Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеню бакалавра

студента *Латишева Дмитра Олександровича*

академічної групи *125-17-1*

спеціальності *125 Кібербезпека*

спеціалізації[1]

за освітньо-професійною програмою *Кібербезпека*

на тему *Розробка засобів захисту веб-сайту підприємства «RubyGlobal» від несанкціонованого доступу*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	д.ф.-м.н., проф. Кагадій Т.С			
розділів:				
спеціальний	ст. викл. Мєшков В.І.			
економічний	к.е.н., доц. Пілова Д.П.			
Рецензент				
Нормоконтролер	ст. викл. Мєшков В.І.			

Дніпро
2021

ЗАТВЕРДЖЕНО:

завідувач кафедри
безпеки інформації та телекомунікацій
_____ д.т.н., проф. Корнієнко В.І.

«_____»_____20___року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня бакалавра

студенту ***Латишеву Дмитру Олександровичу*** **академічної групи** ***125-17-1***
(прізвище ім'я по-батькові) (шифр)

спеціальності ***125 Кібербезпека***
(код і назва спеціальності)

на тему ***Розробка засобів захисту веб-сайту підприємства «RubyGlobal» від несанкціонованого доступу***

затверджену наказом ректора НТУ «Дніпровська політехніка» від 07.06.2021 № 317-С

Розділ	Зміст	Термін виконання
Розділ 1	Стан питання. Постановка задачі. Проаналізовані актуальні атаки, інформація, що зберігається на сайті, повна характеристика сайту, проаналізовані доступи до інформації	29.03.2021 – 23.04.2021
Розділ 2	Створена модель порушника та модель загроз, проведене тестування та виправлення на наявність вразливостей у коді. Обраний профіль захищеності	26.04.2021 – 28.05.2021
Розділ 3	Розрахували економічну цінність та актуальність впровадження системи протидії витоку інформації	31.05.2021 - 11.06.2021

Завдання видано _____ _____
(підпис керівника) (прізвище, ініціали)

Дата видачі: 08.01.2021р.

Дата подання до екзаменаційної комісії: 12.06.2021р.

Прийнято до виконання _____ _____
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 71 с., 8 рис., 13 табл., 4 додатка, 8 джерел.

Об'єкт дослідження: WEB-сторінка підприємства «RubyGlobal» під назвою «Bookstore».

Мета роботи: розробити систему захисту інформації від несанкціонованого доступу.

Методи роботи: аналіз, розробка, порівняння.

В першому розділі був проведений повний аналіз WEB-сторінки, що розглядається, а саме: наведена карта сайту, наведена статистика атак, наданий повний перелік програмних засобів, що використовувались при розробці проекту, наданий перелік прав доступу до інформації персоналу адміністративної частини сайту.

В спеціальній частині проведений аналіз загроз та створена модель порушника, проведене тестування на наявність вразливостей в програмній частині сайту за допомогою спеціальних програмних засобів та бібліотек. Надані рекомендації для протидії вразливостям, які були виявлені при тестування та аналізу загроз.

В економічному розділі проведено розрахунок щодо доцільності впровадження запропонованих організаційних та програмних рішень.

Практичне значення роботи складається у тому, щоб дослідити систему безпеки інформації WEB-сторінки підприємства «RubyGlobal».

Результати досліджень, що здійснені під час кваліфікаційної роботи можуть бути використані для поліпшення системи безпеки WEB-сторінки підприємства «RubyGlobal».

ЗАГРОЗИ ВРАЗЛИВОСТІ WEB-СТОРІНКА КОНФІДЕНЦІЙНІСТЬ ПОРУШНИК ДАНІ

РЕФЕРАТ

Пояснительная записка: 71 с., 8 рис., 13 табл., 4 приложения, 8 источников.

Объект исследования: WEB-страница предприятия «RubyGlobal» под названием «Bookstore».

Цель работы: разработать систему защиты информации от несанкционированного доступа.

Методы работы: анализ, разработка, сравнения.

В первой главе был проведен полный анализ WEB-страницы, рассматривается, а именно: приведенная карта сайта, приведенная статистика атак, предоставлен полный перечень программных средств, которые использовались при разработке проекта, предоставленный перечень прав доступа к информации персонала административной части сайта.

В специальной части проведен анализ угроз и создана модель нарушителя, проведено тестирование на наличие уязвимостей в программной части сайта с помощью специальных программных средств и библиотек. Даны рекомендации для противодействия уязвимостям, которые были обнаружены при тестирования и анализа угроз.

В экономическом разделе проведен расчет о целесообразности внедрения предложенных организационных и программных решений.

Практическое значение работы состоит в том, чтобы исследовать систему безопасности информации WEB-страницы предприятия «RubyGlobal».

Результаты исследований, совершенные во время квалификационной работы могут быть использованы для улучшения системы безопасности WEB-страницы предприятия «RubyGlobal».

УГРОЗЫ УЯЗВИМОСТИ WEB-СТРАНИЦА КОНФИДЕНЦИАЛЬНОСТЬ НАРУШИТЕЛЬ ДАННЫЕ

ABSTRACT

Explanatory note: 71p., 8 pic., 13 tables, 4 supplements, 8 sources

Object of research: WEB-page of the enterprise "RubyGlobal" called "Bookstore".

Purpose of work: to develop a system for protecting information from unauthorized access.

Working methods: analysis, development, comparison.

In the first chapter, a complete analysis of the WEB page was carried out, it is considered, namely: the given sitemap, the given statistics of attacks, a complete list of software tools that were used in the development of the project, a list of access rights to information of the staff of the administrative part of the site.

In a special part, an analysis of threats was carried out and a model of an intruder was created, testing for the presence of vulnerabilities in the software part of the site was carried out using special software tools and libraries. Recommendations are given for countering vulnerabilities that were discovered during testing and threat analysis.

In the economic section, a calculation was made on the feasibility of introducing the proposed organizational and software solutions.

The practical significance of the work is to investigate the information security system of the enterprise "RubyGlobal" WEB-page.

The research results made during the qualification work can be used to improve the security system of the WEB-page of the enterprise "RubyGlobal".

VULNERABILITY THREATS WEB PAGE CONFIDENTIALITY BREAKER DATA

СПИСОК УМОВНИХ СКОРОЧЕНЬ

CSS - Cascading Style Sheets;

DNS - Domain Name System;

HTML - Hyper Text Mark up Language;

HTTP - Hyper Text Transer Protocol;

HTTPS - Hypertext Transfer Protocol Secure;

MVC - Model-View-Controller;

SMTP - Simple Mail Transfer Protocol;

SSH - Secure Shell;

SQL - Structured query language;

АС - Автоматизована система;

ІТС - Інформаційно-телекомунікаційна система;

КЗЗ - Комплекс засобів захисту;

КСЗІ - Комлпексні системи захисту інформації;

ПЗ - Програмне забезпечення;

СЗІ - Служба захисту інформації.

ЗМІСТ

ВСТУП

В наш час найпоширенішим видом зв'язку є мережа Інтернет - сукупність мереж та обчислювальних засобів, які використовують стек протоколів TCP/IP, спільний простір імен та адрес для забезпечення до інформаційних ресурсів мережі будь-якій особі.

Мережевим інформаційним ресурсом, до якого звертаються користувачі є WEB-сторінка, яка надається користувачу у вигляді документу, створений за допомогою мови гіпертекстової розмітки HTML (Hyper Text Mark-up Language). Кожна WEB-сторінка розгорнута на WEB-сервері, який обслуговує запити клієнтів згідно з протоколом HTTP (Hyper Text Transer Protocol), забезпечує актуалізацію, збереження інформації WEB-сторінки, зв'язок з іншими серверами. На сайті може надаватися інформація будь-якої тематики та характеру. Користувачі отримують доступ тільки до тієї інформації, яка є у вільному доступі на сайті або мають права на неї (наприклад та інформація, що доступна тільки після успішної аутентифікації та авторизації). На сайті обов'язково присутня корпоративна або адміністративна інформація, до якої звичайний користувач не має прав доступу.

Так як, для більшості населення мережа Інтернет є невід'ємною частиною життя через можливість завжди бути на зв'язку, робити покупки не виходячи з дому, працювати, вчитися, тощо, «діра» у захищеності будь-якої сторінки, може призвести до колосальних збитків як для користувача так і для власника WEB-сторінки.

На цьому етапі виникає питання правильного захисту і розмежування доступу до корпоративної інформації сайту від несанкціонованого доступу. Кожен сайт может стати «жертвою» різновидних атак, таких як: DDos або спам атаки для виведення з ладу вже серверної частини сторінки. отримання зловмисниками інформації, до якої звичайний відвідувач WEB-сторінки не має доступу.

На даний момент, у світі існує безліч видів вразливостей і загроз для WEB-сторінки та шляхів протидії даним проблемам. Основною метою цієї кваліфікаційної роботи буде аналіз для виявлення основних вразливостей, загроз,

які виникають через описані вразливості та їх реалізація на прикладі реальної WEB-сторінки та подальший опис знешкодження вразливостей, що перешкоджають безпечному і захищенному існуванню, зберіганню та обробці інформації на WEB-сторінці.

РОЗДІЛ 1 СТАН ПИТАННЯ. ПОСТАНОВКА ЗАДАЧІ

1.1 Загальні відомості

Прикладом web-сторінки з використанням технології Т2, яка підлягає перевірці на виконання вимог документу «Вимоги до захисту інформації WEB-сторінки від несанкціонованого доступу» НД ТЗІ 2.5-010 виступає https://dimasfullxq-bookstore-app.herokuapp.com. На прикладі даної WEB-сторінки будуть розглянуті вид і стан інформації, що зберігається на сайті, несанкціонований доступ до цієї інформації. Буде розглянуто повний опис сайту як об'єкту, а саме:

- карта сайту (сторона користувача)
- який хостинг використовується;
- що надає даний тип хостингу;
- яка операційна система використовується на самому сервері;
- яка мова програмування була використана для написання даної WEB-сторінки;
- яке додаткове програмне забезпечення було використане при написанні і використанні WEB-сторінки;
- персонал, який приймає участь в адміністративній частині сайту;
- права доступу до ресурсів, розміщених на сайті.

1.1.1 Карта сайту (сторона користувача)

На сайті присутня голова сторінка, каталог книг, окрема сторінка для кожної книги. Присутня сторінка реєстрації та автентифікації. Окремо прописана сторінка налаштувань особоистого облікового запису користувача і список особистих активних, завершених або відхилених замовлень користувача. Так як сайт виступає сторінкою продажу книг, присутня сторінка кошику в якому створюється замовлення і переходить на усі етапи статусу замовлення. З будь-якої сторінки сайту можна перейти на будь-яку із перерахованих (з умовою успішної авторизації та автентифікації, де вона потрібна)

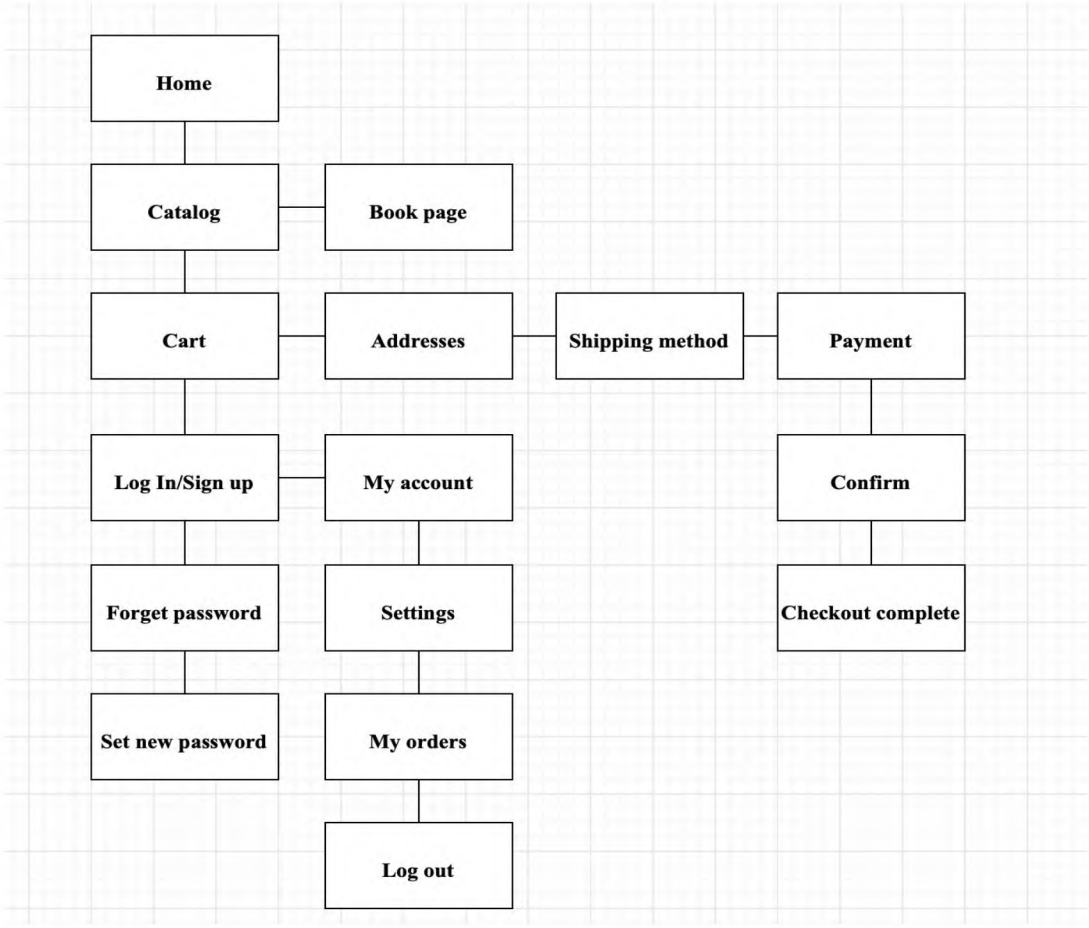

Рисунок 1.1 – Карта сайту

1.2 Інформація, що зберігається на сайті

На WEB-сторінці, що розглядається зберігається:

– інформація про дані всіх товарів, що були і є у продажі на сайті.

– Дана інформація є інформацією вільного доступу, що означає, що навіть якщо ця інформація будь-яким чином буде вкрадена або використана зловмисником у своїх цілях, ці дії не несуть ніякої шкоди або витрат для власника WEB-сторінки. Втрати конфіденційності відсутні;

– інформація про особисті дані існуючих користувачів (електронна пошта, мобільний телефон, адреса (шипінгова та білінгова адреса), платіжна система, дані про банківську карту/карти, які були вказані для оплати). Дана інформація є конфіденційною, так як інформація включає в себе особисті дані користувачів. Розкриття цієї інформації може призвести до серйозних збитків;

– інформація про замовлення, які були створені через сайт. Дана інформація є конфіденційною, так як включає в себе особисту та платіжну інформацію клієнтів. Розкриття цієї інформації може призвести до серйозних збитків;

– адміністративна інформація та електронний облік товару. Дана інформація є конфіденційною. Розкриття цієї інформації призведе до серйозних збитків.

1.3 Актуальні атаки, які можуть використовуватись та протидія цим атакам

Посилаючись на книгу [1] найбільш популярною атакою є «Insufficient transport layer protection» - отримання даних під час передавання. Дана атака може бути виконана для 70% всіх WEB-ресурсів. Для виключення можливості проведення таких атак достатньо використовувати протокол HTTPS (протокол HTTP, який використовується з додатковим рівнем безпеки передачі даних) для передачі усіх даних.

Витік інформації («Information leakage»). Дану атаку можна виконати на 56% ресурсів. Витік інформації з додатків виникає в результаті відмови або неправильної роботи програми, а також у разі порушення її логіки. Для виключення можливості проведення даної атаки необхідно ретельно тестувати програмну частину ресурсу, проводити перевірку повідомлень на стороні сервера, моніторинг оповіщень про помилки (наприклад використовуючи сервіс сповіщень Sentry).

Атаку «Cross-site scripting» - міжсайтове використання сценаріїв, можливо виконати на 47% ресурсів. Атака дозволяє передати JavaScript-код на виконання в браузер користувача. Атаку такого роду часто називають HTML-ін'єкціями, адже механізм цього впровадження дуже схожий із SQL-ін'єкціями, але на відміну від останніх, впроваджуваний код використовується в браузері користувача. Для захисту від цього виду атак необхідно проводити очищення та валідацію вхідних даних.

Генерацію великої кількості запитів або підбір паролів («Brute force») можливо виконати на 29% ресурсів. Для захисту необхідно забезпечити використання

паролів високої складності, налаштування сервера на аналіз вхідних запитів та налаштування повідомлень для загальної валідації, а не специфічної на окреме поле у формі.

Атака «Content spoofing» - підміна даних через заміну контенту сторінок можлива для 26% ресурсів. Використовуючи цю техніку, зловмисник змушує користувача повірити, що сторінка згенерована WEB-сервером, а не передана із зовнішнього джерела. Для захисту від даного виду атак потрібно відмовитися від використання фреймів і, найголовніше, ніколи не передавати в параметрах абсолютні або локальні шляхи до файлів.

Вид атак на відвідувачів WEB-сайтів, який використовує недоліки протоколу HTTP - «Cross-site request forgery». Якщо жертва заходить на сайт, створений зловмисником, від її особи таємно відправляється запит на інший сервер (наприклад, на сервер платіжної системи), який здійснює якусь шкідливу операцію (наприклад, переказ грошей на рахунок зловмисника). Дану атаку можливо виконати на 24% ресурсів. Для захисту необхідно проводити перевірку вхідних даних з форм, наприклад шляхом додавання унікального токену.

Перенаправлення на інші сайти через підміну початкових посилань, атака «URL redirector abuse». Цей вид вразливостей, також як і багато інших перерахованих вище, є різновидом помилок перевірки вхідних даних і можлива на 16% ресурсів. Вирішенням є використання валідації на вхідних даних.

Ще однією популярною атакою є «Predictable resource location» - знаходження прихованого функціоналу та даних. Доступна на 15% ресурсів і вирішується шляхом контролю доступу до файлів сервера.

Виходячи з наведених даних, можна зробити висновки про те, що для захисту від більшості популярних видів атак достатньо належним чином перевіряти вхідні дані. Також рекомендовано використовувати шифрований протокол HTTPS та будувати програмний додаток ресурсу на одному з відомих каркасів (Frame-Work), в якому вбудовані механізми перевірки, шифрування та валідації.

Таблиця 1.1 – Класифікація видів атак, їхня розповсюдженість і методи протидії

Вид атаки	Вразливість WEB-ресурсів, %	Протидія
Insufficient transport layer protection	70	Використання протоколу HTTPS
Information leakage	56	Тестування програмної частини ресурсів, перевірка повідомлень на стороні сервера, моніторинг оповіщень про помилки
Cross-site scripting	47	Очищення та валідація вхідних даних
Brute force	29	Використання паролів високої складності, налаштування сервера та аналіз вхідних запитів
Content spoofing	26	Відмовитися від використання фреймів і не передавати в параметрах абсолютні або локальні шляхи до файлів
Cross-site request forgery	24	Перевірка вхідних даних з форм
URL redirector abuse	16	Валідація вхідних даних
Predictable resource location	15	Контроль доступу до файлів сервера

1.4 Характеристика WEB-сторінки

1.4.1 Хостинг

WEB-сторінка, що розглядається знаходиться за адресою https://dimasfullxq-bookstore-app.herokuapp.com. Хостинг, що використовується heroku.com - на безкоштовній підписці, що лише надає можливість заходити на цей сайт будь-

якому користувачу, звідки завгодно. Недоліком даної форми підписки є факт того, що якщо на відсутні запити більше ніж двадцять хвилин - сайт «засинає» і наступний запит до сторінки потребую часу для підгрузки всієї WEB-сторінки.

1.4.2 Операційна система на віддаленому ресурсі

На віддаленому ресурсі, який «підіймає» весь додаток використовується Операційна Система Ubuntu 18.04, що надає можливість використовувати усі відомі команди терміналу unix системи прямо на сервері (наприклад для написання скриптів на автоматичне оновлення пакетів і бібліотек прямо на сервері або скриптів для усунення програмних помилок). Для підтримки і коректного використання даної WEB-сторінки (саме програмної частини) рекомендується використовувати операційну систему типу unix (Mac OS/Linux) так як мова програмування і додаткове програмне забезпечення, яке використовуються у програмній частині додатку може частково або повністю не працювати на іншій системі (операційні системи сімейства Microsoft Windows).

1.4.3 Мова програмування та бібліотеки, що були використані при написанні back-end (серверної) частини WEB-сторінки

WEB-сторінка, що розглядається була написана на об'єктно-орієнтованій мові програмування Ruby версії 2.7.1 з використанням MVC (Model-View-Controller) FrameWork'у Ruby on Rails версії 6.0 [2, 3].

Перелік бібліотек, надалі gem (гем/джем), опис яких знаходиться на WEB-ресурсі [4] на офіційніц сторінці кожного з перерахованих гемів, що були використані та надалі використовуються при розробці сайту:

Для production середовища сайту використовувались:

– gem «aasm» - це бібліотека для емуляції state-machine була використана для простішого змінення статусу об'єкту і переходу з одного стану в інший в автоматизованому режимі без зайвих перевірок і складних алгоритмів;

– gem «acts_as_list» - бібліотека, що використовується для простішої емуляції (без зайвих алгоритмів і коду) внутрішнього списку однієї сутності в іншій;

– gem «activeadmin» - це бібліотека для Ruby on Rails додатку для генерації інтерфейсів у стилі адміністрування. Він абстрагує загальні шаблони бізнес-застосунків, щоб спростити розробникам реалізацію красивих та елегантних інтерфейсів з дуже не великими зусиллями;

– gem «bcrypt» - це бібліотека, яка використовується для шифрування і зберігання у зашифрованому вигляді паролю або інших полів для запобігання викрадення даних;

– gem «carrierwave» - ця бібліотека забезпечує простий і надзвичайно гнучкий спосіб завантаження файлів із програм Ruby;

– gem «draper» - бібліотека, що додає об'єктно-орієнтований рівень логіки презентацій до вашого додатку Rails. Без Draper ця функція могла б заплутатися у процедурних помічниках або додати велику кількість до ваших моделей. За допомогою декораторів Draper ви можете обгортати свої моделі логікою, пов'язаною з презентаціями, щоб набагато ефективніше організувати та протестувати цей шар програми;

– gem «fog-aws» - бібліотека, що використовується для зв'язку додатку з хмарним сховищем (Amazon Web Services), в якому зберігаються такі дані як текстові документи, картинки, фотографії;

– gem «friendly_id» - бібліотека, яка використовується для створення гарні URL-адреси та працювати зі зручними «людськими» рядками так, ніби це числові ідентифікатори;

– gem «mini_magick» - бібліотека, яка використовується для форматування зображень, які потрібні для додатку «на льоту» під час виконання запиту на сервер і зберігання у новому вигляді у базі даних;

– gem «omniauth-facebook» - бібліотека, яка використовується суміжно з devise для автентифікації, але саме для можливості автентифікуватися використовуючи сторінку facebook;

– gem «pagy» - бібліотека, яка використовується для гнучкої пагінації товарів на сторінці магазину;

– gem «pg» - бібліотека, яка використовується для підключення до Rails-додатку бази даних PostgreSQL;

– gem «pundit» - бібліотека, яка використовується для авторизації (розмежування доступу до ресурсів або окремих полів) на WEB-сторінці, шляхом написання так званих Policy правил доступу до окремих ресурсів сайту для кожного типу користувача (якщо такі присутні);

– gem «sidekiq» - бібліотека, яка використовується для асинхронизації і паралельних процесів у Ruby/Ruby on Rails додатку. Sidekiq використовує потоки для обробки багатьох завдань одночасно в одному процесі. Для цього не потрібні Rails, але вони будуть тісно інтегровані з Rails, щоб зробити фонову обробку простою. Гем після запуску «підіймає» точну копію додатку, яка працює асинхронно з використанням тої самої бази даних і налаштувань. Прикладом використання є відправка, наприклад, щомісячних повідомлень клієнтам через електронну пошту;

– gem «truemail» - бібліотека, що використовується для валідації електронної пошти через регулярні вирази, DNS, SMTP. Гем навіть перевіряє чи дійсно надана електронна пошта є існуючою серед усіх існуючих електронних пошт;

– gem «devise» - це гнучке рішення для автентифікації Rails на базі Warden.

Він складається з 10 модулів:

– Database Authenticatable: хешує та зберігає пароль у базі даних для перевірки справжності користувача під час входу. Аутентифікація може бути здійснена як за допомогою запитів POST, так і за допомогою базової автентифікації HTTP;

– Omniauthable: додає підтримку OmniAuth;

– Confirmable: надсилає електронні листи з інструкціями підтвердження та перевіряє, чи вже підтверджено обліковий запис під час входу;

– Recoverable: скидає пароль користувача та надсилає інструкції щодо скидання;

– Registerable: обробляє реєстрацію користувачів у процесі реєстрації, а також дозволяє редагувати та знищувати свій обліковий запис;

– Rememberable: управляє генерацією та очищенням маркера для запам'ятовування користувача із збереженого файлу cookie;

– Trackable: кількість входів, позначки часу та IP-адреса;

– Timeoutable: закінчується сеанс, який не був активним протягом певного періоду часу;

– Validatable: забезпечує перевірку електронної пошти та пароля. Це необов’язково, і його можна налаштувати, тому ви можете визначити власні перевірки;

– Lockable: блокує обліковий запис після вказаної кількості невдалих спроб входу. Можна розблокувати електронною поштою або через певний проміжок часу;

Для development середовища використовувались:

– gem «pry-rails» - бібліотека, що використовується для дебагінгу (налагодження) додатку;

– gem «brakeman» - бібліотека статичного аналізу, яка перевіряє додатки Ruby on Rails на наявність вразливих місць безпеки;

– gem «bullet» - бібліотека розроблена, щоб допомогти підвищити продуктивність програми, зменшивши кількість запитів, які вона робить. Bullet буде спостерігати за вашими запитами під час розробки вашого додатка та повідомлятиме вас, коли слід додати eager завантаження (запити N + 1), коли ви використовуєте eager завантаження, яке не є необхідним, і коли слід використовувати кеш-лічильник;

– gem «bundler-audit» - бібліотека, яка використовується для аналізу «свіжості» і «безпечності» інших встановлених бібліотек. Надає такі можливості:

– перевірка на наявність уразливих версій гемів у Gemfile.lock (файл в якому знаходиться список усіх гемів і їх залежностей);

– перевірка небезпечних джерел всіх гемів (http: //);

– дозволяє ігнорувати певні рекомендації, які були оброблені вручну;

– друкує довідкову інформацію;

– не вимагає підключення до мережі.

– gem «database_consistency» - це бібліотека, яка використовується, щоб забезпечити простий спосіб перевірити відповідність обмежень бази даних валідаціям програм. Цей гем надає можливості для пошуку таких відмінностей між моделлю та записом у базі даних:

1) Знайти відсутні нульові обмеження (ColumnPresenceChecker);

2) Знайти перевірні відсутні довжини (LengthConstraintChecker);

3) Знайти відсутні перевірки присутності (NullConstraintChecker);

4) Знайти відсутні перевірки унікальності (UniqueIndexChecker);

5) Знайти відсутні зовнішні ключі для асоціацій BelongsTo (BelongsToPresenceChecker);

6) Знайти відсутні унікальні індекси для перевірки унікальності (MissingUniqueIndexChecker);

7) Знайти відсутні індекси для асоціацій HasOne та HasMany (MissingIndexChecker);

8) Знайти первинні ключі з цілочисельним / послідовним типом (PrimaryKeyTypeChecker);

9) Знайти невідповідність типів первинних ключів їх зовнішнім ключам (ForeignKeyTypeChecker);

10) Знайти надлишкові не унікальні індекси (RedundantIndexChecker);

11) Знайти надлишкове обмеження унікальності (RedundantUniqueIndexChecker);

– gem «letter_opener» - бібліотека, що використовується для відкриття електронних листів, які відправляються з Ruby on Rails додатку, в окремому вікні браузера замість реальної відправки. Цей гем надає можливість не турбуватись про випадкові відправки електронної пошти на чиюсь реальну адресу у development середовищі;

– gem «lefthook» - бібліотека, що використовується для самоперевірки на чистоту твого коміту (знімок останніх дій) або пушу (відправка змін на віддалений репозиторій) завдяки запуску всіх лінтерів та аналізаторів коду, які були прописані в конфігурації даного гему. Якщо хоча б одна перевірка не проходить, то коміт/пуш не буде виконаний;

– gem «rubocop» - бібліотека, яка виступає аналізатором статичного коду Ruby (інакше лінтером) та програматором коду. З під коробки гему він буде застосовувати багато вказівок, викладених у спільноті Ruby Style Guide. Окрім повідомлення про проблеми, виявлені у вашому коді, RuboCop також може автоматично виправити багато з них для вас.

Для test середовища і тестування всього додатку були написані юніт-тести - тести усіх окремих частин додатку таких як сервіси, декоратори, презентори, моделі, контролери та інтеграційні тести - тести на окрему повну функціональну частину додатку (feature) або на окремий endpoint (кінцева точка). Усі тести написані з допомогою RSpec [5] - фреймворку для тестування Ruby/Ruby on Rails додатків, який встановлюється як звичайний gem.

Для test середовища використовувались:

– gem «capybara» - бібліотека, яка використовується для тестування WEB-програми, змоделювавши, як реальний користувач взаємодіє з вашим додатком. Гем надає можливість протестувати будь-яку окрему сторінку або групу пов'язаних між собою сторінок додатку, шляхом написання інтеграційних тестів;

– gem «capybara-screenshot» - бібліотека, що використовується для відтворення знімку сторінки, яка тестувалась з допомогою capybara та тест не пройшов;

– gem «factory_bot_rails» - бібліотека, яка використовується для генерації моделей як для test так і для development середовища, для спрощеного і швидкого створення та емуляції усіх потрібних сутностей для потрібних тестів

– gem «faker» - бібліотека, яки використовується як в test так і в development середовищі для генерації випадкових даних, таких як імена, електронні пошти, паролі, заголовки тощо;

– gem «pundit_matchers» - бібліотека, що використовується для простішого тестування правил розмежування доступу написаних з допомогою гему для авторизації pundit;

– gem «shoulda_matchers» - бібліотека, що надає сумісні з RSpec однолінійки для тестування загальних функціональних можливостей Rails, які, якщо писати їх від руки, будуть набагато довшими, складнішими та схильними до помилок;

– gem «simplecov» - бібліотека аналізу охоплення коду для Ruby. Він використовує вбудовану в Ruby бібліотеку Coverage для збору даних про покриття коду, але значно полегшує обробку результатів, забезпечуючи чистий API для фільтрування, групування, об'єднання, форматування та відображення цих результатів, надаючи вам повний набір покриття коду, який можна налаштовано лише на кілька рядків коду;

– gem «site_prism» - бібліотека для створення об’єктної моделі сторінки для capybara. SitePrism надає простий, чистий і семантичний DSL для опису вашого сайту за допомогою шаблону об’єктної моделі сторінки, для використання Capybara при автоматизованому тестуванні інтеграційних сторінок.

1.4.4 Автентифікація

Автентифікація - це процес перевірки достовірності наданого користувачем ідентифікатора. Найчастіше використовується базова автентифікація - найпростіший спосіб обмеження доступу до веб-документів.

Можливість автентифікуватися надається за допомогою гему Devise, що описаний вище.

На сайті налаштована можливість автентифікуватися з допомогою таких способів:

– автентифікуватися з логіном (електронна пошта) та паролем (пароль у базі даних зберігається у зашифрованому вигляді, що запобігає можливості використання паролю користувача будь-яким адміністратором сайту);

– автентифікуватися з використанням існуючого (або створити новий) облікового запису соціальної мережі Facebook;

– автентифікуватися без використання паролю, так звана «швидка реєстрація». В даному випадку користувач вводить лише свою електронну пошту, пароль користувача генерується автоматично і на введену пошту відправляється лист з інструкцією для анулювання наданого паролю і введенням нового паролю, який користувач зможе використовувати для звичайної автентифікації (використання логіну та паролю).

1.4.5 Авторизація

Авторизація - процес розмежування доступу і прав до ресурсів WEB-сторінки між користувачами, адміністраторами та іншими ролями, які прописані на сайті.

Авторизація на сайті, що розглядається, налаштована з використанням гему Pundit з написанням відповідних policy (правил доступу) для кожної з ролей.

1.4.6 Додаткове програмне забезпечення, яке використовувалось при написанні сайту

1.4.6.1 База даних

Дуже важливим аспектом при написанні будь-якого додатку є вибір бази даних, яка буде використовуватись для зберігання усіх даних та обробки запитів для виводу даних. Так як на даному проекті більша частина дій пов’язаних з базою даних є зчитування великого об’єму даних з наявністю великої кількості зв'язків між сутностями і записами, потрібно використовувати реляційну базу даних (тип бази даних, яка підтримує зв’язки між сутностями).

Основною базою даних, яка використовується на сайті є PostgreSQL 12 для зберігання основного багажу даних проекту.

Також на проекті використовується додаткова NoSql (не реляційна) база даних Redis, яка після переповнення свого місця знищує найстаріші записи. Redis - це сховище структур даних з відкритим кодом (з ліцензією BSD), яке

використовується як база даних, кеш-пам'ятки та посередник повідомлень. Redis надає такі структури даних, як рядки, хеші, списки, набори, відсортовані набори із запитами діапазонів, растровими зображеннями, гіперлогічними журналами, геопросторовими індексами та потоками. Ця база даних використовується для зберігання проміжних даних, які навіть якщо будуть втрачені, від цього наслідків не буде.

1.4.6.2 Віртуалізація

Для створення віртуального образу всього проекту з «підняттям» серверу додатку, бази даних, асинхронних процесів та конфігурація додатку для подальшего деплою на будь-якій платформі використовується інструментарій для управління ізольованими Linux-контейнерами Docker.

1.4.6.3 Розгортання на віддаленному сервері

Додаток був розгорнутий (задеплоєний) на платформі Heroku. Для зв'язку з цією платформою і управлінням додатком на віддаленому сервері була використана консольна утіліта Heroku CLI.

1.5 Інформаційні потоки

Інформаційні потоки - потік повідомлень в мовної, паперовій або електронній формах, призначений для реалізації керуючих функцій в логістичній системі і обумовлений конкретним матеріальним потоком. Інформаційні потоки в логістичній системі розглядаються як супутні [6].

Цей потік можна аналізувати в трьох аспектах:

- синтаксичному — встановлює формальні правила (параметри) побудови інформаційного потоку, взаємозв'язок між його елементами;
- семантичному — встановлює правила інтерпретації кожного елементу інформаційного потоку;
- прагматичному — встановлює ступінь корисності кожного елементу інформаційного потоку для цілей управління;

Інформаційні потоки мають наступні характеристики:

– взаємозалежність - потік не може існувати в одній точці, потік обов'язково з'єднує дві точки, а це створює взаємозалежність, в іншому випадку буде мати місце інформаційна ізоляція;

– диференційованість - подібно всім електронним потокам, електронна інформація визначається через відмінність. Дані відносяться до інформації через контекст, який пов'язує їх з іншими даними. Дві частини даних можуть бути досить різні, щоб при зв'язку один з одним з'явилася нова відмінність, але в той же час дві частини даних повинні бути досить подібні, щоб бути пов'язаними в принципі;

– мінливість - потік даних не просто з'єднує два вузли, поєднуючи, він змінює те, що пов'язує. Зміна, однак, ні додає, ні віднімає значення течії потоків. Нові потоки інформації можуть змінити все. Взаємозалежність вузлів створює інформаційне середовище, в якій потоки, проходячи через кожен вузол, збільшуються або зменшуються в розмірі або швидкості;

– залежність від часу - найважливіша характеристика потоку. В навколишньому середовищу, де потоки даних циркулюють дуже швидко і проходять зі швидкістю світла через комп'ютерні мережі, нові взаємозв'язки виникають з такою ж швидкістю, з якою вмирають старі зв'язки. Час - центральний фактор в процесі. Ми живемо в епоху інформації, швидкість і обсяг якої обумовлені новими технологіями, які ведуть до об'єднання світових інформаційних мереж. Інформаційні потоки як елемент соціальних змін виявляються тільки після позначення їх дії в реальному часі і в одній інформаційній системі, що зв'язує прогресивний соціум.

Види інформаційних потоків:

– горизонтальні - інформаційні потоки між рівними по службовому положенню і статусу працівниками або групами працівників (наприклад між керівниками різних відділів підприємства);

– вертикальні - інформаційні потоки між працівниками або групами працівників, які знаходяться на різних рівнях ієрархії підприємства (наприклад між керівником і підлеглими).

В свою чергу, вертикальні інформаційні потоки розподіляються на низхідні, коли інформація передається від керівництва до рядових робітників по ієрархії, та висхідні - в цьому випадку інформація надається від підлеглих посад до керівних відповідно.

Горизонтальні інформаційні потоки

Найчастіше даний вид інформаційних потоків мають неформальний характер. Горизонтальні інформаційні потоки є найефективнішими, з комунікативної точки зору. В них зберігається приблизно 90% відомостей. Це означає, що втрата інформації при передачі даних таким шляхом є мінімально. Обусловлюється це тим, що співробітникам, що знаходяться на одному службовому рівні, психологічно легше зрозуміти один одного, так як вони вирішують однотипні задачі і стикаються зі схожими проблемами.

Вертикальні низхідні інформаційні потоки

Даний вид інформаційних потоків може бути як формальним, так і неформальним. З боку їх комунікативної ефективності, ситуація виглядає наступним чином: чим більше передавальних ланок проходить інформація, тим більше вона втрачається і видозмінюється. Протягом передачі інформації даним шляхом відбувається процес спотворення інформації. На практиці менеджер повинен розуміти, що кожна передавальна ланка спотворює до 50% інформації, що поступає. Парадокс даного виду інформаційного потоку в тому, що інформація, яка приходить «зверху» підприємства, не приховується і не спотворюється ким-то навмисно. Повноті передачі інформації перешкоджають комунікабельні бар'єри, тобто у низхідних інформаційних потоках спостерігається ефект «зіпсованого телефону».

Крім прямого збитку виробничої діяльності неорганізовані, неефективно налагоджені низхідні інформаційні потоки можуть призвести до різноманітних психологічних наслідків у всього персоналу:

1) Викликати у підлеглих стійке почуття «гвинтика», їх необхідність повазі та визнанні не буде задовільним;

2) Породити плітки, якими завжди буде заповнюватись інформаційний вакуум;

3) Викликати почуття страху, невпевненості у наступному дні;

4) Призвести до зменшення мотивації до роботи.

5) Вертикальні визхідні інформаційні потоки

Даний вид інформаційних потоків дуже рідко буває неформальним. Спотворення інформації в цьому потоці може досягти 90%. При цьому інформація, що в ньому зберігається, меньше всього аналізується.

Для покращення визхідних потоків можно використовувати:

1) Систему дій, визначених терміном «політика відкритих дверей». Ця політика визначає готовність керівника будь-якого положення в ієрархії підприємства вислухати пропозиції рядових співробітників. Для запобігання перевищення використання робочого часу керівника та упорядкувати контакти керівника та підлеглих:

– керівник самостійно визначає вигідний для нього час для зустрічей з підлеглим, таким чином не відмовляє співробітникам в аудієнції;

– використання писемної форми викладення інформації.

Система дій, що має назву «виведення управління за межі кабінету». В цьому випадку керівництво практикує обхід всього підприємства при цьому паралельно вирішує усі виробничі процеси.

При поєднаному використанні цих двох тактик, ефективність визхідних інформаційних потоків підвищується приблизно на 40%.

Можна виділити ще один вид інформаційних потоків, який можна назвати зовнішнім інформаційним потоком. Зовнішній інформаційний потік - це інформація, яка надходить у підприємство ззовні і йде з фірми зовні. Такий процес відбувається тому, що жодна соціальна система не може існувати без обміну інформацією з зовнішнім світом. І такі потоки теж мають специфічні особливості. По-перше, їх майже неможливо контролювати, тому що складно визначити яка

інформація: важлива або не важлива, негативна чи позитивна, секретна або відкрита - йде з організації. По - друге, зовнішній інформаційний потік майже не піддається свідомому регулюванню. Єдиний спосіб управління ним - використовувати такий потік для створення "образу підприємства", його іміджу в очах громадської думки. Потрібно зробити так, щоб в засобах масової інформації регулярно з'являлися матеріали про фірму, що мають вигідний для вас характер. Якщо фірма велика, необхідно організувати відділ зі зв'язків з громадськістю, або прес-відділ. Якщо фірма невелика, цим може займатися керівник. Не потрібно нехтувати можливістю інформувати суспільство про те, що справи організації йдуть прекрасно.

З функціональної точки зору інформаційний потік можна представити у наступному вигляді. Для прикладу взято процес розробки програми діяльності підприємства на майбутній період.

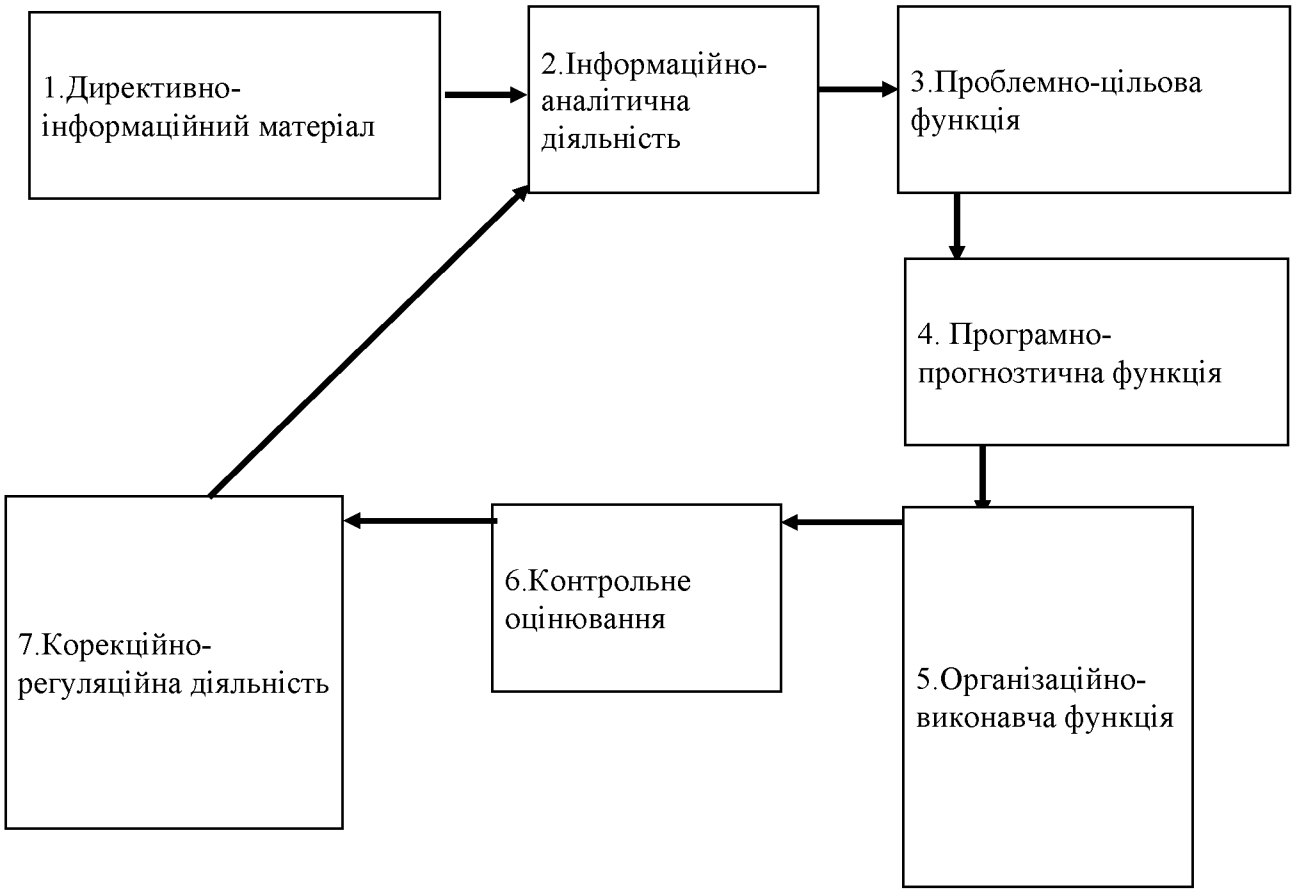

Рисунок 1.2 – Інформаційні потоки

1.6 Персонал, що взаємодіє з адміністративною частиною сайту та його права доступу до основних сутностей, що проводять процеси сайту в дію

Серед усього персоналу підприємства взаємодіє саме з адміністративною частиною сайту:

- адміністратор;
- суперадміністратор;
- контент-менеджер;
- сео-менеджер;
- сейлз-менеджер.

Перелік сутностей до яких адміністративний персонал может мати доступ:

- книги;
- автори книг;
- категорії книг;
- детальні фото книг;
- відгуки користувачів;
- адреси користувачів (білінгова та шипінгова адреса);
- облікові записи користувачів;
- персональні дані користувачів;
- платіжна інформація користувачів;
- замовлення користувачів;
- адміністратори.

Таблиця 1.2 - Перелік прав доступу

Умовне позначення права доступу	Пояснення
C	Право на створення
R	Право на читання
U	Право на змінення (запис)
D	Право на видалення

Середовище користувачів:

Таблиця 1.3 - Адміністратор та його права

Назва сутності	Права доступу
Книга	RD
Автор книги	RD
Категорія книги	RD
Детальне фото книги	RD
Адреса користувача	R
Обліковий запис користувача	RU
Відгуки користувачів	RUD
Персональні дані користувачів	RU
Платіжна інформація користувачів	R
Замовлення користувачів	R
Адміністратори	R

Таблиця 1.4 - Суперадміністратор та його права

Назва сутності	Права доступу
Книга	RD
Автор книги	RD
Категорія книги	RD
Детальне фото книги	RD
Адреса користувача	R
Обліковий запис користувача	RU
Відгуки користувачів	RUD
Персональні дані користувачів	RU
Платіжна інформація користувачів	R
Замовлення користувачів	R
Адміністратори	CRUD

Таблиця 1.5 - Контент-менеджер та його права

Назва сутності	Права доступу
Книга	CRUD
Автор книги	CRUD
Категорія книги	CRUD
Детальне фото книги	CRUD
Адреса користувача	-
Обліковий запис користувача	-
Відгуки користувачів	R
Персональні дані користувачів	-
Платіжна інформація користувачів	-
Замовлення користувачів	R
Адміністратори	R

Таблиця 1.6 - Сео-менеджер та його права

Назва сутності	Права доступу
Книга	RU
Автор книги	R
Категорія книги	RU
Детальне фото книги	RU
Адреса користувача	-
Обліковий запис користувача	-
Відгуки користувачів	R
Персональні дані користувачів	-
Платіжна інформація користувачів	-
Замовлення користувачів	R
Адміністратори	R

Таблиця 1.7 - Сейлз-менеджер та його права

Назва сутності	Права доступу
Книга	R
Автор книги	R
Категорія книги	R
Детальне фото книги	R
Адреса користувача	R
Обліковий запис користувача	R
Відгуки користувачів	R
Персональні дані користувачів	R
Платіжна інформація користувачів	R
Замовлення користувачів	RUD
Адміністратори	R

Окрему роль на сайті має звичайний користувач. Користувача можна розділити на два типи:

– авторизований користувач;

– гість.

Будь-який користувач має можливість переглядати основні сторінки сайту:

1) головна сторінка;

2) сторінка списку усіх книг;

3) сторінка окремої книги;

4) сторінка списку книг окремої категорії;

5) переглядати відгуки до книг.

Додатково тільки авторизований користувач має право на такі дії:

1) Додавати книгу до електронного кошика;

2) Переглядати тільки власний кошик;

3) Залишати відгуки на книги;

4) Переглядати налаштування персонального облікового запису на сайті;

5) Редагувати дані персонального облікового запису на сайті;

6) Створювати замовлення;

7) Переглядати особисті замовлення та їх статус.

1.7 Висновок

В першому розділі кваліфікаційної роботи виконано аналіз загального стану питання та основного опису WEB-сторінки, яка розглядається як об'єкт інформаційної діяльності. Були розглянуті усі потрібні аспекти інформаційної діяльності, а саме:

– інформація, що зберігається на сайті і під яку потрібно розробляти політику безпеки від несанкціонованого доступу;

– програмна та апаратна частина WEB-сторінки, налаштування і конфігурація якої впливає на реалізацію політики захисту;

– права і ролі доступу персоналу до ресурсів розміщених на сайті, який приймає участь у роботі адміністративної частини WEB-сторінки.

Відштовхуючись від усієї проаналізованої інформації буде надалі проведений аналіз загроз та складені модель загроз і порушника, складений профіль захищеності WEB-сторінки.

РОЗДІЛ 2 СПЕЦІАЛЬНА ЧАСТИНА

2.1 Загальні відомості

В даному розділі на прикладі WEB-сторінки, що розглядається буде:

- проведений аналіз загроз;
- складена модель загроз та порушника;
- проведене тестування на реалізацію знайдених загроз;
- складений профіль захищеності Т2 системи посилаючись на «Вимоги до захисту інформації WEB-сторінки від несанкціонованого доступу» НД ТЗІ 2.5-010;
- проведена реалізація системи захисту;
- повторне тестування на реалізацію визначених загроз.

2.2 Модель порушника

Модель порушника - це формальний або неформальний опис дій порушника, який відображає його практичні та теоретичні можливості, його знання, час і місце дії можливого порушення. Як порушника ми розглядаємо особу, що може одержати несанкціонований доступ до інформації, що зберігається на сайті.

Модель порушника повинна визначати:

- можливі цілі порушника за ступенем небезпечності для WEB-сторінки та інформації, що потребує захисту;
- гіпотеза про кваліфікацію порушника;
- умовивід про характер дій можливого порушника;

Метою порушника можуть бути:

- отримання необхідної інформації у потрібному обсязі в подальшому для використання в особистих цілях;
- здобути можливість вносити зміни в інформаційні потоки у відповідності зі своїми намірами;
- нанесення збитків шляхом знищення інформаційних цінностей сайту.

Базово можливі порушники поділяються на дві основі групи: зовнішні та внутрішні.

До зовнішніх можливих порушників WEB-сторінки, що розглядається відноситься будь-який користувач сайту (гість або авторизований користувач), який може намагатися підібрати паролі до облікових записів інших користувачів або намагатися провести SQL/CSS-ін'єкції та інші види відомих атак.

До внутрішніх можливих порушників WEB-сторінки, що розглядається відносяться будь-який персонал адміністративної частини сайту, так як майже усі ролі адміністративного персоналу мають доступ до читання майже усіх даних, що зберігаються на сайті. Також до внутрішніх порушників відносяться активні в даний час розробники сайту (сайт на даний час вже є працюючим, тому розробники приймають участь у програмній підтримці додатку), які мають права суперадміністраторів та єдині особи, які мають доступ до повної програмної і серверної частини WEB-сторінки включаючи доступ до docker-контейнерів проекту, що знаходяться на інстансу (повний обсяг даних додатку і всіх його частин на віддаленому сервері) через мережевий протокол SSH. Окремо потрібно виділити тестувальників, які у цілях тестування повинні перевірити весь обсяг функціоналу усіх користувачів на всіх сторінках та оболонках додатку, маючи при цьому права одночасно всіх ролей, так як функціонал адміністративної частини сайту також потребує перевірки на наявність будь-яких помилок та розбіжностей у специфікації та актуального функціоналу сайту.

2.2.1 Специфікація моделі порушника

2.2.1.1 Специфікація моделі порушника за мотивом здійснення можливого порушення

– M1 - Безвідповідальність (не розуміння наслідків під час скоєння будь-якого порушення)

– M2 - Самоствердження (порушник намагається довести собі або оточуючим чого він/вона вартий)

– М3 - Корисливий мотив (метою порушення виступає особиста невідома вигода)

2.2.1.2 Специфікація моделі порушника за рівнем кваліфікації та обізнаності

– К0 - Не знає функціональної особливості системи, основні закономірності формування даних та запитів, які приходять до WEB-сторінки;

– К1 - Знає функціональні особливості системи, основні закономірності формування даних та запитів, які приходять до WEB-сторінки;

– К2 - Володіє високим рівнем обізнаності основних технічних і програмних засобів роботи WEB-сторінки, слабке розуміння параметрів, що працюють із запитами, що приходять до WEB-сайту;

– К3 - Володіє усією інформацією, що пов'язана з роботою WEB-сторінки, запитами, які приходять до WEB-сайту, параметрами і функціональною частиною додатку.

2.2.1.3 Специфікація моделі порушника за місцем дії

Дана специфікація моделі порушника не є актуальною для даного виду «об'єкту» ІТС, так як це WEB-сайт. Доступ до WEB-сайту доступний будь-якому користувачу з будь-якого девайсу і будь-якої частини світу через вихід до Інтернету. Доступ до адміністративної частини сайту надається лише авторизованим адміністраторам та їх суміжним ролям також через вихід до Інтернету з будь-якої частини світу, з будь-якого девайсу. Аналогічна ситуація є і з доступом docker-контейнерів в актуальних розробників сайту.

2.2.1.4 Специфікація моделі порушника за часом дії:

– Ч1 - у робочий час під час робочого тижня, коли реагування на будь-яку атаку найшвидше;

– Ч2 - на вихідних, коли реагування на будь-яку атаку буде менш швидким;

– Ч3 - у неробочий час (вночі) у робочі дні або на вихідних, коли реагування на будь-яку атаку буде найменш швидким.

Реагування на атаку на сайті проводиться через логування будь-яких дій на сервері, базі даних. Якщо під час обробки запиту на сервер або до бази даних виникає помилка або проводиться спроба будь-яких із відомих атак, сервер відправляє нотифікацію (повідомлення) до закритого каналу сповіщень корпоративного мессенджера Slack з текстом про помилку або іншу видиму проблему у додатку та посиланням на зовнішній ресурс Sentry. Цей ресурс спеціально налаштований для подібних проблем, в ньому розкриваються усі помилки або проблеми, які виникли при роботі сайту з повним розширенням і текстом (навіть з номер рядку у коді, де виникла помилка, якщо помилка виникла на сервері). Також при виникненні подібних ситуацій одночасно відправляється таке саме повідомлення усім адміністраторам, суперадміністраторам та розробникам сайту.

Так як користування сайтом може проходити у будь-який час та звідусіль, найбільш вразливим часом буде проміжок з 10 години вечора по 8 ранку, коли реагування на повідомлення з логуванням буде найменш успішним і результативним через людський фактор.

2.2.1.5 Специфікація моделі порушника за показником можливості подолання системи захисту інстансу

Для того, щоб отримати доступ до середовища додатку на віддаленому сервері і взаємодіяти з docker-контейнерами WEB-сайту використовується мережевий протокол SSH. Для встановлення зв’язку клієнт-сервер є два варіанти:

– клієнт має збережений на своїй локальній машині приватний ключ сервера, до якого клієнт намагається під’єднатись з допомогою протоколу SSH;

– публічний ключ клієнта записаний в список дозволених хостів на віддаленному сервері, до якого клієнт намагається під’єднатись з використанням SSH.

Посилаючись на такі дані можна визначити такі специфікації моделі порушника за даним типом:

– З0 - не має збереженого на локальній машині приватного ключа сервера і на сервері в дозволених хостах публічний ключ даного клієнта – відсутній;

– З1 - має збережений на локальній машині приватний ключ сервера або на сервері в дозволених хостах записаний публічний ключ даного клієнта.

2.2.1.6 Модель порушника зовнішнього типу

Таблиця 2.1 - Модель порушника зовнішнього типу

Роль по відношенню до сайту	Мотив	Кваліфікація	Можливість обійти захист	Час дії	Сума загроз
Гість	М1	К0	З0	Ч3	4
Авторизований користувач	М2	К0	З0	Ч3	5
Конкурент	М3	К2	З0	Ч1	6
Хакер	М3	К3	З0	Ч3	9

2.2.1.7 Модель порушника внутрішнього типу

Таблиця 2.2 - Модель порушника внутрішнього типу

Роль по відношенню до сайту	Мотив	Кваліфікація	Можливість обійти захист	Час дії	Сума загроз
Суперадміністратор	М3	К3	З1	Ч3	10
Адміністратор	М2	К2	З0	Ч2	6
Сео-менеджер	М2	К1	З0	Ч1	4
Сейлз-менеджер	М2	К1	З0	Ч1	4
Контент-менеджер	М2	К1	З0	Ч1	4

Окремими моделями порушника виступають актуальні розробники і тестувальники сайту. По оцінкам модель порушника для цих ролей:

– розробник - має роль суперадміністратора, тому сума загроз для цієї моделі дорівнює 10;

– тестувальник - має одночасно всі ролі (переназначаются кожну ітерацію тестування), максимальну суму загроз має роль суперадміністратора, тому прирівнюємо дану оцінку і тестувальнику (10).

2.3 Модель загроз та вразливостей

2.3.1 Властивості інформації

– К - Конфіденційність - властивість інформації, що гарантує доступ до інформації лише авторизованим особам [7];

– Ц - Цілісність - властивість інформації, що гарантує можливість модифікації інформації лише авторизованим особам [7];

– Д - Доступність - властивість інформації, що гарантує доступ до інформації лише авторизованим користувачам не очікуючи довше заданого інтервалу часу [7].

Таблиця 2.3 - Коефіцієнти можливості реалізації загрози

Коефіцієнт реалізації загрози	Опис
1	Майже неможливо
2	Малоймовірно
3	Можливо, але недоцільно
4	Можливо та доцільно
5	Висока ймовірність

Таблиця 2.4 - Оцінка критичності наслідків реалізації загрози

Коефіцієнт реалізації загрози	Опис
1	Не критичні
2	Низька критичність
3	Середня критичність
4	Висока критичність
5	Неймовірна висока критичність

2.3.2 Перелік загроз

Таблиця 2.5 - Загрози з визначенням порушень властивостей інформації

Опис загрози	Джерело загрози	Наслідки	Порушення	Коефіцієнт можливості реалізації	Критичність наслідків	Оцінка загрози
Підбір приватного ключа від інстансу	Зовнішнє	Порушник має доступ до всіх docker-контейнерів. Порушник може змінити налаштування доступів для інших адміністративних користувачів	К,Ц,Д	1	5	6

Продовження таблиці 2.5

Опис загрози	Джерело загрози	Наслідки	Порушення	Коефіцієнт можливості реалізації	Критичність наслідків	Оцінка загрози
Підбір логіну та паролю одного із авторизованих користувачів	Зовнішнє	Порушник має доступ до одного із облікових записів сайту (персональні дані, платіжна інформація). Сайт втрачає клієнта і репутація, як сайту, що є безпечним.	К	1	3	4
Навмисна передача конфіденційних даних адміністративної частини сайту або особистих даних користувачів	Внутрішнє	Фінансові втрати як у підприємства так і користувачів сайту	К,Ц	4	5	9
Випадкове видалення деяких даних з адміністративної частини сайту	Внутрішнє	Тимчасова втрата існуючого товару або замовлень клієнтів з даних сайту	Д,Ц	3	2	5

Перехоплення даних під час передавання	Зовнішнє	Дані, що передаються під час запитів до сайту можуть бути перехоплені	К.Ц	1	4	5

Продовження таблиці 2.5

Опис загрози	Джерело загрози	Наслідки	Порушення	Коефіцієнт можливості реалізації	Критичність наслідків	Оцінка загрози
Витік інформації	Внутрішнє	Через відсутність ретельного тестування дорелізного продукту, до production середовища потрапляє функціонал з витоком даних, що може призвести до колосальних збитків	К,Ц.Д	2	5	7
Міжсайтове скриптування	Зовнішнє	Через несвоєчасу очистку даних з прийманих запитів, можно проводити підбір даних для заповнення форми або використовувати дану атаку для спаму	К.Ц	2	4	6

Посилаючись на зібрані дані, щодо загроз, які можуть бути реалізовані на сайті, найбільш критичною загрозою є навмисна передача адміністративних да-

них додатку третім особам з боку персоналу з адміністративними правами. Для запобігання і зменшення ризику реалізації цієї загрози потрібно більш чітко розмежувати права доступу між ролями адміністративного персоналу, так як більша частина всього персоналу має право на читання усіх даних, що зберігаються на сайті.

Для запобігання передачі приватного ключа інстансу третім особам від розробників, найкращим варіантом буде - не використання приватного ключа для доступа до інстансу через мережевий протокол SSH. Замість цього на ісінтансі додати в список хостів додати публічні ключі всіх авторизованих членів персоналу, яким дозволений вхід до інстансу через SSH.

2.4 Тестування WEB-сторінки на програмні вразливості за спеціальних гемів, таких як brakeman та bundle audit

Перевірка на вразливості в коді з brakeman

```
== Warnings ==

Confidence: Medium
Category: SQL Injection
Check: SQL
Message: Possible SQL injection
Code: ActiveRecord::Base.connection.execute("DELETE FROM cart_items WHERE cart_items.id = #{id}")
File: app/services/cart_item_service.rb
Line: 14

Confidence: Weak
Category: Dynamic Render Path
Check: Render
Message: Render path contains parameter value
Code: render(partial => "checkout/#{CheckoutService.new(current_order, current_user, params).current_step}/checkout_#{CheckoutService.new(current_order, curre
nt_user, params).current_step}", { :locals => ({ :service => CheckoutService.new(current_order, current_user, params).current_service, :price_service => Price
Service.new(current_order) }) })
File: app/views/checkout/_current_step.html.haml
Line: 1
```

Рисунок 2.1 – Реузльтат перевірки brakeman

Як видно на рисунку 2.1 у додатку є дві вразливості:

– середня по критичності вразливість, пов’язана з можливим проведенням SQL ін’єкції, яку можливо реалізувати при видаленні конкретного пункту електронного кошику. Причиною є передача унікального ідентифікатору (id) пункту кошику напряму у строгий SQL-запит через інтерполяцію рядку, що є небезпечним через свою динамічність. Для вирішення і захисту від даної вразливості треба виключити випадки передачі параметрів у строгий SQL-запит через інтерполяцію рядків. Вирішенням даної проблеми в цій конкретній ситуації буде: переписати строгий SQL запит на rails допоміжні методи роботи з моделлю (оболонка між

абстрактною сутністю та записом у базі даних), а саме ActiveRecord методи, в який буде переданий тільки унікальний ідентифікатор сутності, що видаляється, а вже «під коробкою» метода буде виконаний потрібний SQL-запит на видалення запису з бази даних;

– мінімальна по критичності вразливість пов'язана з рендерингом (візуалізацією) відповіді сервера у вигляді певної HTML-сторінки, а саме динамічна передача параметрів для рендерінгу сторінки. У цьому випадку назва файлу з розширенням html передається в параметрах з контроллеру і передається в html відповідь і передається для рендеренгу допоміжної сторінки. Для вирішення і захисту від даної вразливості є два варіанти. Першим виступає відмова від додаткової оболонки рендеренгу допоміжної сторінки, але це рішення не є доцільним для даного випадку та проекту, тому що ускладнює можливість розмежування відповідальності і надання одній сторінці відповідати лише за відображення одного випадку бізнес-логіки. Другим варіантом вирішення даної проблеми є передача назви файлу з html розширенням не через локальні параметри, а через окремий блок collection (колекція) і передавати назву файлу як єдиний елемент масиву, що повністю вирішує проблеми з динамічним рендерингом сторінки. Цей спосіб є найбільш релевантним для вирішення даної вразливості.

Перевірка на вразливості в залежностях інших гемів з bundle audit

На рисунку В.1 - Реультат тестування bundle audit 1 зображені вразливості пов'язані з гемами actionpack, activerecord та carrierwave з вказанням середньої та невідомої критичності (criticality) загрози, актуальної версії гемів, ідентифікатора цієї вразливості (advisory), по якому можна проводити пошук ручних вирішень даної вразливості або вказати гему цей ідентифікатор для ігнорування конкретної вразливості, базове вирішення вразливості.

На рисунку В.2 - Реультат тестування bundle audit 2 зображені вразливості пов'язані з гемом nokigiri та carrierwave з позначенням високої критичності вразливості.

На рисунку В.3 - Реультат тестування bundle audit 2 зображені вразливості високої критичності пов'язані з гемом nokogiri та puma.

Зробивши висновки з результатів аналізу bundle audit, в додатку присутня чимала кількість вразливостей від залежностей допоміжних або основних гемів. Основним і єдиним рішенням буде оновити геми до потрібних версій, але для якісного виправлення даної вразливості потрібно притримуватись правильного алгоритму і розуміння від яких гемів можуть залежати вказані і від цього буде залежати, що і як буде оновлюватись.

Для виправлення перших п'яти вразливостей показаних на рисунку В.1 потрібно оновити не геми actionpack і activerecord, що вказані, а саме гем rails до вказаної або до найновішої версії.

Для виправлення останньої залежності на рисунку В.1 та усіх вразливостей з рисунку В.2 достатньо оновити вказані геми до вказаних рекомендованих версій або до найновіших версій, але ці дії потрібно обов'язково проводити тільки після оновлення основного гему rails (тільки в даному випадку, так як rails потребує оновлення, якщо такого не потребується, оновлювати одразу потрібні геми).

Для виправлення усіх залежностей, які вказані на рисунку В.3, окрім залежності omniauth, достатньо просто оновити позначені геми до вказаних версій або більш нових (попередньо оновивши rails, бо цього потребує дана ситуація).

Щодо залежності omniauth, нажаль, цю залежність виправити в даний час неможливо, тому що версія, яка вказана на рисунку В.3 не є лігитивною в даний час. Оновити цей гем до потрібної версії тільки тоді, коли розробник і власник гему випустить остаточний генеральний реліз із вказанням цієї версії. Тому на даний момент попередження про цю вразливість можна ігнорувати.

2.5 Повторне тестування WEB-сторінки на програмні вразливості за спеціальних гемів, таких як brakeman та bundle audit

Після отримання рекомендацій для виправлення усіх вразливостей, що були виявлені під час аналізу програмної частини сайту з допомогою гемів brakeman та bundle audit, були виконані рекомендовані дії, а саме:

– виключили можливість проведення SQL-ін'єкції переписавши строгий SQL запит на rails допоміжний метод роботи з моделлю, а саме ActiveRecord метод, в який буде переданий тільки унікальний ідентифікатор сутності, що видаляється, а вже «під коробкою» метода буде виконаний потрібний SQL-запит на видалення запису з бази даних;

– припинена динамічна передача назви файлу з html розширенням у локальних параметрах для рендерінгу сторінки. Замість цього назва файлу передається в параметрах неначе колекція, що запобігає вразливості, що виникла;

– виконані всі рекомендовані дії і алгоритми для оновлення гемів для виправлення вразливостей у залежностях бібліотек, що використовуються у програмній і серверній частині сайту.

Результат повторного тестування з гемом brakeman після виправлення всіх попереджень:

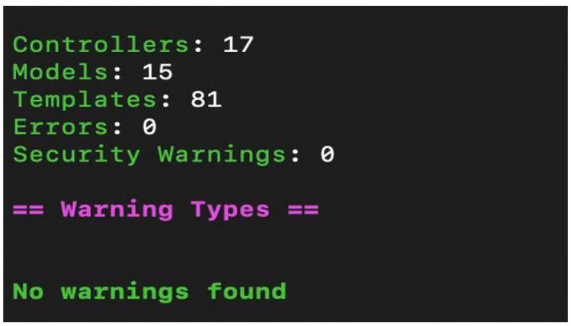

Рисунок 2.5 – Повторна перевірка brakeman

Результат повторного тестування з гемом bundle audit:

Рисунок 2.6 – Повторна перевірка bundle audit

Як видно по результатам обох перевірок (рисунок 2.5 та рисунок 2.6), всі вразливості були виправлені, шляхом введення всіх рекомендацій гемів, які перевіряли програмну частину сайту на вразливості.

2.6 Профіль захищеності

Для обраного об’єкту з використанням технології Т2 посилаючись на [8] профіль захищенності має вигляд:

{КА-2, КВ-1, ЦА-1, ЦО-1, ЦВ-1, ДВ-1, ДР-1, НР-2, НИ-2, НК-1, НО-1, НЦ-1, НТ-1, НВ-1}

КА-2 (Базова адміністративна конфіденційність) - реалізована за допомогою програмних засобів гемів Pundit, Devise та ActiveAdmin

Ця послуга дозволяє адміністратору безпеки керувати потоками інформації від захищених об'єктів до користувачів.

КЗЗ повинен здійснювати розмежування доступу на підставі атрибутів доступу користувача і захищеного об’єкта. Доступ до загальнодоступної інформації встановлюється для користувачів усіх категорій. Призначення атрибутів доступу користувачам і процесам до захищених об'єктів здійснюється адміністратором безпеки на основі аналізу функціональних та службових обов'язків окремих користувачів.

КВ-1 (Конфіденційність при обміні) - реалізована з допомогою програмних засобів гемів Pundit та Devise

Ця послуга дозволяє забезпечити захист об’єктів від несанкціонованого ознайомлення з інформацією, що міститься в них, під час їх експорту/імпорту через незахищене середовище.

ЦА-1 (Мінімальна адміністративна цілісність) - реалізована з допомогою програмних засобів

Ця послуга дозволяє керувати потоками інформації від користувачів до захищених об’єктів WEB-сторінки.

Політика мінімальної адміністративної цілісності стосується: користувачів усіх категорій; загальнодоступної інформації WEB-сторінки; файлової системи та

функціонального ПЗ, що використовується для актуалізації, захисту загальнодоступної інформації та супроводження WEB- сторінки; створеної в процесі супроводження WEB-сторінки технологічної інформації КСЗІ та технологічної інформації щодо управління АС.

КЗЗ повинен здійснювати розмежування доступу на підставі атрибутів доступу користувачів і захищених об'єктів. Розмежування доступу здійснюється на рівні надання (встановлення заборони) користувачеві прав модифікувати об'єкт.

ЦО-1 (Відкат) - реалізовано

Можливість відновлення даних у базі даних при втраті надається через backup-версій усіх записів бази даних. Можливість відновлення структури таблиць у базі даних надається через властивості накатування і відкатування міграцій. Можливість відкату об'єкту до пвеного стану надається через базові властивості і команди гілкування та відкату змін системи контролю версій git.

Ця послуга забезпечує можливість відмінити окрему операцію або послідовність операцій і повернути захищений об'єкт після внесення до нього змін до попереднього наперед визначеного стану.

Факт використання послуги має реєструватись в системному журналі. Відміна операції не повинна призводити до видалення з журналу запису про операцію, яка пізніше була відмінена, якщо остання підлягала реєстрації відповідно до вимог послуги безпеки НР-2.

ЦВ-1 (Мінімальна цілісність при обміні) - частково реалізована

Ця послуга дозволяє забезпечити захист WEB-сторінки від несанкціонованої модифікації інформації, яка передається між WEB-сервером та робочими станціями у разі використання технології Т2, під час експорту/імпорту інформації через незахищене середовище. Політика послуги стосується всіх об'єктів, що передаються.

КЗЗ повинен забезпечувати контроль за цілісністю інформації в повідомленнях, які передаються, а також бути здатним виявляти факти їх несанкціонованого видалення або дублювання.

КЗЗ повинен забезпечувати можливість реєстрації подій, які призвели до порушення цілісності повідомлень, їх несанкціонованого видалення або дублювання.

ДВ-1 (Відновлення після збоїв) - реалізована з допомогою backup-версій бази даних і основних можливостей гілкування та відкату змін через систему контролю версій git.

Політика відновлення після збоїв, що реалізується КЗЗ, стосується: системного та функціонального програмного забезпечення; засобів захисту інформації та засобів управління КСЗІ; засобів адміністрування та управління обчислювальною системою АС – і гарантує повернення АС у відомий захищений стан після відмов або переривання обслуговування, спричинених помилковими діями користувачів, неврахованою функціональною недостатністю програмного та апаратного забезпечення (наприклад, можливою наявністю не виявлених під час проектування незадекларованих функцій), іншими непередбачуваними ситуаціями.

Політика відновлення, яка реалізується КЗЗ, повинна визначати множину типів відмов WEB-сторінки і переривань обслуговування, після яких можливе повернення у відомий захищений стан без порушення політики безпеки. Для кожної з відмов повинні бути чітко визначені і задокументовані рівні відмов, у разі перевищення яких необхідна повторна інсталяція WEB-сторінки.

ДР-1 (Використання ресурсів) - реалізована з допомогою програмних засобів Docker

Ця послуга дозволяє керувати використанням користувачами послуг та ресурсів.

Політика використання ресурсів, що реалізується КЗЗ, стосується: користувачів загальнодоступної інформації; адміністратора безпеки та користувачів, яким надано повноваження щодо управління АС; файлової системи; системного та функціонального програмного забезпечення; технологічної інформації щодо управління АС; окремих периферійних пристроїв (принтерів, накопичувачів інформації і т.ін.); обчислювальних ресурсів АС і передбачає можливість встановлення обмежень на їх використання.

Обмеження щодо використання окремим користувачем та/або процесом обсягів обчислювальних ресурсів АС або кількості об’єктів встановлюються адміністратором безпеки або користувачами, яким надано повноваження щодо управління АС. Запити на зміну встановлених обмежень повинні оброблятися КЗЗ тільки в тому випадку, якщо вони надходять від зазначених користувачів.

Спроби користувачів перевищити встановлені обмеження на використання ресурсів повинні реєструватися в системному журналі.

НР-2 (Реєстрація) - реалізовано

Послуга дозволяє контролювати небезпечні відповідно до політики безпеки WEB-сторінки дії користувачів всіх категорій із захищеними об’єктами.

Політика реєстрації стосується: користувачів усіх категорій; публічної інформації WEB-сторінки; системного та функціонального програмного забезпечення, що використовується для актуалізації, захисту публічної інформації та супроводження WEB-сторінки; створеної в процесі супроводження WEB-сторінки технологічної інформації КСЗІ та технологічної інформації щодо управління АС.

КЗЗ повинен забезпечувати реєстрацію всіх подій, які мають безпосереднє відношення до безпеки. До них відносяться наступні класи подій:

– вхід/вихід або намагання входу/виходу в/із системи користувачами будь-яких категорій;

– реєстрація та видалення або намагання реєстрації та видалення користувачів будь-якої категорії в системі;

– зміна атрибутів доступу користувачем будь-якої категорії та дії, що призвели до цього;

– отримання або намагання отримання доступу користувачем будь-якої категорії до будь-яких захищених процесів і об'єктів АС;

– створення користувачем будь-якої категорії твердих копій та виведення їх на друкуючі пристрої;

– модифікація або спроби модифікації захищених процесів і об'єктів АС, у тому числі факти та спроби порушення цілісності КЗЗ;

– спроби використання обчислювальних ресурсів АС з перевищенням встановлених квот;

– інші події, обов'язковість реєстрації яких передбачена політикою реалізації окремих послуг безпеки інформації.

КЗЗ повинен надавати можливість визначення переліку реєстраційних подій виключно адміністратору безпеки.

Реєстрація всіх подій, що мають безпосереднє відношення до безпеки, здійснюється в журналі реєстрації, який повинен містити інформацію стосовно дати, часу, місця, типу і наслідків зареєстрованої події (успішність/неуспішність), ім'я (IP-адресу) та/або ідентифікатор причетного до цієї події користувача. Реєстраційна інформація повинна бути достатньою для однозначної ідентифікації користувача, процесу і/або об'єкта, що мали відношення до кожної зареєстрованої події.

НИ-2 (Ідентифікація і автентифікація) - реалізовано з допомогою програмних засобів гемів Devise та Pundit

Ідентифікація і автентифікація дозволяють КЗЗ визначити і перевірити особу суб'єкта, що намагається одержати доступ до захищених об'єктів WEB-сторінки.

Політика ідентифікації і автентифікації стосується: всіх користувачів WEB-сторінки, які намагаються одержати доступ до системного та функціонального програмного забезпечення, що використовується для актуалізації, захисту публічної інформації та супроводження WEB-сторінки; створеної в процесі супроводження WEB-сторінки технологічної інформації КСЗІ та технологічної інформації щодо управління АС; задіяного для цього периферійного обладнання.

НК-1 (Достовірний канал) - реалізовано

Ця послуга повинна гарантувати користувачу будь-якої категорії можливість безпосередньої взаємодії з КЗЗ, а також те, що ніяка взаємодія користувача з АС не може бути модифікованою іншим користувачем або процесом. Послуга визначає вимоги до механізму встановлення достовірного зв'язку між користувачем і КЗЗ.

Політика достовірного каналу стосується користувачів усіх категорій та компонентів системного та функціонального програмного забезпечення, які задіяні для реалізації механізмів КЗЗ.

Достовірний канал повинен використовуватися для початкової ідентифікації і автентифікації. Зв'язок з використанням даного каналу повинен ініціюватися виключно користувачем.

НО-1 (Розподіл обов'язків) - реалізовано з допомогою програмних можливостей Pundit та ActiveAdmin

Ця послуга дозволяє розмежувати повноваження користувачів, визначивши категорії користувачів з певними і притаманними для кожної з категорій функціями (ролі). Послуга призначена для зменшення потенційних збитків від навмисних або помилкових дій користувачів і обмеження авторитарності керування АС.

Політика розподілу обов'язків, що реалізується КЗЗ, стосується користувачів усіх категорій і повинна визначати щонайменше такі ролі:

- адміністратора безпеки;

- користувачів, яким надано право доступу до певних видів інформації (публічної, технологічної, системного та функціонального ПЗ).

Кількість користувачів, які мають доступ до технологічної інформації та системного і функціонального ПЗ повинна бути мінімізована, щоб обмежити їх коло тільки тими, кому необхідний такий доступ для виконання функціональних обов'язків, що передбачаються експлуатаційною та розпорядчою документацією на WEB-сторінку.

Адміністратору безпеки дозволяється доступ до всієї інформації WEB-сторінки. У разі необхідності його роль може дублюватися уповноваженим співробітником СЗІ. Повноваження всіх інших користувачів щодо доступу до інформації надаються їм адміністратором безпеки.

НЦ-1 (Цілісність комплексу засобів захисту) - частково реалізовано

Ця послуга визначає міру здатності КЗЗ WEB-сторінки захищати себе і гарантувати свою спроможність керувати захищеними об'єктами.

Політика цілісності КЗЗ повинна визначати склад КЗЗ, механізми контролю цілісності його компонентів та порядок їх використання.

Політика цілісності КЗЗ стосується: адміністратора безпеки; окремих компонентів системного та функціонального програмного забезпечення, які задіяні для реалізації механізмів КЗЗ; засобів захисту інформації, а також технологічної інформації КСЗІ - і забезпечує взаємодію зазначених об’єктів.

Політика реалізації послуги повинна гарантувати, що всі послуги безпеки доступні тільки через інтерфейс КЗЗ і всі запити на доступ до захищених об'єктів контролюються КЗЗ. Якщо існують обмеження, недотримання яких може призвести до надання послуг в обхід інтерфейсу КЗЗ і порушення цілісності КЗЗ, то такі обмеження повинні бути описані і задокументовані. До користувачів має бути доведено порядок їх роботи з дотриманням цих обмежень, а КЗЗ повинен надавати адміністратору можливість здійснення контролю за цим порядком.

НТ-1 (Самотестування) - реалізовно з допомогою програмних засобів фреймворку для тестування RSpec та всіма гемами-аналізаторами, лінтерами, які самостійно запускаються перед розгортанням додатку на віддаленому сервері з допомогою неперервної інтеграції CircleCI.

Самотестування дозволяє КЗЗ перевірити і на підставі цього гарантувати правильність функціонування і цілісність певної множини функцій захисту WEB-сторінки.

Політика самотестування поширюється на адміністратора безпеки, компоненти системного та функціонального програмного забезпечення, які задіяні для реалізації механізмів КЗЗ, засоби захисту інформації.

У склад КЗЗ повинна входити множина тестових процедур, яка враховує особливості функціонування компонентів конкретної WEB-сторінки і достатня для оцінки правильності виконання всіх критичних для безпеки публічної та технологічної інформації КСЗІ функцій, а сам КЗЗ повинен бути здатним контролювати їх виконання.

КЗЗ повинен забезпечувати виконання тестів за запитом адміністратора безпеки.

НВ-1 (Ідентифікація і автентифікація при обміні) - реалізовано з допомогою програмних засобів гемів Devise та Pundit

Ця послуга дозволяє у разі використання технології Т2 компонентам КЗЗ WEB-сервера і віддаленої робочої станції здійснити взаємну ідентифікацію, перш ніж розпочати взаємодію.

Послуга ідентифікації і автентифікації при обміні стосується адміністратора безпеки та користувачів, яким надані повноваження щодо супроводження WEB-сторінки, технологічної інформації КСЗІ.

КЗЗ повинен надавати доступ до процесів, що забезпечують ініціалізацію обміну даними, тільки адміністратору безпеки і користувачам, яким надано повноваження щодо супроводження WEB-сторінки.

2.7 Висновок

В другому розділі кваліфікаційної роботи були складені модель можливого порушника та модель загроз посилаючись на ролі персоналу адміністративної частини, які були розглянуті у першому розділі.

Надалі було проведене тестування для виявлення програмних вразливостей у коді. Посилаючись на знайдені вразливості був описаний алгоритм чіткого виправлення всіх попереджень. Після виправлення попереджень на виході було отримане покарщення в коді та оновленні основні геми, на яких базується робота додатку.

Надалі був складений профіль захищеності об’єкта з використанням технології Т2 та складений план реалізації політики захисту для нереалізованих функцій профілю захищеності.

РОЗДІЛ 3 ЕКОНОМІЧНИЙ РОЗДІЛ

Розробка засобів захисту інформації WEB-сторінки потребує обґрунтування економічної її доцільності, виходячи з аналізу витрат на розробку та впровадження. Тому метою економічного розділу є здійснення відповідних розрахунків, які дозволять встановити економічного ефекту від впровадження та налагодження комплексів засобів захисту інформації WEB-сторінки підприємства «RubyGlobal».

3.1 Розрахунок (фіксованих) капітальних витрат

Капітальні інвестиції – це кошти, призначені для створення і придбання основних фондів і нематеріальних активів, що підлягають амортизації.

До капітальних витрат належать витрати на розробку політики безпеки інформації, які визначаються виходячи з трудомісткості розробки політики безпеки інформації.

Визначення трудомісткості розробки політики безпеки інформації

Трудомісткість розробки політики безпеки інформації визначається тривалістю кожної робочої операції, починаючи з складання технічного завдання і закінчуючи оформленням документації (за умови роботи одного спеціаліста з інформаційної безпеки):

$$t = tmз + tв + ta + tвз + tозб + {}^{tовр} + tд, \text{ годин,}$$

де ^{tmз} – тривалість складання технічного завдання на розробку політики безпеки інформації;

^{tв} – тривалість розробки концепції безпеки інформації у організації;

^{ta} – тривалість процесу аналізу ризиків;

^{tвз} – тривалість визначення вимог до заходів, методів та засобів захисту;

^{tозб} – тривалість вибору основних рішень з забезпечення безпеки інформації;

^{tовр} – тривалість організації виконання відновлювальних робіт і забезпечення неперервного функціонування організації;

$t_д$ – тривалість документального оформлення політики безпеки.

Визначено, що відповідно до етапів розробки політики безпеки інформації, тривалість операцій склала наступні величини:

$t_{тз}$= 30 годин, $t_в$= 30 годин, $t_{тз}$= 22 годин, $t_{вз}$= 21 годин, $t_{озб}$= 12 годин, $t_{овр}$= 9 годин, $t_д$= 8 годин.

Отже, t=30+30+22+21+12+9+8= 132 години,

Розрахунок витрат на створення політики безпеки інформації

Витрати на розробку політики безпеки інформації Крп складаються з витрат на заробітну плату спеціаліста з інформаційної безпеки Ззп і вартості витрат машинного часу, що необхідний для розробки політики безпеки інформації Змч.

$$К_{рп} = З_{зп} + З_{мч}.$$

$$К_{рп} = З_{зп} + З_{мч} = 33000 + 788,04 = 33788,04 \text{ грн.}$$

$$З_{зп} = t\, З_{пр} = 132 \cdot 250 = 33000 \text{ грн.}$$

де t – загальна тривалість розробки політики безпеки, годин;

$З_{пр}$ – середньогодинна заробітна плата спеціаліста з інформаційної безпеки з нарахуваннями, грн/годину.

Вартість машинного часу для розробки політики безпеки інформації на ПК визначається за формулою:

$$З_{мч} = t \cdot С_{мч} = 132 \cdot 5,97 = 788,04 \text{ грн.}$$

де $t_д$ – трудомісткість підготовки документації на ПК, годин;

$С_{мч}$ – вартість 1 години машинного часу ПК, грн./година.

Вартість 1 години машинного часу ПК визначається за формулою:

$$Смч = 0,9 \cdot 3 \cdot 1,64 + \frac{3800 \cdot 0,4}{1920} + \frac{7200 \cdot 0,2}{1920} = 5,97 \text{ грн.}$$

Капітальні (фіксовані) витрати на створення політики інформаційної безпеки підприємства складають:

$$К = К_{рп} + К_{зпз} + К_{пз} + К_{аз} + К_{навч} + К_н = 33788,04 + 20000 + 30000 + 5000 = 88\,788,04 \text{ грн.}$$

де $К_{рп}$ – вартість розробки проекту інформаційної безпеки та залучення для цього зовнішніх консультантів, тис. грн;

$К_{зпз}$ – вартість закупівель ліцензійного основного й додаткового програмного забезпечення (ПЗ) (преміальна форма сервісу Heroku, на якому розгортається сайт), 20000 грн;

$К_{пз}$ – вартість створення основного й додаткового програмного забезпечення (сервіси ведення журналу для сповіщення у корпоративний мессенджер для реагування не загрози або атаки), 30000 грн;

$К_{аз}$ – вартість закупівлі апаратного забезпечення та допоміжних матеріалів, 0 грн;

$К_{навч}$ – витрати на навчання технічних фахівців і обслуговуючого персоналу, 5000 грн;

$К_{н}$ – витрати на встановлення обладнання та налагодження системи інформаційної безпеки, 0 грн.

Розрахунок поточних витрат

Річні поточні витрати на функціонування системи інформаційної безпеки складають:

$$С = С_{в} + С_{к} + С_{ак}, \text{ грн.}$$

де $С_{в}$ - вартість відновлення й модернізації системи ($С_{в} = 0$);

$С_{к}$ - витрати на керування системою в цілому;

$С_{ак}$ - витрати, викликані активністю користувачів системи інформаційної безпеки ($С_{ак} = 0$ грн.).

Витрати на керування системою інформаційної безпеки ($С_{к}$) складають:

$$С_{к} = С_{н} + С_{а} + С_{з} + С_{ел} + С_{о} + С_{тос}, \text{ грн.}$$

Витрати на навчання адміністративного персоналу й кінцевих користувачів визначаються ($С_{н} = 8000$ грн.).

Річні амортизаційні відрахування за обслуговування та додаткову підтримку з боку розробників складає 20000 грн із корисним строком використання 2 роки, за прямолінійним методом нарахування амортизації складуть:

$$С_а = \frac{20000}{2} = 10000 \text{ грн.}$$

Річний фонд заробітної плати програмно-технічного персоналу, що обслуговує систему інформаційної безпеки ($С_з$), складає:

$$С_з = З_{осн} + З_{дод}, \text{ грн.}$$

Основна заробітна плата визначається, виходячи з місячного посадового окладу, а додаткова заробітна плата – в розмірі 10% від основної заробітної плати.

Основна заробітна плата одного спеціаліста з інформаційної безпеки на місяць складає 22000 грн. Додаткова заробітна плата – 10% від основної заробітної плати. Отже,

$$С_з = 22000 \cdot 12 + 22000 \cdot 12 \cdot 0,1 = 264000 + 26400 = 290400 \text{ грн.}$$

Ставка ЄСВ для всіх категорій платників з 01.01.2021 р. складає 22%.

$$С_{єв} = 22000 \cdot 0,22 = 63888 \text{ грн.}$$

Вартість електроенергії, що споживається апаратурою системою інформаційної безпеки протягом року ($С_{ел}$), визначається за формулою:

$$С_{ел} = Р \cdot F_р \cdot Ц_е, \text{ грн.,}$$

де Р – встановлена потужність апаратури інформаційної безпеки, (Р=2,3 кВт);

$F_р$ – річний фонд робочого часу системи інформаційної безпеки ($F_р$ = 1920 год.);

$Ц_е$ – тариф на електроенергію, ($Ц_е$ = 1,68 грн./кВт за годину).

Вартість електроенергії, що споживається апаратурою системою інформаційної безпеки протягом року:

$$С_{ел} = 2,3 \cdot 1920 \cdot 1,68 = 7418,88 \text{ грн.}$$

Витрати на технічне й організаційне адміністрування та сервіс системи інформаційної безпеки визначаються у відсотках від вартості капітальних витрат - 1% (Стос = 88788,04 * 0,01=887,88 грн).

Витрати на керування системою інформаційної безпеки ($С_к$) визначаються:

$$С_к = 8000+10000+290400+63888+7418,88+887,88 = 380594,76 \text{ грн.}$$

Таким чином, річні поточні витрати на функціонування системи інформаційної безпеки складають 380594,76 грн.

3.2 Оцінка можливого збитку від атаки на вузол або сегмент мережі

3.2.1 Оцінка величини збитку

Для розрахунку вартості такого збитку можна застосувати наступну спрощену модель оцінки для умовного підприємства.

Необхідні вихідні дані для розрахунку:

$t_п$ – час простою сервера внаслідок атаки, 8 години;

$t_в$ – час відновлення після атаки персоналом, що обслуговує серверну частину WEB-сторінки, 7 година;

$t_{ви}$ – час повторного введення загубленої інформації співробітниками атакованного вузла або сегмента корпоративної мережі, 5 години;

$З_о$ – заробітна плата персоналу, що обслуговує серверну частину WEB-сторінки (розробник), 22000 грн./міс.;

$З_с$ – заробітна плата співробітників атакованого вузла або сегмента корпоративної мережі, 15000 грн./міс.;

$Ч_о$ – чисельність обслуговуючого персоналу (адміністраторів та ін.), 1 особи;

$Ч_с$ – чисельність співробітників атакованого вузла або сегмента корпоративної мережі, 9 осіб.;

О – обсяг прибутку атакованного сервера, 4500000 грн. у рік;

$П_{зч}$ – вартість відновлення, 15000 грн;

I – число атакованих сегментів корпоративної мережі, 2;

N – середнє число атак на рік, 20.

Упущена вигода від простою атакованого сегмента корпоративної мережі становить:

$$U = П_п + П_в + V,$$

де $П_п$ – оплачувані втрати робочого часу та простої співробітників атакованого вузла або сегмента корпоративної мережі, грн;

$П_в$ – вартість відновлення працездатності вузла або сегмента корпоративної мережі (переустановлення системи, зміна конфігурації та ін.), грн;

V – втрати від зниження обсягу продажів за час простою атакованого вузла або сегмента корпоративної мережі, грн.

Втрати від зниження продуктивності співробітників атакованого вузла або сегмента корпоративної мережі являють собою втрати їхньої заробітної плати (оплата непродуктивної праці) за час простою внаслідок атаки:

$$П_{П} = \frac{\sum Зс}{F} \cdot t_{П} = \frac{((15000 \cdot 9) \cdot 8)}{176} = 6136{,}36 \text{ грн.}$$

де F – місячний фонд робочого часу (при 40-а годинному робочому тижні становить 176 год).

Витрати на відновлення працездатності вузла або сегмента корпоративної мережі включають кілька складових:

$$П_{В} = П_{ВИ} + П_{ПВ} + П_{ЗЧ},$$

де $П_{ВИ}$ – витрати на повторне введення інформації, грн.;

$П_{ПВ}$ – витрати на відновлення WEB-сервера, грн;

$П_{ЗЧ}$ – вартість заміни устаткування або запасних частин, 0 грн.

Витрати на повторне введення інформації $П_{ВИ}$ розраховуються виходячи з розміру заробітної плати співробітників атакованного вузла або сегмента корпоративної мережі $З_{С}$, які зайняті повторним введенням втраченої інформації, з урахуванням необхідного для цього часу $t_{ВИ}$:

$$П_{ВИ} = \frac{\sum Зс}{F} \cdot t_{ВИ} = \frac{((15000 \cdot 9) \cdot 5)}{176} = 3835{,}22 \text{ грн.}$$

Витрати на відновлення сегмента корпоративної мережі $П_{ПВ}$ визначаються часом відновлення після атаки $t_{В}$ і розміром середньогодинної заробітної плати обслуговуючого персоналу (адміністраторів):

$$П_{ПВ} = \frac{\sum Зо}{F} \cdot t_{В} = \frac{((22000 \cdot 1) \cdot 7)}{176} = 875 \text{ грн.}$$

$$П_{В} = 3835{,}22 + 875 + 0 = 4710{,}22 \text{грн.}$$

Втрати від зниження очікуваного обсягу прибутків за час простою атакованого вузла або сегмента корпоративної мережі визначаються виходячи із середньогодинного обсягу прибутку і сумарного часу простою сегмента корпоративної мережі:

$$V = \frac{O}{F_Г} \cdot (t_П + t_В + t_{ВИ})$$

$$V = \frac{4500000}{2080} \cdot 20 = 43269{,}23 \text{ грн.}$$

де $F_г$ – річний фонд часу роботи філії (52 робочих тижні, 5-ти денний робочий тиждень, 8-ми годинний робочий день) становить близько 2080 год.

U = 6136,36+4710,22+43269,23= 54115,81 грн.

Таким чином, загальний збиток від атаки на WEB-сервер організації складе:

$$B = \sum 2 \sum 20 \; 54115{,}81 = 2164632{,}4 \text{ грн.}$$

3.2.2 Загальний ефект від впровадження системи інформаційної безпеки

Загальний ефект від впровадження системи інформаційної безпеки визначається з урахуванням ризиків порушення інформаційної:

$$E = B \cdot R - C \text{ грн.,}$$

де B – загальний збиток від атаки у разі перехоплення інформації, тис. грн.;

R – вірогідність успішної реалізації атаки на сегмент мережі, частки одиниці (30%);

С – щорічні витрати на експлуатацію системи інформаційної безпеки.

Загальний ефект від впровадження системи інформаційної безпеки визначається з урахуванням ризиків порушення інформаційної безпеки:

$$E = 2164632{,}4 \cdot 0{,}3 - 380594{,}76 = 268794{,}96 \text{ грн.}$$

3.3 Визначення та аналіз показників економічної ефективності системи інформаційної безпеки

Коефіцієнт повернення інвестицій ROSI показує, скільки гривень додаткового прибутку приносить одна гривня капітальних інвестицій на впровадження системи інформаційної безпеки:

$$ROSI = \frac{E}{K} \text{, частки одиниці,}$$

де Е – загальний ефект від впровадження системи інформаційної безпеки грн.;

К – капітальні інвестиції за варіантами, що забезпечили цей ефект, грн.

Коефіцієнт повернення інвестицій ROSI:

$$ROSI = \frac{268794,96}{88788,04} = 3,02 \text{ , частки одиниці,}$$

Проект визнається економічно доцільним, якщо розрахункове значення коефіцієнта повернення інвестицій перевищує величину річної депозитної ставки з урахуванням інфляції:

$$ROSI > (N_{деп} - N_{інф})/100),$$

де $N_{деп}$ – річна депозитна ставка, (7,5 %);

$N_{інф}$ – річний рівень інфляції, (5%).

Розрахункове значення коефіцієнта повернення інвестицій:

$$3,08 > \frac{7,5-5}{100} = 3,02 > 0,25.$$

Термін окупності капітальних інвестицій T_o показує, за скільки років капітальні інвестиції окупаються за рахунок загального ефекту від впровадження системи інформаційної безпеки:

$$To = \frac{1}{3,02} = 0,33 \text{ роки.}$$

3.4 Висновок

Розробка засобів захисту інформації WEB-сторінки «RubyGlobal» є економічно доцільним, оскільки капітальні та експлуатаційні витрати будуть меншими за можливий відвернений збиток. Капітальні витрати складають 88 788,04 грн., експлуатаційні – 375594,76 грн. Величина річного економічного ефекту складає 268794,96 грн. Коефіцієнт повернення інвестицій ROSI складає 3,08 грн./грн.

ВИСНОВКИ

Проаналізувані загрози, що реалізуються через визначені вразливості. Розроблені засоби захисту від загроз несанкціонованого доступу до інформації, що зберігається на сайті, відштовхуючись від програмних засобів, які використовувались та надалі використовуються при розробці проекту.

Проведений аналіз загроз та створена модель порушника, проведене тестування на наявність вразливостей в програмній частині сайту за допомогою спеціальних програмних засобів та бібліотек. Надані рекомендації для запобігання реалізації визначених загроз. Виправлені вразливості в коді проекту та проведене повторне тестування на наявність вразливостей, по висновкам якого, перерахованих вразливостей більше намає.

Проведено розрахунок щодо доцільності впровадження запропонованих організаційних та програмних рішень.

ПЕРЕЛІК ПОСИЛАНЬ

1. Hacking Exposed Web Applications, Third Edition by Joel Scambray, 2003. 384 c;

2. Офіційна документація мови програмування Ruby [Електронний ресурс] – https://ruby-doc.org;

3. Офіційна документація фреймворку Ruby on Rails [Електронний ресурс] – https://rubyonrails.org;

4. Онлайн платформа систему контролю версій та зберігання репозиторіїв бібліотек і проектів [Електронний ресурс] - https://github.com (до наданого посилання додається назва програмної бібліотеки, яка використовувалась);

5. Офіційна документація фреймоврку для тестування Ruby/Ruby on Rails додатків RSpec [Електронний ресурс] – https://relishapp.com;

6. Інформаційні потоки в глобальних комп'ютерних системах - Д.В. Ланде, 2009. 295с;

7. Закон України «Про інформацію» (ВВР № 2658-XII від 02.10.92) № 48, ст 651

8. «Вимоги до захисту інформації WEB-сторінки від несанкціонованого доступу» НД ТЗІ 2.5-010, Київ 2003.

ДОДАТОК А Відомість матеріалів кваліфікаційної роботи

№	Формат	Найменування	Кількість листів	Примітка
1	А4	Реферат	3	
2	А4	Список умовних скорочень	1	
3	А4	Зміст	2	
4	А4	Вступ	2	
5	А4	1 Розділ	23	
6	А4	2 Розділ	21	
7	А4	3 Розділ	8	
8	А4	Висновки	1	
9	А4	Перелік посилань	1	
10	А4	Додаток А	1	
11	А4	Додаток Б	1	
12	А4	Додаток В	3	
13	А4	Додаток Г	1	
14	А4	Додаток Д	1	

ДОДАТОК Б Перелік документів на оптичному носії

1. Титульна сторінка.docx
2. Завдання.docx
3. Реферат.docx
4. Список умовних скорочень.docx
5. Зміст.docx
6. Вступ.docx
7. Розділ 1.docx
8. Розділ 2.docx
9. Розділ 3.docx
10. Висновки.docx
11. Перелік посилань.docx
12. Додаток А.docx
13. Додаток Б.docx
14. Додаток В.docx
15. Додаток Г.docx
16. Презентація.pptx

Додаток В

```
Name: actionpack
Version: 6.0.3.4
Advisory: CVE-2021-22885
Criticality: Unknown
URL: https://groups.google.com/g/rubyonrails-security/c/NiQl-48cXYI
Title: Possible Information Disclosure / Unintended Method Execution in Action Pack
Solution: upgrade to ~> 5.2.4.6, ~> 5.2.6, ~> 6.0.3.7, >= 6.1.3.2

Name: actionpack
Version: 6.0.3.4
Advisory: CVE-2021-22881
Criticality: Unknown
URL: https://groups.google.com/g/rubyonrails-security/c/zN_3qA26l6E
Title: Possible Open Redirect in Host Authorization Middleware
Solution: upgrade to ~> 6.0.3.5, >= 6.1.2.1

Name: actionpack
Version: 6.0.3.4
Advisory: CVE-2021-22904
Criticality: Unknown
URL: https://groups.google.com/g/rubyonrails-security/c/Pf1TjkOBdyQ
Title: Possible DoS Vulnerability in Action Controller Token Authentication
Solution: upgrade to ~> 5.2.4.6, ~> 5.2.6, ~> 6.0.3.7, >= 6.1.3.2

Name: actionpack
Version: 6.0.3.4
Advisory: CVE-2021-22902
Criticality: Unknown
URL: https://groups.google.com/g/rubyonrails-security/c/_5ID_ld9u1c
Title: Possible Denial of Service vulnerability in Action Dispatch
Solution: upgrade to ~> 6.0.3.7, >= 6.1.3.2

Name: activerecord
Version: 6.0.3.4
Advisory: CVE-2021-22880
Criticality: Medium
URL: https://groups.google.com/g/rubyonrails-security/c/ZzUqCh9vyhI
Title: Possible DoS Vulnerability in Active Record PostgreSQL adapter
Solution: upgrade to ~> 5.2.4, >= 5.2.4.5, ~> 6.0.3.5, >= 6.1.2.1

Name: carrierwave
Version: 2.1.0
Advisory: CVE-2021-21288
Criticality: Medium
URL: https://github.com/carrierwaveuploader/carrierwave/security/advisories/GHSA-fwcm-636p-68r5
Title: Server-side request forgery in CarrierWave
Solution: upgrade to ~> 1.3.2, >= 2.1.1
```

Рисунок 1 – Реультат тестування bundle audit 1

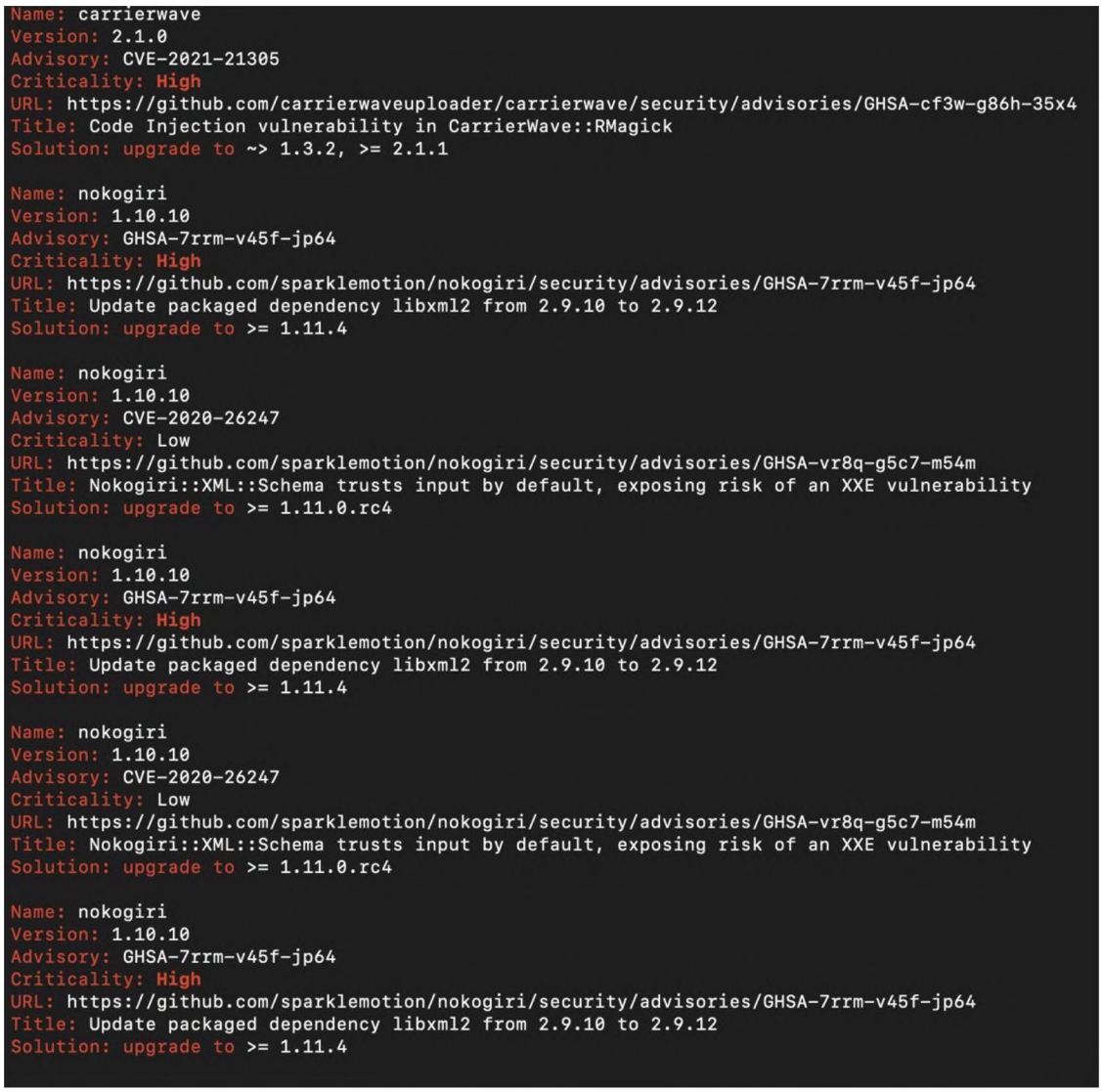

Рисунок 2 - Реультат тестування bundle audit 2

```
Name: nokogiri
Version: 1.10.10
Advisory: CVE-2020-26247
Criticality: Low
URL: https://github.com/sparklemotion/nokogiri/security/advisories/GHSA-vr8q-g5c7-m54m
Title: Nokogiri::XML::Schema trusts input by default, exposing risk of an XXE vulnerability
Solution: upgrade to >= 1.11.0.rc4

Name: nokogiri
Version: 1.10.10
Advisory: GHSA-7rrm-v45f-jp64
Criticality: High
URL: https://github.com/sparklemotion/nokogiri/security/advisories/GHSA-7rrm-v45f-jp64
Title: Update packaged dependency libxml2 from 2.9.10 to 2.9.12
Solution: upgrade to >= 1.11.4

Name: nokogiri
Version: 1.10.10
Advisory: CVE-2020-26247
Criticality: Low
URL: https://github.com/sparklemotion/nokogiri/security/advisories/GHSA-vr8q-g5c7-m54m
Title: Nokogiri::XML::Schema trusts input by default, exposing risk of an XXE vulnerability
Solution: upgrade to >= 1.11.0.rc4

Name: omniauth
Version: 1.9.1
Advisory: CVE-2015-9284
Criticality: High
URL: https://github.com/omniauth/omniauth/wiki/Resolving-CVE-2015-9284
Title: CSRF vulnerability in OmniAuth's request phase
Solution: upgrade to >= 2.0.0

Name: puma
Version: 4.3.6
Advisory: CVE-2021-29509
Criticality: High
URL: https://github.com/puma/puma/security/advisories/GHSA-q28m-8xjw-8vr5
Title: Keepalive Connections Causing Denial Of Service in puma
Solution: upgrade to ~> 4.3.8, >= 5.3.1

Name: puma
Version: 4.3.6
Advisory: CVE-2021-29509
Criticality: High
URL: https://github.com/puma/puma/security/advisories/GHSA-q28m-8xjw-8vr5
Title: Keepalive Connections Causing Denial Of Service in puma
Solution: upgrade to ~> 4.3.8, >= 5.3.1
```

Рисунок 3 - Реультат тестування bundle audit 3

ДОДАТОК Г Відгуки керівників розділів

Відгук керівника економічного розділу:

Керівник розділу _____ _____

(підпис) (ініціали, прізвище)

ДОДАТОК Д ВІДГУК

на кваліфікаційну роботу студента групи 125-17-1
Латишева Дмитра Олександровича на тему: «Розробка засобів захисту веб-сайту підприємства «RubyGlobal» від несанкціонованого доступу»

Пояснювальна записка складається зі вступу, трьох розділів і висновків викладених на 71 стор.

Метою роботи є розробка системи захисту інформації від несанкціонованого доступу веб-сайту підприємства «RubyGlobal».

Тема кваліфікаційної роботи тісно пов'язана з об'єктом діяльності бакалвара спеціальності 125 «Кібербезпека». Для досягнення поставленої мети в кваліфікаційній роботі вирішуються наступні задачі: повний аналіз WEB-сторінки як об'єкта ІТС, аналіз моделі порушника та загроз.

На основі моделі загроз було розроблено елементи комлексної системи захисту інформації і обраний профіль захищеності, відштовхуючись від якого були виконані всі вимоги захищеності інформації.

За час дипломування Латишев Д.О проявив себе фахівцем, здатним самостійно вирішувати поставлені задачі та заслуговує присвоєння кваліфікації бакалавра за спеціальністю 125 Кібербезпека, освітньо-професійна програма «Кібербезпека»

Рівень запозичень у кваліфікаційній роботі не перевищує вимог «Положення про систему виявлення та запобігання плагіату».

Кваліфікаційна робота заслуговує оцінки _____.

Керівник