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

КАФЕДРА СИСТЕМ ТА ТЕХНОЛОГІЙ КІБЕРБЕЗПЕКИ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Технологія забезпечення кібербезпеки веб-додатків на базі рішення OWASP»

зі спеціальності *125 Кібербезпека та захист інформації*
(код, найменування спеціальності)

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

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

_____ Денис ЯЛОВИК
(підпис)

Виконав: здобувач вищої освіти групи БСДМ-63
ЯЛОВИК Денис
(прізвище, ім'я)

Керівник д.ф., МАРЧЕНКО Віталій
(науковий ступінь, вчене звання, прізвище, ім'я)

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

Київ 2024

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ 2
ВСТУП 4
1 АНАЛІЗ ПРОБЛЕМИ ЗАБЕЗПЕЧЕННЯ КІБЕРБЕЗПЕКИ 7
1.1. Актуальність проблеми забезпечення інформаційної безпеки 7
1.2. Аналіз проблеми забезпечення захисту інформаційної системи 8
1.3. Мета та завдання забезпечення захисту інформаційної системи 13
1.4. Аналіз існуючих технологій забезпечення захисту інформаційної системи 15
2 МЕТОДИ ЗАБЕЗПЕЧЕННЯ КІБЕРБЕЗПЕКИ ВЕБ-ДОДАТКІВ НА ОСНОВІ OWASP 22
2.1. Призначення, можливості та функції Open Worldwide Application Security Project 22
2.2. Компоненти та архітектура рішення OWASP Top 10 24
2.3. Призначення та архітектура рішення OWASP Security Knowledge Framework (SKF) 28
2.4. Вимоги до системи для інсталяції OWASP Juice Shop 32
2.5. Аудит і тестування безпеки веб-додатків на базі рішень OWASP 39
3 РОЗРОБЛЕННЯ ВАРІАНТА ТЕХНОЛОГІЇ ЗАБЕЗПЕЧЕННЯ ЗАХИСТУ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ БАЗІ РІШЕННЯ OWASP 45
3.1. Розроблення варіанта топології системи забезпечення захисту інформаційної безпеки на базі OWASP 45
3.2. Інтеграція рішень OWASP у життєвий цикл розробки 49
3.3. Розроблення рекомендацій щодо застосування технології забезпечення захисту інформаційної безпеки 57
ВИСНОВКИ 62
ПЕРЕЛІК ПОСИЛАНЬ 64
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація) 66

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

IDS — Intrusion Detection System

RBAC — Role-Based Access Control

IPS — Intrusion Prevention System

VPN — Virtual Private Network

MFA — Multi-Factor Authentication

SAMM — Software Assurance Maturity Model

SQL — Structured Query Language

LDAP — Lightweight Directory Access Protocol

API — Application Programming Interface

WAF — Web Application Firewall

ASVS — Application Security Verification Standard

RBAC — Role-Based Access Control

ABAC — Attribute-Based Access Control

XSS — Cross-Site Scripting

ВСТУП

Актуальність дослідження. Щороку все більше організацій застосовують веб-технології для підвищення своєї ефективності та залучення нових клієнтів. Це стосується як комерційних підприємств, так і державних та місцевих установ. Однак, незважаючи на численні переваги, які надають інтернет-сервіси, вони також збільшують ризики кіберзагроз.

Зростає кількість кіберзагроз, спрямованих на веб-додатки. Веб-додатки є однією з найбільш вразливих частин сучасних інформаційних систем, оскільки вони часто містять чутливу інформацію, що може стати об'єктом атак. Атаки, як-от SQL-ін'єкції, міжсайтові скрипти (XSS), атаки на автентифікацію та інші, можуть призвести до серйозних наслідків, таких як витік даних, фінансові збитки чи порушення функціонування систем. Тому питання захисту веб-додатків стало надзвичайно актуальним.

Web-додатки сьогодні використовуються в багатьох сферах діяльності: від банківської справи та електронної комерції до медичних і освітніх систем. Ці додатки обробляють величезну кількість чутливої інформації, що робить їх привабливими для зловмисників. У зв'язку з цим розробка надійних технологій захисту веб-додатків стає критично важливою.

Велике значення має й те, що OWASP (Open Web Application Security Project) став одним із основних стандартів для забезпечення безпеки веб-додатків. Організація надає корисні інструменти, методики та рекомендації, зокрема відомий список OWASP Top 10, який визначає найбільш поширені уразливості веб-додатків. Використання цих рекомендацій допомагає розробникам запобігати основним загрозам і підвищувати рівень безпеки додатків.

Крім того, технології кібербезпеки постійно розвиваються, оскільки зловмисники знаходять нові методи атак. OWASP надає відкриті інструменти та програмне забезпечення, що дає змогу фахівцям з безпеки швидко адаптуватися до нових викликів і вдосконалювати захист веб-додатків.

Не менш важливими є економічні та репутаційні ризики, пов'язані з

кібератаками. Втрата даних або збої в роботі веб-додатків можуть призвести до значних фінансових збитків і зниження довіри клієнтів. Тому забезпечення кібербезпеки є важливим не лише з точки зору захисту інформації, але й для підтримки репутації організацій.

Враховуючи також постійно зростаючі вимоги до захисту даних, такі як GDPR в Європейському Союзі або CCPA в Каліфорнії, що регулюють обробку персональних даних, компанії зобов'язані дотримуватися високих стандартів кібербезпеки. Це ще більше підкреслює важливість застосування технологій захисту веб-додатків.

Об’єкт дослідження – процес безпечного функціонування веб-додатків

Предмет дослідження – технологія забезпечення кібербезпеки веб-додатків на базі рішення OWASP

Мета роботи – підвищення рівня інформаційної безпеки шляхом впровадження технічних та програмних рішень для посилення безпеки веб-додатків.

Наукові завдання:

- розроблення варіанта топології системи забезпечення захисту інформаційної безпеки на базі рішення OWASP
- проведення аналізу існуючих технологій забезпечення захисту інформаційної системи
- проведення аналізу наукової та технічної літератури за темою кваліфікаційної роботи.
- провести дослідження актуальності проблеми забезпечення інформаційної безпеки
- розробити методи та засоби забезпечення кібербезпеки

Методи дослідження – теорія інформації, міжнародні стандарти у сфері кібербезпеки, політики безпеки.

Практичне значення одержаних результатів полягає в розробці технології підвищення рівня інформаційної безпеки шляхом впровадження технічних та програмних рішень для посилення безпеки веб-додатків.

Апробація результатів. Результати даного дослідження доповідались на Всеукраїнській науковій конференції «Актуальні проблеми кібербезпеки», що проходила у Навчально-науковому інституті кібербезпеки та захисту інформації 25 жовтня 2024 року. Назва тези «ПОБУДОВА OPEN WORLDWIDE APPLICATION SECURITY PROJECT (OWASP)».

1 АНАЛІЗ ПРОБЛЕМИ ЗАБЕЗПЕЧЕННЯ КІБЕРБЕЗПЕКИ

1.1. Актуальність проблеми забезпечення інформаційної безпеки

Актуальність проблеми забезпечення інформаційної безпеки є надзвичайно високою в умовах сучасного цифрового світу, оскільки інформаційні технології, Інтернет та цифрові дані стали основою економічної, соціальної та політичної діяльності.

З розвитком технологій зростає кількість і складність кіберзагроз. Хакерські атаки, шкідливе програмне забезпечення, фішинг, шантаж через викрадення даних та інші форми кіберзлочинності створюють серйозні загрози для конфіденційності, цілісності та доступності інформації.

У сучасному світі більшість компаній, організацій і навіть урядів активно використовують цифрові системи для зберігання і обробки даних. Будь-яка загроза цим системам може призвести до значних фінансових і репутаційних втрат, а також порушення нормальної роботи державних установ і бізнесу.

Збільшення обсягів даних (як особистих, так і корпоративних) потребує більш ефективних методів їх захисту. Дані, що містять персональну, фінансову або іншу чутливу інформацію, є привабливою ціллю для кіберзлочинців.

У глобалізованому світі кіберзагрози не мають кордонів. Держави, бізнеси та окремі громадяни можуть стати жертвами міжнародних кіберзлочинців. Це вимагає міжнародної співпраці для вирішення проблем кібербезпеки.

З розвитком технологій виникають нові виклики для законодавства і нормативних актів у сфері кібербезпеки. Законодавство має постійно адаптуватися до нових загроз, щоб забезпечити ефективний захист прав і свобод громадян.

У випадку державних чи військових кібероперацій, інформаційна безпека набуває стратегічного значення для національної безпеки. Кібернапади можуть вплинути на критичну інфраструктуру, енергетику, фінансові установи та інші важливі сектора держави.

Забезпечення інформаційної безпеки є життєво важливим для стабільного

розвитку сучасного суспільства. Відповідно, держави та організації повинні приділяти особливу увагу розвитку стратегій кіберзахисту, створенню безпечних інфраструктур і вдосконаленню технологій захисту даних.

1.2. Аналіз проблеми забезпечення захисту інформаційної системи

Забезпечення захисту інформаційних систем є важливою і складною проблемою в умовах швидкого розвитку інформаційних технологій та збільшення обсягу і різноманітності загроз. Інформаційні системи стали критично важливими для функціонування майже всіх аспектів сучасного суспільства: від економіки та державного управління до наукових досліджень і особистої діяльності громадян. Разом з тим, зростання залежності від інформаційних технологій призводить до збільшення ризиків, пов'язаних з їх використанням.

Загрози інформаційним системам можуть бути різноманітними: від кібератак і зловмисних програм до несанкціонованого доступу та фізичних пошкоджень обладнання. Крім того, швидкість еволюції кіберзагроз вимагає від організацій постійного вдосконалення заходів безпеки, а також регулярного оновлення програмного забезпечення та апаратних засобів.

Одним з основних аспектів захисту є забезпечення конфіденційності, цілісності та доступності даних, а також управління доступом до них. Це включає впровадження надійних методів шифрування, аутентифікації та моніторингу активностей в мережах і системах. Важливо також враховувати людський фактор — співробітники часто стають мішенями фішингових атак або помилково надають доступ до чутливої інформації.

Наразі особливу увагу приділяють таким напрямам, як розвиток систем виявлення та реагування на вторгнення (IDS/IPS), аналіз загроз за допомогою великих даних та штучного інтелекту, а також впровадження комплексних стратегій кібербезпеки для забезпечення безпеки на всіх рівнях — від кінцевих користувачів до корпоративних мереж і хмарних інфраструктур.

Інформаційні системи стикаються з різноманітними загрозами, що постійно

змінюються та еволюціонують.(рис.1.1) Це можуть бути:

- **Кіберзлочинність**, зокрема хакерські атаки (DDoS-атаки, SQL-ін'єкції, зловмисне програмне забезпечення).
- **Інсайдерські загрози**, коли доступ до чутливих даних мають співробітники або контрагенти, що використовують ці дані з метою крадіжки або шантажу.
- **Фізичні загрози**, такі як крадіжка обладнання або втручання в роботу серверів і пристроїв.

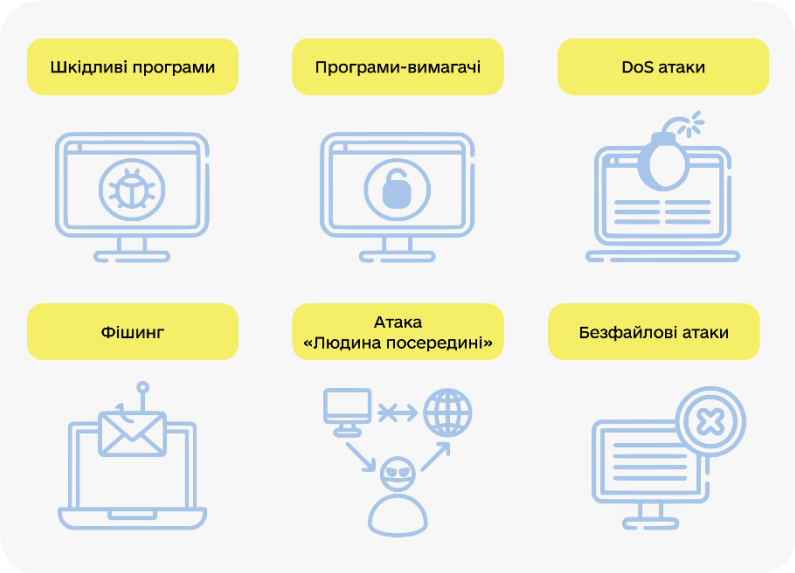

Рис.1.1.Види кіберзагроз

Такі загрози змушують постійно оновлювати методи захисту і використовувати новітні технології для виявлення та нейтралізації загроз.

Для ефективного захисту інформаційної системи потрібно застосовувати комплексний підхід, що включає:

- **Технічний захист** — використання сучасних засобів шифрування, антивірусного програмного забезпечення, фаєрволів, систем виявлення вторгнень (IDS), брандмауерів.
- **Організаційний захист** — розробка політик безпеки, регламентів доступу до даних, контроль за використанням ресурсів.
- **Правовий захист** — дотримання норм законодавства у сфері захисту

інформації, впровадження корпоративних стандартів безпеки.

Однак через швидкий розвиток технологій і постійну еволюцію загроз, важко підтримувати баланс між вартістю забезпечення захисту і необхідними інвестиціями в сучасні рішення. Системи безпеки, які сьогодні здаються надійними, можуть стати вразливими через кілька місяців або навіть тижнів після їх впровадження, оскільки кіберзлочинці постійно розробляють нові методи атак. Тому організаціям необхідно не тільки вкладати значні кошти в захист своїх систем, але й бути готовими до постійного оновлення і адаптації своїх підходів до безпеки.

Однією з основних проблем є те, що інвестиції в кібербезпеку можуть бути великими, а повний захист — фактично недосяжний. Кожна нова загроза вимагає від компаній додаткових витрат на модернізацію обладнання, програмного забезпечення та навчання персоналу. Більше того, обсяг даних, які потрібно захищати, постійно збільшується, що також додає витрат. Це створює складне завдання для організацій, які повинні балансувати між наданням високого рівня захисту і забезпеченням фінансової ефективності.

Один з можливих підходів для досягнення цього балансу полягає в пріоритезації загроз, щоб витрачати ресурси на найбільш критичні аспекти захисту, де потенційний збиток від атаки може бути найбільшим. Однак це потребує глибокого аналізу й оцінки ризиків, що є не лише технологічним, а й стратегічним завданням для керівництва компанії.

Зростаючий обсяг та складність загроз змушують компанії звертатися до новітніх технологій, таких як штучний інтелект, машинне навчання і автоматизація, які дозволяють виявляти й реагувати на загрози в реальному часі. Проте такі технології також вимагають значних інвестицій і можуть створювати додаткові виклики в управлінні і підтримці безпеки.

Вразливості інформаційних систем

Багато інформаційних систем мають вразливості, які можуть бути використані для порушення безпеки:

- **Недосконалість програмного забезпечення**. Помилки у коді,

уразливості, неправильні налаштування можуть стати причиною компрометації системи.

- **Невідповідність стандартам безпеки**. Часто організації не впроваджують найкращі практики в питаннях безпеки, такі як оновлення програмного забезпечення або належне шифрування даних.
- **Невиправлені патчі**. Програмне забезпечення часто має вразливості, для яких випущені патчі, але вони не встановлюються своєчасно.

Управління доступом до інформаційних систем є критично важливим аспектом захисту. Однак, із ростом кількості користувачів, пристроїв та додатків стає складніше гарантувати, що доступ мають лише уповноважені особи. У сучасних умовах, коли організації використовують численні цифрові платформи, мобільні пристрої, а також інтеграцію хмарних технологій, контроль доступу набуває все більшої складності. Багато організацій не можуть зберегти централізовану систему управління доступом, оскільки їх інфраструктура стає все більш динамічною та розподіленою.

Однією з основних проблем є необхідність ідентифікації та автентифікації користувачів, що вимагає використання ефективних і надійних методів для забезпечення доступу до чутливої інформації. Традиційні паролі, хоча й досі широко використовуються, все більше піддаються критиці через низький рівень безпеки, особливо в умовах постійних атак на бази даних і значне збільшення кількості зламаних паролів. Водночас, багатофакторна автентифікація (MFA), яка включає кілька етапів перевірки особи (наприклад, комбінацію пароля, біометрії та одноразового коду), є однією з найефективніших практик для посилення захисту доступу.

Проте, навіть із використанням сучасних методів автентифікації, організації стикаються з проблемами, пов'язаними з управлінням правами доступу. Правильне надання та відновлення доступу, а також своєчасне відкликання прав доступу після завершення співпраці з працівниками або зміни їхніх ролей у компанії, є складними і критичними задачами. Виникають ситуації, коли користувачі отримують доступ до ресурсів, які їм більше не потрібні, що створює додаткові ризики для безпеки.

Іншою складністю є використання різних типів пристроїв (мобільних, робочих, особистих), що підвищує ймовірність несанкціонованого доступу через злом або витік інформації. У таких умовах важливо впроваджувати політики контролю доступу на основі ролей (RBAC), що дозволяють точно визначити, хто має право на доступ до певних даних чи ресурсів залежно від їхньої посади чи функціональних обов’язків.

Крім того, зростання використання хмарних технологій та сторонніх сервісів також ускладнює управління доступом, оскільки ці платформи можуть мати різні системи безпеки та вимоги до автентифікації, що потребує інтеграції й узгодження політик доступу з численними зовнішніми постачальниками.

Актуальність стратегій резервного копіювання

Однією з основних проблем у захисті інформаційних систем є організація ефективного резервного копіювання даних. Втрата критичної інформації може мати катастрофічні наслідки для організації.

- **Резервне копіювання на місці (on-premise)** або у **хмарі** має бути продуманим і регулярним. Однак резервні копії можуть бути заражені шкідливим програмним забезпеченням, якщо їх не ізолювати від основної системи.
- Процес відновлення після катастрофи також є важливим, оскільки в разі втрати даних потрібно швидко відновити систему, що може бути проблематичним, якщо не були належно підготовлені стратегії відновлення.

У зв'язку з посиленням вимог до захисту персональних даних, таких як **GDPR** в Європейському Союзі, виникає необхідність забезпечення відповідності законодавчим вимогам щодо зберігання і обробки даних.

- Невиконання цих вимог може призвести до штрафів і значних репутаційних втрат для організацій.
- Захист персональних даних вимагає реалізації належних заходів безпеки, таких як шифрування, обмеження доступу, аудит доступу.

Виклики у підтримці безпеки в умовах мобільності

З поширенням мобільних пристроїв (смартфонів, планшетів) і віддаленої

роботи, організації стикаються з додатковими труднощами в управлінні безпекою. Мобільні пристрої можуть бути вразливими до зловмисних програм, а віддалений доступ до корпоративних ресурсів через Інтернет створює додаткові точки входу для кіберзагроз.

1.3. Мета та завдання забезпечення захисту інформаційної системи

Метою забезпечення захисту інформаційної системи є **збереження конфіденційності, цілісності та доступності** інформації в рамках цієї системи, а також захист її від внутрішніх і зовнішніх загроз, таких як несанкціонований доступ, кібернапади, витік даних, зловмисне програмне забезпечення, помилки користувачів і технічні неполадки. Основні цілі забезпечення захисту інформаційної системи можна сформулювати наступним чином:

1. **Забезпечення конфіденційності**: Гарантувати, що інформація доступна лише тим особам і організаціям, які мають відповідний дозвіл.
2. **Забезпечення цілісності**: Забезпечити, щоб інформація не була змінена або пошкоджена без належного дозволу, а також підтримувати точність і достовірність даних.
3. **Забезпечення доступності**: Гарантувати, що інформація та ресурси системи доступні для авторизованих користувачів у необхідний час і в потрібному обсязі.
4. **Захист від несанкціонованих впливів**: Унеможливити несанкціонований доступ до даних і ресурсів системи, а також відновлення після порушень безпеки.

Завдання забезпечення захисту інформаційної системи

Для досягнення поставленої мети, необхідно вирішити низку завдань, серед яких можна виділити такі основні:

1. **Аналіз ризиків та загроз**:
   a. Ідентифікація потенційних загроз для системи (кіберзлочинність,

інсайдерські загрози, природні катастрофи, технічні неполадки).

b. Оцінка рівня ризику та ймовірності виникнення загроз.

c. Визначення критичності інформаційних активів і ресурсів для організації.

2. **Розробка політики безпеки**:

a. Визначення чітких правил і політик безпеки для всіх користувачів системи, включаючи політики щодо доступу до даних, використання паролів, управління правами доступу та аудитів.

b. Створення процедур для управління конфіденційними даними, їх зберігання та обробки.

3. **Захист даних**:

a. Впровадження механізмів шифрування для захисту даних, як при їх передачі через мережу, так і при їх зберіганні.

b. Використання механізмів для забезпечення автентифікації і контролю доступу (паролі, біометрія, двофакторна аутентифікація).

c. Захист від втрати даних шляхом регулярного резервного копіювання і розробки плану відновлення після катастрофи.

4. **Моніторинг і виявлення загроз**:

a. Встановлення систем моніторингу і виявлення аномалій, що дозволяють оперативно реагувати на можливі порушення безпеки.

b. Використання систем виявлення вторгнень (IDS) і систем запобігання вторгненням (IPS).

c. Проведення регулярних перевірок та аудитів для виявлення вразливостей і слабких місць в системі.

5. **Аудит та контроль доступу**:

a. Регулярне перевіряння журналів доступу та дій користувачів для виявлення підозрілих або несанкціонованих дій.

b. Впровадження принципу найменшого привілею, що означає обмеження доступу до системи тільки тими правами, які необхідні для виконання конкретних завдань.

6. **Забезпечення фізичної безпеки**:

a. Захист серверів і іншого обладнання від фізичного доступу, в тому числі через встановлення засобів контролю доступу, відеоспостереження, сигналізації.

b. Забезпечення фізичної безпеки даних, зокрема шляхом захисту від крадіжок або пошкоджень обладнання.

7. **Навчання та підвищення обізнаності співробітників**:

a. Проведення навчань та тренінгів для співробітників щодо безпечного поводження з інформацією, використання засобів захисту та дотримання політик безпеки.

b. Підвищення обізнаності про загрози соціальної інженерії, фішингу та інших форм маніпуляцій.

8. **Відновлення після катастроф**:

a. Розробка та тестування планів відновлення інформаційної системи після можливих атак або технічних катастроф.

b. Забезпечення безперервності бізнес-процесів і мінімізація втрат у разі порушення безпеки.

9. **Забезпечення відповідності стандартам і законодавству**:

a. Дотримання вимог національних і міжнародних стандартів безпеки (наприклад, ISO/IEC 27001, GDPR).

b. Забезпечення відповідності законодавчим вимогам щодо обробки та захисту персональних даних.

1.4. Аналіз існуючих технологій забезпечення захисту інформаційної системи

Забезпечення захисту інформаційних систем є ключовим елементом для забезпечення безпеки даних, ефективності роботи і довіри до систем. Існує велика кількість технологій, які дозволяють забезпечити різні аспекти безпеки: захист від кіберзагроз, зберігання конфіденційності та цілісності даних, управління доступом, а також відновлення після катастроф. Далі наведено аналіз основних технологій

захисту інформаційних систем.

Шифрування (Encryption)

Шифрування є одним з основних методів захисту даних, який дозволяє зробити інформацію нерозбірливою для неавторизованих осіб.

- **Типи шифрування**:
  - **Симетричне шифрування** (AES, DES) — використовується один ключ для шифрування і розшифровки даних. Це ефективно і швидко, але ключ повинен залишатися в безпеці.
  - **Асиметричне шифрування** (RSA, ECC) — використовується пара ключів (відкритий і закритий). Це більш безпечно для передачі даних через ненадійні канали, але займає більше ресурсів.
- **Переваги**: Гарантія конфіденційності даних при їх зберіганні або передачі.
- **Недоліки**: Шифрування може уповільнити роботу системи, особливо при великих обсягах даних.

Аутентифікація та управління доступом

Ці технології використовуються для контролю доступу до інформаційної системи.

- **Механізми аутентифікації**:
  - **Паролі** — найбільш традиційний метод, але він може бути вразливим до атак (як фішинг, брутфорс).
  - **Біометрія** (відбитки пальців, розпізнавання обличчя, райдужка ока) — надає додаткову безпеку, але вимагає спеціального обладнання.
  - **Багатофакторна аутентифікація (MFA)** — поєднання кількох методів аутентифікації (наприклад, пароль і одноразовий код на телефоні), що значно підвищує рівень безпеки.
- **Управління доступом**:
  - **Модель доступу на основі ролей (RBAC)** — доступ надається в

залежності від ролі користувача в організації.

- **Модель доступу на основі атрибутів (ABAC)** — доступ залежить від атрибутів користувача, таких як час доби або географічне місце.

- **Переваги**: Захист від несанкціонованого доступу до даних.
- **Недоліки**: Необхідність ускладнити і контролювати облікові дані і процеси аутентифікації для зменшення людського фактора.

Антивірусне програмне забезпечення(рис.1.2)

Рис1.2.Види антивірусних програмних забеспечень

Антивірусні програми дозволяють виявляти, блокувати та видаляти шкідливе програмне забезпечення (віруси, троянці, шпигунські програми).

- **Типи антивірусних технологій**:

- **Підписки на базі сигнатур** — порівняння файлів з базою даних відомих загроз.

- **Методи евристичного аналізу** — виявлення невідомих загроз на основі аномальних патернів.

- **Технології на основі поведінкового аналізу** — аналіз поведінки програм для виявлення шкідливих дій.

- **Переваги**: Захист від відомих і нових загроз.

- **Недоліки**: Програми можуть не завжди виявляти нові або адаптовані загрози, а також уповільнювати систему.

Системи виявлення і запобігання вторгнень (IDS/IPS)

- **IDS (Intrusion Detection System)** — системи для виявлення несанкціонованого доступу або аномальної активності в мережі або на серверах.
- **IPS (Intrusion Prevention System)** — системи, які не тільки виявляють, а й блокують атаки в реальному часі.
- **Типи IDS/IPS**:
  - **Аналіз на основі сигнатур** — пошук відомих шаблонів атак.
  - **Аналіз на основі аномалій** — виявлення аномальної поведінки системи або користувача.
  - **Гібридні системи** — поєднують обидва методи для підвищення ефективності.
- **Переваги**: Забезпечують виявлення та попередження атак на ранніх етапах.
- **Недоліки**: Можливі хибні спрацьовування, необхідність регулярного оновлення бази сигнатур.

Віртуалізація та сегментація мережі

Віртуалізація допомагає створювати ізольовані середовища (віртуальні машини), що можуть бути більш захищеними від атак, ніж фізичні сервери.

- **Сегментація мережі** — розділення внутрішньої мережі на кілька частин для обмеження доступу між ними.

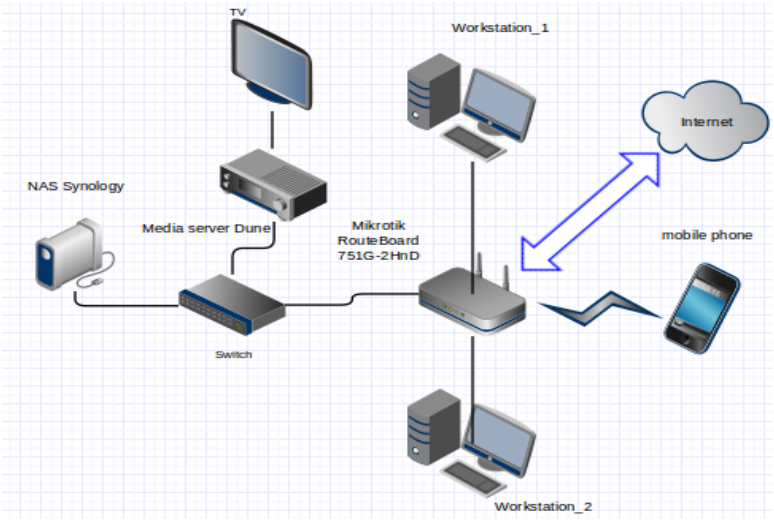

Рис.1.3. Сегментація мережі, використання протоколу 802.1Q

- **Віртуальні приватні мережі (VPN)** — забезпечують зашифроване з'єднання між віддаленими користувачами та корпоративною мережею.
- **Переваги**: Зменшується кількість точок атаки на систему та обмежується можливість поширення загрози.
- **Недоліки**: Необхідність додаткових ресурсів і управління.

Резервне копіювання та відновлення даних

Технології резервного копіювання даних дозволяють відновити систему після атак (наприклад, після шифрування даних у результаті атаки Ransomware).

- **Типи резервних копій**:
  - **Повні резервні копії** — копіюються всі дані.
  - **Інкрементальні резервні копії** — копіюються лише зміни з останньої резервної копії.
  - **Диференційні резервні копії** — копіюються зміни з останньої повної резервної копії.
- **Переваги**: Гарантія відновлення після збоїв або атак.
- **Недоліки**: Високі витрати на зберігання і управління великими обсягами даних.

Фаєрволи (Firewall)

Фаєрволи є основним засобом захисту мережі від несанкціонованого доступу, фільтруючи трафік між внутрішньою та зовнішньою мережею.(рис.1.4)

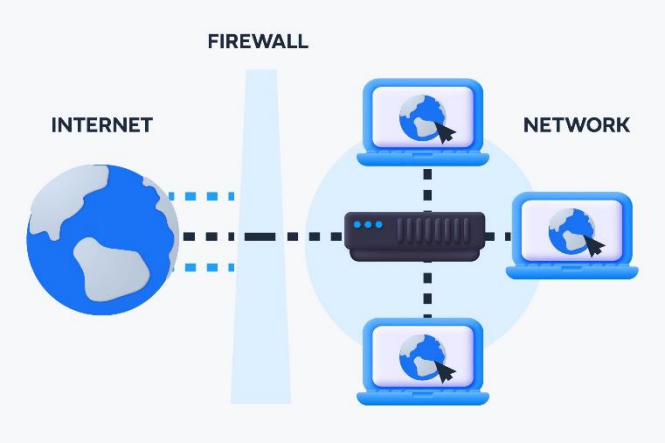

Рис.1.4.Приклад роботи фаєрволів

- **Типи фаєрволів**:
  - **Мережеві фаєрволи** — захищають від атак на рівні мережі (IP, порти).
  - **Програмні фаєрволи** — захищають конкретні пристрої від атак.
- **Переваги**: Захист від зовнішніх атак і несанкціонованого доступу.
- **Недоліки**: Не завжди можуть захистити від внутрішніх загроз або більш складних атак.

Висновки до розділу 1

Забезпечення інформаційної безпеки є життєво важливим для стабільного розвитку сучасного суспільства. Відповідно, держави та організації повинні приділяти особливу увагу розвитку стратегій кіберзахисту, створенню безпечних інфраструктур і вдосконаленню технологій захисту даних.

Забезпечення захисту інформаційної системи є складним і постійно змінюваним завданням. Це вимагає комплексного підходу, використання новітніх технологій і стратегій, а також адаптації до нових загроз і змін у законодавстві. Постійне оновлення систем безпеки, освіта співробітників, належне управління

доступом і резервне копіювання даних є важливими складовими ефективного захисту інформаційних систем. Завдання забезпечення захисту інформаційної системи є багатоаспектним і включають як технічні, так і організаційні елементи. Їх успішне вирішення дозволяє забезпечити надійний захист від кіберзагроз, мінімізувати ризики порушення безпеки та забезпечити безперервність роботи системи.

Існуючі технології захисту інформаційних систем включають різноманітні інструменти та методи, що дозволяють знижувати рівень ризику та підвищувати безпеку. Кожна технологія має свої переваги та обмеження, і для досягнення максимальної ефективності необхідно використовувати комплексний підхід, що поєднує різні методи захисту в межах єдиної системи. Важливою умовою є також постійне оновлення та вдосконалення технологій, оскільки з розвитком загроз з’являються нові методи атаки.

2 МЕТОДИ ЗАБЕЗПЕЧЕННЯ КІБЕРБЕЗПЕКИ ВЕБ-ДОДАТКІВ НА ОСНОВІ OWASP

2.1. Призначення, можливості та функції Open Worldwide Application Security Project

OWASP (Open Worldwide Application Security Project) — це міжнародна організація, що займається розробкою відкритих стандартів і ресурсів для покращення безпеки веб-застосунків. Її основна мета — підвищити обізнаність щодо проблем безпеки програмного забезпечення та надати організаціям інструменти та методології для забезпечення безпеки їхніх веб-застосунків.

Призначення OWASP

OWASP була створена з метою допомогти розробникам, адміністраторам, тестувальникам і підприємствам забезпечити безпеку своїх веб-застосунків. Основне призначення організації полягає в наступному:

- **Покращення безпеки веб-додатків** шляхом розробки стандартів, практик, інструментів і методів, що допомагають усунути вразливості в програмному забезпеченні.
- **Освітні ініціативи**: OWASP активно займається просуванням знань про безпеку додатків і розробку безпечного програмного забезпечення через публікації, тренінги та конференції.
- **Розвиток стандартів**: OWASP створює відкриті стандарти і рекомендації, які дозволяють організаціям і розробникам будувати безпечні додатки з самого початку, замість того, щоб додавати безпеку пізніше.

Можливості OWASP

OWASP пропонує широкий спектр можливостей для покращення безпеки додатків:

1. **Інструменти для тестування безпеки**: OWASP надає безкоштовні інструменти для тестування безпеки веб-застосунків, які дозволяють виявляти вразливості в програмному забезпеченні та застосунках.

a. **OWASP ZAP (Zed Attack Proxy)** — один із найбільш відомих

інструментів для автоматизованого тестування безпеки веб-застосунків.

b. **OWASP Dependency-Check** — інструмент для виявлення вразливостей у сторонніх бібліотеках та компонентах програмного забезпечення.

2. **Ресурси для навчання**: OWASP організовує тренінги, онлайн-курси, конференції, семінари і вебінари, що дозволяють підвищити обізнаність щодо безпеки веб-додатків серед розробників і фахівців із безпеки.

3. **Стандарти і методології**: OWASP активно працює над створенням і популяризацією стандартів безпеки, таких як:

a. **OWASP Top 10** — рейтинг найбільш поширених вразливостей веб-застосунків, який є важливим інструментом для розробників і тестувальників для виявлення та усунення найпоширеніших загроз.

b. **OWASP Software Assurance Maturity Model (SAMM)** — модель оцінки зрілості процесів забезпечення безпеки на різних етапах розробки програмного забезпечення.

4. **Оцінка безпеки**: OWASP пропонує методи і підходи для проведення оцінки безпеки додатків і їх компонентів, що допомагає організаціям виявляти уразливості до того, як вони будуть використані зловмисниками.

5. **Спільнота та співпраця**: OWASP має велику і активну спільноту розробників, фахівців з безпеки та ентузіастів, що працюють разом над створенням відкритих інструментів, стандартів і документації для покращення безпеки програмного забезпечення.

Функції OWASP

OWASP виконує кілька ключових функцій, що допомагають покращити безпеку веб-додатків і програмного забезпечення в цілому:

1. **Розробка інструментів для забезпечення безпеки** OWASP розробляє відкриті інструменти, які дозволяють тестувати веб-застосунки на наявність вразливостей. Вони включають як автоматизовані, так і ручні методи тестування, що дають можливість розробникам перевірити свої додатки на відповідність стандартам безпеки.

2. **Просвітницька діяльність** OWASP активно сприяє поширенню знань

про безпеку програмного забезпечення серед розробників і організацій. Це включає публікацію документів, організацію тренінгів, конференцій і семінарів. Ключовою частиною просвітницької діяльності є **OWASP Top 10**, який є важливим ресурсом для розробників та інженерів безпеки, а також популяризація хороших практик безпеки.

3. **Аудити безпеки та оцінка уразливостей** OWASP надає методології для проведення безпекових аудитів та оцінки уразливостей у додатках, що дозволяє організаціям виявляти можливі загрози та усувати їх до того, як вони стануть реальними проблемами.

4. **Створення стандартів та керівництв** OWASP формує стандарти для розробників і тестувальників, які допомагають забезпечити базову безпеку програмного забезпечення на всіх етапах життєвого циклу. Це включає як стандарти для тестування, так і найкращі практики для розробки безпечних програм.

5. **Спільнота та колективна розробка** OWASP активно підтримує і розвиває спільноту фахівців, що працюють над покращенням безпеки програмного забезпечення. Це дозволяє обмінюватися досвідом, ділитися новими ідеями та технологіями, що сприяє розвитку інновацій у сфері безпеки.

6. **Моделі оцінки зрілості безпеки** Однією з важливих функцій OWASP є розробка моделей оцінки зрілості організацій у сфері безпеки. **OWASP SAMM (Software Assurance Maturity Model)** дозволяє організаціям оцінити рівень зрілості їхніх процесів щодо забезпечення безпеки і створює план для їх подальшого розвитку.

2.2. Компоненти та архітектура рішення OWASP Top 10

OWASP Top 10 — це найбільш відомий і широко використовуваний стандарт для розуміння, аналізу та усунення найбільш поширених вразливостей у веб-застосунках. Цей список, що оновлюється регулярно, допомагає організаціям зосередити увагу на найбільших загрозах для безпеки додатків і забезпечити базовий рівень захисту.(рис.2.1)

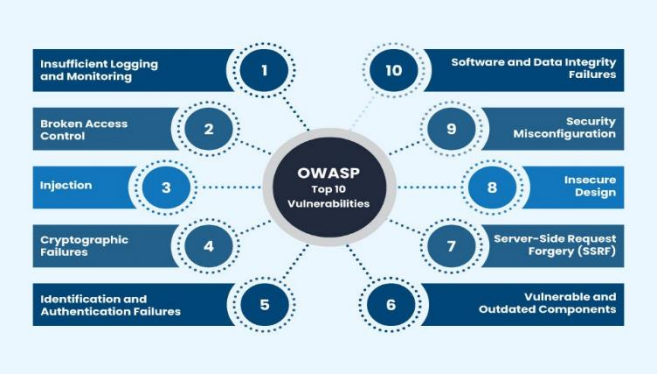

Рис.2.1.Список десяти найбільш критичних вразливостей веб-додатків OWASP

Завдяки тому, що **OWASP Top 10** є наборами найпоширеніших вразливостей, можна визначити стратегії для розробки, тестування і впровадження додатків, що максимально захищені від цих загроз.

Компоненти OWASP Top 10

OWASP Top 10 є списком 10 найбільш поширених загроз для веб-додатків. Зазвичай у кожному циклі випускається оновлена версія цього списку з урахуванням нових типів загроз. Ось компоненти, що складають **OWASP Top 10**:

1. **A01:2021 - Broken Access Control (Порушення контролю доступу)**

a. Вразливості, пов'язані з тим, що користувачі можуть доступити або змінювати дані, які їм не належать. Це може стосуватися як фізичного доступу до ресурсів, так і доступу до веб-застосунку через неправильні налаштування ролей і прав доступу.

2. **A02:2021 - Cryptographic Failures (Криптографічні помилки)**

a. Включає помилки в обробці та зберіганні чутливих даних, таких як паролі, токени або платіжні реквізити. Це стосується ненадійного шифрування або його відсутності, що може призвести до витоків даних.

3. **A03:2021 - Injection (Ін'єкції)**

a. Це клас вразливостей, де зловмисник вставляє шкідливий код в запит (SQL, NoSQL, LDAP ін'єкції), що дає можливість виконати небажані команди на сервері або викрасти чутливі дані.

4. **A04:2021 - Insecure Design (Небезпечний дизайн)**

a. Охоплює вразливості, що виникають через помилки в процесах проектування додатків, таких як відсутність механізмів захисту від атак або невизначеність принципів безпеки на етапі проектування.

5. A05:2021 - Security Misconfiguration (Неправильна конфігурація безпеки)

a. Вразливості, що виникають через неправильні налаштування, які можуть включати залишкові конфігурації за замовчуванням, невикористовувані сервіси або помилки в налаштуванні прав доступу.

6. A06:2021 - Vulnerable and Outdated Components (Уразливі та застарілі компоненти)

a. Використання застарілих або незахищених бібліотек і компонентів програмного забезпечення, що можуть містити відомі вразливості.

7. A07:2021 - Identification and Authentication Failures (Помилки ідентифікації та аутентифікації)

a. Включає проблеми з автентифікацією і керуванням сеансами, наприклад, слабкі паролі, вразливості у механізмах сесій або зломи паролів.

8. A08:2021 - Software and Data Integrity Failures (Помилки цілісності програмного забезпечення та даних)

a. Помилки в обробці даних або програм, які можуть дозволити зловмисникам маніпулювати або коригувати критичні дані або програмні файли.

9. A09:2021 - Security Logging and Monitoring Failures (Помилки в журналах безпеки та моніторингу)

a. Недостатнє ведення журналів і моніторинг безпеки, що ускладнює виявлення або відстеження атак, а також створення механізмів для реагування на інциденти.

10. A10:2021 - Server-Side Request Forgery (SSRF)

a. Вразливість, яка дозволяє зловмисникам змусити сервер додатка зробити запит до іншого сервера в локальній мережі або до інтернет-ресурсів, які в іншому випадку були б недоступними.

Архітектура рішення OWASP Top 10

Рішення OWASP Top 10 не є однією конкретною технологією чи інструментом. Це швидше за все набір рекомендацій, стандартів та найкращих практик для забезпечення безпеки веб-застосунків на всіх етапах їхнього життєвого циклу. Ось як виглядає архітектура рішення, орієнтуючись на найкращі практики, згадані в OWASP Top 10:

1. **Проектування і планування**

a. Під час розробки нових додатків, команда повинна враховувати безпеку на самому початку (захищений дизайн). Усі компоненти системи повинні бути спроектовані з урахуванням можливих загроз і вразливостей.

b. Включення криптографії для захисту чутливих даних і перевірка конфігурацій на предмет правильних налаштувань.

2. **Розробка**

a. Всі компоненти додатка повинні бути перевірені на наявність вразливостей, таких як ін'єкції, неправильні налаштування доступу та проблеми з аутентифікацією.

b. Застосування принципу "мінімального доступу" та використання сильних методів автентифікації, таких як двофакторна автентифікація(2FA).(рис.2.2)

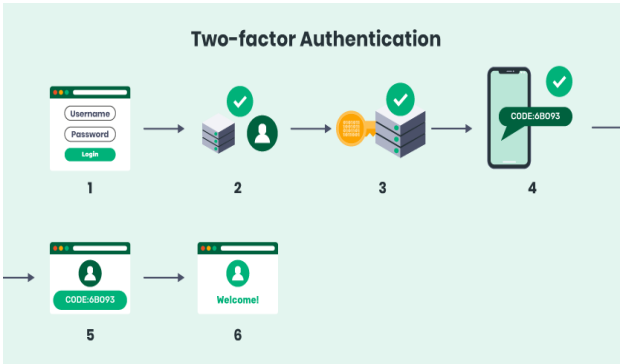

Рис.2.2.Приклад двофакторної аутентифікації

3. **Тестування та аудит**

a. Регулярне проведення тестування на вразливості, таких як SQL ін'єкції, XSS, SSRF та інші. Для цього використовуються автоматизовані інструменти, такі як **OWASP ZAP** або **Burp Suite**, а також методи ручного тестування.

b. Проведення аудиту використаних компонентів і бібліотек для

виявлення застарілих або незахищених частин коду (наприклад, старі версії компонентів з відомими уразливостями).

4. Реалізація моніторингу та логування

a. Забезпечення належного рівня моніторингу і ведення журналів для виявлення спроб атак або несанкціонованих доступів.

b. Використання рішень для збору і аналізу логів, таких як **SIEM** (Security Information and Event Management), для виявлення аномалій і швидкого реагування на інциденти безпеки.

5. Оновлення та обслуговування

a. Регулярне оновлення програмного забезпечення і компонентів, а також впровадження процесу оновлень і патчів для виправлення вразливостей і помилок у роботі додатка.

Ключові стратегії захисту

- **Захищений код**: Використання захищених практик програмування, таких як перевірка вхідних даних, запобігання SQL ін'єкціям і XSS, захист від CSRF атак.
- **Захист на рівні мережі**: Використання фаєрволів, VPN, сегментації мережі і систем для виявлення атак (IDS/IPS).
- **Забезпечення конфіденційності**: Використання сильного шифрування для зберігання та передачі чутливих даних.

2.3. Призначення та архітектура рішення OWASP Security Knowledge Framework (SKF)

OWASP Security Knowledge Framework (SKF) — це відкрита ініціатива від OWASP (Open Worldwide Application Security Project), мета якої полягає в тому, щоб допомогти організаціям та окремим розробникам розуміти, як правильно інтегрувати безпеку в усі етапи розробки програмного забезпечення. (рис.2.3) SKF — це набір знань, інструментів та ресурсів, який покликаний підвищити рівень обізнаності про безпеку серед розробників і тестувальників, допомагаючи їм

приймати обґрунтовані рішення для забезпечення захищеності програм.

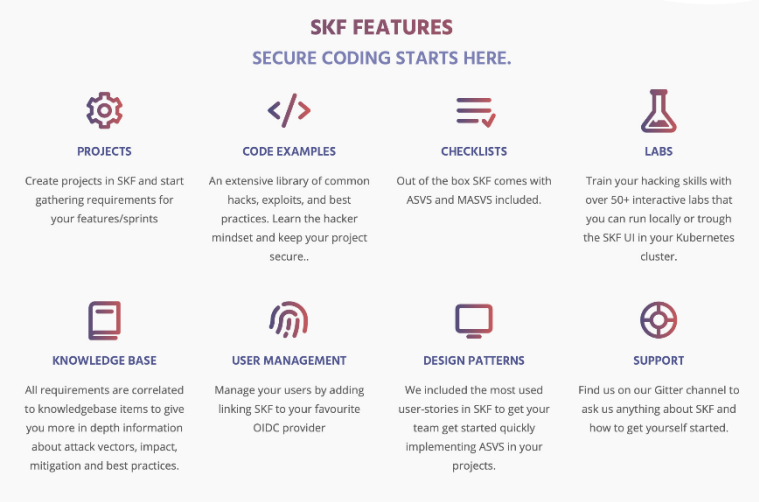

Рис.2.3.Можливості OWASP Security Knowledge Framework (SKF)

Призначення OWASP Security Knowledge Framework

OWASP SKF був створений для того, щоб:

1. **Навчати та підвищувати обізнаність розробників** щодо безпеки додатків і найкращих практик захисту даних.
2. **Підтримувати безпечну розробку програмного забезпечення** на всіх етапах його життєвого циклу, від проектування до розгортання і підтримки.
3. **Забезпечити зручний доступ до знань і інструментів** для розробників і тестувальників, щоб вони могли застосовувати правильні методи безпеки в процесі розробки додатків.
4. **Допомогти в прийнятті рішень по безпеці**: SKF дозволяє командам обирати відповідні заходи безпеки залежно від контексту, з яким вони працюють, і рівня складності рішення.
5. **Забезпечити кращу інтеграцію безпеки** в процеси розробки, що дозволяє зменшити кількість вразливостей у програмному забезпеченні та підвищити його стійкість до атак.

SKF є своєрідним мостом між розробниками і експертами з безпеки, дозволяючи покращити їхню співпрацю для досягнення кращого результату в плані захисту додатків.

Архітектура рішення OWASP Security Knowledge Framework

Архітектура OWASP SKF включає кілька основних компонентів, які разом формують повноцінне середовище для підтримки безпеки програмного забезпечення:

1. **Основні компоненти SKF**:

a. **Рамки знань**: SKF включає базу знань, що охоплює різні аспекти безпеки додатків, такі як захищений код, контролі доступу, криптографія, управління сеансами, тестування на безпеку тощо. Ці знання зібрані у вигляді документів, статей, посібників і прикладів.

b. **Інтерактивні інструменти**: SKF надає різноманітні інтерактивні інструменти, які допомагають користувачам зрозуміти, як застосовувати різні техніки безпеки під час розробки. Це включає інтерактивні інтерфейси для вибору відповідних заходів безпеки для конкретних ситуацій, інструменти для аналізу ризиків та рекомендації по безпеці.

c. **Практичні сценарії та шаблони**: SKF надає шаблони та практичні кейси, які можна використовувати при розробці безпечних додатків. Це допомагає забезпечити узгодженість і стандартизованість підходів до безпеки на всіх етапах життєвого циклу додатка.

d. **Ресурси для навчання**: SKF містить навчальні матеріали, що дозволяють розробникам і тестувальникам освоїти найкращі практики безпеки та дізнатися про новітні тенденції в області кіберзахисту.

e. **Моделі загроз і аналіз ризиків**: SKF допомагає організаціям оцінювати загрози для їхніх систем і вибирати відповідні заходи для зменшення цих загроз через аналітичні моделі та оцінки ризиків.

2. **Основні елементи та принципи роботи SKF**:

a. **Знання, засновані на практиці**: SKF фокусується на реальних проблемах і рішеннях, які розробники можуть застосовувати у своїй роботі. Ось чому він базується на реальних кейсах і використанні конкретних інструментів для тестування безпеки.

b. **Інтерактивні сценарії**: Використання інтерактивних сценаріїв дозволяє не лише пояснити концепції безпеки, а й навчити розробників

застосовувати їх на практиці.

c. **Індивідуалізація безпекових практик**: SKF дозволяє налаштувати підхід до безпеки в залежності від типу застосунку (наприклад, веб-додаток, мобільний додаток, хмарний сервіс) та специфічних вимог компанії або проекту.

d. **Інтеграція з іншими інструментами**: SKF можна інтегрувати з іншими інструментами безпеки, що забезпечує більш повний і системний підхід до тестування безпеки додатків і управління ризиками.

3. **Складові структури SKF**:

a. **Категорії знань**: SKF організовано по категоріях знань, що охоплюють всі аспекти безпеки, такі як управління доступом, автентифікація, криптографія, тестування безпеки, захист від атак, і так далі.

b. **Структура контенту**: Контент SKF організовано так, щоб покрити весь життєвий цикл додатка, починаючи з проектування і закінчуючи тестуванням та підтримкою. Це включає не тільки технічні аспекти, а й організаційні практики щодо безпеки.

c. **Рівні складності**: SKF має ресурси для різних рівнів складності — від початкового рівня для новачків до висококваліфікованих розробників і експертів з безпеки.

Основні компоненти архітектури SKF:

1. **База знань (Knowledge Base)**:

a. Це основний компонент, що містить теоретичну і практичну інформацію щодо забезпечення безпеки. Тут зібрані рекомендації, приклади, шаблони, best practices для безпеки додатків.

2. **Інтерактивні інструменти (Interactive Tools)**:

a. Використовуються для вивчення і застосування знань на практиці. Наприклад, це може бути інтерактивний вибір підходів для безпеки на основі сценаріїв або конкретних вимог користувача.

3. **Моделі загроз (Threat Models)**:

a. Структури для аналізу загроз і виявлення ризиків, специфічних для конкретних додатків або типів бізнесу.

4. **Навчальні ресурси (Training Resources)**:

a. Посібники, курси, відеоуроки та інші матеріали, що дозволяють користувачам поглиблювати свої знання і розвиватися в напрямку забезпечення безпеки.

5. **Шаблони та практичні кейси (Templates & Use Cases)**:

a. Збірники шаблонів і прикладів для вирішення типових завдань безпеки, що допомагають розробникам застосовувати кращі практики для забезпечення захищеності своїх продуктів.

Переваги використання OWASP SKF

1. **Простота інтеграції**: SKF легко інтегрується в процеси розробки і підтримки додатків, що дозволяє швидко отримувати корисну інформацію і приймати рішення по безпеці.

2. **Полегшення навчання**: SKF є чудовим ресурсом для навчання новачків і вдосконалення знань досвідчених розробників у сфері безпеки.

3. **Практичні рекомендації**: SKF пропонує конкретні інструменти, методи і шаблони для впровадження безпеки у розробку.

4. **Адаптивність**: SKF адаптується до конкретних умов організації і проекту, що дозволяє гнучко підходити до застосування безпеки.

2.4. Вимоги до системи для інсталяції OWASP Juice Shop

OWASP Juice Shop — це один із найбільш популярних проектів OWASP, який представляє собою веб-застосунок, створений спеціально для навчання та тестування безпеки. Він симулює веб-застосунок для онлайн-магазину, але при цьому включає ряд уразливостей, які користувачі можуть спробувати експлуатувати в процесі навчання з безпеки.(рис.2.5)

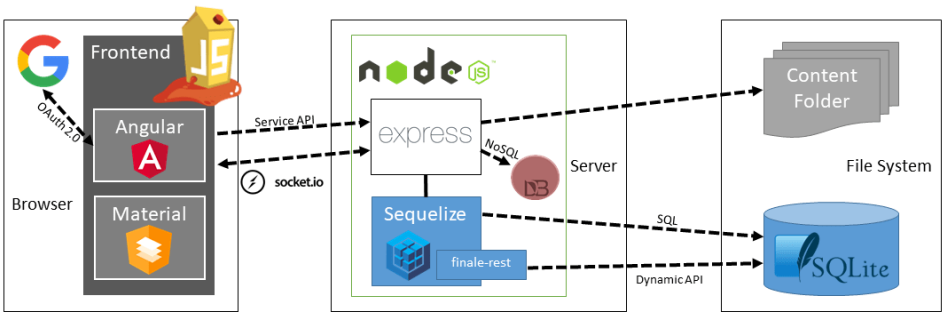

Рис.2.5.Ряд уразливостей в OWASP Juice Shop

Щоб встановити та запустити **OWASP Juice Shop**, необхідно виконати кілька кроків. Нижче наводяться основні **вимоги до системи** та кроки для інсталяції.

1. Операційні системи:

OWASP Juice Shop можна встановити на різних операційних системах, таких як:

- **Windows**
- **Linux**
- **MacOS**

2. Необхідні програмні компоненти:

Для інсталяції **OWASP Juice Shop** потрібно мати встановленими наступні програмні компоненти:

a) Node.js

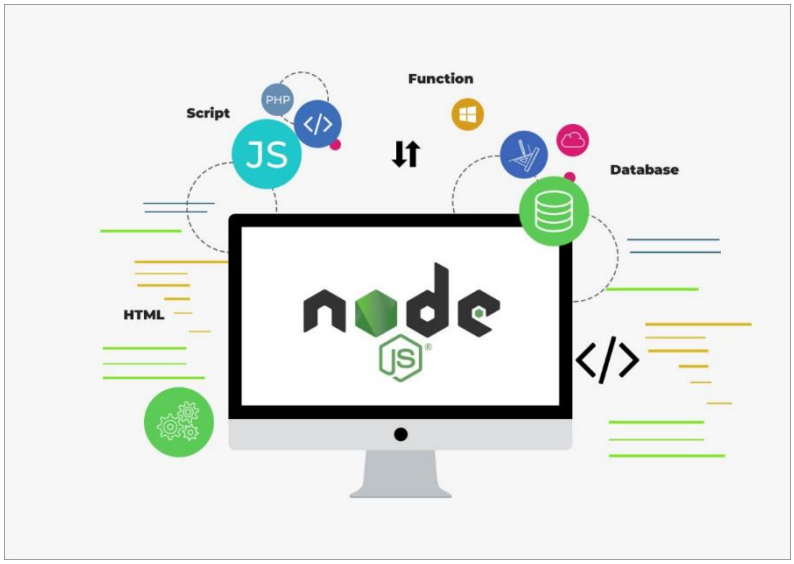

Рис.2.6.Архітектура Node.js

Node.js — це відкритий, крос-платформний середовище виконання, яке дозволяє запускати JavaScript код поза браузером. Він базується на движку JavaScript Chrome V8 і дозволяє писати серверний код, що робить його популярним для створення масштабованих веб-додатків і бекенд-сервісів.

Переваги Node.js:

- **Масштабованість**: Чудово підходить для обробки великої кількості одночасних запитів, що робить його ідеальним для великих додатків.
- **JavaScript всюди**: Розробники можуть використовувати JavaScript як на фронтенді, так і на бекенд, що робить розробку більш узгодженою і ефективною.
- **Підтримка спільноти**: Велика спільнота розробників і обширна документація допомагають швидко вирішувати проблеми.

Недоліки:

- **Обмеження однопоточності**: Хоча Node.js ефективний, для обробки ресурсомістких задач (наприклад, важких обчислень) він може стати менш ефективним, оскільки всі операції відбуваються в одному потоці.
- **Callback Hell**: Управління вкладеними коллбеками може стати проблемою, хоча нові підходи, такі як async/await та Promises, допомагають полегшити це.

OWASP Juice Shop є **Node.js**-застосунком, тому необхідно встановити

Node.js — середовище виконання JavaScript на сервері.

- Мінімальна версія: **Node.js 14.x** або новіша.
- Завантажити Node.js можна з офіційного сайту: https://nodejs.org/

Перевірити, чи Node.js встановлений на вашій системі, можна через команду:

```
node -v
```

b) npm (Node Package Manager)

`npm` є інструментом для управління пакетами в Node.js, і він зазвичай встановлюється разом з Node.js.

Перевірити наявність **npm** можна через команду:

```
npm -v
```

c) Git (для завантаження проекту з репозиторію)

Якщо ви плануєте клонувати репозиторій **OWASP Juice Shop** з GitHub, потрібно мати встановлений Git.

Завантажити та встановити Git можна з офіційного сайту: https://git-scm.com/

Перевірити, чи Git встановлений:

```
git –version
```

d) Браузер

Для доступу до веб-застосунку Juice Shop через локальний сервер, вам потрібен браузер. Сумісні всі сучасні браузери, такі як:

- Google Chrome
- Mozilla Firefox
- Safari
- Microsoft Edge

3. Необхідні ресурси для запуску:

a) Місце на диску

Для локального розгортання **OWASP Juice Shop** потрібно мати кілька мегабайт вільного місця на жорсткому диску для завантаження репозиторію і встановлення залежностей.

b) Інтернет-з'єднання

Для завантаження залежностей через npm або клонування репозиторію з GitHub буде потрібно інтернет-з'єднання.

4. Інсталяція OWASP Juice Shop:

Ось кроки для інсталяції OWASP Juice Shop на вашій машині:

Крок 1: Клонування репозиторію з GitHub

1. Створіть директорію для проекту, якщо це необхідно.(рис.2.6)

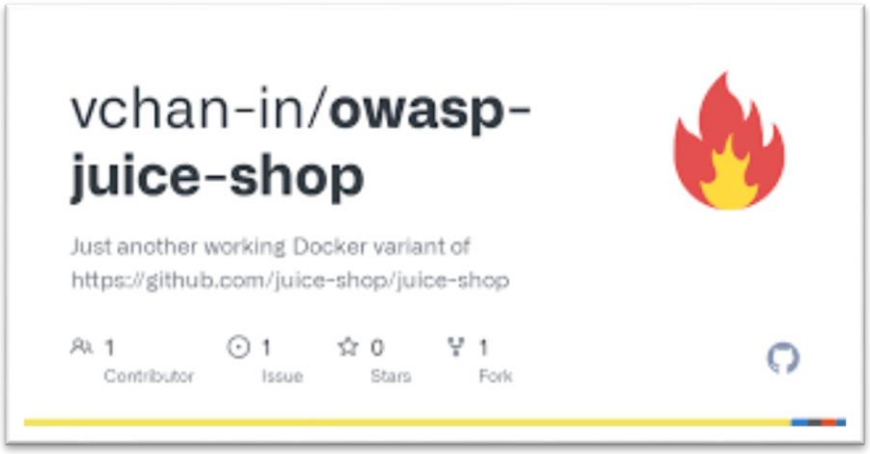

Рис.2.6.Створення директорії в GitHub

2. Переходимо в директорію і клонуємо репозиторій: `git clone https://github.com/owasp/juice-shop.git`

Крок 2: Встановлення залежностей

Переходимо у каталог проекту:

```
cd juice-shop
```

Встановлюємо всі необхідні пакети через npm(рис2.7):

```
npm install
```

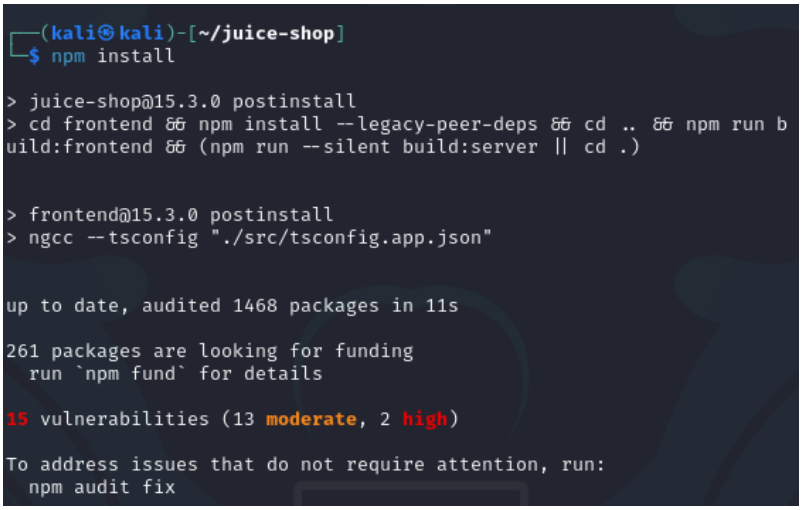

Рис.2.7.Встановлення необхідних компонентів

Ця команда завантажить і встановить всі залежності, які визначені в `package.json` файлі.

Крок 3: Запуск Juice Shop

Після завершення встановлення залежностей, можна запустити проект(рис.2.8):

```
npm start
```

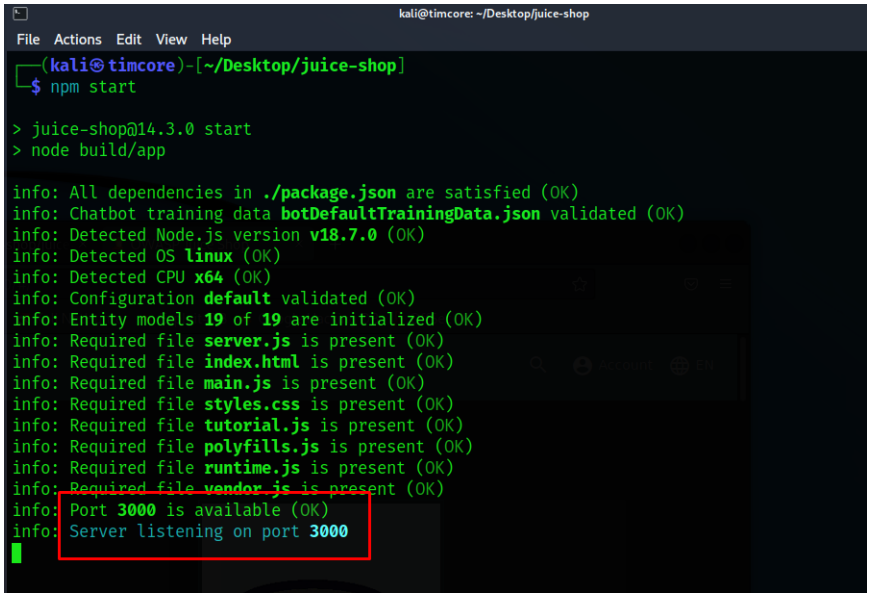

Рис.2.8.Запуск Juice Shop

Ця команда запустить сервер і зробить **OWASP Juice Shop** доступним за замовчуванням на **http://localhost:3000**.

5. Рекомендовані додаткові компоненти для розгортання:

a) Docker (для контейнеризації)

Якщо потрібно запустити **OWASP Juice Shop** у контейнері, можна використовувати **Docker**.

1. Встановлюємо Docker: https://www.docker.com/get-started
2. Завантажуємо Docker образ: `docker pull owasp/juice-shop`
3. Запускаємо контейнер: `docker run -d -p 3000:3000 owasp/juice-shop`(рис.2.9)

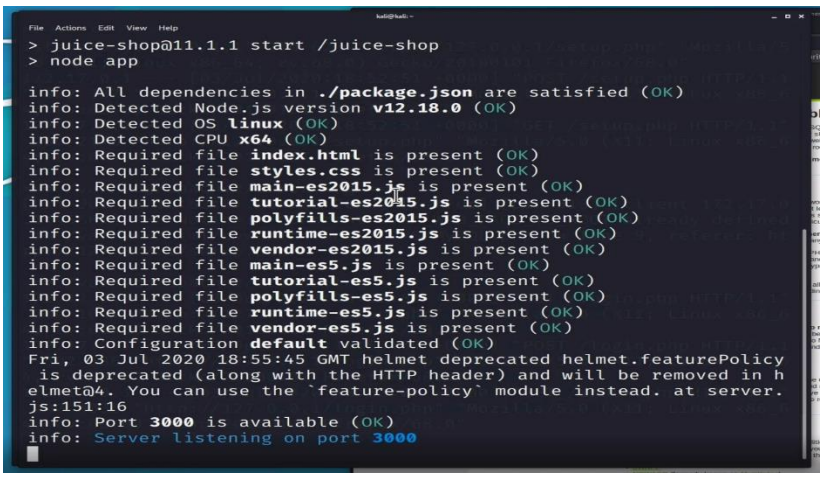

Рис.2.9.Запуск OWASP Juice Shop використовуючи Docker

Це дозволить запускати **OWASP Juice Shop** у контейнері, що забезпечить більш ізольоване середовище для тестування.

b) Docker Compose (для розгортання багатокомпонентних середовищ)

Для запуску **OWASP Juice Shop** в складі багатокомпонентного середовища (наприклад, з базою даних), можна використовувати **Docker Compose**. Для цього потрібно створити відповідний файл `docker-compose.yml`, який дозволить автоматично налаштувати середовище для додатка.

6. Перевірка успішності інсталяції

Після того як успішно запустимо сервер за допомогою команди `npm start` або через Docker, відкриваємо браузер і переходимо за адресою:

- **http://localhost:3000**

Після цього побачимо головну сторінку **OWASP Juice Shop**, яка виглядатиме як типовий інтернет-магазин, що містить ряд уразливостей для навчання.(рис.2.10)

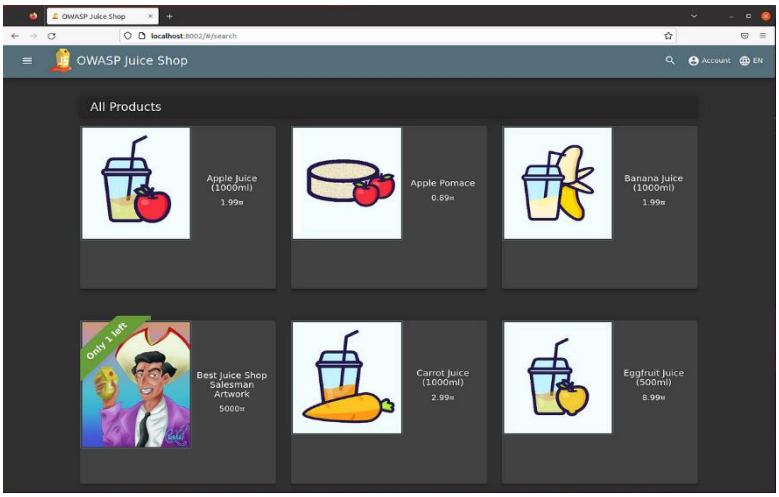

Рис.2.10.Головна сторінка OWASP Juice Shop

2.5. Аудит і тестування безпеки веб-додатків на базі рішень OWASP

Аудит і тестування безпеки веб-додатків є важливою складовою частиною забезпечення їхньої стійкості до атак і вразливостей. Використання рішень і стандартів, розроблених OWASP (Open Worldwide Application Security Project), допомагає систематично оцінювати рівень безпеки додатків, виявляти слабкі місця та забезпечити належний захист від можливих загроз.

OWASP надає різноманітні інструменти, методології та ресурси для тестування безпеки веб-додатків, що дозволяють як професіоналам з безпеки, так і розробникам вбудовувати практики безпеки на всіх етапах розробки та експлуатації.

У цьому контексті, основними інструментами для тестування безпеки є **OWASP ZAP (Zed Attack Proxy)**, **OWASP Dependency-Check**, **OWASP Dependency-Track**, **OWASP Juice Shop** (як платформа для навчання), а також **OWASP Top 10** — набір вразливостей, на які варто звернути увагу при тестуванні.

Ключові методи та інструменти для аудиту та тестування безпеки веб-додатків на основі рішень OWASP

1. OWASP Top 10 (2021)

OWASP Top 10 — це набір найпоширеніших та найкритичніших вразливостей веб-додатків, на які слід звертати увагу при проведенні аудиту безпеки.

- **A01:2021 - Broken Access Control (Порушення контролю доступу)**
- **A02:2021 - Cryptographic Failures (Криптографічні помилки)**
- **A03:2021 - Injection (Ін'єкції)**
- **A04:2021 - Insecure Design (Небезпечний дизайн)**
- **A05:2021 - Security Misconfiguration (Неправильна конфігурація безпеки)**
- **A06:2021 - Vulnerable and Outdated Components (Уразливі та застарілі компоненти)**
- **A07:2021 - Identification and Authentication Failures (Помилки ідентифікації та аутентифікації)**
- **A08:2021 - Software and Data Integrity Failures (Помилки цілісності програмного забезпечення та даних)**
- **A09:2021 - Security Logging and Monitoring Failures (Помилки в журналах безпеки та моніторингу)**
- **A10:2021 - Server-Side Request Forgery (SSRF)**

Цей набір вразливостей є основою для тестування безпеки веб-додатків. Тестувальники повинні ретельно перевіряти наявність цих вразливостей за допомогою різних інструментів і методів, таких як сканери вразливостей, ручне тестування та аудити конфігурацій.

Основні інструменти для тестування безпеки веб-додатків на базі рішень OWASP

1. OWASP ZAP (Zed Attack Proxy)

OWASP ZAP — це потужний, відкритий інструмент для тестування безпеки веб-додатків, який дозволяє автоматизувати багато етапів тестування, а також використовувати його для ручного тестування та дослідження вразливостей. ZAP є

одним з найпопулярніших інструментів для тестування на вразливості.

Основні можливості OWASP ZAP:

- **Автоматичне сканування на вразливості**: ZAP може автоматично сканувати веб-додатки на наявність типових вразливостей (XSS, SQL Injection, CSRF, і т.д.).
- **Ручне тестування**: ZAP має розширені можливості для ручного аналізу і тестування.
- **Перехоплення трафіку**: ZAP дозволяє перехоплювати HTTP/HTTPS запити та відповіді для детальнішого аналізу.
- **Аналіз компонентів**: Він може допомогти з ідентифікацією уразливих сторонніх бібліотек і компонентів, що використовуються в додатку.
- **Інтеграція з CI/CD**: ZAP може бути інтегрований у процеси безперервної інтеграції для автоматизованого тестування на безпеку.

Як використовувати ZAP:

1. Встановлюємо ZAP з офіційного сайту: https://www.zaproxy.org/download/.
2. Запускаємо ZAP і налаштовуємо його для сканування веб-додатка.
3. Використовємо функції для тестування вразливостей, перегляду запитів та відповідей, а також використовуємо автоматичне сканування для аналізу загроз.

2. OWASP Dependency-Check

OWASP Dependency-Check — це інструмент для виявлення уразливих бібліотек і компонентів, що використовуються у вашому веб-додатку. Застарілі або незахищені компоненти є одним з основних ризиків для безпеки веб-додатків, і цей інструмент допомагає ідентифікувати такі компоненти.(рис.2.11)

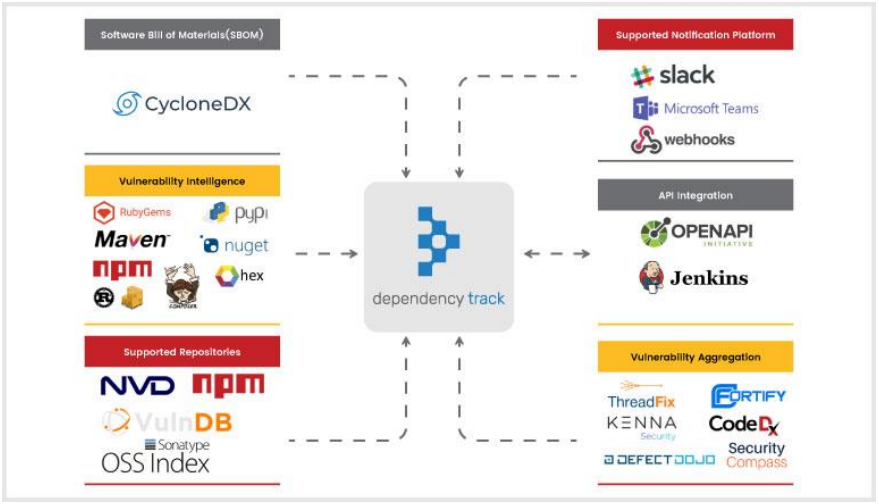

Рис.2.11.Використання OWASP Dependency-Check для покращення життєвого циклу програми

Основні можливості:

- Аналізує проєкти на наявність уразливих компонентів і бібліотек.
- Підтримує різні типи проєктів (Java, .NET, JavaScript тощо).
- Виявляє відомі уразливості у бібліотеках через бази даних CVE (Common Vulnerabilities and Exposures).

Як використовувати:

1. Завантажуємо OWASP Dependency-Check з офіційного сайту.
2. Запускаємо його для сканування проєкту.
3. Отримуємо звіт про виявлені уразливості та оновлюємо компоненти.

3. OWASP Dependency-Track

OWASP Dependency-Track — це інструмент для моніторингу безпеки компонентів програмного забезпечення в режимі реального часу. Він дозволяє відслідковувати та керувати уразливими компонентами, які використовуються в різних додатках, а також підтримує інтеграцію з CI/CD процесами.

Основні можливості:

- Можливість інтеграції з іншими інструментами (наприклад, OWASP Dependency-Check).

- Моніторинг вразливих компонент через вбудовану базу даних CVE.
- Підтримка автоматичних оновлень компонентів.
- Звітність про ризики на основі компонентів.

Тестування на основі OWASP Top 10

Тестування безпеки веб-додатків на основі **OWASP Top 10** включає перевірку вразливостей, пов'язаних з найбільш поширеними проблемами безпеки:

1. **Broken Access Control**: Тестування на порушення доступу до ресурсів (наприклад, перевірка ролей користувачів, доступ до файлів, баз даних).
2. **Cryptographic Failures**: Аналіз використання шифрування для захисту чутливих даних.
3. **Injection**: Перевірка на наявність SQL, NoSQL, LDAP та інших типів ін'єкцій.
4. **Insecure Design**: Аналіз проектних рішень на безпеку, включаючи захист від логічних атак.
5. **Security Misconfiguration**: Перевірка конфігурацій сервера, помилок у налаштуваннях безпеки (наприклад, залишкові файли конфігурацій).
6. **Vulnerable and Outdated Components**: Виявлення застарілих або вразливих сторонніх бібліотек.
7. **Identification and Authentication Failures**: Перевірка на помилки в аутентифікації та управлінні сесіями.
8. **Software and Data Integrity Failures**: Перевірка на цілісність програмного забезпечення та даних.
9. **Security Logging and Monitoring Failures**: Оцінка систем моніторингу і журналювання для виявлення атак.
10. **Server-Side Request Forgery (SSRF)**: Тестування на вразливості, які дозволяють зловмиснику змусити сервер робити непередбачувані запити.

Висновки до розділу 2

OWASP є важливою організацією, що надає безкоштовні ресурси для підвищення рівня безпеки веб-застосунків. Призначення OWASP полягає в підвищенні обізнаності про безпеку програмного забезпечення і наданні інструментів, методологій і стандартів для тестування та покращення безпеки додатків. Її можливості включають інструменти для тестування безпеки, ресурси для навчання, методи оцінки вразливостей і стандарти, що дозволяють побудувати надійні системи безпеки. Функції OWASP охоплюють широкий спектр дій від створення інструментів і стандартів до підтримки глобальної спільноти фахівців з безпеки.

OWASP Security Knowledge Framework (SKF) — це потужний інструмент для підвищення рівня безпеки в процесах розробки додатків. Завдяки інтерактивним інструментам, шаблонам, знанням і рекомендаціям, SKF допомагає розробникам впроваджувати найкращі практики безпеки на всіх етапах життєвого циклу програмного забезпечення. Використання SKF дозволяє зменшити кількість вразливостей у програмному забезпеченні та забезпечити його стійкість до атак.

OWASP Juice Shop — це потужний інструмент для навчання безпеці веб-застосунків. Для його інсталяції на локальному середовищі вам потрібно мати підтримку Node.js, npm, Git і бажано Docker для контейнеризації. Після інсталяції ви отримаєте доступ до вразливого веб-застосунку, який стане чудовою практичною базою для тестування та навчання з безпеки додатків.

Аудит і тестування безпеки веб-додатків є важливою частиною забезпечення захисту

від атак. Використання інструментів OWASP, таких як OWASP ZAP, OWASP Dependency-Check і OWASP Dependency-Track, дозволяє ефективно перевіряти додатки на наявність вразливостей. Комбінація автоматизованих інструментів і ручного тестування, орієнтованого на OWASP Top 10, допомагає забезпечити високий рівень безпеки для веб-додатків і зменшити ризики кіберзагроз.

3 РОЗРОБЛЕННЯ ВАРІАНТА ТЕХНОЛОГІЇ ЗАБЕЗПЕЧЕННЯ ЗАХИСТУ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ БАЗІ РІШЕННЯ OWASP

3.1. Розроблення варіанта топології системи забезпечення захисту інформаційної безпеки на базі OWASP

Забезпечення інформаційної безпеки є критично важливим для будь-якої організації, що розробляє або експлуатує веб-застосунки. Використання рішень OWASP (Open Worldwide Application Security Project) дозволяє створити ефективну систему безпеки, орієнтуючись на найкращі практики і відкриті інструменти. Розробка топології системи захисту інформаційної безпеки на базі OWASP передбачає інтеграцію кількох компонентів, які дозволяють забезпечити багаторівневий захист.

У цій дипломній роботі буде представлений приклад топології системи забезпечення захисту інформаційної безпеки для веб-застосунків на основі рішень OWASP, що включає різноманітні інструменти для оцінки та забезпечення безпеки, а також механізми моніторингу та реагування на інциденти.

1. Основні компоненти системи безпеки на базі OWASP

1. **Веб-додаток** (Web Application)
2. **Захист периметра** (Perimeter Security)
3. **Аудит безпеки та тестування** (Security Audit & Testing)
4. **Моніторинг і реагування на інциденти** (Incident Response & Monitoring)
5. **Управління уразливостями** (Vulnerability Management)
6. **Захист даних** (Data Protection)
7. **Система управління безпекою інформаційної безпеки** (Security Management System)

Ці компоненти повинні бути інтегровані в єдину топологію, що дозволяє забезпечити системний підхід до захисту веб-додатків.

2. Топологія системи захисту інформаційної безпеки на базі OWASPВеб-

додаток і сервери (Web Application & Servers)

Веб-додатки та сервери, на яких вони працюють, є центральними елементами системи, що потребують захисту від атак. OWASP рекомендує забезпечити захист на рівні додатка і на рівні серверної інфраструктури.

- **OWASP ZAP (Zed Attack Proxy)** — інструмент для тестування безпеки веб-додатків, що використовує методи **динамічного аналізу безпеки** для виявлення вразливостей (SQL injection, XSS, CSRF, тощо). Цей інструмент може бути інтегрований в процес CI/CD для автоматизованого тестування безпеки під час розробки.
- **OWASP Dependency-Check** — інструмент для виявлення вразливих сторонніх бібліотек, що використовуються в додатку (наприклад, через CVE).
- **Інструменти захисту серверів**: на серверному рівні повинні бути впроваджені механізми безпеки, такі як **WAF (Web Application Firewall)**, **IDS/IPS (Intrusion Detection/Prevention Systems)** та **антивірусні рішення** для запобігання атак на рівні мережі.

Захист периметра (Perimeter Security)

Захист периметра є основним елементом системи, який дозволяє контролювати доступ до критичних ресурсів.

- **WAF (Web Application Firewall)** — система, яка фільтрує HTTP/HTTPS трафік, що йде до веб-серверів, щоб захистити від атак, таких як SQL injection, XSS, LFI, RCE тощо.
- **Firewall** — міжмережевий екран для фільтрації трафіку і обмеження доступу до сервісів на основі IP-адрес, портів і протоколів.
- **VPN (Virtual Private Network)** — забезпечує захищене підключення до корпоративної мережі для віддалених користувачів, захищаючи конфіденційність і цілісність переданих даних.

Аудит безпеки та тестування (Security Audit & Testing)

Аудит і тестування безпеки повинні бути постійно інтегровані в процес

розробки та експлуатації веб-додатка.

- **OWASP Juice Shop** — відкритий веб-застосунок, що містить вразливості, на основі якого можна навчатися технікам тестування безпеки і знаходити потенційні проблеми в реальних додатках.
- **OWASP ZAP** для **динамічного тестування** (DAST): сканування додатків на наявність вразливостей в реальному часі.
- **OWASP Dependency-Check** і **Dependency-Track** для **статичного аналізу** (SAST): перевірка на використання застарілих або вразливих компонентів.
- **SAST (Static Application Security Testing)** — аналіз вихідного коду додатку на наявність вразливостей.
- **DAST (Dynamic Application Security Testing)** — аналіз роботи додатку в реальному середовищі з фокусом на зовнішні інтерфейси.

Моніторинг і реагування на інциденти (Incident Response & Monitoring)

Це компонент системи, який дозволяє оперативно виявляти та реагувати на кіберінциденти.

- **SIEM (Security Information and Event Management)**: системи моніторингу і збору інформації про безпеку (наприклад, **ELK Stack**, **Splunk**), які допомагають виявляти підозрілі активності та автоматично реагувати на загрози.
- **IDS/IPS**: системи для виявлення (IDS) і запобігання (IPS) атак у реальному часі, що забезпечують безпеку мережі.
- **WAF логування** та **Firewall логування** для відстеження спроб атак на веб-додаток і запобігання небажаному доступу.

Управління уразливостями (Vulnerability Management)

Цей компонент дозволяє ідентифікувати та усувати вразливості на всіх етапах життєвого циклу програмного забезпечення.

- **OWASP Dependency-Check**: виявлення уразливих бібліотек і сторонніх компонентів.
- **Моніторинг CVE-баз даних**: регулярне відслідковування нових

вразливостей у бібліотеках, фреймворках та компонентах.

- **Тестування на основі OWASP Top 10**: регулярне тестування додатків на наявність найпоширеніших уразливостей, що визначені OWASP.

Захист даних (Data Protection)

Захист даних включає в себе використання шифрування та безпечних протоколів для захисту чутливої інформації.

- **Шифрування даних**: застосування протоколів TLS для захисту даних під час передачі та шифрування даних на сервері (AES-256).
- **Обробка конфіденційних даних**: дотримання стандартів безпеки для роботи з конфіденційними даними, таких як **GDPR**, **PCI-DSS**, **HIPAA**.

Система управління безпекою інформаційної безпеки (Security Management System)

Це система, яка забезпечує контроль за впровадженням і підтримкою політик безпеки в організації. Вона повинна включати:

- Політики доступу (RBAC, ABAC).
- Оцінку ризиків.
- Навчання і підвищення обізнаності співробітників.
- Стандарти безпеки на основі OWASP (наприклад, **OWASP ASVS**).

3. Загальний опис топології

1. **Зовнішні елементи**:

a. **WAF** та **Firewall** захищають від зовнішніх загроз.

b. **VPN** для віддаленого доступу.

c. **SIEM-система** для моніторингу подій безпеки.

2. **Основні компоненти системи**:

a. **Web Application** і **Servers** — тестуються за допомогою **OWASP ZAP**.

b. **Моніторинг уразливостей** через **OWASP Dependency-Check** та

OWASP Dependency-Track.

3. **Інтеграція інструментів**:

a. Автоматизоване тестування за допомогою **OWASP ZAP**.

b. Перевірка на вразливості через **OWASP Juice Shop**.

c. Використання **SIEM

** та **IDS/IPS** для моніторингу інцидентів.

- Регулярні **аудити безпеки** на основі **OWASP Top 10**.

3.2. Інтеграція рішень OWASP у життєвий цикл розробки

Інтеграція рішень OWASP (Open Worldwide Application Security Project) у життєвий цикл розробки програмного забезпечення є критично важливим кроком для забезпечення високого рівня безпеки веб-додатків. Принцип "безпека з першого дня" (Security by Design) набуває особливої важливості в умовах постійних кіберзагроз, тому важливо включати тести на безпеку на кожному етапі розробки програмного забезпечення.

Основні етапи життєвого циклу розробки програмного забезпечення (SDLC)

Життєвий цикл розробки програмного забезпечення (SDLC) можна умовно поділити на кілька етапів:

1. **Планування** та **Аналіз вимог**
2. **Проєктування** та **Архітектура**
3. **Розробка**
4. **Тестування**
5. **Розгортання**
6. **Експлуатація** та **підтримка**

Для кожного з цих етапів можна застосувати рішення OWASP, що

допоможуть забезпечити належний рівень безпеки.

1. Планування та Аналіз вимог

На цьому етапі важливо визначити, які вимоги до безпеки повинні бути виконані для конкретного проекту. Це включає в себе:

- **Визначення політик безпеки**: необхідно на початковому етапі визначити основні вимоги до безпеки, такі як конфіденційність, цілісність, доступність даних.
- **Залучення команд з безпеки**: команда розробників повинна активно взаємодіяти з експертами з безпеки для визначення відповідних заходів.

Рішення OWASP на цьому етапі:

- **OWASP Software Assurance Maturity Model (SAMM)**(рис.3.1): надає рамки для оцінки поточної зрілості процесу безпеки програмного забезпечення і допомагає в плануванні інтеграції безпеки в SDLC.

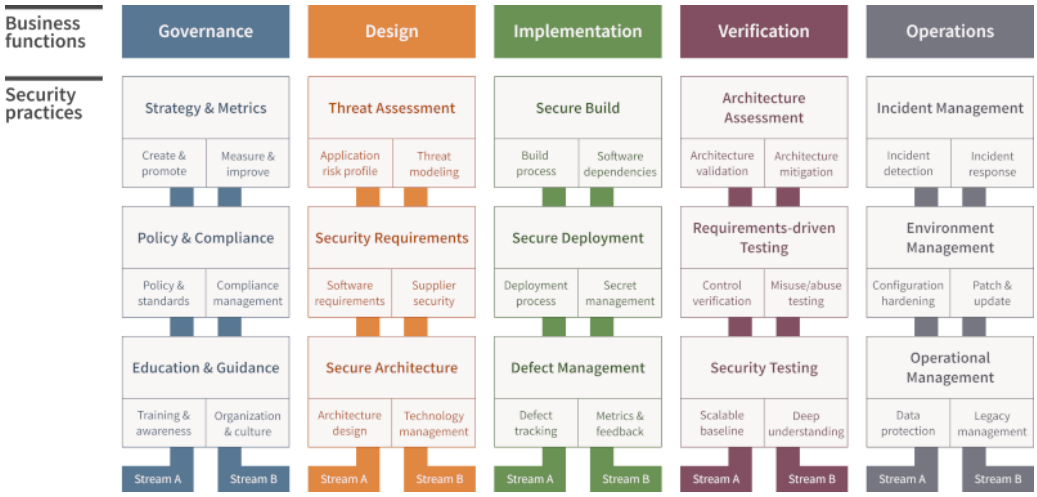

Рис.3.1.Модель зрілості процесу безпеки програмного забезпечення

- **OWASP Risk Rating Methodology**: методологія оцінки ризиків, яка допомагає визначити, які загрози можуть бути актуальними для конкретного

проекту.(рис.3.2)

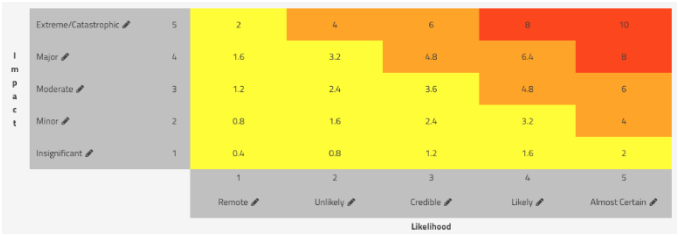

Рис.3.2. Методологія оцінки ризиків

2. Проєктування та Архітектура

На етапі проєктування розробники повинні закласти основи для безпеки системи, передбачивши необхідні механізми захисту на рівні архітектури додатка. Важливо, щоб архітектура не була вразливою до стандартних атак, таких як SQL-ін'єкції, міжсайтові скрипти (XSS), несанкціонований доступ тощо.

Рішення OWASP на цьому етапі:

- **OWASP Top 10**: цей набір найпоширеніших вразливостей веб-додатків допомагає команді розробників і архітекторів зрозуміти, які загрози найчастіше трапляються в реальних веб-додатках, і вчасно вжити необхідних заходів для їх уникнення.
- **OWASP ASVS (Application Security Verification Standard)**: стандарт верифікації безпеки веб-додатків, який пропонує чіткі критерії для перевірки безпеки додатків на етапі архітектури. ASVS забезпечує комплексну перевірку безпеки на різних рівнях: від моделювання загроз до впровадження конкретних

технологій захисту.(рис.3.3)

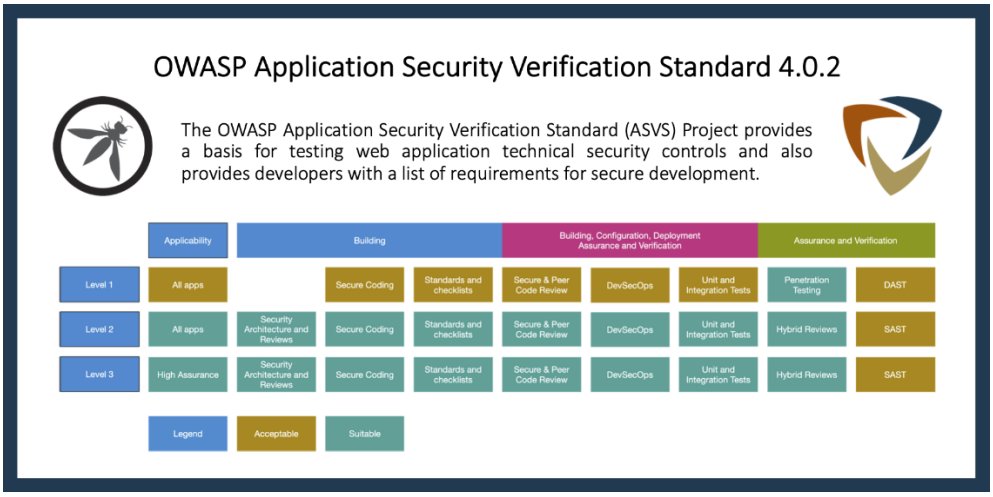

Рис.3.3.Критерії перевірки безпеки веб-додатків OWASP ASVS

3. Розробка

На етапі розробки необхідно інтегрувати практики безпеки безпосередньо в код. Потрібно запобігати поширеним помилкам програмування, що можуть призвести до вразливостей. Це включає в себе використання безпечних бібліотек, валідацію введених даних, а також забезпечення захисту від атак.

Рішення OWASP на цьому етапі:

- **OWASP Dependency-Check**: інструмент для перевірки сторонніх бібліотек на наявність вразливостей, що допомагає виявляти небезпечні компоненти на ранніх етапах розробки.(рис.3.5)

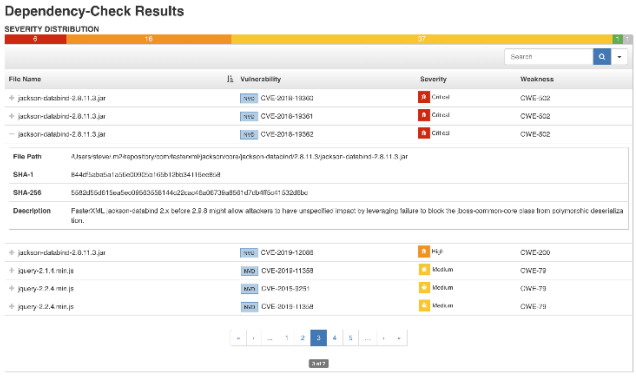

Рис.3.5.Інтерактивна таблиця з результатами бібліотек на наявність вразливості

- **OWASP Dependency-Track**: платформа для управління ризиками, пов'язаними з використанням сторонніх компонентів, що дозволяє моніторити безпеку бібліотек та фреймворків у реальному часі.(рис.3.6)

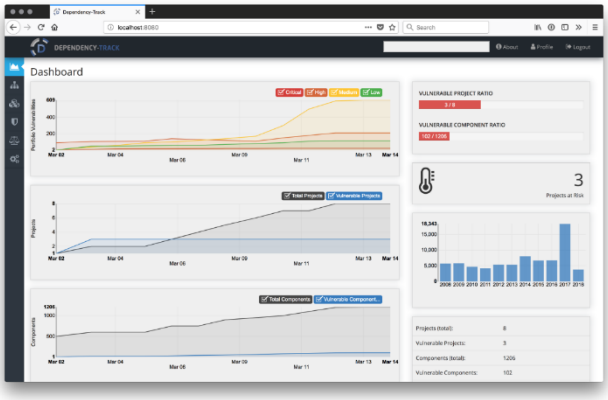

Рис.3.6.Моніторинг бібліотек та фреймворків

- **OWASP Secure Coding Practices**(рис.3.7): набір рекомендацій щодо безпечного програмування, включаючи захист від атак типу SQL Injection, XSS, CSRF, та ін.

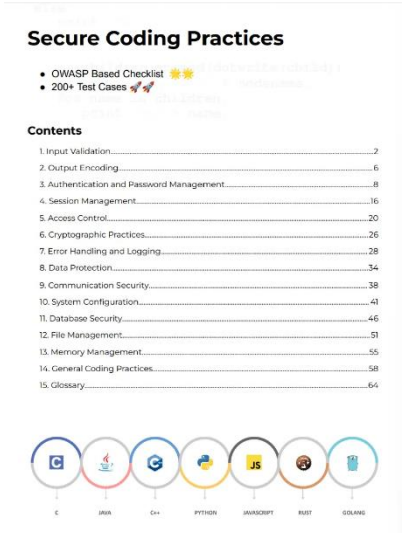

Рис.3.7.Набір рекомендацій OWASP Secure Coding Practices

- **Static Application Security Testing (SAST)**(рис.3.8): на етапі розробки варто використовувати статичний аналіз коду для виявлення уразливостей на ранніх етапах. Інструменти, як **SonarQube**(рис.3.9) або **Checkmarx**(рис3.10), можуть бути інтегровані в процес CI/CD для автоматичного тестування коду на

вразливості.

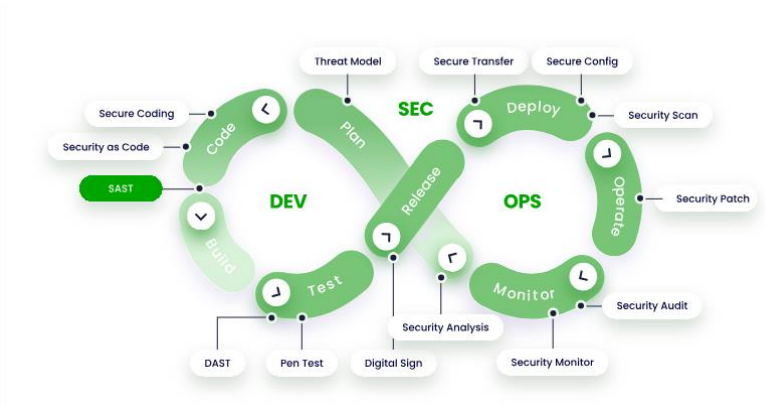

Рис.3.8.Принцип роботи Static Application Security Testing (SAST)

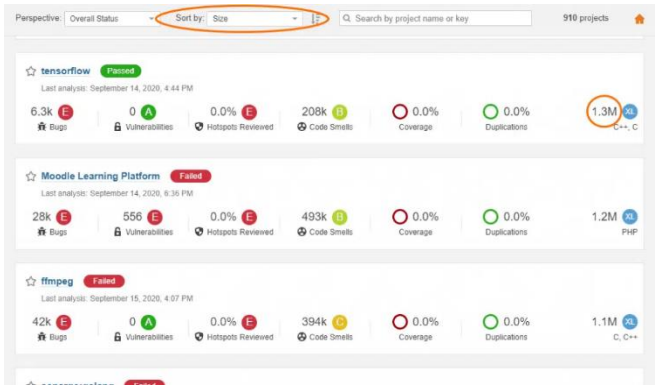

Рис.3.9.Перегляд коду та статичного аналізу за допомогою інструмента SonarQube

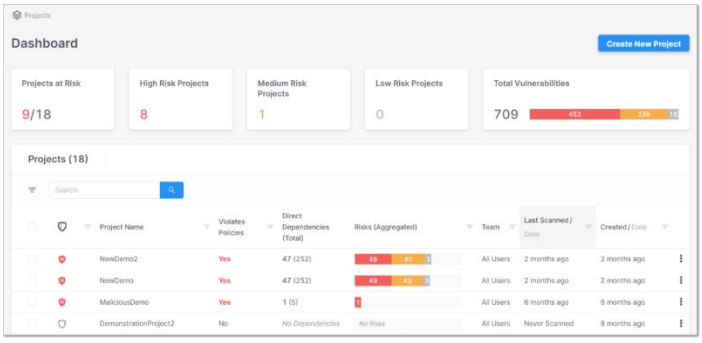

Рис.3.10. Перегляд коду та статичного аналізу за допомогою інструмента Checkmarx

4. Тестування

На етапі тестування важливо провести як статичний, так і динамічний аналіз безпеки, щоб виявити вразливості, які можуть бути непомічені на попередніх етапах. Це включає використання автоматизованих інструментів для тестування безпеки та ручний аудит додатків.

Рішення OWASP на цьому етапі:

- **OWASP ZAP (Zed Attack Proxy)**: інструмент для динамічного тестування веб-додатків. ZAP дозволяє проводити сканування на наявність вразливостей, таких як SQL ін'єкції, XSS, CSRF і багатьох інших.(рис.3.11)

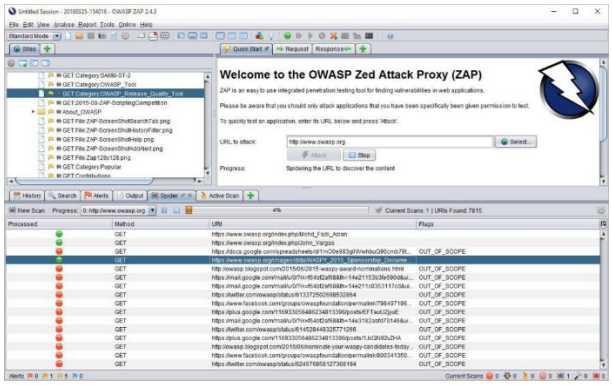

Рис.3.11.Сканування на наявність вразливостей

- **OWASP Juice Shop**: використовувати для навчання та тестування навичок з безпеки веб-додатків. Це також хороший інструмент для імітації реальних атак на веб-додатки.(рис.3.12)

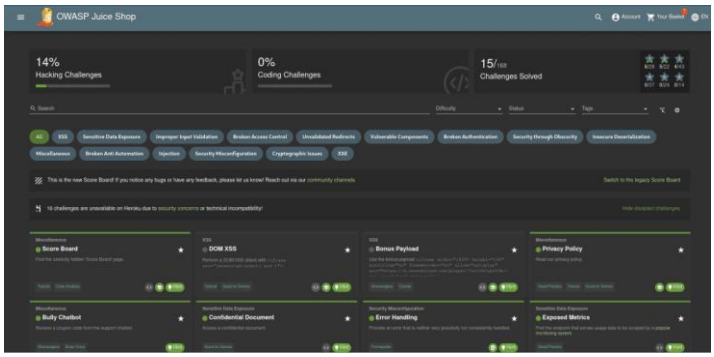

Рис.3.12. Аналіз обмежень OWASP JuiceShop як цільового орієнтиру для бенчмаркінгу інструментів DAST

- **OWASP Security Knowledge Framework (SKF)**: система для навчання та підвищення кваліфікації розробників і тестувальників з безпеки.

5. Розгортання

На етапі розгортання необхідно забезпечити безпеку як інфраструктури, так і додатків, які будуть працювати в реальному середовищі. Важливо правильно налаштувати сервери, мережеву інфраструктуру та бази даних, а також впровадити механізми моніторингу безпеки.

Рішення OWASP на цьому етапі:

- **OWASP Top 10**: на етапі розгортання потрібно перевірити конфігурацію системи відповідно до рекомендацій OWASP Top 10, щоб уникнути таких вразливостей, як неправильні налаштування безпеки або недостатнє шифрування.
- **Web Application Firewall (WAF)**: для захисту веб-додатків від атак, таких як SQL Injection, Cross-Site Scripting, і інших загроз, можна розгорнути WAF.

6. Експлуатація та підтримка

Після того як веб-додаток розгорнутий і працює в реальному середовищі, потрібно забезпечити моніторинг та регулярне оновлення системи, щоб підтримувати високий рівень безпеки. Важливо оперативно реагувати на виявлені вразливості і оновлювати програмне забезпечення.

Рішення OWASP на цьому етапі:

- **OWASP Dependency-Check**: використання для регулярного моніторингу вразливостей у компонентах програмного забезпечення, що використовуються в додатку.
- **Моніторинг та аудит безпеки**: регулярний моніторинг веб-додатків на наявність нових загроз та атак за допомогою **SIEM**-систем і **IDS/IPS**.
- **Incident Response**: інтеграція системи для реагування на інциденти

безпеки, щоб мати змогу швидко усувати наслідки атак.

3.3. Розроблення рекомендацій щодо застосування технології забезпечення захисту інформаційної безпеки

Забезпечення інформаційної безпеки є важливою складовою в процесі розробки та експлуатації програмних систем. Оскільки ризики кіберзагроз постійно зростають, важливо застосовувати комплексні заходи для захисту інформації та систем. Розробка рекомендацій щодо використання технологій забезпечення інформаційної безпеки допоможе знизити ймовірність атак і покращити загальну стійкість систем.

Ці рекомендації можуть бути поділені на кілька ключових категорій, що охоплюють різні аспекти безпеки: від захисту даних до безпеки мережі та додатків.

1. Захист периметра та мережі

1.1 Використання міжмережевих екранів (Firewall)

- **Рекомендація**: Встановити та налаштувати міжмережеві екрани (Firewall) для фільтрації трафіку, що проходить через мережу. Це дозволить блокувати небажаний трафік та мінімізувати потенційні вектори атак.
- **Типи міжмережевих екранів**:
  - **Network-based Firewalls**: для фільтрації трафіку на мережевому рівні.
  - **Host-based Firewalls**: для контролю за трафіком між пристроєм та мережею.

1.2 Захист від атак на рівні додатків (WAF)

- **Рекомендація**: Встановити **Web Application Firewall (WAF)** для захисту веб-додатків від атак, таких як SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), та інші типові веб-загрози.
- **Інструменти**:
  - **ModSecurity**, **Imperva WAF**, **Cloudflare WAF** — популярні

інструменти для захисту веб-додатків.

1.3 Використання VPN для захищених з'єднань

- **Рекомендація**: Використовувати **VPN (Virtual Private Network)** для забезпечення безпечного з'єднання між віддаленими користувачами та корпоративною мережею.
- **Рішення**: Рішення на основі **OpenVPN** або **WireGuard** можуть забезпечити шифроване з'єднання, що мінімізує ризик перехоплення даних.

2. Захист даних та їх шифрування

2.1 Шифрування даних на всіх етапах

- **Рекомендація**: Усі чутливі дані повинні бути зашифровані як під час зберігання, так і під час передачі.
  - **Під час передачі**: використовувати **TLS/SSL** для забезпечення захищених з'єднань між клієнтом і сервером.
  - **Під час зберігання**: використовувати **AES-256** для шифрування баз даних і файлів на серверах.
- **Інструменти**:
  - **Let's Encrypt** для безкоштовного SSL сертифіката.
  - **OpenSSL** для шифрування та генерації сертифікатів.

2.2 Захист паролів

- **Рекомендація**: Для зберігання паролів використовувати безпечні хеш-функції з додатковими солями (наприклад, **PBKDF2**, **bcrypt**, **Argon2**).
- **Інструменти**: **Password Hashing API** для валідації паролів під час реєстрації та автентифікації користувачів.

2.3 Використання політик для управління ключами

- **Рекомендація**: Встановити чітку політику щодо управління криптографічними ключами, включаючи їх генерацію, зберігання, оновлення та

відключення.

- **Інструменти**:
  - **AWS KMS (Key Management Service)** або **Azure Key Vault** для безпечного зберігання та управління ключами.

3. Захист веб-додатків

3.1 Впровадження принципів безпечного кодування

- **Рекомендація**: Дотримуватися принципів **Secure Coding Practices** для запобігання загрозам типу SQL Injection, XSS, CSRF, Command Injection, та іншим.
- **Рішення**: Використовувати вбудовані механізми безпеки фреймворків (наприклад, **OWASP ESAPI**, **OWASP Secure Coding Practices**).
- **Інструменти**:
  - **OWASP ZAP** для динамічного аналізу безпеки (DAST).
  - **OWASP Dependency-Check** для виявлення вразливостей у бібліотеках.

3.2 Регулярні аудит і тестування безпеки

- **Рекомендація**: Регулярно проводити тестування на вразливості і аудит безпеки додатків, застосовуючи статичний (SAST) і динамічний (DAST) аналіз.
- **Інструменти**:
  - **OWASP ZAP** або **Burp Suite** для тестування безпеки в реальному часі.
  - **SonarQube**, **Checkmarx** для статичного аналізу коду.

3.3 Використання політики управління доступом

- **Рекомендація**: Для доступу до веб-додатків використовувати принципи найменших привілеїв та контролю доступу на основі ролей (RBAC) або атрибутів (ABAC).
- **Інструменти**:
  - **OAuth 2.0**, **OpenID Connect** для реалізації безпечного автентифікації та авторизації.

4. Управління уразливостями та реагування на інциденти

4.1 Виявлення та управління вразливостями

- **Рекомендація**: Регулярно перевіряти системи на наявність

вразливостей, особливо у сторонніх компонентах, таких як бібліотеки, фреймворки чи модулі.

- **Інструменти**:
  - **OWASP Dependency-Check** для виявлення вразливостей у сторонніх бібліотеках.
  - **Qualys**, **Nessus** для сканування мереж і серверів на наявність вразливостей.

4.2 Моніторинг та аудит безпеки

- **Рекомендація**: Встановити систему моніторингу для виявлення аномальних дій та вторгнень, а також вести аудити безпеки.
- **Інструменти**:
  - **SIEM-системи**: **Splunk**, **ELK Stack**, **Graylog** для збору і аналізу логів.
  - **IDS/IPS**: **Snort**, **Suricata** для виявлення і запобігання вторгнень.

4.3 Реагування на інциденти

- **Рекомендація**: Розробити план реагування на інциденти (IRP) і проводити регулярні тренування команди безпеки щодо потенційних сценаріїв атак.
- **Інструменти**:
  - **TheHive**, **Cortex** для автоматизації процесу реагування на інциденти.

5. Моніторинг та управління ризиками

5.1 Використання систем управління ризиками

- **Рекомендація**: Оцінювати та управляти інформаційними ризиками за допомогою спеціалізованих платформ, які допомагають аналізувати та класифікувати ризики.
- **Інструменти**:
  - **OWASP SAMM (Software Assurance Maturity Model)** для оцінки рівня зрілості практик безпеки.
  - **NIST Cybersecurity Framework** для загальної оцінки ризиків і

управління ними.

5.2 Проведення регулярних оцінок ризиків

- **Рекомендація**: Проводити регулярні оцінки ризиків і моделювання загроз на всіх етапах життєвого циклу розробки програмного забезпечення.
- **Інструменти**:
  - **OWASP Risk Rating Methodology** для визначення та оцінки ймовірності загроз.
  -

Висновки до розділу 3

Розробка топології системи забезпечення захисту інформаційної безпеки на базі рішень OWASP дозволяє побудувати багаторівневу систему захисту для веб-додатків. Використання інструментів, таких як **OWASP ZAP**, **OWASP Dependency-Check**, **WAF**, **SIEM** та інших, дозволяє ефективно моніторити, тестувати і управляти безпекою веб-додатків, знижуючи ймовірність атак і забезпечуючи високу стійкість до вразливостей. Інтеграція рішень OWASP у життєвий цикл розробки програмного забезпечення дозволяє ефективно впроваджувати безпеку на кожному етапі розробки. Це включає використання таких інструментів, як OWASP ZAP, OWASP Dependency-Check, OWASP ASVS, а також застосування принципів безпеки, визначених у **OWASP Top 10**. Важливо не лише інтегрувати безпеку на етапі тестування, але й закласти її на етапах планування, проєктування та розробки, що допоможе зменшити ризики та підвищити стійкість додатка до атак. Інтеграція технологій забезпечення захисту інформаційної безпеки в організаційні процеси і життєвий цикл розробки програмного забезпечення є ключовим елементом забезпечення кібербезпеки. Рекомендується застосовувати багаторівневий підхід до безпеки, включаючи захист мережі, шифрування даних, безпеку додатків, моніторинг та реагування на інциденти. Регулярне оновлення технологій, використання новітніх інструментів для аналізу та тестування вразливостей, а також інтеграція безпеки на етапі розробки допоможуть знизити ймовірність кіберзагроз і підвищити стійкість систем до атак.

ВИСНОВКИ

В кваліфікаційній роботі отримано наступні наукові та науково-практичні результати:

1. Досліджено актуальність проблеми забезпечення інформаційної безпеки, оскільки зростаюча залежність від цифрових технологій та інтернет-ресурсів створює нові виклики в сфері захисту даних. Атаки на інформаційні системи стають все більш складними та зухвалими, що вимагає постійного вдосконалення заходів безпеки для мінімізації ризиків втрати або компрометації чутливої інформації.

2. Проаналізовано наукову та технічну літературу за темою кваліфікаційної роботи, що охоплює різноманітні аспекти забезпечення інформаційної безпеки в умовах сучасних викликів цифрової трансформації. У рамках дослідження було розглянуто праці вітчизняних та зарубіжних авторів, що аналізують як теоретичні основи, так і практичні методи захисту інформаційних систем.

3. Проаналізовано існуючі технології забезпечення захисту інформаційної системи, зокрема, методи та інструменти, що дозволяють ефективно протидіяти загрозам інформаційній безпеці на різних етапах обробки та зберігання даних. Особливу увагу приділено сучасним підходам до захисту інформаційних ресурсів, що базуються на комплексних рішеннях, здатних інтегруватися з існуючою інфраструктурою організацій.

4. Розроблено варіант топології системи забезпечення захисту інформаційної безпеки на базі рішення OWASP, що включає використання ряду ефективних інструментів та практик для забезпечення безпеки веб-додатків. Зокрема, цей підхід спрямований на захист від основних загроз, які визначаються в рамках OWASP Top 10 (найпоширеніші уразливості та атаки на веб-додатки).

5. Розроблено методи та засоби забезпечення кібербезпеки, що включають комплексний підхід до захисту інформаційних систем від різноманітних кіберзагроз. Вони орієнтовані на забезпечення конфіденційності, цілісності та доступності даних, а також на мінімізацію ризиків, пов'язаних з кіберінцидентами.

Застосування рішень OWASP та інших сучасних технологій безпеки у розробці, впровадженні та експлуатації інформаційних систем дозволяє значно знизити ризики кіберзагроз, підвищити ефективність захисту та забезпечити належний рівень безпеки для організацій та їх користувачів. Важливо не лише реалізувати технології безпеки, але й постійно їх адаптувати, навчаючи команду та вдосконалюючи практики на кожному етапі життєвого циклу програмного забезпечення.

ПЕРЕЛІК ПОСИЛАНЬ

1. OWASP Foundation. *OWASP Top 10 - 2021*. веб-сайт. URL: https://owasp.org/www-project-top-ten/ (дата звернення: 15.10.2024)

2. Яловик Д. В. ПОБУДОВА OPEN WORLDWIDE APPLICATION SECURITY PROJECT (OWASP). Актуальні проблеми кібербезпеки : Всеукр. науково-практ. конф., м. Київ, 25 жовт. 2024 р. Київ, 2023. С. 217–219.

3. OWASP Foundation. *OWASP ASVS (Application Security Verification Standard)*. веб-сайт. URL: https://owasp.org/www-project-application-security-verification-standard/ (дата звернення: 2.11.2024)

4. OWASP Foundation. *OWASP ZAP (Zed Attack Proxy)*. веб-сайт. URL: https://owasp.org/www-project-zap/ (дата звернення: 19.10.2024)

5. OWASP Foundation. *OWASP Dependency-Check*. веб-сайт. URL: https://owasp.org/www-project-dependency-check/ (дата звернення: 15.10.2024)

6. OWASP Foundation. *OWASP SAMM (Software Assurance Maturity Model)*. веб-сайт. URL: https://owasp.org/www-project-samm/ (дата звернення: 19.10.2024)

7. OWASP Foundation. *OWASP Dependency-Track*. веб-сайт. URL: https://owasp.org/www-project-dependency-track/ (дата звернення: 17.10.2024)

8. OWASP Foundation. *OWASP Secure Coding Practices* (v2.0). веб-сайт. URL: https://owasp.org/www-project-secure-coding/ (дата звернення: 19.10.2024)

9. OWASP Foundation. *OWASP Risk Rating Methodology*. веб-сайт. URL: https://owasp.org/www-community/Risk_Rating_Methodology (дата звернення: 17.10.2024)

10. OWASP Foundation. *OWASP Security Knowledge Framework (SKF)*. веб-сайт. URL: https://owasp.org/www-project-security-knowledge-framework/ (дата звернення: 19.10.2024)

11. OWASP Foundation. *OWASP Juice Shop*. веб-сайт. URL: https://owasp.org/www-project-juice-shop/ (дата звернення: 19.10.2024)

12. OWASP Foundation. *OWASP Web Application Security Testing Checklist*. веб-сайт. URL: https://owasp.org/www-project-web-security-testing-guide/ (дата

звернення: 19.10.2024)

13. National Institute of Standards and Technology (NIST). (2018). *Framework for Improving Critical Infrastructure Cybersecurity (NIST CSF)*. веб-сайт. URL: https://www.nist.gov/cyberframework (дата звернення: 29.11.2024)

14. Microsoft. (2021). *Secure Coding Practices* by Microsoft. веб-сайт. URL: https://docs.microsoft.com/en-us/security/ (дата звернення: 5.11.2024)

15. OWASP Foundation. *OWASP ZAP - Beginner's Guide*. веб-сайт. URL: https://owasp.org/www-project-zap/ (дата звернення: 19.10.2024)

16. Qualys. *Qualys Vulnerability Management*. веб-сайт. URL: https://www.qualys.com/ (дата звернення: 1.11.2024)

17. Cloudflare. *WAF (Web Application Firewall)*. веб-сайт. URL: https://www.cloudflare.com/products/web-application-firewall/ (дата звернення: 19.10.2024)

18. OpenVPN. *OpenVPN - Open Source VPN*. веб-сайт. URL: https://openvpn.net/ (дата звернення: 29.11.2024)

19. Splunk. *Splunk SIEM*. веб-сайт. URL: https://www.splunk.com/ (дата звернення: 20.11.2024)

20. Burp Suite. *Burp Suite Professional - Web Vulnerability Scanner*. веб-сайт. URL: https://portswigger.net/burp (дата звернення: 24.11.2024)

21. SonarQube. *SonarQube - Continuous Inspection of Code Quality*. веб-сайт. URL: https://www.sonarqube.org/ (дата звернення: 29.11.2024)

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)

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

КВАЛІФІКАЦІЙНА РОБОТА
НА ТЕМУ:

«ТЕХНОЛОГІЯ ЗАБЕЗПЕЧЕННЯ КІБЕРБЕЗПЕКИ ВЕБ-ДОДАТКІВ НА БАЗІ РІШЕННЯ OWASP»

Виконав:
здобувач вищої освіти
групи БСДМ-63
ЯЛОВИК ДЕНИС

Керівник:
д.ф., **МАРЧЕНКО ВІТАЛІЙ**

Київ 2024

Об'єкт дослідження – процес безпечного функціонування веб-додатків

Предмет дослідження – технологія забезпечення кібербезпеки веб-додатків на базі рішення OWASP

Мета роботи – підвищення рівня інформаційної безпеки шляхом впровадження технічних та програмних рішень для посилення безпеки веб-додатків.

Наукові завдання:

- розроблення варіанта топології системи забезпечення захисту інформаційної безпеки на базі рішення OWASP
- проведення аналізу існуючих технологій забезпечення захисту інформаційної системи
- проведення аналізу наукової та технічної літератури за темою кваліфікаційної роботи.
- провести дослідження актуальності проблеми забезпечення інформаційної безпеки
- розробити методи та засоби забезпечення кібербезпеки

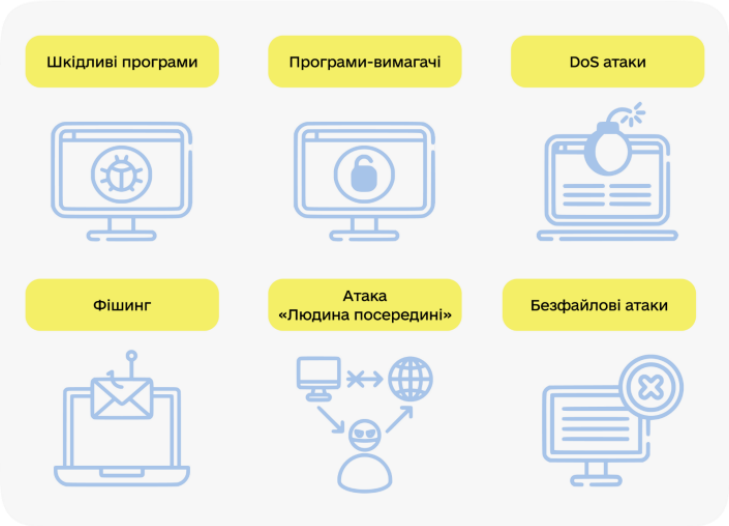

Рис.1.Види кіберзагроз

3

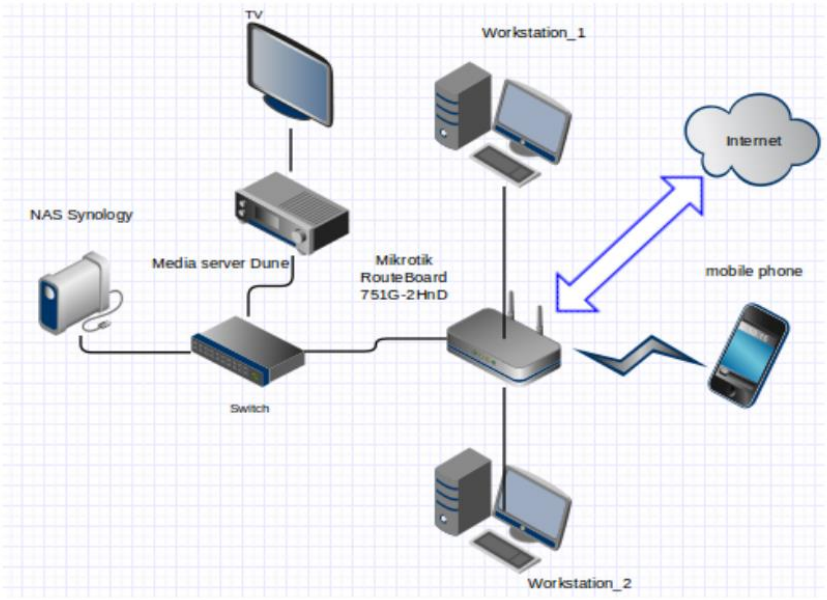

Рис.2. Сегментація мережі, використання протоколу 802.1Q

4

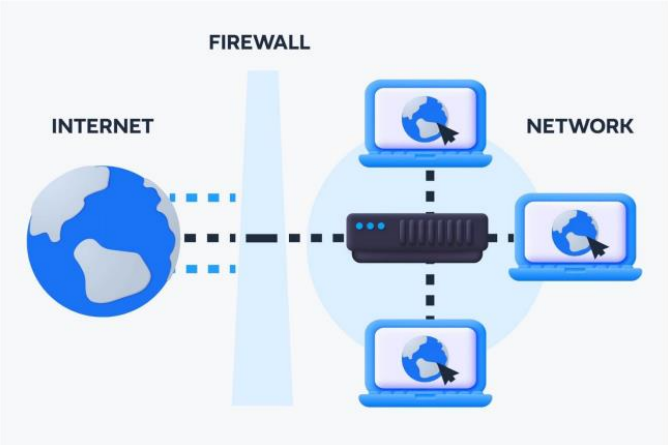

Рис.3.Приклад роботи фаєрволів

Рис.4.Різниця між програмним та апаратним шифруванням

5

Рис.5.Список десяти найбільш критичних вразливостей веб-додатків OWASP

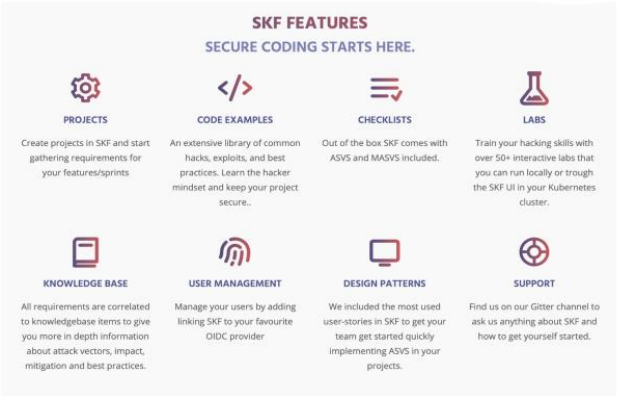

Рис.6.Можливості OWASP Security Knowledge Framework (SKF)

6

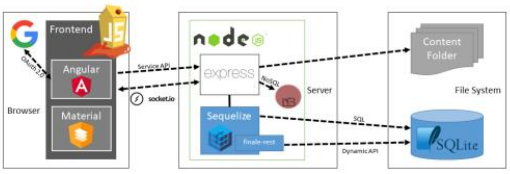

Рис.7.Ряд уразливостей в OWASP Juice Shop

Рис.8.Архітектура Node.js

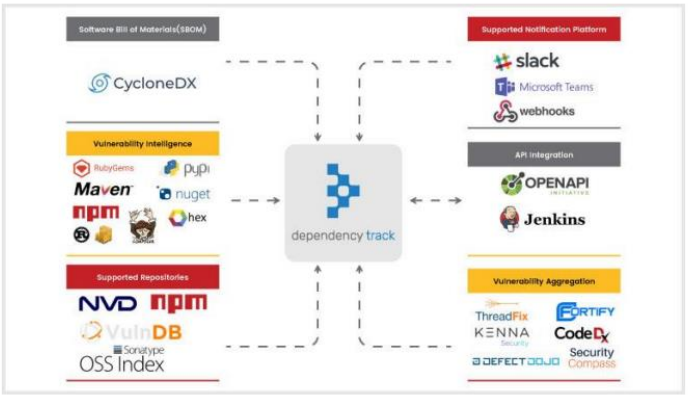

Рис.9.Використання OWASP Dependency-Check для покращення життєвого циклу програми

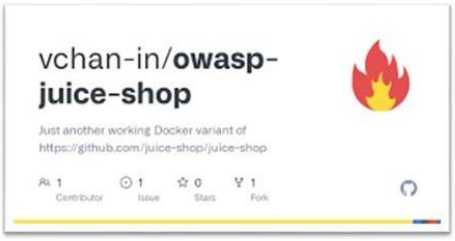

Рис.10.Створення директорії в GitHub

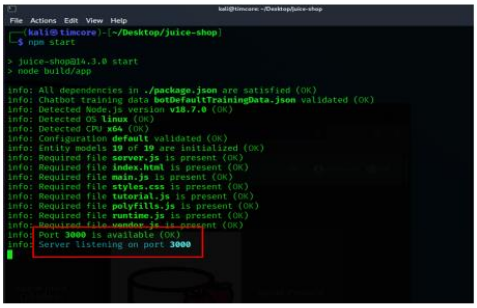

Рис.11.Запуск Juice Shop

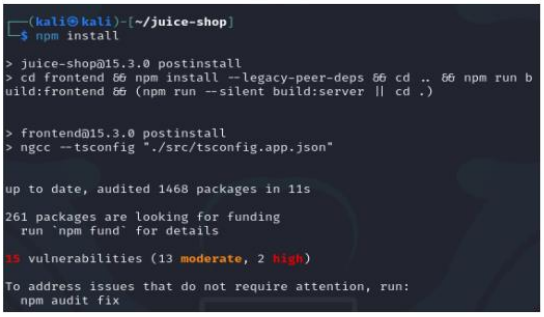

Рис.12.Встановлення необхідних компонентів

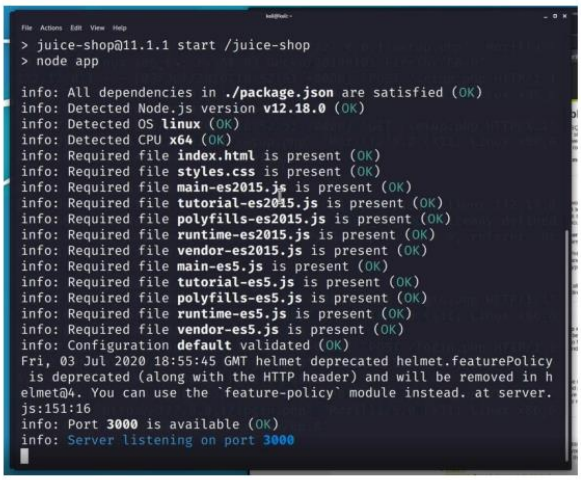

Рис.13.Запуск OWASP Juice Shop використовуючи Docker

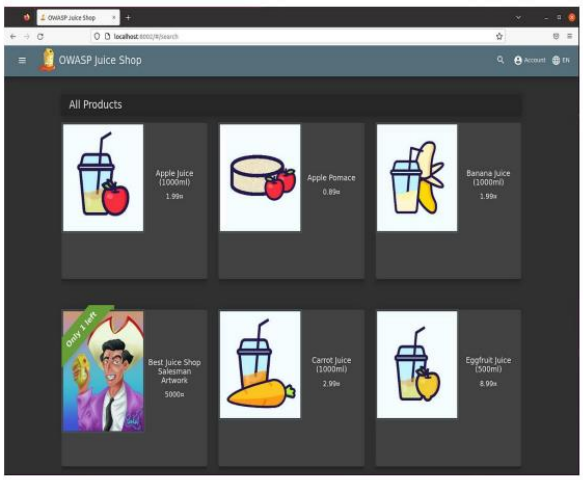

Рис.14.Головна сторінка OWASP Juice Shop

ВИСНОВКИ

1) Досліджено актуальність проблеми забезпечення інформаційної безпеки, оскільки зростаюча залежність від цифрових технологій та інтернет-ресурсів створює нові виклики в сфері захисту даних. Атаки на інформаційні системи стають все більш складними та зухвалими, що вимагає постійного вдосконалення заходів безпеки для мінімізації ризиків втрати або компрометації чутливої інформації.
2) Проаналізовано наукову та технічну літературу за темою кваліфікаційної роботи, що охоплює різноманітні аспекти забезпечення інформаційної безпеки в умовах сучасних викликів цифрової трансформації. У рамках дослідження було розглянуто праці вітчизняних та зарубіжних авторів, що аналізують як теоретичні основи, так і практичні методи захисту інформаційних систем.
3) Проаналізовано існуючі технології забезпечення захисту інформаційної системи, зокрема, методи та інструменти, що дозволяють ефективно протидіяти загрозам інформаційній безпеці на різних етапах обробки та зберігання даних. Особливу увагу приділено сучасним підходам до захисту інформаційних ресурсів, що базуються на комплексних рішеннях, здатних інтегруватися з існуючою інфраструктурою організацій.
4) Розроблено варіант топології системи забезпечення захисту інформаційної безпеки на базі рішення OWASP, що включає використання ряду ефективних інструментів та практик для забезпечення безпеки веб-додатків. Зокрема, цей підхід спрямований на захист від основних загроз, які визначаються в рамках OWASP Top 10 (найпоширеніші уразливості та атаки на веб-додатки).
5) Розроблено методи та засоби забезпечення кібербезпеки, що включають комплексний підхід до захисту інформаційних систем від різноманітних кіберзагроз. Вони орієнтовані на забезпечення конфіденційності, цілісності та доступності даних, а також на мінімізацію ризиків, пов'язаних з кіберінцидентами.