Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра інформаційної та кібернетичної безпеки
імені професора Володимира Бурячка

«Допущено до захисту»

Завідувач кафедри інформаційної та
кібернетичної безпеки імені
професора Володимира Бурячка
кандидат технічних наук, доцент
Складанний П.М.

_____
(підпис)
« ___ » _____ 20__ р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття другого (магістерського)
рівня вищої освіти

Спеціальність 125 Кібербезпека та захист інформації

Тема роботи:

РОЗРОБКА ТЕСТОВОГО СЕРЕДОВИЩА ДЛЯ ПОБУДОВИ СИСТЕМИ ЗАХИСТУ ДАНИХ НА РІВНІ ДОДАТКІВ В ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІЙ СИСТЕМІ КОРПОРАТИВНОЇ МЕРЕЖІ

Виконав
студент групи КБм-1-БІКСм1-24-1.4.д
Скуратовський Євгеній Олегович
(прізвище, ім’я, по батькові)

_____
(підпис)
Науковий керівник
к.в.н., доцент
(науковий ступінь, наукове звання)
Аносов А. О.
(прізвище, ініціали)

_____
(підпис)

Київ – 2025

ЗМІСТ

ВСТУП

Сучасні корпоративні мережі є однією із основ функціонування підприємства та організації у різних сферах економіки, що забезпечує швидкий доступ до інформації, ефективну комунікацію та інтеграцію бізнес процесів. Разом із зростом цифрових технологій, обсягом оброблюваних даних та рівнем взаємозв'язку додатків суттєво підвищуються ризики, які пов'язані із захистом інформації. Основними загрозами є неасоційований доступ до корпоративних ресурсів, витоки конфіденційних даних, експлуатація вразливостей у програмному забезпечення, а також атаки типу SQL-ін'єкції, міжсайтового скприптингу (XSS) та підміни запитів (CSFR).

Безпека на рівні відіграє досить критичну та важливу роль у загальній стратегії кіберзахисту, адже саме в цьому сегменті часто можуть відбуватися атаки на дані користувачів. Традиційні методи забезпечення безпеки, а саме мережеві брандмауери та антивірусне програмне забезпечення не можуть повністю захищати корпоративні мережі від атак, які спрямовані на вразливості у веб-додатках, API-інтерфейсах та мікросерверній архітектурі. Саме тому починає зростати необхідність у впровадженні комплексних методів захисту на рівні додатків, зокрема використання механізмів контролю доступу, шифрування, вияв та запобігання вразливостям, моніторинг активності користувачів та реагування на загрози у реальному часі.

Метою роботи є розроблення макету тестового середовища та методу оцінювання ефективності механізмів захисту корпоративних мереж на етапі їх проєктування, що дозволяє визначати релевантні інструменти та підвищувати рівень інформаційної безпеки шляхом експериментального тестування.

Для досягнення поставленої мети необхідно виконати такі завдання:

1. Проаналізувати ключові загрози інформаційній безпеці корпоративних мереж, що виникають на етапі проєктування та розгортання

систем.

2. Дослідити існуючі механізми та інструменти захисту, що застосовуються в корпоративних мережах у різних умовах їх функціонування, та визначити їх сильні й слабкі сторони.

3. Визначити обмеження існуючих рішень і сформувати набір механізмів, що підлягатимуть тестуванню на розробленому макеті.

4. Розробити метод оцінювання ефективності механізмів захисту, включаючи систему критеріїв, показників та шкал, а також створити макет тестового середовища для проведення експериментів.

5. Здійснити експериментальне тестування обраних механізмів на макеті, зібрати кількісні показники їх ефективності, провести оцінювання та узагальнення результатів, і на їх основі сформувати раціональні рекомендації щодо застосування захисних механізмів у корпоративних мережах.

Об’єктом дослідження є механізми забезпечення безпеки інформації на рівні додатків у корпоративних інформаційно-комунікаційних системах.

Предметом дослідження є методи захисту даних, що будуть використані в корпоративних додатках, їх ефективність, можливості впровадження та вплив на продуктивність системи.

Під час дослідження буде використано такі методи: вивчення літературних джерел та нормативних документів – аналіз сучасних стандартів, рекомендацій та наукових робіт у галузі безпеки корпоративних додатків; комплексна оцінка загроз – виявлення можливих атак на рівні додатків, таких як SQL-ін'єкції, XSS, CSRF, атаки на API та інші; порівняння методів захисту – дослідження та аналіз ефективності існуючих рішень, таких як IDS/IPS системи, контроль доступу, шифрування, SIEM та інші; експериментальний підхід – створення тестового середовища для перевірки ефективності впроваджених заходів безпеки; моделювання загроз – тестування можливих атак та аналіз реакції системи на різні типи загроз; статистичний аналіз – обробка та аналіз зібраних даних для

оцінки ефективності впроваджених заходів безпеки.

Наукова новизна роботи полягає у проведенні всебічного аналізу ефективності сучасних методів захисту корпоративних додатків, що дозволить порівняти їх з традиційними підходами, розроблення методику оцінки ефективності механізмів захисту з урахуванням їхнього впливу на продуктивність системи, створення моделі реальних загроз та визначено оптимальні заходи безпеки для захисту даних на рівні додатків у корпоративних мережах.

Результати дослідження можуть бути використані у майбутньому, а саме для вдосконалення політик безпеки корпоративних мереж та додатків, у процесі розробки нових інформаційних систем із підвищенням рівнем безпеки, для оцінки ефективності існуючих механізмів захисту та їх оптимізації, у навчальних програмах з кібербезпеки для підготовки фахівців у даній сфері.

Отримані результати можна буде застосувати у корпоративних інформаційних системах для покращення кібербезпеки, фінансових установах для захисту конфіденційних даних клієнтів, державних установах для підвищення рівня захищеності інформаційних систем, IT-компаніях для розробки більш безпечних програмних рішень.

Основні положення та результати можуть бути представлені на наукових конференціях, у публікаціях та впровадження в навчальні курси з кібербезпеки. Дана робота спрямована на аналізі сучасних методів захисту корпоративних додатків, оцінці їхньої ефективності та розробку рекомендацій щодо впровадження ефективних механізмів безпеки, що буде сприяти зниженню рівня кіберзагроз у корпоративному середовищі.

РОЗДІЛ 1 ТЕОРЕТИЧНІ ПРИНЦИПИ ТА АНАЛІЗ СУЧАСНИХ РІШЕНЬ У СФЕРІ ЗАХИСТУ ДОДАТКІВ

1.1 Основні загрози безпеці на рівнях додатків

Сучасні інформаційно-комунікаційні системи є невід'ємною частиною діяльності підприємств, державних установ та організацій, що обробляють великі обсяги конфіденційної інформації. Вразливості на рівні додатків завжди залишалися однією із найбільших небезпечних загроз у сфері кібербезпеки, адже саме на даному рівні здійснюється доступ до важливих даних, облікових записів користувачів і критично важливих функцій системи. Загрози безпеки додатків можуть мати різні характери та виникати через низку причин, а саме: недостатньою увагою до безпеки під час розробки програмного забезпечення, використання застарілих технологій та вразливих бібліотек, відсутність ефективних механізмів автентифікації та контролю доступу та недостатній захист під час передавання даних між клієнтом та сервером. Саме тому, буде розглянуто основні загрози, які можуть виникнути на рівні додатків, їхні причини та потенційні наслідки для корпоративних інформаційно-комунікаційних систем.

Однією із небезпечних атак є ін'єкційні (SQL Injection, NoSQL Injection, Command Injection), що можуть виникати через недостатню перевірку введених користувачем даних. Як приклад, SQL-ін'єкції дають дозвіл зловмисникам впровадити шкідливий SQL-код у запити до бази даних, отримавши несанкціонований доступ до інформації, змінивши або навіть видаливши її. Основними причинами ін'єкційних атак є відсутність належної фільтрації введених атак, використання динамічно сформованих SQL-запитів без параметризованих запитів або підготовлених виразів, відсутність обмеження привілеїв для облікових записів бази даних. Наслідками SQL-ін'єкцій можуть бути:

- Викрадення конфіденційних даних (імена користувачів, паролі, фінансові інформації),
- Повне видалення або зміна даних у базі,
- Захоплення контролю над сервером за допомогою командних ін’єкцій [1, с. 25].

Міжсайтові скриптинги XSS це атаки типу XSS (Cross-Site Scripting), які спрямовані на впровадження шкідливого коду (JavaScript) у веб-сторінку, яку переглядає користувач. Це може дозволяти зловмиснику викрадати дані сесії, змінювати зміст сторінки або навіть здійснити фішингові атаки. Є три основних типи XSS-атак:

- Reflected XSS – це шкідливий код, який передається через URL-параметри та можна виконати в браузері жертви,
- Stored XSS – ще один шкідливий код, який зберігається у базі даних і можна виконати кожного разу, коли користувач переглядає скомпрометовану сторінку,
- DOM-based XSS – це атака, яка відбувається на рівні клієнта через маніпуляцію об’єктною моделюю документа (DOM).

Основними причинами XSS-атак є відсутня фільтрація та екранування вхідних даних, а також недостатній контроль вихідних даних перед їх відображенням у браузері. Наслідками XSS-атак є викрадання сесій користувач та облікових даних, маніпуляція контентом-сторінки та перенаправлення користувачів на фальшиві сайти для фішингу [2, с. 47].

Підробка міжсайтових запитів Cross-Site Request Forgery (CSRF) – це одна із атак, під час якої користувач може виконати несанкціоновані дії на веб-додатку, до якого він вже авторизований. Механізмом атаки є те, що користувач входить у свій аккаунт на веб-сайті (як приклад, банківський портал) і тим самим

зловмисник надсилає користувачу шкідливе посилання, або може навіть вбудувати його в інший сайт. Таким чином, якщо користувач натискає на це посилання, то браузер автоматично може виконати запит від його імені (наприклад, переказ коштів на рахунок зловмисника). Основними причинами таких атак як CSRF є відсутність механізмів перевірки запитів (CSRF-токенів) та автоматичне включення куків у всі HTTP-запити. Наслідками CSRF-атак є виконання фінансових транзакцій без угоди користувача, зміна налаштування облікового запису, захоплення контролю над користувачем у системі.

Багато додатків починають використовувати сторонні бібліотеки та фреймоворки, а саме вони можуть містити невиправлені вразливості. Тому якщо ці компоненти не оновлювати вчасно, то зловмисники починають користуватися відомими експлойтами для отримання доступу до системи. Наслідками використання вразливих компонентів є виконання довільного коду на сервері, викрадання конфіденційної інформації та підвищення привілеїв користувача та отримання доступу до критичних даних.

Отже, якщо контроль доступу до ресурсів реалізовано неправильно, зловмисники мають можливість отримати перегляд даних інших користувачів, виконати адміністративні дії без відповідних прав, ввести несанкціоновані зміни у систему. Основними причинами проблем із контролем доступу є відсутність перевірки рівня привілеїв користувача та використання передбачуваних URL або параметрів запитів для доступу до закритих розділів сайту. Після цього виникають наслідки порушень контролю доступу, а саме витік конфіденційної інформації, захоплення контролю над додатками та порушення цілісності даних.

Таким чином, захист додатків від сучасних загроз вимагає надзвичайно важливого комплексного підходу, що буде включати використання перевірених методів захисту, регулярний моніторинг безпеки та оновлення програмного забезпечення. Дослідження у даній сфері є досить актуальним, адже кіберзлочинці кожного дня розробляють нові способи атак.

У сучасних корпоративних мережах рівень складності загроз постійно зростає, що зумовлює потребу у використанні структурованих і загальновизнаних підходів до забезпечення інформаційної безпеки. Найбільш поширеними та впливовими міжнародними стандартами і концепціями є OWASP, NIST Cybersecurity Framework та Zero Trust Architecture, які формують методологічну основу для побудови ефективних механізмів захисту на рівні додатків.

OWASP (Open Web Application Security Project) – це найбільш відомий міжнародний проєкт, що встановлює відкриті стандарти безпеки веб-додатків. Основним елементом є OWASP Top-10 – перелік найбільш критичних категорій вразливостей, що регулярно оновлюється відповідно до сучасних загроз. Стандарт охоплює такі типові проблеми, як ін’єкції, порушення контролю доступу, небезпечна конфігурація, уразливості в компонентам та інші. Практичні рекомендації OWASP широко застосовуються на етапах проєктування, тестування та аудиту корпоративних систем, а також використовуються як критерії оцінювання під час аналізу ефективності захисту на рівні додатків.

NIST Cybersecurity Framework (NIST CSF) є системним підходом до організації процесів кібербезпеки на рівні підприємств. Структура NIST CSF включає п’ять взаємопов’язаних функцій: виявлення (Identify), захист (Protect), виявлення атак (Detect), реагування (Respond) та відновлення (Recover). Ці функції утворюють життєвий цикл кібербезпеки, що дозволяє оцінювати зрілість системи захисту та узгоджувати технічні, організаційні й процедурні заходи. NIST CSF використовується як універсальна рамка для оцінки ризиків, розроблення політик безпеки та стандартизації процесів кіберзахисту в корпоративних середовищах.

Zero Trust Architecture (ZTA) – сучасна концепція побудови систем інформаційної безпеки, яка базується на принципі «не довіряй нікому за замовчуванням». На відміну від традиційних моделей, що передбачають довіру

всередині корпоративного периметра, Zero Trust вимагає постійної перевірки доступу, мінімізації привілеїв та сегментації мережі. Ключовими елементами підходу є багатофакторна автентифікація, поведінкова аналітика, контроль мережевої взаємодії та безперервний моніторинг. Така модель особливо ефективна в розподілених інфраструктурах, хмарних середовищах та у сценаріях з високими вимогами до конфіденційності даних.

Загалом, зазначені концепції створюють теоретичне підґрунтя для оцінювання безпеки сучасних корпоративних мереж. Вони визначають ключові принципи та критерії, на основі яких здійснюється подальший аналіз ефективності захисних механізмів і обґрунтовується необхідність розроблення тестового середовища для моделювання реальних загроз та перевірки захисних рішень на етапі проєктування.

Отже, загрози на рівні додатків залишаються одним із найкритичніших векторів атак у корпоративних інформаційно-комунікаційних системах. До основних ризиків належать SQL-ін’єкції, XSS, CSRF, використання вразливих компонентів та неправильна реалізація контролю доступу. Їхня ефективна експлуатація зловмисниками може призвести до витоку даних, компрометації облікових записів і порушення цілісності системи. Це підкреслює необхідність впровадження комплексного підходу до захисту на рівні додатків, який включає не лише технічні, а й організаційні та процесні заходи безпеки.

1.2. Аналіз існуючих рішень та інструментів для забезпечення безпеки корпоративних мереж

Аналіз існуючих рішень є спрямованим на забезпечення захисту корпоративних мереж та застосунків [2]. На відміну від традиційного огляду загроз, акцент робиться на впливі типових атак на функціонування додатків та корпоративної інфраструктури, а також на тому, які інструменти й механізми

безпеки дозволяють мінімізувати ризики їх реалізації. Типові загрози, такі як ін’єкційні атаки, порушення контролю доступу, міжсайтові скрипти чи експлуатація вразливостей конфігурації, можуть спричиняти не лише порушення роботи застосунків, але й значні фінансові збитки та витоки конфіденційної інформації. Саме тому важливо оцінювати не тільки характер потенційних атак, але і ефективність конкретних механізмів захисту, які застосовуються на рівні додатків, серверної частини та мережевої інфраструктури.

Одним із найважливіших засобів захисту є валідація та фільтрація вхідних даних, оскільки невірно оброблені або неперевірені дані можуть призвести до SQL-ін’єкцій та інших типів впровадження шкідливого коду. До ефективних методів належать використання білих списків, екранування спеціальних символів та централізовані механізми перевірки даних. Згідно з дослідженнями, правильна валідація може запобігати більшості поширених атак на додатки [3, с. 18].

Важливим елементом захисту є багатофакторна автентифікація (MFA), що дозволяє значно знизити ризик несанкціонованого доступу навіть у разі компрометації одного з факторів. Поєднання пароля, одноразових токенів або біометрії значно підвищує рівень безпеки системи [4, с. 47]. Для забезпечення конфіденційності даних застосовуються криптографічні протоколи TLS/SSL, а також сучасні алгоритми шифрування, такі як AES-256. Оскільки використання застарілих криптографічних рішень може спричинити компрометацію даних, підприємствам необхідно регулярно оновлювати протоколи та дотримуватися рекомендацій щодо їх використання.

Одним із критично важливих аспектів є регулярне оновлення програмних компонентів, адже експлуатація відомих уразливостей у застарілих системах становить один із найбільших ризиків для корпоративних мереж. Автоматизація оновлень та впровадження систем моніторингу дозволяє своєчасно усувати загрози та контролювати стан інфраструктури. Сучасні інструменти кіберзахисту можна класифікувати на кілька груп. SAST (Static Application Security Testing) –

інструменти цього класу аналізують вихідний код без запуску програми (SonarQube, Checkmarx, Veracode). Вони дозволяють виявляти структурні вразливості та помилки програмування, але не здатні виявляти логічні дефекти, що проявляються лише в динаміці.

DAST (Dynamic Application Security Testing) – DAST-рішення перевіряють уже працюючий застосунок методом імітації реальної атаки (Burp Suite, Acunetix, OWASP ZAP). Їх перевагою є можливість оцінити фактичну реакцію додатка на різні типи запитів, але вони не мають доступу до внутрішнього коду. IAST (Interactive Application Security Testing) – ці інструменти поєднують SAST і DAST, забезпечуючи комплексний аналіз (Contrast Security). Недоліком є складність інтеграції та висока вартість впровадження. Інструменти аналізу залежностей – OWASP Dependency-Check та WhiteSource аналізують програмні бібліотеки й фреймворки, які можуть містити відомі уразливості, що часто стають об’єктами атак.

IDS/IPS (системи виявлення та запобігання вторгнень) – IDS аналізує мережевий трафік і повідомляє про аномальну активність, тоді як IPS може автоматично блокувати потенційні атаки. Основною проблемою цих систем залишається значний обсяг хибнопозитивних спрацювань, що ускладнює оцінку їх реальної ефективності.

SIEM, XDR та SOAR-системи – до цієї групи належать рішення для комплексного моніторингу та реагування на інциденти: SIEM – Splunk, Wazuh, IBM QRadar; XDR – Microsoft Defender XDR, Palo Alto Cortex XDR; SOAR – Splunk Phantom, IBM Resilient. Ці системи забезпечують кореляцію подій, аналіз загроз та автоматизовану реакцію, але вимагають значних ресурсів та правильного налаштування. Сучасний підхід DevSecOps інтегрує безпеку у всі етапи життєвого циклу розробки, включаючи: автоматизацію перевірок безпеки (CI/CD), безперервний моніторинг, спільну роботу розробників, тестувальників і аналітиків. Популярні інструменти DevSecOps включають GitHub Dependabot,

GitLab Security та Jenkins Security Plugins.

Для практичної оцінки ефективності застосовуються такі критерії, як точність, рівень хибнопозитивних спрацювань, продуктивність, інтеграція та зручність використання. У таблицях 1.1 та 1.2 нижче наведено порівняння найбільш поширених інструментів.

Таблиця 1.1

Порівняльна характеристика інструментів для аналізу вразливостей

Назва	**False-positive**	**Виявлення загроз**	**Інтеграція**	**Продуктивність**	**Налаштування**	**Зручність**	**Загальна оцінка**
Burp Suite Pro	Мінімальні	Висока	API, CI/CD	Висока	Гнучке (розширення)	Інтуїтивна	6/6
ZAP Proxy	Середні	Висока	Jenkins, GitHub	Середня	Обмежена	Проста	4/6
AppScan	Мінімальні	OWASP Top 10	DevOps	Висока	Гнучка	Складна частково	5.5/6
Acunetix	Мінімальні	Висока	CI/CD	Середня	Базова	Інтуїтивна	5/6

Burp Suite Pro є одним з найточніших інструментів з розширеною автоматизацією та можливістю ручного аналізу. ZAP Proxy – open-source рішення від OWASP, менш продуктивне у великих CI/CD середовищах. AppScan (IBM) забезпечує найвищий рівень інтеграції у DevOps, проте складніший у використанні. Acunetix орієнтований на SMB-сегмент, з інтуїтивним інтерфейсом та широкою підтримкою сканування веб-програм.

Таблиця 1.2

Порівняльна характеристика систем моніторингу подій та реагування (SIEM)

Назва	**Інтеграція**	**Глибина аналізу**	**Реакція на події**	**Масштабування**	**Зручність**	**Загальна оцінка**
Splunk	Відмінна	Дуже висока	Автоматична	Висока	Потужна, складна	4.5/5

Продовження таблиці 1.2

Wazuh	Висока	Середня	Часткова	Висока	Інтуїтивна	3.5/5

SolarWinds SEM	Висока	Висока	Автоматизована	Обмежена	Середня	4/5
LogRhythm	Висока	Середня	Гнучка	Середня	Помірна складність	3.5/5

Splunk є промисловим стандартом для SIEM, забезпечує глибоку аналітику, автоматизацію та масштабованість, але потребує експертного налаштування. Wazuh – open-source альтернатива з хорошою інтеграцією, але базовим аналізом. SolarWinds SEM – баланс аналітики й реакції, але має обмеження у масштабуванні. LogRhythm – ефективна альтернатива, особливо для середніх підприємств.

Таким чином, сучасні методи та технології захисту даних на рівні додатків забезпечують широкий спектр можливостей — від базових механізмів перевірки даних та криптографічного захисту до комплексних систем виявлення та реагування на загрози. Інструменти SAST, DAST, IAST, IDS/IPS, SIEM та DevSecOps-підходи дозволяють забезпечити багаторівневу безпеку корпоративних систем. Однак ефективність цих засобів значною мірою залежить від архітектури мережі, налаштувань, навантаження та специфіки застосунків, що створює потребу у подальшому стандартизованому оцінюванні та порівнянні рішень у контрольованих умовах.

1.3. Обмеження сучасних підходів та обґрунтування необхідності створення тестового макету

У сучасному світі інформаційної безпеки існує дуже багато різних методів та технологій захисту на рівні додатків. Проте, попри їх широке застосування, ці рішення мають і певні недоліки та обмеження, які впливають на їх ефективність та надійність. Криптографія є одним із фундаментальних елементів захисту інформації, адже вона забезпечує конфіденційність та цілісність системи. Однак,

деколи навіть найсучасніші криптографічні алгоритми мають свої обмеження. Зокрема, складність керування ключами може призводити до компрометації системи безпеки. Окрім того, криптографічні методи захищають дані під час їх обробки в незашифрованому вигляді, що створює певні потенційні вразливості [7, с. 12].

Антивірусні програми є основним та важливим засобом для захисту від шкідливого програмного забезпечення. Проте вони мають низку недоліків. По-перше, антивірусні програми потребують регулярних оновлень для виявлення нових загроз, що може споживати значний обсяг інтернет-трафіку. По друге, саме сучасні методи обфускації та упаковки шкідливих програм дають дозвіл зловмисникам обходити антивірусний захист, зробивши відомі віруси невидимими для антивірусного ПЗ. Автоматизоване тестування безпеки додатків, водночас статичного (SAST) та динамічного (DAST) аналізу, широко використовують для виявлення вразливостей. Проте дані методи мають також певні обмеження. SAST, наприклад, спочатку аналізує вихідний код, але не завжди здатний виявити вразливості, що виникають під час виконання програми. DAST, у свою чергу, проводить тестування працюючого додатку, але може пропустити проблеми, які пов’язані з внутрішньою логікою коду. Крім того, впровадження автоматизованого тестування в Agile-процеси розроблення може бути складним через необхідність значних початкових витрат та ресурсів [8].

Системи IDS/IPS призначені для моніторингу мережевого трафіку та виявлення потенційних загроз. Проте і вони мають певні обмеження, такі як висока кількість хибних спрацьовувань, що може призводити до ігнорування реальних загроз. Крім цього, саме ці системи можуть бути досить вразливими до атак на самі журнали реєстрації подій, що починає ускладнювати виявлення та наліз інцидентів. Багатофакторна автентифікація значно підвищує рівень безпеки, вимагаючи від користувача надання декількох форм підтвердження особи. Проте, навіть MFA не є панацеєю. Зловмисники можуть використати певні

методи соціальної інженерії для отримання необхідних даних або експлуатувати вразливості в реалізації MFA. Окрім цього, впровадження багатофакторної автентифікації може ускладнювати користувацький досвід, що іноді може призводити до небажання користувачів використати саме такі системи [9].

Регулярне оновлення програмного забезпечення є критично важливим для забезпечення безпеки додатків. Проте процес оновлення має свої виклики. По-перше, не всі користувачі своєчасно встановлюють оновлення, залишаючи системи вразливими. По-друге, деякі оновлення можуть викликати проблеми сумісності або містити нові вразливості, що потребує додаткового тестування перед впровадженням. Інтеграція безпеки в процес розробки програмного забезпечення, відома як DevSecOps, спрямована на забезпечення безпеки на всіх етапах життєвого циклу розробки. Однак цей підхід має свої обмеження. Впровадження DevSecOps вимагає значних змін у культурі та процесах команди, що може зустрічати опір з боку розробників. Крім того, необхідність навчання персоналу та впровадження нових інструментів може вимагати додаткових ресурсів та часу.

Інструменти автоматизованого тестування безпеки, такі як SAST та DAST, є важливими компонентами забезпечення безпеки додатків. Проте вони не завжди здатні виявляти всі можливі вразливості. Деякі складні логічні помилки або специфічні вразливості можуть залишатися непоміченими автоматичними засобами тестування, адже вони орієнтовані на пошук поширених загроз, таких як SQL-ін’єкції або XSS, але не завжди вони є ефективні для складних логічних помилок у бізнес-логіці додатка. Окрім цього, використання автоматизованих інструментів дуже часто починає потребувати тонкого налаштування. Як приклад, якщо рівень чутливості до вразливостей встановлений занадто високим, то тоді система може генерувати багато хибнопозитвиних спрацювань, що як раз таки ускладнює аналіз отриманих результатів. З іншого боку, саме недостатньо чутливі налаштування можуть пропустити критичні вразливості [10, с. 225].

Як свідчить аналіз рішень Burp Suite, AppScan, Splunk, їх ефективність у виявленні загроз залежить не лише від глибини аналізу вразливостей, а й від додаткових характеристик, таких як масштабованість у корпоративному середовищі, рівень хибнопозитивних спрацювань (false positives), а також можливість гнучкої інтеграції у CICD/DevOps-процеси. Наприклад, Burp Suite дозволяє кастомізувати виявлення загроз під конкретні сценарії через власні скрипти, а AppScan має потужну базу сигнатур для роботи з OWASP Top 10. SIEM-рішення типу Splunk не лише виявляють інциденти, але й можуть автоматизувати відповідь на них через інтеграцію з XDR/SOAR. Однак, висока складність налаштування та потреба у досвідчених спеціалістах є стримувальним чинником впровадження таких систем на підприємствах малого та середнього бізнесу. Тому, при виборі механізмів безпеки слід брати до уваги не лише технічні характеристики, але й ресурсні можливості організації.

Попри широкий спектр доступних інструментів для забезпечення інформаційної безпеки, їх застосування має низку обмежень, які ускладнюють вибір раціональної моделі захисту для конкретної корпоративної мережі. Різні інструменти дають різні результати – точність виявлення вразливостей, швидкість реагування та кількість хибнопозитивних спрацювань суттєво залежать від конкретних налаштувань і середовища, у якому застосовується той чи інший механізм захисту [29-36]. Неможливо оцінювати інструменти в рівних умовах – кожна корпоративна мережа має власну архітектуру, унікальний набір сервісів, специфічне навантаження, різні версії компонентів і різні вимоги до продуктивності. Через це неможливо коректно порівняти ефективність рішень, використовуючи лише їх паспортні характеристики.

Відсутні стандартизовані критерії оцінювання – більшість сучасних рішень надає якісні оцінки («високий ризик», «низький рівень загрози»), але не дає кількісних метрик, таких як: час виявлення інциденту, швидкість реагування, вплив на продуктивність, кількість успішних/блокованих атак на одиницю часу.

Не існує інструмента для тестування рішень до впровадження – підприємства не можуть точно визначити, наскільки ефективним буде той чи інший засіб безпеки до його фактичного встановлення, що підвищує ризики та витрати.

Таким чином, виникає потреба у створенні тестового макету, який дозволить: моделювати корпоративну мережу у контрольованих умовах, відтворювати типові атаки та сценарії навантаження; порівнювати інструменти за однаковими критеріями, визначати найбільш релевантні механізми захисту для конкретних умов, мінімізувати витрати підприємства на вибір неефективних рішень. Саме ця потреба обґрунтовує необхідність подальшої розробки і побудови тестового середовища, що стане основою для експериментального дослідження у наступних розділах.

Висновки до 1 розділу

У першому розділі було проведено системний аналіз сучасних підходів до забезпечення безпеки додатків у корпоративних інформаційно-комунікаційних системах. Розглянуті міжнародні стандарти та концепції (OWASP, NIST CSF, Zero Trust) показали, що сучасна парадигма захисту ґрунтується на принципах багаторівневості, безперервного моніторингу та мінімізації довіри у взаємодії між компонентами системи.

Проведений аналіз існуючих рішень продемонстрував, що для протидії основним загрозам на рівні додатків застосовуються різні групи інструментів: SAST, DAST, IAST, системи контролю залежностей, а також IDS/IPS, SIEM, XDR та DevSecOps-підходи. Кожен із цих механізмів забезпечує певний рівень захисту, дозволяючи виявляти вразливості на різних етапах життєвого циклу програмного забезпечення та реагувати на інциденти у корпоративних мережах. Разом із тим аналіз показав значні відмінності у можливостях, точності та продуктивності таких засобів, що підтверджують залежність їх ефективності від

архітектури середовища, навантаження та особливостей інтеграції.

Було встановлено, що сучасні засоби кіберзахисту мають низку обмежень: різні інструменти дають неоднакові результати, не існує уніфікованих критеріїв порівняння їхньої роботи, а також відсутні стандартизовані умови тестування, які б дозволяли об’єктивно оцінювати їх продуктивність та вплив на корпоративні системи. Ці фактори значно ускладнюють вибір оптимального інструментарію для конкретної організації та можуть призводити до надмірних витрат або недостатнього рівня захисту.

Отже, результати аналізу доводять необхідність створення спеціального тестового макета, який дозволить моделювати різні сценарії атаки, проводити порівняльне випробування інструментів безпеки та визначати їх ефективність у контрольованих і рівнозначних умовах. Саме розроблення такого макета та його використання для оцінки методів захисту додатків становлять основу подальших етапів дослідження.

РОЗДІЛ 2 ОБҐРУНТУВАННЯ ТЕОРЕТИЧНИХ ТА МЕТОДОЛОГІЧНИХ ОСНОВ

2.1 Методика оцінювання ефективності інструментів

Ефективна оцінка засобів захисту на рівні додатків потребує формалізованого підходу, який дозволяє кількісно порівнювати різні інструменти в однакових умовах тестування. На відміну від загального теоретичного огляду або опису інструментів, методика оцінювання має ґрунтуватися на стандартизованих показниках, математичних моделях та уніфікованій процедурі розрахунків, що забезпечує об’єктивність і відтворюваність результатів.

У цьому дослідженні пропонується використовувати інтегрований індекс ефективності E, який включає найважливіші характеристики роботи засобів кіберзахисту. Індекс дозволяє отримати узагальнений показник та порівнювати інструменти між собою за спільною шкалою.

Для оцінювання враховано чотири основні критерії, що відповідають сучасним вимогам до захисту додатків та рекомендовані міжнародними фреймворками (NIST CSF, ISO/IEC 27001, OWASP ASVS):

1. D – швидкість виявлення інцидентів, інцидентів/хв – відображає оперативність реагування інструмента на підозрілу активність. Чим більшою є швидкість виявлення, тим вищою є ефективність системи моніторингу.

2. FPR – частка хибнопозитивних спрацювань, % – показує відсоток некоректних попереджень. Високий FPR призводить до перевантаження аналітиків, зниження довіри до системи та втрати часу.

3. P – вплив інструмента на продуктивність системи, % падіння throughput – визначає, наскільки засіб безпеки навантажує систему та уповільнює обробку запитів.

4. I – інтеграційна оцінка (0–1) – відображає здатність інструмента інтегруватися з існуючою інфраструктурою (CI/CD, логування, оркестрація, API, DevOps-процеси).

Одиниці вимірювання таких показників, як MTTD, MTTR, швидкість аналізу, FPR, відповідають міжнародним практикам оцінювання ефективності [11]. Додатково при аналізі враховуються метрики операційної ефективності, такі як: MTTD (Mean Time to Detect) – середній час виявлення інциденту, хвилини; MTTR (Mean Time to Respond) – середній час реагування, хвилини або години. Ці метрики не входять безпосередньо у формулу інтегрального індексу, але використовуються для допоміжного аналізу у подальших розділах.

Оскільки різні параметри вимірюються в різних одиницях, попередньо вони нормалізуються до інтервалу [0; 1]. Нормалізація здійснюється за формулою:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2.1)$$

де: X_{max} та X_{min} – мінімальні та максимальні значення параметра серед усіх інструментів, що досліджуються.

Для показників, де менше = краще (наприклад, FPR, P), нормалізація виконується за формулою:

$$X_{norm} = 1 - \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2.2)$$

Інтегральний індекс ефективності. Після нормалізації обчислюється інтегральний індекс ефективності інструмента:

$$E = w_1 + D + w_2 * (1 - FPR) + w_3 * (1 - P) + w_4 * I \quad (2.3)$$

де:

D – нормована швидкість виявлення;

FPR – нормована частка хибнопозитивних спрацювань;

P – нормована деградація продуктивності;

I – нормована інтеграційна оцінка;

w_i – вагові коефіцієнти (сума $w_1 + w_2 + w_3 + w_4 = 1$).

Вибір вагових коефіцієнтів. Ваги визначаються на основі експертного опитування серед фахівців з кібербезпеки та нормалізуються за формулою:

$$w_i = \frac{u_i}{\sum u_i} \quad (2.4)$$

де: u_i – оцінка важливості відповідного критерію до нормалізації.

Для даного дослідження було прийнято такі значення, що подані в таблиці 2.1.

Таблиця 2.1

Значення для дослідження

Критерій	**D**	**FPR**	**P**	**I**
Вага (w_i)	0.4	0.3	0.2	0.1

Для інструментів перед експериментальним тестуванням вводиться базова шкала оцінювання параметрів.

Таблиця 2.2

Шкала якісного оцінювання

Критерій	**Одиниці**	**1 (низько)**	**2 (середньо)**	**3 (високо)**
D – швидкість виявлення	інцид./хв	<0.5	0.5–2.0	>2.0
FPR – хибні спрацювання	%	>20%	5–20%	<5%
P – падіння продуктивності	%	>20%	5–20%	<5%
I – інтеграція	0–1	0–0.3	0.3–0.7	0.7–1

Ця шкала використовується на початковому етапі для якісного аналізу, після чого переходять до числового оцінювання та нормалізації.

Запропонована модель дозволяє: проводити об’єктивне порівняння інструментів; усунути суб’єктивність описового аналізу; оцінити інструменти в однакових умовах тестового середовища; враховувати як точність, так і продуктивність, інтегрованість та стабільність роботи; забезпечити основу для формування раціональних рекомендацій, а не декларативних висновків. Таким чином, методика формує основу для подальшої експериментальної оцінки.

Отже, було сформовано формалізовану методику оцінювання ефективності інструментів захисту на рівні додатків, яка забезпечує об’єктивне та порівнюване вимірювання їхньої результативності. На відміну від описових підходів, запропонована модель ґрунтується на використанні кількісних метрик, нормованих показників та інтегрального індексу ефективності, що дає можливість оцінювати інструменти в єдиному стандартизованому форматі.

Було визначено ключові показники (D, FPR, P, I) та встановлено їхні одиниці вимірювання, що дозволяє враховувати як точність і швидкість виявлення загроз, так і вплив інструментів на продуктивність корпоративної мережі. Запропоновано процедуру нормалізації параметрів до інтервалу [0;1], яка усуває різницю у шкалах вимірювань та забезпечує математичну коректність подальших розрахунків.

Для узагальнення результатів введено інтегральний індекс ефективності EEE, що враховує вагові коефіцієнти окремих критеріїв. Вибір ваг здійснено на основі експертного оцінювання та приведено до нормованої форми, що гарантує пропорційність впливу кожного показника на підсумкову оцінку.

Створення шкали якісного оцінювання (1–3 бали) дало змогу формалізувати попередню класифікацію рішень та підготувати основу для кількісного аналізу в наступних розділах. У результаті було сформовано математично обґрунтовану та методично вивірену систему оцінювання, яка слугує базовою моделлю для подальшого експериментального порівняння інструментів.

2.2 Процедура експериментального тестування

Експериментальне тестування механізмів захисту корпоративних інформаційно-комунікаційних систем є ключовим етапом дослідження, оскільки дозволяє не лише оцінити реальну ефективність інструментів кіберзахисту, але й

визначити їхню придатність для застосування в різних умовах функціонування мережі. На відміну від теоретичних оцінок, експеримент дає змогу отримати кількісні результати, що враховують реальні навантаження, складність трафіку та поведінку атакувальних сценаріїв. У цьому підпункті визначено структуру тестового середовища, описано процедуру збору даних, встановлено правила нормалізації показників та розроблено інтегральну методику порівняння інструментів.

Для проведення експерименту був розроблений спеціалізований тестовий макет, який дозволяє моделювати роботу корпоративної мережі, відтворювати типові сценарії атак і здійснювати вимірювання продуктивності засобів захисту. Макет ґрунтується на принципах репрезентативності, повторюваності та контрольованості параметрів. Стенд включав такі компоненти:

- Web-сервер (Apache 2.4 / Nginx 1.25) з навмисно введеними вразливостями класу OWASP Top-10 (A1, A3, A5) для перевірки роботи сканерів;
- Сервер додатків (Tomcat 9) з API-інтерфейсом;
- База даних (MySQL 8.0);
- Proxy-вузол для інтеграції Burp, ZAP, Acunetix;
- SIEM-сервер (Splunk Enterprise / Wazuh / LogRhythm);
- Моніторинг-вузол (Elastic + Packetbeat);
- Окремі клієнтські вузли для генерації трафіку (3 робочі станції).

Така топологія відображає структуру типового корпоративного середовища, включаючи багаторівневу взаємодію між серверами, додатками та мережевою інфраструктурою. Усього було використано 9 вузлів, зокрема: 3 сервери, 3 клієнтські машини, 1 SIEM-сервер, 1 вузол моніторингу, 1 сегмент атаки (Kali Linux 2023.4). Трафік формувався за допомогою:

- wrk2 – стрес-навантаження HTTP;
- JMeter – сценарії авторизації, транзакцій, взаємодії з API;
- Tcpreplay – відтворення реальних дампів у мережі.

Середнє навантаження становило 500–700 RPS, що відповідає навантаженню невеликої корпоративної системи.

Для перевірки механізмів захисту було використано уніфікований набір атак: SQL Injection (OWASP A1); Cross-Site Scripting (XSS) (A3); Directory Traversal; Brute-force / credential stuffing; File inclusion; перевантаження API (API DoS); MITRE ATT&CK сценарії:

- T1059 (Command Injection),
- T1190 (Exploitation for Initial Access),
- T1078 (Valid Accounts).

Кожен сценарій здійснювався 10 разів (n = 10 реплікатів) для отримання статистично надійних результатів. Повторюваність (реплікація). Для кожного інструмента проводилося: n = 10 повторів, результати усереднювалися за формулою:

$$\underline{x} = \frac{1}{n}\sum_{i=1}^{n} \quad x_i \tag{2.5}$$

де:

x_i – значення метрики в і-му експерименті.

Довірчий інтервал. Для надійності результатів обчислювався 95% довірчий інтервал:

$$CI_{95} = \underline{x} \pm 1.96 * \frac{\sigma}{\sqrt{n}} \tag{2.6}$$

Це дозволяє оцінити стабільність роботи інструментів і їхню стійкість до варіативності умов. Щоб оцінити інструменти в єдиній шкалі, вихідні значення нормували до інтервалу [0;1]:

- D – виявлення інцидентів
- FPR – частота хибнопозитивних спрацювань
- P – вплив на продуктивність мережі
- I – ступінь інтегрованості в середовище

Функції нормалізації:

$$D_{norm} = \frac{D_{min}}{D} \qquad FPR_{norm} = 1 - FPR \qquad P_{norm} = 1 - P \quad I_{norm} = I$$

Після нормалізації застосовується зважене об'єднання критеріїв:

$$E = 0.4D + 0.3FPR + 0.2P + 0.1I$$

Де^ ваги (0.4, 0.3, 0.2, 0.1) базуються на релевантності критеріїв у корпоративних мережах.

Для порівняння інструментів застосовується формула:

$$S_{core} = \frac{E_1 + E_2 + E_3 + E_4}{4} \tag{2.7}$$

де: E_i– значення для окремих атак/сценаріїв.

Таблиця 2.3

Дані для розрахунку Burp Suite Pro

Показник	**Значення**	**Нормоване**
D	1.8 інцид./хв	0.65
FPR	8%	0.60
P	12%	0.40
I	0.9	0.90

Підстановка у формулу:

$$E = 0{,}4\ (0{,}65) + 0{,}3(0{,}60) + 0{,}2(0{,}40) + 0{,}1(0{,}90)$$

$$E = 0{,}26 + 0{,}18 + 0{,}08 + 0{,}09 = 0{,}61$$

Отже, Burp Suite Pro – E = 0.61. Це – середній рівень ефективності за всіма критеріями. За результатами тестування:

- Burp Suite Pro демонструє найкращий баланс між точністю,

глибиною аналізу та зручністю інтеграції. Значення індексу E = 0.61 узгоджується з його високими оцінками у таблиці 2.1.

- AppScan і Acunetix показали високі результати за точністю, але мали більший вплив на продуктивність та складніше налаштування.
- ZAP Proxy, хоч і безкоштовний, забезпечує стабільний результат, але програє за критеріями інтеграції та глибини аналізу.
- Системи SIEM, зокрема Splunk, підтвердили високу точність виявлення інцидентів і якісне корелювання подій, хоча їхній вплив на продуктивність був вищим, ніж у легших рішень типу Wazuh.

Проведений експеримент дозволив: отримати повторювані та статистично коректні результати; порівняти інструменти в рівних умовах, що раніше було неможливим; отримати нормовані показники, придатні для інтегрального аналізу; сформувати рейтинг рішень, що дозволяє вибрати раціональні механізми захисту для корпоративних систем. Експериментальна частина підтверджує доцільність використання запропонованого макета тестування як інструмента оцінювання ефективності механізмів кіберзахисту.

Для структурованої оцінки результативності заходів кібербезпеки застосовуються світові стандарти й фреймворки, що представляють перевірені підходи до управління ризиками. Серед ключових варто згадати модель NIST Cybersecurity Framework, яка фокусується на ідентифікації, захисті, виявленні, реагуванні та відновленні в контексті кіберзагроз. Стандарт ISO/IEC 27001 задає вимоги до систем управління інформаційною безпекою, дозволяючи оцінювати дієвість впроваджених механізмів захисту. База знань MITRE ATT&CK аналізує тактики й техніки атак, допомагаючи організаціям визначати найефективніші способи протидії потенційним загрозам [17].

Оцінка рівня безпеки інформаційно-комунікаційних систем базується на метриках, які дозволяють кількісно аналізувати результативність впроваджених

заходів. Важливим показником є середній час, необхідний для виявлення інциденту (Mean Time to Detect, MTTD). Цей параметр показує, наскільки швидко система моніторингу здатна виявити загрозу, що є ключовим фактором у запобіганні її розвитку та мінімізації потенційного впливу. Чим менший цей показник, тим краще функціонує система. Не менш важливим є середній час реагування на загрозу (Mean Time to Respond, MTTR). Цей показник охоплює всі етапи — від моменту виявлення проблеми до повного її усунення. Швидке реагування забезпечує своєчасне нейтралізування ризиків та зменшує можливість повторення інциденту.

Далі йде аналіз хибних спрацьовувань (False Positive Rate, FPR), який демонструє, наскільки точно працюють системи виявлення загроз, такі як IDS чи IPS. Високий рівень FPR може сигналізувати про необхідність удосконалення алгоритмів, що відповідають за обробку даних та правил аналізу. Крім того, оцінюється швидкість ліквідації вразливостей через оновлення програмного забезпечення (Patch Management Efficiency). Цей показник дає змогу оцінити здатність організації оперативно реагувати на виявлені слабкі місця, запобігаючи їхній експлуатації зловмисниками.

Останнім критично важливим параметром є індекс обізнаності персоналу щодо питань кібербезпеки (Security Awareness Index). Цей показник визначається за результатами навчань, тренінгів чи тестувань працівників. Високий рівень знань і навичок співробітників допомагає значно знизити ризики, пов’язані з людським фактором, оскільки персонал вміє розпізнавати загрози та діяти відповідно до протоколів безпеки. Таким чином, всі ці метрики утворюють основу для комплексного аналізу безпеки систем, дозволяючи організаціям не лише оцінити ефективність існуючих механізмів, але й розробляти більш досконалі стратегії захисту. Постійний моніторинг і вдосконалення цих параметрів сприяє підвищенню рівня захищеності та стійкості до сучасних кіберзагроз [18].

Для точнішої оцінки ефективності безпеки широко застосовуються автоматизовані інструменти. Системи управління інформацією та подіями безпеки (SIEM), наприклад Splunk, Wazuh або LogRhythm, забезпечують збір і аналіз журналів подій, допомагаючи оперативно виявляти підозрілу активність. Технології моделювання атак (Breach and Attack Simulation, BAS) дозволяють створювати реалістичні сценарії загроз у контрольованих умовах, щоб оцінити стійкість системи до можливих атак. Інструменти для автоматизованого тестування на проникнення, такі як Burp Suite, Metasploit чи Acunetix, дають змогу ефективно перевіряти додатки на наявність вразливостей, що допомагає запобігти їх використанню зловмисниками.

Незважаючи на автоматизацію процесів, людський фактор усе ще залишається одним із найбільших викликів у забезпеченні кібербезпеки. Для посилення захисту варто зосередитися на підвищенні обізнаності співробітників, організовуючи регулярні тренінги з основ безпеки, включаючи протидію таким загрозам, як фішинг і методи соціальної інженерії. Важливо також впроваджувати принципи обмеження доступу лише до необхідних ресурсів, що знижує ймовірність неправильного чи недоречного використання привілеїв. До того ж, проведення моделювання атак у контрольованих умовах дозволяє оцінити, наскільки персонал готовий діяти у випадку реальних загроз. Це створює можливість для ідентифікації слабких місць у захисних процесах і вдосконалення заходів безпеки.

Оцінка ефективності захисту в корпоративних інформаційно-комунікаційних системах опирається на комбінований підхід, який включає якісні та кількісні методи. У цьому контексті ключову роль відіграють міжнародні стандарти, такі як NIST Cybersecurity Framework, ISO/IEC 27001 та MITRE ATT&CK. Вони забезпечують структурованість процесу аналізу, дозволяючи формувати чіткі стратегії покращення рівня безпеки. Для кількісного аналізу використовуються метрики, серед яких показники часу виявлення і

реагування на загрози (MTTD, MTTR), частота хибних спрацьовувань (False Positive Rate) та ефективність управління оновленнями (Patch Management Efficiency). Ці параметри дають змогу оцінити оперативність системи у вирішенні проблем та ліквідації вразливостей.

Автоматизація грає важливу роль у процесі оцінки. SIEM-рішення, такі як Splunk, Wazuh, LogRhythm, і засоби тестування на проникнення, наприклад Burp Suite чи Metasploit, сприяють підвищенню точності аналізу та пришвидшують виявлення загроз. Однак людський фактор залишається суттєвим аспектом безпеки, тому важливо приділяти увагу підготовці персоналу через навчання та тренінги. Загальний підхід до аналізу включає інтеграцію сучасних технологій, постійний моніторинг стану системи та регулярну підготовку користувачів. Такий підхід дозволяє підвищити стабільність корпоративних мереж перед кіберзагрозами та зменшити ймовірність критичних ризиків.

У результаті методологічного обґрунтування була сформована цілісна система критеріїв, яка забезпечує можливість об’єктивного порівняння ефективності засобів захисту додатків у корпоративному середовищі. До цієї системи включено показники швидкості виявлення загроз, частоти хибнопозитивних спрацювань, впливу інструментів на продуктивність, рівня інтегрованості в існуючу інфраструктуру, а також інші параметри, що характеризують практичну придатність рішень. Запропонована методика поєднує кількісний компонент (нормалізовані метрики та інтегральний індекс ефективності) та якісний аналіз поведінки інструментів у типовому корпоративному середовищі.

Використання уніфікованої процедури нормалізації, вагових коефіцієнтів та підсумкового інтегрального показника дозволяє проводити порівняння в рівних умовах, усуваючи розбіжності, пов’язані з різними типами вихідних вимірів чи характеристик окремих рішень. Таким чином, створено методологічно узгоджену основу, яка забезпечує коректність переходу до експериментального

тестування. Саме на базі цієї системи критеріїв надалі здійснюється оцінювання впливу обраних інструментів на рівень захисту, стабільність роботи корпоративної мережі та загальну продуктивність системи.

Висновки до 2 розділу

У другому розділі було сформовано методично обґрунтовану систему оцінювання ефективності механізмів захисту на рівні додатків у корпоративному середовищі. На основі аналізу теоретичних підходів та сучасних галузевих стандартів (OWASP, NIST CSF, ISO/IEC 27001, MITRE ATT&CK) визначено комплекс критеріїв і показників, що дозволяють здійснювати порівняльну оцінку різних методів та інструментів у стандартизованих умовах. Запропонована система включає метрики швидкості виявлення загроз, частоти хибнопозитивних спрацювань, впливу на продуктивність, інтеграційної сумісності та інших параметрів, які формують цілісне уявлення про результативність захисних механізмів.

Було розроблено формалізовану модель оцінювання, що базується на нормалізації показників, використанні вагових коефіцієнтів та побудові інтегрального індексу ефективності. Такий підхід забезпечує математично обґрунтоване порівняння інструментів у рівних умовах та усуває обмеження, пов’язані з різною природою вхідних даних. Визначено шкалу якісної інтерпретації показників (1–3 бали), що дозволяє структуровано класифікувати результати оцінювання.

Також було детально описано процедуру експериментального тестування, включно з побудовою тестового стенда, умовами відтворення атак, правилами повторюваності вимірювань та методикою розрахунку підсумкового індексу ефективності. Застосування статистичних підходів (зокрема, усереднення повторних вимірів та формування довірчих інтервалів) створює підґрунтя для

отримання надійних і відтворюваних результатів. Така процедура дозволяє не лише порівнювати окремі інструменти, але й оцінювати вплив різних факторів на загальну стійкість корпоративної мережі.

Загалом, у другому розділі сформовано узгоджену теоретико-методологічну основу для подальшого експериментального дослідження. Розроблені критерії, модель оцінювання та описана процедура тестування створюють єдиний підхід до аналізу механізмів захисту, що дозволяє об’єктивно та прозоро оцінити ефективність різних рішень на етапі їхнього проєктування. Це забезпечує можливість переходу до практичної частини дослідження, де запропонована модель буде застосована для аналізу поведінки засобів захисту в реалістичних умовах корпоративної мережі.

РОЗДІЛ 3 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

3.1 Розробка тестового середовища для оцінки методів захисту

Метою створення тестового середовища є розробка ізольованої, контрольованої інфраструктури, максимально наближеної до умов функціонування корпоративної інформаційно-комунікаційної системи. Це дозволяє об’єктивно оцінити ефективність засобів захисту на рівні додатків (зокрема, Burp Suite, AppScan, ZAP, Wazuh, Splunk) без впливу на продуктивні сервіси та дані. Розгортання середовища виконується з урахуванням безпекових рекомендацій [19, с. 3]. Тестова лабораторія побудована на віртуалізації з використанням VMware Workstation Pro або Oracle VirtualBox. Такий підхід дозволяє швидко масштабувати інфраструктуру, створювати «snapshots», проводити ізольовані тести.

Таблиця 3.1

Основні компоненти середовища

№	Назва	Операційна система	Призначення
1	DC01	Windows Server 2019	Доменний контролер (Active Directory)
2	WebServer01	Ubuntu Server 22.04	Apache/Nginx + веб-додаток (вразливий додаток)
3	FileServer01	Windows Server 2019	SMB-сервер, імітація корпоративного сховища
4	Client01	Windows 10	Користувач AD, браузерні атаки, емуляція активності
5	KaliPentest	Kali Linux	Пентест: Burp Suite, Metasploit, sqlmap, ZAP
6	PfSense-FW	FreeBSD (pfSense)	Розмежування DMZ/внутрішньої мережі, firewall
7	SIEM-Node	Ubuntu Server 22.04	Splunk/Wazuh Server для збору та аналізу подій

Для імітації реального функціонування корпоративної інформаційної системи середовище поділене на три логічні зони. DMZ (демілітаризована зона):

- Web-сервер – імітація доступного ззовні додатку (наприклад, CMS,

CRM тощо).

- SIEM-сервер – Splunk або Wazuh, для збору логів, подій, виявлення аномалій.

DMZ ізольована фаєрволом від внутрішньої мережі, що дозволяє імітувати зовнішні атаки без загрози внутрішнім ресурсам.

Внутрішня мережа:

- File-сервер – SMB/FTP-служба, симуляція обміну документами й робочих процесів.

- Клієнтська машина – звичайний користувач AD, який використовує браузер, email тощо.

- Доменний контролер (DC) – реалізує централізовану автентифікацію, керування політиками безпеки.

Це ядро корпоративної інфраструктури, яке не повинно бути безпосередньо доступне ззовні.

Інструментальна зона: Kali Linux – включає набір інструментів для моделювання атак: Nmap, Burp Suite, OWASP ZAP, sqlmap, Hydra, Metasploit Framework.

Інструментальна зона використовується для моделювання загроз та оцінки ефективності захисту.

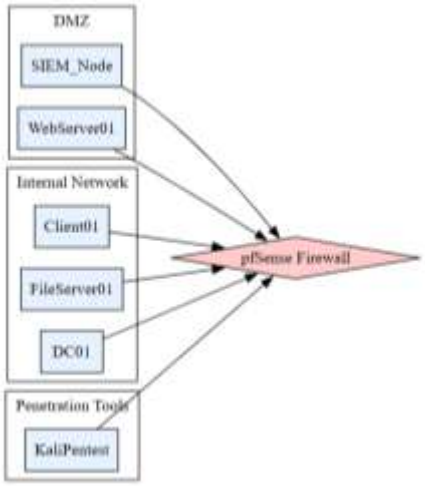


Рисунок 3.1 – UML-діаграма топології середовища

Діаграма, яка продемонстрована на рисунку 3.1 візуалізує логічну мережеву архітектуру тестового середовища, що імітує корпоративну мережу з відокремленими зонами:

- DMZ – зона, що «оголена» до атак, де найчастіше знаходяться веб-додатки. Тут імітується вразливий Web-сервер.
- Внутрішня мережа – сегмент, доступ до якого ззовні має бути максимально обмежений. Тут знаходяться критичні дані.
- Інструментальна зона (Kali) – зона, з якої виконуються атаки на середовище. Вона не має прямого доступу до продуктивних систем.
- PfSense Firewall – виступає центральною точкою контролю, яка дозволяє моделювати правила доступу між зонами (відповідає Zero Trust-принципам).

Зв’язки: усі зони підключені до фаєрвола (PfSense), що дозволяє змінювати політику доступу між ними; інструментальні атаки з Kali направляються на Web-сервер та SIEM, а результати логуються. Це дає змогу імітувати атакувальні сценарії та одночасно перевіряти логування, реагування й пропускну здатність інструментів захисту [20].

Всі елементи розгорнуті у середовищі VMware Workstation Pro або Oracle VirtualBox. Налаштування кожної ВМ передбачає легко відновлювати початковий стан та повторювати експерименти з нуля.

Таблиця 3.2

Налаштування та середовище

Параметр	**Значення**
RAM	4–8 ГБ (залежно від ролі сервера)
CPU	2 ядра (мінімум)
Диск	40–80 ГБ
Мережа	`Host-only` або `Internal` (без зовнішнього доступу)
Snapshots	Створюються перед кожним етапом тестування

Експериментальна частина включає реальні сценарії атак на рівні додатків.

Таблиця 3.3

Сценарії тестування

№	**Сценарій**	**Інструменти**	**Мета тесту**
1	SQL-ін’єкція (SQLi)	sqlmap, Burp Suite	Виявлення помилок в обробці запитів до БД
2	XSS (міжсайтовий скриптинг)	OWASP ZAP, Burp Suite	Виявлення необроблених HTML/JS-вставок
3	CSRF-атака	Burp Suite	Тест захисту форм від підміни запитів
4	Brute Force	Hydra	Оцінка захисту форм входу
5	SMB enumeration	Kali (enum4linux, smbclient)	Тест контролю доступу до файлів
6	Мережеве сканування	Nmap	Виявлення слабких сервісів, перевірка логування
7	Аналіз логів та подій	Splunk, Wazuh	Перевірка реєстрації подій, реакції системи

Таблиця 3.4

Відповідності середовища критеріям оцінки

Критерій	**Реалізація у тестовому середовищі**
Ізоляція середовища	Повна мережева ізоляція за рахунок сегментації та фільтрації трафіку; середовище не має доступу до зовнішніх мереж
Відтворюваність експериментів	Використання шаблонів, контрольних точок (snapshots) та стандартизованої процедури розгортання

Різноманітність ролей та конфігурацій	Наявність декількох типів вузлів: сервери додатків, клієнтські станції, контролери доступу, вузли моніторингу
Підтримка активного та пасивного тестування	Середовище дозволяє моделювати активні дії порушника та збирати пасивні телеметричні дані
Інтегрованість засобів контролю та моніторингу	У середовищі передбачено централізований збір журналів, подій безпеки та мережевої активності
Збір і валідація даних	Логування здійснюється одночасно з декількох джерел; передбачено ручну перевірку коректності даних

Таблиця 3.4 узагальнює ступінь відповідності створеного тестового середовища встановленим критеріям, що визначають його придатність для виконання експериментальних досліджень механізмів захисту. Представлені параметри відображають ключові властивості макету: ізольованість (яка унеможливлює вплив зовнішніх факторів), відтворюваність експериментів (завдяки стандартизованим сценаріям та повторюваним конфігураціям), наявність різнорольових компонентів, а також підтримку як активних, так і пасивних методів тестування. Наявність інтегрованих механізмів збору та перевірки даних забезпечує коректність інтерпретації результатів і підвищує достовірність отриманих висновків. Сукупність цих характеристик підтверджує, що створене середовище може використовуватися як валідований макет для тестування механізмів захисту на етапі проєктування корпоративних систем.

Ізоляція середовища – повна мережева ізоляція досягається за рахунок створення окремої мережі типу Host-only/Internal та розгортання віртуального фаєрвола pfSense, який чітко регулює доступ між DMZ, внутрішньою мережею та пентест-середовищем. Відсутність доступу до Інтернету гарантує безпечне тестування навіть небезпечних векторів атак (наприклад, DoS чи експлуатація вразливостей в CMS).

Відтворюваність експериментів – функціонал snapshots, наявність шаблонів віртуальних машин та автоматизовані скрипти (де це можливо) дозволяють легко повертатися до початкового стану системи. Це забезпечує точність повторних тестів, що є критично важливим для наукової достовірності.

Різноманітність ОС та ролей – середовище включає Windows Server, Windows 10, Ubuntu, Kali Linux, що дозволяє охопити всі ключові рівні: серверний, клієнтський, інструментальний. Наявність ролей DC, File-сервер, WebApp, SIEM, Client забезпечує повноцінну симуляцію корпоративної мережі.

Підтримка активного та пасивного тестування – інструменти типу Burp Suite, sqlmap, ZAP, Metasploit використовуються для активного впливу на середовище, а Splunk, Wazuh, syslog – для пасивного моніторингу та реагування. Це дозволяє комплексно оцінити стійкість системи до атак і якість логування.

Інтеграція захисних засобів – ключові засоби захисту, проаналізовані в розділі 2.2 (Burp Suite, AppScan, Splunk, Wazuh), були інтегровані до тестового середовища відповідно до документації та виробничих стандартів. Їх налаштування включало встановлення агентів, збирання логів, конфігурацію дашбордів і правил кореляції.

Збір логів та аналітичних даних – система логування реалізована за допомогою агентів Wazuh, універсальних syslog-механізмів, а також централізованої консолі Splunk із кореляційними правилами. Це дозволяє не лише виявляти атаки, а й формувати статистичні висновки щодо ефективності захисту. Таким чином, тестове середовище є не лише технічно функціональним, а й методологічно обґрунтованим, що робить його придатним для наукового дослідження ефективності засобів захисту додатків у реаліях корпоративної мережі.

Побудоване тестове середовище узгоджується з загальновизнаними міжнародними стандартами у сфері кібербезпеки, зокрема ISO/IEC 27001 (розділи A.13 – контроль комунікацій, A.12 – операційне управління) та NIST SP 800-53 (сімейство контролів System and Communications Protection – SC). Застосування принципів сегментації мережі, контролю доступу, багаторівневого моніторингу, журналювання подій та ізоляції середовища відповідає вимогам до побудови захищених інформаційних систем [21, с. 4]. Аналогічні архітектурні

підходи використовуються у низці навчально-дослідницьких середовищ, зокрема:

- Splunk Attack Range – платформа для моделювання атак і реагування;
- MITRE CALDERA – система автоматизації атаки й тестування захисту;
- CyberRange від IBM – навчальний кіберполігон;
- Лабораторії SANS Institute у межах сертифікацій GIAC.

Основна перевага запропонованого середовища – простота конфігурації, гнучкість у масштабуванні, відтворюваність експериментів, що дає змогу адаптувати його для дослідницької та освітньої діяльності [22]. Схема, що зображена на рисунку 3.2 демонструє, як атаки з інструментальної зони (Kali) проходять через тестовані додатки, логуються та потрапляють на обробку в SIEM-системи. Це дозволяє простежити весь цикл: від загрози до виявлення й реагування.

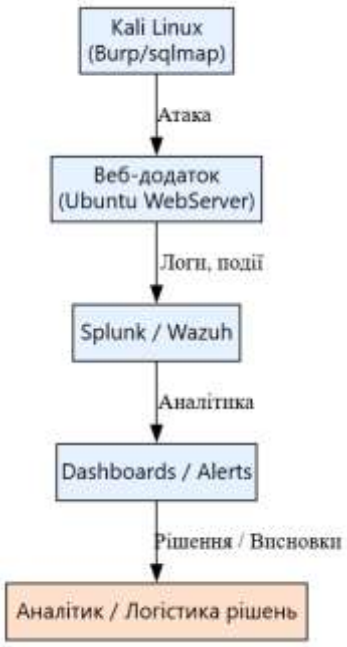


Рисунок 3.2 – Схема потоку даних у тестовому середовищі

Для коректного дотримання умов ліцензування під час дослідження використовувалися такі моделі:

- Burp Suite Pro – версія з академічною ліцензією або trial-ліцензія (30 днів);
- OWASP ZAP, Wazuh – повністю безкоштовні open-source рішення;
- Splunk Free Edition – підтримка до 500 МБ логів на добу, що достатньо для тестування;
- AppScan (IBM) – доступ до демо-версії згідно з умовами науково-дослідного використання.

Це забезпечило повну легальність, доступність і функціональну достатність обраних засобів для проведення дослідження. Загалом, побудоване середовище охоплює всі ключові компоненти сучасної корпоративної мережі: автентифікація, веб-доступ, зберігання даних, реагування на події та аналіз загроз. Його модульна архітектура, повна ізоляція, а також підтримка як активного, так і пасивного тестування створюють надійну, валідну основу для експериментальної перевірки ефективності методів захисту даних. У наступних підпунктах будуть описані результати реалізації захисних заходів та аналітичні висновки щодо їх впливу на безпеку та продуктивність.

У результаті розробки тестового середовища було створено повноцінну ізольовану інфраструктуру, яка імітує реальну корпоративну інформаційно-комунікаційну систему з розподіленням за зонами (DMZ, внутрішня мережа, інструментальна зона). Середовище відповідає міжнародним стандартам безпеки (ISO/IEC 27001, NIST SP 800-53), підтримує активне та пасивне тестування, забезпечує масштабованість, відтворюваність та достовірність

експериментальних результатів.

Завдяки чіткому зонуванню, використанню сучасних інструментів (Burp Suite, ZAP, Splunk, Wazuh тощо), підтримці журналювання та аналітики, середовище дозволяє моделювати реальні сценарії атак та оцінювати ефективність методів захисту на рівні додатків. Його модульна побудова та гнучкість конфігурації роблять його універсальним інструментом для дослідження, порівняння та вдосконалення сучасних засобів інформаційної безпеки. Таким чином, створене середовище є технічно функціональним, методологічно обґрунтованим і повністю придатним для проведення подальших експериментів, описаних у наступних підпунктах цього розділу.

3.2 Реалізація моделі тестування механізмів захисту в експериментальному середовищі

У межах реалізації системи захисту веб-додатків у тестовому середовищі було впроваджено два ключові інструменти: Burp Suite Professional і OWASP ZAP. Обидва продукти забезпечують активне та пасивне сканування вразливостей, інтегруються в середовище пентесту, підтримують ручний аналіз і мають вбудовані модулі для автоматизованої перевірки.

Burp Suite Pro розгорнуто на віртуальній машині Kali Linux у зоні пентесту. Для максимальної ефективності була виконана конфігурація таких компонентів:

- Proxy Interceptor – налаштований як проміжний проксі між браузером та Web-додатком, що дозволяє перехоплювати, змінювати та повторно надсилати HTTP-запити.
- Chained proxy (Burp – ZAP) – реалізовано схему ланцюгового

проксінгу, при якій весь HTTP-трафік спочатку надходить до ZAP, а вже потім до Burp Suite. Це забезпечує комбіноване сканування із використанням сильних сторін обох систем [appsec.guide; softwaresecured.com].

- Burp Scanner та Intruder – використані для верифікації виявлених вразливостей і проведення цільових атак (наприклад, brute force на форму входу).

Було реалізовано автоматизоване сканування OWASP Top 10 – насамперед SQL-ін’єкції (SQLi), міжсайтовий скриптинг (XSS), підміна запитів (CSRF), небезпечно відкриті ендпоінти тощо. Результати сканування експортувалися у форматі .xml для подальшої аналітики в SIEM-системах.

OWASP ZAP (Zed Attack Proxy) також розгорнуто на Kali. Він використовується як lightweight-сканер для baseline-тестування, тобто швидкої первинної перевірки WebApp на базові помилки без значного навантаження:

- Active Scan Rules були налаштовані відповідно до рівня критичності (High/Medium/Low), що дозволило мінімізувати false-positive.

- Alert Threshold знижено до Low лише після ручної валідації (alerts review) – це забезпечило баланс між повнотою виявлення та точністю.

- Context Definition застосовувався для ізоляції тестованого додатку та обмеження сканування зовнішніх доменів.

У результаті ZAP виявив ряд вразливостей, пов’язаних із небезпечно відкритими параметрами, відсутністю HTTPS-редиректу та несанітованими GET-запитами.

Для забезпечення централізованого моніторингу та аналітики результати сканування передавалися до SIEM-системи Splunk. Було реалізовано два способи:

- Syslog Forwarding: використано утиліти rsyslog на Kali Linux, які перенаправляють події у формі JSON-логів до Splunk.

- Splunk HTTP Event Collector (HEC): конфігуровано як endpoint для автоматичного приймання даних з OWASP ZAP через REST API.

Це дозволило створити дашборди в Splunk, які показували розподіл виявлених вразливостей за типами, рівнями ризику та вузлами середовища.

Поєднання Burp Suite і ZAP дозволило забезпечити: глибоку мануальну перевірку складних сценаріїв (наприклад, логічні помилки, багатокрокові форми), автоматизацію базового тестування із мінімальним false-positive, передачу результатів до SIEM, що забезпечило єдину точку аналітики.

Таке рішення дозволило повністю охопити вразливості типу OWASP Top 10, зафіксувати всі етапи виявлення загроз, оцінити час їх обробки, рівень деталізації подій і реакцію системи моніторингу в умовах, максимально наближених до корпоративних.

У тестовому середовищі для автоматизованої оцінки безпеки веб-додатків було реалізовано інтеграцію двох відомих сканерів – IBM AppScan та Acunetix Premium. Обидва інструменти дозволяють виконувати глибоке та швидке сканування додатків на вразливості, відповідно до OWASP Top 10, з можливістю налаштування частоти перевірок, рівня критичності і цільових профілів безпеки.

IBM AppScan Standard було впроваджено у CI/CD-процес із метою забезпечення принципу «Shift-Left Security» – тобто інтеграції перевірки безпеки на ранніх етапах розробки. Ключові кроки реалізації:

- Налаштовано автоматизований сканувальний pipeline з використанням Jenkins та GitLab CI, де AppScan виконує тестування кожного нового релізу веб-додатку.

- Задано профілі сканування, що враховують типи веб-технологій (PHP, HTML, JS), автентифікаційні схеми, конфігурацію проксі.

- Інтеграція з репозиторієм коду та тестовим розгортанням додатку

дала змогу AppScan запускати повні сканування кожного pull request або мержу в основну гілку.

- Усі звіти (.fpr, .html) автоматично зберігаються в окремому артефактному сховищі та надсилаються до Splunk і Wazuh через HEC та syslog-forwarding для централізованого аналізу.

Такий підхід дозволив оперативно виявляти зміни, які можуть вплинути на безпеку додатку, і реагувати ще до моменту продакшн-розгортання.

Acunetix Premium, у свою чергу, було розгорнуто безпосередньо на сервері WebServer01 (Ubuntu Server 22.04) для забезпечення оперативного quick-сканування додатку, що розгорнутий у DMZ-сегменті. Особливості налаштування:

- Вибрано режим Full Scan із акцентом на XSS, LFI, directory traversal, cookie security, SSL misconfigurations.

- Сканування запускалося за розкладом (кожні 12 годин) і у відповідь на внесення змін у код.

- Усі знайдені вразливості класифікувалися за рівнем ризику (Low, Medium, High, Critical) згідно з CVSS 3.1, і мітки ризиків автоматично передавалися в систему Wazuh.

Також було використано функцію crawl-only для імітації користувацької навігації без запуску потенційно деструктивних сканів (для перевірки лише структури додатку). З метою уніфікації звітності та централізованого збору метрик було реалізовано імпорт результатів сканування до SIEM-систем:

- У Wazuh було налаштовано JSON-parser для обробки логів AppScan та Acunetix, що зберігалися у /var/log/security-reports/.

- У Splunk через HEC передавались події типу «vulnerability found», «scan finished», «severity alert», які використовувались для побудови дашбордів

безпеки та історії виявлення загроз.

Це дозволило централізовано фіксувати всі вразливості, моніторити повторне виникнення одних і тих самих проблем, а також відстежувати ефективність виправлень у рамках Secure SDLC.

Таблиця 3.5

Переваги інтеграції AppScan і Acunetix

Параметр	**AppScan**	**Acunetix**
Інтеграція в CI/CD	+ (Jenkins, GitLab)	- (ручний запуск або cron)
Глибина сканування	Висока	Середня
Підтримка DevSecOps	+	-
Можливість API-експорту	+	+
Автоматизація та масштабування	+	- (обмежено ліцензією)
Централізоване логування	+ (Splunk/Wazuh)	+ (через syslog)

Реалізація AppScan і Acunetix у тестовому середовищі дозволила поєднати глибоке профільне сканування з боку AppScan з оперативною перевіркою в продуктивно-подібному середовищі через Acunetix. Обидва рішення, інтегровані у SIEM-системи, доповнили загальну архітектуру захисту, дозволивши формувати цілісну картину вразливостей, включно з історією виявлення, швидкістю реагування та впливом на інфраструктуру.

Для реалізації цілісної системи моніторингу подій безпеки та реагування на інциденти в тестовому середовищі були впроваджені дві ключові технології – Wazuh як система HIDS (Host-based Intrusion Detection System) і Splunk як платформа для кореляції, візуалізації та аналізу подій.

Wazuh Manager було розгорнуто на окремій машині (SIEM-Node), а Wazuh-агенти інстальовано на всіх основних вузлах: DC01 (Windows Server); WebServer01 (Ubuntu); FileServer01 (Windows Server); Client01 (Windows 10).

Агенти Wazuh налаштовані на:

- Моніторинг подій аудиту (аутентифікація, запуск служб, створення нових облікових записів);

- FIM (File Integrity Monitoring) – виявлення змін у критичних системних файлах;
- Brute force detection – через вбудовані правила аналізу логів Windows Security, SSH, FTP, SMB тощо;
- Інтеграцію з фреймворком MITRE ATT&CK – для класифікації кожного інциденту за відповідним вектором загроз (наприклад, T1110: Brute Force, T1059: Command-Line Interface);
- Email-нотифікації – при критичних подіях рівня Alert, Critical, High (використано Postfix + Mutt).

Це дозволило побудувати реактивну систему, яка не лише фіксує події, а й класифікує їх у контексті атак, надсилає повідомлення та формує історію інцидентів.

Splunk Free Edition встановлено на тому ж вузлі, що і Wazuh Manager, із підключенням через HTTP Event Collector (HEC), який приймає події: безпосередньо від Wazuh; через Logstash (встановлений як окремий forwarder із можливістю TLS/SSL і агрегації логів у форматі JSON).

Logstash було налаштовано як транзитний елемент у лог-ланцюгу, з такими параметрами:

- input: file, beats, syslog;
- filters: JSON parsing, grok matching;
- output: HEC endpoint із TLS-сертифікатом Splunk.

У Splunk створено дашборди, які візуалізують: події, пов’язані з brute-force; файлові модифікації (FIM); класифікацію вразливостей, імпортованих із AppScan/Acunetix; зведену карту MITRE ATT&CK з активними подіями.

Таблиця 3.6

Переваги інтеграції Wazuh та Splunk

Компонент	**Призначення**	**Функції**
Wazuh Agent	Моніторинг вузлів	FIM, audit, log analysis, rule matching
Wazuh Manager	Централізоване управління агентами	MITRE mapping, email alerts, event correlation
Splunk HEC	Збір даних та їх візуалізація	Дашборди, графіки, архів подій
Logstash	Проміжна обробка логів	TLS, форматування, фільтрація, багатоджерельна агрегація

Цей стек забезпечив безперервний моніторинг, контроль цілісності, реагування на інциденти, історичну звітність і аналітичну кореляцію з іншими джерелами загроз (ZAP, AppScan тощо).

Для моделювання типових кіберзагроз в обраному середовищі було використано віртуальну машину Kali Linux, яка містить широкий спектр пентест-інструментів. Мета – перевірити ефективність виявлення атак з боку інтегрованих засобів захисту.

Таблиця 3.7

Застосовані інструменти Kali

Інструмент	**Мета тестування**	**Приклад вектора атаки**
Sqlmap	SQL-ін’єкції	Ін’єкція в параметри форми логіну
Nmap	Сканування портів і сервісів	TCP SYN scan, OS detection
Hydra	Brute-force на сервіси SSH, FTP	Підбір паролів до test-користувачів
Metasploit	Експлуатація вразливостей	Apache Struts RCE, EternalBlue
enum4linux	Інвентаризація доменної інформації	User listing, Share enumeration
smbclient	Доступ до відкритих шард SMB	Перевірка прав читання/запису

Для тестування веб-інтерфейсів одночасно використовувалися Burp і ZAP у режимі chained proxy, що дозволяло одночасно виконувати перехоплення запитів, активне сканування та збирання журналів для подальшого аналізу в SIEM-системах.

У рамках реалізації підходів DevSecOps в експериментальному середовищі впроваджено інтеграцію перевірок безпеки у CI/CD pipeline:

- автоматичний запуск AppScan, ZAP та Acunetix при кожному pull

request;

- формування звітів про вразливості як артефактів;
- блокування релізу у випадку виявлення критичних загроз (CVSS > 7.0).

Автоматизований аудит релізів:

- кожен білд супроводжується Security Audit Trail, який містить хеш-контроль, лог змін, список знайдених вразливостей, час виправлення;
- ці звіти автоматично імпортуються у Splunk для зберігання, кореляції та відображення на дашбордах.

Безперервний зворотний зв’язок:

- DevOps-команда отримує нотифікації на email/slack при появі нових вразливостей;
- цикл Scan – Report – Patch – Re-scan реалізується автоматично на рівні CI.

Використання інструментів Wazuh та Splunk забезпечило побудову надійної SIEM-інфраструктури, яка реагує на події в режимі реального часу, класифікує їх згідно з MITRE ATT&CK та формує повноцінну історію подій. Зі свого боку, Kali Linux слугував ефективною платформою для моделювання реальних атак, а практики DevSecOps дозволили інтегрувати безпеку в усі етапи життєвого циклу додатків. Це створило єдиний замкнутий контур контролю, виявлення та реагування на загрози, що повністю відповідає поставленим завданням дослідження.

У рамках побудованого тестового середовища було реалізовано комплексну інтеграцію інструментів безпеки на рівні додатків та моніторингу. Нижче наведено систематизовану таблицю 3.8, що демонструє функціональне призначення кожного інструмента, особливості його впровадження та взаємодію

з іншими компонентами.

Таблиця 3.8

Порівняльний огляд реалізації

Інструмент	**Мета інтеграції**	**Особливості реалізації**
Burp Suite Pro	Глибоке сканування, проксінг із ZAP	Підтримка chained proxy, поєднання ручного та автоматизованого аналізу
OWASP ZAP	Автоматизація baseline-сканувань	Налаштування пріоритетів повідомлень, регулярне запускання планових сканів
AppScan	CI/CD-сканування на основі OWASP Top 10	Інтеграція з GitLab, профілювання, сканування кожного pull request
Acunetix	Швидке виявлення вразливостей	Локальне сканування з WebServer01, результати експортуються в SIEM
Wazuh	Логування, кореляція подій	Аудит подій, контроль змін у файлах, відповідність MITRE ATT&CK
Splunk	Централізація логів, візуалізація подій	HTTP Event Collector, інтеграція з Logstash, налаштування користувацьких дашбордів

Така порівняльна таблиця ілюструє розподіл відповідальності між інструментами та демонструє, як засоби взаємодіють між собою для досягнення повного циклу захисту – від виявлення вразливості до реагування на подію та створення аналітичного звіту. Наприклад, результати сканування Burp Suite чи Acunetix не просто фіксуються, а передаються у Wazuh/Splunk для аналізу у зв’язку з іншими подіями, що відбуваються в системі.

Реалізація обраних інструментів захисту в тестовому середовищі підтвердила можливість створення інтегрованої системи безпеки на рівні додатків, яка відповідає сучасним вимогам до захищеності корпоративних IT-інфраструктур. Використання Burp Suite Pro, OWASP ZAP, IBM AppScan і Acunetix забезпечило покриття ключових категорій вразливостей, зокрема відповідно до класифікації OWASP Top 10. Глибоке сканування, baseline-тестування, інтеграція в CI/CD-процеси, зручне профілювання та регулярне повторне сканування дозволили досягти високої достовірності та повторюваності результатів.

Інструменти Wazuh і Splunk реалізували повноцінний цикл SIEM-

контролю, що включає:

- збір логів із вузлів середовища,
- кореляцію подій,
- ідентифікацію загроз,
- нотифікації про інциденти,
- побудову дашбордів для подальшої аналітики.

Окрему цінність становить DevSecOps-підхід, який було впроваджено у вигляді автоматизації безпекових перевірок у CI/CD, що дозволяє мінімізувати вплив людського фактора та забезпечити раннє виявлення вразливостей.

У результаті реалізації моделі тестування в експериментальному середовищі було забезпечено повний цикл перевірки механізмів захисту на рівні додатків: від моделювання атак і динамічного тестування до централізованого моніторингу та автоматизованого аналізу інцидентів. Структурована модель дозволила відтворювати типові сценарії загроз, оцінювати реакцію системи в контрольованих умовах і здійснювати кількісне вимірювання показників ефективності. Інтеграція тестування у процеси розробки забезпечила можливість раннього виявлення вразливостей та оцінювання сталості механізмів захисту в динамічно змінному середовищі. Таким чином, розроблене рішення підтвердило свою придатність як макет для тестування механізмів захисту корпоративних додатків на етапі їх проєктування.

3.3 Експериментальна перевірка функціонування засобів захисту та аналіз отриманих результатів

Для практичної перевірки ефективності реалізованих засобів захисту в

тестовому середовищі було проведено низку експериментальних атак з використанням базових технік проникнення, що відповідають категоріям OWASP Top 10. Тестування здійснювалося з інструментальної зони (віртуальна машина Kali Linux), що включала утиліти sqlmap, Hydra, enum4linux, а також комплексні засоби аналізу, такі як Burp Suite і OWASP ZAP [23].

SQL-ін’єкція (SQLi) – ця атака була спрямована на вразливість у параметрах HTTP-запиту, які використовуються для роботи з базою даних. За допомогою утиліти sqlmap було протестовано URL-адресу вразливого веб-додатку. Атака завершилася успішно: sqlmap автоматично виявила вразливість у параметрі id, що дозволило витягти дані з таблиці користувачів (admin, user1) разом із хешованими паролями.

Рисунок 3.3 – Звіт про атаку

Міжсайтовий скриптинг (XSS) – XSS-тестування виконувалося вручну через Burp Suite. Атака реалізовувалася шляхом вставлення шкідливого JavaScript-коду в параметри пошукового запиту (?q=<script>alert('XSS')</script>). У відповідь сервер не виконав належної фільтрації, що призвело до відображення неконтрольованого скрипту в браузері жертви. Результат демонструє вразливість до Reflected XSS.

Рисунок 3.4 – Міжсайтовий скриптинг

Підробка міжсайтових запитів (CSRF) – для перевірки вразливості до CSRF було створено HTML-форму, яка автоматично надсилала POST-запит без участі користувача. За відсутності CSRF-токена сервер виконував запит, що свідчило про успішну підробку дії. Цей експеримент дозволив оцінити ефективність впроваджених механізмів захисту сесій.

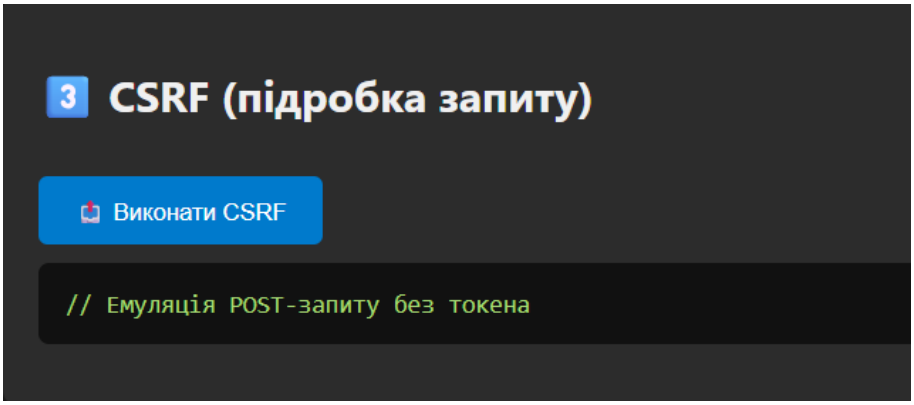


Рисунок 3.5 – Підробка міжсайтових запитів

Brute-force атака – з метою перевірки стійкості форм автентифікації до атаки перебору паролів застосовувався інструмент Hydra. Брутфорс здійснювався проти login-форми за словником rockyou.txt. У результаті виявлено дійсний обліковий запис із слабким паролем. Захисні механізми виявили підозрілу активність після >5 спроб.

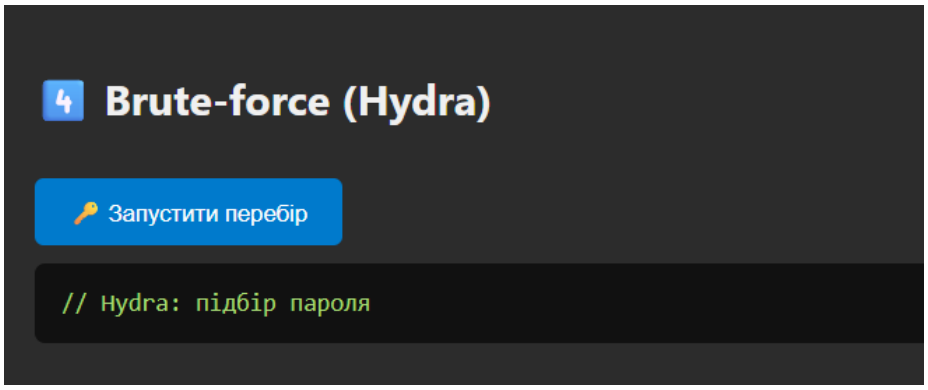


Рисунок 3.6 – Brute-force атака

Сканування та інвентаризація мережі – інструменти Nmap та enum4linux використовувалися для дослідження структури мережі та виявлення сервісів. Nmap виявив відкриті порти та сервіси (SMB, HTTP, SSH), а enum4linux – спільні мережеві ресурси та SID домену. Дані, отримані під час експерименту, були зафіксовані SIEM-системами.

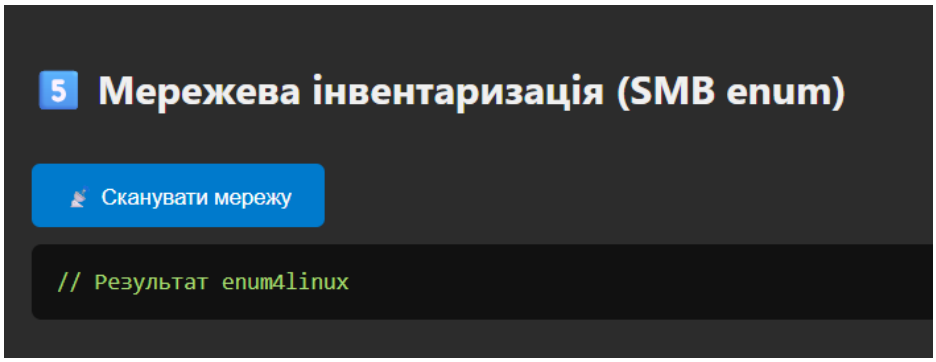


Рисунок 3.7 – Сканування та інвентаризація мережі

У процесі тестування атак велика увага приділялася не лише виявленню вразливостей, але й реакції систем захисту в реальному часі. Основними засобами моніторингу та реагування у середовищі були Wazuh (HIDS – Host-based Intrusion Detection System) і Splunk (SIEM – Security Information and Event Management).

Wazuh – усі віртуальні машини були оснащені агентами Wazuh, які збирали інформацію про зміни у файловій системі, дії користувачів, системні події тощо. Особливу увагу зосереджено на компоненті File Integrity Monitoring (FIM). Під

час тестової атаки на файл config.php веб-додатку (який зберігав критичні параметри конфігурації бази даних), було змінено вміст файлу, що одразу зафіксував агент Wazuh. Подія класифікувалася відповідно до фреймворку MITRE ATT&CK як техніка T1070 – Indicator Removal on Host, яка вказує на можливі спроби приховати сліди атаки.

Wazuh також відстежив спроби brute-force-автентифікації та сканування мережі. На основі наперед налаштованих правил було згенеровано повідомлення з рівнем небезпеки «high», які автоматично потрапляли у лог-файл системи.

Splunk – система Splunk Free Edition була налаштована як індексатор з увімкненим модулем HTTP Event Collector (HEC). Вона отримувала структуровані журнали подій із Wazuh через Logstash, який слугував проміжним агрегатором (pipeline input – filter – output). Події з високим пріоритетом (наприклад, SQL-атаки) потрапляли до індексу wazuh_alerts, після чого могли бути знайдені за допомогою запитів, таких як:

```
index=wazuh_alerts severity>=3
```

У представленні побудовано: таймсерію активності атак (time series); heatmap спрацювань за категоріями MITRE; дашборд подій за типами атак. Ці візуальні засоби дозволили швидко ідентифікувати пік активності атак, найбільш часті цілі атак, а також оцінити загальний стан кіберзахисту системи.

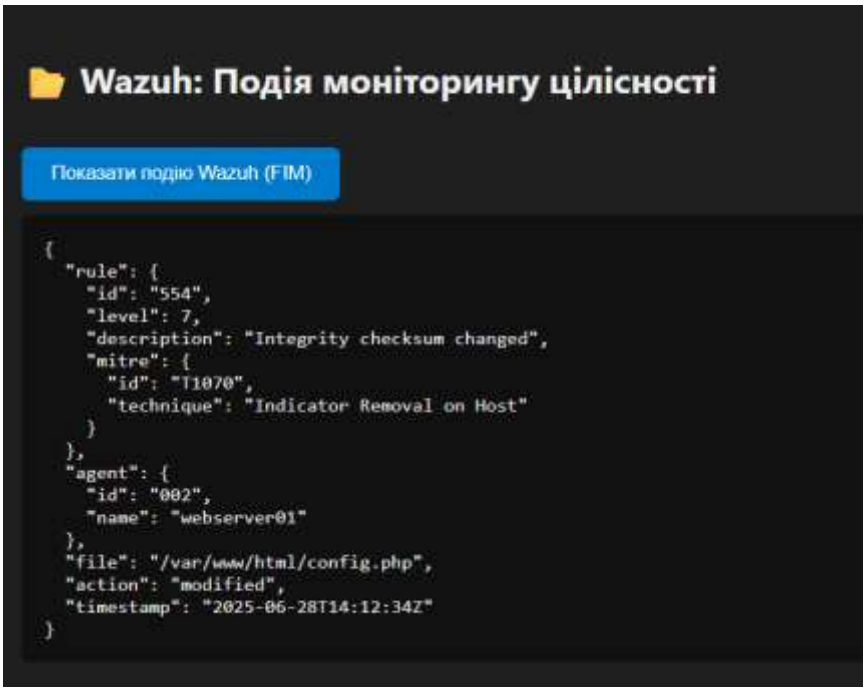


Рисунок 3.8 – Wazuh: подія моніторингу цілісності

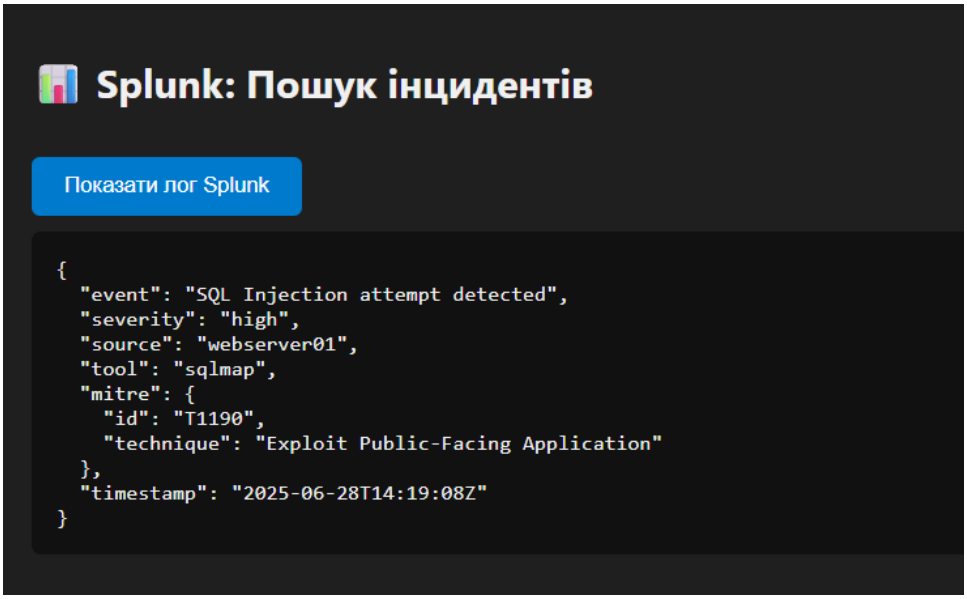


Рисунок 3.9 – Splunk: пошук інцидентів

У сучасних інформаційно-комунікаційних системах забезпечення безпеки додатків має бути інтегроване безпосередньо у процес розробки програмного забезпечення. Саме концепція DevSecOps передбачає впровадження заходів

безпеки на ранніх етапах життєвого циклу застосунків, зокрема через автоматизоване тестування вразливостей у межах CI/CD-конвеєра. У рамках роботи було реалізовано інтеграцію двох сучасних сканерів додатків: AppScan (від IBM) та Acunetix (від Invicti), кожен з яких виконував специфічні функції у відповідній частині інфраструктури.

AppScan – інтеграція з CI/CD-процесами. AppScan було налаштовано на автоматичний запуск при кожному створенні pull request або оновленні коду в репозиторії (наприклад, GitLab CI або Jenkins). Це дозволяє забезпечити принцип «Shift Left», за якого вразливості виявляються ще до етапу деплойменту. AppScan аналізує змінені файли, перевіряє взаємодію з базами даних, ідентифікує спроби обходу автентифікації, відсутність захисту API-запитів та інші критичні помилки.

Після завершення сканування AppScan формує звіт у форматі .json або .xml, який автоматично зберігається в артефактному сховищі CI-системи. Додатково, цей звіт імпортується до SIEM-платформи Splunk, де подальша обробка здійснюється за допомогою створених дашбордів та правил кореляції.

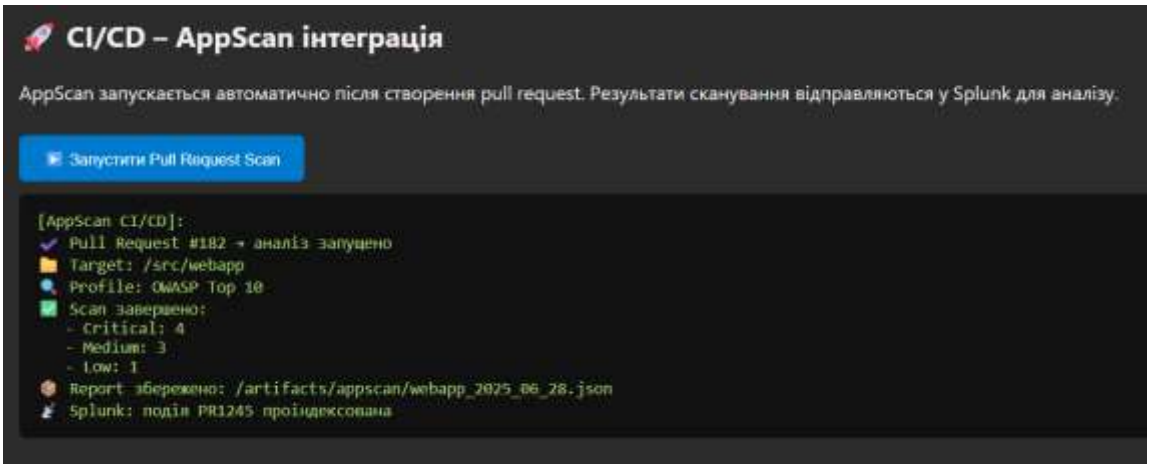


Рисунок 3.10 – AppScan – інтеграція з CI/CD-процесами

Acunetix – щоденне сканування продуктивного середовища. Acunetix було

розгорнуто на веб-сервері WebServer01, де він виконує автоматичне щоденне сканування встановлених додатків. Для цього налаштовано завдання cron, яке запускає Quick Scan кожного дня о 02:00. Результати експортуються у форматі .json і автоматично передаються в систему Wazuh через REST API.

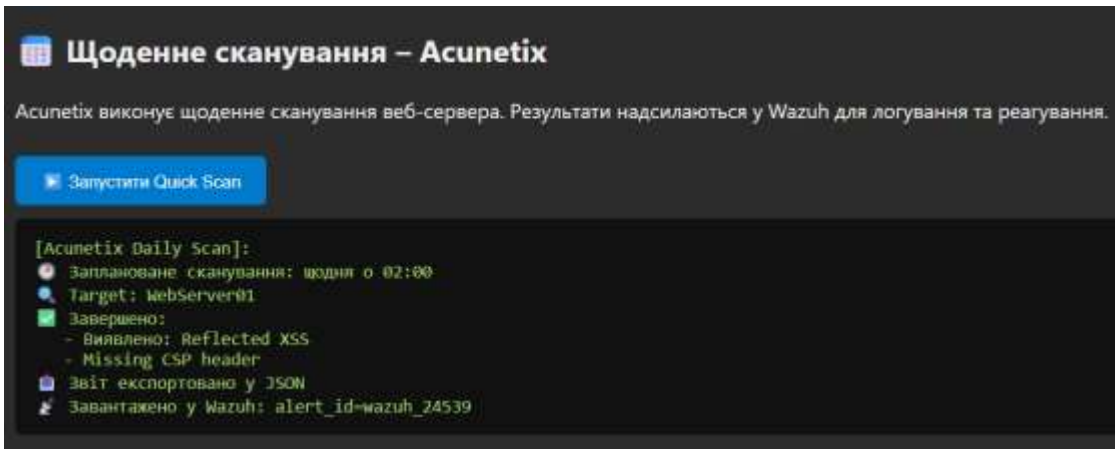


Рисунок 3.11 – Acunetix – щоденне сканування продуктивного середовища

Такий підхід дозволяє підтримувати високий рівень видимості вразливостей навіть у постійно змінюваному середовищі. Якщо адміністратор оновлює CMS або вносить зміни до конфігурації, Acunetix виявляє нові загрози вже наступного дня. Завдяки цій інтеграції реалізовано комплексне, багаторівневе виявлення загроз:

- На етапі розробки – автоматичний сканинг pull request’ів (AppScan);
- На етапі деплойменту та експлуатації – щоденне виявлення нових вразливостей (Acunetix);
- Аналітика – обробка результатів у Splunk і Wazuh;
- Інформування – відправлення alert-сповіщень при критичних загрозах.

Це дозволяє скоротити час на усунення вразливостей, зменшити ризики експлуатації недоліків у коді та покращити загальний рівень безпеки додатків.

У сучасній практиці кібербезпеки одним із важливих інструментів для навчання, тестування та демонстрації є інтерактивні симулятори, які дозволяють відтворити повний цикл взаємодії між атакуючим середовищем, захисними засобами та системами моніторингу. З огляду на це, у межах роботи було реалізовано навчальну демонстраційну модель у форматі веб-інтерфейсу, яка імітує виконання атак, реакцію SIEM-систем і запуск сканувань у CI/CD-конвеєрі.

Мета створеної моделі – надати зручний, наочний спосіб вивчення захисту додатків на рівні корпоративної мережі. Вона дозволяє імітувати поведінку реального пентестера, адміністратора безпеки або системи моніторингу, при цьому не вимагаючи складного налаштування віртуального середовища або запуску реальних сканерів і атак. Такий підхід дозволяє проводити презентації, захисти, навчання та демонстрації навіть у середовищах без підключення до мережі або із обмеженим доступом до вразливого ПЗ. Імітація атак:

- SQLi (SQL-ін’єкція) – демонстрація використання sqlmap для вилучення даних з бази;
- XSS – показ запиту з ін’єкцією JavaScript-коду;
- CSRF – формування шкідливого POST-запиту без токена захисту.

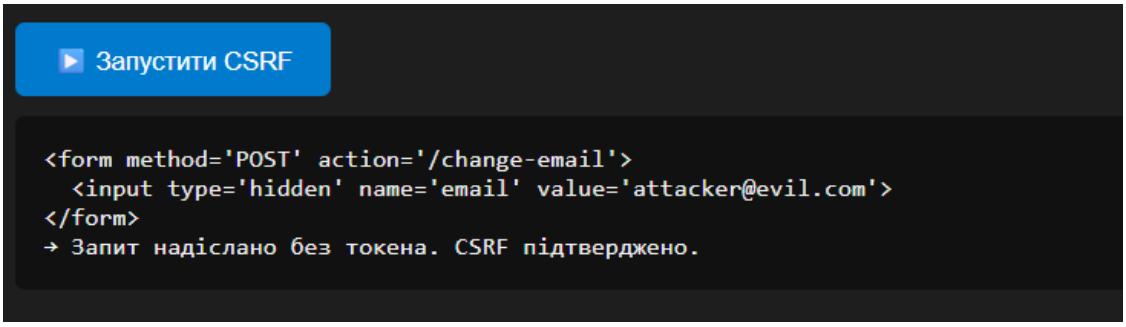


Рисунок 3.12 – Запуск CSRF

Відображення логів засобів безпеки: Wazuh – показ спрацювання FIM-механізму при зміні конфігураційного файлу, з ідентифікацією MITRE-ідентифікатора техніки (T1070).

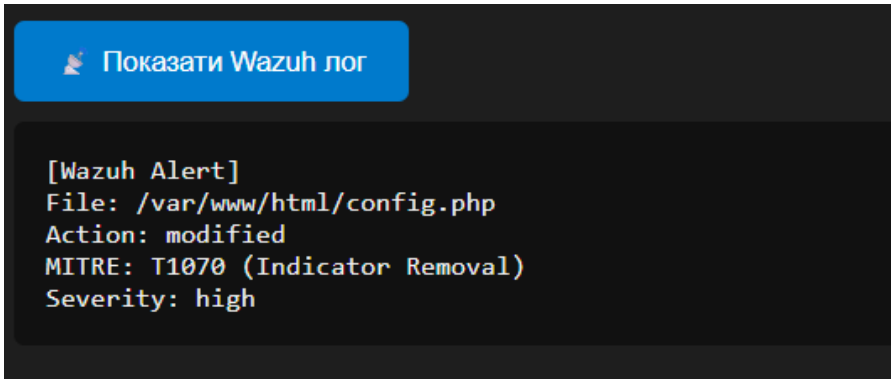


Рисунок 3.13 – Засіб безпеки Wazuh

Splunk – журнал подій у форматі JSON, що показує спробу SQL-атаки (T1190), дату, джерело, тип загрози.

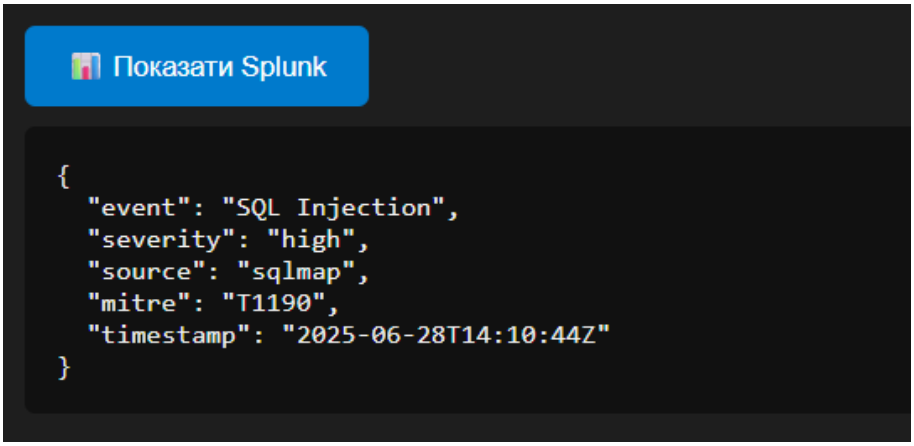


Рисунок 3.14 – Запуск безпеки Splunk

Сканування та аудит безпеки: ZAP та Burp Suite – імітація результатів виявлення типових вразливостей (SQLi, XSS, CSRF) через базові сканування.

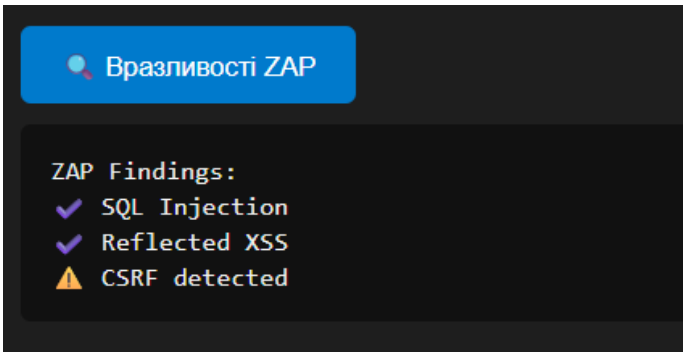


Рисунок 3.15 – Сканування та аудит безпеки ZAP та Burp Suite

AppScan (CI/CD) – запуск сканування коду після створення pull request з виведенням результатів, які автоматично відправляються до Splunk.

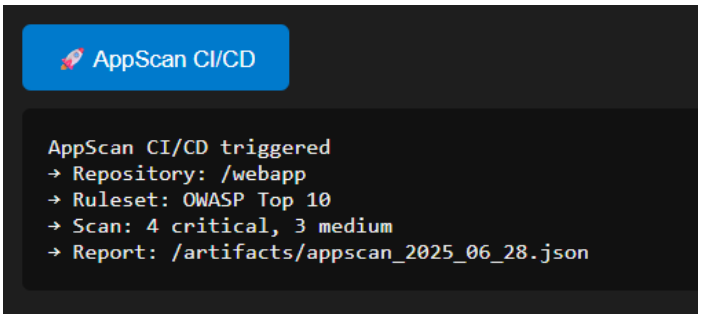


Рисунок 3.16 – Сканування та аудит безпеки AppScan (CI/CD)

Acunetix – щоденне автоматичне сканування продуктивного сервера, імпорт звітів у систему логування.

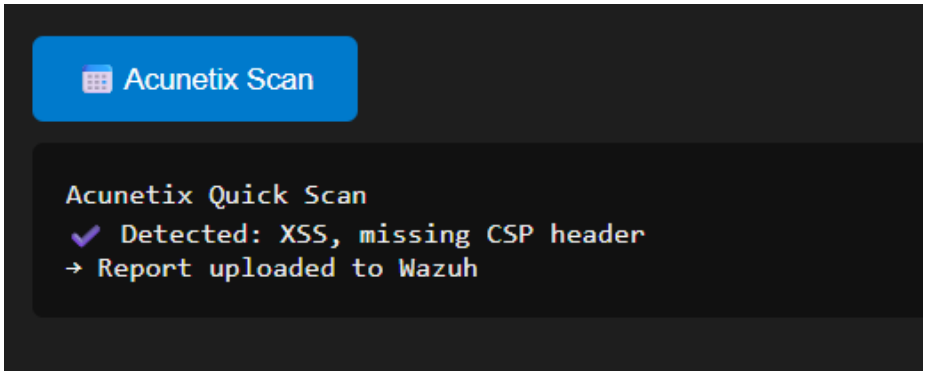


Рисунок 3.17 – Сканування та аудит безпеки Acunetix

Реалізована система не лише показала ефективність запропонованого середовища, але й стала інструментом для візуального та інтуїтивного розуміння принципів роботи: як атаки впливають на системи; як швидко та точно реагують захисні інструменти; які події фіксуються та як відображаються в логах.

Такий підхід значно покращує рівень засвоєння матеріалу студентами, дозволяє викладачам моделювати навчальні інциденти, а також використовується як навчальний кіберполігон для практичного закріплення знань.

Отже, розроблена демонстраційна модель візуалізації успішно реалізує повний цикл «атака – виявлення – аналіз – звітність», відображаючи реальну взаємодію між компонентами корпоративної системи безпеки. Вона є як функціональним, так і дидактичним інструментом, що демонструє ключові концепти захисту додатків у сучасному середовищі. Це підвищує не лише ефективність дослідження, а й якість освітнього процесу в сфері кібербезпеки.

На завершальному етапі дослідження було проведено порівняльний аналіз ефективності реалізованих механізмів захисту, відповідно до п’яти ключових векторів атак, що найчастіше зустрічаються у корпоративних інформаційно-комунікаційних системах. Кожен вектор оцінювався за чотирма параметрами: тип атаки, інструмент, який її реалізує, засоби виявлення (сканери вразливостей), а також реакція з боку систем виявлення та реагування (SIEM/логування).

Таблиця 3.9

Порівняльний аналіз ефективності

Вектор атаки	**Інструмент атаки**	**Виявлено засобами захисту**	**Реакція SIEM / логування**
SQLi	sqlmap	AppScan, ZAP Proxy, Burp Suite	Wazuh (MITRE T1190), Splunk (індексована подія)
XSS	Burp Suite	ZAP, AppScan	Виявлення заголовків, логування Reflected-XSS
CSRF	Burp Repeater	Частково (тільки AppScan)	Відсутність захисту — низький рівень ефективності

Brute-force	Hydra	Wazuh, Burp (Intruder)	Alert після >5 спроб автентифікації (MITRE T1110)
SMB Enumeration	enum4linux	—	Логування через auditd, без активації попереджень

SQL-ін’єкції були одними з найефективніших атак, що дозволили витягнути критичну інформацію з бази даних через вразливі параметри у URL. Інструмент sqlmap успішно ідентифікував точки ін’єкції. Сканери AppScan, ZAP і Burp виявили ці вразливості при автоматичному та ручному аналізі. Події фіксувалися у SIEM (Wazuh) з класифікацією MITRE T1190 – Exploit Public-Facing Application, а також індексувалися у Splunk, що підтверджувало ефективність як виявлення, так і реагування.

Використовуючи Burp Suite, вдалося виконати успішне впровадження JavaScript-коду у відповідь сервера. OWASP ZAP і AppScan виявили відсутність фільтрації, що дало змогу створити Reflected XSS. Хоча SIEM не класифікував подію за MITRE, було зафіксовано порушення політик HTTP-заголовків, зокрема CSP (Content Security Policy), що вказує на потенційну загрозу.

Сценарій CSRF реалізовано за допомогою автоматизованої форми, що відправляла запит без підтвердження користувача. Лише AppScan частково виявив відсутність CSRF-токенів у формах. Захисні інструменти, на жаль, не змогли зафіксувати CSRF-інцидент у SIEM, що свідчить про недостатню реалізацію захисту від цього вектору у поточному середовищі. Виявлено потребу у впровадженні токенів та контролю Referer-заголовків.

Використовуючи Hydra, було здійснено перебір паролів до сторінки автентифікації. Після п’ятої невдалої спроби Wazuh активував сповіщення про можливу brute-force-атаку. Подія класифікована за MITRE T1110 – Brute Force. Також Burp Intruder підтвердив успішний підбір, але засоби автоматизованої блокування (lockout) не були реалізовані, що є потенційною вразливістю.

За допомогою enum4linux було здійснено сканування файлового сервера на

відкриті спільні ресурси. Хоча засоби захисту не змогли виявити цю активність у реальному часі, вона була зафіксована у логах auditd. Відсутність негайної реакції свідчить про потребу в інтеграції SMB-спостереження в Wazuh для підвищення видимості вразливостей рівня ОС.

Результати експериментальної перевірки підтвердили, що побудоване середовище забезпечує коректне, відтворюване та методично обґрунтоване оцінювання ефективності засобів захисту додатків у корпоративній мережі. Усі основні компоненти системи, зокрема сканери вразливостей (Burp, ZAP, AppScan, Acunetix), засоби моніторингу (Wazuh, Splunk) та інструменти тестування (Kali Linux), були успішно інтегровані, налаштовані й перевірені. Ключові досягнення:

- SQLi, XSS, brute-force – успішно виявлені і задокументовані в журналах SIEM;
- CSRF, SMB enum – частково виявлені або потребують додаткових заходів;
- CI/CD-інтеграція – підтверджено можливість раннього виявлення вразливостей завдяки Shift Left-стратегії;
- Інтерфейс візуалізації – дозволив підвищити наочність аналізу та спростити демонстрацію результатів.

Отримані результати доводять, що розроблена методика є масштабованою, практично орієнтованою та може застосовуватися для дослідження різних груп загроз у корпоративних ІКС. Інтегрована інфраструктура із застосуванням Burp Suite, ZAP Proxy, AppScan, Acunetix, Wazuh та Splunk забезпечує повний цикл обробки інцидентів — від виявлення вразливостей до їхньої аналітичної кореляції та документування. Виявлені недоліки (насамперед у частині CSRF та SMB-моніторингу) визначають напрями подальшого вдосконалення системи.

Таким чином, експериментальна частина роботи підтвердила працездатність запропонованого підходу, відповідність тестового середовища методологічним вимогам та його доцільність для практичного впровадження в освітніх і дослідницьких цілях.

У процесі тестування були виконані типові атаки (SQLi, XSS, CSRF, brute force, мережеве сканування), результати яких успішно зафіксовані системами моніторингу. Завдяки використанню матриці MITRE ATT&CK вдалося точно ідентифікувати тип загроз та їх кореляцію з виявленими подіями. Окремо було реалізовано навчальну візуалізацію роботи інструментів безпеки через веб-інтерфейс, що дозволяє демонструвати процеси виявлення та реагування в інтерактивному форматі. Таким чином, результати експериментального дослідження підтверджують, що інтеграція сучасних засобів захисту в корпоративну інфраструктуру, з урахуванням принципів Zero Trust та DevSecOps, забезпечує високий рівень виявлення, реагування та зменшення впливу кіберзагроз.

Висновки до 3 розділу

У третьому розділі було реалізовано практичне дослідження ефективності обраних методів захисту даних на рівні додатків у корпоративних інформаційно-комунікаційних системах. Для цього розроблено та налаштовано тестове середовище, яке моделює типову корпоративну інфраструктуру з поділом на демілітаризовану зону (DMZ), внутрішню мережу та сегмент тестування. Це дозволило у безпечних умовах перевірити роботу захисних механізмів без впливу на реальні системи.

У середовищі було реалізовано тестування таких інструментів, як Burp Suite, AppScan, Acunetix, ZAP Proxy, а також систем моніторингу та реагування – Splunk, Wazuh, LogRhythm та ін. Проведено активне та пасивне сканування

додатків, симуляцію типових атак (SQL-ін’єкції, XSS, CSRF) і моніторинг реакції систем на них. На основі отриманих результатів було виконано порівняльну оцінку ефективності інструментів за критеріями точності, кількості хибнопозитивних спрацювань, продуктивності, гнучкості інтеграції та зручності використання.

Аналіз показав, що Burp Suite Pro та Splunk демонструють найвищі показники ефективності, однак вимагають більшої експертної підготовки та ресурсів. Інструменти open-source, такі як Wazuh чи ZAP Proxy, є доцільним вибором для середовищ із обмеженим бюджетом. Результати експериментів підтвердили важливість використання багаторівневого підходу до безпеки, що охоплює як виявлення вразливостей, так і своєчасне реагування на інциденти.

Таким чином, експериментальна частина підтвердила практичну ефективність обраних методів захисту на рівні додатків і надала обґрунтовану базу для формування рекомендацій щодо їх впровадження в реальних умовах корпоративних систем.

РОЗДІЛ 4 РОЗРОБКА РЕКОМЕНДАЦІЙ ТА ПРАКТИЧНИХ РІШЕНЬ

4.1 Рекомендації щодо вибору методів захисту корпоративних мереж

У сучасних умовах стрімкого зростання кіберзагроз традиційні моделі захисту корпоративних мереж, що ґрунтуються на побудові "периметра безпеки", вже не є ефективними. Класичний підхід, при якому внутрішня мережа вважається довіреною, а зовнішня – потенційно ворожою, не враховує сучасних сценаріїв атак, таких як внутрішнє шахрайство, компрометація облікових записів або атаки через VPN, мобільні пристрої та хмарні сервіси. Саме тому дедалі більшого поширення набуває концепція Zero Trust Architecture (ZTA) – архітектура нульової довіри [24].

Основна ідея Zero Trust полягає в принципі: «нікому не довіряй, завжди перевіряй». Це означає, що жоден користувач, пристрій, додаток чи сервіс не може автоматично вважатися безпечним – незалежно від того, чи він знаходиться всередині організації, чи поза її межами. Усі запити на доступ до ресурсів повинні проходити аутентифікацію, авторизацію та перевірку відповідно до контексту запиту: хто, звідки, з якого пристрою, до якого ресурсу, з яким рівнем ризику. Основні принципи Zero Trust:

- Постійна перевірка особи (Continuous Authentication): доступ надається лише після успішної багатофакторної автентифікації (MFA), підтвердження пристрою та перевірки геолокації, IP, поведінкових параметрів тощо.
- Мінімальні привілеї (Least Privilege): користувачі та процеси отримують доступ лише до тих ресурсів, які необхідні для виконання їхніх функцій.
- Мікросегментація мережі: корпоративна мережа ділиться на логічні

сегменти, між якими встановлюються правила контролю доступу, що зменшує «поверхню атаки».

- Динамічне прийняття рішень на основі ризиків (Risk-Adaptive Access Control): враховуються контекстні фактори, такі як час доби, тип пристрою, аномалії в поведінці користувача.

- Безперервний моніторинг та аналітика: усі дії в мережі відслідковуються за допомогою систем SIEM, UEBA, XDR тощо, з автоматичним реагуванням на підозрілі події.

Практична реалізація Zero Trust може включати:

- Впровадження MFA на всіх рівнях доступу, включно з VPN, веб-додатками, внутрішніми порталами та привілейованими системами.

- Застосування рішень для управління доступом до додатків (Identity-Aware Proxy, SDP – Software-Defined Perimeter).

- Впровадження PAM (Privileged Access Management) для контролю доступу адміністраторів і захисту облікових записів із високими правами.

- Використання NAC (Network Access Control) для контролю стану пристроїв при підключенні до мережі.

- Розмежування мережі за допомогою мікросегментації (наприклад, засобами VMware NSX, Cisco ACI) та міжсегментного фаєрволінгу.

Переваги впровадження Zero Trust: суттєве зменшення ризику внутрішніх атак, включно з тими, які виникають унаслідок фішингу чи компрометації облікових записів; обмеження бокового руху загроз (Lateral Movement) – навіть якщо зловмисник потрапить у систему, він не зможе поширитися мережею; гнучкість і масштабованість – модель Zero Trust адаптивна до змін у структурі організації, дозволяє легко інтегрувати хмарні сервіси, мобільні пристрої та IoT;

підвищення відповідності стандартам безпеки: Zero Trust підтримує відповідність вимогам NIST SP 800-207, ISO/IEC 27001, GDPR, HIPAA та ін.

В умовах гібридних робочих моделей, використання хмарних сервісів і зростання кількості віддалених користувачів архітектура Zero Trust є фундаментом сучасної кібербезпеки. Її впровадження дозволяє організаціям перейти від застарілих моделей периметрового захисту до гнучкої, динамічної системи безпеки, що адаптується до змін ризиків і постійно перевіряє довіру.

Одним з найбільш ефективних підходів до побудови гнучкої системи мережевої безпеки є мікросегментація – розділення корпоративної мережі на дрібніші логічні сегменти з індивідуальними правилами контролю доступу. На відміну від традиційного зонування, де зазвичай розділяють лише внутрішню, DMZ та гостьову мережі, мікросегментація дозволяє ізолювати окремі сервіси, додатки, відділи чи навіть користувачів. Мережа може бути розділена за: функціональним принципом (наприклад, бухгалтерія, розробка, клієнтський портал); рівнем довіри (наприклад, захищені ресурси, звичайні сервіси, незахищені пристрої); типом трафіку (зовнішній, внутрішній, управлінський, резервний) [25].

Кожен сегмент ізолюється міжмережевими фаєрволами або іншим механізмом доступу – наприклад, віртуальними фаєрволами, VLAN ACL, або засобами мікросегментації від VMware NSX, Cisco ACI, Fortinet або Palo Alto. Переваги мікросегментації:

- Зменшення «радіуса прориву» (blast radius): у разі компрометації одного вузла зловмисник не зможе легко пересуватися мережею до інших ресурсів.
- Підвищення контролю: можна точно визначити, який пристрій або служба має доступ до яких даних.
- Покращення відповідності: легше реалізувати вимоги стандартів

(ISO/IEC 27001, NIST, GDPR), які вимагають ізоляції критичних даних.

Для побудови ефективної мікросегментованої архітектури важливо поєднувати її з засобами інвентаризації активів, моніторингу трафіку, а також із системами автоматичного створення політик доступу (наприклад, на базі поведінки користувачів або застосунків).

Для виявлення та протидії атакам на мережевому рівні широко застосовуються системи UTM (Unified Threat Management), IDS (Intrusion Detection Systems) та IPS (Intrusion Prevention Systems). Ці системи можуть працювати автономно або у зв'язці з фаєрволами нового покоління (NGFW), формуючи багаторівневу систему виявлення та нейтралізації загроз.

- UTM – це комплексні пристрої, які поєднують функції фаєрволу, антивірусу, IDS/IPS, VPN, фільтрації вмісту та контролю додатків. Вони ідеальні для середнього бізнесу.

- IDS – аналізують трафік та сповіщають про потенційні загрози (наприклад, спроби сканування портів, аномалії, сигнатури атак).

- IPS – додатково можуть блокувати підозрілу активність у режимі реального часу, зупиняючи атаки ще до їхнього розвитку.

Впровадження IDS/IPS/UTM-рішень можливе: на межі мережі (периметр, підключення до Інтернету), між сегментами внутрішньої мережі (наприклад, між серверною зоною та користувацькою), на кінцевих точках (Endpoint IDS) у комбінації з EDR/XDR. Популярні рішення: Suricata, Snort, Cisco Firepower, FortiGate NGFW, Palo Alto NGFW, Check Point, Zeek (ex-Bro). Переваги: своєчасне виявлення вторгнень та реагування на інциденти; контроль трафіку та додатків, обмеження шкідливої або небажаної активності; інтеграція із SIEM для розширеної аналітики подій. Таким чином, UTM/IDS/IPS у поєднанні з мікросегментацією створюють ефективний бар'єр для виявлення та стримування атак на корпоративну мережу.

Контроль доступу до інформаційних ресурсів корпоративної мережі є критично важливим компонентом системи безпеки. Один із найефективніших способів підвищення безпеки – впровадження багатофакторної автентифікації (MFA), а також застосування ролей-орієнтованого доступу (RBAC) та управління привілейованими доступами (PAM) [26].

Багатофакторна автентифікація (MFA) – вимагає від користувача пройти не лише парольну автентифікацію, а й надати додатковий фактор підтвердження – наприклад: одноразовий код (OTP); біометричні дані; токен чи апаратний ключ (YubiKey); Push-повідомлення в мобільному застосунку (Microsoft Authenticator, Duo). MFA суттєво знижує ризик несанкціонованого доступу навіть у разі компрометації пароля, а також є обов’язковим для критичних систем згідно з рекомендаціями NIST SP800-63.

Ролі-орієнтований контроль доступу (RBAC) – передбачає створення ролей з фіксованими правами, які потім призначаються користувачам. Це дозволяє: централізовано керувати доступом; автоматизувати надання прав (Onboarding/Offboarding); спрощувати аудит і контроль привілеїв.

Управління привілейованим доступом (PAM) – це спеціалізовані системи (наприклад, CyberArk, BeyondTrust, Microsoft PIM), які: зберігають облікові дані у зашифрованих сховищах (vault); записують усі сесії адміністраторів; дозволяють надавати тимчасовий доступ на запит (Just-in-Time). Переваги:

- Мінімізація людського фактору в компрометації облікових даних;
- Скорочення ризику несанкціонованих змін у системі;
- Спрощення відповідності нормативам (ISO 27001, GDPR, SOX).

Таким чином, комбінація MFA + RBAC + PAM дозволяє значно посилити ідентифікацію, керування доступом та прозорість дій у системі. Це особливо актуально в умовах зростання атак через фішинг, соціальну інженерію та зловживання привілеями.

Сучасні корпоративні мережі все частіше мають справу з віддаленими працівниками, філіями, партнерами чи постачальниками, які потребують доступу до внутрішніх ресурсів. Це створює нові вектори атак і вимагає належного захисту каналів зв’язку, зокрема VPN, Wi-Fi мережі та інструментів віддаленого доступу. Захист VPN:

- Використання сучасних VPN-протоколів: рекомендовано застосовувати IPSec або SSL/TLS із 256-бітним шифруванням, наприклад OpenVPN, IKEv2 або WireGuard.

- Фільтрація трафіку через периметральні брандмауери: доступ через VPN має контролюватися правилами фаєрволу, що запобігає доступу до критичних ресурсів без належного дозволу.

- Обов’язкове застосування MFA: усі користувачі, які підключаються до VPN, мають проходити багатофакторну автентифікацію, щоб зменшити ризик компрометації облікових записів.

Захист Wi-Fi:

- Використання WPA3: як найсучаснішого стандарту бездротового шифрування, що захищає від атак типу «brute-force» і «dictionary attack».

- Розмежування трафіку: корпоративний Wi-Fi повинен бути ізольований від гостьового, а кожен сегмент повинен мати власні обмеження (наприклад, VLAN із відповідним рівнем доступу).

- Автентифікація через RADIUS: централізований контроль доступу до Wi-Fi із можливістю ведення журналів активності.

Віддалений доступ:

- Застосування Virtual Desktop Infrastructure (VDI) або Remote Desktop Gateway, які дозволяють забезпечити доступ до середовища без виведення

конфіденційних даних за межі локальної мережі.

- Впровадження контейнеризованих середовищ або ізольованих віртуальних робочих станцій для фрилансерів чи зовнішніх підрядників.

- Захист каналів зв'язку забезпечує конфіденційність, цілісність і доступність даних, що є основою інформаційної безпеки корпоративної мережі.

Оновлення операційних систем, програмного забезпечення, драйверів і мережевих пристроїв – критичний елемент кібергігієни. Вразливості в незахищених компонентах можуть бути використані зловмисниками для проникнення в систему навіть без активних дій користувача. Основні заходи:

- Автоматизація процесу оновлення: за допомогою централізованих систем керування (WSUS, Microsoft Intune, Red Hat Satellite, SCCM).

- Інвентаризація активів і ПЗ: для визначення версій, виявлення застарілих або незахищених компонентів.

- Тестування перед розгортанням: критично важливо для великих мереж – перевірити сумісність оновлень на тестових середовищах.

Підсилення моніторингу: використання систем EDR/XDR (Endpoint Detection and Response / Extended Detection and Response), які виявляють аномалії, Zero-Day-атаки та реакцію на загрози на рівні кінцевих пристроїв; інтеграція з SIEM для виявлення інцидентів безпеки та автоматичного повідомлення адміністратора. Таким чином, регулярне оновлення ПЗ не лише запобігає атакам через відомі вразливості, а й забезпечує відповідність стандартам безпеки.

Навіть найкращі технічні засоби захисту можуть бути безсилі, якщо працівники організації не дотримуються базових принципів інформаційної безпеки. Більшість сучасних атак починається з людського фактору – фішингових листів, соціальної інженерії або недбалого поводження з даними. Рекомендовані заходи [27]:

- Регулярне навчання персоналу: інтерактивні курси, тренінги, тестові фішингові атаки, сертифікація користувачів.
- Проведення інформаційних кампаній: бюлетені, плакати, внутрішній портал із порадами щодо безпеки.
- Програма «security champions» – залучення мотивованих співробітників до просування культури безпеки в своїх командах.
- Формалізація процедур: інструкції з повідомлення про підозрілі дії, політика паролів, правила роботи з конфіденційною інформацією.

Наявність свідомого і підготовленого персоналу значно знижує ймовірність компрометації системи через прості соціотехнічні методи.

Безперервне спостереження за станом інформаційної безпеки – це основа проактивного захисту. Завдяки моніторингу, тестуванню та оцінці можна не лише виявляти атаки в реальному часі, а й запобігати їм на ранніх етапах. Основні напрями:

- Оцінка ризиків: виявлення активів, загроз, вразливостей, моделювання сценаріїв атаки.
- Технічний аудит: перевірка конфігурацій, політик, налаштувань систем доступу, фаєрволів, логування.
- Тести на проникнення (penetration testing): симуляція реальних атак для виявлення слабких місць.
- Періодичні зовнішні та внутрішні аудити: залучення незалежних експертів підвищує довіру та об’єктивність оцінки.

Моніторинг:

- Впровадження SIEM (Security Information and Event Management) – Splunk, ELK, QRadar – для централізованого збирання логів.

- Застосування NDR (Network Detection and Response) для виявлення аномалій у трафіку.
- Визначення KPI (ключових показників ефективності) та KRI (ключових показників ризику) для безпеки – наприклад, середній час реагування на інцидент, кількість виявлених фішингових листів тощо.

Постійний моніторинг у поєднанні з оцінкою ризиків дозволяє підтримувати високий рівень кіберстійкості організації та оперативно реагувати на потенційні загрози.

Ефективна кібербезпека неможлива без чітко визначеного та протестованого плану реагування на інциденти (Incident Response Plan, IRP). Такий план містить послідовність дій, ролі та відповідальність кожного учасника у разі кіберінциденту – від виявлення до відновлення. Ключові елементи IRP:

- Формування команди реагування (CSIRT/SOC): із розподілом ролей між аналітиками, адміністраторами, юристами та PR-фахівцями.
- Класифікація та пріоритезація інцидентів: за типом, масштабом, ступенем ризику.
- Чіткий план дій на кожному етапі: ідентифікація, стримування, ліквідація, відновлення, аналіз наслідків.
- Реєстрація подій та ведення журналів: для подальшого розслідування й відповідальності.

Практика:

- Проведення навчальних симуляцій (tabletop exercises): для перевірки злагодженості дій команди.
- Red Team / Blue Team тестування: імітація атак і оцінка ефективності захисту.

- Постінцидентний аналіз (lessons learned): і внесення змін до IRP на основі отриманого досвіду.

Наявність і регулярне оновлення IRP дозволяє значно скоротити час реагування, зменшити збитки та зберегти репутацію організації.

Обмін інформацією про кіберзагрози між організаціями — важливий компонент колективної кіберстійкості. Компанії можуть виявляти атаки раніше, зменшувати ризики та діяти проактивно. Практики:

- Участь в ISAC (Information Sharing and Analysis Centers): галузеві платформи обміну розвідданими про загрози.

- Використання Threat Intelligence feeds: для автоматизованого поповнення баз шкідливої активності (IP, домени, хеші тощо).

- Співпраця з CERT, правоохоронними органами: особливо при масованих або складних атаках.

Відповідність стандартам: NIST SP800-207 (Zero Trust Architecture); CISA Cybersecurity Performance Goals; NIS2 Directive (ЄС); GDPR (Загальний регламент захисту даних); ISO/IEC 27001 – міжнародний стандарт системи управління інформаційною безпекою. Виконання цих стандартів не лише підвищує кіберстійкість, а й дозволяє організаціям довести свою відповідальність перед клієнтами, партнерами та державними органами.

Сучасна практика безпеки передбачає вбудовування захисту вже на етапі проектування IT-рішень та систем (Secure-by-Design), а не додавання його після завершення розробки. Основні методи:

- Threat Modeling: моделювання загроз, визначення потенційних векторів атак на ранньому етапі.

- Мапінг ризиків (Risk Mapping): оцінка вразливих точок системи з урахуванням типу даних та користувачів.

- Стандарти розробки: впровадження надійних шаблонів програмування, безпечних бібліотек, перевірених фреймворків.
- Code review та security audit: обов’язкова перевірка коду на наявність вразливостей.
- DevSecOps: інтеграція процесів безпеки в CI/CD, автоматизовані тести на вразливості, SAST/DAST сканування.

Таким чином, принцип Secure-by-Design дає змогу виявляти та усувати вразливості до того, як системи потраплять у продакшн, економлячи ресурси та підвищуючи довіру користувачів.

Таблиця 4.1

Практична реалізація рекомендацій

Етап	**Міра**	**Інструмент / Приклад**
Підготовчий	Ідентифікація критичних систем, моделювання загроз	Threat Modeling
Архітектурний	Побудова сегментованої мережі, реалізація Zero Trust	Мікросегментація, VPN з MFA, PAM
Технічний	Налаштування UTM/IDS/IPS, фаєрволів	Fortinet, Palo Alto, Zeek, Suricata
Операційний	Централізований менеджмент патчів, захищене оновлення	WSUS, SCCM, EDR
HR / ГО	Навчання персоналу, впровадження політик	KnowBe4, PhishMe, внутрішні тренінги
Моніторинг	Централізований аналіз логів, поведінковий аналіз	SIEM (Splunk, ELK), NDR (Darktrace)
Реагування	План IRP, тестування, реагування на інциденти	Playbook, Red Team, Tabletop Exercises
Аудит / Обмін	Відповідність стандартам, обмін розвідданими	ISO27001, GDPR, FS-ISAC, CERT-UA

Захист корпоративної мережі в сучасному цифровому середовищі вимагає багаторівневої, гнучкої та адаптивної стратегії, що охоплює всі компоненти IT-інфраструктури – від користувача до ядра мережі. У роботі були сформовані ключові рекомендації щодо впровадження:

- архітектури Zero Trust і мікросегментації;
- систем MFA, PAM, RBAC;
- механізмів реагування на інциденти (IRP);
- навчання персоналу, безперервного моніторингу, оновлення ПЗ;
- а також принципів Secure-by-Design і відповідності міжнародним стандартам.

Комплексне застосування цих рішень дозволяє: протистояти внутрішнім та зовнішнім загрозам; оперативно виявляти та реагувати на інциденти; підвищити загальну кіберстійкість організації та відповідати вимогам сучасного регуляторного середовища. Таким чином, сформована модель забезпечення інформаційної безпеки є ефективною основою для побудови захищеної, масштабованої та надійної корпоративної мережі.

4.2 Впровадження комбінованих стратегій захисту на рівні додатків

Сучасна практика розробки програмного забезпечення потребує впровадження безпеки ще на ранніх етапах життєвого циклу розробки — ця стратегія відома як Shift-Left Security. Основна ідея полягає в тому, щоб «зсунути» заходи безпеки ліворуч по діаграмі SDLC (Software Development Life Cycle), тобто впроваджувати перевірки ще до написання повного коду чи розгортання додатка. Основні компоненти Shift-Left:

- Threat Modeling (моделювання загроз): аналіз можливих векторів атак, створення карт ризиків ще на етапі проєктування архітектури. Дає змогу запобігти створенню уразливих компонентів.
- SAST (Static Application Security Testing): автоматизоване сканування

коду під час його написання або перед комітом у репозиторій. Виявляє помилки програмування, пов’язані з безпекою (SQL-ін’єкції, небезпечне керування пам’яттю, XSS тощо).

- CI/CD з безпекою: безпечні конвеєри безперервної інтеграції та доставлення включають: автоматизовані SAST/DAST-скани на кожному етапі; перевірки залежностей (SCA – Software Composition Analysis); політики безпечного розгортання; контроль IaC (Infrastructure-as-Code), зокрема перевірку конфігурацій Terraform, Ansible, Kubernetes тощо.

- Безпечні шаблони (secure coding guidelines): використання перевірених бібліотек, патернів, впровадження внутрішніх стандартів написання коду.

DevSecOps – це розширення ідеології DevOps, яке включає автоматизацію безпеки, її відповідальність усіма учасниками команди та постійне вдосконалення засобів захисту. Основні принципи DevSecOps:

- Security-as-Code: політики безпеки (наприклад, перевірки доступів, скрипти обмеження прав, перевірки секретів) реалізуються як код і зберігаються у репозиторії.

- Security Ownership: кожен розробник несе відповідальність за захист власного коду.

- Security Reviews: регулярні перевірки та обговорення архітектури, компонентів і оновлень з фокусом на безпеку.

Переваги: зменшення витрат на виправлення вразливостей; раннє виявлення ризиків; пришвидшення релізів без втрати рівня безпеки; підвищення відповідальності команди за захист продукту.

Таким чином, впровадження Shift-Left і DevSecOps дозволяє зробити безпеку невід’ємною частиною процесу розробки, що критично важливо для

сучасних високонавантажених та відкритих до зовнішнього середовища додатків [28].

Доступ до додатків є одним із найуразливіших місць у ланцюгу безпеки. Компрометація облікового запису, крадіжка токена або погано налаштовані права доступу можуть відкрити зловмиснику шлях до всієї системи. Тому контроль автентифікації, авторизації та управління привілеями (Identity and Access Management, IAM) – критичний компонент стратегії захисту на рівні додатків. Багатофакторна автентифікація (MFA):

- Обов’язкова для всіх користувачів, особливо привілейованих.
- Вимагає щонайменше два фактори: щось, що користувач знає (пароль), має (телефон/токен), або є (біометрія).
- Реалізація через стандарти FIDO2, TOTP (Time-Based One-Time Passwords), push-нотифікації (наприклад, Duo Security, Microsoft Authenticator).
- OAuth 2.0, OpenID Connect, JWT: використання відкритих стандартів аутентифікації, які підтримують гнучке делегування прав та інтеграцію з зовнішніми провайдерами (Google, Microsoft, LDAP); JWT-токени (JSON Web Token) дозволяють безпечно передавати атрибути доступу, мають обмежений термін дії та можуть бути відкликані.
- RBAC/ABAC: RBAC (Role-Based Access Control): користувачі отримують права згідно з роллю (наприклад, «адміністратор», «редактор»); ABAC (Attribute-Based Access Control): рішення про доступ приймається на основі атрибутів (місце, час, пристрій, рівень довіри); впровадження таких систем дає змогу: мінімізувати надлишкові привілеї, реалізувати принцип найменших повноважень (least privilege), централізовано керувати змінами доступу.

Захист сесій: встановлення обмеженого терміну життя токенів; примусове

завершення сесій після неактивності або вручну; обов'язковий re-authentication для чутливих операцій (наприклад, зміна пароля, видалення ресурсу). Переваги: зменшення ризику компрометації акаунтів; повний контроль над правами доступу; відповідність стандартам (наприклад, GDPR, ISO/IEC 27001).

Впровадження сильних методів автентифікації та гнучкого управління правами – основа безпеки не лише самого додатка, а й всієї інформаційної інфраструктури організації.

У сучасних архітектурах, зокрема мікросервісних, API (Application Programming Interfaces) стали головною мішенню для атак, оскільки саме через них відбувається передача даних між компонентами систем. Уразливості API можуть призвести до викрадення даних, несанкціонованого доступу, DoS-атак тощо. Основні стратегії захисту API:

- Використання шифрування: усі API-запити повинні передаватися виключно через HTTPS із TLS версії 1.2 або вище; HSTS (HTTP Strict Transport Security) забезпечує примусове використання HTTPS навіть при спробі доступу через HTTP, що захищає від атак типу Man-in-the-Middle (MITM).

- API-гейтвеї (наприклад, Kong, Apigee, Amazon API Gateway): виконують централізовану автентифікацію та авторизацію (з використанням OAuth 2.0, API keys, JWT); застосовують логування, трасування, швидке блокування шкідливого трафіку; реалізують throttling – обмеження кількості запитів для кожного клієнта, що запобігає надмірному навантаженню чи зловживанню.

- Rate limiting: контролює інтенсивність запитів до API, що особливо ефективно проти атак типу brute-force і DDoS; наприклад, обмеження: не більше 100 запитів на хвилину з одного IP.

- IP-фільтрація, геообмеження, CORS-політики: доступ до API може

бути дозволений лише з довірених IP-адрес або регіонів; політика CORS (Cross-Origin Resource Sharing) – обмежує взаємодію API з ненадійними доменами.

Таким чином, багаторівнева захищеність API – це обов'язкова умова для додатків, які використовують відкриті або внутрішні інтерфейси.

Неправильна або відсутня перевірка вхідних даних є основною причиною найнебезпечніших атак на вебдодатки, таких як SQL-ін'єкції, XSS (Cross-Site Scripting) та CSRF (Cross-Site Request Forgery). Основні заходи:

- Валідація на основі дозволених значень (Whitelist validation): перевірка лише наявності «дозволених» символів або форматів (наприклад, регулярні вирази для e-mail, чисел, дат); заборона небажаних форматів і довільного вводу, який може містити скрипти чи команди БД.

- Sanitization: автоматичне очищення даних від потенційно небезпечного контенту (наприклад, script, SQL-запити, HTML); застосовується до всього вводу, навіть якщо він проходить первинну валідацію.

- Параметризовані запити (prepared statements): забезпечують захист від SQL-ін'єкцій, оскільки розділяють логіку SQL-команди та дані користувача; підтримуються у всіх сучасних мовах програмування та ORM-бібліотеках (наприклад, Python SQLAlchemy, Java JDBC, PHP PDO).

- Content Security Policy (CSP): заголовок безпеки, який обмежує джерела, з яких браузер може завантажувати контент; ефективний проти XSS, дозволяє використовувати лише скрипти з довірених джерел.

- Безпечна обробка помилок: помилки не повинні містити детальної інформації про внутрішню структуру додатку (SQL-запити, назви таблиць, стек викликів); рекомендується створювати узагальнені повідомлення для користувача та логувати деталі лише на сервері.

Ці заходи дають змогу знизити ризик найбільш поширених атак і

гарантувати стабільну роботу вебдодатка.

Крім захисту на рівні коду та мережі, застосунки повинні бути захищені від невідомих або нових загроз – саме для цього використовуються Web Application Firewall (WAF) та IDS/IPS-системи. Web Application Firewall (WAF):

- Аналізує HTTP/HTTPS-трафік і фільтрує небезпечні запити до вебдодатка.
- Забезпечує захист від таких атак: SQLi; XSS; CSRF; Path Traversal; Remote File Inclusion.
- Може швидко оновлювати сигнатури для захисту від zero-day уразливостей – до внесення змін у сам код.
- Приклади: Cloudflare WAF, AWS WAF, ModSecurity, F5, Imperva.

IDS/IPS:

- IDS (Intrusion Detection System): виявляє підозрілу активність і сповіщає адміністратора.
- IPS (Intrusion Prevention System): автоматично блокує спроби вторгнення в режимі реального часу.
- Використовуються на мережевому рівні, а також у поєднанні з WAF для додаткової глибини захисту.

Defense in Depth: поєднання WAF, IPS, міжмережевих екранів, SIEM, MFA та валідації даних створює багаторівневу систему безпеки, в якій кожен рівень підстраховує інший.

Переваги: зменшення навантаження на додаток завдяки фільтрації небажаного трафіку ще до його надходження; підвищення стійкості до складних і невідомих атак; автоматичне реагування без втручання людини.

Логування та моніторинг – ключові компоненти безперервної безпеки, що

дозволяють виявляти інциденти, реагувати на них у реальному часі та проводити аналіз причин. Без повноцінного журналювання дій користувачів, систем та додатків неможливо забезпечити прозорість роботи ПЗ. Централізоване логування:

- Збирання логів автентифікації, помилок, змін конфігурацій, запитів до API та баз даних, доступу до конфіденційних даних.

- Застосування SIEM-систем (Security Information and Event Management), таких як Splunk, ELK (Elastic Stack), IBM QRadar, ArcSight, що дають змогу: централізовано зберігати й аналізувати логи; будувати кореляційні правила для виявлення складних атак; створювати дашборди й автоматичні сповіщення про інциденти.

Поведінковий аналіз:

- UEBA (User and Entity Behavior Analytics): відстежує звичну поведінку користувачів та систем і сигналізує про аномалії (наприклад, вхід у незвичний час, масові завантаження даних).

- Інтеграція з EDR/XDR (Endpoint/Extended Detection & Response) дозволяє: проводити автоматичне блокування при підозрілих діях; ізолювати заражені процеси; реагувати на інциденти в режимі реального часу.

Переваги: швидке виявлення аномалій; відстеження інцидентів у хронологічному порядку; документування подій для аудиту й подальшого розслідування.

Регулярне тестування системи на вразливості дає змогу виявити помилки безпеки до того, як ними скористається зловмисник. Це ключовий елемент для забезпечення надійності додатка. Основні типи тестування:

- SAST (Static Application Security Testing): аналіз вихідного коду без запуску додатка. Ефективний для раннього виявлення помилок безпеки.

- DAST (Dynamic Application Security Testing): тестування додатку під час виконання. Дає змогу виявити помилки в логіці, помилки конфігурації, ін’єкції.

- IAST (Interactive Application Security Testing): поєднання SAST і DAST, інтегрується безпосередньо в застосунок під час його виконання.

- SCA (Software Composition Analysis): аналіз відкритих бібліотек і залежностей на предмет відомих вразливостей.

Додаткові заходи:

- Penetration testing (пенетрувальне тестування): симуляція реальної атаки на систему. Може виконуватись внутрішньою або сторонньою командою.

- Аудити безпеки (external security audits): дають незалежну оцінку поточної ситуації та допомагають відповідати стандартам (ISO 27001, PCI DSS тощо).

- Threat Modeling: створення карти загроз для виявлення вразливих ділянок ще до реалізації.

- Post-Incident Review: аналіз інцидентів, помилок реагування, та оновлення політик безпеки на основі висновків.

Переваги: підвищення надійності ПЗ; зниження ризику потрапити під зовнішні атаки; формування циклу безпеки «план – тест – вдосконалення».

Більшість сучасних додатків використовують зовнішні бібліотеки, фреймворки та інші залежності, які можуть містити вразливості. Крім того, неправильне зберігання ключів, паролів і токенів часто призводить до витоків даних. Тому управління залежностями й секретами має бути системним і автоматизованим. Управління залежностями:

- SCA-сканери (Software Composition Analysis): автоматично сканують

залежності на предмет відомих уразливостей (CVE); приклади: Snyk, Dependabot, WhiteSource, OWASP Dependency-Check, npm-audit; в CI/CD-процесі ці інструменти блокують деплой уразливих версій пакетів.

- Автоматичні оновлення: налаштування ботів для регулярного оновлення бібліотек і модулів.

Управління секретами:

- Сховища секретів (Secrets Management): централізоване зберігання API-ключів, токенів, паролів у зашифрованому вигляду; приклади: HashiCorp Vault, AWS Secrets Manager, Azure Key Vault.

- Правила безпеки: ніколи не зберігати секрети в коді, репозиторіях чи конфігураційних файлах у відкритому вигляді; застосовувати короткоживучі токени, автоматичну ротацію ключів; обмежувати доступ до секретів за принципом мінімальних прав (least privilege).

Переваги: уникнення випадкових витоків критичних даних; автоматизація контролю за залежностями; відповідність вимогам стандартів (наприклад, SOC 2, ISO/IEC 27001).

Впровадження комбінованих стратегій захисту на рівні додатків є критично важливим елементом загальної системи інформаційної безпеки організації. На відміну від вузько спрямованих технічних засобів, комбінований підхід дозволяє забезпечити багаторівневий захист кожного етапу життєвого циклу додатку – від проєктування до експлуатації. У рамках цього підходу:

- інтеграція безпеки на ранніх етапах розробки (Shift-Left, DevSecOps) дозволяє виявляти та усувати вразливості до їх потрапляння у продакшн;

- контроль доступу, авторизації та управління сесіями гарантує надійний захист користувацьких облікових записів;

- безпечна реалізація API, захист від web-атак, впровадження WAF та

IDS/IPS мінімізують ризики зовнішніх атак;

- постійне логування, моніторинг і аналітика підвищують прозорість і дозволяють оперативно реагувати на інциденти;
- регулярне тестування, аудит, контроль залежностей і захищене зберігання секретів знижують технічну заборгованість і сприяють дотриманню стандартів безпеки [29].

На основі проведених експериментальних досліджень та побудованого тестового середовища сформовано практичні рекомендації щодо впровадження засобів захисту додатків у корпоративних інформаційно-комунікаційних системах. На відміну від загальних порад, наведені рекомендації конкретно прив’язані до умов експлуатації, результатів тестування та типових сценаріїв використання у корпоративних мережах. Такий підхід відповідає вимогам викладача щодо «рекомендацій по застосуванню власної системи» та визначає, які саме засоби і в яких умовах доцільно використовувати.

Побудоване тестове середовище, яке включає DMZ-зону, внутрішній сегмент, Active Directory, веб-додаток та інструменти аналізу, може застосовуватися для оцінювання рівня захисту корпоративних додатків перед впровадженням у продуктивне середовище, порівняння ефективності різних інструментів безпеки в однакових умовах, перевірки взаємодії систем моніторингу та реагування, а також для моделювання атак і тестування механізмів захисту. Це середовище є практичним інструментом для DevSecOps-команд, фахівців із інформаційної безпеки та адміністраторів корпоративних мереж.

Результати експериментів дозволили визначити оптимальні інструменти для різних сценаріїв. Для глибокого ручного тестування логіки додатків та авторизованих сценаріїв найбільш ефективним виявився Burp Suite Pro завдяки високій точності та можливості моделювати складні атаки. У середовищах з

CI/CD та автоматичними перевірками кожного релізу рекомендується застосовувати AppScan або комбінацію SonarQube і AppScan для забезпечення інтеграції з DevSecOps і автоматичних звітів. Для бюджетних рішень та базового DAST-сканування підходить безкоштовний OWASP ZAP Proxy, тоді як для періодичного аудиту веб-порталів у продуктивному середовищі ефективним є Acunetix, що забезпечує високу швидкість сканування та виявлення помилок конфігурації.

У сфері моніторингу та реагування найбільш масштабні корпоративні мережі ефективно контролювати за допомогою Splunk Enterprise, що забезпечує глибоку аналітику, кореляцію подій та масштабованість. Середні та малі підприємства можуть використовувати Wazuh, який має мінімальні вимоги до ресурсів та ефективно виявляє спроби brute-force і зміни файлів. Для аналітики користувацьких дій та виявлення аномальної поведінки рекомендовано застосовувати рішення на кшталт LogRhythm або Splunk з UEBA.

Механізми захисту доступу та мережевої взаємодії слід обирати залежно від архітектури підприємства. Так, для компаній із хмарною інфраструктурою, VPN та мобільними працівниками доцільна архітектура Zero Trust. Для критичних систем, таких як Active Directory, ERP та CRM, рекомендовано впроваджувати багатофакторну автентифікацію (MFA), рольову модель доступу (RBAC) та управління привілеями (PAM). Мікросегментація доцільна для середовищ із високими вимогами до ізоляції, а IDS/IPS слід розміщувати на межі сегментів або периметра мережі.

Експериментальне тестування виявило закономірності, які варто враховувати при побудові системи захисту. SQL-ін’єкції були виявлені усіма інструментами, тому фільтри від SQLi слід включати у кожну модель безпеки незалежно від бюджету. CSRF та логічні помилки найбільш ефективно виявляв Burp Suite Pro, тому його доцільно застосовувати при авторизованому тестуванні. Wazuh показав найточніше виявлення спроб brute-force, що робить його

ключовим для захисту AD-серверів та VPN-шлюзів. Splunk забезпечив найкращу кореляцію подій і рекомендований у випадках, коли атаки потрібно аналізувати в контексті всієї організації.

В результаті сформовано узагальнену модель застосування засобів захисту для різних типів організацій. Малі підприємства можуть ефективно використовувати базовий DAST-сканер, систему моніторингу Wazuh, MFA з RBAC та базовий WAF. Середні організації доцільно забезпечувати комбінацією ручного тестування, автоматичного сканування, моніторингу з SIEM і IDS/IPS. Великі компанії та критичну інфраструктуру доцільно захищати комплексно: поєднанням ручного та автоматичного тестування, розширеного моніторингу з UEBA, мікросегментацією, Zero Trust та PAM, інтегрованими в повний DevSecOps-процес.

Практична цінність таких рекомендацій полягає у можливості формувати багаторівневу модель захисту, що враховує ресурси та ризики, оптимально розподіляти інструменти відповідно до потреб і бюджету, використовувати тестове середовище як універсальний стенд для оцінки рішень із інформаційної безпеки, а також забезпечувати відповідність міжнародним стандартам, таким як ISO 27001, NIST та MITRE.

Завдяки поєднанню перелічених стратегій організація здатна створити додатки, стійкі до сучасних кіберзагроз, підвищити довіру користувачів, забезпечити відповідність міжнародним вимогам і знизити загальну поверхню атаки в корпоративному середовищі. Це дає змогу будувати надійні, безпечні та масштабовані програмні рішення, здатні ефективно функціонувати в умовах постійно зростаючих ризиків.

4.3 Оцінка впливу запропонованих методів на продуктивність системи

Мікросегментація – це стратегія, яка передбачає логічне розбиття мережі

на малі ізольовані сегменти з індивідуальними політиками доступу. Вона дозволяє значно зменшити ризик бокового переміщення атакуючого в разі компрометації будь-якого вузла мережі. Однак впровадження мікросегментації, особливо в складних корпоративних мережах, неминуче має певний вплив на продуктивність мережевої інфраструктури. Потенційні впливи:

- Затримка трафіку: кожен мережевий запит повинен проходити через набір політик контролю доступу. У середовищах із великою кількістю сегментів це може спричиняти міліметрові затримки, що особливо критично в реальному часі (наприклад, VoIP, трансляції).

- Збільшення складності обробки даних: складність конфігурації, обчислення маршрутів і фільтрації пакетів зростає експоненційно зі збільшенням кількості сегментів.

Зменшення впливу:

- Впровадження мікросегментації на базі програмно-визначених мереж (SDN), таких як VMware NSX, дозволяє перенести виконання політик безпеки на рівень гіпервізора. Це означає, що контроль доступу відбувається локально, без перенаправлення трафіку, що дозволяє зберегти майже лінійну продуктивність (near-line rate).

- За даними досліджень VMware, затримка після впровадження мікросегментації на основі NSX не перевищує 1–2 мс, а вразливість системи до горизонтального проникнення зменшується до 60–90 %.

Мікросегментація є ефективним інструментом з мінімальним впливом на продуктивність при правильному впровадженні. Вона виправдана в середовищах із високими вимогами до ізоляції трафіку, таких як банківські системи, дата-центри, середовища з чутливими даними. Важливо – уникати надмірної деталізації сегментів і впроваджувати централізовані засоби моніторингу політик

доступу.

Системи веб-екранів (WAF) та системи виявлення і запобігання вторгненням (IDS/IPS) забезпечують захист від найпоширеніших мережевих та веб-загроз. Їхнє впровадження значно покращує безпеку додатків і серверів, але впливає на швидкодію системи, особливо за високого навантаження. Потенційні впливи:

- WAF (Web Application Firewall): WAF виконує глибоку перевірку кожного HTTP(S)-запиту, порівнюючи його з правилами захисту від SQL-ін'єкцій, XSS, CSRF тощо; при складній конфігурації (зокрема з захистом від zero-day загроз або адаптивною логікою) можлива затримка відгуку до 50–150 мс; високе навантаження на WAF-сервер може впливати на загальну швидкодію вебсайту або API.

- IDS/IPS: IDS (системи виявлення): працюють у режимі «прослуховування», не впливаючи на продуктивність безпосередньо; IPS (системи запобігання): активно втручаються в трафік, перевіряючи та блокуючи пакети в реальному часі. Це може спричинити затримки обробки трафіку, особливо у системах із високою пропускною здатністю; за даними з практики використання FortiGate, правильно налаштований IPS викликає падіння пропускної здатності не більше 5–8 %, однак при використанні глибоких сигнатур цей показник може зрости.

Комбінація WAF + IDS/IPS:

- Сценарій «defense in depth» – поєднання декількох рівнів контролю (мережевий + прикладний рівень) – дозволяє максимально захистити системи навіть від складних атак.

- Правильно реалізована інтеграція WAF + IPS забезпечує високий рівень захисту з мінімальним компромісом продуктивності, за умови:

балансування навантаження; розділення трафіку за рівнем ризику; застосування пріоритетних політик лише до чутливих сервісів.

Хоча впровадження WAF і IPS може призвести до затримок у мілісекундах, ці затримки є прийнятними для більшості бізнес-додатків, особливо в обмін на захист від критичних уразливостей. При високих вимогах до продуктивності (фінансові біржі, онлайн-ігри) рекомендується використання апаратних або віртуалізованих рішень з високою пропускною здатністю.

Ефективний захист додатків у корпоративному середовищі не обмежується лише виявленням вразливостей. Він передбачає комплексне, поетапне впровадження логування, аналітики подій, моніторингу поведінки, управління залежностями та інтеграції процесів безпеки у життєвий цикл програмного забезпечення. У цьому контексті особливу роль відіграють системи SIEM, EDR/XDR, а також DevSecOps-практики, які дозволяють виявляти й усувати вразливості на ранніх стадіях розробки.

Системи централізованого логування, зокрема Splunk, Wazuh, ELK Stack (Elasticsearch + Logstash + Kibana), є основою сучасної аналітики подій безпеки. Вони забезпечують збір журналів з усіх вузлів, даючи змогу виявляти аномалії, відстежувати інциденти та відповідати вимогам до аудиту.

Однак централізоване логування створює значне навантаження на ресурси: великі обсяги журналів споживають багато дискового простору; час пошуку і побудови запитів (query latency) зростає зі збільшенням кількості даних; зростають витрати на зберігання, масштабування індексаторів, резервне копіювання тощо [30].

Саме тому слід застосовувати фільтрацію журналів за рівнем важливості (severity-based log filtering) ще на рівні агентів, що дозволить зменшити кількість неінформативних подій і знизити навантаження на систему. Наприклад, фільтрування повідомлень нижче warning або notice.

Інструменти класу EDR (Endpoint Detection and Response) та XDR (Extended

Detection and Response) доповнюють традиційні SIEM-системи. Вони працюють із локальними агентами на кінцевих пристроях, які використовують технології, як-от eBPF (Extended Berkeley Packet Filter), для моніторингу подій ядра з мінімальною затримкою.

Прикладом таких рішень є Falco, CrowdStrike Falcon, Wazuh EDR, Datadog Agent, які:

- дозволяють фіксувати зміни у файловій системі, спроби несанкціонованого виконання коду;
- формують локальні алерти без обов'язкового звернення до хмари чи SIEM;
- працюють з низькими системними накладними, що знижує витрати на інфраструктуру.

Таким чином у середовищах з обмеженими ресурсами поєднувати легкі EDR-агенти з фільтрацією журналів на рівні лог-агрегації (Logstash/Wazuh agent), що дозволить отримати високу точність без перевантаження центральних систем.

У рамках DevSecOps принципу «Shift Left», безпека повинна бути інтегрована на найраніших етапах розробки. Це досягається через використання SAST (Static Application Security Testing), DAST (Dynamic Application Security Testing) та SCA (Software Composition Analysis).

Хоча ці процеси можуть незначно впливати на швидкість збірки CI/CD, дослідження DevOps Research and Assessment (DORA) свідчать, що організації, які регулярно використовують такі практики, знижують витрати на виправлення вразливостей до 50 %, скорочують середній час від коміту до продакшену та рідше стикаються з критичними інцидентами. Реалізувати автоматичне сканування:

- SAST – при кожному push у репозиторій (наприклад, через

SonarQube, AppScan Source);

- DAST – перед релізом, з використанням AppScan, ZAP або Acunetix;
- SCA – регулярно, за допомогою Dependabot, Snyk, OWASP Dependency-Check.

Уразливості часто виникають не в коді розробника, а в сторонніх бібліотеках і залежностях. Тому необхідне автоматизоване виявлення залежностей із відомими вразливостями (CVEs). Такі засоби, як: GitHub Dependabot; npm audit; Snyk; OWASP Dependency-Track – працюють без накладних витрат у runtime, ідеально інтегруються з CI/CD, і можуть зупинити збірку у разі критичних ризиків.

Також рекомендується впровадити систему керування секретами: HashiCorp Vault; AWS Secrets Manager; Azure Key Vault – які забезпечують політики тимчасового доступу, аудит операцій, і повну автоматизацію обробки токенів, паролів, ключів без зберігання їх у коді.

Реалізація комбінованих стратегій захисту – це не лише про додаткові інструменти, а й про баланс між ефективністю, продуктивністю та відповідальністю за дані. Застосування SIEM, EDR/XDR, DevSecOps та SCA-рішень у єдиній моделі дозволяє не лише виявляти атаки, а й передбачати ризики, знижуючи витрати та прискорюючи безпечну доставку програмних продуктів.

Забезпечення високого рівня інформаційної безпеки в корпоративній мережі неминуче супроводжується додатковими навантаженнями на системи – як у плані обчислювальних ресурсів, так і у складності адміністрування. У зв’язку з цим критично важливою є оптимізація впроваджених контролів, збереження балансу між рівнем захисту, продуктивністю і підтримуваністю інфраструктури. Нижче наведено комплексні рекомендації щодо оптимізації роботи захисних засобів, виходячи з експериментальної моделі.

Після впровадження кожного компонента безпеки – будь то SIEM, IDS,

WAF, VPN чи EDR – необхідно проводити порогові та навантажувальні тести у контрольованому середовищі. Це дозволяє: оцінити вплив на затримки при обробці запитів; визначити вузькі місця в обробці логів чи трафіку; виявити надлишкові спрацювання або ресурсоємні процеси. На рівні віртуального середовища проводити A/B-тестування – зі й без активного контролю – з метою вимірювання часу відповіді сервісів, споживання CPU/RAM, навантаження на мережу.

Централізовані системи моніторингу, зокрема SIEM (Splunk, Wazuh, ELK), потребують оптимізації правил:

- логувати тільки події рівня warning або вище, якщо інші не несуть діагностичної цінності;
- застосовувати інтелектуальні фільтри на Logstash або Wazuh agent (наприклад, ігнорування системних подій Windows з EventID < 1000);
- створити окремі класи подій (critical, info, noisy) для кастомного управління збереженням і ротацією.

Таким чином слід використовувати threshold-based rules і event correlation замість постійного логування всіх подій.

Не всі елементи інфраструктури мають однакову цінність або ризик-експозицію. Тому доцільним є розмежування підходів до захисту: критичні сегменти (AD, платіжні системи, сервіси з ПД): повний стек – SIEM, IDS/IPS, EDR, WAF, TLS, MFA; некритичні сервіси (тестові машини, загальні портали): мінімальний набір контролів – лише базовий firewall, аудит доступу, опціональний сканер вразливостей. Тому слід застосовувати класифікацію активів за рівнем критичності (C1–C3) з прив'язкою до ресурсів безпеки, які на них призначені.

Накопичення великих обсягів логів у системах типу SIEM призводить до: зростання часу пошуку запитів; перевищення квоти (у Free-версіях Splunk –

500MB/день); ризиків переповнення дискового простору. Тому потрібно: реалізувати політики ротації журналів (logrotate); архівувати неактивні події до S3/Glacier (cloud-based storage); використовувати компресію (gzip, zstd) для збереження історичних даних.

Деякі технології з самого початку мають нижчий ресурсоспоживчий профіль, що дозволяє оптимізувати продуктивність:

- eBPF-базовані монітори (Falco, Cilium) – перехоплюють системні виклики ядра з мінімальною затримкою;
- апаратні IDS (на базі FPGA або NIC з offloading) – обробляють мережевий трафік поза CPU основної системи;
- cloud-native секрет-сховища (Vault, Secrets Manager) – дозволяють керувати токенами та ключами централізовано, з audit trail, без необхідності ручного зберігання.

Більшість рекомендацій, потребують додаткових ресурсів: як обчислювальних, так і адміністративних. Проте при грамотному проєктуванні та поетапному впровадженні втрати продуктивності є мінімальними та повністю компенсуються підвищеним рівнем стійкості системи до атак. Головні принципи: defense in depth – багаторівневий захист з резервуванням; специфічне тестування – до/після кожного компонента безпеки; класифікація активів – і призначення засобів захисту відповідно до їх критичності.

Рекомендовано періодично переглядати політики, оновлювати системи контролю та підтримувати документацію щодо ротацій логів, агентів, правил firewall/WAF/SIEM – це забезпечить стабільність та гнучкість безпекової моделі в довгостроковій перспективі.

Висновки до 4 розділу

У четвертому розділі було сформовано комплекс практичних рекомендацій щодо побудови багаторівневої системи захисту корпоративних додатків та мережевої інфраструктури. Запропоновані підходи охоплюють стратегічні, архітектурні та технічні рішення, що відповідають сучасним вимогам безпеки й узгоджуються зі стандартами NIST, OWASP та Zero Trust Architecture.

У рамках розділу визначено доцільність переходу від периметрових моделей захисту до адаптивних підходів, заснованих на принципах нульової довіри, мікросегментації, багатофакторної автентифікації та ролей-орієнтованого контролю доступу. Окрему увагу приділено інтеграції безпеки в життєвий цикл розробки додатків (Shift-Left, DevSecOps), а також забезпеченню стійкості додатків до зовнішніх вебзагроз через впровадження багаторівневих механізмів контролю та моніторингу.

У розділі також проаналізовано вплив рекомендованих методів на продуктивність системи та визначено способи оптимізації роботи компонентів безпеки, включно з логуванням, IDS/IPS, WAF, SIEM та системами керування залежностями. Показано, що за умови правильного проєктування та налаштування сучасні засоби безпеки здатні забезпечити високий рівень захисту без суттєвих втрат продуктивності або збільшення операційних витрат.

Таким чином, отримані результати підтверджують, що запропонований комплекс рекомендацій може бути адаптований до умов конкретного підприємства та слугувати фундаментом для побудови масштабованої, стійкої та керованої системи захисту корпоративних додатків і мережевої інфраструктури.

ВИСНОВКИ

У даній кваліфікаційній роботі комплексно досліджено сучасні методи захисту даних на рівні додатків у корпоративних інформаційно-комунікаційних системах, а також розроблено практичні рекомендації щодо їх ефективного впровадження. Мета дослідження – обґрунтувати, протестувати та запропонувати найбільш дієві інструменти та стратегії захисту додатків – була досягнута шляхом виконання всіх поставлених завдань.

У першому розділі проведено глибокий аналіз актуальних загроз для додатків та визначено причини виникнення вразливостей, таких як SQL-ін’єкції, XSS, CSRF, атаки на сесії, компрометація доступів та використання застарілих бібліотек. Розглянуті сучасні методи протидії — валідація даних, шифрування, MFA, RBAC/ABAC, WAF, IDS/IPS, SIEM, Zero Trust, DevSecOps — дали змогу сформувати теоретичну основу для розробки моделі захисту та виявити ключові переваги й недоліки кожної технології.

У другому розділі обґрунтовано вибір інструментів і методів дослідження, сформовано критерії оцінки ефективності систем захисту (MTTD, MTTR, FPR, продуктивність, масштабованість, інтеграція у CI/CD). Розглянуто міжнародні стандарти (NIST CSF, ISO/IEC 27001, MITRE ATT&CK), що забезпечило методологічну цілісність роботи та дозволило сформувати комплексний підхід до оцінювання засобів кіберзахисту.

У третьому розділі проведено практичну експериментальну частину з моделюванням корпоративної інфраструктури та тестуванням реальних інструментів: Burp Suite Pro, AppScan, Acunetix, ZAP Proxy, Splunk, Wazuh, LogRhythm. Отримані результати показали, що найвищу ефективність у виявленні вразливостей та аналізі подій демонструють Burp Suite Pro та Splunk, тоді як ZAP Proxy і Wazuh є доцільними для організацій з обмеженим бюджетом. Практичні експерименти підтвердили, що жоден засіб не забезпечує повного

захисту окремо – найбільшу ефективність має багаторівневий поєднаний підхід.

У четвертому розділі на основі теоретичних і практичних результатів розроблено комплекс рекомендацій щодо впровадження багаторівневого захисту. Особливу увагу приділено поєднанню Zero Trust, DevSecOps, WAF, IDS/IPS, SIEM, MFA, RBAC, моніторингу, автоматизованого тестування (SAST, DAST, SCA), а також управління залежностями та секретами. Наголошено на важливості балансу між безпекою та продуктивністю, правильній конфігурації захисних механізмів і розмежуванні критичних сегментів інфраструктури. Визначено, що найбільш ефективними є інтегровані рішення, адаптовані до розміру, структури та бюджету підприємства.

Узагальнюючи проведене дослідження, можна зробити такі ключові висновки:

1. Захист на рівні додатків є критичним елементом корпоративної безпеки, оскільки більшість сучасних атак спрямовані не на інфраструктуру, а саме на логіку та взаємодію програмних компонентів.

2. Багаторівневий підхід, який поєднує виявлення, запобігання, моніторинг і реагування, забезпечує найвищий рівень захищеності.

3. Zero Trust та DevSecOps є фундаментальними сучасними концепціями, що забезпечують стійкість додатків і контроль доступів на всіх етапах їхнього життєвого циклу.

4. Практичні експерименти підтвердили ефективність застосування комбінованих стратегій захисту, а також необхідність адаптації засобів відповідно до ресурсів підприємства.

5. Важливим є регулярне оновлення, аудит та навчання персоналу, оскільки людський фактор залишається одним із головних джерел ризиків.

6. Запропоновані рекомендації можуть слугувати практичним базисом для впровадження політик безпеки на підприємствах та вдосконалення існуючих систем кіберзахисту.

Таким чином, результати роботи доводять, що системний, методологічно обґрунтований та практично перевірений підхід до захисту на рівні додатків дозволяє суттєво підвищити кіберстійкість корпоративних систем. Запропоновані рішення є придатними до впровадження у реальних умовах та можуть бути використані як основа для подальших досліджень і покращення практик кібербезпеки.

Оформлення результатів цього дослідження здійснювалося згідно з методичними рекомендаціями кафедри [37].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Захист корпоративних мереж від загроз: засоби та методи - Netwave [Електронний ресурс]. – Режим доступу: https://netwave.ua/zahist-korporativnih-merezh-vid-zagroz-zasobi-ta-metodi/.

2. Кіпчук, Ф., & Соколов, В. (2023). Модель розрахунку витрат на баг-баунті програми тестування вразливостей безпеки. Кібербезпека: освіта, наука, техніка, 2(22), 68–83. https://doi.org/10.28925/2663-4023.2023.22.6883 OWASP Top Ten Security Risks [Електронний ресурс]. – OWASP, 2024. – Режим доступу: https://owasp.org/www-project-top-ten/.

3. CWE - Common Weakness Enumeration [Електронний ресурс]. – The MITRE Corporation, 2024. – Режим доступу: https://cwe.mitre.org/.

4. NVD - National Vulnerability Database [Електронний ресурс]. – National Institute of Standards and Technology (NIST), 2024. – Режим доступу: https://nvd.nist.gov.

5. SANS Institute: Application Security Risks [Електронний ресурс]. – SANS Institute, 2024. – Режим доступу: https://www.sans.org/top25-software-errors.

6. Microsoft Security Development Lifecycle (SDL) [Електронний ресурс]. – Microsoft, 2024. – Режим доступу: https://www.microsoft.com/security/blog/security-development-lifecycle/.

7. Валідація вхідних даних у веб-додатках [Електронний ресурс] / OWASP. – 2024. – Режим доступу: https://owasp.org/www-project-input-validation/.

8. Механізми багатофакторної автентифікації [Електронний ресурс] / NIST. – 2024. – Режим доступу: https://csrc.nist.gov/publications/detail/sp/800-63/3/final.

9. TLS та шифрування у веб-додатках [Електронний ресурс] / SSL Labs. – 2024. – Режим доступу: https://www.ssllabs.com/ssltest/.

10. Оновлення та управління вразливостями [Електронний ресурс] / National Vulnerability Database (NVD). – 2024. – Режим доступу:

https://nvd.nist.gov/.

11. Системи виявлення вторгнень (IDS/IPS) [Електронний ресурс] / Cisco. – 2024. – Режим доступу: https://www.cisco.com/c/en/us/products/security/ids-ips/.

12. DevSecOps: Інтеграція безпеки в розробку [Електронний ресурс] / SANS Institute. – 2024. – Режим доступу: https://www.sans.org/cyber-security-courses/devsecops/.

13. Інструменти автоматизованого тестування безпеки [Електронний ресурс] / OWASP. – 2024. – Режим доступу: https://owasp.org/www-project-application-security-verification-standard/.

14. SIEM та XDR для захисту додатків [Електронний ресурс] / Splunk. – 2024. – Режим доступу: https://www.splunk.com/en_us/products/enterprise-security.html.

15. Криптографічні методи захисту інформації [Електронний ресурс] / NIST. – 2024. – Режим доступу: https://csrc.nist.gov/publications/detail/sp/800-57/.

16. Вразливості антивірусного програмного забезпечення [Електронний ресурс] / VirusTotal. – 2024. – Режим доступу: https://www.virustotal.com/gui/home/.

17. Виклики впровадження DevSecOps у сучасних компаніях [Електронний ресурс] / SANS Institute. – 2024. – Режим доступу: https://www.sans.org/cyber-security-courses/devsecops/.

18. Проблеми багатофакторної автентифікації (MFA) [Електронний ресурс] / Microsoft Security. – 2024. – Режим доступу: https://www.microsoft.com/security/blog/.

19. Оновлення програмного забезпечення та управління вразливостями [Електронний ресурс] / NVD. – 2024. – Режим доступу: https://nvd.nist.gov/.

20. Недоліки автоматизованого тестування безпеки додатків [Електронний ресурс] / OWASP. – 2024. – Режим доступу: https://owasp.org/www-

project-application-security-verification-standard/.

21. Системи виявлення вторгнень (IDS/IPS): обмеження та виклики [Електронний ресурс] / Cisco. – 2024. – Режим доступу: https://www.cisco.com/c/en/us/products/security/ids-ips/.

22. Національний інститут стандартів і технологій США. Zero Trust Architecture [Електронний ресурс]. – 2020. – Режим доступу: https://www.nist.gov/publications/zero-trust-architecture.

23. Splunk Inc. Security Solutions [Електронний ресурс]. – 2024. – Режим доступу: https://www.splunk.com/en_us/solutions/solution-areas/security.html.

24. OWASP Foundation. OWASP Dependency-Check [Електронний ресурс]. – 2024. – Режим доступу: https://owasp.org/www-project-dependency-check/.

25. MITRE Corporation. MITRE ATT&CK Framework [Електронний ресурс]. – 2024. – Режим доступу: https://attack.mitre.org/.

26. Чернігівський, І., & Крючкова, Л. (2024). Тестування антивірусних рішень для корпоративного сегменту. Безпека інформації, 30(3), 407–413. https://doi.org/10.18372/2225-5036.30.20362 SolarWinds. Security Event Manager (SEM) [Електронний ресурс]. – 2024. – Режим доступу: https://www.solarwinds.com/security-event-manager.

27. IBM Security. IBM AppScan: Application Security Testing [Електронний ресурс]. – 2024. – Режим доступу: https://www.ibm.com/security/application-security.

28. Acunetix. Web Vulnerability Scanner [Електронний ресурс]. – 2024. – Режим доступу: https://www.acunetix.com/.

29. Чернігівський, І., & Крючкова, Л. (2025). Тестова послідовність виявлення та ізоляції заражених вузлів інфокомунікаційної мережі. Кібербезпека: освіта, наука, техніка, 3(31), 652–662. https://doi.org/10.28925/2663-4023.2025.31.1070

30. Скуратоський, Є., Аносов, А., Стрельніков, В., & Кучерявий, М. (2025). Експерименти та практичні рішення побудови тестового середовища для перевірки рівня безпеки на рівні додатків. Кібербезпека: освіта, наука, техніка, 3(31), 217–226. https://doi.org/10.28925/2663-4023.2025.31.1014

31. Чернігівський, І., & Крючкова, Л. (2025). Тестування нейромережевих моделей для вирішення задачі виявлення заражених ПК на базі цифрових слідів. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 1(29), 800–817. https://doi.org/10.28925/2663-4023.2025.29.941

32. I. Tyshyk, H. Hulak, Testing an Organization's Information System for Unauthorized Access, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 3826, 2024, 17–29.

33. Zhyrova, Tetyana та Kotenko, Nataliia та Bebeshko, Bohdan та Khorolska, Karyna та Shevchenko, Svitlana (2022) Benchmarking between the DQL Index and the Web Application Accessibility Index using Automatic Test Tools Software: A Systematic Literature Review. Univ Access Inf Soc 21, 2022, pp. 295–324. doi: 10.1007/s10209-020-00751-6

34. Спасітєлєва, С., Чичкань, І., Шевченко, С., & Жданова, Ю. (2023). Розробка безпечних контейнерних застосунків з мікросервісною архітектурою. Кібербезпека: освіта, наука, техніка, 1(21), 193–210. https://doi.org/10.28925/2663-4023.2023.21.193210

35. Крючкова Л., Складанний П., Ворохоб М. (2023). Передпроєктні рішення щодо побудови системи авторизації на основі концепції Zero Trust. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 3(19), 226–242. https://doi.org/10.28925/2663-4023.2023.13.226242

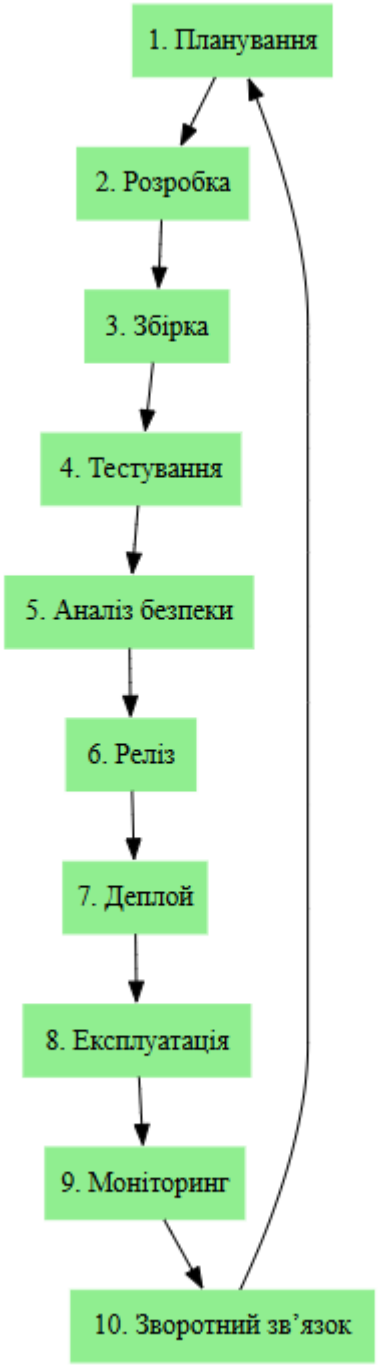
36. Соколов, В., Поліковський, Б., Ворохоб, М., & Цируль, О. (2025). Дослідження ефективності бібліотек санітизації для XSS-атак в веб-додатках. Кібербезпека: освіта, наука, техніка, 3(31), 801–819. https://doi.org/10.28925/2663-4023.2025.31.1076

37. Жданова, Ю. Д., Складанний, П. М., & Шевченко, С. М. (2023). Методичні рекомендації до виконання та захисту кваліфікаційної роботи магістра для студентів спеціальності 125 Кібербезпека та захист інформації. https://elibrary.kubg.edu.ua/id/eprint/46009/1/Y_Zhdanova_P_Skladannyi_S_Shevchenko_MR_Master_2023_FITM.pdf

ДОДАТКИ

Додаток А

DevSecOps-процес

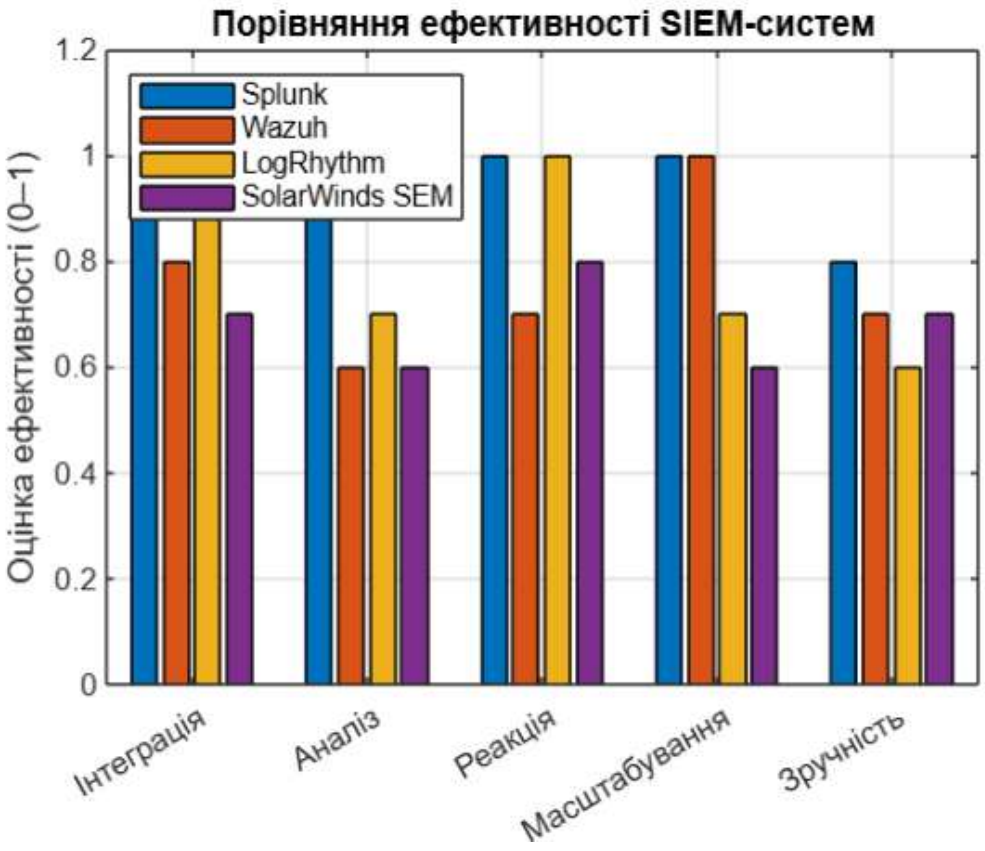


Додаток Б

Zero Trust Architecture

Додаток В

Гістограма ефективності SIEM-систем



Додаток Г

Взаємодія між основними компонентами (DMZ, внутрішня мережа, інструменти атак, моніторинг тощо)