ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА СИСТЕМ ТА ТЕХНОЛОГІЙ КІБЕРБЕЗПЕКИ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Засоби виявлення вразливостей в інформаційній системі організаціїї на базі Splunk»

зі спеціальності *125 Кібербезпека*
(код, найменування спеціальності)

освітньо-професійної програми *Інформаційна та кібернетична безпека*
(назва програми)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Данило НАУМЕНКО
(підпис)

Виконав: здобувач вищої освіти групи БСД-42
НАУМЕНКО Данило
(прізвище, ім'я)

Керівник БОЙКО Анна
(науковий ступінь, вчене звання, прізвище, ім'я)

Рецензент _____
(науковий ступінь, вчене звання, прізвище, ім'я)

Київ 2025

ВСТУП

Актуальність дослідження. В умовах постійного зростання кількості кіберзагроз і технологічної складності інформаційних систем, виявлення вразливостей набуває критичного значення для забезпечення безпеки цифрової інфраструктури організацій. Вразливості можуть існувати на рівні програмного забезпечення, мережевих протоколів, конфігурацій, архітектури або людського фактора, і саме їх своєчасне виявлення дозволяє запобігти реалізації широкого спектра атак, зокрема експлуатації нульового дня, несанкціонованого доступу та витоку даних.

Особливої ваги набувають системи класу SIEM (Security Information and Event Management), які поєднують можливості збору, зберігання, аналітики та кореляції подій безпеки в режимі реального часу. Серед таких систем Splunk посідає провідне місце завдяки своїй модульній архітектурі, гнучкості налаштувань, масштабованості та інтеграції з великою кількістю джерел подій. Splunk дозволяє реалізовувати повноцінний процес виявлення вразливостей — від збору телеметрії до автоматизованого реагування.

Платформа надає змогу впроваджувати складні правила кореляції, створювати інтерактивні дашборди, здійснювати оповіщення, проводити поведінковий аналіз, інтегруватися з threat intelligence-фідами, а також будувати сценарії реагування через SOAR-рішення. Усе це дозволяє не лише виявляти окремі інциденти, а й фіксувати складні ланцюги атак відповідно до моделей MITRE ATT&CK.

Вищевикладене визначає актуальність теми даної кваліфікаційної роботи, основною метою якої є дослідження можливостей платформи Splunk у контексті виявлення вразливостей та розробка практичних рекомендацій щодо її використання в інформаційній системі організації.

Об’єкт дослідження — інформаційна система організації.

Предмет дослідження — методи і засоби виявлення вразливостей інформаційної системи на базі платформи Splunk.

Мета роботи — розробити рекомендації щодо застосування Splunk як інструменту виявлення вразливостей та загроз в IT-інфраструктурі організації.

Наукові завдання:

- дослідити сутність проблеми вразливостей в інформаційних системах;
- класифікувати типи вразливостей і методи їх виявлення;
- проаналізувати сучасні технології виявлення вразливостей;
- вивчити архітектуру та функціональні можливості Splunk;
- розробити приклади механізмів виявлення загроз, кореляційних правил та дашбордів у Splunk;
- сформулювати рекомендації щодо практичного впровадження Splunk у систему кіберзахисту.

Методи дослідження — аналіз літературних джерел, нормативних документів, технічної документації Splunk; порівняльний аналіз інструментів виявлення вразливостей; моделювання інфраструктури та експериментальна побудова сценаріїв виявлення загроз.

Практичне значення одержаних результатів полягає в тому, що сформульовані рекомендації можуть бути використані фахівцями з інформаційної безпеки для впровадження Splunk як SIEM-рішення в корпоративному середовищі, побудови ефективної логіки виявлення та реагування на інциденти, а також підвищення загального рівня кіберстійкості організації.

1 ТЕОРЕТИЧНІ ОСНОВИ ВИЯВЛЕННЯ ВРАЗЛИВОСТЕЙ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

1.1. Поняття вразливостей та їх класифікація

Вразливості інформаційних систем становлять собою сукупність недоліків у програмному забезпеченні, апаратному забезпеченні, мережевій архітектурі або організаційних процесах, які можуть бути використані для компрометації конфіденційності, цілісності або доступності інформації. У нормативно-правовому полі термін "вразливість" визначається як слабке місце, яке може бути цілеспрямовано використане загрозою задля нанесення шкоди IT-інфраструктурі або даним. Саме наявність вразливостей є передумовою більшості кіберінцидентів [1].

Класифікація вразливостей є важливим етапом управління ризиками та дозволяє систематизувати загрози відповідно до джерела, характеру впливу, рівня критичності та способів усунення [2].

Основними критеріями класифікації виступають:

- джерело виникнення (людський фактор, програмна помилка, помилка конфігурації);
- рівень реалізації (мережевий, прикладний, системний);
- спосіб експлуатації (віддалене/локальне використання);
- критичність (за шкалою CVSS або NVD).

Нижче наведено таблицю, яка демонструє основні типи вразливостей інформаційних систем (табл. 1.1):

Таблиця 1.1

Класифікація вразливостей в інформаційних системах

Категорія вразливості	**Приклади**	**Механізм впливу**	**Потенційні наслідки**
Програмні	Buffer Overflow, Use-After-Free	Порушення роботи пам’яті	Виконання довільного коду, DoS
Валідаційні	SQL Injection, XSS, CSRF	Некоректна обробка вхідних даних	Неавторизований доступ, виконання скриптів
Конфігураційні	Default Passwords, Open Ports	Неналежне налаштування	Повний контроль над системою
Автентифікації та доступу	Weak MFA, Insecure Sessions	Обхід механізмів контролю доступу	Компрометація облікових записів
Криптографічні	MD5, SHA-1, TLS misconfig	Використання слабких або застарілих алгоритмів	Розшифрування даних, втрати конфіденційності
Архітектурні	Відсутність сегментації, Zero Trust	Прорахунки в дизайні системи	Повне проникнення зловмисника в мережу
Людського фактора	Фішинг, соціальна інженерія	Маніпуляція користувачами	Розкриття облікових даних, початковий доступ
Мережеві	DoS, BGP hijack, MITM	Використання вразливостей мережевих протоколів	Відмова в обслуговуванні, перехоплення даних
Ланцюга постачання	SolarWinds, Log4Shell	Вплив через стороннє ПЗ або бібліотеки	Масова компрометація клієнтів

Застосування наведеної класифікації дозволяє не лише впорядковувати відомі загрози, а й прогнозувати нові сценарії атак, будувати ефективні політики реагування та розробляти процедури усунення вразливостей. Враховуючи динамічний характер середовища загроз, важливим є постійний моніторинг, застосування стандартів та інтеграція засобів автоматичного виявлення (наприклад, SIEM/EDR/XDR).

1.2. Методи і засоби виявлення вразливостей

Виявлення вразливостей є невід'ємною складовою системного управління інформаційною безпекою. Цей процес полягає в ідентифікації недоліків, слабких місць або помилок в архітектурі, програмному забезпеченні чи організаційних політиках, які можуть бути використані для реалізації загроз. Відповідно до підходу, викладеного в NIST SP 800-115, виявлення вразливостей – це один із базових етапів оцінки безпеки інформаційних систем, спрямований на визначення потенційних точок проникнення або компрометації активів організації [3].

Залежно від часу застосування, контексту використання та типу ресурсів, що аналізуються, методи виявлення вразливостей поділяються на проактивні та реактивні. Проактивні методи орієнтовані на випереджальне виявлення вразливостей ще до того, як вони будуть використані зловмисниками. Вони застосовуються під час розробки, впровадження або планових аудитів інформаційних систем. Реактивні методи, своєю чергою, спрямовані на виявлення вже активних або нещодавно реалізованих вразливостей шляхом аналізу цифрових слідів, логів або поведінкових аномалій.

З іншого боку, за ступенем автоматизації виділяють ручні та автоматизовані методи. Ручні передбачають участь експерта (наприклад, під час пентесту або перевірки коду), в той час як автоматизовані базуються на використанні спеціалізованих програмних засобів (сканери вразливостей, SIEM, EDR тощо), що

значно прискорює процес, особливо у великих середовищах. Варто зазначити, що навіть найефективніші засоби потребують грамотної конфігурації та інтерпретації результатів, отже людський фактор залишається критичним на всіх етапах.

Класифікацію методів виявлення також варто розглядати через призму застосовуваних джерел даних: деякі методи працюють зі статичною інформацією (наприклад, аналіз вихідного коду чи конфігурацій), інші – з динамічними даними (лог-файли, трафік, поведінкові метрики). Згідно з ISO/IEC 27002:2022, обидва підходи мають бути інтегрованими у політику безпеки організації з урахуванням актуальності ризиків та критичності активів.

У сучасних умовах значної складності IT-середовищ та збільшення обсягів даних, ключовим фактором стає ефективне поєднання різних методів виявлення вразливостей у межах єдиного процесу. Саме такий інтегрований підхід, відомий як defense-in-depth, є основою побудови стійких до атак інформаційних систем і дозволяє вчасно виявляти та нейтралізувати загрози ще на ранніх стадіях.

Проактивні методи виявлення вразливостей застосовуються до моменту реалізації загроз і мають на меті попередити виникнення інцидентів шляхом своєчасного виявлення та усунення слабких місць. Вони є ключовими в контексті впровадження моделі “shift left” у кібербезпеці, яка передбачає перенесення практик захисту на ранні етапи життєвого циклу IT-систем.

Одним із поширених методів є сканування вразливостей. Цей підхід ґрунтується на автоматичному аналізі конфігурацій, відкритих портів, версій програмного забезпечення та інших характеристик системи з метою пошуку відповідностей (співпадінь) у відомих базах даних вразливостей, таких як CVE (Common Vulnerabilities and Exposures) або NVD (National Vulnerability Database). Типовими представниками програмних засобів такого класу є Nessus, Qualys та OpenVAS. Їх основними перевагами є швидкість, системність і висока покритість, а недоліками – ймовірність хибнопозитивних спрацювань і залежність від актуальності баз даних

Аналіз вихідного коду, що реалізується у двох варіантах – статичному (SAST) та динамічному (DAST). Перший здійснюється без запуску програми й дозволяє виявити синтаксичні, логічні помилки, неправильне використання API або вразливі бібліотеки. Динамічний аналіз, навпаки, передбачає виконання програмного забезпечення в контрольованому середовищі та аналіз його поведінки під час взаємодії з тестовими запитами. Такий підхід є ефективним у виявленні XSS, SQL-ін'єкцій, небезпечної десеріалізації тощо. Аналіз коду особливо актуальний для організацій, які розробляють власні IT-рішення та прагнуть інтегрувати безпеку в процес DevOps (так званий DevSecOps-підхід).

Іншим інструментом проактивного підходу є моделювання загроз, яке полягає в побудові карти потенційних атак на систему на основі її архітектури, логіки взаємодії компонентів і попередньо виявлених слабких місць. Для реалізації такого підходу використовуються методики STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege), а також DREAD (Damage, Reproducibility, Exploitability, Affected Users, Discoverability) (рис. 1.1) [4].

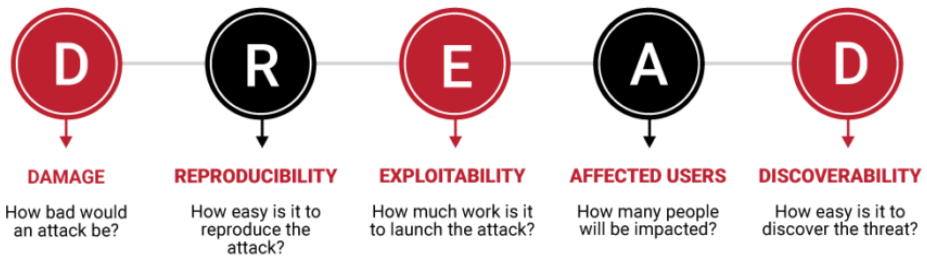


Рис. 1.1. Модель загроз DREAD

Таке моделювання загроз дозволяє їх пріоритезувати та зорієнтувати ресурси на захист найбільш критичних активів.

До проактивних методів відносять тестування на проникнення (penetration testing), яке полягає в імітації дій потенційного зловмисника з метою виявлення реальних сценаріїв експлуатації вразливостей. Пентест проводиться як вручну (з використанням фреймворків типу Metasploit, Cobalt Strike), так і частково автоматизовано, й має на меті перевірити не лише наявність уразливостей, а й ефективність засобів захисту, готовність персоналу та відповідність політик реагування. Методологічною основою можуть слугувати стандарти OSSTMM (Open Source Security Testing Methodology Manual) або NIST 800-115.

Таким чином, проактивні методи забезпечують можливість виявлення значної частини вразливостей до того, як вони будуть експлуатовані, що критично важливо для побудови ефективної системи превентивного захисту.

Реактивні методи виявлення вразливостей реалізуються у відповідь на події, що вже відбулися, або під час активного моніторингу інформаційної системи. Ці методи орієнтовані на своєчасне виявлення підозрілих або аномальних дій, які можуть свідчити про спробу експлуатації вже наявних вразливостей. На відміну від проактивних підходів, які спрямовані на попередження, реактивні методи дозволяють забезпечити оперативне реагування та мінімізувати наслідки інцидентів.

Одним із ключових напрямів є аналіз логів і подій, який становить основу більшості сучасних систем інформаційної безпеки. Лог-файли – це структуровані записи подій, що відбуваються у компонентах IT-інфраструктури: серверах, робочих станціях, мережевих пристроях, базах даних, веб-застосунках. Аналіз логів дозволяє виявляти аномальні входи, спроби ескалації привілеїв, доступ до критичних об’єктів чи зміну конфігурацій. Важливу роль тут відіграють SIEM-системи (Security Information and Event Management), які дозволяють централізовано збирати, зберігати та корелювати події з різних джерел у режимі реального часу. Прикладом таких систем є Splunk, IBM QRadar, ArcSight, які забезпечують створення складних правил виявлення атак та генерацію відповідних оповіщень.

Іншим важливим реактивним методом є виявлення поведінкових аномалій, що базується на аналізі відхилень від типового шаблону роботи користувачів або систем. Це можуть бути: різке збільшення обсягу переданих даних, доступ до нетипових ресурсів у неробочий час, повторювані невдалі спроби автентифікації тощо. Застосування алгоритмів машинного навчання дозволяє автоматично визначати такі аномалії та підвищити точність виявлення нових загроз, які ще не мають фіксованих сигнатур. Подібні механізми реалізуються як у SIEM-рішеннях, так і в окремих системах класу UEBA (User and Entity Behavior Analytics).

Особливе місце серед реактивних методів займає використання інформації про кіберзагрози (threat intelligence або TI). Мова йде про інтеграцію зовнішніх фідів з індикаторами компрометації (IOC), що включають IP-адреси, доменні імена, хеш-суми файлів, URL-адреси тощо, пов’язані з відомими атаками або інструментами зловмисників. Використання TI дозволяє корелювати локальні події з глобальними знаннями про атаки, підвищуючи ефективність виявлення. Для обміну цією інформацією використовуються стандартизовані формати STIX (Structured Threat Information Expression) і TAXII (Trusted Automated Exchange of Intelligence Information), а також платформи типу MISP (Malware Information Sharing Platform) [5].

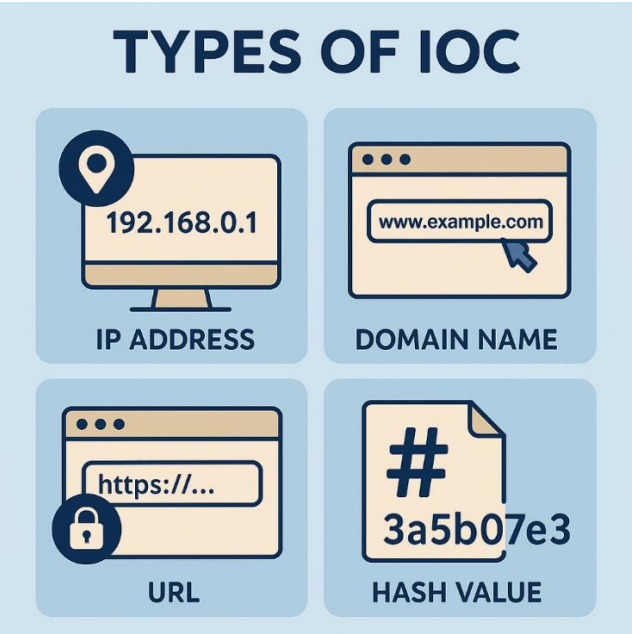


Рис 1.2. Типи індикаторів компрометації(IOC)

Реактивні методи дозволяють не лише виявити спроби експлуатації вразливостей, а й забезпечити збереження цифрових доказів для подальшого аналізу інцидентів. Це критично важливо в контексті відповідності нормативним вимогам (наприклад, ISO/IEC 27001 або GDPR), що зобов'язують організації забезпечувати як безперервний моніторинг, так і фіксацію подій безпеки.

Отже, реактивні методи є ключовим елементом будь-якої ефективної системи кіберзахисту, оскільки дозволяють забезпечити постійну оперативну обізнаність щодо стану безпеки організації, своєчасно ідентифікувати інциденти та мінімізувати їх наслідки.

Засоби виявлення вразливостей становлять основу інструментарію спеціаліста з інформаційної безпеки. Їх класифікація базується на принципах дії, функціональному призначенні, типах джерел даних, що обробляються, а також на рівні інтеграції в IT-інфраструктуру. Кожен клас таких засобів реалізує власну методику виявлення та відображає певний підхід до захисту.

Одним із найбільш поширених класів є сканери вразливостей. Вони забезпечують автоматичне виявлення відомих вразливостей у системах, конфігураціях та програмному забезпеченні, порівнюючи зібрані дані з базами CVE або вендорськими бюлетенями. Найбільш популярними представниками цього класу є Nessus (Tenable), Qualys Vulnerability Management, Greenbone Vulnerability Manager (OpenVAS). Ці засоби широко використовуються в процесах управління ризиками, сертифікації безпеки та планових аудитах.

Другий вагомий клас – це системи управління подіями та інформацією безпеки (SIEM). Вони здійснюють збір, нормалізацію, аналіз та кореляцію подій з багатьох джерел: серверів, мережевого обладнання, EDR, хмарних сервісів тощо. Прикладами таких систем є Splunk Enterprise Security, IBM QRadar, Micro Focus ArcSight. SIEM дозволяє не лише ідентифікувати прояви активної експлуатації вразливостей, а й формувати комплексну картину загроз завдяки використанню правил кореляції, поведінкових профілів і аналітичних дашбордів.

Наступну категорію становлять системи виявлення загроз на кінцевих точках (EDR) та розширеного реагування (XDR). Вони аналізують активність на робочих станціях, серверах або мобільних пристроях у режимі реального часу. Сучасні рішення, такі як CrowdStrike Falcon, Microsoft Defender for Endpoint, SentinelOne, дозволяють виявляти підозрілі процеси, сценарії lateral movement, ескалацію привілеїв і створювати повноцінну ланцюгову аналітику інцидентів. XDR, своєю чергою, поєднує телеметрію з мережі, хмари, пошти та кінцевих точок, формуючи уніфіковану платформу моніторингу загроз.

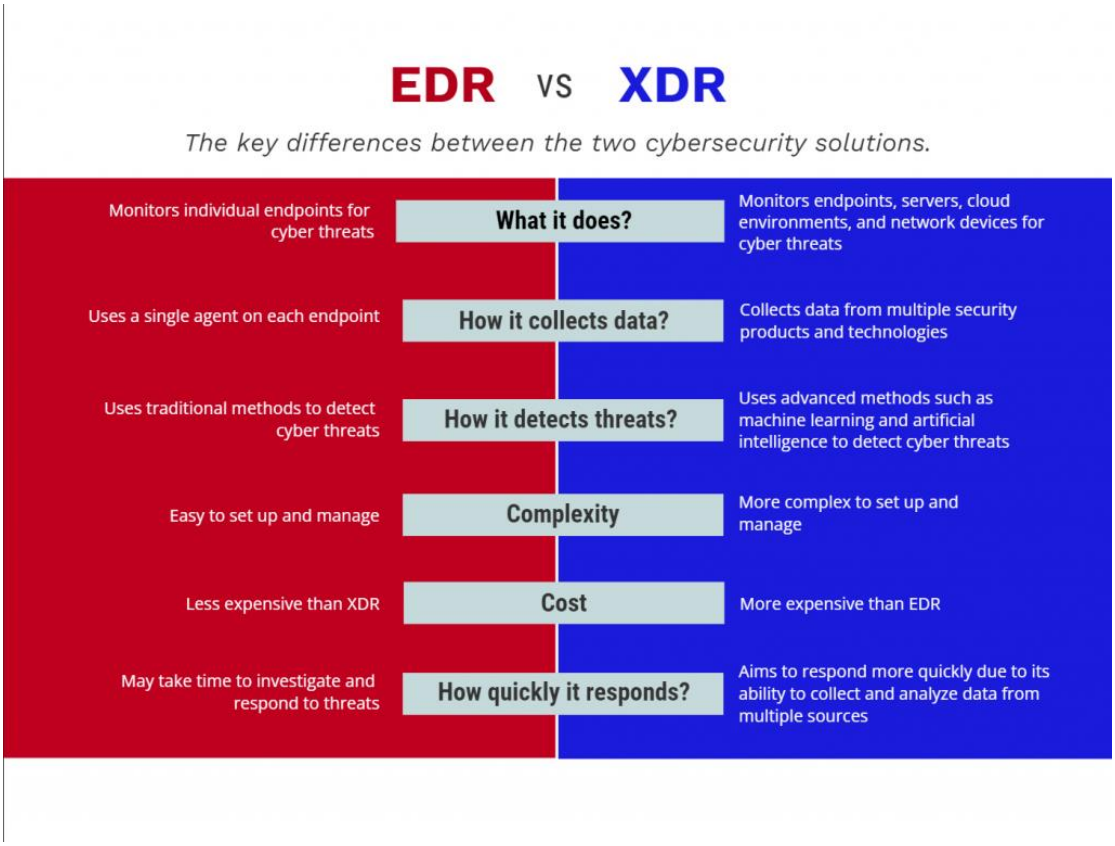


Рис. 1.3. EDR та XDR

До окремого класу належать інструменти аналізу безпеки програмного коду, що застосовуються в контексті DevSecOps. Статичні (SAST) і динамічні (DAST) сканери, такі як SonarQube, Checkmarx, Burp Suite, використовуються для виявлення небезпечних шаблонів у коді, бібліотеках та інтерфейсах, і дозволяють знизити ризики ще до розгортання системи в продуктивному середовищі. Їх

інтеграція в CI/CD-пайплайни є однією з кращих практик сучасної безпеки розробки.

Варто також згадати платформи автоматизації реагування (SOAR), які, хоча й не виявляють вразливості напряму, значно підвищують ефективність їх обробки. Splunk SOAR, Cortex XSOAR, IBM Resilient дозволяють автоматизувати типові сценарії: повідомлення відповідальних осіб, створення інциденту, запуск сканування або ізоляцію активу. Такі платформи реалізують концепцію машинного реагування на основі попередньо налаштованих playbook'ів.

Застосування кожного з наведених класів засобів залежить від специфіки організації, її IT-архітектури та рівня зрілості системи управління інформаційною безпекою. У більшості випадків найбільш ефективним є поєднання кількох інструментів різного рівня – від кінцевих точок до мережевої та прикладної аналітики – у межах єдиного ланцюга реагування.

Ефективне виявлення вразливостей в інформаційних системах неможливе без комплексного підходу, що поєднує як проактивні, так і реактивні методи. Кожен із них виконує свою унікальну функцію в системі захисту – від попереджувального сканування та аналізу на етапі розробки до моніторингу поведінки користувачів і подій у реальному часі. Використання лише одного типу методів – наприклад, виключно сканерів або тільки SIEM – створює сліпі зони, які можуть бути використані зловмисниками.

Застосування комбінованого підходу, що включає SAST/DAST, пентест, сканування вразливостей, кореляційний аналіз логів, UEBA і threat intelligence, дозволяє організаціям охопити повний цикл виявлення, класифікації та реагування на загрози. Такий підхід відповідає концепції глибинного захисту (defense-in-depth), рекомендованій у стандартах ISO/IEC 27001 і NIST Cybersecurity Framework, згідно з якою безпека повинна бути багаторівневою, взаємозалежною та адаптивною.

Особливе значення має інтеграція виявлення вразливостей у загальну систему управління інформаційною безпекою підприємства. Це передбачає

централізований облік подій, автоматизоване корелювання подій із різних джерел, візуалізацію загроз і можливість оперативного реагування. Саме таку функціональність надають сучасні SIEM-платформи, які виступають не лише сховищем логів, а повноцінним аналітичним центром кібербезпеки.

Одним із провідних рішень у цій категорії є Splunk, що завдяки своїй гнучкій архітектурі, можливостям масштабування, підтримці тисячі типів джерел даних і потужному механізму обробки подій (Splunk Processing Language, SPL) забезпечує не лише виявлення вразливостей, а й повноцінний цикл управління інцидентами.

1.3. Аналіз рішень для виявлення вразливостей в організації

Актуальним є огляд основних класів технологічних рішень, які найчастіше використовуються для виявлення вразливостей у корпоративному середовищі.

Однією з найбільш поширених категорій є сканери вразливостей. Такі рішення, як Nessus, Qualys та OpenVAS, дозволяють швидко інвентаризувати активи, виявити конфігураційні помилки та недоліки у версіях ПЗ, що відповідають відомим CVE. Їх основна функція – своєчасне сповіщення про наявні технічні вади, які можуть бути легко усунені засобами патч-менеджменту.

Другий клас – це системи управління подіями безпеки (SIEM), які забезпечують централізовану кореляцію подій, збір логів із тисячі джерел і побудову аналітики загроз. Splunk, IBM QRadar та ArcSight стали лідерами у цій галузі завдяки гнучкості, підтримці уніфікованих форматів даних і потужному механізму створення правил виявлення. Вони дозволяють не лише зафіксувати ознаки експлуатації вразливостей, а й розкрити повну картину інциденту: початковий вектор, ланцюг дій зловмисника, уражені активи.

Не менш важливу роль відіграють рішення класу EDR/XDR, що працюють на рівні кінцевих точок. CrowdStrike Falcon, SentinelOne та Microsoft Defender for Endpoint реалізують постійний моніторинг поведінки систем, що дозволяє

виявляти не лише вже відомі типи атак, але й нові, на основі аномалій або відхилень від типових шаблонів.

На рівні розробки велику увагу привертають засоби аналізу вихідного коду – SAST/DAST-рішення на кшталт SonarQube, Checkmarx або Veracode, що інтегрують у DevOps-процеси. Вони дозволяють зменшити кількість вразливостей ще до релізу продукту, що економить ресурси та знижує ризики.

Останнім елементом є SOAR-платформи (наприклад, Splunk SOAR або Cortex XSOAR), які автоматизують обробку подій: запуск сканування, сповіщення аналітиків, виконання ізоляцій активів тощо.

Таким чином, ефективна система виявлення вразливостей базується не на використанні одного конкретного рішення, а на комбінації інструментів, які доповнюють один одного на різних рівнях: від розробки до експлуатації, від кінцевих точок до хмари. У центрі цієї багаторівневої системи дедалі частіше розташовується SIEM-платформа, яка виконує роль координатора, корелятору та аналітичного центру реагування.

2 АНАЛІЗ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВИЯВЛЕННЯ ВРАЗЛИВОСТЕЙ НА БАЗІ SPLUNK

2.1. Архітектура рішення та принцип роботи Splunk

У сучасному IT-середовищі, де організації функціонують у гетерогенних, розподілених та хмароорієнтованих інфраструктурах, ризики зростають експоненційно. Вразливість одного елемента системи може стати точкою входу для складних атаках, спрямованих на розширення доступу, підвищення привілеїв або витік критичних даних [6]. Це особливо актуально в умовах розширеної відповідальності за захист персональних даних (GDPR, CCPA), цифрової трансформації та активного використання SaaS/PaaS-рішень.

Інтеграція централізованих рішень для виявлення вразливостей, таких як Splunk, дозволяє побудувати системний підхід до безпеки: здійснювати безперервний моніторинг, збір і аналіз подій, автоматизовану обробку сигналів тривоги, а також формувати узгоджену аналітику щодо потенційних ризиків. Такі системи функціонують як центральні вузли обробки телеметрії з різних джерел (мережеве обладнання, сервери, додатки, хмарні сервіси), забезпечуючи побудову повної картини стану безпеки в організації.

Таким чином, впровадження платформи виявлення вразливостей – це не просто технологічне рішення, а стратегічний крок у побудові адаптивної, стійкої до загроз цифрової інфраструктури, що дозволяє організаціям швидко та ефективно реагувати на виклики кібербезпеки, дотримуватись нормативних вимог та захищати найбільш критичні ресурси бізнесу.

Проєктування системи виявлення вразливостей потребує чіткого визначення її функціонального призначення та експлуатаційних характеристик. Визначення вимог є критичним етапом, що безпосередньо впливає на вибір архітектури, інструментів, методів обробки даних і стратегії розгортання. Ефективна система

має задовольняти як функціональні потреби (тобто що саме вона повинна робити), так і нефункціональні (вимоги до продуктивності, масштабованості, безпеки тощо).

До функціональних вимог належать:

- Централізований збір журналів подій із широкого спектра джерел (серверів, робочих станцій, мережевого обладнання, хмарних сервісів, платформ розробки тощо).
- Попередня обробка, нормалізація та зберігання даних у структурованому вигляді з можливістю подальшої індексації.
- Механізми виявлення потенційно небезпечної активності, включаючи сигнатурний, евристичний та поведінковий аналіз.
- Інтерфейс для аналітиків безпеки, що дозволяє формулювати запити, виводити статистику, будувати дашборди, налаштовувати автоматизовані оповіщення.
- Підтримка розслідувань та аудиту, включаючи збереження подій, формування звітів, інтеграцію з іншими системами (наприклад, CMDB, SOAR, threat intelligence-фідами).

Нефункціональні вимоги формуються з урахуванням сучасних реалій цифрової інфраструктури та високих навантажень на систему:

- Продуктивність – обробка великих обсягів телеметрії у режимі, наближеному до реального часу.
- Масштабованість – підтримка горизонтального розширення за рахунок додавання вузлів або ресурсів.
- Висока доступність і відмовостійкість – реплікація даних, балансування навантаження, автоматичне відновлення.
- Безпека самої системи – автентифікація, контроль доступу до даних, захист трафіку (TLS), аудит дій користувачів.
- Інтеграційна сумісність – взаємодія з іншими системами безпеки (EDR, DLP, AD, Cloud API), підтримка форматів журналів (CEF, Syslog, JSON тощо).

Важливо, що як функціональні, так і нефункціональні вимоги були адаптованими до структури IT-інфраструктури. Система має бути здатною до оперативного розширення, оновлення механізмів виявлення, застосування нових джерел загроз і підтримки регуляторних вимог, які постійно еволюціонують.

Саме завдяки відповідності цим вимогам платформа Splunk набула широкого визнання у сфері корпоративної безпеки, оскільки поєднує гнучку архітектуру, високий рівень кастомізації, багаторівневу аналітику та масштабованість.

Архітектура Splunk побудована за модульним принципом, що забезпечує масштабованість, гнучкість та ефективне розподілення навантаження між компонентами системи. Кожен компонент виконує чітко визначену роль у процесі збору, обробки, зберігання, аналізу та візуалізації даних. Такий підхід дозволяє розгортати Splunk у конфігураціях різної складності – від однокомпонентного варіанту до розподілених кластерів корпоративного рівня [7].

Основні компоненти архітектури Splunk:

- Universal Forwarder (UF) – це легкий агент, який встановлюється на джерела даних (сервери, мережеві пристрої, хмарні системи) і виконує функцію збору журналів. UF не обробляє дані, а лише транспортує їх до індексера. Він характеризується низьким споживанням ресурсів і високою швидкістю передачі, що робить його оптимальним для масштабного розгортання.
- Indexer, що отримує дані від Forwarder-ів, виконує парсинг, нормалізацію, індексацію та зберігання подій. Індексери створюють пошукові файли, які забезпечують можливість швидкого доступу до інформації. У розподіленій архітектурі можливе об'єднання кількох індексерів у кластер для забезпечення відмовостійкості та масштабування.
- Search Head (SH) – призначений для взаємодії з користувачем, виконання пошукових запитів, побудови аналітики, дашбордів, формування оповіщень і звітів. Він інтерпретує запити, написані мовою SPL (Search Processing Language), та здійснює агрегацію результатів. У великих розгортаннях

використовується кластер Search Head-ів для розподілення запитів між кількома вузлами.

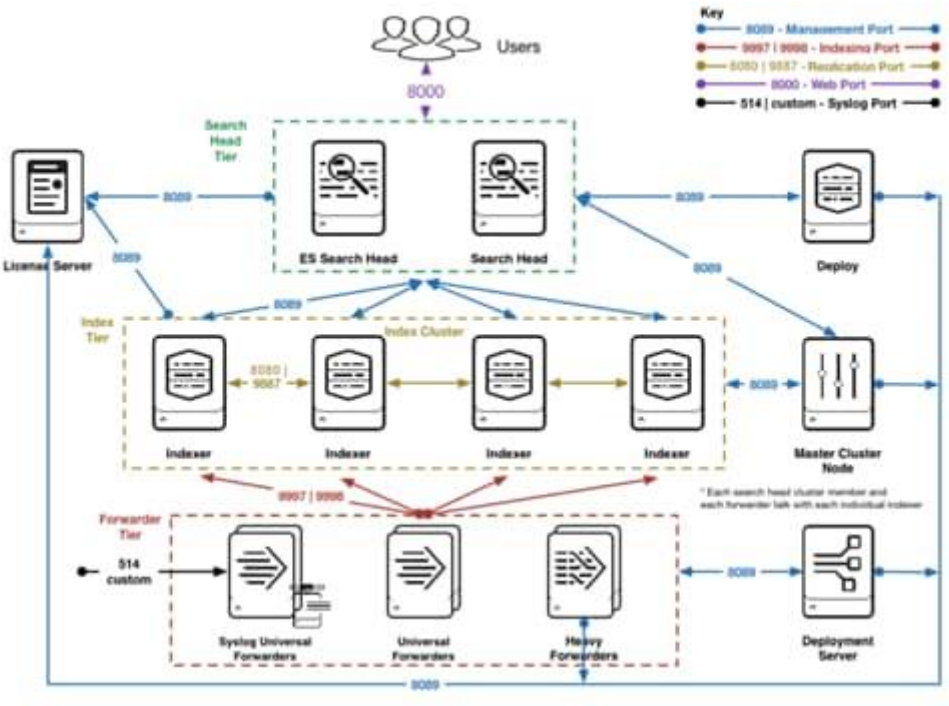


Рис. 2.1. Принцип роботи Splunk

Усі ці компоненти взаємодіють у межах узгодженого протоколу, забезпечуючи наскрізний потік даних – від первинного збору до кінцевої аналітики. Завдяки чіткому розмежуванню ролей, архітектура Splunk легко адаптувати до потреб організації, дозволяючи збільшити обчислювальні ресурси у разі зростання обсягів даних чи кількості користувачів.

2.2. Проектування механізмів збору й обробки логів

Сучасні системи забезпечення кібербезпеки повинні не лише фіксувати інциденти, а й виявляти потенційні загрози ще до того, як вони спричинять реальну шкоду. Особливої ваги в цьому контексті набуває можливість оперативного виявлення типових атак, які часто застосовуються у реальних умовах: спроби

перебору облікових даних (brute-force), звернення з підозрілих IP-адрес, а також атаки на рівні веб-застосунків, зокрема SQL- та XSS-ін'єкції.

У системах класу SIEM, таких як Splunk, процес виявлення загроз базується на аналізі телеметрії з численних джерел: логів веб-серверів, систем автентифікації, мережевого обладнання, проксі-серверів, хмарних сервісів тощо. Ці дані проходять процес нормалізації, індексації та зберігаються у форматі, придатному для подальшого аналітичного оброблення.

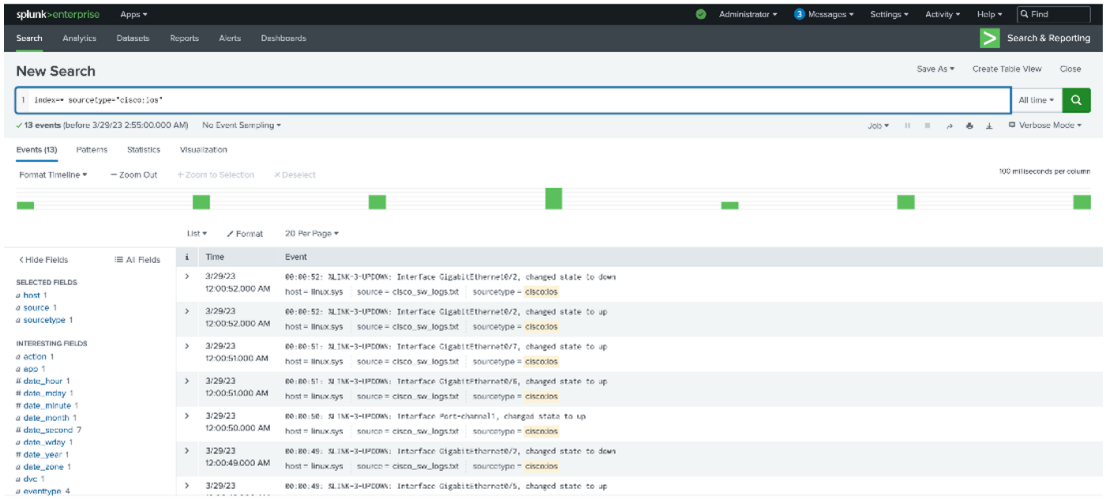


Рис. 2.2. Приклад обробки логів в Splunk

Виявлення загроз відбувається за трьома основними підходами:

- Сигнатурний підхід – полягає у виявленні відомих шаблонів атак, що повторюються. Наприклад, певна послідовність SQL-запиту у URL або типовий User-Agent, що використовується в інструментах автоматизації атак.
- Поведінковий аналіз – ґрунтується на виявленні аномалій у звичній активності користувача або системи. Наприклад, логін у нічний час з невідомої країни, нехарактерне навантаження на певний ресурс, або часта зміна сесій [8].
- Кореляційний підхід – передбачає виявлення логічних залежностей між подіями, які поодинці можуть здаватися нешкідливими, але у сукупності утворюють ланцюг атаки.

Практична ефективність SIEM-системи багато в чому визначається її здатністю виявляти найбільш поширені вектори атак. У цьому підпункті

розглянуто три типові сценарії, які часто фіксуються в корпоративних середовищах і піддаються ефективному аналізу за допомогою Splunk.

Brute-force атака передбачає масовий перебір пар логін-пароль з метою підбору дійсних облікових даних. Ознаками такої активності є велика кількість невдалих спроб входу за короткий часовий інтервал із тієї самої IP-адреси або щодо одного облікового запису [9].

У Splunk такий сценарій реалізується через агрегування логів аутентифікації, зокрема на основі поля «action=failure», фільтрації за полем «user» або «src_ip», та використання операцій «stats count by user, src_ip». За умови перевищення заданого порогу (наприклад, >5 невдалих спроб за 1 хвилину), генерується сповіщення.

Типовий фрагмент SPL-запиту:

```
index=auth action=failure
| stats count by user, src_ip
| where count > 5
```

Таке правило дозволяє не лише виявити активну атаку, але й спрогнозувати спроби компрометації облікових даних у майбутньому.

Активність із зовнішніх IP-адрес, які мають погану репутацію або розташовані в нетипових для організації регіонах, може свідчити про спробу сканування, несанкціонованого доступу або тестування інфраструктури на слабкі місця.

У Splunk подібна активність визначається через геолокаційну фільтрацію (GeoIP), звірку з зовнішніми фідами IOC (indicators of compromise), аномальний час доступу або кількість запитів до обмежених ресурсів.

Приклад запиту з використанням lookup-таблиці репутацій:

```
index=network
| lookup malicious_ips ip_address AS src_ip OUTPUT threat_level
| where threat_level="high"
```

Таким чином аналітик може виявити взаємодію з інфраструктурою зловмисників ще до реалізації власне атаки.

Веб-атаки типу SQL-ін’єкція та міжсайтове скриптування (XSS) становлять серйозну загрозу для веб-застосунків, особливо за відсутності валідації введених даних. У журналах доступу (web access logs) вони можуть виглядати як запити, що містять специфічні конструкції, ключові слова або теги.

У Splunk це реалізується через пошук регулярних виразів у параметрах HTTP-запитів, наприклад, у полях uri_query, uri_path, http_method.

Приклад фрагменту запиту для SQL-ін’єкції:

index=web_logs uri_query="*SELECT*" OR uri_query="*1=1*"

Приклад для XSS:

index=web_logs uri_query="*<script*"

Такі запити можна вдосконалювати за допомогою поля «user_agent», яке дозволяє визначити, чи йдеться про автоматизовану активність.

Splunk також дозволяє комбінувати ці підходи у межах складних кореляційних правил, що об’єднують сигнатурні елементи, часові шаблони та контекст поведінки, забезпечуючи детальні сценарії реагування [10].

Розглянуті приклади свідчать про здатність платформи Splunk ефективно виявляти як прямі загрози (brute-force, SQL/XSS), так і непрямі індикатори потенційних атак (репутаційні IP, нетипова поведінка). Завдяки використанню механізмів пошуку, агрегації та фільтрації подій, а також підтримці регулярних виразів і збагачених джерел даних, користувачі можуть не лише ідентифікувати інциденти, а й будувати логіку попередження атак на ранніх стадіях.

Доповненням до процесу виявлення є візуалізація отриманих результатів. У реальних умовах аналітики рідко працюють безпосередньо з пошуковими запитами – натомість вони використовують готові дашборди, панелі індикаторів та системи сповіщень. Саме ці інструменти дозволяють зосередитися на інтерпретації загроз і ухваленні рішень.

2.3. Побудова дашбордів і оповіщень у Splunk

У сучасних SIEM-рішеннях особливу роль відіграють засоби візуалізації даних (дашборди) та механізми оповіщення, що дозволяють не лише виявляти інциденти, а й ефективно ними керувати.

Дашборди є основним інструментом представлення результатів аналізу в Splunk. Вони дають змогу в реальному часі переглядати ключові показники безпеки, відстежувати динаміку атак, виявляти аномалії та отримувати миттєве уявлення про стан IT-інфраструктури. Від простих таблиць і гістограм до комплексних інтерактивних heatmap-ів – Splunk надає широкі можливості створення аналітичних панелей під конкретні задачі служби безпеки[11].

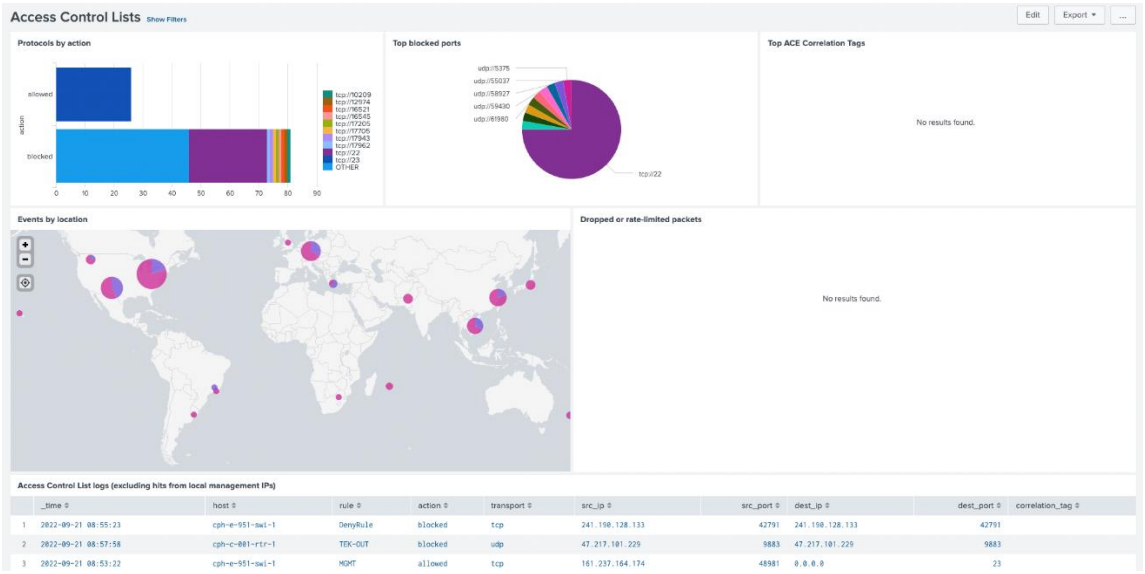


Рис. 2.3. Дашборд Splunk

Оповіщення (alerts), своєю чергою, автоматизують контроль за виявленням критичних подій. Вони дозволяють задати умови спрацювання (кількість, час, вміст подій), частоту перевірки, тип повідомлення і навіть дії, які система має виконати [12]. Це критично для скорочення часу реагування на інциденти – від моменту виявлення до початку дій.

Побудова дашбордів у Splunk – це ключовий етап трансформації даних з подій та журналів у доступну й оперативно зрозумілу інформацію для фахівців з

безпеки. Дашборди дозволяють створювати єдине вікно спостереження (single-pane-of-glass), в якому об’єднуються результати різних запитів, графіки, таблиці та KPI-показники. Вони використовуються як для поточного моніторингу, так і для презентації історичних трендів, активності атак чи ефективності реагування.

У Splunk дашборди створюються на основі SPL-запитів (Search Processing Language), результати яких виводяться у вигляді графічних віджетів. Візуалізація даних може бути реалізована через:

- Таблиці з розбиттям по загрозах (тип події, IP, кількість, геолокація);
- Кругові діаграми (розподіл загроз за категоріями або критичністю);
- Гістограми/лінійні графіки (динаміка подій у часі);
- Теплові карти (heatmaps) (наприклад, активність за IP або регіонами);
- Графіки KPI (кількість атак за добу, середній час виявлення, кількість alert'ів).

Приклад візуального запиту для розподілу типів загроз:

index=threat_logs

| stats count by threat_type

| sort – count

Цей запит може бути виведений як кругова діаграма з типами загроз (SQLi, brute-force, XSS) та кількістю виявлень.

Інший приклад: розподіл IP-адрес з підозрілою активністю:

index=network_logs status=denied

| stats count by src_ip

| sort - count

| head 10

Його зручно виводити як горизонтальну гістограму для швидкого визначення “найактивніших” зловмисників.

У Splunk доступні два варіанти інтерфейсів для побудови дашбордів [13]:

- Classic Dashboard Editor – простий інтерфейс із шаблонними елементами;
- Dashboard Studio – розширене середовище для візуального дизайну, анімації, інтерактивних фільтрів, адаптації до KPI.

Один дашборд може містити віджети з різних джерел: логів веб-додатків, систем автентифікації, проксі-серверів, брандмауерів або навіть зовнішніх TI-джерел.

Рис. 2.4. Дашборд з віджетами

Автоматичні оповіщення (alerts) у Splunk є ефективним інструментом реагування на інциденти безпеки. Вони дозволяють системі самостійно ініціювати дії на основі заданих умов, що суттєво скорочує час між виявленням загрози та початком її нейтралізації [12].

Оповіщення в Splunk формуються на базі будь-якого SPL-запиту, результати якого перевіряються за вказаними критеріями. Основні умови спрацювання включають:

- Кількісні пороги: наприклад, більше ніж 5 невдалих спроб входу з однієї IP-адреси за 5 хвилин;
- Наявність значень у результатах (подій певного типу, записів з критичним статусом);
- Часові інтервали: регулярне спрацювання щохвилини/щогодини/щодня;
- Періодичність перевірки та “cooldown” для уникнення дублювання алертів.

Приклад SPL-запиту для алерту на brute-force активність:

```
index=auth_logs action=failure
| stats count by user, src_ip
| where count > 5
```

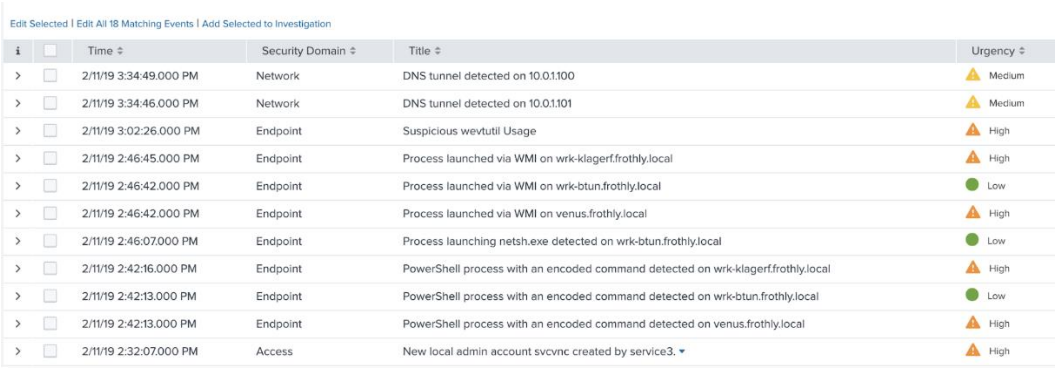
Після спрацювання оповіщення Splunk підтримує такі типи дій:

- Надсилання електронного листа (із шаблонізованим звітом);
- HTTP Webhook (інтеграція з Slack, MS Teams, Telegram, Jira тощо);
- Виклик скриптів для ініціювання блокування, зміни правил фаєрволу;
- Створення інциденту у зовнішніх системах: ServiceNow, SOAR, TheHive тощо.

Інтерфейс налаштування оповіщень дозволяє:

- задати умови спрацювання (тригери),
- вибрати формат повідомлення,
- додати відфільтровані поля (IP, час, тип події),
- вказати графік виконання та політику повторень.

У Splunk також реалізовано механізм «notable events», що дозволяє виносити критичні події у спеціальний розділ для швидкого доступу. Такі події можуть агрегуватись у таймлайни, прив’язуватись до користувачів або систем, що є основою для реагування в рамках SOC-процесів.



Edit Selected | Edit All 18 Matching Events | Add Selected to Investigation

i	Time	Security Domain	Title	Urgency
>	2/11/19 3:34:49.000 PM	Network	DNS tunnel detected on 10.0.1.100	Medium
>	2/11/19 3:34:46.000 PM	Network	DNS tunnel detected on 10.0.1.101	Medium
>	2/11/19 3:02:26.000 PM	Endpoint	Suspicious wevtutil Usage	High
>	2/11/19 2:46:45.000 PM	Endpoint	Process launched via WMI on wrk-klagerf.frothly.local	High
>	2/11/19 2:46:42.000 PM	Endpoint	Process launched via WMI on wrk-btun.frothly.local	Low
>	2/11/19 2:46:42.000 PM	Endpoint	Process launched via WMI on venus.frothly.local	High
>	2/11/19 2:46:07.000 PM	Endpoint	Process launching netsh.exe detected on wrk-btun.frothly.local	Low
>	2/11/19 2:42:16.000 PM	Endpoint	PowerShell process with an encoded command detected on wrk-klagerf.frothly.local	High
>	2/11/19 2:42:13.000 PM	Endpoint	PowerShell process with an encoded command detected on wrk-btun.frothly.local	Low
>	2/11/19 2:42:13.000 PM	Endpoint	PowerShell process with an encoded command detected on venus.frothly.local	High
>	2/11/19 2:32:07.000 PM	Access	New local admin account svcvnc created by service3.	High

Рис. 2.5. Механізм «notable events».

Таким чином, у поєднанні з дашбордами, оповіщення у Splunk реалізують наступний цикл: від виявлення, створення сповіщення та відповідної реакції.

3 РЕКОМЕНДАЦІЇ ЩОДО ЗАСТОСУВАННЯ SPLUNK ДЛЯ ЗАХИСТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1. Варіанти впровадження Splunk в організації

Найпоширенішим є використання хмарної моделі Splunk Cloud Platform, яка надає високу доступність і зменшує потребу у локальних потужностях. У середовищах з підвищеними вимогами до контролю над інфраструктурою застосовується локальне розгортання на власних серверах, що дає змогу гнучко керувати політиками зберігання та доступу до даних. Гібридна модель — компроміс між хмарною гнучкістю та локальною безпекою — використовується у великих компаніях, де частина логів обробляється локально, а частина — через хмарні ресурси.

Ключовим етапом є інтеграція Splunk з джерелами подій організації. Для цього використовується набір агентів: Universal Forwarder для базових джерел, Heavy Forwarder для попередньої обробки даних та маршрутизації, а також API-інтеграція для SaaS-рішень. Наприклад, для збору логів з Windows Server потрібно встановити Universal Forwarder, вказати порт зв’язку з індексатором (наприклад, 9997), а також вказати шляхи до логів (наприклад, C:\Windows\System32\winevt\Logs\Security.evtx). Аналогічно, для логів з мережевих пристроїв використовують syslog-протокол або REST API, залежно від типу обладнання.

Масштабованість системи забезпечується за рахунок архітектурного розділення функціональності: індексатори відповідають за обробку та збереження даних, пошукові вузли (search heads) — за аналітичні запити, deployment server — за централізоване оновлення конфігурацій на форвардерах. Наприклад, у середовищі на 3000 пристроїв оптимально використовувати як мінімум три індексатори та один пошуковий вузол із балансуванням навантаження.

Успішне впровадження вимагає не лише підключення даних, а й ефективної організації структури індексів. Рекомендується створювати окремі індекси за типами систем або функціональними зонами (наприклад, index=windows, index=firewall, index=gmail_logs). Це спрощує контроль доступу, аудит і підвищує продуктивність пошукових запитів.

Для оптимізації ресурсів при розгортанні Splunk важливо ретельно розраховувати обсяг даних, які надходять у систему, адже ліцензування базується на обсязі щоденного індексування. Наприклад, якщо середній обсяг логів від одного сервера становить близько 500 МБ на добу, а в інфраструктурі 100 серверів, то щоденний обсяг індексації перевищить 50 ГБ. У таких випадках варто:

- виключити малозначущі логи з конфігурації агента (наприклад, WinEventLog:Setup);
- використовувати фільтри на рівні Heavy Forwarder для усунення дублювання подій;
- агрегацію подій (наприклад, замість кожного успішного входу — агреговані підсумки на годину).

```
[WinEventLog:Security]
blacklist1 = EventCode="4662"
blacklist2 = EventCode="4688" Message="*powershell*"
```

Після підключення логів до Splunk доцільно побудувати базові дашборди моніторингу, які візуалізують активність та дозволяють виявляти аномалії. Наприклад, для виявлення спроб сканування портів або входів під обліковими записами адміністраторів можна налаштувати панель, що містить:

- графік за кількістю невдалих логінів;
- географічну карту джерел трафіку;
- топ джерел із високим обсягом запитів.

Розгортання Splunk потребує врахування масштабу організації. Для малих підприємств (до 50 хостів) можливо розмістити всі компоненти на одному сервері

з економною ліцензією (до 5 ГБ/день). Такий підхід дозволяє швидко впровадити базовий моніторинг із відносно низькими витратами. Натомість для корпоративного середовища необхідна розділена архітектура, що передбачає:

- кілька індексаторів із реплікацією даних;
- окремі Search Head-кластери;
- власний сервер для керування агентами;
- окрему групу правил RBAC (рольового доступу) для аналітиків, аудиторів, адміністраторів.

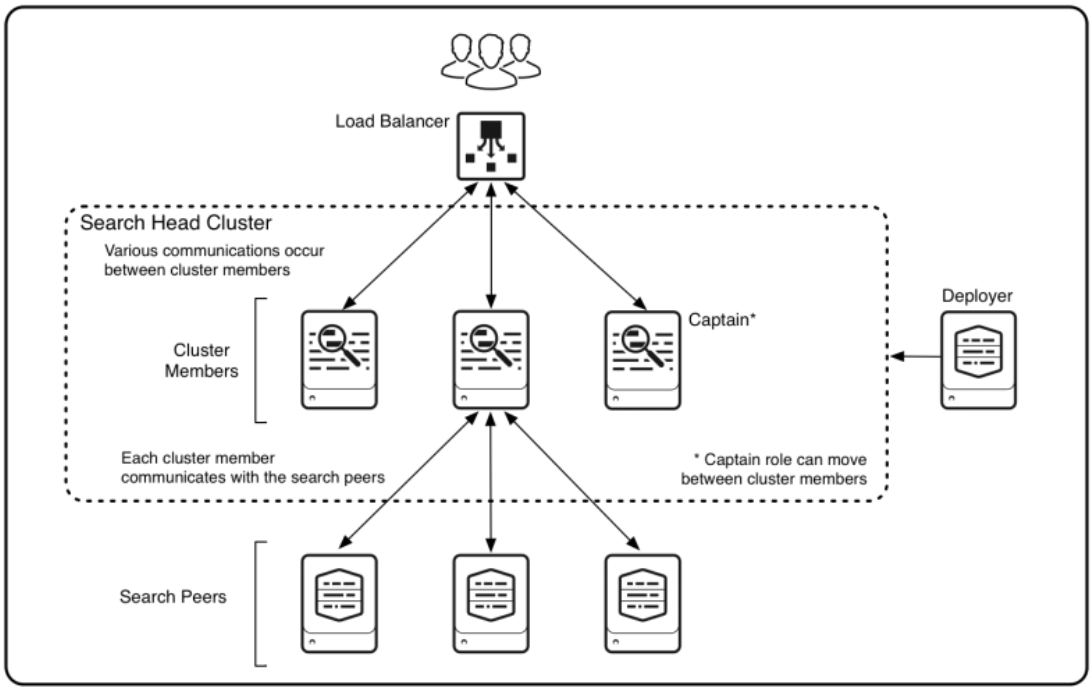


Рис. 3.1. Архітектура Splunk у великій організації з регіональними офісами

Кожна філія має Universal Forwarder, який передає дані через Heavy Forwarder у центральний кластер індексаторів. Аналіз даних здійснюється через кластер пошукових вузлів. SaaS-дані інтегруються через HTTP Event Collector (HEC) [14].

У рамках налаштування прав доступу необхідно ретельно продумати ролі в межах Splunk. Наприклад, аудиторам потрібен доступ лише на читання до певних індексів (role=auditor), тоді як аналітики мають доступ до збережених запитів, дашбордів і alert'ів (role=soc_analyst). Усе це реалізується через web-інтерфейс:

Settings > Access Controls > Roles.

Також для управління впровадженням доцільно використовувати Deployment Server, який централізовано поширює конфігураційні файли на всі агенти. Це забезпечує уніфікацію правил збирання логів та спрощує адміністрування розподіленого середовища.

Ще одним важливим аспектом є контроль за змінами та активністю в системі. Splunk надає можливість аудиту змін, що дозволяє визначити, хто додавав нові індекси, вносив правки у конфігурації або створював алерти. Для цього використовується індекс audit:

index=_audit action=edit OR action=create user!="splunk-system-user"

Ці дані дозволяють адміністраторам здійснювати прозорий моніторинг змін і дій усіх залучених осіб, що критично для регламентованих галузей (банківський сектор, держслужби).

3.2. Створення кореляційних правил для виявлення потенційних загроз

Впровадження правил для виявлення потенційних вразливостей та інцидентів безпеки, дозволяє автоматизовано зіставляти розрізнені події з різних джерел — від мережевих журналів до EDR/XDR-платформ — з метою виявлення аномалій або шаблонів, що можуть вказувати на зловмисну активність. У контексті організації, яка використовує Splunk як SIEM-рішення, процес створення кореляційних правил вимагає не лише глибокого технічного розуміння структури подій, а й знання специфіки IT-інфраструктури, бізнес-логіки та поведінкових моделей користувачів.

Першим кроком у створенні ефективного механізму кореляції є виявлення джерел даних, які необхідно індексувати. Це включає події з Active Directory, firewall, проксі-серверів, антивірусів, EDR-агентів, cloud-ресурсів, а також журнали доступу до внутрішніх веб-додатків і баз даних. Дані мають бути нормалізовані

згідно з CIM (Common Information Model) — це забезпечує уніфікований формат подій і полегшує побудову уніфікованих запитів [15]. Наприклад, журнал з Palo Alto Firewall і Windows Event Log можуть бути приведені до однієї логічної моделі "Network Traffic", що дозволяє зіставляти спробу доступу до певного ресурсу з подальшою успішною аутентифікацією чи запуском PowerShell.

Одним з класичних сценаріїв кореляції є виявлення послідовних дій, що відповідають техніці з MITRE ATT&CK, наприклад Initial Access → Execution → Lateral Movement. У Splunk це реалізується за допомогою SPL (Search Processing Language), де будується логіка об'єднання подій за ключовими атрибутами, такими як src_ip, user, process, command_line, dest, timestamp [16]. Зазвичай використовується оператор transaction, join або stats у поєднанні з eval, where, lookup, regex тощо. Наприклад, наступне правило може зіставляти появу нової обліковки адміністратора з попереднім виконанням підозрілої PowerShell-команди:

```
index=windows EventCode=4720 OR (EventCode=4104
command_line="*Invoke-Mimikatz*")
| stats earliest(_time) as first_seen latest(_time) as last_seen by user, host
| where last_seen - first_seen < 300
```

У цьому випадку правило реагує на швидку послідовність подій, що може свідчити про ескалацію привілеїв. Подібні правила потребують калібрування, щоб зменшити кількість false positives. Для цього в практиці організацій використовують whitelist-підходи, threat-intel lookup’и, контекстну інформацію про критичність хостів (наприклад, через Tag Management у Splunk ES), та таймфрейми кореляції, які дозволяють уникати шуму у великій подієвій базі.

Кореляційні правила можуть також ґрунтуватися на зовнішніх Threat Intelligence-даних. Наприклад, інтеграція з платформами VirusTotal, MISP або CrowdStrike Intelligence дозволяє збагачувати журнали даних про IP, URL, домени чи файли, що згадуються в телеметрії, та перевіряти їх на відповідність відомим зловмисним об’єктам. Використовуючи lookup або inputlookup, Splunk може зіставити зовнішні IoC з внутрішньою активністю:

```
index=network_traffic | lookup malicious_iocs ip as dest_ip OUTPUT ip as malicious_ip
| search malicious_ip=*
```

У середовищах із високими вимогами до відповідності стандартам (наприклад, ISO/IEC 27001, PCI DSS, NIS2), кореляційні правила також включають аудит доступу до критичних систем, контроль змін у конфігураціях безпеки, а також події, пов’язані з відмовою у доступі чи зміною політик MFA. У Splunk Enterprise Security можна використовувати кореляційні пошуки як частину Adaptive Response Framework, де виявлення конкретної загрози автоматично ініціює відповідну дію — наприклад, блокування IP через firewall API або тимчасове відключення облікового запису у Active Directory через SOAR-інтеграцію.

Реальне розгортання таких правил у продакшн-середовищі потребує не лише розробки SPL-запитів, а й створення тестового середовища для перевірки на якість спрацювання. Це передбачає емуляцію атак за допомогою сценаріїв MITRE CALDERA, Atomic Red Team або власних playbook’ів. За результатами таких симуляцій проводиться тонке налаштування умов спрацювання — наприклад, встановлення граничних значень для кількості подій за певний період, зміни параметрів агрегації, або підбір специфічних атрибутів об’єктів, які унікальні для організації. Наприклад, в банківському секторі критичною вважається активність у системах SWIFT чи Core Banking, тому правила можуть бути орієнтовані на нестандартний доступ до відповідних URL, портів або логінів у неробочий час.

Окремо варто згадати специфіку розробки кореляційних правил у багатофіліальній організації. За наявності територіально розподілених офісів із різною мережею і політиками журналювання, Splunk може отримувати дані через Universal Forwarder або Heavy Forwarder, з централізованою обробкою на Indexer’ах і виконанням запитів на Search Head. Для зменшення навантаження критично правильно розносити обчислення: прості правила варто виконувати на рівні Indexer, а складні — залишати на Search Head із обмеженим cron-плануванням.

Ще одним важливим аспектом є використання кореляційних правил як інструменту прогнозної аналітики в рамках розширеного моніторингу. Наприклад, за допомогою машинного навчання, яке доступне через Splunk Machine Learning Toolkit, можливо автоматично генерувати шаблони нормальної поведінки для окремих облікових записів або систем. Виявлення відхилень від цих шаблонів може запускати кореляційні сповіщення без необхідності чітко визначати кожен можливий сценарій загрози. Такий підхід, хоч і складніший в налаштуванні, дозволяє виявляти нові вектори атак, що ще не були ідентифіковані у класичних MITRE ATT&CK або для яких ще не існує CVE.

Кореляційні правила у Splunk також застосовуються як компонент складних playbook'ів у рамках інтеграції з платформами SOAR (Security Orchestration, Automation and Response). Наприклад, виявлення запуску шкідливого процесу може запускати ланцюг дій, що включає: перевірку хосту на наявність інших слідів компрометації, сповіщення відповідального аналітика через Slack, створення інциденту у ServiceNow та ініціацію AV-сканування через CrowdStrike API. Такий зв'язок дозволяє скоротити час реагування з годин до хвилин, що особливо важливо при обробці масштабованих атак типу ransomware.

Особливу увагу потрібно приділити викликам, що виникають під час розгортання великої кількості кореляційних правил. По-перше, це навантаження на обчислювальні ресурси: кожен SPL-запит споживає CPU та пам'ять, і при неправильно спроєктованій інфраструктурі може спричинити деградацію продуктивності Splunk. По-друге, некоректно сформульовані правила або дублювання запитів можуть викликати алерт-шторм, коли система генерує сотні непотрібних сповіщень, ускладнюючи роботу аналітиків SOC. Тому важливо реалізовувати внутрішню політику пріоритизації правил за критичністю, частотою спрацювання та перевіреною точністю (наприклад, за метриками TPR/FPR — true positive rate/false positive rate). Splunk дозволяє створити власний фреймворк категоризації правил із поділом на: базові (baseline), детективні (detective),

адаптивні (adaptive), реактивні (reactive), що дозволяє ефективно керувати масштабною логікою кореляції.

Щодо процесу оновлення й оптимізації правил, організації повинні впровадити регулярні цикли перегляду кореляційної логіки. Це особливо важливо з урахуванням динамічної природи загроз — щомісячні CVE, нові TTP від APT-груп, змінена політика доступу в організації тощо. Найбільш ефективним виявляється підхід DevSecOps, коли оновлення правил включається до CICD-пайплайнів, перевіряється в тестовому середовищі, версіонується через Git, і лише після цього застосовується в продуктиві. Такий підхід дозволяє зберігати контроль за історією змін, повертатись до стабільних версій у разі збоїв, та інтегрувати спільну роботу кількох команд — SOC, аналітиків загроз, DevOps та адміністраторів системи.

Ще однією ефективною практикою є візуалізація спрацювань кореляційних правил за допомогою Splunk Dashboard або Glass Table. Це дозволяє у режимі реального часу контролювати, які правила найчастіше генерують події, які хости чи облікові записи згадуються найчастіше, а також визначати «шумні» джерела даних. Згідно з [17], більшість зламаних організацій мали журнальовані події, які могли б сигналізувати про атаку — однак через відсутність кореляції ці події не були помічені. Саме тому візуалізація кореляцій — не лише технічна зручність, а й стратегічний інструмент в операційній безпеці.

В окремих випадках, Splunk може також інтегруватися з зовнішніми платформами аналітики загроз — наприклад, з OpenCTI або ThreatQ — які можуть автоматично оновлювати репутаційні списки, правила поведінки або надавати сигнатури для внутрішнього використання. Це відкриває можливості для динамічного створення кореляційних умов, де правила автоматично адаптуються до нових загроз. Наприклад, новий IP з категорією C2 (command and control) потрапляє у threat feed, автоматично додається до lookup у Splunk, і всі спроби з'єднання з цим IP одразу потрапляють під кореляційне правило.

Наступним логічним кроком після створення кореляційних правил є їх інтеграція у загальний процес виявлення та реагування на інциденти. У середовищі Splunk це реалізується через модуль Enterprise Security (ES), який дозволяє об'єднувати окремі правила у концептуально зв’язані Use Case-и — наприклад, «Compromised Account Detection» або «Suspicious Network Beaconing». Кожен Use Case включає набір правил, метрик і KPI, що дозволяє вимірювати їхню ефективність у динаміці: кількість спрацювань, середній час до реагування (MTTR), кількість помилкових позитивів. Це надзвичайно важливо для управління ризиками та побудови звітності перед керівництвом чи аудиторами [18].

Також значущою функціональністю Splunk є можливість використання макросів (macros), тегів (tags), eventtypes, а також datamodels, які дозволяють будувати більш гнучкі та повторно використовувані правила. Наприклад, кореляційне правило на пошук аномального SSH-доступу може бути виражене як окремий макрос, який потім імпортується в кілька правил. Це не тільки пришвидшує розробку нових детекцій, а й забезпечує їхню узгодженість між різними командами (SOC, SecOps, Threat Hunting).

Важливу роль у точності кореляцій відіграє якість вхідних даних. Наприклад, якщо журнали подій не містять чіткої інформації про джерело, користувача, тип доступу, геолокацію чи хеш файлу — значна частина потенційної аналітики втрачається. Тому однією з практик перед розробкою правил є аудит доступних джерел та їх збагачення. Це може включати додавання geoip-даних, asset inventory, CMDB-зв’язків, даних із cloud API або синхронізацію з AD-групами для поділу користувачів на ризикові категорії.

Приклад: розробка правила виявлення підозрілих з'єднань до зовнішніх IP у неробочий час для критичних обліковок. Тут необхідно об’єднати дані з трьох джерел: (1) firewall-логів, (2) даних про обліковки з AD (з міткою privileged=true), (3) календаря робочого часу (або ручного списку годин).

```
index=firewall_logs dest_ip!=internal_ranges
| lookup privileged_users user OUTPUT privileged
| where privileged=true
| eval hour=strftime(_time,"%H")
| where hour<7 OR hour>20
| stats count by user, src_ip, dest_ip
```

Таке правило є класичним прикладом поведінкової аналітики з врахуванням контексту та часової зони. У разі позитивного спрацювання подія може бути автоматично передана в Splunk SOAR для запуску скрипту блокування IP або тимчасового призупинення облікового запису, з паралельним створенням інциденту в системі обліку.

У великомасштабних інфраструктурах важливо розуміти, що збільшення кількості правил не завжди призводить до зростання ефективності. Згідно з аналітичними звітами [2], найбільш продуктивними вважаються ті організації, які дотримуються принципу «інтелектуальної щільності правил»: менше, але контекстуально влучніші кореляції, з урахуванням конкретної ролі IT-активів у бізнесі. Наприклад, правило на виявлення SQL-ін’єкції у веб-додатку буде абсолютно різним для систем внутрішнього документообігу і для фінансової CRM-системи, яка доступна ззовні. Тому важливо враховувати топологію систем, їх критичність та бізнес-процеси.

З метою підтримки високої якості правил рекомендовано також застосовувати принципи Threat-Informed Defense — тобто тестування правил за допомогою симульованих TTP з MITRE ATT&CK, наприклад, використовуючи Purple Team-інструменти: Red Canary Atomic Red Team, Prelude Operator, Infection Monkey або Caldera. Ці інструменти дозволяють перевірити, наскільки точно створені кореляційні правила відображають реальні загрози — наприклад, чи буде зафіксовано lateral movement, або чи розпізнається credential dumping, виконаний з нестандартного процесу.

Створення та підтримка кореляційних правил повинна бути задокументована — з описом мети, логіки, джерел даних, прикладів спрацювання, відповідальних осіб, та сценаріїв реагування. Така документація дозволяє не лише підтримувати знання в команді SOC, а й пришвидшувати адаптацію нових працівників або аудитів, особливо в умовах динамічної зміни штатного складу або аутсорсингу частини функцій. Splunk дозволяє зберігати цю документацію прямо у вигляді полів мета-даних у кожному правилі або у вигляді окремої таблиці (наприклад, на базі kvstore або lookup).

Таким чином, впровадження кореляційних правил у Splunk — це комплексний процес, що вимагає не лише технічної компетентності у побудові запитів, а й стратегічного бачення безпеки, глибокого розуміння IT-ландшафту, взаємодії з бізнес-процесами, аналітики загроз та постійного вдосконалення. Система, побудована за такими принципами, стає не просто джерелом логів, а повноцінною платформою ситуаційної обізнаності, готовою до проактивної оборони у складному середовищі цифрових ризиків.

У практичному середовищі організації створення кореляційних правил базується на типовому підході до багатошарової логіки виявлення — починаючи від базових технік атак (таких як PowerShell-експлуатації), до складніших ланцюгів дій, характерних для APT-груп. У контексті реального застосування Splunk, одним з перших правил, які впроваджуються, є детекція шкідливих команд у PowerShell.

Наприклад, правило для виявлення команд, пов’язаних з використанням утиліти Mimikatz для викрадення облікових даних, може виглядати наступним чином:

```
index=wineventlog EventCode=4104
| search CommandLine="*Invoke-Mimikatz*"
| stats count by user, host, CommandLine, _time
```

У цьому запиті Splunk звертається до Windows журналу PowerShell Script Block Logging (EventCode 4104) і фільтрує всі команди, які містять згадку Invoke-Mimikatz. Це базовий приклад детекції індикатора компрометації, який можна

удосконалити, додавши порівняння з white-list системними адміністраторами або допустимими сценаріями запуску.

Іншим критичним правилом є виявлення сценарію password spraying. Така атака характеризується множинними спробами входу в систему з різними обліковими записами, але з одним і тим самим паролем. У Splunk це можна реалізувати наступним чином:

```
index=wineventlog EventCode=4625
| stats count by user, src_ip, _time
| where count > 5
```

Це правило аналізує події невдалого входу (EventCode 4625), групує їх за користувачем та IP-адресою джерела й виводить ті, де кількість спроб перевищує поріг у 5. Для підвищення точності його можна розширити, додавши часові рамки або перетворивши на transaction, щоб виявити сесії за проміжок до блокування.

Ще один приклад — виявлення активності, що відбувається у неробочий час, коли нормальна операційна діяльність відсутня. Для цього використовуються eval-функції часу та перевірки на нестандартну поведінку.

```
index=wineventlog EventCode=4624
| eval hour=strftime(_time,"%H")
| where hour<6 OR hour>22
| stats count by user, src_ip, host
```

Тут визначається година входу в систему (EventCode 4624 означає успішний логін), і всі події за межами 6:00–22:00 виносяться на аналіз. Це правило особливо актуальне для організацій із фіксованим робочим графіком, наприклад, банківських структур або урядових установ.

Окрему увагу варто приділити сценаріям lateral movement, зокрема через Remote Desktop Protocol. Підозріла активність у вигляді з’єднань RDP між неадміністративними хостами або з невідомих джерел фіксується через логіку з використанням подій з мережевого журналу або логів Windows:

```
index=wineventlog EventCode=4624 LogonType=10
| lookup asset_inventory host OUTPUT role
| where role!="server" AND src_ip!="known_admin_ip"
| stats count by user, src_ip, dest
```

Таке правило спирається на події входу через RDP (LogonType 10), і перевіряє, чи виконується доступ з IP-адрес, що не належать до службових або адміністративних. Lookup-таблиця asset_inventory дозволяє визначити роль пристрою в інфраструктурі для фільтрації допустимих сценаріїв.

У середовищах, де використовується хмарна інфраструктура, особливо актуальними є правила, що стосуються логів доступу до API, IAM змін, та створення нових токенів доступу. Наприклад, для Google Workspace чи AWS CloudTrail:

```
index=aws_cloudtrail eventName=CreateAccessKey OR eventName=CreateUser
| stats count by userIdentity.arn, sourceIPAddress, eventTime
```

Це дозволяє фіксувати всі випадки створення нових доступів — дій, які часто є індикатором компрометації або початку APT-кампанії (T1078 — Valid Accounts). Якщо додати до цього правило на виявлення трафіку до відомих C2-серверів (за threat intel lookup), ми можемо створити комплексну логіку умовного «ланцюга атаки».

Наступний рівень зрілості кореляційних правил — це зв'язування подій між собою на основі часової послідовності. У Splunk для цього використовується transaction, який дозволяє об’єднати події в один ланцюг за ключовими полями:

```
index=wineventlog (EventCode=4688 OR EventCode=4624)
| transaction user host maxspan=10m
| search process_name="cmd.exe" AND LogonType=3
```

Цей запит об’єднує в одну транзакцію події запуску процесів та логінів, і дозволяє виявити випадки, коли після входу на машину відбувається підозріле виконання командного рядка. Такі сценарії характерні для технік T1059.003 (CMD) і є критичними на етапі постексплуатації.

На завершення важливо зазначити, що кожне правило має супроводжуватися політикою реагування, метаданими (priority, owner, expected FPR), а також проходити валідацію в sandbox-умовах. Тестування кореляційних правил на основі реальних атак (наприклад, через MITRE CALDERA або Atomic Red Team) дозволяє уникнути ситуації, коли в продакшн потрапляють неякісні або шумні умови спрацювання.

Таким чином, логіка побудови кореляційних правил у Splunk базується на інтеграції технічної аналітики, знання TTP атак, внутрішнього контексту організації та перевірених джерел індикаторів. Саме ця багаторівнева інтеграція забезпечує не лише детекцію ізольованих інцидентів, а й виявлення складних, послідовних атак із глибоким зануренням у внутрішнє середовище, даючи змогу організації випереджати загрози на крок вперед.

3.3. Рекомендації щодо використання Splunk для підвищення безпеки систем

У процесі забезпечення інформаційної безпеки організаційного середовища одним із найбільш ефективних підходів є впровадження системи моніторингу та кореляційної аналітики подій, здатної оперативно виявляти та реагувати на загрози. У цьому контексті платформа Splunk виступає не просто як інструмент лог-менеджменту, а як повноцінна SIEM-рішення, здатне стати центральною ланкою у системі захисту інформаційної інфраструктури. Рекомендації щодо її використання для підвищення безпеки систем повинні враховувати архітектурні можливості, гнучкість у налаштуванні, інтеграцію з іншими засобами захисту та відповідність сучасним тактикам атак згідно з MITRE ATT&CK.

Насамперед доцільно акцентувати увагу на важливості створення у Splunk повноцінного ланцюга контролю: від збору та нормалізації даних до автоматизованого реагування. Це передбачає побудову наскрізного пайплайну обробки подій, де джерела даних мають бути репрезентативними для усіх важливих

об’єктів: кінцевих точок, мережевого трафіку, систем аутентифікації, хмарних сервісів, поштових шлюзів тощо. Для цього необхідно правильно конфігурувати механізми Data Inputs, HEC та Universal Forwarder у ролі агентів збору. У рекомендаціях слід враховувати, що чим вища щільність і якість зібраних логів, тим точнішою буде аналітика.

Другою ключовою рекомендацією є створення динамічних кореляційних правил, які базуються не лише на сигнатурних патернах, а й на поведінковому аналізі. Наприклад, у разі виявлення підозрілої активності, пов’язаної зі зміною атрибутів облікового запису у позаробочий час у поєднанні з аномальним географічним підключенням, Splunk повинен запускати розширене оповіщення, що враховує кілька умов. Це реалізується за допомогою мови SPL, у поєднанні з макросами, lookup-таблицями та моделей ризику (risk-based alerting). Такий підхід відповідає концепції багаторівневої аналітики SIEM нового покоління.

Окрім написання складних правил, варто також інтегрувати Splunk із зовнішніми джерелами Threat Intelligence. Це дозволяє збагачувати внутрішні події безпеки контекстом про відомі IOC, шкідливі IP-адреси, домени, цифрові сертифікати або контрольні суми файлів. Найбільш ефективно такі сценарії реалізуються через Splunk Enterprise Security (ES) або додаткові модулі, такі як Threat Intelligence Framework та підтримка таксономії STIX/TAXII [18]. Такий збагачений аналіз дозволяє не лише виявляти загрози, а й проводити ретроспективні розслідування, що підвищує рівень incident response та threat hunting.

Рекомендується також створити власні дашборди із контекстною візуалізацією ключових показників безпеки. Наприклад, варто зосередитись на індикаторах компрометації (IoC), розподілі типів подій за часом, heatmap поведінкових відхилень та детекторів lateral movement. Використовуючи Splunk Dashboard Studio або Simple XML, можна формувати кастомізовані панелі, адаптовані під специфіку галузі (банківська сфера, енергетика, держсектор) та політику безпеки конкретної організації.

Окремо варто надати рекомендації щодо автоматизації дій у відповідь. Splunk може бути інтегрований із платформами SOAR, наприклад, Splunk SOAR або сторонніми рішеннями (IBM Resilient, Cortex XSOAR), що дозволяє реалізувати playbook-сценарії для автоматичної реакції на події. Наприклад, при спрацюванні правила на витік облікових даних, SOAR може ініціювати блокування доступу, відкриття тікета у системі ServiceNow, а також сповіщення команди безпеки через Slack або електронну пошту. Такі механізми не лише знижують час реагування (MTTR), а й мінімізують вплив людського фактору [2].

Наступним важливим аспектом є забезпечення постійної валідації ефективності впроваджених засобів моніторингу. Зокрема, доцільно реалізувати механізми регулярного тестування кореляційних правил та оповіщень шляхом симуляції атак або запуску умовних тригерів у тестовому середовищі. Це дозволяє не лише перевірити працездатність правил, а й адаптувати їх до нових сценаріїв з урахуванням розвитку технік противника, що фіксуються у фреймворку MITRE ATT&CK. Наприклад, методи з підкатегорії “Credential Access” або “Persistence” можуть оновлюватися, і без регулярного апдейту логіки детекції Splunk втрачає релевантність у реальному середовищі атак.

У частині кібергігієни та людського чинника рекомендується використовувати Splunk не лише як інструмент реагування, а й як засіб підвищення обізнаності персоналу. Через систему сповіщень та централізованих дашбордів можна інформувати користувачів про інциденти, що їх безпосередньо стосуються — наприклад, спроби входу в обліковий запис з нової геолокації або виявлені аномальні дії. Такі повідомлення можуть автоматично супроводжуватись інструкціями або запитами на підтвердження дій через MFA чи зміну пароля. Подібна стратегія поєднує технологічну складову з поведінковою адаптацією користувача, що особливо ефективно у середовищах з високою розподіленістю ресурсів (віддалені працівники, філії, підрядники) [2].

Також доцільно впровадити моделі управління ризиками на основі ризик-скорингу подій. Splunk дозволяє кожній події призначати умовну «вагу ризику»

залежно від поєднання факторів: наприклад, одночасне спрацювання правил, які охоплюють різні стадії kill chain, свідчить про високий ступінь загрози. Такі механізми можуть бути підключені до внутрішніх процесів реагування, пріоритизації інцидентів або політик обмеження доступу. Це дозволяє перейти від реактивного реагування до проактивної моделі аналізу ризиків, що відповідає принципам Zero Trust та підходам NIST CSF.

У контексті адаптації до галузевих нормативів рекомендується конфігурувати Splunk відповідно до вимог таких стандартів, як ISO/IEC 27001, PCI DSS, NIS2 або HIPAA (для медичних організацій). Splunk має низку вбудованих App-модулів для контролю відповідності (compliance), які забезпечують візуалізацію статусу аудиту, виявлення невідповідностей та звітність для внутрішніх і зовнішніх перевірок. Наприклад, App PCI Compliance for Splunk дозволяє створити готові аналітичні панелі відповідно до стандарту PCI DSS 4.0, з чіткими індикаторами на основі агрегованих подій.

Окремо заслуговують уваги питання масштабування та продуктивності Splunk при нарощуванні обсягу подій. У рекомендаціях для середніх та великих організацій слід враховувати архітектуру з окремими інстансами Indexer'ів, Search Head Cluster, ліцензування ingestion rate, а також оптимізацію зберігання шляхом використання summary indexing, retention policy або cold storage. Неправильна балансування навантаження або використання неефективних SPL-запитів може призвести до затримок у детекції, що критично для SOC-процесів. Тому рекомендується регулярно аналізувати performance metrics через internal Splunk логіку (index=_introspection) та використовувати App Performance Monitoring (Observability Suite) для виявлення вузьких місць у розгортанні.

У контексті інтеграції з екосистемою безпеки, Splunk може виступати як центральний вузол агрегації подій, синхронізуючи дані з системами DLP, EDR (наприклад, CrowdStrike Falcon), хмарними платформами (AWS, GCP, Azure) та мережевими пристроями (Cisco, Palo Alto, Fortinet). Для цього передбачені стандартні Add-on і TA (Technology Add-ons), які дозволяють нормалізувати логи у

формат CIM (Common Information Model), що є критично важливим для коректної роботи кореляційних правил. Наявність такого інтеграційного шару значно підвищує якість аналітики, дозволяє створювати уніфіковані дашборди та мінімізує ймовірність хибнопозитивних спрацювань.

Ще одним критичним аспектом ефективного використання Splunk для підвищення безпеки систем є організація циклу повного життєвого циклу управління інцидентами — від детекції до пост-інцидентного аналізу. У рамках цього підходу рекомендується будувати структуру взаємодії між Splunk, системами управління кейсами (наприклад, Jira або ServiceNow) та командами SOC. Зокрема, створення кейсу в системі керування інцидентами має ініціюватися безпосередньо із спрацювання певного оповіщення у Splunk. Це дозволяє скоротити час виявлення (MTTD) та реагування (MTTR), а також створити уніфіковану базу знань для наступного аналізу інцидентів, яка доповнює базу сценаріїв і playbook-ів для майбутніх атак.

Також рекомендовано впроваджувати процес post-mortem аналізу загроз із використанням Splunk як джерела хронологічної картини подій. За допомогою функцій transaction, timeline visualization, а також збережених пошукових шаблонів можна реконструювати повний маршрут атаки, виявити слабкі місця в захисті та формалізувати відповідні рекомендації. Такий ретроспективний аналіз має бути частиною систематичної програми вдосконалення захисту — на рівні конфігурацій, політик доступу, архітектурних змін або підвищення обізнаності персоналу [2].

У специфічних середовищах, де важливим є безперервність роботи та стабільність сервісів (наприклад, критична інфраструктура або великі підприємства реального сектору), рекомендується також створити окремі моніторингові дашборди для системної стабільності Splunk. Це можуть бути метрики використання CPU/пам’яті на Indexer’ах, затримки індексації, обсяг даних, що не потрапили до індексу, кількість неуспішних запитів тощо. Такі індикатори можна агрегувати в окремий health dashboard, який буде сигналізувати про деградацію продуктивності до того, як це вплине на ефективність виявлення загроз.

У процесі формування рекомендацій варто враховувати, що захист не може бути ефективним без адекватної аналітики активності користувачів (UEBA — User and Entity Behavior Analytics). Splunk Enterprise Security має вбудовані засоби побудови профілю поведінки користувача, які дозволяють виявляти відхилення від звичного патерну доступу, переміщення в мережі або запитів до критичних сервісів. Наприклад, виявлення того, що користувач бухгалтерського відділу намагається отримати доступ до репозиторію з кодом або бази даних CRM, може бути тригером для запуску розслідування. Такий рівень контекстуальної безпеки дозволяє поєднувати SIEM та Insider Threat Detection без потреби у додаткових системах .

Також актуальним є застосування машинного навчання у контексті Splunk Machine Learning Toolkit (MLTK). З його допомогою можна створювати моделі прогнозування (наприклад, зростання кількості помилкових логінів), кластери аномальної поведінки або трендові аналітичні моделі. Найбільш доцільно застосовувати такі рішення для виявлення low-and-slow атак або багатостадійних сценаріїв, які важко виявити звичайними правилами. Використання таких підходів значно підвищує точність та адаптивність захисту.

На завершення, доцільно звернути увагу на питання кадрової підготовки та внутрішньої кіберкультури. Ефективне використання Splunk потребує не лише технічного впровадження, а й постійного розвитку знань аналітиків, інженерів безпеки та адміністраторів. Рекомендується проводити внутрішні навчання на базі власних даних (table-top exercises), оцінювання ефективності правил, участь у тренінгах Splunk Education (Fundamentals 1/2/3, Security Use Case Labs) та періодичну сертифікацію фахівців (Splunk Certified Power User, Splunk Certified Admin, Splunk Certified Consultant).

Таким чином, використання Splunk для підвищення безпеки інформаційної системи має базуватись на комплексному підході: від технічного налаштування, інтеграцій та аналітики — до внутрішньої організаційної культури реагування, навчання персоналу, розробки сценаріїв реагування та системної оптимізації. Лише

така мультиаспектна стратегія дозволяє повноцінно реалізувати потенціал платформи Splunk як ключового елемента в архітектурі сучасного захисту організаційної інформаційної інфраструктури.

ВИСНОВКИ

У роботі проведено дослідження проблеми виявлення вразливостей в інформаційній системі організації. Визначено мету, завдання, об’єкт і предмет дослідження, а також окреслено актуальні виклики, пов’язані зі зростанням складності IT-інфраструктур і рівня загроз, які стають дедалі більш витонченими. Вразливості залишаються ключовими точками ризику, експлуатація яких може призвести до серйозних наслідків для конфіденційності, цілісності та доступності даних.

Проведено аналіз основних підходів до класифікації вразливостей, методів та засобів їх виявлення. Окрему увагу приділено розмежуванню проактивних і реактивних методів, а також інструментам, що забезпечують аналіз як статичних, так і динамічних характеристик IT-систем. Розглянуто засоби сканування, тестування на проникнення, поведінкову аналітику та кореляцію подій.

Проведено детальне дослідження можливостей платформи Splunk у контексті виявлення вразливостей та реагування на загрози. Проаналізовано її архітектуру, принципи функціонування, механізми збору даних, побудову дашбордів, а також реалізацію кореляційних правил і систем оповіщень. Продемонстровано ефективність Splunk у виявленні таких типових атак, як brute-force, SQL/XSS-ін’єкції, підозрілі IP-з’єднання та аномальна активність користувачів.

Розроблено практичні рекомендації щодо впровадження Splunk в інформаційну інфраструктуру організації: від вибору моделі розгортання (локальна, хмарна, гібридна) до налаштування агентів, індексів, правил доступу, а також інтеграції з платформами SOAR, threat intelligence та зовнішніми джерелами даних. Сформовано підходи до побудови кореляційних правил з урахуванням MITRE ATT&CK, специфіки бізнес-процесів, ризик-скорингу подій та аналізу поведінкових відхилень.

Отже, виявлення вразливостей та кореляційна аналітика стали ключовими елементами сучасної архітектури кібербезпеки організацій. Splunk — це універсальне SIEM-рішення, яке дозволяє централізовано збирати, аналізувати й інтерпретувати події безпеки, автоматизувати реагування на інциденти, будувати ефективні дашборди та забезпечувати повну ситуаційну обізнаність.

Splunk може допомогти організаціям ефективно реалізувати концепцію глибокого захисту (defense-in-depth), відповідати міжнародним стандартам безпеки (ISO/IEC 27001, NIST CSF, PCI DSS), впровадити найкращі практики реагування на інциденти та скоротити час до виявлення (MTTD) і реагування (MTTR) на загрози.

Таким чином, правильне впровадження платформи Splunk у середовищі організації сприяє підвищенню загального рівня кіберстійкості, дозволяє забезпечити ефективний контроль за вразливостями, підвищити точність виявлення загроз та сформувати адаптивну модель інформаційної безпеки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. NIST SP 800-30 Rev. 1 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf (дата звернення: 29.05.2025).

2. ENISA threat landscape 2023 URL: https://www.enisa.europa.eu/sites/default/files/publications/ENISA%20Threat%20Landscape%202023.pdf (дата звернення: 29.05.2025).

3. SP 800-115, technical guide to information security testing and assessment | CSRC. NIST Computer Security Resource Center | CSRC. URL: https://csrc.nist.gov/pubs/sp/800/115/final (дата звернення: 25.05.2025).

4. Threats - microsoft threat modeling tool - azure. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats (дата звернення: 23.05.2025).

5. ENISA threat landscape 2024 URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2024 (дата звернення: 12.03.2025).

6. Vulnerabilities, threats & risk explained | splunk. Splunk. URL: https://www.splunk.com/en_us/blog/learn/vulnerability-vs-threat-vs-risk.html (дата звернення: 29.05.2025).

7. Components of a splunk enterprise deployment - splunk documentation. Documentation - Splunk Documentation. URL: https://docs.splunk.com/Documentation/Splunk/9.4.2/Capacity/ComponentsofaSplunkEnterprisedeployment (дата звернення: 04.06.2025).

8. Threat hunting vs. threat detecting: what's the difference? | splunk. Splunk. URL: https://www.splunk.com/en_us/blog/learn/threat-hunting-vs-threat-detecting.html (дата звернення: 04.06.2025).

9. Brute force attacks: techniques, types & prevention | splunk. Splunk. URL: https://www.splunk.com/en_us/blog/learn/brute-force-attacks.html (дата звернення: 04.06.2025).

10. Splunk docs. Splunk Docs. URL: https://help.splunk.com/en/splunk-enterprise-security-8/administer/8.0/detections/use-detections-to-search-for-threats-in-splunk-enterprise-security (дата звернення: 04.06.2025).

11. Splunk enterprise product features | splunk. Splunk. URL: https://www.splunk.com/en_us/products/splunk-enterprise-features.html (дата звернення: 04.06.2025).

12. Create real-time alerts - Splunk Documentation. Documentation - Splunk Documentation. URL: https://docs.splunk.com/Documentation/Splunk/9.4.2/Alert/DefineRealTimeAlerts (дата звернення: 04.06.2025).

13. Splunk | The Key to Enterprise Resilience. URL: https://www.splunk.com/en_us/pdfs/solution-guide/splunk-dashboards-quick-reference-guide.pdf (дата звернення: 04.06.2025).

14. Splunk. Data collection architecture. Splunk Lantern. URL: https://lantern.splunk.com/Splunk_Success_Framework/Platform_Management/Data_collection_architecture (дата звернення: 04.06.2025).

15. Overview of the Splunk Common Information Model - Splunk Documentation. Documentation - Splunk Documentation. URL: https://docs.splunk.com/Documentation/CIM/latest/User/Overview (дата звернення: 04.06.2025).

16. Splunk. Assessing and expanding MITRE ATT&CK coverage in Splunk Enterprise Security. Splunk Lantern. URL: https://lantern.splunk.com/Security/UCE/Guided_Insights/Cyber_frameworks/Assessing_and_expanding_MITRE_coverage_in_Splunk_Enterprise_Security (дата звернення: 04.06.2025).

17. Verizon: Wireless, Internet, TV and Phone Services | Official Site. URL: https://www.verizon.com/business/resources/reports/2024-dbir-executive-summary.pdf (дата звернення: 05.03.2025).

18. Implement security use cases using the Use Case Library in Splunk Enterprise Security - Splunk Documentation. Documentation - Splunk Documentation. URL: https://docs.splunk.com/Documentation/ESCU/5.6.0/user/UCLKeyFeatures (дата звернення: 04.06.2025).