https://orcid.org/0000-0001-5568-7629
https://orcid.org/0000-0002-5451-626X
https://orcid.org/0000-0003-4431-1989

УДК 621.37:637.142

В.А. ЛИТВИНОВ[*,], О.М. М'ЯКШИЛО[**], В.О. БРАЦЬКИЙ[**]**

ЗАДАЧА ЦЕНТРАЛІЗОВАНОГО УПРАВЛІННЯ ЖУРНАЛАМИ В МЕРЕЖІ СИТУАЦІЙНИХ ЦЕНТРІВ ОДВ І ПІДХОДИ ДО ПРОТОТИПУВАННЯ ЇЇ ПРОГРАМНИХ РІШЕНЬ

[*]Інститут проблем математичних машин і систем НАН України, м. Київ, Україна
[**]Національний університет харчових технологій, м. Київ, Україна

Анотація. *Журналізація подій у розподілених системах є одним із найважливіших факторів забезпечення належного моніторингу і управління IT-системами, використання інформації журналів є важливою сферою діяльності команд DevOps і DevSecOps, що забезпечують ефективну взаємодію розробників, тестувальників і IT-фахівців із безпеки. У статті розглядаються можливі підходи до прототипування рішень щодо реалізації централізованої системи LMS (Log Management System) у створюваній Національній мережі Ситуаційних центрів органів державної влади (СЦ ОДВ). У рамках першого підходу, що полягає у використанні готових ринкових продуктів, проведено огляд декларованих можливостей, переваг і недоліків популярних free open-source систем і окремих інструментів LMS (стек ELK, Graylog, Grafana Loki, Logstash, Fluentd, LOGalyze, Filebeat тощо). У контексті сформульованих базових вимог до централізованої LMS, з урахуванням наявного досвіду використання розглянутих інструментів, обґрунтовано доцільність вибору рішень серед двох комплексних повнофункціональних систем, а саме стеку ELK (комплексу Elasticsearch + Logstash + Kibana) та повного самодостатнього пакета Graylog. Розглядаються переваги і недоліки кожної системи, наводяться узагальнюючі дані щодо впровадження ELK – Graylog, їх використання та оцінки реальними користувачами, сформовані на основі матеріалів, представлених дослідницькою компанією Gartner. Прикладом можливої реалізації другого підходу до створення прототипу LMS-системи, що полягає у створенні власних засобів, слугує розроблена спеціалізована система діагностики помилок, зареєстрованих у лог-файлах. Описані структура системи, функції основних компонентів, результати апробації в корпоративній банківській мережі.*
Ключові слова: *мережа СЦ, журналізація подій, LMS, прототипування рішень.*

Abstract. *Event logging in distributed systems is one of the most important factors for ensuring proper monitoring and management of IT systems, and the use of log information is an important area of activity of DevOps and DevSecOps teams that ensure effective interaction between developers, testers, and IT security professionals. The article discusses some possible approaches to prototyping solutions for the implementation of a centralized LMS (Log Management System) in the National Network of Situation Centers of Public Authorities (SCPA). As part of the first approach, which consists in the use of ready-made market products, a review of the declared capabilities, advantages, and disadvantages of popular free open-source systems and individual LMS tools (ELK Stack, Graylog, Grafana Loki, Logstash, Fluentd, LOGalyze, Filebeat, etc.) is carried out. In the context of the formulated basic requirements for a centralized LMS, taking into account the existing experience of using the tools under consideration, the expediency of choosing solutions among two complex, full-featured systems, namely the ELK Stack (Elasticsearch + Logstash + Kibana complex) and the complete, self-sufficient Graylog package, is substantiated. The advantages and disadvantages of each system are considered, and the generalized data on the implementation of ELK – Graylog, their use and evaluation by real users, formed on the basis of materials presented by the research company Gartner, are provided. An example of the possible implementation of the second approach to creating a prototype of LMS, which consists in creating new tools, is the developed specialized system for diagnosing errors registered in log files. The structure of the system, the functions of the main components, and the results of testing in a corporate banking network are described.*
Keywords: *SC network, event logging, LMS, solution prototyping.*

DOI: 10.34121/1028-9763-2023-4-33-42

1. Вступ

Журналізація подій у розподілених системах, тобто їх реєстрація в журналах різних типів (журналах серверів, додатків, безпеки тощо), є одним із найважливіших факторів забезпечення належного моніторингу і управління ІТ-системами [1]. Моніторинг та аналіз подій у системі є необхідними для пошуку помилок у програмному забезпеченні, обладнанні або діях користувачів, несанкціонованих дій, відхилень від «нормальних» дій, а також, можливо, для розуміння дій користувачів у рамках нормальної поведінки. Аналіз журналів дозволяє відтворити ланцюжок подій, які передували і, можливо, стали причиною виникнення проблеми, і ефективно усунути її. Інформація щодо подій, які реєструються в системах, зберігається в файлах журналів (лог-файлах), переважно текстового формату, і використовуються спеціалістами різних профілів – операційного, безпекового, аналітичного та ін. Загалом методи і засоби управління журналами і обробки лог-файлів є невід'ємною частиною інфраструктури моніторингу і забезпечення працездатності будь-якої ІТ-системи.

Одними з найважливіших ІТ-систем національного масштабу є системи, що забезпечують підтримку діяльності органів державної влади (ОДВ). Необхідною передумовою успішного виконання їх функцій є створення національної мережі ситуаційно-кризових центрів (СКЦ) і забезпечення їх взаємодії. На жаль, до теперішнього часу проблема створення окремих відомчих СКЦ ОДВ, які належним чином відповідали б потрібним вимогам, запропонованим, зокрема, в [2], та їх інтеграції в рамках єдиної загальнодержавної мережі поки що далеко не вирішена. Прискорити процес її вирішення має розробка типових підходів, рішень і прототипів їх реалізації з подальшим відпрацюванням, розвитком і впровадженням [2].

Важливість моніторингу і журналізації подій у вузлах мережі СЦ та централізованої обробки журналів визначається двома основними факторами.

1. Послідовне масштабоване прототипування версій програмного забезпечення (ПЗ) в рамках еволюційних технологій проєктування пов'язане з підвищеним рівнем виникнення локальних і інтеграційних помилок ПЗ, користувачів тощо. В таких умовах використання інформації журналів, зокрема, системних журналів і журналів застосунків, є особливо важливою сферою діяльності команд DevOps (Development Operations), що забезпечують ефективну взаємодію розробників, тестувальників та інших ІТ-фахівців при впровадженні, супроводженні та модифікації програмного забезпечення. Інструменти управління журналами дозволяють визначити обставини і причину виникнення помилок і «нестиковок» застосунків, користувачів, компонентів системного середовища.

2. Мережа СЦ ОДВ вимагає підсиленої уваги до безпекових питань функціонування. В цьому аспекті слід відзначити, що інформація журналів є переважною частиною усієї сукупності даних, які необхідні для функціонування програмного забезпечення загальної системи управління безпекою SIEM (Security Information and Event Management) і фахівців DevSecOps, що забезпечують інтеграцію тестування безпеки в етапах розробки прототипів ПЗ [3, 4].

Отже, *мета статті* полягає в розгляді задачі реалізації управління обробкою лог-файлів у мережі СЦ і підходів до прототипування відповідних програмних рішень.

2. Централізоване управління журналами

Управління журналами визначається як безперервний процес централізованого збору, парсінгу, зберігання та аналізу журнальних даних, що генеруються всіма програмними застосунками та іншими компонентами ІТ-інфраструктури з метою надання корисної інформації для підтримки усунення несправностей, підвищення продуктивності і моніторингу безпеки [5–7].

Принциповою основою ефективного управління журналами є їх агрегація і централізація. Без об'єднання журналів розробники мають вручну впорядковувати, готувати та шукати дані журналів із численних джерел, щоб витягувати з них необхідну інформацію. Можливі різні рішення щодо централізації, починаючи із простої реплікації файлів журналів у деяке центральне місцезнаходження. Але кращим вибором для агрегації журналів у великих IT-системах стало створення своєрідного автоматизованого конвеєра, що забезпечує систематичний збір, відправку і обробку журналів на постійній основі (рис. 1).

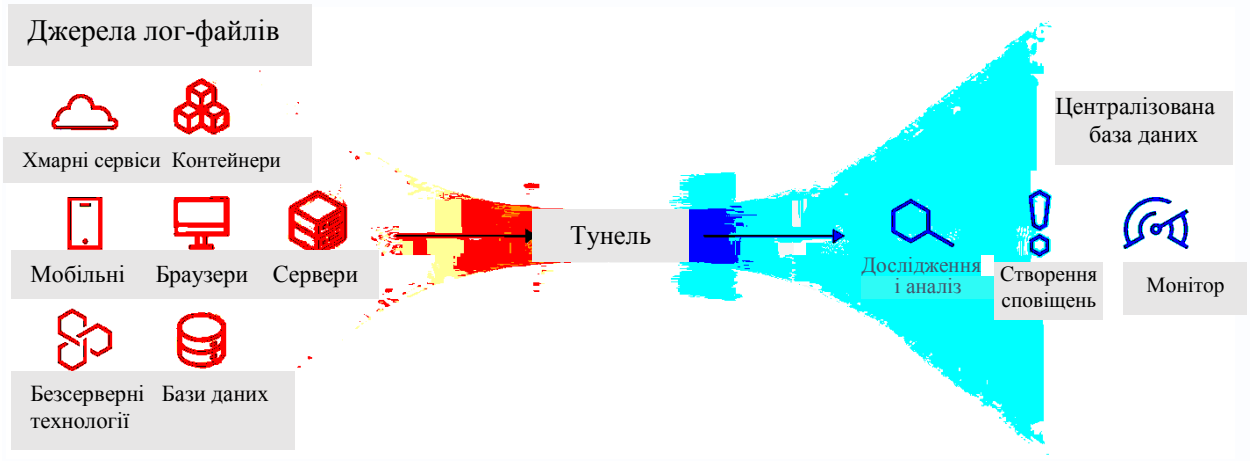

Рисунок 1 – Схема централізованої обробки журналів

Основні вимоги до функцій і відповідних інструментів платформ, що мають забезпечити ефективну агрегацію журналів та їх конвеєрну обробку:

– впорядкований збір даних із різних джерел із підтримкою всіх основних форматів журналів, зокрема, текстових файлів, JSON, XML тощо;

– парсінг (синтаксичний аналіз і обробка) журналів із метою впорядкування інформації – фільтрації за заданими критеріями, стандартне форматування і сортування даних;

– моніторинг даних – відстежування заданих подій, а також часу і обставин їх виникнення;

– аналіз даних – перегляд даних для упереджуваного виявлення помилок, загроз безпеки та інших проблем;

– сповіщення відносно здійснення заданих подій та/або виконання заданих критеріїв у зареєстрованих подіях;

– зберігання даних на протязі визначеного часу.

З урахуванням особливостей призначення і специфіки функціонування мережі СЦ ОДВ вважаються можливими два підходи до прототипізації відповідних рішень щодо програмної реалізації систем управління журналами (LMS-систем):

1) адаптація і використання готових free open-sourse систем;

2) розробка власних спеціалізованих програмних систем, орієнтованих безпосередньо на задачі мережі СЦ ОДВ.

Розглянемо наявні можливості реалізації відзначених підходів.

3. Огляд готових free open-source засобів управління журналами

З декількох десятків інструментів роботи з журналами, що пропонуються в теперішній час на ринку для реалізації LMS-систем [8–11], статус «free open-source» мають такі основні розглянуті засоби.

Стек ELK

Стек ELK являє собою интегрований комплекс инструментів агрегації журналів, що містить:

– масштабовану нереляційну (NoSQL) документоорієнтовану пошукову і аналітичну систему Elasticsearch, що забезпечує зберігання та високопродуктивний пошук даних. Вона є центральним елементом комплексу і об'єднує функції зберігання даних, індексації і повнотекстового пошуку за всіма полями у всіх документах із можливостями нечіткого пошуку, а також функції статистичної аналітики;

– інструмент збору та агрегації журналів Logstash, що забезпечує прийом логповідомлень із багатьох джерел, їх парсінг, приведення до єдиного формату і завантаження в Elasticsearch;

– інструмент користувацького інтерфейсу Kibana, що є функціональним шаром над Elasticsearch і забезпечує організацію пошуку даних, їх аналітики і побудову візуалізацій.

Схема комплексу ELK наведена на рис. 2.

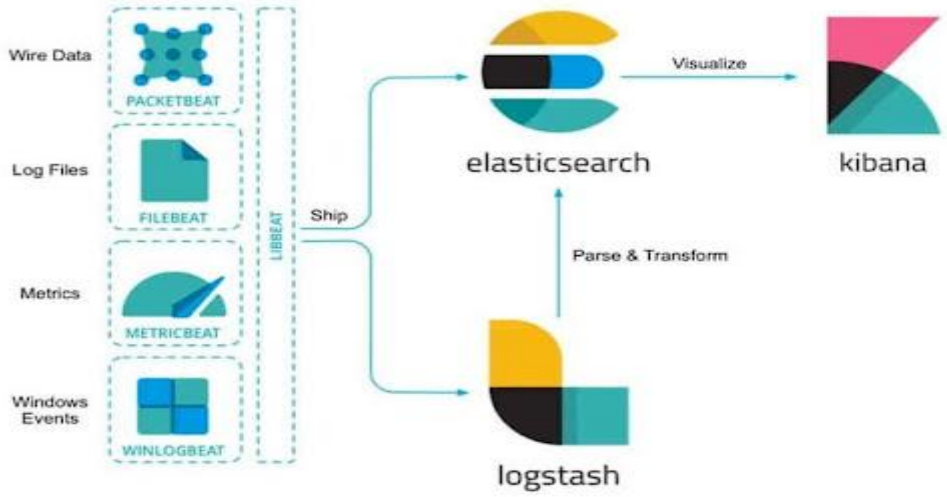

Рисунок 2 – Схема комплексу ELK

Elasticsearch і Kibana є прийнятими невід'ємними компонентами ELK, а для Logstash можливі й іноді використовуються деякі альтернативні варіанти. Відзначається, що в цілому ELK може надати достатньо ефективні відповідні можливості для системи SIEM [3].

Комплекс Graylog

Це повнофункціональний комплекс, що виконує всі базові функції аналогічно ELK. На відміну від ELK, що містить три окремих компоненти, Graylog побудований, як один цілісний пакет, який виконує збір, парсінг, індексацію, пошук і аналіз даних. Інтерфейс користувача Graylog функціонально аналогічний інтерфейсу Kibana.

Graylog складається з (рис. 3):

– вбудованої Elasticsearch, що використовується як інструмент індексації, зберігання, пошуку і аналізу даних;

– вбудованої СУБД MongoDB для зберігання налаштувань та іншої метаінформації;

– сервера Graylog, що реалізує функції агрегації логів (аналогічно Logstach), інтерфейсу користувача (аналогічно Kibana) і управління процесом функціонування комплексу.

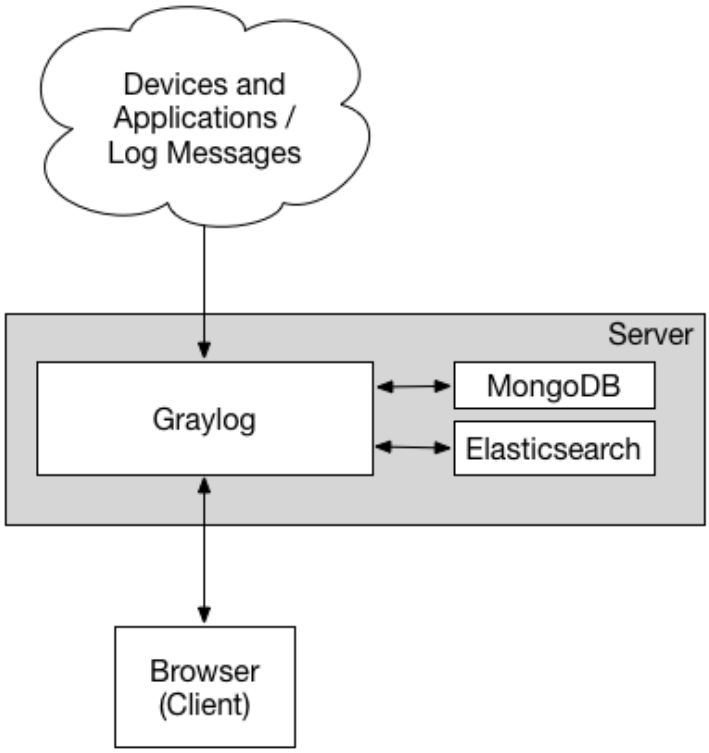

Рисунок 3 – Структурна схема комплексу Graylog

Комплекс Grafana Loki

Grafana Loki, відносно нещодавнє доповнення до комплексу інструментів централізованого управління журналами, є полегшеною альтернативою ELK і Graylog. Його «полегшеність» проявляється у двох основних аспектах:

– парсинг та приведення лог-повідомлень до загального формату виконується зовнішніми агентами, як, наприклад, Logstash;

– індексація лог-повідомлень виконується не по повному тексту, а тільки по метаданим, отже прямий повнотекстовий пошук по індексу неможливий і його треба здійснювати у два етапи. Рис. 4 ілюструє варіанти реалізації Grafana Loki.

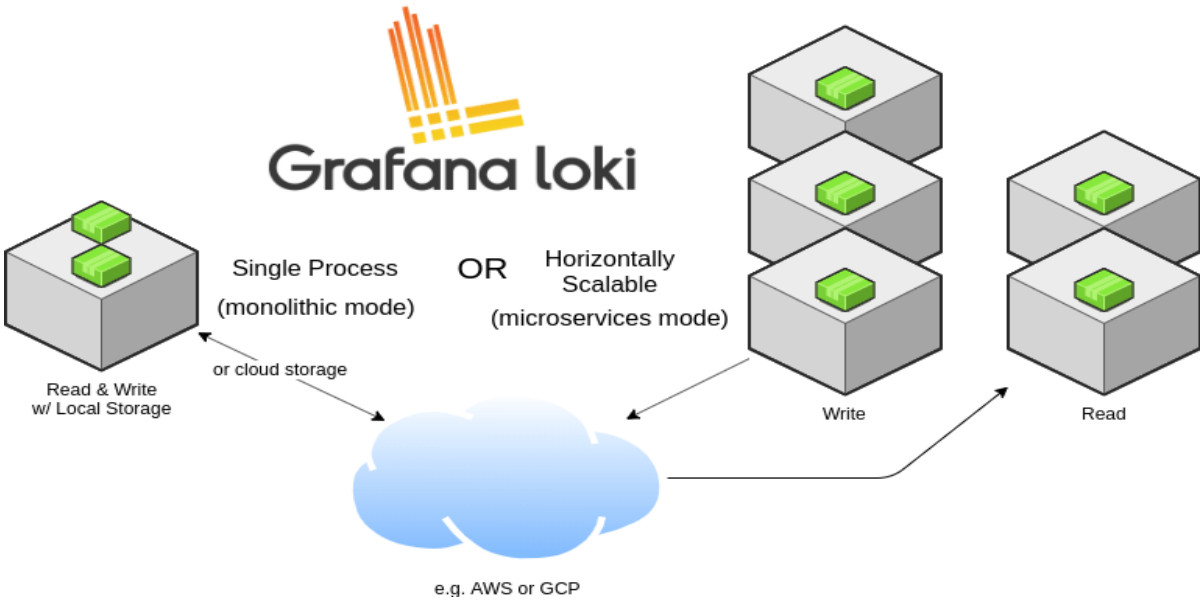

Рисунок 4 – Варіанти реалізації комплексу Grafana Loki

Logstash

Автономний інструмент збору, обробки та пересилання даних журналів є частиною ELK і може використовуватися самостійно. Постачається з великим набором плагінів, що забезпечує широкі можливості вводу, фільтрації і перетворення лог-повідомлень із різних джерел.

Fluentd

Як і Logstash, він виконує збір, аналіз, фільтрацію і перетворення даних із різних джерел та виведення даних у місця призначення, тобто він є альтернативою Logstash. Але на відміну від нього має кращу продуктивність та використання ресурсів (правда, ціною деякого обмеження функціонала).

LOGalize

Ще одна високопродуктивна, дещо спрощена, альтернатива Logstash.

Logagent

Легкий високопродуктивний інструмент доставки журналів – дуже спрощена альтернатива Logstash.

Filebeat

Легкий і простий у користуванні засіб доставки журналів, розроблений у доповнення до logstash, як його попередній агент (рис. 5). Але може передавати дані і напряму в Elasticsearch.

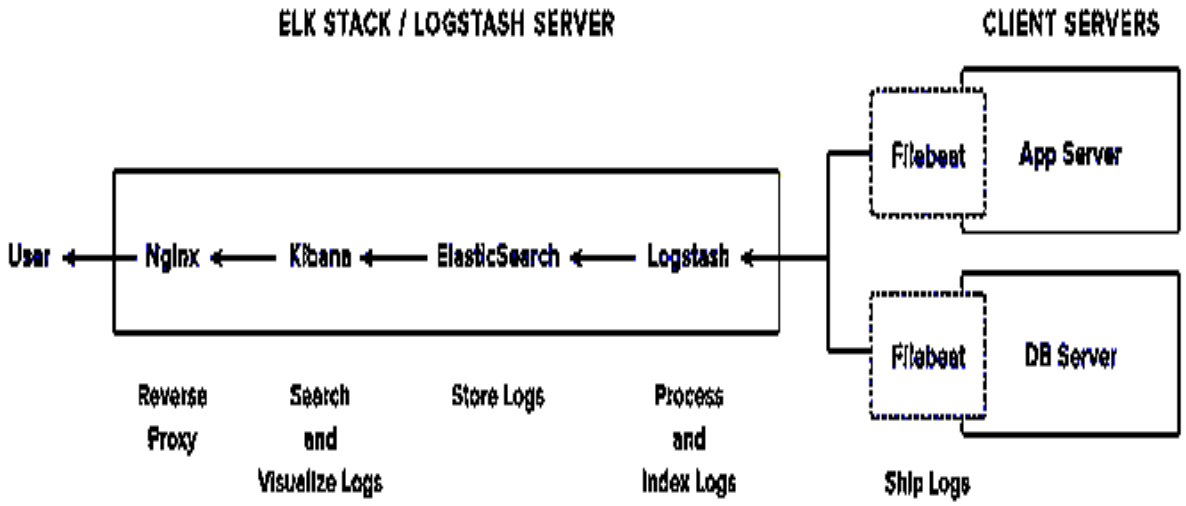

Рисунок 5 – Застосування засобу доставки журналів *Filebeat*

Rsyslog ma syslog-ng

Універсальні системні інструменти супроводження журналів, що можуть считувати дані з декількох джерел, попередньо обробляти і відправляти в місця призначення.

Оцінюючи можливості і перспективи використання розглянутих інструментів, слід прийняти до уваги необхідність виконання комплексу базових функцій та наявність відповідних інструментів, необхідних, як було наведено вище, для централізованих систем управління журналами. Орієнтація на зазначені вимоги, з урахуванням наявного досвіду використання відповідних систем, дає підстави для звуження області прийнятних рішень до двох комплексних повнофункціональних інтегрованих систем. Це стек ELK та комплекс Graylog. З точки зору базової функціональності, обидва рішення цілком співставні,

кожна з відзначених систем має свої переваги і недоліки стосовно цільового застосування за критеріями повноти функціонала, масштабування, хмарної реалізації, складності опанування і розвитку тощо.

Ключова відмінність між системами, що розглядаються, полягає в їх загальному призначенні. Комплекс Graylog призначений для прийому, зберігання, пошуку і обробки виключно даних журналів, а стек ELK – для аналізу великих даних взагалі. У зв'язку з цим, ELK вважається більш повнофункціональним рішенням щодо збору і аналізу даних (зокрема, і даних журналів). Тобто ELK – це багатоцільовий, модульний інструмент із можливостями адаптації та масштабування. ELK, на відміну від Graylog, підтримує велику кількість типів даних (більше 25 відповідних «офіційних» власних плагінів і більше 100 автономних пакетів різного походження), має безліч фільтрів обробки (парсингу) вхідних даних (близько 30 плагінів), розвинуті засоби візуалізації даних (Kibana) і засоби збору даних (тракт «системні журнали – Filebeat – Logstash).

У той же час ELK:

– більш складна в налаштуваннях і опануванні;

– має менш інтерактивний і зручний інтерфейс;

– у безкоштовній версії відсутня можливість сповіщень щодо заданих подій (можливе використання сторонніх плагінів для цього);

– потребує більше ресурсів і менш продуктивна при високому трафіку.

Обґрунтування рішення щодо вибору «ELK – Graylog» має високий попит при створенні систем моніторингу IT-інфраструктури, і цілеспрямоване порівняння цих інструментів під різними кутами зору, за різними критеріями, наведено в досить значній кількості публікацій, наприклад, [12–14].

Показовими є такі узагальнюючі дані (табл. 1) щодо впровадження ELK – Graylog, використання та оцінки реальними користувачами, сформовані на основі матеріалів, представлених дослідницькою компанією Gartner [15].

Таблиця 1 – Порівняння систем ELK і Graylog

КРИТЕРІЇ	ELK	Graylog
Користувачі, кількість представлених оцінок	342	149
З них		
з оцінкою 5	47%	40%
з оцінкою 4	46%	51%
з оцінкою 3	6%	9%
з оцінкою менше 3	0%	0%
Користувачі впевнено рекомендують для використання	86%	86%
Можливості, загальна (безпосередня) оцінка	4,6	4,4
Оцінка за окремими критеріями		
Моніторинг у реальному часі	4,6	4,5
Аналітика загроз	4,0	3,9
Профілювання поведінки	4,1	3,8
Моніторинг даних і користувачів	4,5	4,2
Аналітика	4,4	4,2
Управління журналами і звітність	4,6	4,5
Простота розгортання/підтримки	4,2	4,1
Інтеграція і розгортання, загальна оцінка	4,4	4,4
Оцінка за окремими критеріями		
Простота розгортання	4,2	4,2
Якість навчання кінцевих користувачів	4,2	3,9

Продовж. табл. 1

Простота інтеграції з використанням стандартних API та інструментів	4,4	4,4
Доступність сторонніх ресурсів	4,2	4,2
Підтримка, загальна оцінка	4,3	4,3
Оцінка за окремими критеріями		
Своєчасність реакції служби підтримки	4,4	4,3
Якість підтримки	4,3	4,2
Якість співтовариства користувачів	4,4	4,4

Із наведених даних у табл. 1 видно, що оцінки реальними користувачами рішень щодо використання ELK і Graylog примірно рівноцінні, з деякою стійкою перевагою ELK. У ELK також значно більше користувачів (судячи, зокрема, по кількості відгуків з оцінками) і, відповідно, досвід використання.

4. Діагностична система аналізу лог-файлів

Обнадійливим прикладом можливої реалізації другого підходу до створення прототипу LMS-системи слугує спеціалізована система діагностики помилок, зареєстрованих у лог-файлах [16]. Структурна схема системи наведена на рис. 6.

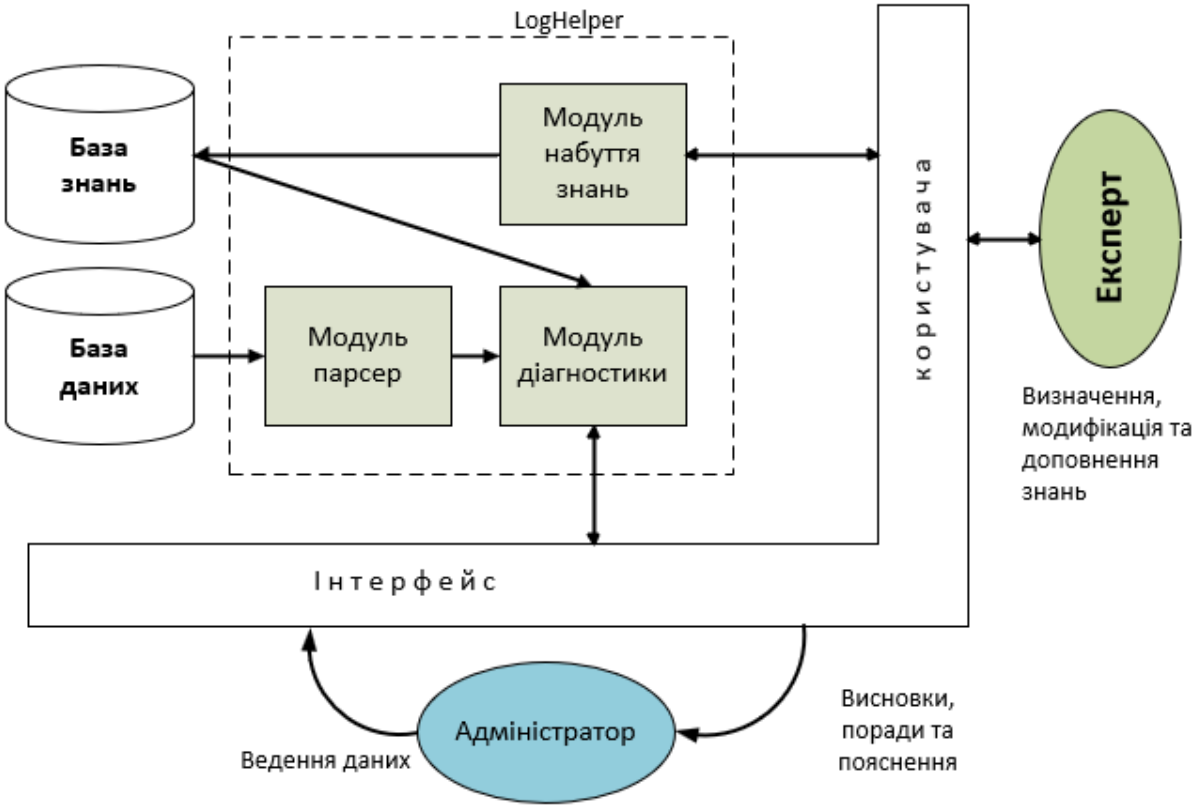

Рисунок 6 – Структурна схема діагностичної системи аналізу log-файлів

Парсер виконує синтаксичний аналіз файлів журналів (лог-файлів), фільтрації за заданими критеріями і формування лог-об'єктів стандартного формату, придатних для зберігання в базі даних, і подальшого змістовного аналізу.

Модуль діагностики виконує змістовний аналіз лог-об'єктів із метою ідентифікації заданих помилок, визначення їх можливих причин та шляхів усунення в подальшому з наданням відповідних рекомендацій користувачу.

Модуль набуття знань накопичує інформацію про результати ідентифікації помилок і їх можливих та реальних (фактичних) причин із метою подальшого використання у процесі функціонування модуля діагностики.

База даних і база знань, що містять лог-файли, лог-об'єкти та інформацію про результати роботи модуля діагностики, реалізовані на основі NoSQL СУБД MongoDB.

LogHelper забезпечує управління системою – сумісну роботу модулів та їх взаємодію з базою даних і базою знань у середовищі MongoDB.

Роботу діагностичної системи апробовано і протестовано в корпоративній банківській мережі. Результати апробації і впровадження показали дієвість системи та її ефективність для аналізу функціонування клієнтських додатків, а також визначили шляхи її вдосконалення та подальшого розвитку функціонала.

5. Висновки

Таким чином, нами обґрунтовано доцільність і «рамково» розглянуті наявні інструментальні можливості програмної реалізації двох принципових підходів до прототипізації рішень зі створення централізованої LMS-системи в мережі СЦ ОДВ. Для першого підходу (адаптація і використання готових free open-sourse інструментів) вважається доцільною орієнтація на застосування комплексних повнофункціональних популярних систем ELK або Graylog, для яких є достатньо великий досвід експлуатації і якісні співтовариства користувачів. Для другого підходу (створення власних інструментів) розглянута розроблена діагностична система аналізу лог-файлів, що пройшла успішну апробацію в корпоративній банківській мережі. Запропоновані підходи і рішення, які визначають властивості діагностичної системи, створюють, на думку авторів, підстави для обґрунтування можливості її використання після відповідної доробки як початковий прототип LMS-систем для мережі СЦ ОДВ.

Для можливого подальшого розвитку досліджень і дослідно-конструкторських робіт в одному з розглянутих напрямів необхідні загальні політично-технічні рішення відносно вибору підходів.

СПИСОК ДЖЕРЕЛ

1. What Is Log Analysis: A Complete Guide to Logging Use Cases & Best Practices You Need to Know About. URL: https://sematext.com/blog/log-analysis/.
2. Гречанінов В.Ф., Кузьменко Г.Є., Лопушанський А.В., Морозов А.О. Мережа ситуаційних центрів органів державної влади – базис для підвищення ефективності їх діяльності (взаємодії). *Математичні машини і системи*. 2018. № 3. С. 32–39.
3. SIEM vs Log Management. URL: https://www.motadata.com/blog/siem-vs-log-management/.
4. SIEM vs Log Management. URL: https://www.crowdstrike.com/cybersecurity-101/observability/siem-vs-log-management/.
5. What is Log Management. URL: https://sematext.com/guides/log-management/.
6. What is Log Management. The importance of logging and best practices. URL: https://www.crowdstrike.com/cybersecurity-101/observability/log-management/.
7. Log Aggregation Overview. URL: https://www/datadoghq.com/knowledge-center/log-management/.
8. Best Log Management Tools for Monitoring, Analytics & More: Pros & Cons Comparison. 2023. URL: https://sematext.com/blog/best-log-management-tools/.
9. Best Log Management Tools. URL: https://www.comparitech.com/net-admin/log-management-tools/.
10. Best Log Management Tools: Useful Tools for Log Management, Monitoring, Analytics, and More. URL: https://stackify.com/best-log-management-tools/.
11. Top 8 BEST Log Management Software | Log Analysis Tool Review. 2023. URL: https://www.softwaretestinghelp.com/log-management-software/.

12. Log Management: Graylog vs ELK. URL: https://expertise.jetruby.com/log-management-graylog-vs-elk-d6e8f0492323.
13. Graylog vs ELK. URL: https://www.geeksforgeeks.org/graylog-vs-elk/.
14. ELK vs Grylog: Log Management Comparison. URL: https://www.atatus.com/blog/elk-vs-graylog-log-management-comparison/#:~:text=Key%20Difference%20Between%20Graylog%20vs%20ELK,-Let's%20look%20at&text=Graylog%20is%20primarily%20for%20log,up%20separately%20from%20the%20others/.
15. Elastic vs Graylog. URL: https://www.gartner.com/reviews/market/security-information-event-management/compare/elasticsearch-vs-graylog.
16. Брацький В.О., М'якшило О.М., Литвинов В.А. Діагностична система аналізу log-файлів із віддалених вузлів обробки даних. *Математичні машини і системи*. 2022. № 1. С 62–73.

Стаття надійшла до редакції 26.09.2023