергономічним вимогам (див. рис. 4.1).

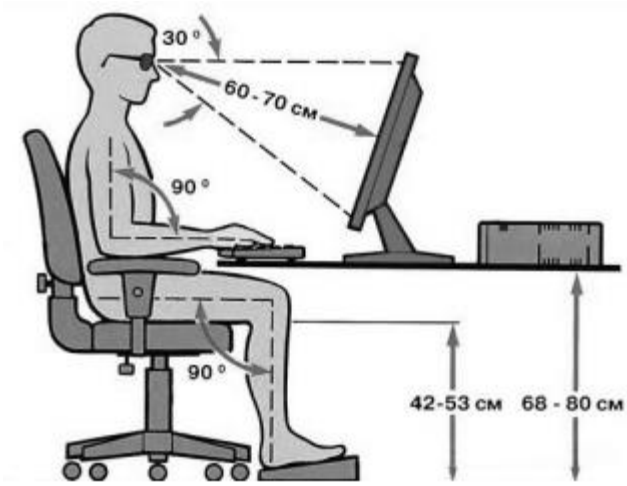


Рисунок 4.1 - Робоче місце і робоча поза користувача ПК

Найпоширенішими у процесі праці є пози сидячи і стоячи. Проектуючи робоче місце, потрібно враховувати, що при виконанні роботи з фізичним навантаженням бажана поза стоячи, а при малих зусиллях - сидячи.

Робоча поза стоячи втомлює людину більше, ніж сидяча. Вона вимагає на 10% більше енергії, спричиняє підвищення артеріального і венозного тиску крові, розширення вен на ногах, пошкодження ступень, викривлення хребта [15].

У розділі про вимоги ергономіки до організації робочого місця оператора ПК було розглянуто основні аспекти, які сприяють комфорту, безпеці та продуктивності користувача. Забезпечення правильного розташування монітора, клавіатури та миші допомагає уникнути напруження та травм, пов'язаних з поганою позицією тіла. Крісло з належним регулюванням та підтримкою для спини сприяє правильній підтримці тіла та зменшує ризик розвитку втоми та болю у спині.

Підсумовуючи розділ про вимоги ергономіки до організації робочого місця оператора ПК, важливо зазначити, що правильна організація робочого місця має велике значення для комфорту, безпеки та продуктивності користувача.

ВИСНОВКИ

У даній роботі було проведено детальний аналіз важливості та значення лог-файлів в контексті інформаційної безпеки. Метою роботи була розробка методу аналізу лог-файлів для виявлення потенційних загроз безпеки з використанням системи ELK.

Метою даного дипломного проєкту було дослідження можливостей використання стеку ELK для аналізу лог-файлів з метою виявлення потенційних загроз в безпеці. Дослідження обгрунтовує актуальність використання стеку ELK у сфері інформаційної безпеки та вплив цієї технології на якість та ефективність аналізу лог-файлів.

В ході роботи було розглянуто основні види лог-файлів та особливості їх зберігання та аналізу. Було проведено порівняльний аналіз систем журналювання подій Windows та Linux, що дозволило більш точно визначити їх відмінності та спільні риси.

Важливу частину роботи складає розгляд принципів збору та аналізу лог-файлів з використанням ELK. Було досліджено основні компоненти цієї системи - Elasticsearch, Logstash та Kibana, та розглянуто їх взаємодію та особливості використання.

Робота містить детальний аналіз загроз безпеки інформації, включаючи класифікацію активних та пасивних атак. Окрім того, було розглянуто, як система ELK може допомогти в аналізі та виявленні цих загроз.

Практична частина роботи включає встановлення Elasticsearch, Logstash і Kibana, а також налаштування цих компонентів для обробки лог-файлів. Особлива увага приділена розробці фільтрів для Logstash та шаблонів для Elasticsearch, що є ключовим елементом ефективного аналізу лог-файлів.

Таким чином, можна зробити висновок, що стек ELK є потужним інструментом для аналізу лог-файлів та виявлення потенційних загроз безпеки. Він дозволяє компаніям ефективно виявляти, відстежувати та

аналізувати можливі проблеми, що забезпечує високий рівень захисту від несанкціонованого доступу."

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Turnbull J. The Logstash Book / James Turnbull., 2013. - 262 с.
2. Gormley C., Tong Z. Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine. 2015. 724 p.
3. Srivastava A. Mastering Kibana 6.x: Visualize your Elastic Stack data with histograms, maps, charts, and graphs. Packt Publishing, 2018. 365 p.
4. Aggarwal M. Network Security with pfSense: Architect, deploy, and operate enterprise-grade firewalls. Packt Publishing, 2018. 152 p.
5. Elasticsearch Guide [8.8] | Elastic. *Elasticsearch Platform - Find real-time answers at scale | Elastic*. URL: https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html (date of access: 01.04.2023).
6. Event Logging (Event Logging) - Win32 apps. *Microsoft Learn: Build skills that open doors in your career*. URL: https://learn.microsoft.com/en-us/windows/win32/eventlog/event-logging (date of access: 05.05.2023).
7. Vulnerability detection - Use cases Â· Wazuh documentation. *Wazuh documentation*. URL: https://documentation.wazuh.com/current/getting-started/use-cases/vulnerability-detection.html (date of access: 11.06.2023).
8. Антон П. Советы и рекомендации по преобразованию неструктурированных данных из логов в ELK Stack используя GROK в LogStash. *Хабр*. URL: https://habr.com/ru/articles/509632/ (дата звернення: 14.06.2023).
9. Chuvakin A. Logging and Log Management: The Authoritative Guide to Understanding the Concepts Surrounding Logging and Log Management. Syngress, 2012. 460 p.
10. Kali Linux. Тестування на проникнення і безпеку / А. Замм та ін. ; ред. Н. Гринчик ; пер. з англ. А. Герасименко. 4-те вид. Packt Publishing, 2018. 528 с.

11. Kurose J., Ross K. Computer Networking: A Top-Down Approach. 7th ed. Pearson, 2016. 864 p.

12. Speciner M., Perlman R., Kaufman C. Network Security: Private Communications in a Public World. 2nd ed. Pearson, 2002. 1103 p.

13. Зеркалов Д.В. Безпека життєдіяльності та основи охорони праці. Навчальний посібник. К.: «Основа». 2016. - 267 с.

14. Яремко З. М. Безпека життєдіяльності: Навч. посіб. — Львів., 2005. - 301 с.

15. Желібо Є. П. Заверуха Н.М., Зацарний В.В. Безпека життєдіяльності. Навчальний посібник. - К.; Каравела, 2004. -328 с.

16. Morgan S. Cybercrime To Cost The World 8 Trillion Annually In 2023. Cybercrime Magazine. URL: https://cybersecurityventures.com/cybercrime-to-cost-the-world-8-trillion-annually-in-2023/ (date of access: 16.06.2023).

17. Morgan S. 2022 Official Cybercrime Report. 83 Main Street, 2nd Flr., Northport, N.Y. 11768 : eSentire, 2022. 32 p. URL: https://s3.ca-central-1.amazonaws.com/esentire-dot-com-assets/assets/resourcefiles/2022-Official-Cybercrime-Report.pdf.

18. Cyber Heatmap: CyberRisk Is Rising Across 70 Global. www.moodys.com. URL: https://www.moodys.com/research/Moodys-Cyber-Heatmap-Cyber-Risk-Is-Rising-Across-70-Global--PBC_1343021.

19. McLean E. Managing Cyber Risk. 2022 Cybercrime Report. Erin McLean, CMO & Tia Hopkins, Field CTO, eSentire. SoundCloud. URL: https://soundcloud.com/cybercrimemagazine/managing-cyber-risk-2022-cybercrime-report-erin-mclean-cmo-tia-hopkins-field-cto-esentire (date of access: 16.06.2023).

20. Morgan S. Top 10 Cybersecurity Predictions and Statistics For 2023. Cybercrime Magazine. URL: https://cybersecurityventures.com/top-5-cybersecurity-facts-figures-predictions-and-statistics-for-2021-to-2025/ (date of access: 16.06.2023).

21. Mehrotra K., Turton W. CNA Financial Paid $40 Million in Ransom After March Cyberattack. bloomberg.com. URL: https://www.bloomberg.com/news/articles/2021-05-20/cna-financial-paid-40-million-in-ransom-after-march-cyberattack#xj4y7vzkg?leadSource=uverify%20wall (date of access: 16.06.2023).

22. Internet Crime Report. FEDERAL BUREAU OF INVESTIGATION, 2022. 32 p. URL: https://www.ic3.gov/Media/PDF/AnnualReport/2022_IC3Report.pdf.

Додаток А - Лістинги повідомлень логів про перевищення швидкості передачі пакетів та про відкинуті пакети

Лістинг А1 - Приклад повідомлення, яке відправляється, коли швидкість передачі пакетів перевищує поріг виявлення.

```
Jun 16 23:08:46 172.30.107.11 action="Allow",hostname="lon-i5800-
1.pme.itc.f5net.com",bigip_mgmt_ip="172.30.107.11",context_name="/Common/www_10_103_2_80_80",date_time="Jun 17 2021 22:58:12",dest_ip="10.103.2.80",dest_port="80",device_product="DDoS Hybrid Defender",device_vendor="F5",device_version="15.1.2.1.0.317.10",dos_attack_event="Attack Sampled",dos_attack_id="550542726",dos_attack_name="TCP Push Flood",dos_packets_dropped="0",dos_packets_received="117",errdefs_msgno="23003138",errdefs_msg_name="Network DoS Event",flow_id="0000000000000000",severity="4",dos_mode="Enforced",dos_src="Volumetric, Per-SrcIP, VS-specific attack, metric:PPS",partition_name="Common",route_domain="0",source_ip="10.103.6.10",source_port="39219",vlan="/Common/vlan3006_client"
```

Лістинг А2 - Приклад повідомлення про відкинуті пакети:

```
Jun 17 23:05:03 172.30.107.11 action="Drop",hostname="lon-i5800-
1.pme.itc.f5net.com",bigip_mgmt_ip="172.30.107.11",context_name="Device",date_time="Jun 17 2021 22:54:29",dest_ip="10.103.2.80",dest_port="0",device_product="DDoS Hybrid Defender",device_vendor="F5",device_version="15.1.2.1.0.317.10",dos_attack_event="Attack Sampled",dos_attack_id="3221546531",dos_attack_name="Bad TCP flags (all cleared)",dos_packets_dropped="152224",dos_packets_received="152224",errdefs_msgno="23003138", errdefs_msg_name="Network DoS Event",flow_id="0000000000000000",severity="4",dos_mode="Enforced",dos_src="Volumetric, Aggregated across all SrcIP's, Device-Wide attack, metric:PPS",partition_name="Common",route_domain="0",source_ip="10.103.6.10",source_port="12826",vlan="/Common/vlan3006_client"
```

Додаток Б - Лістинг файлу learn.conf

Лістинг Б1 - Початковий файл конфігурації «learn.conf»

```
input {
  stdin { } # Вхідний потік для отримання даних з консолі
}
output {
  stdout {
    codec => rubydebug # Вивід даних у форматі Ruby-відладки
  }
  elasticsearch {
    hosts => ["http://localhost:9200"] # Адреси серверів Elasticsearch
    index => "test.logstash" # Індекс, у який будуть зберігатись дані
    user => "elastic" # Користувач Elasticsearch
    password => "3guzVh6WIwXCb*qVPQfp" # Пароль користувача Elasticsearch
  }
}
```

Додаток В - Лістинг прикладу створення компонентного шаблону

Лістинг В1 - Запити на створення компонентних шаблонів

```
PUT _component_template/component_template1
{
  "template": {
    "mappings": {
      "properties": {
        "@timestamp": {
          "type": "date"
        }
      }
    }
  }
}


PUT _component_template/runtime_component_template
{
  "template": {
    "mappings": {
      "runtime": {
        "day_of_week": {
          "type": "keyword",
          "script": {
            "source":
"emit(doc['@timestamp'].value.dayOfWeekEnum.getDisplayName(TextS
tyle.FULL, Locale.ROOT))"
          }
        }
      }
    }
  }
}
```

Додаток Г - Лістинг шаблону індексу

Лістинг Г1 - Конфігурація Logstash-файлу «learn.conf» для обробки журналу HTTP

```
UT _index_template/template_1
{
  "index_patterns": ["te*", "bar*"],
  "template": {
    "settings": {
      "number_of_shards": 1
    },
    "mappings": {
      "_source": {
        "enabled": true
      },
      "properties": {
        "host_name": {
          "type": "keyword"
        },
        "created_at": {
          "type": "date",
          "format": "EEE MMM dd HH:mm:ss Z yyyy"
        }
      }
    },
    "aliases": {
      "mydata": { }
    }
  },
  "priority": 500,
  "composed_of": ["component_template1",
"runtime_component_template"],
  "version": 3,
  "_meta": {
    "description": "my custom"
  }
}
```