Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(освітній рівень)

на тему: *" Розробка інформаційної системи для виявлення вразливостей веб-сайтів"*

Виконав: студент (ка)

Спеціальності:

125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)

Чех Т.П.
підпис (прізвище та ініціали)

Керівник *Стадник М.А.*
підпис *(прізвище та ініціали)*

Нормоконтроль *Лобур Т.Б.*
підпис (прізвище та ініціали)

Завідувач кафедри *Загородна Н.В.*
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м.Тернопіль – 2022

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

Студенту Чеху Тарасу Павловичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інформаційної системи для виявлення вразливостей веб-сайтів

Керівник роботи Стандик Марія Андріївна, к.т.н., старший викладач каф. КБ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «23» 03 2022 року № 4/7-178

2. Термін подання студентом завершеної роботи 17.06.2022

3. Вихідні дані до роботи

4. Зміст роботи (перелік питань, які потрібно розробити)

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці	Пулька Ч.В., проф. кафедри МТ		

7. Дата видачі завдання 16.02.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	16.02 – 19.02	*Виконано*
2.	Підбір джерел про основні вразливості сайтів	20.02 – 27.02	*Виконано*
3.	Опрацювання джерел в галузі дослідження	28.02 – 16.03	*Виконано*
4.	Вибір програмних засобів розробки	17.03 – 20.03	*Виконано*
5.	Реалізація програмного забезпечення	20.03-05.04	*Виконано*
6.	Оформлення розділу «Теоретична частина»	06.03 – 17.04	*Виконано*
7.	Оформлення розділу «Вибір програмних засобів розробки»	18.04 – 29.04	*Виконано*
8.	Оформлення розділу «Практична частина. Проєктування програмного забезпечення»	30.04 – 13.05	*Виконано*
9.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи хорони праці»	14.05 – 21.05	*Виконано*
10.	Оформлення кваліфікаційної роботи	22.05 – 05.06	*Виконано*
11.	Нормоконтроль	06.06 – 12.06	*Виконано*
12.	Перевірка на плагіат	10.06 – 15.06	*Виконано*
13.	Попередній захист кваліфікаційної роботи	16.06 – 19.06	*Виконано*
14.	Захист кваліфікаційної роботи	24.06.2022	

Студент _____ Чех Т.П.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Стадник М.А.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Розробка інформаційної системи для виявлення вразливостей веб-сайтів // Кваліфікаційна робота ОР «Бакалавр» // Чех Тарас Павлович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБс-42 // Тернопіль, 2022 // С. , рис. – , табл. – , кресл. – , додат. – .

Ключові слова: БЕЗПЕКА ВЕБ-ЗАСТОСУВАНЬ, ІНФОРМАЦІЙНА БЕЗПЕКА, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ОЦІНКИ РІВНЯ БЕЗПЕКИ.

Актуальність визначається необхідністю проєктуванням і розробкою веб-програми для проведення аналізу безпеки веб-сайтів. Розроблена система буде дозволяти підвищувати рівень захисту персональних даних, що оброблятимуться і зберігатимуться у веб-додатках.

Мета даного дипломного проєкту є проєктування та розробка веб-програми для проведення аналізу безпеки веб-сайтів.

Для того, щоб досягнути поставленої мети потрібно вирішити такі задачі:

– привести огляд методів аналізу безпеки застосувань;

– провести огляд та вибір засобів розробки програмного забезпечення;

– виконати проєктування програмного забезпечення для оцінки безпеки веб-додатків;

– провести тестування розробленого застосування;

Об'єктом дослідження є процес оцінки ризиків уразливості веб-додатків.

Предметом дослідження є методи та інструменти оцінки ризиків уразливості веб-додатків.

ANNOTATION

Development of an information system to identify website vulnerabilities // Qualification thesis of educational level "Bachelor" // Chekh Taras Pavlovych // Ternopil National Technical University named after Ivan Pulyuy, Faculty of Computer Information Systems and software engineering, Department of Cybersecurity, СБс-42 group // Ternopil, 2022 // P. , fig. - , table. - , chair. - , added. - .

Keywords: WEB APPLICATION SECURITY, INFORMATION SECURITY, SECURITY ASSESSMENT SOFTWARE.

Relevance is determined by the need to design and develop a web application for conducting website security analysis. The developed system will increase the level of security of personal data that is stored and processed in web applications.

The purpose of the diploma project is to design and develop a web application for conducting website security analysis.

To achieve this goal, you need to solve the following tasks::

- provide an overview of application security analysis methods;

- conduct an overview and selection of software development tools;

- perform software development to evaluate the security of web applications;

- conduct testing of the developed application;

The object of research is the process of assessing the vulnerability risks of web applications.

The subject of the research is Methods and tools for assessing the vulnerability risks of web applications.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ	Програмне забезпечення
ОС	Операційна система
ПДН	Персональні дані
КЗ	Контрольована зона
ЛОМ	Локальна обчислювальна мережа
СЗІ	Системи захисту інформації

ВСТУП

На сьогоднішній час значно виросла потреба в Інтернеті при розробці програмних продуктів. Програми, які створюються з допомогою Web-технологій, є найкращим вирішенням завдань з різноманітних областей. Такі розробки витісняють ті програми, які розроблялись на застарілих технологіях. У Web-сервісах поступово ускладнюються структури та реалізації, а це призводить до використання розподілених архітектур. Нові розробки потребують новіших, більш вдосконалених вимог щодо їх безпеки.

За останні кілька років у сфері інформаційної безпеки намітилася чітка тенденція – веб-додатки зазнають атак. Веб-програми продовжують залишатися основним вектором атак для злочинців, і ця тенденція не слабшає. Щодня з'являються нові повідомлення про високоорганізовані кібератаки на веб-сайти. Рівень веб-застосунків є легкою мішенню для зловмисників через вразливості, що існують у веб-додатках.

Безумовно, безпечна веб-розробка та написання безпечного коду – найефективніший метод мінімізації вразливостей веб-додатків. Однак написання безпечного коду та безпечна розробка набагато простіше, і включає різні питання. Більш того, в сьогоднішню конкурентну епоху, коли програми виводяться в мережу за дуже короткий час, безпеці не надається належного значення при розробці багатьох веб-додатків. Крім того, реалізація безпеки з використанням безпечного життєвого циклу розробки займає більше часу, потребує кваліфікованого персоналу та вищих витрат, що не є перевагами в організаціях малого та середнього розміру. Більш того, для деяких організацій витрати на безпечну розробку веб-додатків можуть бути не дуже розумним рішенням. оскільки вони не містять конфіденційної інформації, але вони можуть віддати перевагу безпеці бренду. Іншими словами, можна сказати, що безпека потрібна для кожної веб-програми, але рівень безпеки може варіюватися від організації до організації та від типу веб-програми, оскільки основна мета веб-додатків - отримати грошові кошти або таке інше. Веб-розробники можуть

реалізувати важливі функції безпеки, не витрачаючи багато часу на пошук вимог безпеки та дотримуючись суворого безпечного життєвого циклу розробки.

Використання нових технологій та інформаційних систем створює певну сукупність ризиків. А їх оцінка потрібна, щоб здійснювати контроль ефективної діяльності у сфері вітчизняної безпеки інформації, створення унікальних захисних систем і прийняття правильних заходів щодо ефективного захисту. Основними причинами реальних потенційних ризиків є існуючі загрози для національної безпеки. Через це постає важливість захисту Web-додатків. Це можна досягти шляхом виявлення порушення конфіденційності та класифікації всіх загроз в інформаційній безпеці. Ще дуже важливим кроком стане інтеграція управління ризиками в життєвий цикл інформаційних систем.

Об'єктом дослідження є процес оцінки ризиків уразливості веб-додатків.

Предметом дослідження є методи та інструменти оцінки ризиків уразливості веб-додатків.

Актуальність визначається необхідністю проєктуванням і розробкою веб-програми для проведення аналізу безпеки веб-сайтів. Розроблена система дозволить підвищити рівень захищеності персональних даних, які зберігаються і обробляються у веб-додатках.

Метою дипломного проєкту є проєктування та розробка веб-програми для проведення аналізу безпеки веб-сайтів.

Для досягнення поставленої мети потрібно вирішити такі задачі:

– привести огляд методів аналізу безпеки застосувань;

– провести огляд та вибір засобів розробки програмного забезпечення;

– виконати проєктування програмного забезпечення для оцінки безпеки веб-додатків;

– провести тестування розробленого застосування;

1 ТЕОРЕТИЧНА ЧАСТИНА

1.1 Огляд методів аналізу безпеки веб-застосунків

Завдяки досліджень у сфері інформаційної безпеки, що входить до складу державної інформаційної політики можна вирішити цілий комплекс питань. В результаті дуже важливим стає вияснення функцій та рис державної інформаційної політики у сфері інформаційної безпеки, а й виявлення основних факторів, які на неї впливають.

Уразливість безпеки веб-програм може призвести до крадіжки конфіденційних даних, порушення цілісності даних або вплинути на доступність веб-програм. Таким чином, завдання безпеки веб-додатків є однією з найбільш актуальних на даний момент: згідно з опитуванням Acunetix [1], 60% виявлених вразливостей зачіпають веб-додатки. Найпоширенішим способом захисту веб-додатків є пошук та усунення вразливостей у них. Приклади інших способів захисту веб-додатків включають безпечну розробку, впровадження систем виявлення та/або захисту від вторгнень та брандмауерів веб-додатків.

Згідно з OWASP [5], найефективніший спосіб пошуку вразливостей у веб-додатках – це перевірка коду вручну. Цей метод дуже трудомісткий, вимагає експертних навичок і схильний до помилок. Тому суспільство безпеки активно розробляє автоматизовані підходи до пошуку вразливостей безпеки. Ці підходи можна розділити на дві широкі категорії: «чорний ящик» та «білий ящик».

Перший підхід базується на аналізі веб-застосунків з боку користувача, за умови, що вихідний код програми недоступний. Ідея полягає в тому, щоб відправляти різні шкідливі шаблони (наприклад, за допомогою SQL-ін'єкцій або атак з використанням міжсайтових сценаріїв) у форми веб-застосунків і аналізувати їх результати після цього. Якщо спостерігаються будь-які помилки програми, робиться припущення про можливу вразливість. Такий підхід не гарантує ні точності, ні повноти одержаних результатів. Другий підхід базується на аналізі веб-додатків із боку сервера, за умови, що вихідний код програми

доступний. І тут можуть застосовуватися методи динамічного чи статичного аналізу.

Найбільш поширеною моделлю вразливостей під час перевірки вхідних даних є модель Tainted Mode [3]. Ця модель була реалізована за допомогою статичного чи динамічного аналізу.

Інший підхід до моделювання вразливостей під час перевірки вхідних даних полягає у моделюванні синтаксичної структури для аргументів чутливих операцій. Ідея полягає в тому, що веб-додаток піддається ін'єкційній атаці, якщо синтаксична структура для чутливих аргументів операції залежить від введення користувача. Цей підхід був реалізований за допомогою аналізу рядків і був застосований для виявлення вразливостей SQLI та XSS у PHP. Зрештою, цей підхід був реалізований для виявлення ін'єкційних атак під час виконання.

Одним з основних недоліків статичного аналізу в цілому є його схильність до помилкових спрацьовувань, викликаних неминучими неточностями аналізу. Це посилюється динамічною природою мов сценаріїв. Проте методи статичного аналізу зазвичай виконують консервативний аналіз, який враховує всі можливі шляхи контролю.

Навпаки, одним з основних недоліків динамічного аналізу є те, що він виконується на шляхах, що виконуються, і не дає жодних гарантій щодо шляхів, не охоплених під час даного виконання. Проте динамічний аналіз, який має доступ до внутрішніх компонентів процесу виконання веб-додатків, потенційно може бути більш точним.

Хакери можуть атакувати користувачів у 9 із 10 веб-додатків. Атаки включають перенаправлення користувачів на контрольований хакерами ресурс, крадіжку облікових даних при фішинг-атаках та зараження комп'ютерів шкідливим ПЗ [5].

Несанкціонований доступ до програм можливий на 39% сайтів. У 2019 році повний контроль над системою було отримано у 16 відсотках веб-додатків. У 8% систем повний контроль над сервером веб-додатків дозволяв атакувати локальну мережу [5].

Порушення конфіденційних даних були загрозою у 68 відсотках веб-додатків. Найбільш достовірні дані мали особистий характер (47% порушень) чи облікові дані (31%).

Статистика вразливостей [5]:

– 82 відсотки вразливостей було виявлено в коді програми.

– Середня кількість вразливостей на одну веб-програму скоротилася на третину порівняно з 2018 роком. У середньому кожна система містила 22 вразливості, у тому числі 4 мали високий рівень небезпеки.

– Одна з п'яти вразливостей має високий рівень небезпеки.

Відсоток веб-додатків, що містять вразливість із високим ступенем ризику, у 2019 році значно знизився на 17 пунктів порівняно з попереднім роком (рис.1.1). Середня кількість серйозних вразливостей на веб-додаток також скоротилася майже на третину.

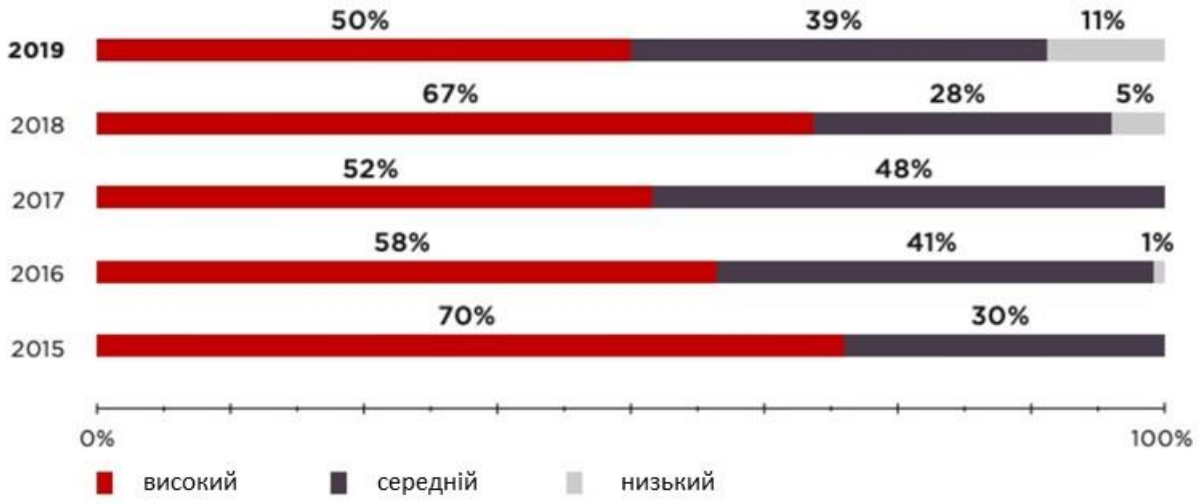

Рисунок 1.1 – Відсоток веб-застосунків Відсоток веб-програм, що містять вразливості з високим ступенем ризику за роками

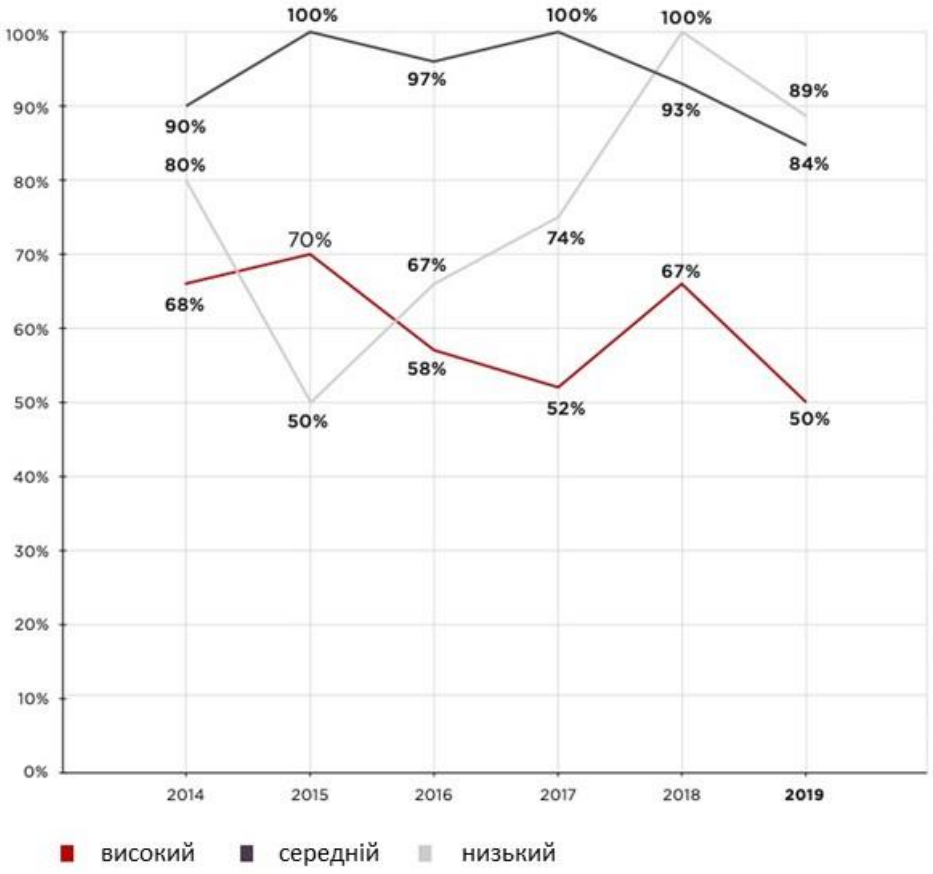

Рисунок 1.2 - Частка сайтів, що містять серйозні вразливості, за роками

Останні п'ять років показують зниження частки сайтів, що містять серйозні вразливості. Це обнадійлива ознака, що узгоджується із загальним покращенням безпеки (рис.1.2).

Найуразливіші вразливості веб-додатків у 2019 році включали неправильне налаштування безпеки. Кожна з п'яти протестованих додатків містила вразливості, що дозволяють хакерам атакувати сеанс користувача, наприклад конфіденційні файли cookie без прапорів HttpOnly і Secure[5]. Зловмисники можуть використовувати такі недоліки для виконання міжсайтових сценаріїв (XSS), щоб захопити ідентифікатор сеансу користувача та видати себе за користувача у програмі.

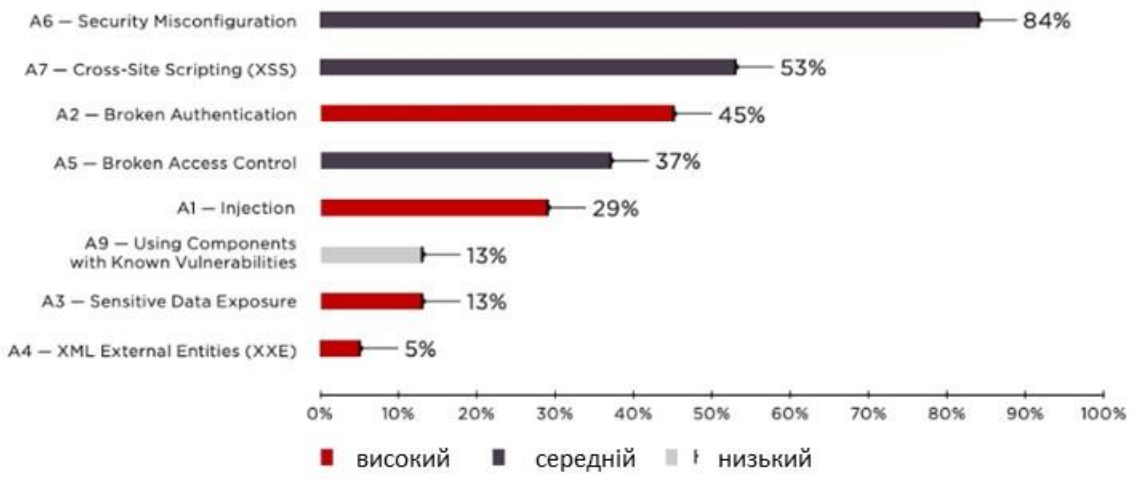

Рисунок 1.3 - Найпоширеніші уразливості OWASP Top 10 (відсоток веб-додатків)[5]

Злом автентифікації було виявлено в 45 відсотках веб-додатків (рис.1.3). Майже третина таких вразливостей полягає у нездатності правильно обмежити кількість спроб аутентифікації. Зловмисник може використовувати це для злому облікових даних та доступу до веб-програми. Наприклад, одна з програм може бути доступна з правами адміністратора лише після 100 спроб.

Тільки парольна автентифікація є фактором, що сприяє більшості атак автентифікації. Встановлено, що вимоги до віку та складності пароля, які раніше були «золотим стандартом», підривають безпеку. Згідно з останніми рекомендаціями NIST, організаціям слід перейти до багатофакторної аутентифікації, якщо вони ще цього не зробили [6, 7].

Кожен третій додаток у 2019 році мав зламаний контроль доступу. Обхід обмежень доступу зазвичай призводить до несанкціонованого розкриття, зміни або знищення даних. В одному протестованому додатку небезпечна авторизація дозволила змінити профіль будь-якого користувача. Спеціалісти Positive Technologies з'ясували ім'я користувача адміністратора програми, замінили відповідну адресу електронної пошти у профілі користувача на власну адресу, а потім використали стандартну процедуру скидання пароля для доступу до сайту з правами адміністратора [5].

Зазвичай можна мінімізувати вразливість автентифікації та авторизації, шляхом утримання життєвого повного циклу цілісної розробки ПЗ (SSDLC) під час розробки веб-програми.

На додаток до 10 вразливостей 2017 року, OWASP вказує на ряд недоліків, які необхідно перевірити «Ми виявили, що третина веб-додатків вразлива для клік-джекінгу (спотворення інтерфейсу критичної інформації, CWE451). Ще третина була вразлива для підробки міжсайтових запитів (CSRF)». У CSRF-атаці хакер використовує спеціально створені сценарії для виконання дій, видаючи себе за користувача, що увійшов до вразливого веб-додатку. Уявіть, що ви увійшли на веб-сайт, вразливий для CSRF, наприклад, онлайн-банк. Ви отримуєте фішингове повідомлення з посиланням та натискаєте це посилання. Потім хакер відправляє спеціально створений запит вразливому сайту (вашому онлайн-банку) для виконання певних дій, які хоче хакер (наприклад, переказ коштів на рахунки хакера). Онлайн-банк не зможе відрізнити цей несанкціонований запит від законного, якщо не використовує захист від CSRF-атак. Захист зазвичай потребує наявності унікальних одноразових ключів (токенів CSRF), підтвердження справжності (наприклад, за допомогою пароля), перевірки того, що запит виконується фактичним користувачем (можливо, за допомогою CAPTCHA), або встановлення додаткового прапора SameSite.

Атаки на клієнтів залишалися загрозою для дев'яти з кожних десяти додатків у 2019 році, як і у 2018 році. Міжсайтовий скриптинг (XSS) залишається однією з найважливіших причин. Зловмисники можуть заражати комп'ютери шкідливими програмами, організовувати атаки фішинга, наприклад, захоплювати облікові дані і виконувати дії, видаючи себе за користувача. Як загальна рекомендація, веб-програми повинні очищати все введення користувача, яке згодом відображається в браузері, включаючи поля заголовка HTTP-запиту, такі як User-Agent і Referer. Потенційно небезпечні символи, які можна використовувати при форматуванні сторінки HTML, мають бути замінені неформатними еквівалентами. Також рекомендується використовувати сучасні брандмауери веб-застосунків, оскільки вони можуть блокувати міжсайтовий скриптинг.

Порушення важливої інформації є другою найбільш актуальною загрозою безпеці сайту. Майже в половині всіх порушень (47%) особисті дані зазнавали ризику. Повноваження користувача також займали чільне місце (31%). Як видно з аналізу інцидентів кібербезпеки 2019, інформація є головною метою хакерів, коли вони націлені на організації [5].

У 16 відсотках веб-програм серйозні вразливості дозволили взяти під контроль як додаток, так і ОС сервера.

Наприклад, доступ до веб-програми можна використовувати для додавання аналізатора JavaScript до його коду та атаки на користувачів сайту. Сніфери можуть викрадати облікові дані, так і особисті дані, а також дані платіжної картки. Атаки з використанням сніферів JavaScript були найбільш небезпечними атаками на окремих осіб у 2018-2019 роках.

При цілеспрямованій атаці на вразливість веб-додатків можуть допомогти в зборі даних про внутрішню мережу компанії, такий як структура сегментів мережі, портів і служб. У багатьох випадках хакери можуть навіть отримати доступ до внутрішніх мережевих ресурсів і конфіденційних даних, що зберігаються там.

Хакери можуть об'єднувати набори вкрадених даних та використовувати їх для атак на інші веб-ресурси, використовуючи так звані облікові дані. У травні 2019 року такі атаки торкнулися півмільйона клієнтів двох інтернет-магазинів.

Справедливо сказати, що безпека більшості веб-застосунків все ще залишається низькою. Половина сайтів містить уразливість високого ризику. Однак щороку спостерігається неухильне зниження частки веб-додатків із серйозними вразливістю. Середня кількість таких вразливостей на програму впала на третину порівняно з 2018 роком. Ще одна позитивна тенденція полягає в тому, що компанії більш серйозно ставляться до безпеки не лише загальнодоступних веб-додатків, а й внутрішніх.

Досягнення та постійна підтримка високої безпеки веб-додатків – непростий процес. Однак є два основні правила:

- Якнайшвидше виправите всі виявлені недоліки
- Зробити процеси автоматично скрізь, де це можливо

Щоб дотримуватись цих правил, компанії повинні навчати розробників безпечним методам розробки. Інструменти автоматичного аналізу вихідного коду є гарним доповненням до аналізу безпеки веб-додатків. Разом вони зменшать недоліки та вразливості, що виникають у процесі розробки. Також рекомендується превентивні заходи, такі як брандмауер веб-застосунків (WAF). Оскільки веб-програми постійно змінюються, а кожна нова зміна пов'язана з ризиком впровадження нової вразливості, WAF пропонують надійний захист. Щоб бути ефективним, WAF повинен не тільки виявляти та запобігати відомим ризикам на рівні додатків та бізнес-логіки. Він також повинен виявляти використання вразливостей нульового дня, запобігати атакам на користувачів, а також аналізувати та зіставляти події для виявлення ланцюжків атак

1.2 Огляд програмного забезпечення для аналізу безпеки веб-застосунків

У теперішній час є дуже актуальним різноманітні розробки в сфері захисту. Через сотні тисяч зовнішніх чи внутрішніх атак на комп’ютерні мережі і системи, потрібно проводити оцінки захисту таких систем, їх слабкі місця, що дозволяють провести злом. А також необхідно провести аналіз дій, які зможуть запобігти негативним наслідкам.

Для того, щоб провести повний аудит безпеки необхідно використати тести на проникнення. Це так званий Пентест.

Пентест є дуже популярною послугою у світі. В сфері інформаційної безпеки пентест стає критично необхідним. Людина, яка дотримується умов проведення даного тесту, ще називається етичним хакером. Для того, щоб знати конкретні умови проведення пентесту, між замовником і виконавцем укладається заздалегідь угода, де все описується. Це допоможе не допустити витоку інформації про стан організованої безпеки у конкретній компанії.

Суть роботи Пентестів заключається у санкціонованій спробі оминути існуючий комплекс засобів захисту інформаційної системи [9].

Переважно у пентестах можуть використовуватись методики Національного інституту стандартів та технології США (NIST).

Для того, щоб тест на проникнення був ефективним, потрібно використовувати різні об'єктами інформаційної інфраструктури. Такими компонентами переважно є робочі станції, мережеве обладнання, бази даних, сервери і т.д.

Завдання пентестера – з’ясувати їхні можливості в експлуатації, після чого ліквідувати ці вразливості.

Найбільш підходящі і ефективні такі методики для проведеня пентестів:

- використання елементів OSSTMM, OISSG;
- стандарт PTES – Penetration Testing Execution Standard;
- методика OWASP – Open Web Application Security Project;
- проект OWASP TOP.

Деякі методики наголошують на технічну складову тесту, інші описують організацію самого процесу тестування. Ці всі методики мають, як завжди свої переваги і недоліки.

Найактуальнішим інструментом для пентесту є Kali Linux [8] - це орієнтовані на платформі Linux Debian, компоненти для тестування на проникнення, створений для відпрацювання саме етичного хакінгу.

Основою роботи даної платформи є застосування методик пентесту, що включає в себе 10 етапів. Деякими з них є: збір інформації про систему (Information Gathering), визначення меж тестування (Target Scoping), соціальна інженерія (Social Engineereing) і інші.

Кожний етап має свої програми, які в свою чергу містять унікальний набір утиліт Kali Linux.

Metasploitable 2 Linux - операційна система, спеціально спроектована виявлення властивостей тестування, тести експлоїт та навчання новачків. Віртуальна машина Metasploitable - це спеціально вразлива версія Ubuntu Linux, призначена для випробувань засобів безпеки та демонстрації найпоширеніших вразливостей. Версія 2 цієї віртуальної машини доступна для всіх користувачів

Інтернету. Це віртуальна машина сумісна з VM Ware, Virtual Box та іншими платформами загальної віртуалізації.

The Mutillidae - це веб-додаток, який містить уразливості з набору OWASP Top Ten. На відміну від DVWA, Mutillidae дозволяє користувачеві змінювати "Security Level" від 0 (цілком небезпечно) до 5 (безпечно). Якщо програма пошкоджена ін'єкціями коду або іншими діями користувача, натиснувши "Reset DB", програма відновиться до оригінального вигляду. Ця програма дозволить випробувати всі сучасні атаки на веб-програми або сайти в інтернеті.

Damn Vulnerable Web App (DVWA) це «дуже вразливий веб-додаток» з PHP / My SQL. безпека веб-додатків на реальному прикладі у навчальних середовищах.

1.3 Постановка задачі на проєктування

Веб-програми в сучасному світі знайшли своє місце практично у всіх галузях діяльності. Це спеціальні програмні забезпечення, що використовуються для найрізноманітніших цілей.

Однак, незважаючи на важливу роль, є й зворотний бік: їхня компрометація може призвести до катастрофічних наслідків. Для звичайного користувача це може призвести до крадіжки особистих даних. Для компаній це може призвести до того, що вона втратить репутацію, фінансові втрати. Саме тому безпека веб-програм не менш важлива, ніж реалізація основних функцій.

Постановка завдання полягає у створенні веб-додатка для оцінки захищеності сайтів, яка допомагає оцінити рівень захищеності оцінку рівня вразливості сайту, що тестується.

1. Безпека веб-застосунків.

Щоб максимально забезпечити безпеку, необхідно знати рівень захищеності веб-додатків, а для цього потрібно використовувати відповідні способи її отримання.

Безпека веб-додатків - це властивість цієї програми функціонувати без прояву різних негативних наслідків конкретної комп'ютерної системи.

Рівень безпеки веб-програм - це показник, який характеризує ймовірність того, що за певних умов використання веб-програми буде отримано потрібний результат.

У роботах [13, 14] були показані існуючі можливі способи перевірки захищеності веб-додатків. Найпоширенішими є такі:

1. Тести проникнення.
2. Сканери безпеки – як інструмент виявлення вразливостей.
3. Аналіз веб-програми вручну. Моделювання загроз.
4. Статичне та динамічне тестування.
5. Тестування або перевірка системи за допомогою такого набору тестів, як перевірка сумісності, тестування продуктивності, функціональне тестування та тестування безпеки.

Метою роботи є проектування та розробка веб-програми для проведення аналізу безпеки веб-сайтів.

Щоб досягти поставлених задач потрібно вирішити такий ряд завдань:

- Виконати аналіз предметної галузі, пов'язаної з оцінкою вразливостей веб-додатків та методів їх виявлення;

- Виконати аналіз інструментів тестування на проникнення та інструментів моделювання проникнення;

- Виконати проектування та розробку веб-додатку для проведення аналізу безпеки веб-сайтів;

- Виконати тестування розробленого веб-додатку

Отже, проведений аналіз дозволив сформулювати основні вимоги до веб-застосунку з оцінки вразливостей веб-сайтів та оформити ці вимоги у вигляді технічного завдання.

2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ РОЗРОБКИ

2.1 Огляд та вибір мов програмування

C# - це об'єктно-орієнтована мова програмування. Він розроблений у компанії Microsoft під керівництвом Андерса Хейлсберга та членів його команди в рамках ініціативи .Net і був прийнятий Європейською асоціацією виробників комп'ютерів (ECMA) та Міжнародною організацією зі стандартизації (ISO).

C# покращено і оновлено різні функції C і C++, включаючи наступні:

C # має суворі логічні дані типів змінних, наприклад, bool, в той час як логічні типи змінних C++ можуть повертатися у вигляді цілих чисел або покажчиків, щоб уникнути поширених помилок програмування.

C # автоматично керує пам'яттю віддалених об'єктів через збирач сміття, що позбавляє розробників від занепокоєння та витоків пам'яті.

Тип C# більш безпечний, ніж C++, і має лише надійні перетворення за замовчуванням (наприклад, ціле розширення), які виконуються під час компіляції або виконання.

Програмісти не можуть виконувати неявні перетворення між логічними значеннями, членами перерахування та цілими числами (крім 0) у тип, що перераховується. Користувальницький переклад має бути явним або мається на увазі, на відміну від неявних операторів перетворення C++ за умовчанням та конструкторів копіювання.

Синтаксис C # виключно виразний, але при цьому простий і легкий у освоєнні. Синтаксис C # спрощує багато труднощів мови C і пропонує унікальні функції, такі як обнулювані, перераховані, делеговані, лямбда-вираження та прямий доступ до пам'яті, які недоступні в Java. C # підтримує стандартні методи та типи, які певною мірою забезпечують кращу безпеку та продуктивність, а також ітератори, які дозволяють розробникам збирати класи для опису користувача поведінки ітерацій, яке клієнтський код може легко використовувати. Вирази запиту, інтегровані у мову (LINQ), роблять добре типізований запит першокласною конструкцією мови.

Хоча конструкції C# безпосередньо слідують традиційним різним високорівневим мовам, C і C++ є об'єктно-орієнтованою мовою програмування. Він дуже схожий на Java, має безліч функцій програмування, які роблять його привабливим для різних програмістів по всьому світу.

Основні важливі функції C#:

- Бульові умови
- Автоматичний збір сміття
- Стандартна бібліотека
- Варіанти монтажу
- Властивості та події
- Управління делегатами та заходами.
- Індексатори
- Умовна компіляція
- Проста багатопоточність
- LINQ та лямбда-вирази

Як об'єктно-орієнтована мова C# підтримує теорію інкапсуляції, наслідування та поліморфізму. Всі змінні та методи разом з основним процесом, точкою входу в додаток, інкапсулюються у визначення класу. Клас може успадковувати від першого класу, щоб реалізовувати кілька інтерфейсів. Для методів, які перевизначають віртуальні методи у первинному (батьківському) класі, потрібне ключове слово override як метод, що дозволяє уникнути незапланованого перевизначення. У C# структура подібна до легкого класу; - це тип, присвоєний стеку, який може виконувати інтерфейси, але несумісний із наслідуванням.

Накопичуючи ці загальні об'єктно-орієнтовані принципи, розробка програмних компонентів полегшується завдяки кільком новаторським мовним конструкціям, які включають:

Сигнатури інкапсульованих методів називають делегатами, які дозволяють отримувати повідомлення про події безпечного типу.

Властивості, які є оцінювачами для приватних змінних-членів.

Коментарі до документації XML онлайн.

Інтегрований із мовою запит (LINQ), який надає можливості інтегрованого запиту у різних джерелах даних.

C# - це просто мова, орієнтована на CLR. C # має такі переваги:

– C# чисто об'єктно-орієнтований, але C++ являє собою поєднання об'єктно-орієнтованого та процедурно-орієнтованого.

– C# безпечний за типом

– Програмістові не потрібно приділяти багато уваги таким проблемам, як втрата пам'яті, яка непокоїть програміста на C++.

– Концепція збирання добре вирішує проблему контролю версій.

– Проста у створенні, багата бібліотека класів спрощує реалізацію багатьох функцій.

– Кросплатформність. Програма буде нормально працювати, тільки якщо на комп'ютері встановлено .NET Framework.

– Підтримка розподіленої системи.

Недоліки:

Програміст не може виконувати низькорівневі речі, наприклад, безпосередньо взаємодіяти з обладнанням через драйвери та прошивки.

Він не постачається з незалежним компілятором, який може прямо інтерпретувати максимальні рівні мови для базової апаратної архітектури чистого асемблера. Він використовує свій байт-код і JIT-компілятор, який значною мірою вбудований в структуру .Net і є основою структури .Net як посередник для машинного коду замість прямої взаємодії з обладнанням.

2.2 Огляд та вибір СУБД

На ринку існує багато систем управління реляційними базами даних.

Прикладами таких реляційних систем управління базами даних можуть бути Microsoft SQL Server, Microsoft Access, Oracle, DB2 і т.д.

Розглянемо особливості My SQL проти іншими системами управління базами даних.

Як порівняння виберемо одну з найпопулярніших систем управління базами даних SQL Server.

My SQL підтримує кілька механізмів зберігання, кожен зі своїми специфікаціями, у той час як інші системи, такі як MS SQL Server підтримують лише один механізм зберігання. Щоб оцінити це твердження, розглянемо два механізми зберігання, підтримувані My SQL.

InnoDB - це механізм зберігання за промовчанням, який надається My SQL з версії 5.5. InnoDB підтримує зовнішні ключі для цілісності посилань, а також підтримує ACID-стандартні транзакції.

My ISAM був механізмом зберігання за умовчанням для My SQL до версії 5.5. My ISAM не підтримує транзакції. Його переваги перед InnoDB включають простоту та високу продуктивність.

My SQL має високу продуктивність у порівнянні з іншими СУБД. Це пов'язано з простотою дизайну та підтримкою двигунів з кількома сховищами.

Розрізняють кілька варіантів видань My SQL: Community безкоштовне видання, Enterprise видання має ліцензійний збір, який є меншим у порівнянні з ліцензійними зборами для таких продуктів, як Microsoft SQL Server.

Крос-платформний My SQL працює на багатьох платформах, що означає, що він може бути розгорнутий на більшості машин. Інші системи, такі як MS SQL Server, працюють лише на платформі Windows.

Щоб взаємодіяти з My SQL, знадобиться інструмент доступу до сервера баз даних, який може взаємодіяти з сервером My SQL. My SQL підтримує кілька підключень користувача.

Таким чином можна зробити такі висновки:

My SQL – це реляційна база даних з відкритим вихідним кодом, яка є крос-платформною.

My SQL підтримує кілька механізмів зберігання, які значно покращують налаштування продуктивності сервера та гнучкість. До версії 5.5 за замовчуванням механізмом зберігання був My ISAM, який не підтримував транзакції, а починаючи з версії 5.5; Механізм зберігання за замовчуванням - InnoDB, який підтримує транзакції та зовнішні ключі.

2.3 Огляд та вибір платформи

Visual Studio Code (VSC) є безкоштовним редактором коду, який легкий і зрозумілий навіть недосвідченим програмістам. Даний редактор допомагає будь-якому програмісту написати код та виправляє його, використовуючи метод intelli-sense.

Згодом деякі мови програмування потребували спеціальної структури та підтримки для подальшого кодування та розробки в ній, що було неможливо з використанням спеціальних редакторів. VI Editor, Sublime Text Editor, є одним із багатьох видів редакторів, які з'явилися на світ. Найвідомішим і майже всіма мовами програмування є VSC. Його функції дозволяють користувачеві змінювати редактор відповідно до його використання, що означає, що користувач може завантажувати бібліотеки з Інтернету та інтегрувати їх з кодом відповідно до своїх вимог.

Visual Studio Code має кілька унікальних функцій:

Підтримка кількох мов програмування: підтримує кілька мов програмування. Раніше програмістам була потрібна веб-підтримка: інший редактор для різних мов, але з вбудованою багатомовною підтримкою. Це також означає, що він легко виявляє, якщо є якась помилка або міжмовне посилання, вона зможе легко її виявити.

Intelli-Sense: він може визначити, чи залишився якийсь фрагмент коду незавершеним. Крім того, оголошення та загальні синтаксиси змінних виконуються автоматично.

Крос-платформна підтримка здатна працювати на трьох платформах: Windows, Linux та Mac.

Web-підтримка: Постачання вбудованої підтримки веб-програм. Таким чином, у VSC є можливість створення та підтримання Web-програм.

Ієрархія в структурі: усі файли коду розміщені в папках і файлах. Коло основних файлів коду можуть бути розташовані декілька інших файлів, які потрібні інший проєктів. Ви можете видалити ці файли для зручності.

Покращення коду: так як деякі частинки коду можуть бути оголошені трохи інакше, то це може допомогти користувачеві при потребі замінити його на варіант, який буде запропонований.

Коментування: звичайна функція, проте деякими мовами вона може не підтримуватись. Але завдяки коментуванню користувач може згадати або відстежити відповідно до бажаної послідовності.

Отже, у розділі розглянуті та вибрані інструменти для розробки та створення веб-програми. Як СУБД пропонується використання My SQL, C# обраний мовою програмування. Як середовище розробки пропонується використовувати VSC.

.

3 ПРАКТИЧНА ЧАСТИНА. ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Опис сценаріїв використання підсистеми

Функціональні вимоги до системи визначають дії системи, які вона має виконувати. Ці вимоги реалізуються через функції системи. Під функцією програмного продукту(ПП) мається на увазі сукупність дій ПП, спрямовану досягнення певної мети чи аспект певного поведінки системи, а під завданням мають на увазі функція чи частина функції ПП, є формалізовану сукупність автоматичних дій, виконання яких призводить до результату заданого виду.

Функціональні вимоги до програмного продукту можна представити графічно за допомогою діаграми прецедентів у нотації UML. Суть даної діаграми у тому, що програмний продукт представляється вигляді безлічі сутностей чи акторів, взаємодіючих із останнім з допомогою про варіантів використання. Варіант використання служить для опису сервісу, що продукт надає актору. Діаграма варіантів використання ПП, що розробляється, представлені у додатку.

На діаграмі зображені два актори - "Експерт" та "Аналітик". Наявність актора "Експерт" обумовлена тим, що для вирішення завдання оцінки рівня ризику ІБ веб-сайту використовується методика, яка передбачає надання експертами значень лінгвістичним змінним.

Відповідно до "Експерта" доступна "Робота з лінгвістичними змінними": завдання значень лінгвістичним змінним, редагування та видалення значень.

Основним прецедентом "Аналітика" є "Розрахунок величини ризику", що включає низку інших прецедентів, таких як: "Опис веб-сайту", "Вибір показників ІБ веб-сайту", "Встановлення коефіцієнтів значущості показників", "Класифікація показників за підмножинами їх значень" виду В", "Розрахунок числового значення величини ризику ІБ веб-сайту" та "Переклад значення величини ризику ІБ АІС у лінгвістичну форму". Діаграма варіантів використання представлена рисунку 3.1.

Рисунок 3.1 – Діаграма варіантів використання

"Користувач" також може працювати з кінцевими результатами розрахунків, наприклад, зберігати їх у БД.

Отже, згідно з вищенаведеним, до основних функціональних вимог проєктованого ПП можна віднести такі:

- система має надавати можливість експертам коректно вводити дані визначення лінгвістичних змінних, і навіть зберігати їх;

- система повинна надавати можливість коректно запроваджувати дані про експертів, а також зберігати їх;

- система повинна надавати можливість користувачеві розраховувати рівень ризику ІБ веб-сайту, використовуючи експертні оцінки;

- система повинна надавати можливість користувачеві коректно вводити необхідні дані для розрахунку рівня ризику ІБ веб-сайту;

- система повинна надавати можливість користувачеві редагувати та видаляти введені раніше дані;

- система має надавати можливість користувачеві зберігати результати розрахунків.

Обмеження - це умови, які модифікують окремі вимоги чи групи вимог до програмної системи, звужуючи вибір можливих шляхів реалізації. Наприклад, однією з таких вимог може бути вимога про наявність коментарів у програмному коді.

3.2 Проєктування інтерфейсу користувача

При виконанні проєктів для повнішого відображення прийнятих проєктних рішень рекомендується будувати діаграми компонентів для наступних уявлень:

1) модель вихідного програмного коду системи;

2) модель програмного коду системи, що виконується;

3) модель артефактів системи, що поставляється замовнику.

На першій діаграмі слід відобразити розподіл класів файлів з вихідним кодом; на другий — представити за допомогою стереотипів розширення «compile» та (або) «build» трансформацію вихідного коду в код, що виконується, тобто показати взаємозв'язок між одними і тими ж артефактами, представленими в різних фізичних форматах; на третій — уявити зв'язок зазначених програмних артефактів з інформаційними та сторонніми артефактами, тобто з базою даних та СУБД.

На рис. 3.2-3.3 представлені приклади перелічених раніше варіантів діаграм компонентів. У реальному проєкті крім проблемних класів, операції яких реалізують бізнес-логіку програми, існує велика кількість допоміжних класів - це всілякі форми, що формують інтерфейс користувача. Тому модель вихідного коду може виявитися дуже об'ємною. Якщо врахувати, що в деяких системах програмування з кожною формою пов'язується не один, а кілька файлів, то для того, щоб діаграма не "потонула" в деталях, або обмежуються вказівкою основних асоційованих файлів, або не виносять допоміжні класи на діаграми.

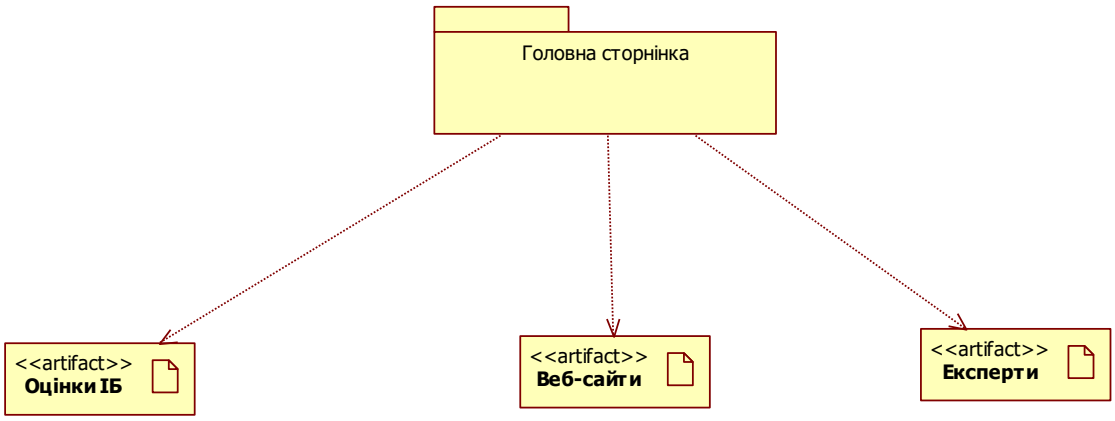

Рисунок 3.2 – Діаграма компонентів модель вихідного програмного коду

Рисунок 3.3 - Діаграма компонентів - модель програмного коду, що виконується.

З рис.3.3 видно, що для сторінок порталу існують .cs-сторінки з програмним кодом, що реалізують динамічну складову для aspx-сторінок порталу. Цей артефакт є вихідним кодом створюваної програми.

Розміщення артефактів програми по фізичних пристроях відображається діаграмою розгортання (рис.3.4)

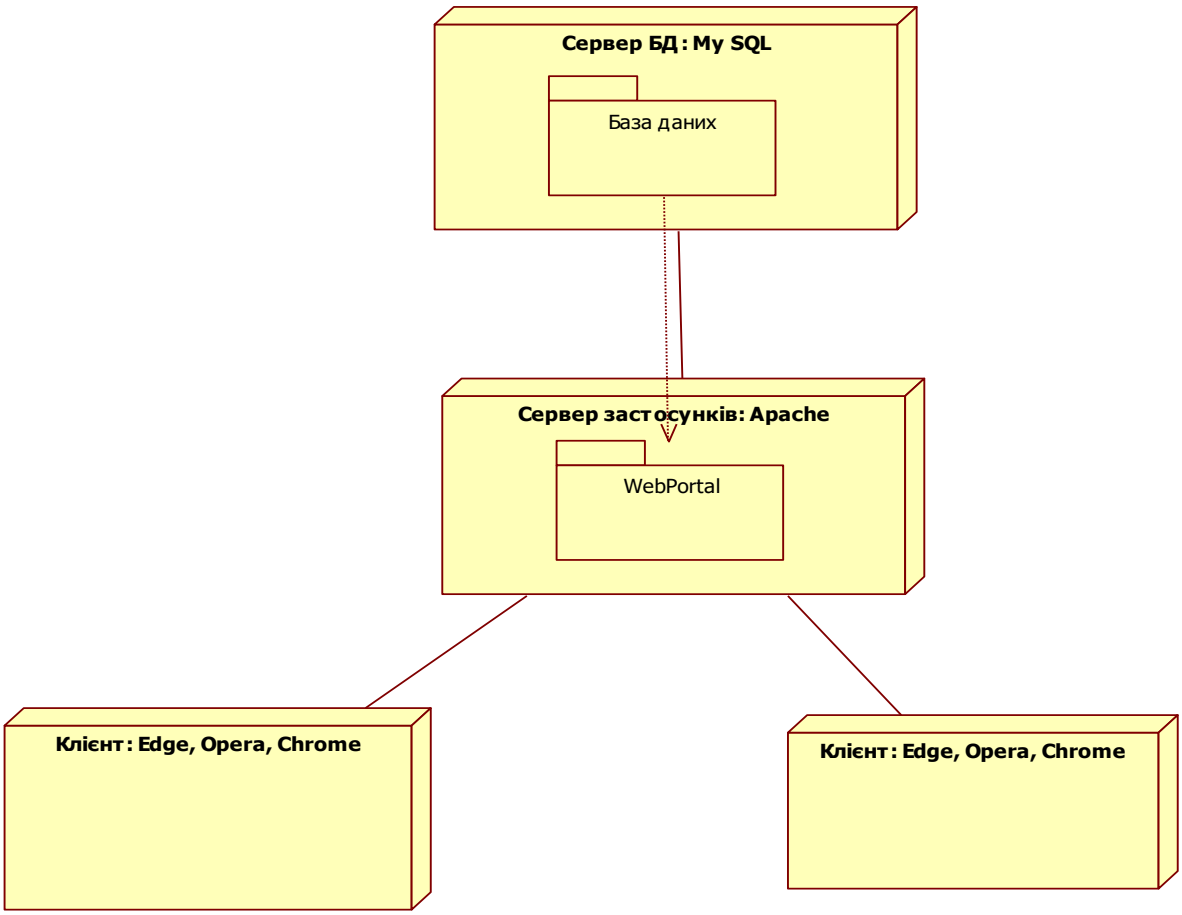

Рисунок 3.4- Діаграма розгортання - артефакти розподілені по трьох комп'ютерах

3.3 Розробка алгоритму оцінки безпеки веб-сайтів

3.3.1 Опис методу аналізу безпеки сайтів

У теперішній час стає дуже складно забезпечувати збільшення ефективності інформаційної безпеки. З цим пов’язано підвищення вимог до веб-застосунків. Саме тому сьогодні є дуже актуальним вдосконалення безпеки управління веб-додатками.

Велику роль у підвищенні рівня інформаційної безпеки є використання математичних методів та моделей з метою оцінки та запобігання ризиків. Але використання цих методів, щоб вирішити різні завдання часто неможливе внаслідок їхньої складності. Тому значно ефективним буде метод експертних оцінок. Даний метод застосовується при опрацюванні оцінок досвідчених фахівців. Ці оцінки формуються на основі інтуїції, знань і досвіду фахівців. Також їхні оцінки базуються на статистичних даних і факторів ризику.

Даний метод застосовується в ситуаціях, які частенько виникають у розробках сучасних додатків з метою оцінки інформаційної безпеки а ще при плануванні на довгий строк.

Ризики безпеки інформаційних систем дуже тісно пов'язані з невизначеністю. Можна визначити два випадки невизначеності: ідентифікацію поточного та майбутнього стану систем.

При вирішенні завдань, пов'язаних з оцінкою та прогнозуванням ризиків безпеки, часто виникає питання щодо якісної інтерпретації тих чи інших рівнів параметрів. Лінгвістична оцінка рівня безпеки чіткіша і найкраще визначає стан захищеності IT-інфраструктури, що у свою чергу спонукає керівника приймати те чи інше рішення.

Щоб ефективно виконати лінгвістичну оцінку, потрібні дві речі:
По-перше, вам потрібно визначити лінгвістичну шкалу для оцінки. Найчастіше використовується п'ятирівневий класифікатор дуже низький (VL) - низький (L) - середній (A) - високий (H) - Дуже високий (VH)".

По-друге, необхідно зібрати всю наявну інформацію визначення лінгвістичної оцінки: кількісні дані, зібрані групи подібних об'єктів спостереження.

Наприклад, для якісної оцінки та прогнозу рівня інформаційної безпеки необхідно зібрати статистичну інформацію щодо аналогічних інформаційних систем за відносно короткий період моніторингу. Це необхідно підтримки стану статистичної однорідності. При цьому необхідно враховувати ті закономірності, які притаманні об'єктам інформаційної безпеки.

Слід зазначити, що немає загальних універсальних правил для точної та оперативної оцінки та прогнозу рівня інформаційної безпеки АІС. При зборі вихідних даних для лінгвістичного аналізу може виникнути ряд проблем.

Основні етапи лінгвістичної класифікації:

1. Проведено дослідження вихідного набору даних та його верифікацію у вигляді квазістатики. Є інформація, що у цих даних прихований певний закон розподілу даних, наприклад, "сіра" шкала Поспєлова.

2. Потім слід визначити основні вузли. За відсутності експертної оцінки вузлові точки можуть бути визначені за простим правилом: вузлова точка – лівий кінець медіа інтервалу, вузлова точка – правий кінець медіа інтервалу, середня точка – відповідає максимальній гістограмі або медіанній гістограмі.

3. Інтервал між двома вузловими точками, що стоять поруч, поділяється на три зони, середня з яких є зоною експертної невизначеності у класифікації. Таким чином, первинна лінгвістична інтерпретація гістограми завершена.

Після класифікаційного визначення можна зробити корекцію пестаскаля. Для цього можна модифікувати вузлові точки класифікації, зближуючи їх та звужуючи зону невизначеності. Також можете замінити вузлову точку абсолютним довірчим інтервалом та спробувати розширити її по обидва боки від вузлової точки. Усі роз'яснення мають бути зроблені на основі погодженої експертної оцінки.

3.3.2 Опис роботи алгоритму

Розглянемо застосування алгоритму методики оцінки рівня ІБ сайту порівняння її ефективності з методом FAIR. Вихідні дані до розрахунку взято з роботи [14].Етап 1. На даному етапі вводяться терм-множини для опису базових множин стану веб-сайту та підмножини станів, що описуються природною мовою:

Повний комплект оцінки стану інформаційної безпеки в ІС розбитий на п'ять підмножин форми:

E1-підмножина станів "вкрай неблагополучний стан рівня захищеності інформації сайту";

E2 - підмножина станів "неблагополучний стан рівня захищеності інформації сайту";

E3 – підмножина станів "середньої якості стан рівня захищеності інформації сайту";

E4 – підмножина станів "щодо безпечного стану рівня захищеності інформації сайту";

E5 – підмножина станів "максимальний безпечний стан рівня захищеності інформації сайту".

Відповідний набір E повний набір ризиків загрози безпеці інформації G поділяються на 5 підгруп:

G1-підмножина "граничний ризик загрози рівню захищеності інформації сайту";

G2 – підмножина "високий ризик загрози рівню захищеності інформації сайту";

G3-підмножина "середній ризик загрози рівню захищеності інформації сайту";

G4-підмножина "низький ризик загрози рівню захищеності інформації сайту";

G5-підмножина "незначна загроза ризику рівню захищеності інформації сайту".

Припустимо, що G набуває значення від 0 до 1 за визначенням.

Для довільного окремого показника оцінки ІБ Xi повний набір його значень *Bi* розбивається на п'ять підмножин:

Bi1 - підмножина "дуже низький рівень індикатора Xi";

Bi2 - підмножина "низький рівень показника Xi";

Bi3 - підмножина "середній рівень показника Xi";

Bi4 - підмножина "високий рівень індикатора Xi";

Bi5 - підмножина "дуже високий рівень показника Xi".

Виконується додаткова умова узгодження множин B, E і G наступного виду: якщо всі ці показники в аналізі мають, згідно з класифікацією, рівень підмножини Bij, то прогнозний стан рівня захищеності ІБ кваліфікується як Ej, а рівень ризику загрози рівню захищеності інформації ІС - як Gj. На правильне визначення рівня важливості показника в системі оцінки та правильну кількісну класифікацію рівнів показників впливає виконання даної умови.

Етап 2. Побудуємо сукупність показників X = {Xi} у кількості N = 4, які, на думку експерта-аналітика, з одного боку, впливають оцінку ризику загрози рівню захищеності інформації сайту, з другого боку, оцінюють різні сторони ІБ сайту.

Таблиця 3.1 - Набір індикаторів X

Характеристика	поточне значення
X1	1.2
X2	0.7
X3	0.025
X4	0.004

Етап 3. Підсумовуємо за кожним показником рівень значущості для аналізу r_1. Для того, щоб можна було оцінити цей рівень, розмістити всі значення в порядку зменшення величини так, щоб дотримувалося правило:

$$r_1 \geq r_2 \geq \ldots \geq r_n \qquad (3.1)$$

Значимість і-го показника повинна визначатися за правилом Фішберна, при умові, що система показників може ставитись в порядку спадання їх значимості [17]:

$$r_i = \frac{1}{N} = \frac{1}{4} = 0.25 \quad (3.2)$$

Правило Фішберна доказує факт, що про показники значущості (1) нічого не відомо. Тоді оцінка (2) відповідає максимальній ентропії існуючої інформаційної невизначеності щодо об'єкта дослідження.

Етап 4. Будується класифікація поточного значення g фактора ризику G як критерій для розбиття цієї множини на підмножину:

Таблиця 3.2 -Значення показника g

Інтервал G	Набір підмножин
0.8 <g< 1	G1-підмножина "граничний ризик загрози рівню захищеності інформації сайту";
0.6 <g< 0.8	G2 – підмножина "високий ризик загрози рівню захищеності інформації сайту";
0.4 <g< 0.6	G3-підмножина "середній ризик загрози рівню захищеності інформації сайту";
0.2 <g< 0.4	G4-підмножина "низький ризик загрози рівню захищеності інформації сайту";
0 <g< 0.2	G5-підмножина "незначна загроза ризику рівню захищеності інформації сайту".

Етап 5. Побудуємо класифікацію поточних значень x показників X як критерій розбиття повного набору їх значень на підмножину типу Bij

Таблиця 3.3 - Поділ підмножини значень

Параметр	Критерії розбиття				
	Bi1	*Bi2*	*Bi3*	*Bi4*	*Bi5*
X1	*x1<0.02*	*0,02<x1<0,16*	*0,16<x1<0,84*	*0,84<x1<1*	*1<x1*
X2	*x2<0.02*	*0,02<x2<0,16*	*0,16<x2<0,84*	*0,84<x2<1*	*1 x2*
X3	*x3<0.02*	*0,02<x3<0,16*	*0,16<x3<0,84*	*0,84<x3<1*	*1<x3*
X4	*x4<0.02*	*0,02 x4<0,16*	*0,16<x4<0,84*	*0,84<x4<1*	*1<x4*

Етап 6. Оцінюється поточний рівень показників.

Таблиця 3.4 – Оцінка рівня показника

Значення показника	поточне значення
Дуже високий (VH)	*X1*>1
високий (H)	0.1<X2<1
середній (M)	0.01<X3<0.1
Низький (L)	0.001<X4<0.01
Дуже низький (VL)	<0.001

Етап 7. Класифікація поточних значень за критерієм таблиці 4. Результатом класифікації заноситься в таблицю 6.

$\lambda_{ij} = 1$, Якщо , і , коли значення не потрапляє у вибраний діапазон класифікації.$b_{i(j-1)} < x_i < b_{ij}\lambda_{ij} = 0$

Таблиця 3.5 – Результат класифікації

Параметр	Значення	Результат класифікації підмножин				
		Bi1	*Bi2*	*Bi3*	*Bi4*	*Bi5*
X1	*0.25*	0	0	0	0	1
X2	*0.25*	0	0	1	0	0
X3	*0.25*	0	1	0	0	0
X4	*0.25*	1	0	0	0	0

Стадія 8. Виконуємо арифметичні дії щодо оцінки прогнозованого рівня захищеності інформації ІС G:

$$G = \sum_{j=1}^{5} g_j \sum_{i=1}^{N} r_i \lambda_{ij} \quad (3.4)$$

де$g_j = 0.9 - 0.2(j-1)$ (3.5)

$G = 0.25 * 0.1 + 0.25 * 0.3 + 0.25 * 0.5 + 0.25 * 0.9 = 0.45$

Значення g відповідає підмножині "середній ризик загрози рівню захищеності інформації ІС".

Отриманий результат оцінки та прогнозування рівня захищеності ІБ сайту відповідає результатам досліджень, наведених у роботі [14].

Отже, у третьому розділі наведено опис сценаріїв використання веб-програми для оцінки рівня безпеки веб-сайтів. Описано основні компоненти системи у вигляді діаграм компонентів та діаграми розгортання. Описано алгоритм визначення рівня ризику безпеки веб-сайту.

3.4 Тестування розробленого програмного забезпечення

Для тестування розробленої інформаційної системи необхідний апаратно-програмний комплекс з такими вимогами:

Сервер бази даних

Процесор: Intel Pentium Xeon від 2GHz

RAM: 2048 Мб

Вільний дисковий простір: 72Gb.

Функціональне тестування – один із видів незалежного тестування ПЗ, за допомогою якого визначається, чи вирішує програмне забезпечення завдання, для яких воно було розроблене, чи відповідає воно потребам замовника/користувача.

Функціональне тестування займає ключову позицію у процесі забезпечення якості програмних продуктів.

У роботі розглядається функціональне тестування програмних компонентів веб-програми.

Розглянемо приклад тестового сценарію для тестування основного функціоналу веб-програми. Розглянемо тестові сценарії.

Назва:	Тест Вхід до системи адміністратора.	
Функція:	Вхід у систему під роллю адміністратора	
Дія	Очікуваний результат	**Результат тесту:** пройдено
Передумова:		
Зайти на сайт програми	Сторінка Авторизації відкрита (рис.4.1)	
Кроки тесту:		
Ввести логін та пароль (semenov/qwerty)	Поля логін та пароль заповнені (рис.4.2)	
Натисніть кнопку Увійти до системи	Якщо пароль та логін введені правильно, то відкриється сторінка Адміністратор, інакше системи видасть повідомлення про помилку (рис.4.3)	
Післяумова:		
Відкриється сторінка Адміністратор	Сторінка Адміністратор доступна для використання (рис.4.4)	

Оцінка рівня безпеки веб-сайту

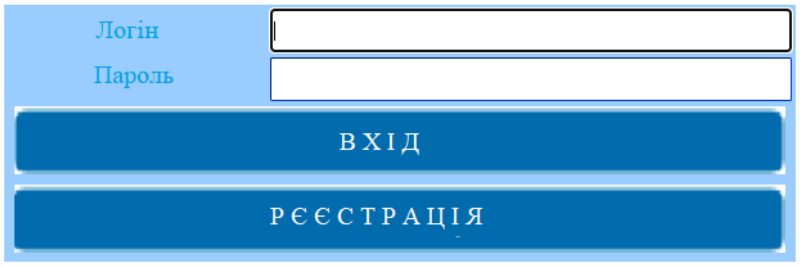

Рисунок 3.6- Сторінка Авторизації

Оцінка рівня безпеки веб-сайту

Логін petrov

Пароль ********

ВХІД

РЄЄСТРАЦІЯ

Рисунок 3.7 - Введення логіна та пароля

Оцінка рівня безпеки веб-сайту

Логін petrov

Пароль

ВХІД

РЄЄСТРАЦІЯ

Помилка введеня логіну та паролю!

Рисунок 3.8 - Помилка введення логіну та паролю

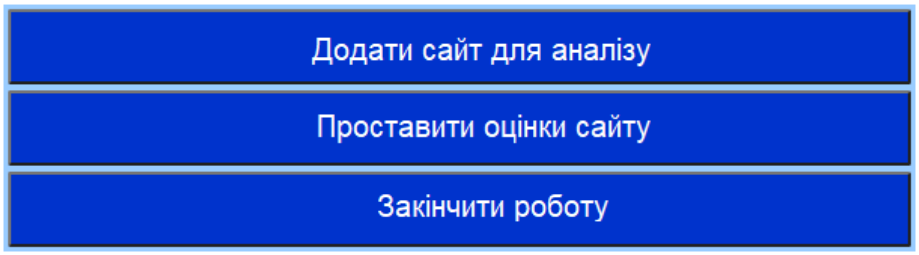

Рисунок 3.9 – Форма Адміністратора

Назва:	Тест : Реєстрація нового користувача.	
Функція:	Додавання інформації про нового користувача до БД	
Дія	Очікуваний результат	**Результат тесту:** – пройдено – ~~провалений~~ – ~~заблокований~~
Передумова:		
Зайти на сторінку реєстрації	Відкрито сторінку Реєстрації (рис.4.5)	
Кроки тесту:		
Ввести дані нового користувача (рис.4.6)	Заповнено всі поля реєстраційної форми	
Натисніть кнопку Реєстрація нового користувача	Якщо дані збережені успішно, з'явиться повідомлення (рис.4.7)	
Постуслів'я:		
Користувач може входити до системи		

Оцінка рівня безпеки веб-сайту

ПІБ
Адреса
Телефон
Логін
Пароль
Пароль ще раз
Реєстрація

Рисунок 3.10 - Сторінка Реєстрації

Рисунок 3.10 -Введення даних на сторінці Реєстрації

Оцінка рівня безпеки веб-сайту

Реєстрація прошла успішно

Повернутись

Рисунок 3.11 - Збереження даних

Отже, у третьому розділі розглянуто процес тестування програмної реалізації веб-програми для онлайн оцінки рівня безпеки сайтів. Розглянуто тест-кейси для основних функцій веб-програми. Наведено список для тестування нефункціональних вимог сайту та графічного інтерфейсу.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ

4.1 Організація служби охорони праці на підприємстві

Закон України «Про охорону праці» передбачає, що роботодавець зобов'язаний створити на робочому місці умови праці та забезпечити додержання вимог законодавства щодо прав працівників у галузі охорони праці. З цією метою роботодавець забезпечує функціонування системи управління охороною праці та несе безпосередню відповідальність за порушення вимог з охорони праці на підприємстві.

На підприємстві з кількістю працюючих менше 50 осіб функції служби охорони праці можуть виконувати в порядку сумісництва особи, які мають виробничий стаж не менше трьох років і пройшли навчання з охорони праці.

На підприємстві з кількістю працюючих 50 і більше осіб роботодавець створює службу охорони праці відповідно до типового положення, що затверджується центральним органом виконавчої влади, що забезпечує формування державної політики у сфері охорони праці.

На підприємстві з кількістю працюючих менше 20 осіб для виконання функцій служби охорони праці можуть залучатися сторонні спеціалісти на договірних засадах, які мають відповідну підготовку.

Служба охорони праці підпорядковується безпосередньо роботодавцю. Керівники та спеціалісти служби охорони праці за своєю посадою і заробітною платою прирівнюються до керівників і спеціалістів основних виробничо-технічних служб.

Спеціалісти служби охорони праці мають право:

– видавати керівникам структурних підрозділів підприємства обов'язкові для виконання приписи щодо усунення наявних недоліків, одержувати від них необхідні відомості, документацію і пояснення з питань охорони праці;

– вимагати відсторонення від роботи осіб, які не пройшли передбачених законодавством медичного огляду, навчання, інструктажу, перевірки знань і не мають допуску до відповідних робіт або не виконують вимог нормативно-правових актів з

охорони праці;

– зупиняти роботу виробництва, дільниці, машин, механізмів, устаткування та інших засобів виробництва у разі порушень, які створюють загрозу життю або здоров'ю працюючих;

– надсилати роботодавцю подання про притягнення до відповідальності працівників, які порушують вимоги щодо охорони праці;

– за поліпшення стану безпеки праці вносити пропозиції про заохочення працівників за активну працю.

Припис спеціаліста з охорони праці може скасувати лише роботодавець. Ліквідація служби охорони праці допускається тільки у разі ліквідації підприємства чи припинення використання найманої праці фізичною особою.

Фінансування охорони праці здійснюється роботодавцем. Фінансування профілактичних заходів з охорони праці, виконання загальнодержавної, галузевих та регіональних програм поліпшення стану безпеки, гігієни праці та виробничого середовища, інших державних програм, спрямованих на запобігання нещасним випадкам та професійним захворюванням, передбачається, поряд з іншими джерелами фінансування, визначеними законодавством, у державному і місцевих бюджетах.

Для підприємств, незалежно від форм власності, або фізичних осіб, які відповідно до законодавства використовують найману працю, витрати на охорону праці становлять не менше 0,5 відсотка від фонду оплати праці за попередній рік.

На підприємствах, що утримуються за рахунок бюджету, розмір витрат на охорону праці встановлюється у колективному договорі з урахуванням фінансових можливостей підприємства, установи, організації.

Суми витрат з охорони праці, що належать до валових витрат юридичної чи фізичної особи, яка відповідно до законодавства використовує найману працю, визначаються згідно з переліком заходів та засобів з охорони праці, що затверджується Кабінетом Міністрів України.

Роботодавець зобов’язаний інформувати працівників або осіб, уповноважених на здійснення громадського контролю за дотриманням вимог нормативно-правових актів з охорони праці, та Фонд соціального страхування України про стан охорони праці, причину аварій, нещасних випадків і професійних захворювань і про заходи, яких вжито для їх усунення та для забезпечення на підприємстві умов і безпеки праці на рівні нормативних вимог.

Працівникам забезпечується доступ до інформації та документів, що містять результати атестації робочих місць, заплановані роботодавцем профілактичні заходи, результати розслідування, обліку та аналізу нещасних випадків і професійних захворювань і звіти з цих питань, а також до повідомлень, подань та приписів органів державного нагляду за охороною праці.

Органи державного управління охороною праці у встановленому порядку інформують населення України, працівників про реалізацію державної політики з охорони праці, виконання загальнодержавної, галузевих чи регіональних програм з цих питань, про рівень і причини аварійності, виробничого травматизму і професійних захворювань, про виконання своїх рішень щодо охорони життя та здоров’я працівників.

4.2 Працездатність людини-оператора.

Під працездатністю людини розуміють можливість її виконувати певну роботу з необхідною якістю та у встановлений час.

Здатність виконувати задану роботу, має такі рівні:

– резервний - здатність працювати в умовах, що вимагають граничної мобілізації всіх фізичних і духовних сил. Природно, що людина в таких умовах не може працювати не тільки постійно, але і будь-яке тривалий час;

– актуальний (актуалізований). Він відноситься до повсякденної трудової діяльності з виконанням вимог певної професії.

Працездатність людини залежить як від зовнішніх факторів, так і від внутрішнього стану (внутрішні фактори).

До зовнішніх факторів належать: кількість та форма отриманої інформації, зручність робочого місця, характер взаємовідносин в колективі, вплив факторів середовища існування.

До внутрішніх факторів належать: рівень підготовки, тренованість людини та її емоційна стійкість.

Розглядаючи зміни функціонального стану та якості роботи людини у процесі одного трудового циклу (зміни), виділяють 4 фази працездатності: пристосування до праці. стійкої працездатності, субкомпенсації, втоми. Тривалість усіх фаз та усього циклу роботи залежить від рівня підготовки людини до роботи.

Фаза пристосування до праці - це час, протягом якого людина адаптується до умов праці. Основний показник ефективності праці поступово досягає свого встановленого значення. Тривалість періоду пристосування організму до умов праці залежить від багатьох факторів, серед яких основними є інтенсивність роботи (чим інтенсивніша робота, тим цей період коротший) та рівень готовності людини до майбутньої роботи.

Фаза стійкої працездатності характеризується найвищою якістю праці при оптимальних рівнях функціонування фізіологічних систем організму. Тривалість цього періоду залежить від інтенсивності роботи. Чим інтенсивніша праця, тим коротший цей період. Найоптимальніша динамічна робота, коли цей період може бути в десятки разів довшим, ніж при статичній діяльності.

На процес стійкої працездатності впливають емоції. Негативні **(**страх**,** невпевненість, поганий настрій) знижують працездатність. Позитивні (впевненість, спокій, бадьорий настрій) значно продовжують період стійкої працездатності.

Продовження періоду стійкої працездатності можна забезпечити:

- оптимальним рівнем напруги психофізіологічних функцій;
- комфортними умовами праці;
- правильним поєднанням режимів праці та відпочинку;
- емоційним розвантаженням;

– використанням тонізуючих напоїв (кава, чай), фармакологічних засобів, зокрема препаратів рослинного походження (вітаміни, препарати, які впливають на енергетичні та метаболічні процеси);

– інформуванням людини про наслідки її діяльності, наглядом та контролем її роботи.

Фаза субкомпенсації розглядається як початок розвитку втоми. В цей період якість праці ще зберігається на високому рівні, але тільки за рахунок перенапруги відповідних функцій організму.

Фаза втоми характеризується чітко вираженим зниженням якості роботи при подальшому погіршенні функціонального стану людини. Об'єктивними показниками втоми є зміна частоти пульсу, дихання, зорової та слухової чутливості.

Наступною фазою життєдіяльності людини повинна бути фаза відновлення працездатності (відпочинку), яка може тривати від декількох хвилин до декількох годин і навіть декілька діб.

Працездатність людини обумовлена його здатністю до нагромадження енергетичних резервів організму та психіки людини. Межа працездатності - величина змінна. Вона залежить від багатьох чинників: типу нервової системи, загального здоров'я, кваліфікації, мотивації, співвідношення праці та відпочинку, умов робочого середовища та ін.

Знання закономірностей динаміки працездатності дозволяє оптимізувати процес зростання працездатності людини з урахуванням специфіки його діяльності, психофізіологічного стану та індивідуальних особливостей.

ВИСНОВКИ

У роботі розглянуто актуальну проблему проєктування та розробки веб-додатка з оцінки рівня інформаційної безпеки веб-сайтів.

Метою роботи було проєктування та розробка веб-програми для проведення аналізу безпеки веб-сайтів

Для досягнення поставленої мети, у роботі, були сформульовані та вирішені наступні завдання:

- Виконано аналіз предметної галузі, пов'язаної з оцінкою вразливостей веб-додатків та методів їх виявлення;

- Виконано аналіз інструментів тестування на проникнення та інструментів моделювання проникнення;

- Виконано проєктування та розробку веб-додатка для проведення аналізу безпеки веб-сайтів;

- Виконано тестування розробленого веб-додатку

Робота складається зі вступу, 4-х розділів та списку використаних літературних джерел.

У вступі описано актуальність роботи, виділено об'єкт та предмет дослідження, сформульовано мету та завдання необхідні для досягнення поставленої мети.

У першому розділі проведено аналіз проблемної галузі. Проведений аналіз дозволив сформулювати основні вимоги до веб-застосунку з оцінки вразливостей веб-сайтів та оформити ці вимоги у вигляді технічного завдання.

У розділі розглянуті та вибрані інструменти для розробки та створення веб-програми. Як СУБД пропонується використання My SQL, C# обраний мовою програмування. Для розробки веб0сайту пропонується середовище розробки Visual Studio Code.

У третьому розділі наведено опис сценаріїв використання веб-програми для оцінки рівня безпеки веб-сайтів. Описано основні компоненти системи у вигляді діаграм компонентів та діаграми розгортання. Описано алгоритм визначення рівня ризику безпеки веб-сайту. Також розглянуто процес тестування

програмної реалізації веб-програми для онлайн оцінки рівня безпеки сайтів. Розглянуто тест-кейси для основних функцій веб-програми. Наведено список для тестування нефункціональних вимог сайту та графічного інтерфейсу.

СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Петров К. Е., Кобзев І. В., Онищенко Ю. М. Політика безпеки web-застосувань та серверів //Вісник Харківського національного університету внутрішніх справ. – 2015. – №. 2. – С. 256-263.

2. Климаш М. М., Шпур О. М., Пелех Н. В. МОНІТОРИНГ ДОСТУПНОСТІ ВЕБ-СЕРВІСУ В РОЗПОДІЛЕНИХ ІНФОКОМУНІКАЦІЙНИХ СИСТЕМАХ //Вісник Університету «Україна» Серія Інформатика, обчислювальна техніка та кібернетика. – 2020. – Т. 1. – №. 28..

3. Бакумов К. О. Методологія тестування на проникнення корпоративної мережі закладу вищої освіти //Архів кваліфікаційних робіт. – 2020.

4. Єрьоменко Н. Н., Кокоулін А. Н. Дослідження методів тестування на проникнення в інформаційних системах // Master's Journal. - 2016. - №. 2. - С. 181-186.

5. Web Applications vulnerabilities and threats: statistics for 2019 – [Електронний ресурс] – Режим доступу: https://www.ptsecurity.com/ww-en/analytics/web-vulnerabilities-2020/

6. Леванова А. С., Рожнова Н. С. Тестування на проникнення та його роль інформаційної безпеки //Слов'янський форум. - 2016. - №. 1. - С. 86-90.

7. Kalchenko V. Огляд методів проведення тестування на проникнення для оцінки захищеності комп’ютерних систем //Системи управління, навігації та зв’язку. Збірник наукових праць. – 2018. – Т. 4. – №. 50. – С. 109-114.

8. Яцюк О. О., Федюшин О. І. Дослідження методологій тестування на проникнення : дис. – ХНУРЕ, 2021.

9. Барибін О. І. Методологія тестування на проникнення веб-сайту закладу вищої освіти //Стандартизація. Сертифікація. Якість. – 2019. – №. 4. – С. 12-18.

10. Бакумов К. О. Методологія тестування на проникнення корпоративної мережі закладу вищої освіти //Архів кваліфікаційних робіт. – 2020.

11. Halton W. et al. Penetration Testing: A Survival Guide. – Packt Publishing Ltd, 2017.

12. Порошин С. М., Можаєв О. О., Можаєв М. О. Методологія проведення pen-тестування веб-додатків //Системи обробки інформації. – 2016. – №. 3. – С. 33-35..

13. Chiem T. P. A study of penetration testing tools and approaches : дис. – Auckland University of Technology, 2014..

14. Лапоніна О. Р., Малаховський С. А. Використання сканера вразливостей ZAP для тестування веб-додатків // International Journal of Open Information Technologies. - 2017. - Т. 5. - №. 8.