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 магістра

студента ***Лупашку Сергія Вікторовича***

академічної групи ***125м-22з-1***

спеціальності ***125 Кібербезпека***

спеціалізації[1]

за освітньо-професійною програмою ***Кібербезпека***

на тему ***Технічні аспекти розслідування кіберінцидентів***

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи:				
розділів:				
спеціальний				
економічний				

Рецензент:				

Нормконтролер:				

Дніпро
2023

ЗАТВЕРДЖЕНО:
завідувач кафедри
безпеки інформації та телекомунікацій
_____д.т.н., проф. Корнієнко В.І.
«_____»_____20___року

ЗАВДАННЯ

на кваліфікаційну роботу ступеня магістра

студенту Лупашку С.В. академічної групи 125м-22з-1

спеціальності *125 Кібербезпека*

спеціалізації[1]

за освітньо-професійною програмою *Кібербезпека*

на тему *Технічні аспекти розслідування кіберінцидентів*

Затверджену наказом ректора НТУ «Дніпровська політехніка» від ____ № ____

Розділ	Зміст	Термін виконання
Огляд літератури		**07.09.23-15.09.23**
Технічні аспекти кіберінцидентів		**15.09.23-08.10.23**
Методи розслідування		**09.10.23-17.11.23**
Кібербезпека та захист від кіберінцидентів		**17.11.23-28.11.23**
Практичні приклади		**29.11.23-30.11.23**
Висновки та рекомендації		**01.12.23-13.12.23**

Завдання видано _____ _____

Дата видачі завдання: _____

Дата подання до екзаменаційної комісії _____

Прийнято до виконання _____ _____

РЕФЕРАТ

Пояснювальна записка: 64 с., 21 рис., 8 табл., 18 джерел.

Об'єкт дослідження: Технічні аспекти розслідування кіберінцидентів.

Мета роботи: Дослідити технічні аспекти розслідування кіберінцидентів з метою виявлення і аналізу потенційних загроз та вироблення ефективних методів їх протидії.

Методи розробки: У даному дослідженні застосовано аналіз наукової літератури, вивчення статистичних даних про кіберінциденти, використання технічних засобів для розслідування, моделювання сценаріїв кібератак та експертну оцінку.

У першому розділі було проведено детальний аналіз переважаючих типів кіберінцидентів, виявлено зв'язок між методами атак і потенційною шкодою, що може бути заподіяна.

У спеціальній частині було визначено ключові етапи кіберрозслідування, включаючи збір електронних доказів, аналіз мережевого трафіку та використання спеціалізованого програмного забезпечення.

В економічному розділі визначено вплив кіберінцидентів на економіку, зокрема збитки компаній, витрати на відновлення та важливість вчасного розслідування для зменшення втрат.

Наукова новизна роботи полягає у поєднанні технічних аспектів розслідування кіберінцидентів з актуальними даними про загрози та використанням сучасних методів аналізу.

Практичне значення роботи полягає у наданні рекомендацій та розробці проактивних підходів до розслідування кіберінцидентів з метою посилення кібербезпеки та захисту від потенційних загроз.

Ключові слова: кіберінциденти, розслідування, технічні аспекти, методи, інструменти, кібербезпека.

ABSTRACT

Explanatory Note: 64 pages, 21 figures, 8 tables, 18 sources. Research Object: Technical aspects of cyber incident investigation.

Objective: To explore the technical aspects of cyber incident investigation, aiming to identify and analyze potential threats and develop effective methods to counteract them.

Research Methods: The study employs a literature review analysis, examination of statistical data on cyber incidents, the use of technical tools for investigation, simulation of cyber attack scenarios, and expert evaluation. In the first chapter, a detailed analysis of prevailing types of cyber incidents was conducted, establishing a connection between attack methods and potential harm inflicted.

The specialized section identified key stages of cyber investigation, including electronic evidence collection, network traffic analysis, and the use of specialized software tools.

The economic section determined the impact of cyber incidents on the economy, including company losses, recovery expenses, and the importance of timely investigations to minimize losses.

The scientific novelty of the work lies in the integration of technical aspects of cyber incident investigation with up-to-date threat data and the utilization of modern analysis methods.

The practical significance of the study lies in providing recommendations and developing proactive approaches to cyber incident investigation, aiming to enhance cybersecurity and protect against potential threats.

Keywords: cyber incidents, investigation, technical aspects, methods, tools, cybersecurity.

СПИСОК УМОВНИХ СКОРОЧЕНЬ

ІА - Інформаційна Атака

КІ - Кіберінцидент

СЗІ - Система Захисту Інформації

ФІШ - Фішинг

ВІШ - Вішинг

АТО - Аналіз Технічних Ознак

РТЗ - Реагування на Технічні Загрози

ДФЗ - Деталізоване Фахове Зведення

СОА - Системи Оперативного Аналізу

ТБЗ - Технічні Безпекові Заходи

АПТ - Адвансед Персистент Треат

ІДЗ - Інцидентний Детектор Загроз

СІМ - Система Управління Інцидентами та Безпекою

СТО - Сервіс Тривожного Оповіщення

ПЗ - Програмне Забезпечення

ОЗ - Організаційні Заходи

ІнтС - Інтелектуальна Система

ДЗЗ - Деталізоване Завдання Захисту

ЕПЗ - Електронна Підписова Завада

ТМІ - Технічний Моніторинг Інцидентів

СРМ - Система Реагування на Мережеві Загрози

ЛО - Логічне Озброєння

ЕУМ - Електронна Угода Мережі

ТЗ - Технічні Заходи

ЗМІСТ

ВСТУП

Останнім часом глобальна кіберсфера все частіше розглядається світовою спільнотою як один із найважливіших пріоритетів безпеки, оскільки її функціонування стає важливим фактором розвитку військової, соціальної, економічної та інших галузей. Все більш очевидною стає зростаюча мілітаризація кіберпростору, а зусилля світових держав щодо запобігання цьому процесу залишаються, на жаль, малоефективними.

Ці механізми поки що перебувають у стадії формування. Деякі їх потребують вдосконалення, але розвитку більшості та його окремих елементів насамперед бракує концептуального обгрунтування. Крім того, як і раніше, відсутні найважливіші елементи національної системи кібербезпеки від кібератак і розробка заходів щодо захисту від таких атак.

Актуальність даної теми обумовлена необхідністю подолання протиріччя між нинішнім станом наростаючих питань кібербезпеки та далеко не повною готовністю реагувати на нові виклики кібербезпеки від кібератак та розробка заходів щодо захисту від таких атак.

Аналіз останніх наукових досліджень про. Дослідженням питання забезпечення безпеки часто займаються вітчизняні та зарубіжні вчені, зокрема: Л. Абалкін, О. Ареф'єва, І. Бінько, О. Власюк, Г. Пастернак Таранушенко, З. Варналій, Т. Васильцев, З. Герасимчук, З. . Живко, О. Захаров, В. Ковальов, Г. Козаченко, О. Ляшенко, В. Мунтіян, Є. Олійников, В. Пономаренко, М. Єрмошенко, Я. Жаліло, В. Франчук, Л. Шемаєва, С. Шкарлет, В. Шлемко, В. Ярочкін та ін. Проте доцільно відзначити недостатню увагу вчених щодо можливості персоналу підприємств критичної інфраструктури до протидії кіберзагрозам та формуванню відповідного теоретико-методологічного базису її забезпечення у військових умовах.

Мета роботи полягає в обґрунтуванні теоретико-методичних положень та розробці практичних рекомендацій щодо забезпечення корпоративної безпеки,

зокрема, шляхом визначення сукупності ключових зовнішніх та внутрішніх кіберзагроз.

Для досягнення цієї мети було поставлено такі завдання:

– аналіз проблем інформаційної військової безпеки критичної інфраструктури;

– огляд існуючих методологій та стандартів з управління ризиками інформаційної безпеки, порівняльний аналіз, виявлення недоліків та переваг;

– визначення релевантного для військової критичної інфраструктури підходу до оцінки ризиків інформаційної безпеки;

– оцінка ризиків інформаційної безпеки, що належить до критичної інфраструктури;

– визначення рекомендацій щодо зниження ризиків високого рівня.

Методами дослідження обрано: опрацювання літератури на цю тему, а також аналіз документації міжнародних стандартів.

Наукова новизна цієї роботи полягає в адаптації методів управління ризиками інформаційної безпеки та визначенні оптимального підходу до оцінки ризиків. Визначення такого підходу передбачає успішнішу протидію сучасним кіберзловмисникам, забезпечення безпеки, орієнтованої на захист від кіберзагроз шляхом впровадження превентивних засобів захисту.

Практичне значення результатів роботи випливає з можливості використання даного підходу для побудови системи управління інформаційною безпекою, що ґрунтується на процесі управління ризиками інформаційної безпеки, а також на основі використання даного підходу для оцінки ризиків інформаційної безпеки провести обробку ризиків для зменшення їх до прийнятного рівня.

Таким чином, об'єктом дослідження є інформаційна кібербезпека.

Предмет досліджень - методи та підходи до оцінки здатності персоналу підприємств критичної інфраструктури до протидії кібератак.

Методологія Теоретико-методологічною основою проведеного дослідження стали загальнонаукові методи наукового пізнання: монографічне, теоретичне узагальнення, систематизація, аналіз та синтез.

1 СТАН ПИТАННЯ. ПОСТАНОВКА ЗАДАЧІ

1.1 Завдання роботи та методологія

1.1.1 Завдання роботи

У даній роботі необхідно проаналізувати основні види сучасних кібератак та розробити заходи захисту від таких атак. За цим показником сьогодні виділяють шість видів кібератак.

Необхідно провести дослідження за єдиним алгоритмом:

1. З'ясувати мету атаки, яку ставить собі зловмисник.

2. Виявити джерело небезпеки. внутрішнє чи зовнішнє. Під зовнішньою загрозою розумітимемо безпосередньо атаку з Інтернету. Під внутрішньою загрозою розумітимемо можливість здійснення атаки по локальній мережі.

3. Виявити та описати вразливості організації ІКМ, які зловмисник використовує для атаки.

4. Визначити програми, з допомогою яких зловмисник може її здійснити.

5. Визначити об'єкти атаки, тобто сегмент мережі або програмне забезпечення, на яке спрямована атака.

6. Виявити описані типові сучасні заходи нейтралізації цих атак.

1.1.2 Методологія

Основними елементами у сфері кібербезпеки в контексті загроз є:

1. Кіберпростір

2. Кібербезпека

3. Кіберзагрози

4. Кіберризик

Саме визначення цих термінів є ключовим моментом у вирішенні термінологічної проблеми у цій сфері та створенні ефективних механізмів протидії кіберзлочинності. Незважаючи на те, що ці поняття використовуються в законодавчій та професійній практиці різних держав, їх зміст досі залишається

нечітким, що також створює нормативну проблему та ускладнює формування відповідних державних документів, які мають визначати підходи до проблем кібербезпеки.

Під інформаційною безпекою ми розуміємо захищеність інформації та інфраструктури, що її підтримує, від випадкових або навмисних впливів природного чи штучного характеру, які можуть завдати неприйнятної шкоди суб'єктам інформаційних відносин, зокрема власникам і користувачам інформації та допоміжної інфраструктури.

Захист інформації - це комплекс заходів, спрямованих на забезпечення безпеки інформації.

Кібератака – це спроба реалізації кіберзагрози, тобто отримання несанкціонованого доступу до комп'ютерного обладнання чи комп'ютерної мережі з метою отримання даних або з наміром заподіяти шкоду. Кіберзагрози мають широкий спектр атак, починаючи від крадіжки персональних даних, маніпулювання ними, запуску вірусів у мережу, порушення цифрового добробуту та стабільності бізнес-підприємств і підприємств критичної інфраструктури держав [1].

Насправді кібератаки впливають на кожен аспект нашого життя. Вони можуть призвести до відключення електроенергії, знищення військової техніки та викрадення конфіденційних даних, порушуючи національну безпеку країни.

Людей, які здійснюють такі атаки, називають хакерами, кіберзлочинцями. Злочинці можуть діяти поодинці або групами. Вони також можуть стосуватися злочинних угруповань, які працюють з іншими загрозливими суб'єктами, щоб точніше боротися зі слабкими місцями в комп'ютерних мережах.

ІТ-підрозділи, що фінансуються державою, також можуть здійснювати кібератаки. Тоді їх ідентифікують як злочинців, які загрожують національній безпеці держави.

Кібератаки призначені для заподіяння шкоди та мають різні цілі, а саме:

– фінансовий – отримання фінансової вигоди. Здійснюється проти комерційних підприємств і фізичних осіб, спрямований на викрадення

персональних даних і номерів кредитних карток, які потім використовуються для доступу до грошей;

– помста – здійснюється, щоб посіяти хаос і плутанину, розчарування та недовіру. Прикладом таких груп є Anonymous. Їх вважають кіберактивістами, які борються за справедливість і національні інтереси жертв;

– кібервійна – в таких атаках беруть участь усі країни світу.

Кібервійни відбуваються в рамках економічних, політичних чи соціальних суперечок [2].

Найпоширеніші види кібератак включають:

– Відмова в обслуговуванні (DoS-атака) — мережева атака, під час якої компоненти комп'ютерної системи перевантажуються та намагаються зробити пристрій недоступним для користувачів.

– Фішинг – це атака, яка використовує соціальну інженерію для викрадення особистої інформації шляхом створення копій відомих сайтів.

– Malware – це поширена кібератака, яка передбачає запуск шкідливого програмного забезпечення (вірусів, троянів) всередині комп'ютера, яке виконує несанкціоновані дії в системі жертви.

– Ransomware – це атака, яка запускає шкідливе програмне забезпечення всередині комп'ютера, яке блокує та шифрує дані жертви або створює їх копію для подальшого шантажу.

– Man-in-the-Middle — мережева атака, під час якої злочинець отримує дані, що передаються між двома користувачами, і замінює їх фіктивними даними.

– Zero-day exploit – це атака, яка здійснюється на вразливості ліцензійного програмного забезпечення, які ще не відомі розробнику.

– Міжсайтовий скриптинг (XSS) – це атака, яка здійснюється шляхом додавання небезпечного коду на веб-сайт. При використанні такого сайту дані користувача можуть передаватися хакерам.

– Logic bombs – це атака, яка містить набір інструкцій у програмі, яка несе зловмисне корисне навантаження та може атакувати операційну систему [3].

1.2 Поточний стан знань із проблеми

1.2.1 Аналіз статистики кібератак. Класифікація атак і вразливостей

Зараз велика увага приділяється інформаційній безпеці персональних даних. Деякі компанії аналізують статистику атак на персональні дані. У цьому розділі роботи ми розглянемо наступну статистику, а також перерахуємо основні типи атак для досягнення військових цілей.

Як зазначається в [1], основні результати цих досліджень, що характеризують рівень забезпеченості закладів освіти, свідчать про те, що:

– у 77% випадків зовнішнє тестування на проникнення виявило можливість здійснення атак з метою отримання доступу до внутрішніх корпоративних ресурсів шляхом використання вразливостей навчальних закладів;

– 30% кіберінцидентів пов’язані з атаками на веб-ресурси.

Усі ці дані свідчать про те, що навчальні заклади є однією з головних «мішеней» для зловмисників, адже велика кількість невиправлених уразливостей і простота їх експлуатації допомагають зловмисникам успішно досягати поставлених цілей — від викрадення інформації до доступу до внутрішніх ресурсів локального комп’ютера. мережі [1].

Важливо розуміти, що більшість уразливостей можна виявити задовго до атаки, а аналіз вихідного коду веб-сайтів може виявити в рази більше критичних уразливостей, ніж тестування систем без перевірки коду.

Основні результати досліджень, спрямованих на аналіз захисту персональних даних з точки зору інформаційної безпеки, показують, що [1,2]:

1. Усі персональні дані студентів навчального закладу є вразливими. За результатами автоматизованого аналізу вихідного коду було виявлено, що всі навчальні заклади мають уразливості, і лише 6% досліджуваних систем не мають уразливостей високого ризику.

2. Основна ціль – користувачі сайту. 85% перевірених веб-додатків мають уразливості, які дозволяють атакувати користувачів. Використовуючи дані про вразливості, зловмисник може викрасти файли cookie користувачів, провести

фішингові атаки або заразити їхні робочі станції шкідливим програмним забезпеченням.

3. Результати зумовлені тим, що програми навчальних закладів мають більш складну логіку роботи порівняно з інформаційними веб-ресурсами інших організацій, і цей фактор створює передумови для появи більшої кількості високоризикових уразливостей. Використання цих уразливостей може дозволити зловмиснику спробувати порушити роботу програми або виконати довільний код у цільовій системі, що може призвести до повного контролю над сервером, на якому розміщено веб-сайт.

4. Усі перевірені школи можна використовувати для атак на користувачів. Усі досліджені державні установи мають уразливості, які дозволяють атакувати користувачів. Важливо відзначити, що переважна більшість користувачів таких веб-ресурсів дуже погано обізнані в інформаційній безпеці і легко можуть стати жертвами зловмисників.

5. Відмова в обслуговуванні є найпоширенішою загрозою для інтернет-магазинів. 75% перевірених веб-сайтів електронної комерції мали вразливості, які потенційно могли призвести до відмови в обслуговуванні. Реалізація цієї загрози може призвести до значних фінансових втрат власників [1,2].

Хакерська атака - це комплекс дій, спрямованих на пошук вразливостей в цифрових системах. Водночас слід зазначити, що хакери не завжди займаються зловмисною діяльністю, але сьогодні термін «злом» зазвичай вживається в контексті незаконної діяльності, а хакери — це кіберзлочинці, які прагнуть отримати фінансову вигоду, висловити протест та збирати певну інформацію [3].

Автоматизований аналіз безпеки, представлений у [1], показав, що всі перевірені веб-сайти містили вразливості різного ступеня ризику. При класифікації вразливостей за рівнем ризику виявилося, що більшість із них (65%) належать до середнього рівня небезпеки.

Розподіл вразливостей за рівнем ризику показано на рис. 1.1

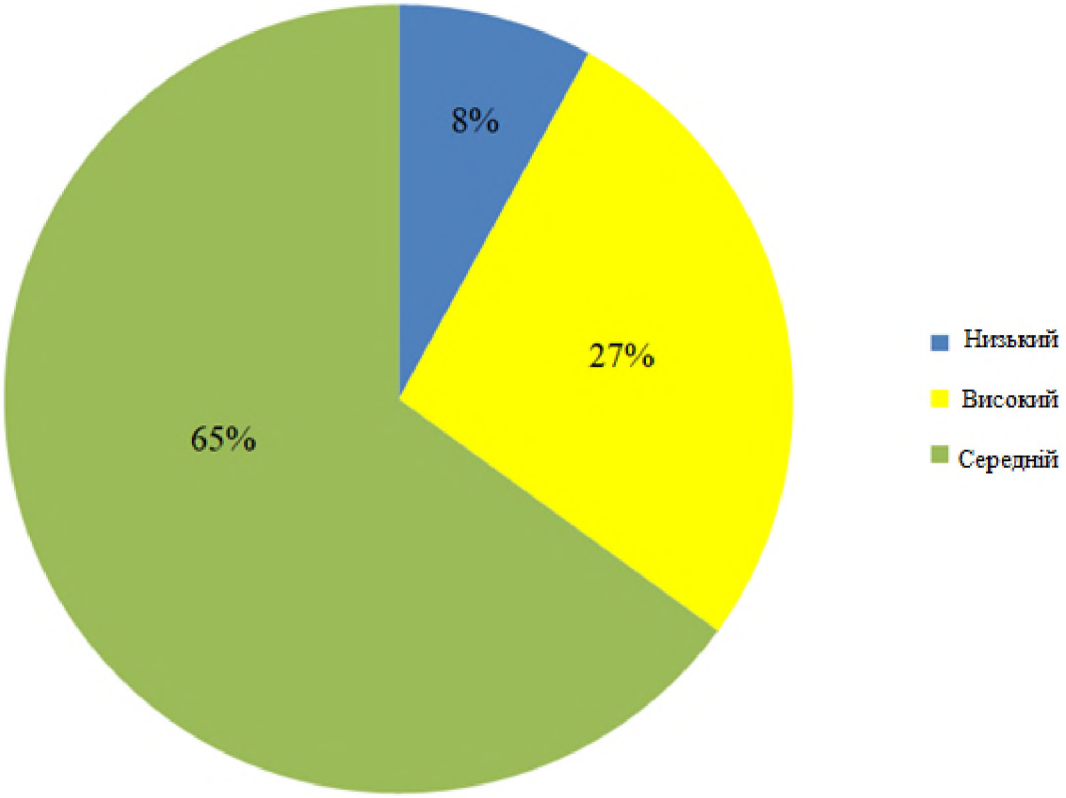

Рисунок 1.1 - Розподіл вразливостей за рівнем ризику

Джерело: власне

При оцінці систем на основі максимального рівня ризику виявлених уразливостей було виявлено, що лише 6% досліджуваних веб-сайтів не мали вразливостей високого ризику. Слід враховувати, що досліджувані сайти є типовими платформами CMS і містять велику кількість коду, написаного їх власниками. Але не варто забувати, що успішна експлуатація навіть однієї критично небезпечної вразливості на практиці може призвести до повної компрометації програми або навіть сервера (рис. 2.2).

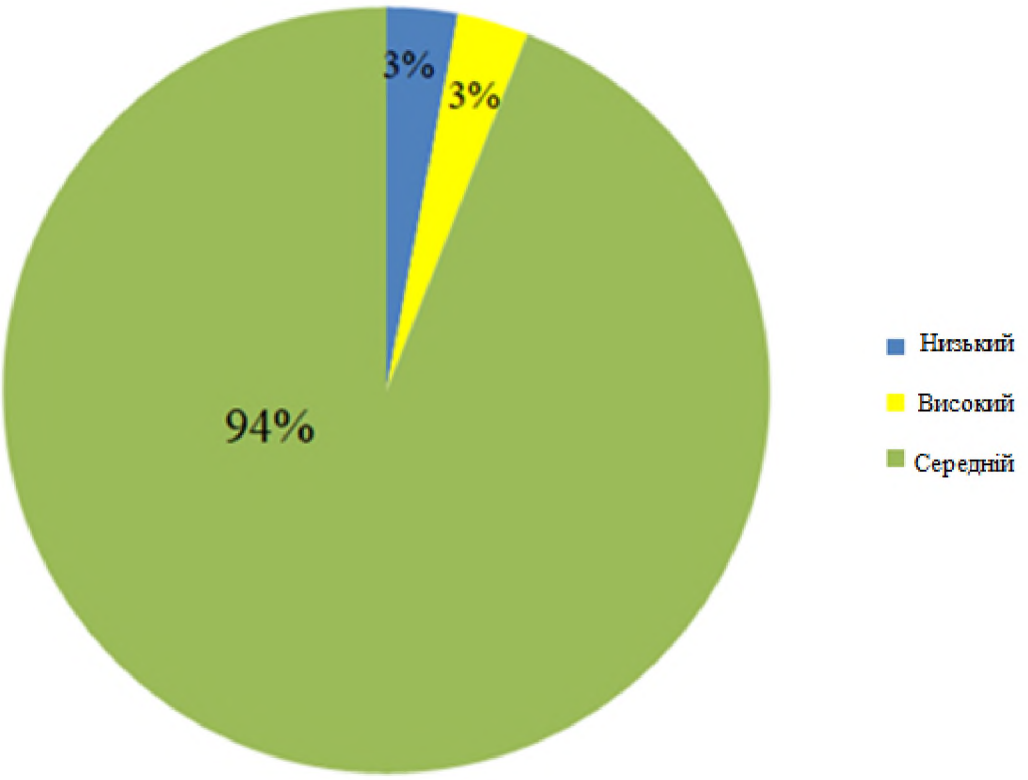

Рисунок 1.2 - Розподіл систем за максимальним рівнем ризику виявлених вразливостей

Джерело: власне

Найпоширеніша вразливість, яка виявилася під час автоматизованого аналізу вихідного коду веб-сайту та перерахована в аналізі [1], — це «міжсайтовий сценарій», який дозволяє зловмиснику здійснювати фішингові атаки на клієнтів веб-сайту або заражати їхні робочі станції шкідливим програмним забезпеченням. Ця уразливість також лідирує в аналогічному рейтингу за результатами ручного тестування веб-сайтів.

Друга найпоширеніша вразливість — «Розподіл відповіді HTTP». У разі успішного використання зловмисник може здійснювати атаки на клієнти веб-програми на основі надсилання браузером подвійної відповіді HTTP, атакуваний вміст заголовків і полів якого частково контролюється зловмисником.

На третьому місці — уразливість «Читання довільного файлу» з високим рівнем ризику, через яку зловмисник може отримати несанкціонований доступ до вмісту довільного файлу на сервері. Таким чином зловмисник може отримати

вихідний код веб-сайту, облікові дані та іншу інформацію, яка обробляється в системі.

З десяти найпоширеніших уразливостей п'ять мають високий рівень ризику, і їх використання може призвести до серйозних наслідків. Наприклад, у деяких випадках, в результаті використання уразливості Arbitrary File Creation, зловмисник може виконати довільний код у цільовій системі та згодом повністю скомпрометувати атакований сервер [1].

Консорціум безпеки веб-додатків (WASC) також класифікує вразливості веб-сайтів.

Консорціум WASC розробив спеціальний документ, який описує класифікацію вразливостей веб-сайтів - WASC Threat Classification [4].

Згідно з документом класифікації загроз WASC, уразливості веб-сайтів поділяються на такі етапи життєвого циклу програмного забезпечення:

– етап проектування – охоплює вразливості, які можуть виникнути через помилки в дизайні сайту;

– етап впровадження - охоплює вразливості, які можуть виникнути через помилки в реалізації компонентів сайту, наприклад, написання програмного коду

– фаза розгортання - охоплює уразливості, які можуть виникнути під час налаштування веб-сайту для роботи, наприклад, - неправильна конфігурація веб-сервера;

Помилки, які можуть бути зроблені на різних етапах життєвого циклу програмного забезпечення, описані в списку CWE (Common Weakness Enumeration), складеному корпорацією MITRE. Згідно з веб-сайтом корпорації MITER, CWE є базовим стандартом для виявлення слабких місць програмного забезпечення та запобігання їх появі [5].

Кожній ваді в списку присвоюється власний ідентифікатор (ID), список містить близько тисячі ідентифікаторів CWE. Під час опису вразливості в базі даних уразливостей ідентифікатор CWE може бути включений для додаткової інформації.

Розглянемо наступні типи атак на сайти [6]:

– вставка інструкцій (Injection) – інструкції SQL вставляються та надсилаються на обробку сайту, який після їх отримання може почати виконувати довільні команди;

– некоректна автентифікація (Broken Authentication) – функції автентифікації можуть бути реалізовані таким чином, що вони дозволяють обійти паролі або отримати ідентифікатори користувачів;

– витік критичних даних (Sensitive Data Exposure) – недостатня безпека персональних даних;

– атаки на аналізатори вхідних даних XML (XML Eternal Entities) – зловмисники можуть завантажувати XML-файли на сервер або включати шкідливий код у XML-документ;

– некоректний контроль доступу (Broken Access Control) - контроль доступу реалізований таким чином, що авторизовані користувачі можуть мати ті повноваження в системі, які вони повинні мати;

– небезпечна конфігурація середовища (Security Misconfiguration) – відсутність оновлень і неправильна конфігурація окремих компонентів веб-сайтів може становити додаткову загрозу безпеці;

– міжсайтовий скриптинг (XSS) – зловмисник може виконувати скрипти в браузері жертви, перехоплювати скрипти користувача та перенаправляти користувачів на інші веб-сайти;

– незахищений процес десеріалізації – зловмисник може порушити логіку веб-програми шляхом підробки програмних об’єктів, що призводить до віддаленого виконання коду зловмисника;

– використання компонентів із відомими вразливими місцями (Using Components with Known Vulnerabilities) – програмне забезпечення, термін підтримки якого вже минув або не було оновлено, що може залучити більше зловмисників до його злому;

– Недостатня реєстрація та моніторинг – погано організовані механізми реєстрації подій та моніторингу можуть призвести до того, що зловмисники зможуть неодноразово атакувати веб-сайти, не будучи виявленими.

Згідно з дослідженням Vercode, компанії з тестування безпеки програмного забезпечення, під час тестування на проникнення 74% програм містять принаймні одну вразливість зі списку Топ-10 OWASP [7].

1.2.2 Проблеми захисту персональних даних

Оскільки технології захисту персональних даних розвиваються, необхідно підтримувати надійні заходи безпеки. Загрози безпеці реальні й виникають по всьому світу. Стандартних заходів уже недостатньо для захисту від нових загроз.

З розвитком технологій постійно з'являються нові вектори атак. Комплексні засоби безпеки зараз потрібні як ніколи. У минулому захист кінцевих точок пристрою та мережева безпека були останніми засобами захисту. Потім з'явилися мобільні та хмарні технології, які суттєво знизили ефективність мережевої безпеки та значно підірвали горизонтально-орієнтований підхід до безпеки, на який так часто покладалися.

Зараз розвивається абсолютно новий підхід до доступу до бек-офісних систем, що відкриває нові можливості для бізнесу – як легальні, так і нелегальні. Зараз організації все більше покладаються на інтерфейси прикладного програмування (API) для впровадження інновацій, прискорення розробки та надання нових можливостей монетизації.

Але згідно з топ-10 OWASP: «За своєю природою API розкривають логіку програми та конфіденційні дані, такі як персональна інформація (PII), що робить API все більш поширеною мішенню для зловмисників». Переміщення критичних компонентів на клієнтську сторону сайтів (тобто поза захистом традиційної мережевої безпеки) створює велику нову поверхню для атак, збільшуючи ризик таких атак, як злом форм, підробка об'єктної моделі документа (DOM), зловживання сеансом, бомбардування, Зловживання API [9].

На жаль, багато організацій досі вважають, що брандмауер веб-додатків (WAF) — це все, що потрібно для захисту веб-сайтів від загроз безпеці. Насправді WAF є лише частиною рішення. Традиційні методи безпеки, такі як WAF, не можуть зупинити сучасні атаки на веб-програми, які в багатьох випадках походять із браузера поза сферою безпеки мережі. До того моменту, коли вони спрацьовують (якщо вони взагалі зникнуть), зловмисник уже не має конфіденційної інформації, і все, що можна зробити, це зосередитися на запобіганні збитку.

Оскільки зловмисники продовжують виявляти нові вектори атак, організації повинні реагувати відповідним чином, щоб забезпечити адекватний і комплексний захист критично важливих активів і даних.

Близько 95% веб-сайтів працюють на JavaScript і HTML5, мовах, які можна легко перехопити, сканувати та зламати. Це робить веб-програми та API вразливими до атак на стороні клієнта, особливо якщо вони покладаються лише на традиційні інструменти захисту периметра, такі як WAF. За даними Symantec, щомісяця зламують понад 4800 сайтів. Атаки з крадіжки даних на стороні клієнта є лише одним із векторів загроз, з якими стикаються веб-сайти, тому важливий багаторівневий підхід до безпеки [10].

Як видно з останніх зломів на веб-сайтах British Airways і Ticketmaster, навіть якби WAF був на місці та правильно налаштований, він не міг би запобігти цим зломам на стороні браузера/клієнта. Коли експлойти були виявлені, сотні тисяч записів клієнтів уже були вилучені.

Тому безпека є ключовою характеристикою розвитку захисту персональних даних. Щоб залишатися конкурентоспроможними на ринку, компанії повинні розробляти нові рішення безпеки для боротьби з хакерами та надавати надійні та безпечні програми своїм клієнтам.

Однак безпека особистих даних значною мірою залежить від того, чи розробники знають про кіберзагрози та регулярно відстежують діяльність програми.

1.2.3 Ієрархія захисту персональних даних

Зростання кількості та якості загроз інформаційній безпеці в комп'ютерних системах не завжди призводить до адекватного реагування з використанням надійної системи та безпечних інформаційних технологій. У більшості комерційних і державних організацій, не кажучи вже про звичайних користувачів, в якості засобів захисту використовуються тільки антивірусні програми і розмежування прав доступу користувачів за допомогою паролів.

Існує кілька способів захисту інформації.

Антивірусне програмне забезпечення є важливою частиною надійної програми безпеки. При правильній конфігурації ризик зараження шкідливим програмним забезпеченням значно знижується (хоча і не завжди - згадайте про вірус Melissa).

Але жодна антивірусна програма не може захистити організацію від зловмисника, який використовує легальну програму входу, або законного користувача, який намагається отримати несанкціонований доступ до файлів.

Будь-яка комп'ютерна система в організації обмежує доступ до файлів шляхом ідентифікації користувача, який увійшов у систему. Якщо система налаштована правильно, з встановленими необхідними дозволами для законних користувачів, існують обмеження на використання файлів, до яких вони не мають доступу. Однак система контролю доступу не забезпечить захист, якщо зловмисник через уразливості отримає доступ до файлів як адміністратор. Така атака розцінюватиметься адміністратором як законна дія.

Перший найпростіший і найважливіший. Основним засобом захисту тут є брандмауер. Брандмауер повинен обмежувати використання послуг, що надаються користувачам. Крім того, брандмауер повинен контролювати всі підключення з того чи іншого боку. Не слід використовувати програми невідомого походження; перевагу слід віддавати ліцензійному ПЗ.

Другий рівень безпеки включає налаштування операційної системи, під якою працює сайт. Кожна операційна система дозволяє створювати контрольні

списки безпеки. Ці налаштування мають відповідати операційним системам виробника. Також важливо, щоб новий сайт був своєчасно доданий до контрольного списку безпеки та не був вимкнений.

Третій рівень – мережевий. Необхідно оснастити мережеве обладнання та програмне забезпечення хостинг-провайдера датчиками атак. Головне, щоб вхідний сигнал про небезпеку був правильно оброблений і нейтралізований.

Четвертий рівень безпеки - встановлення програмного забезпечення на рівні хостингу. Це набагато складніше завдання. По-перше, тут ви можете зустріти заперечення з боку самої хостингової компанії. А по-друге, таке програмне забезпечення набагато складніше простих датчиків. З цієї причини рівень безпеки вище.

Рівень 5 має два підрівні - A і B. Рівень 5A - це встановлення спеціального програмного забезпечення, яке діє як прошарок між операційною системою веб-сервера та всіма сайтами. Цей буфер запобігає атакам хакерів на вразливі програми, які захоплюють контроль над усією операційною системою під час виконання. Це не найдешевший варіант, оскільки, як і раніше, потрібно забезпечити підтримку програмного забезпечення, яке запускає веб-сервери.

Рівень 5B – це встановлення програмних брандмауерів або проксі-серверів. Вони зосереджені на протоколі HTTP та допомагають запобігти атакам до того, як потенційні зловмисники зможуть дістатися до сайтів, запущених на веб-сервері. Однак наявність проксі-сервера є істотним обмеженням у роботі.

Шостий рівень – це свого роду вершина безпеки. Тут вам дозволено використовувати лише перевірені операційні системи та сайти, що працюють під їх контролем. Іншими словами, всі функціонуючі веб-сайти та операційні системи повинні бути максимально адаптовані або розроблені спеціально для конкретних потреб компанії. Це найдорожчий, але і найефективніший спосіб захисту. Крім того, це вимагає спеціального навчання адміністратора мережі, а часто і користувачів, що тягне за собою додаткові витрати.

1.3 Методи пошуку вразливостей. Перспективні методи захисту

Як уже зазначалося, сьогодні кібератаки стають найпопулярнішим інструментом досягнення військових цілей. Однак попередні дослідження показали, що багато пристроїв можуть бути вразливими до різноманітних уразливостей.

Під час тестування безпеки веб-додатків слід переглянути наступний неповний список функцій. Неправильна реалізація кожного з них може призвести до уразливості [11]:

– налаштування програми та сервера. Можливі дефекти пов'язані зі змінами шифрування/криптографії, конфігураціями веб-сервера тощо;

– перевірка введення та обробка помилок. Впровадження SQL, міжсайтовий сценарій (XSS) та інші поширені вразливості впровадження є результатом поганої обробки введення та виведення;

– аутентифікація та керування сесіями. Вразливі місця, які можуть призвести до видавання себе за іншу особу. Необхідно також розглянути розширення можливостей і захист;

– авторизація. Перевірка здатності програми захищатися від вертикальної та горизонтальної ескалації привілеїв.

– бізнес-логіка. Важливо для більшості програм, які забезпечують бізнес-функції.

– клієнтська логіка. У сучасних веб-додатках із інтенсивним використанням JavaScript, на додаток до веб-сторінок, які використовують інші типи клієнтських технологій (наприклад, Silverlight, Flash, Java-аплети), цей тип функціональності стає все більш поширеним.

Отже, тестування веб-безпеки спрямоване на виявлення вразливостей у веб-додатках та їх конфігураціях. Початковою метою є прикладний рівень (тобто те, що працює через протокол HTTP). Перевірка безпеки веб-програми часто передбачає надсилання різних типів вхідних даних, які викликають помилки та

викликають неочікувану поведінку системи. Ці так звані негативні тести перевіряють, чи робить система те, для чого вона не призначена [11].

Важливо також розуміти, що тестування веб-безпеки — це не лише перевірка функцій безпеки (таких як автентифікація та авторизація), які можуть бути реалізовані в програмі. Не менш важливо перевірити, чи безпечно реалізовано інші функції (наприклад, бізнес-логіка та використання належної перевірки вхідних даних і вихідного кодування). Мета полягає в тому, щоб забезпечити безпеку функцій, наданих у веб-додатку.

Згідно з дослідженням OWASP [8], найефективнішим способом виявлення вразливостей веб-додатків є експертний аналіз вихідних кодів веб-додатків (code review). Цей метод дуже трудомісткий, вимагає високої кваліфікації фахівців і не захищений від експертних помилок.

Тому також активно розробляються методи автоматичного виявлення вразливостей веб-додатків.

Методи автоматичного виявлення вразливостей сайту можна розділити на дві основні групи:

– методи аналізу роботи веб-сайту, розгорнутого на стенді, без доступу до вихідних кодів веб-сайтів;

– методи, що аналізують вихідні коди веб-сайту та налаштування конфігурації. Перша група методів розглядає сайти з точки зору зовнішнього користувача, тобто потенційного зловмисника.

Відповідно до класифікації, наведеної в [8,11], типи тестів безпеки веб-додатків поділяються на:

– динамічний тест безпеки програми (DAST). Цей автоматизований тест безпеки програми найкраще підходить для внутрішніх програм із низьким рівнем ризику, які мають відповідати нормативним вимогам безпеки. Для додатків із середнім рівнем ризику та критично важливих додатків, які зазнають незначних змін, найкращим рішенням є поєднання DAST із ручним тестуванням веб-безпеки на загальні вразливості.

– статичний тест безпеки додатків (SAST). Цей підхід до безпеки програми пропонує автоматичні та ручні методи тестування. Він краще підходить для виявлення помилок без запуску програм у робочому середовищі. Це також дозволяє розробникам сканувати вихідний код і систематично знаходити та виправляти вразливості безпеки програмного забезпечення.

– тест на проникнення. Цей ручний тест безпеки програми найкраще підходить для критично важливих програм, особливо тих, які зазнають серйозних змін. Оцінка включає бізнес-логіку та тестування зловмисників для виявлення складних сценаріїв атак.

– самозахист програми під час виконання (RASP). Цей підхід до розвитку безпеки програмного забезпечення включає в себе низку методів інструментарію програмного забезпечення, щоб атаки можна було відстежувати під час їх виникнення та, в ідеалі, блокувати в режимі реального часу.

Методи автоматизованого тестування будуть детально розглянуті в рамках магістерської роботи.

Виявивши основну причину вразливості, на ранніх етапах тестування можна впровадити засоби пом'якшення, щоб запобігти будь-яким проблемам. Крім того, знання про те, як працюють ці атаки, можна використовувати під час тестування безпеки веб-додатків.

Визнання впливу атаки також є ключовим для управління ризиками, оскільки результати успішної атаки можна використовувати для оцінки загальної серйозності вразливості. Якщо під час аудиту безпеки виникають проблеми, визначення їх серйозності дозволяє організації ефективно визначати пріоритети заходів з усунення.

Перш ніж виявити проблему, оцінка потенційного впливу на кожну програму в бібліотеці програм може допомогти визначити пріоритетність тестування безпеки програми. Маючи встановлений список високоякісних програм, можна запланувати тестування веб-безпеки, щоб спочатку націлити на критично важливі програми компанії з більш цілеспрямованим тестуванням для зменшення ризиків [12].

1.4 Використання сканерів та їх можливості

Для пошуку вразливостей використовуються спеціальні сканери вразливостей. Основні завдання сканерів уразливостей включають [13]:

– виявлення та аналіз загроз;

– інвентаризація ресурсів (операційна система, програмне забезпечення, мережеві пристрої);

– формування звітів з описом вразливостей і варіантами виправлення вразливостей системи.

Існує два основних механізми роботи сканерів уразливостей [13]:

– сканування - пасивний аналіз, який полягає в зборі інформації про порти, операційні системи, версії програмного забезпечення, отримані дані порівнюються з правилами визначення інформації про порти, програмне забезпечення та інші компоненти системи та робиться висновок про наявність чи відсутність вразливості;

– зондування – активний аналіз, який є імітацією тестованої вразливості.

Існують різні інструменти пошуку вразливостей для різних етапів перевірки систем на вразливості, від сканерів портів до складних систем, що складаються з сканерів портів і інструментів пошуку вразливостей експлойтів [13].

Приклади інструментів виявлення вразливостей:

– програма дослідження мережі Nmap – дозволяє визначати хости в мережі, встановлені служби та їх версії, операційні системи, які вони використовують, і версії фільтрів пакетів/брандмауерів [13];

– набір інструментів для перевірки веб-додатків Burp Suite – має власний проксі-сервер, сканер вразливостей, інструмент автоматизації атак;

– Metasploit Framework – інструмент для перевірки та використання вразливостей.

Інструменти для пошуку та використання вразливостей можуть бути зібрані в спеціальних операційних системах, призначених для аналізу безпеки та тестування на проникнення комп'ютерних систем.

Для тестування комп'ютерних систем на проникнення доступні такі операційні системи:

– ОС Kali Linux;
– ОС BlackArch Linux;
– ОС Parrot Security;
– ОС BlackBox.

Розглянемо приклади існуючих сканерів вразливостей (табл. 1.1).

Таблиця 1.1 - Існуючі сканери вразливостей та їх порівняння

Назва	Defect Tracking Integration	IAST Module Hybrid Analysis	SAST Module Hybrid Analysis	Flash Scanner	CGI Scanner	Enterprise Console Management Features	Demo
Rapid appspider	+	-	-	+	-	+/-	+
Portswigger Burp Suite	+/-	+	-	+	+	+/-	+
Fortify WebInspect	+	+	+	+	+	+	+
IBM Security AppScan	+	+	+	+	+	+	+
Accunetix Vulnerability Scanner	+	+	-	-	+	+	+
Netsparker Web Application Security Scanner	+	-	-	-	+	+	+
Janusec WebCruiser	-	-	-	-	+	-	+

Давайте коротко розглянемо кожен із запропонованих сканерів.

1) Продукт Rapid7 — це інструмент функціонального тестування веб- та мобільних додатків. Окрім виявлення вразливостей, Appspider може давати конкретні рекомендації щодо їх усунення та розставляти пріоритети.

Існує 95 типів атак, тож програми можна перевірити на стійкість до всіх поширених сьогодні загроз. Крім того, є можливість запускати окремі атаки окремо, щоб перевірити ефективність захисту від них.

Інструмент відмінно працює з усіма сучасними технологіями. Є підтримка Ajax, JSON, GWT та інших стандартів, різних форматів і протоколів, які використовуються в сучасних веб-додатках і браузерах. Також проводиться перевірка на відповідність сучасним стандартам безпеки.

Rapid7 може створювати інтерактивні звіти, представлені у вигляді веб-сторінок.

Їхня головна зручність полягає в можливості детально досліджувати конкретну вразливість і детально вивчати найдрібніші деталі, щоб вирішити проблему. Інструмент може інтегруватися з існуючими засобами безпеки (наприклад, Web Application Firewall), що значно економить час і ресурси. Ви можете спробувати інструмент безкоштовно протягом обмеженого часу.

Інтерфейс сканера показано на рисунку 1.3.

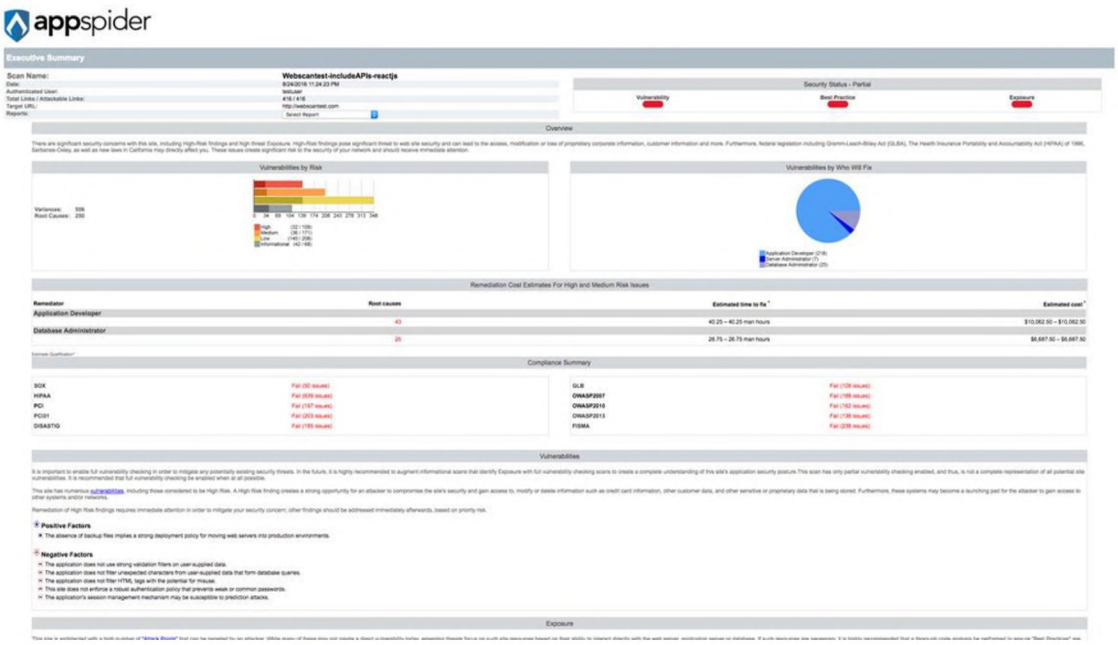

Рисунок 1.3 - Інтерфейс сканера Rapid appspider

Джерело: [12]

2) Portswigger Burp Suite. Цей сканер дозволяє перевіряти безпеку веб-додатків як вручну, так і автоматично. Але основний акцент робиться на ручні перевірки. Burp Suite має для цього ряд інструментів. Наприклад, Intruder дозволяє виявляти незвичайні вразливості, а Repeater використовується для маніпулювання та повторного надсилання окремих запитів. Інтерфейс Burp Suite інтуїтивно зрозумілий, ви можете швидко розібратися з програмою.

Однією з переваг інструменту є наявність безлічі сторонніх плагінів і надбудов, які серйозно розширюють його базовий функціонал. Багато з них були розроблені самими програмістами, які використовували Burp Suite у своїй роботі і точно знали, чого не вистачає оригінальному продукту. Якщо ви не знайшли потрібний вам плагін, ви можете написати його самостійно.

Інтерфейс програми показано на рисунку 1.4.

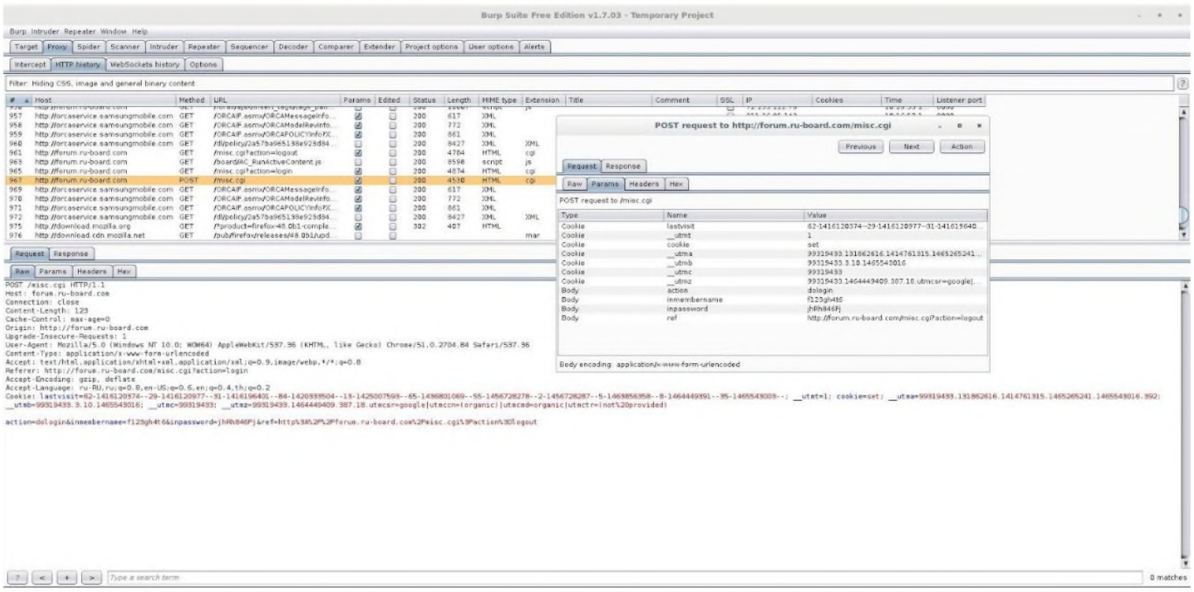

Рисунок 1.4 - Інтерфейс Portswigger Burp Suite

Джерело: [12]

3) Fortify WebInspect. Fortify WebInspect — один із найфункціональніших інструментів. Його можна постачати як ліцензоване програмне забезпечення або використовувати як модель SaaS (програмне забезпечення як послуга), або запустити як демонстраційну версію.

Інструмент може імітувати реальні атаки та методи злому, які найчастіше використовують кіберзлочинці. Сканер підтримує всі сучасні технології, що дозволяє працювати з додатками без урахування його архітектури. Серед найпопулярніших — Adobe Flash і JavaScript/Ajax, які сьогодні дуже часто використовуються при створенні веб-додатків.

Переваги цього продукту включають легкість встановлення, налаштування та масштабованість. Після завершення тестів Fortify WebInspect створює докладні звіти, які будуть корисні та зрозумілі як керівництву компанії, так і розробникам. Вони показують статистику знайдених уразливостей, їх пріоритетність (на що слід звернути увагу в першу чергу), а також детальну інформацію про кожну проблему. У програмі є набір готових шаблонів для звітів, але ви можете створити власні.

Інтерфейс показаний на рисунку 1.5.

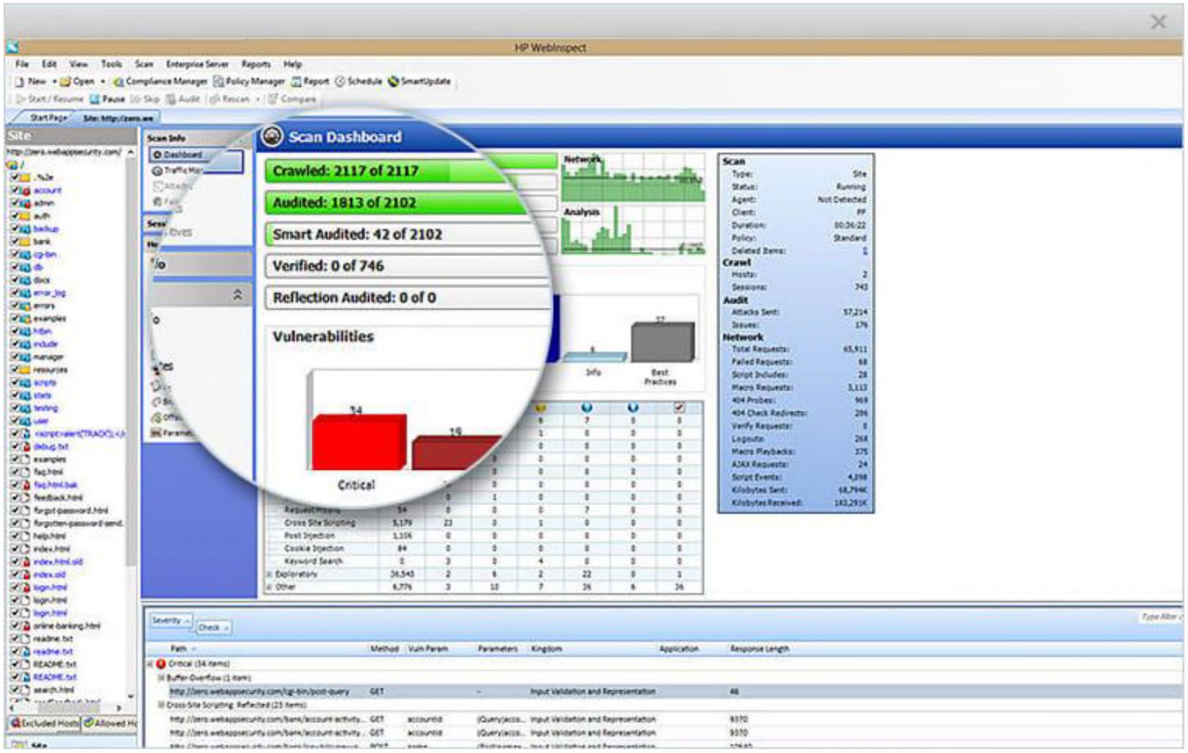

Рисунок 1.5 - Інтерфейс Fortify WebInspect

Джерело: [13]

4) IBM Security AppScan. Security AppScan має різні інструменти для проведення статичних і динамічних тестів, а також перевірки компонентів з відкритим кодом.

Програма підтримує більшість сучасних протоколів, стандартів і архітектур, включаючи JavaScript/Ajax і Adobe Flash. Незважаючи на те, що IBM Security AppScan зосереджена на автоматизованому скануванні, можна виконувати тестування програми вручну. Щоб забезпечити максимальну схожість із реальними атаками, алгоритми перевірки використовують адаптивні процедури, які імітують поведінку людини.

Виявлені проблеми представлені у вигляді зручних звітів. База даних програми містить більше 40 шаблонів звітів про відповідність різним стандартам: ISO 27001, ISO 27002, Basel II та ін.

Для кожної виявленої уразливості надається детальне пояснення та рекомендації щодо швидкого вирішення проблеми.

У цих порадах використовуються підготовлені кроки, які включають приклади коду та список завдань, які потрібно виконати спочатку. Продукт випускається в кількох випусках; ви можете спробувати безкоштовну версію перед покупкою.

Інтерфейс показаний на рисунку 1.6.

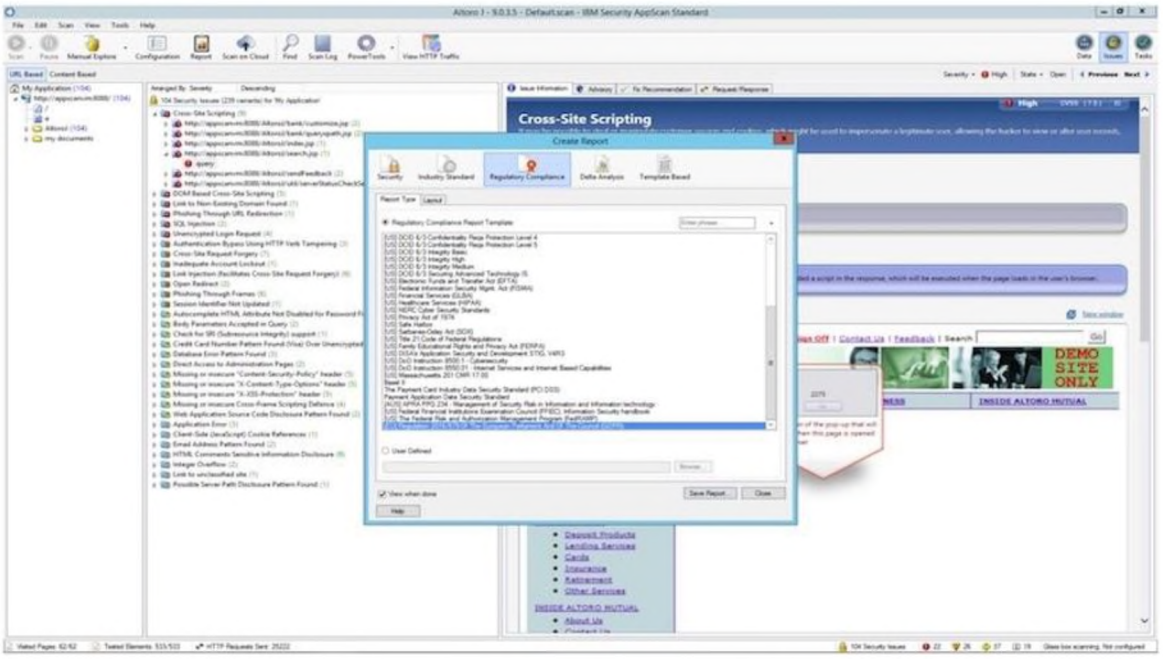

Рисунок 1.6 - Інтерфейс IBM Security AppScan

Джерело[14]

5) Сканер вразливостей Accunetix. Продукти безпеки Acunetix користуються широкою довірою таких клієнтів, як Visa та American Express. Ці продукти включають Acunetix Vulnerability Scanner, сканер безпеки веб-додатків. Цей інструмент має широкий набір функцій для забезпечення автоматичного контролю безпеки програми.

Він виявляє всі поширені типи вразливостей, включаючи впровадження SQL і виконання шкідливих скриптів і кодів. Наприклад, для популярної платформи WordPress цей продукт може виявити до 1200 відомих уразливостей. При цьому він може працювати в багатопоточному режимі, що дозволяє безперервно сканувати тисячі об'єктів з різних платформ.

Усі виявлені проблеми відображаються у зручних звітах. Вони підходять як для фахівців, які будуть безпосередньо вирішувати проблеми, так і для керівників, яким необхідно бути в курсі того, що відбувається, і розуміти загальну картину. Отримані звіти можна зібрати в загальний файл. Ці результати можна порівняти з аналогічними даними минулих сканувань, щоб визначити, які вразливості усунено, а які залишилися.

Інтерфейс програми показано на рисунку 1.7.

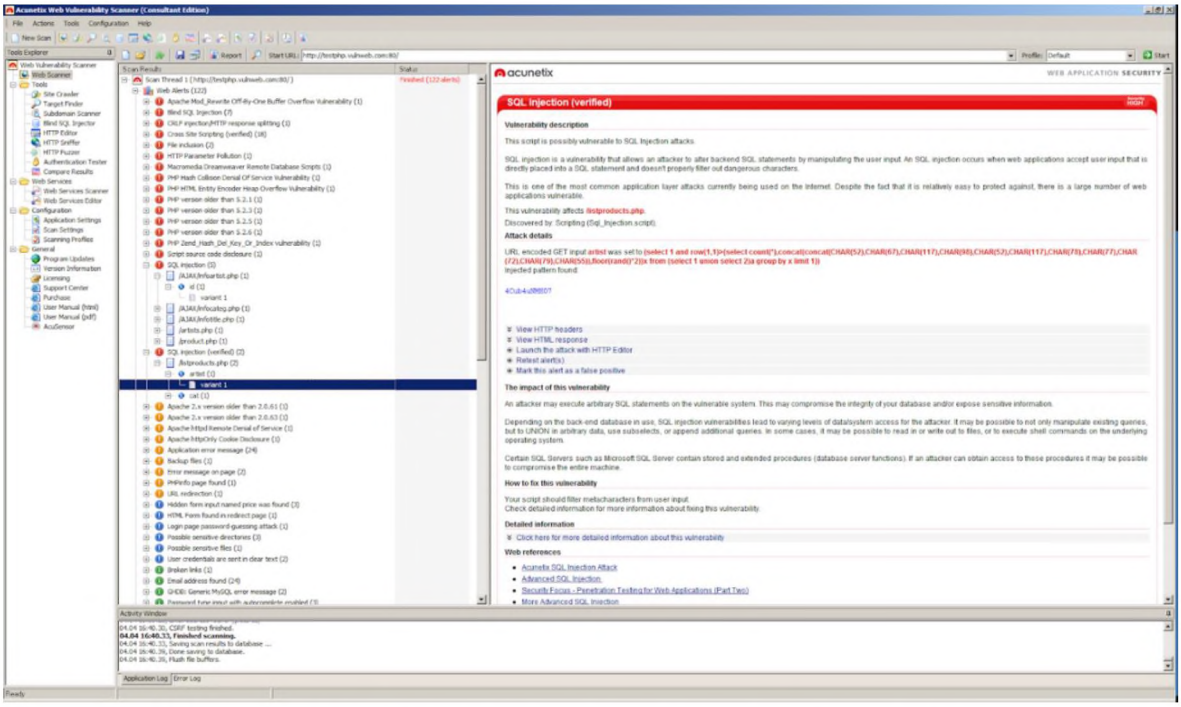

Рисунок 1.7 - Сканер уразливостей Accunetix

Джерело: [15]

6) Сканер безпеки веб-додатків Netsparker. Цей повністю автоматичний сканер має високий рівень виявлення вразливостей і мінімальну кількість помилкових спрацьовувань. Такий результат досягається завдяки фірмовому інструменту Proof-Based Scanning, який не тільки сигналізує про загрозу, але й одразу надає докази того, що це не помилковий знос.

Це економить багато часу та ресурсів, оскільки виявлені загрози не потрібно повторно сканувати вручну, і ви можете негайно приступити до їх усунення.

Сканер безпеки веб-додатків Netsparker може аналізувати веб-додатки та служби на всіх поширених платформах, включаючи Java Script, HTML 5, .NET та багато інших. Перевірка виконується у всіх поширених типах атак.

Інструмент може одночасно працювати з сотнями і тисячами ресурсів, при цьому легко інтегруючись в існуючі системи безпеки. Продукт доступний у настільній, корпоративній та хмарній версіях. Також є пробна версія.

Інтерфейс показаний на рисунку 1.8.

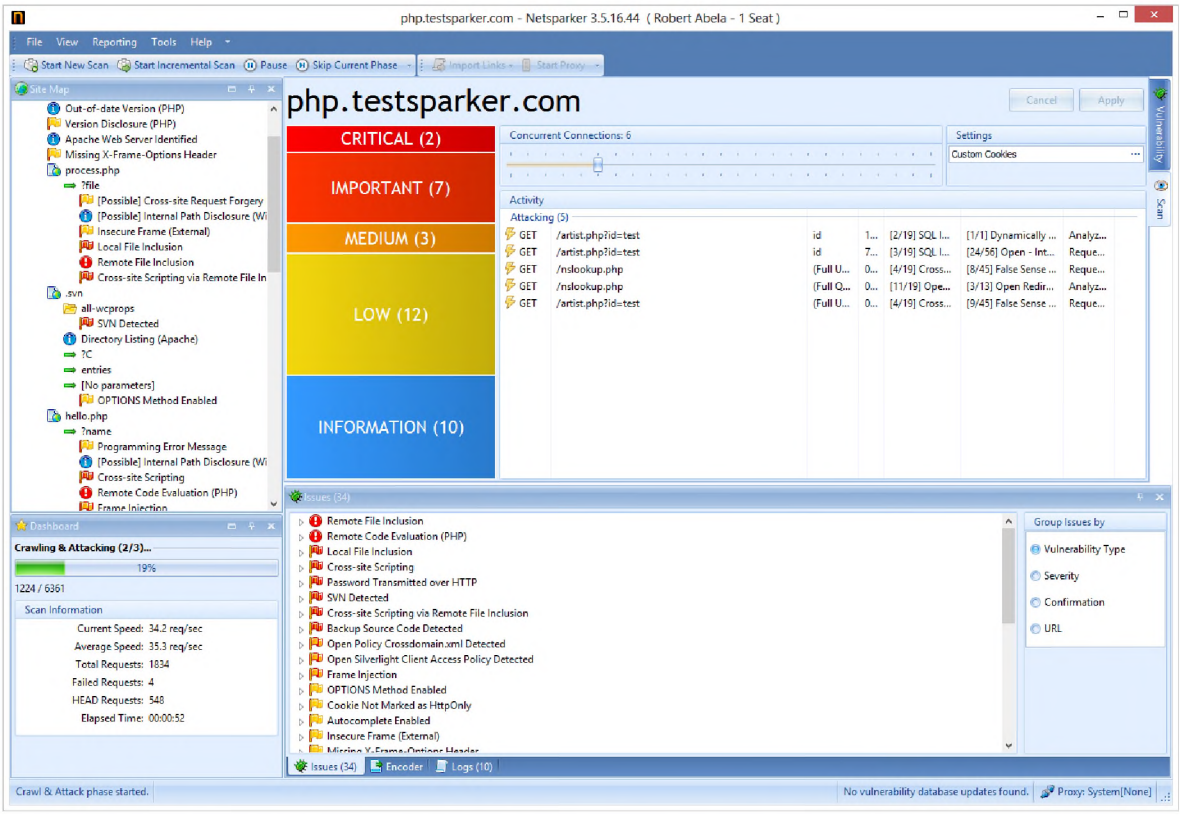

Рисунок 1.8 - Інтерфейс сканера безпеки веб-додатків Netsparker

Джерело: [16]

7) Janusec WebCruiser. Сканер призначений в основному для проведення SQL-ін'єкцій на різних платформах, таких як SQL Server, Oracle і Access. Він також має інструменти для боротьби з файлами cookie, міжсайтовим шифруванням, впровадженням PHP та іншими поширеними загрозами.

Інтерфейс показаний на рисунку 1.9.

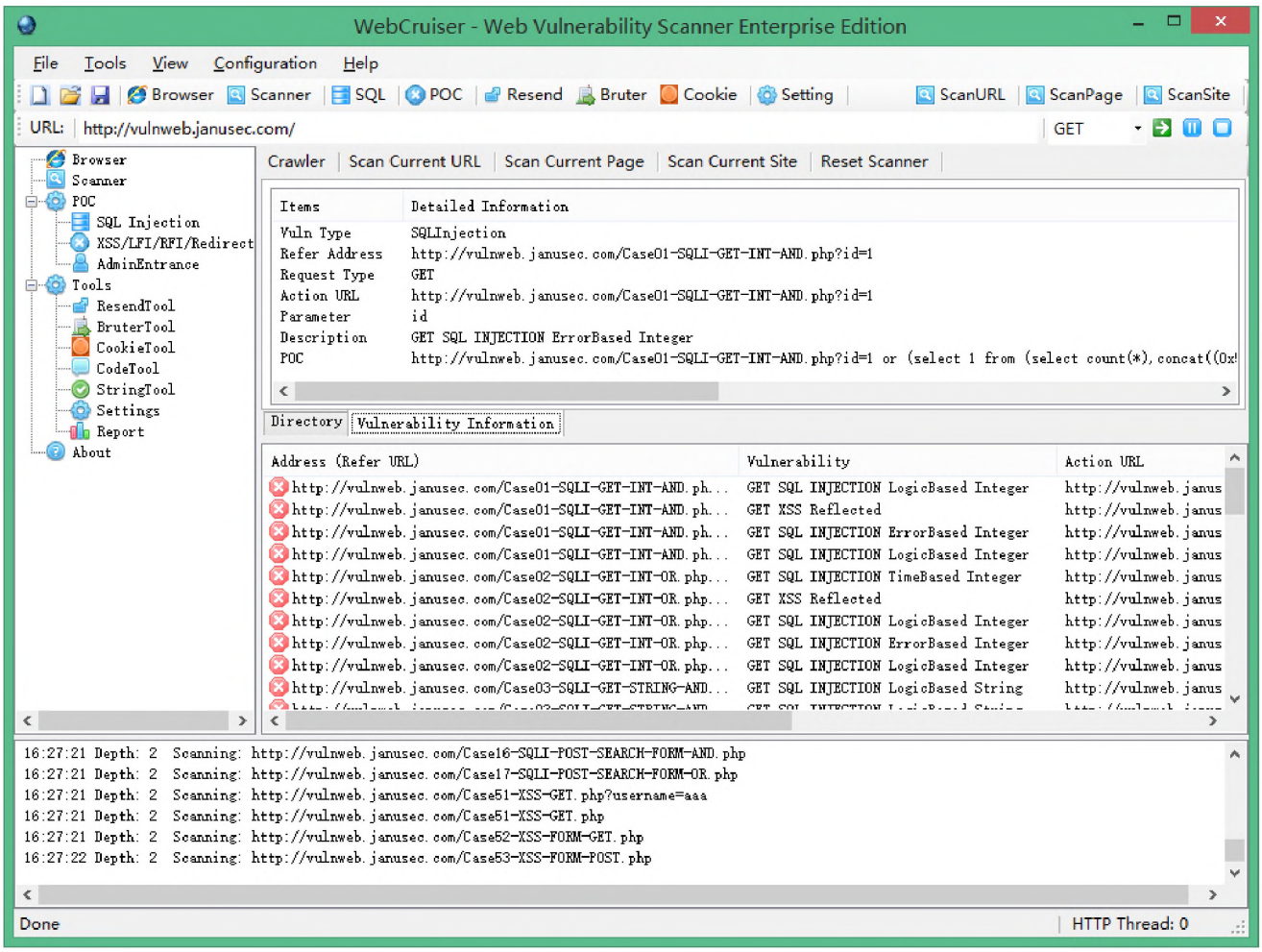

Рисунок 1.9 - Інтерфейс Janusec WebCruiser

Джерело: [17]

8) Несус. Це відомий і популярний сканер уразливостей, який надається безкоштовно для особистого некомерційного використання.

Вперше він був випущений у 1998 році Renurd Derason і зараз опублікований Tenable Network Security. Існує також допоміжний проект Nessus 2 під назвою OpenVAS, опублікований під GPL.

Використовуючи велику кількість засобів перевірки вразливостей, які називаються плагінами Nessus, можна виявити велику кількість відомих уразливостей. Metasploit приймає файли результатів сканування вразливостей від Nessus і OpenVAS у форматі файлу nbe.

Інтерфейс сканера показано на рисунку 1.10.

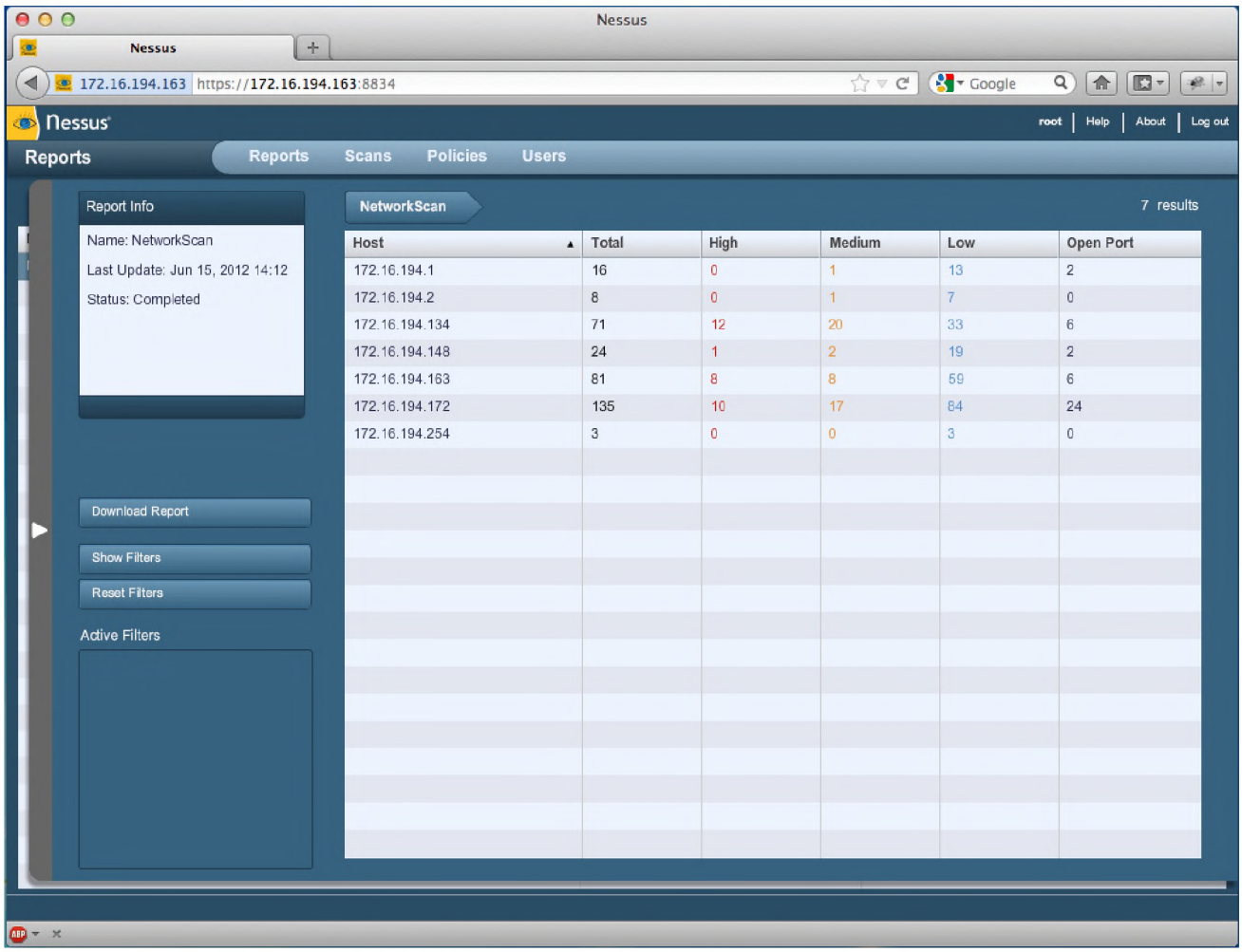

Рисунок 1.10 - Інтерфейс сканера Nessus

Джерело: [18]

Nessus може сканувати такі вразливості:

– уразливості, які можуть дозволити несанкціонований контроль або доступ до конфіденційних даних у системі;

– північна конфігурація (наприклад, open mail relay):

– уразливості відмови в обслуговуванні (DOS);

– стандартні паролі, кілька загальних паролів і порожні/відсутні паролі для деяких системних облікових записів.

Збої програмного забезпечення, відсутні виправлення, зловмисне програмне забезпечення та помилки неправильної конфігурації в широкому діапазоні операційних систем, пристроїв і програм усуваються за допомогою Nessus.

Наразі сервер Nessus доступний для:

– Unix;

– Linux;

– FreeBSD.

Сканер також доступний для:

– операційні системи на базі Unix;

– операційні системи на базі Windows;

Основні характеристики Nessus включають:

– планові аудити безпеки;

– виявлення дірок безпеки на локальних або віддалених хостах;

– імітація атак з виявлення вразливостей;

– виявлення відсутніх оновлень безпеки та виправлень;

– Nessus Professional сканує внутрішню мережу відповідно до вимог PCI DSS 11.2.1.

Приклад результатів сканування представлено на рисунку 1.11.

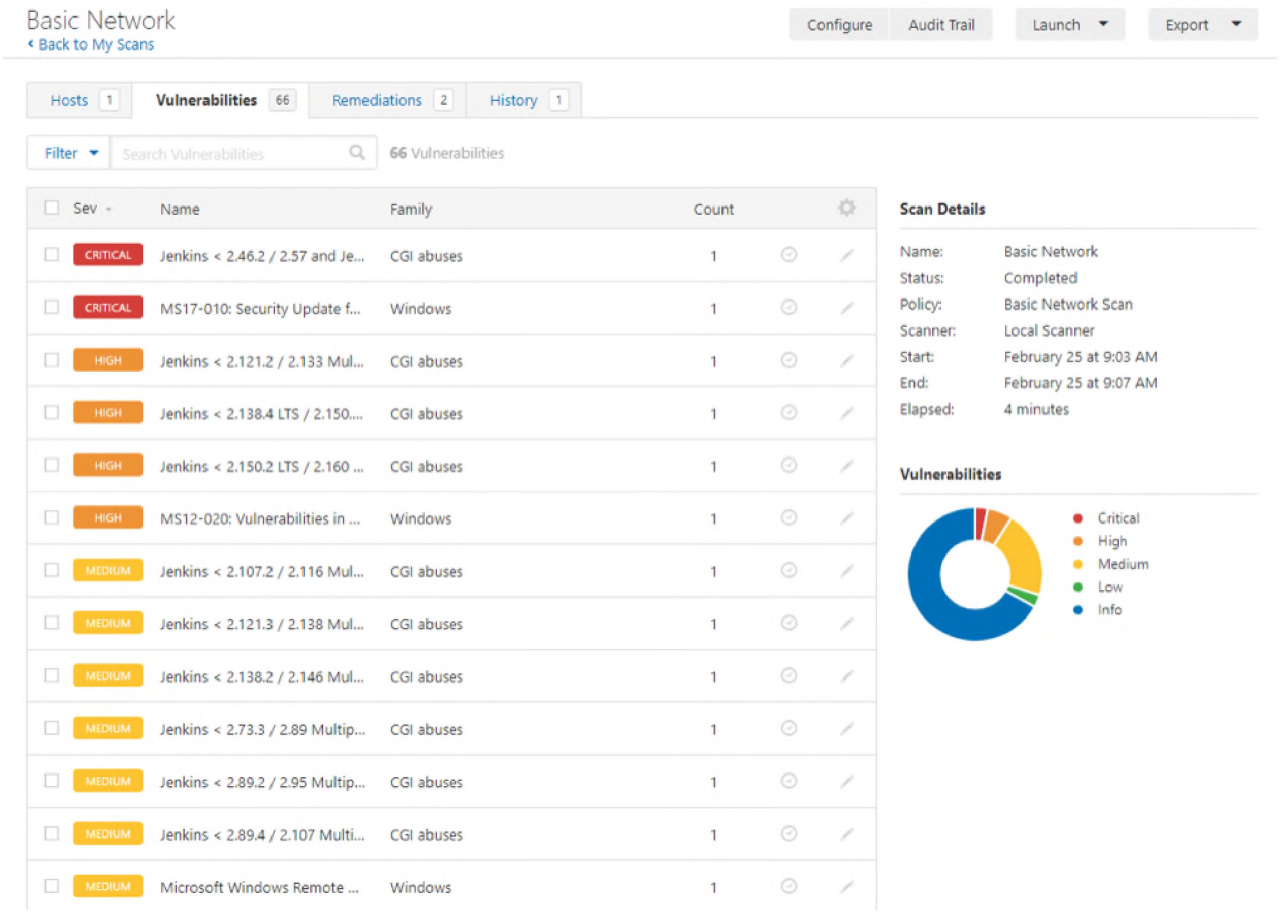

Рисунок 1.11 - Приклад результатів сканування Nessus

Джерело: [18]

Таким чином, існують наступні варіанти пошуку вразливостей - за допомогою комплексних комерційних рішень, таких як Nessus, або за допомогою інструментів, що входять до складу спеціальних операційних систем, таких як Kali Linux і т.д.

Використання спеціального програмного забезпечення для пошуку вразливостей і проведення тестування на проникнення вимагає дотримання певного алгоритму використання вбудованих в програмне забезпечення засобів для підвищення ефективності пошуку вразливостей.

1.5 Автоматизовані системи тестування як ефективний засіб захисту персональних даних

Ручне тестування захисту персональних даних вимагає великих людських ресурсів для проведення повного циклу тестування безпеки на наступних етапах:

- збір інформації;
- ручне сканування вразливостей захисту персональних даних;
- виявлення та тестування вразливостей;
- підготовка звітів;
- корекція [13].

У будь-якій організації ці етапи зазвичай охоплюють багато людей: розробників, аналітиків якості, менеджерів проектів, мережевих адміністраторів, розробників веб-додатків, менеджерів з інформаційної безпеки, аудиторів і менеджерів. Навіть сторонні постачальники беруть участь у проектах з оцінки веб-безпеки. При такій кількості співробітників, які працюють на одну мету, необхідно максимально автоматизувати пошук уразливостей в особистих даних студентів навчального закладу.

Автоматизоване тестування конфіденційності має певні переваги. По-перше, існує ризик дублювання зусиль під час надлишкових тестів. Коли існує

багато складних веб-додатків, як сьогодні у більшості онлайн-компаній, це може призвести до значного обсягу непотрібної роботи [13].

Інша проблема полягає в тому, що немає ні знань, ні часу постійно вручну перевіряти вразливість усіх особистих даних. Якщо перевірка безпеки веб-додатків не автоматизована за допомогою перевіреної автоматизованої системи конфіденційності, яка може перевіряти тисячі потенційних недоліків безпеки, деякі, якщо не всі, серйозні вразливості веб-додатків можуть бути пропущені.

Як приклад розглянемо спеціальну веб-систему планування ресурсів підприємства (ERP). Така система матиме сотні видимих точок входу або поверхонь для атаки, а також багато інших прихованих уразливостей, які потрібно буде перевірити на наявність уразливостей персональних даних, таких як впровадження SQL і міжсайтовий сценарій.

Скажімо, ERP має 200 точок входу для перевірки 100 різних уразливостей веб-додатків. Це означає, що тестувальник проникнення повинен виконати щонайменше 20 000 тестів безпеки. Якби кожен тест займав лише 5 хвилин, фахівцеві з веб-безпеки знадобилося б приблизно 208 робочих днів, щоб провести належний аудит безпеки персональних даних у системі ERP.

Автоматизована система захисту ідентифікаційної інформації може сканувати набагато більші системи ERP на наявність багатьох інших типів уразливостей ідентифікаційної інформації за лічені години. І, на відміну від людини, автоматизований сканер безпеки запам'ятає сканувати вхідний параметр.

Під час виконання ручного тесту безпеки веб-програми проникнення обмежене кількома відомими вразливими місцями, відомими тестеру проникнення. З іншого боку, використовуючи автоматичний сканер веб-вразливостей, ви можете переконатися, що всі налаштування перевіряються на всі типи налаштувань безпеки особистих даних.

Використовуючи автоматизовані системи безпеки, ви можете гарантувати відсутність помилкових спрацьовувань у результатах перевірки безпеки персональних даних, тому вам не потрібно витрачати час на перевірку виявлених вразливостей.

Важливість автоматизації тестування вразливостей підкреслюється популярними дослідженнями в галузі інформаційної безпеки. Рік за роком це дослідження вказує на одні й ті самі основні причини інформаційних ризиків, такі як брак ресурсів, відсутність прозорості та недостатня обізнаність керівництва. Кожен із цих елементів можна вирішити за допомогою автоматизованого тестування безпеки [13].

Як зазначено в [13], не існує ідеального способу перевірити вразливість персональних даних. Однак виконання цього вручну і покладення лише на досвід співробітників не може гарантувати повну достовірність отриманих результатів тестування.

Існують різні інструменти для автоматичного тестування безпеки персональних даних студентів навчального закладу; розглянемо список найкращих інструментів для систем тестування персональних даних на 2022 рік згідно з інформацією, представленою в [12,13].

1. WebLOAD. WebLOAD — це ефективний інструмент тестування продуктивності веб-додатків, який демонструє здатність програми справлятися з навантаженням. Цей інструмент має потужні можливості створення сценаріїв, які спрощують складне тестування. Крім того, його можна використовувати для виявлення аспектів продуктивності, вузьких місць і можливих вразливостей у програмі. Це відповідний інструмент, який може допомогти досягти необхідної ефективності реагування веб-програми. Крім того, WebLOAD також інтегровано з Selenium, Jenkins та іншими інструментами для покращення можливостей тестування.

Інтерфейс показаний на рисунку 1.12.

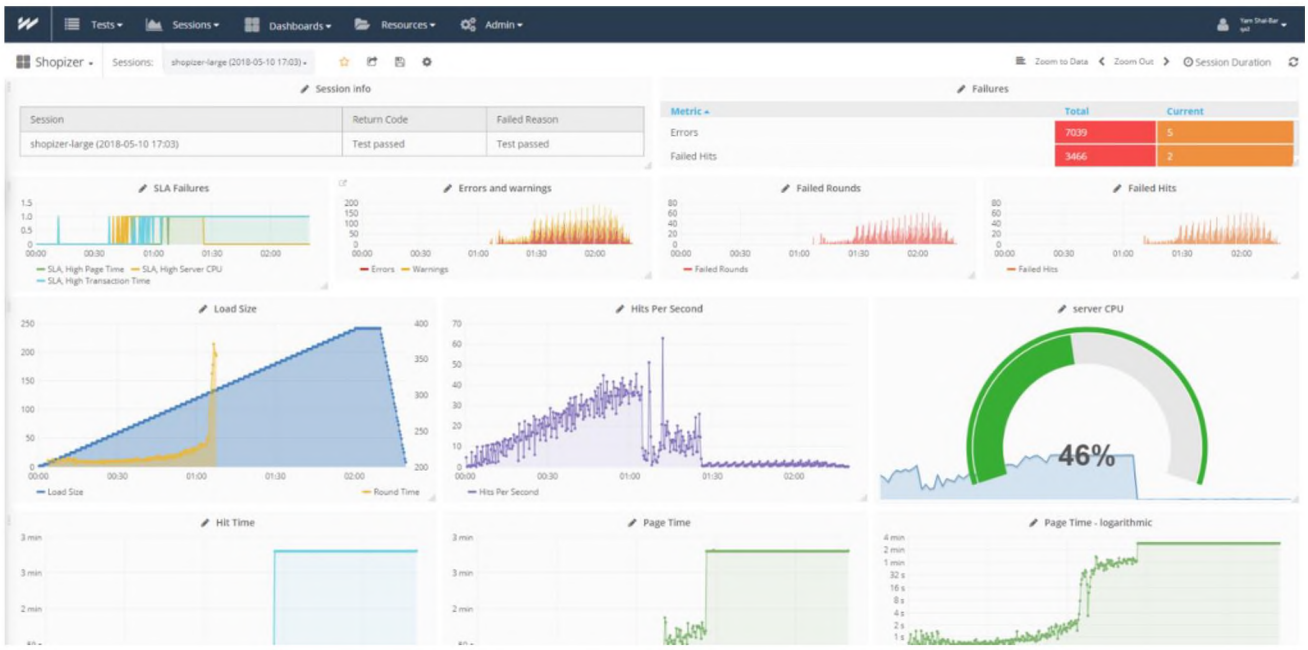

Рисунок 1.12 - Інтерфейс інструменту WebLOAD

Джерело: [12]

2. Акунетикс. Acunetix — це автоматизована версія інструментів тестування безпеки веб-додатків із вбудованим сканером безпеки, перевіреним раніше для виявлення навіть незначних уразливостей у понад 4500 веб-додатках.

Інтерфейс показаний на рисунку 1.13.

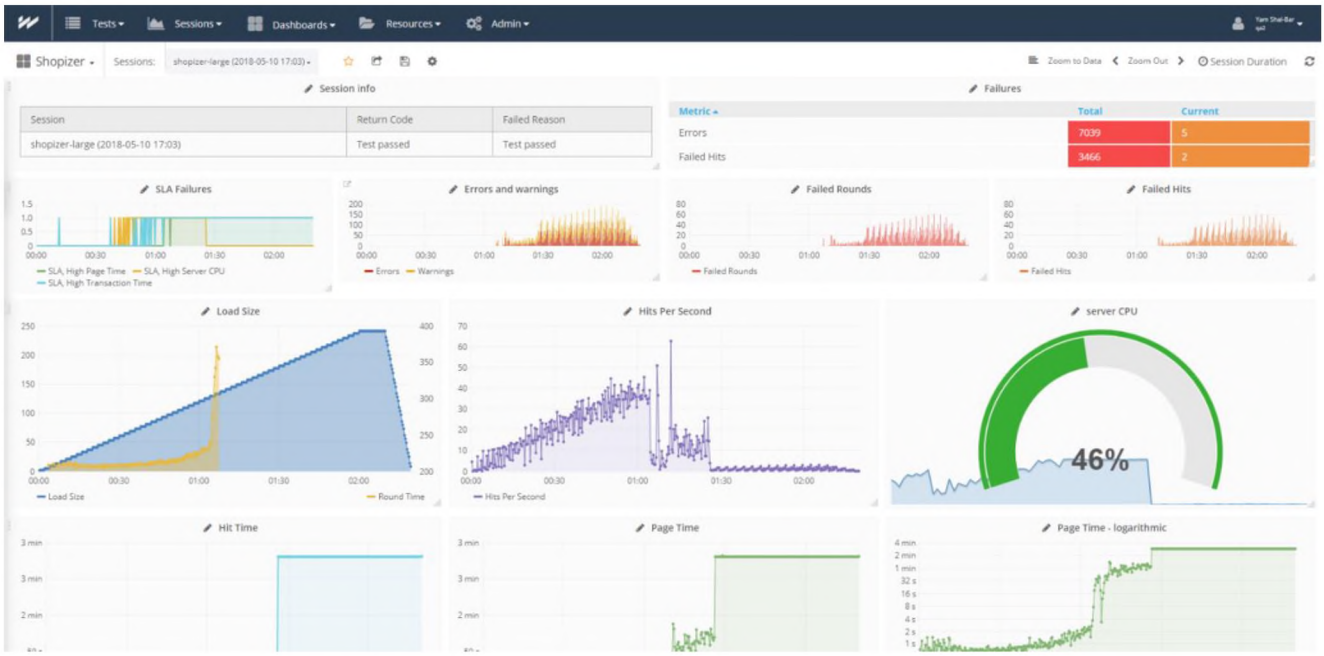

Рисунок 1.13: Інтерфейс автоматизованої системи Acunetix

Джерело: [13]

3. Netsparker. Інструмент тестування веб-додатків Netsparker відомий своєю точністю. Він також має вбудований сканер безпеки для виявлення незначних і серйозних уразливостей майже в усіх веб-інтерфейсах API, про що йшлося раніше. Цей інструмент виявляє помилки та перевіряє, чи є вони справжніми. Це заощадить час порівняно з ручною перевіркою та переглядом кожної виявленої несправності після сканування. Інструмент Netsparker можна використовувати як програмне забезпечення Windows або як постачальник онлайн-послуг. Характеристики та ефективність залишаються однаковими в обох версіях.

Інтерфейс показаний на рисунку 1.14.

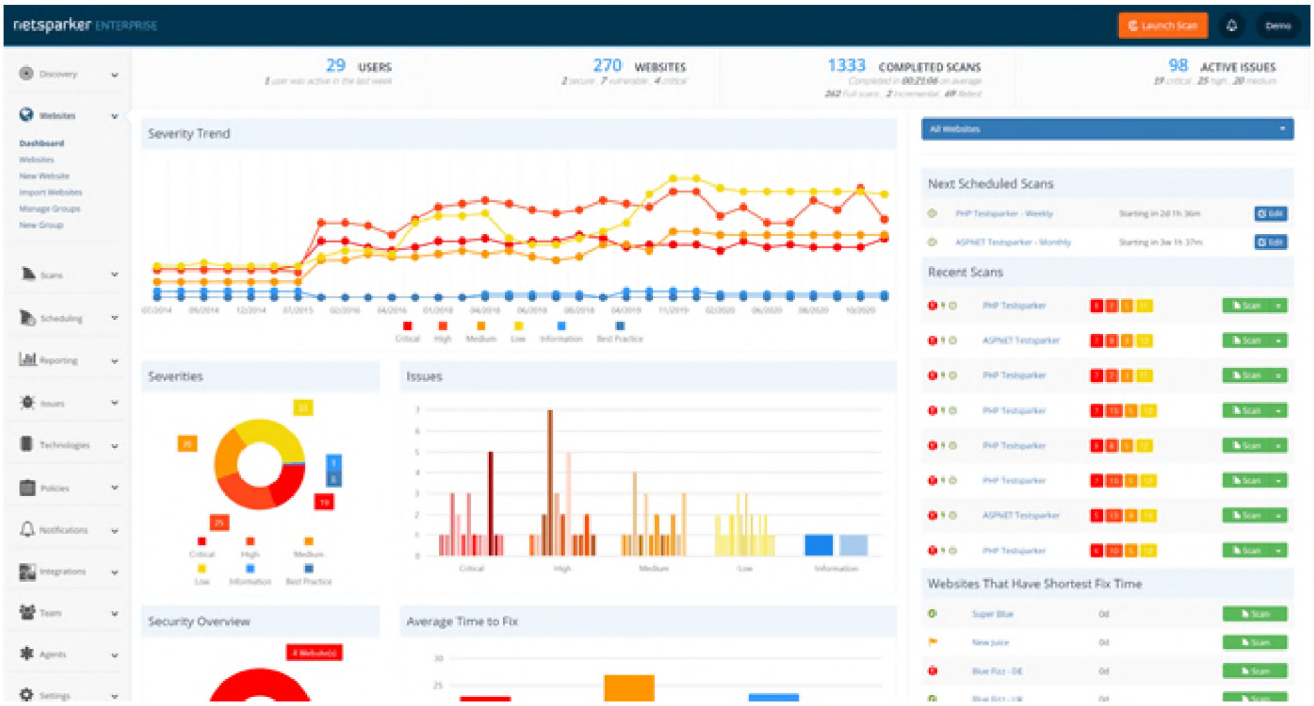

Рисунок 1.14 - Інтерфейс автоматизованої системи Netsparker

Джерело: [14]

4. Експеритест. Experitest дозволяє користувачам тестувати веб-програми на понад 1000 пристроях одночасно в хмарі. Існують як інструкції, так і автоматизовані інструменти перегляду веб-додатків, доступних для тестування веб-додатків.

З Expritest буде достатньо протестувати додаток у будь-якому браузері. Крім того, він інтегрований з Appium і Selenium для досягнення найкращих

результатів. Шаблон тестування цього інструменту працює в реальному часі та налагоджує його, щоб отримати найкращі результати. Expertest може проводити більше 100 тестів одночасно. Крім того, можна провести візуальне тестування, щоб визначити швидкість реагування інтерфейсу користувача при різних роздільних здатностях.

Інтерфейс показаний на рисунку 1.15.

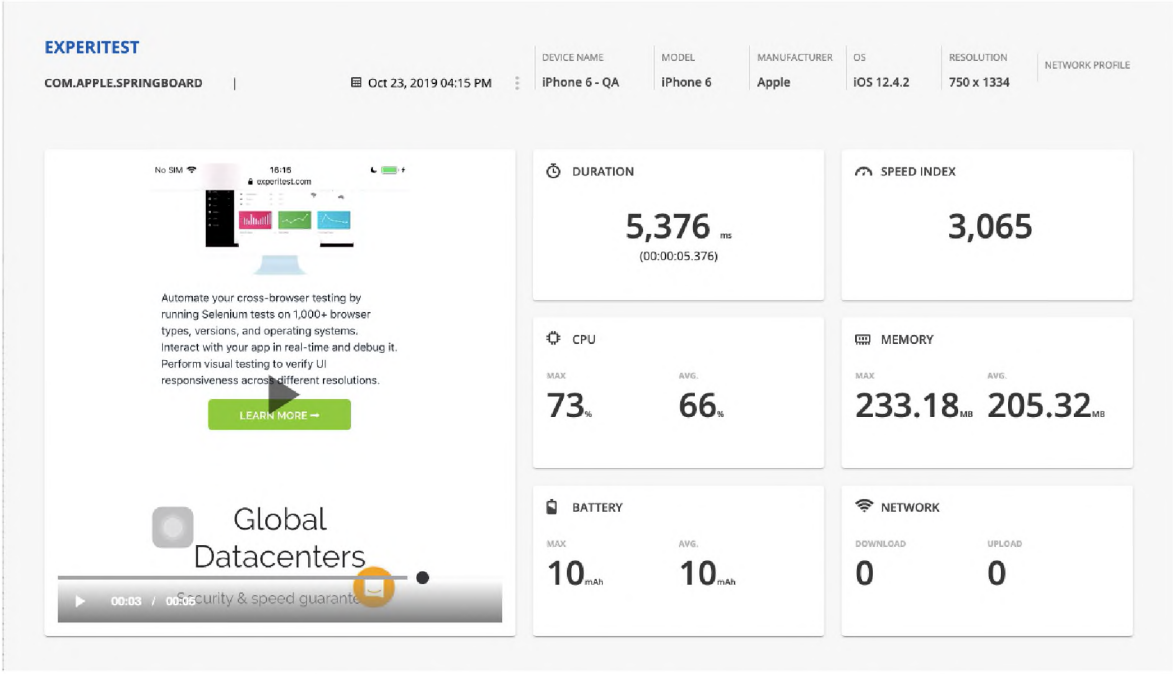

Рисунок 1.15 - Інтерфейс автоматизованої системи Expertest

Джерело: [15]

5. Лямбда-тест. Інструмент Lambda Test розроблено для забезпечення безперебійної роботи всіх елементів веб-додатків, таких як CSS, HTML5 та інші, на всіх пристроях. Цей інструмент використовує схему тестування: ручне, візуальне, автоматизоване тестування або прогресивні результати. Він використовує хмарну інфраструктуру для одночасного запуску кількох тестів.

Інтерфейс показаний на рисунку 1.16.

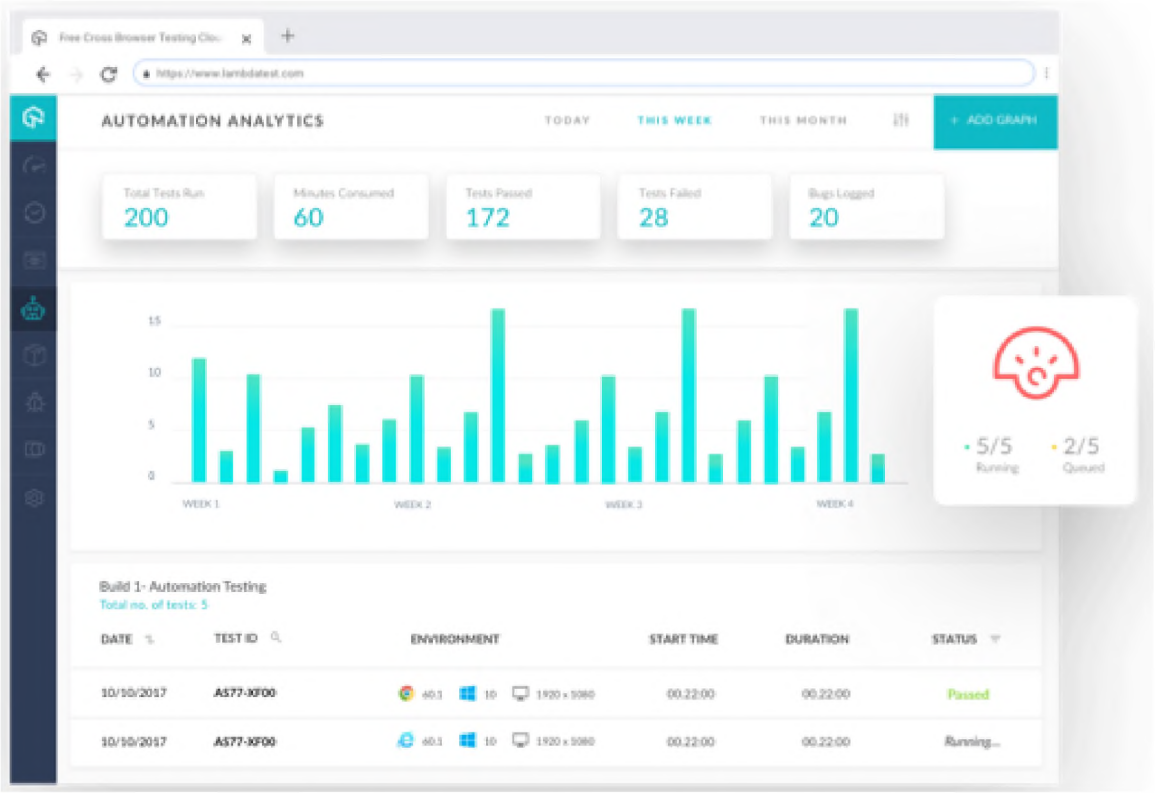

Рисунок 1.16 - Інтерфейс автоматизованої системи Lambda Test

Джерело: [16]

6. Селен. У списку інструментів автоматизації тестування з відкритим кодом для веб-додатків це один із найбільш широко використовуваних інструментів тестувальниками. Selenium широко відомий як один із найкращих інструментів регресійного тестування, який бездоганно працює в різних браузерах і платформах.

Автоматизація тестування є однією з найкращих переваг Selenium, і в нього вбудовано багато окремих інструментів. Кожен інструмент Selenium призначений для обробки різних аспектів тестування веб-додатків. Selenium IDE, Selenium Web Driver, Selenium RC і Selenium Grid — це чотири компоненти Selenium.

Інтерфейс системи показано на рисунку 1.17.

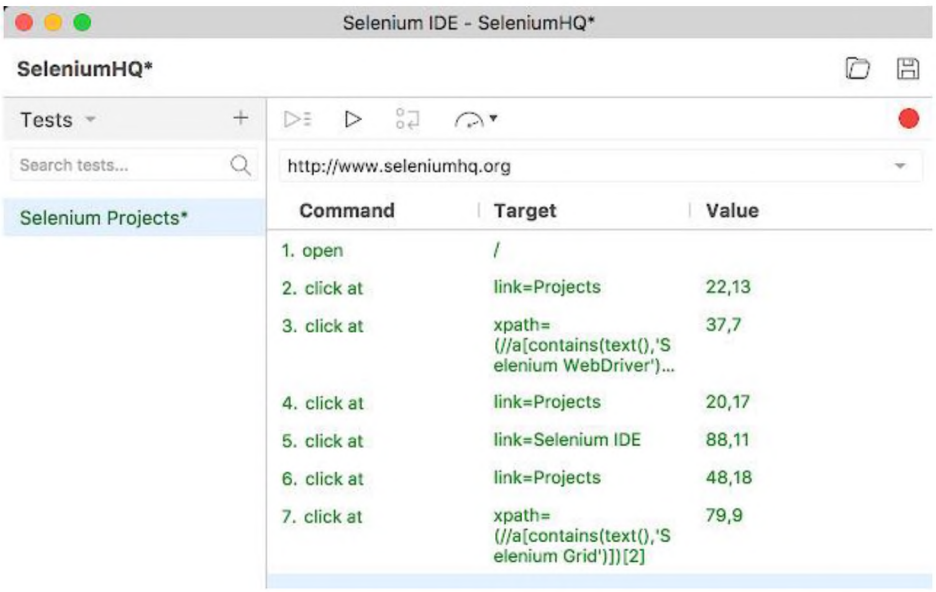

Рисунок 1.17 - Інтерфейс автоматизованої системи Selenium

Джерело: [17]

7. Ватир. Watir — це акронім для тестування веб-додатків у Ruby. Найкраще у Watir те, що він тестує веб-програми так само, як це роблять люди. Цей інструмент тестування використовує такі ідеології, як натискання посилань, перевірка тексту, заповнення форм та інші аспекти для тестування веб-додатків у режимі реального часу для виявлення основних лазівок. Система досить проста у використанні і без проблем працює на різних платформах і браузерах.

Інтерфейс показаний на рисунку 1.18.

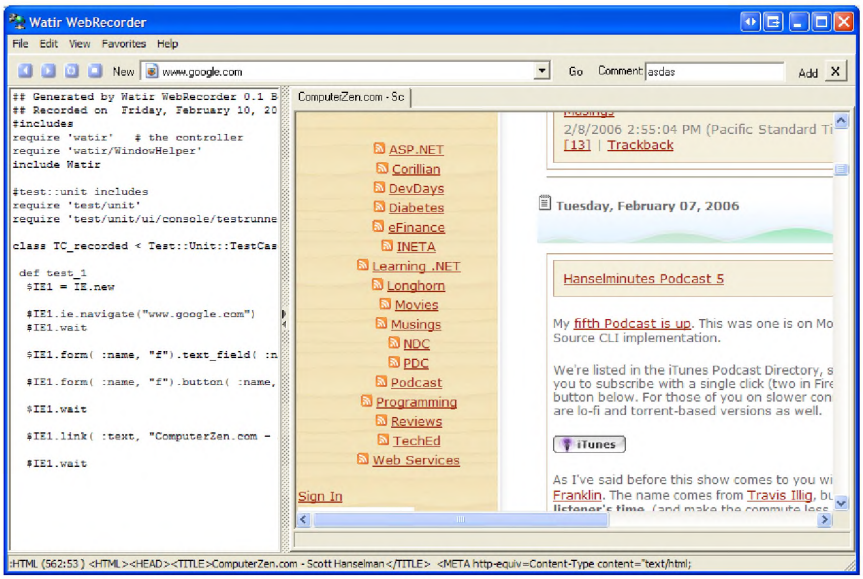

Рисунок 1.18 - Інтерфейс автоматизованої системи Watir

Джерело: [18]

8. Студія Ранорекс. Ranorex розроблено для максимального використання ресурсів для підтримки автоматизованих методів тестування. Це придатний інструмент для початку наскрізного тестування за допомогою симуляторів або реальних пристроїв для більш помітних результатів. Він ідеально працює майже з усіма браузерами, такими як Safari, Microsoft Edge, Chrome та іншими.

Інтерфейс показаний на рисунку 1.19.

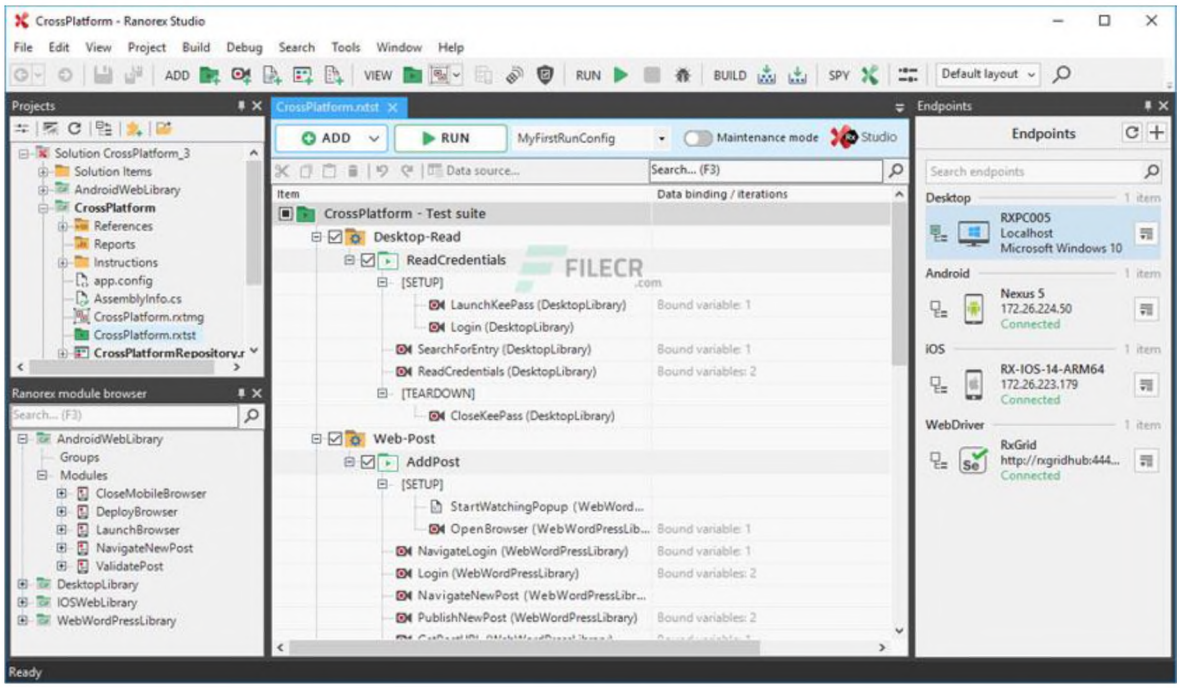

Рисунок 1.19 - Інтерфейс автоматизованої системи Ranorex Studio

Джерело: [19]

9. Безтурботність. Інструмент тестування Serenity гарантує, що написані тести легше підтримувати та достатньо гнучкі для різних програм. Крім того, Serenity створює докладні звіти про тестування, щоб надати розуміння продуктивності веб-додатку. Крім того, він також відстежує весь хід проекту.

Вище наведено 9 найкращих інструментів тестування персональних даних, які широко використовуються компаніями. З відповідними результатами робота над модифікацією програм стане простішою та зручнішою.

2 СПЕЦІАЛЬНА ЧАСТИНА

2.1 Вибір засобів розробки

В якості мови програмування для проектування автоматизованої системи було обрано C++, оскільки це сучасна, гнучка, потужна мова програмування.

Також використовували Windows Forms. Підхід до розробки програм на Windows Forms базується на GDI (Graphics Device Interface, Graphical Device Interface) – інтерфейсі Windows для представлення графічних об'єктів і передачі їх на пристрої відображення, такі як монітори та принтери.

GDI відповідає за малювання ліній і кривих, відображення шрифтів і роботу з палітрами. Він не відповідає за відображення вікон, меню тощо; це завдання покладено на підсистему користувача, розташовану в user32.dll і засновану на GDI.

Однією з переваг Windows Forms є те, що вона дозволяє писати кросплатформні програми. Проекти, написані в Windows Forms, можна легко перенести в іншу операційну систему, якщо на ній встановлено .Net Framework потрібної моделі, де написаний проект.

Для інтелектуалізації автоматизованої системи також використовувалася нейронна мережа. Нейронна мережа є методом прогнозної аналітики. Система намагається передбачити вартість так, як це зробив би людський мозок.

Нейронна мережа працює шляхом створення мережі вхідних вузлів (які є початком мережі), вихідних вузлів (які показують результати/прогнози, коли дані проходять через мережу) і прихованого шару між цими вузлами.

Прихований шар між вхідними та вихідними вузлами робить нейронну мережу такою унікальною та ефективною. Щоразу, коли дані «завантажуються» в нейронну мережу, алгоритм включає дані, що проходять через неї, призначаючи «ваги» вузлам у прихованому шарі, що може змінити результат на початкових вузлах.

Висока точність нейронних мереж викликає питання, чому вони не використовуються частіше, ніж зараз. Як і очікувалося, нейронні мережі також

мають недоліки. Нейронні мережі потребують більшої обчислювальної потужності, ніж звичайні інструменти прогнозування, що робить їх дорожчими. Однак для точності прогнозування вразливості актуальним є використання нейронних мереж.

Метою автоматизованої системи тестування безпеки веб-додатків є сканування сайту на наявність вразливостей, визначення ймовірності вразливості та виявлення вразливостей.

Розроблена автоматизована система тестування структурно складається з 2 кнопок, 5 міток, 2 текстових полів і 1 веб-браузера.

Призначення першої кнопки – розпочати аналіз тексту, призначення другої кнопки – завантажити веб-сторінку.

Призначення першого текстового поля – запис адресного рядка. Друге текстове поле призначене для зберігання звітів

Весь процес сканування веб-додатків відбувається в MyForm.h. До нього підключається заголовний файл і клас myNeuro.h і myNeuro.cpp, де зберігається реалізація алгоритму нейронної мережі для аналізу тексту.

Аналіз тексту виконується методами нейронної мережі. Алгоритм без викладача. У неконтрольованому навчанні модель має набір даних і немає чітких інструкцій щодо того, що робити з цим набором. Нейронна мережа намагається самостійно знаходити кореляції в даних, витягуючи корисні ознаки та аналізуючи їх.

Текст сторінки записується в рядок, а потім за допомогою виклику методу класу цей рядок аналізується, після аналізу вказується можливий відсоток шкідливого контенту.

2.1.1 Розробка структури даних

Структура даних про вразливості — це база даних, призначена для збору, зберігання та розповсюдження інформації про виявлені вразливості, націлені на комп’ютерні системи. База даних зазвичай ідентифікує виражену вразливість і

може оцінити потенційну шкоду для комп'ютерних систем і обхідний шлях, необхідний для запобігання атаці [26].

Існує багато баз даних уразливостей. Після оприлюднення уразливості CERT (Computer Emergency Response Team) у 1989 році всі ці уразливості з'явилися в багатьох нових базах даних, опублікувавши їх у іншому форматі інформації. Існує багато типів уразливостей у кількох базах даних, але вони представлені по-різному.

Наприклад, база даних MS містить інформацію про вразливості в продуктах Microsoft, BugTraq, яка запитує вирішення уразливості, як її вирішити, NVD має рівень оцінки вразливості, OSVDB — це велика база даних, яка містить більше звітів про вразливості [33]. .

Список найвідоміших баз даних [33]:

– SVDB: Метою бази даних є надання точної та об'єктивної інформації про вразливі місця безпеки комп'ютерного обладнання. У блозі OSVDB обговорюються різні теми, пов'язані з уразливістю, зокрема розкриття інформації, запуск бази даних уразливостей (VDB) тощо. Але ця база даних повідомляє недостатньо статистичної інформації;

– Security Focus Bugtraq: містить детальну інформацію про вразливість, а також інформацію про експлойти (за наявності). Але він містить заплутану систему пошуку вразливостей;

– CVE: база даних відомих вразливостей інформаційної безпеки.

– Кожна вразливість містить ідентифікаційний номер, опис і кількість загальнодоступних описових посилань. CVE підтримується MITRE;

– NVD: підтримка уряду США та співпраця з базою даних CVE;

– VULBD: велика база даних уразливостей, з додатковим функціоналом у вигляді статистики та огляду існуючих експлойтів;

– NIPC: уразливість китайських баз даних. Без перекладу на інші мови;

– Бюлетень безпеки Microsoft: містить інформацію лише про вразливості продуктів Microsoft;

– База даних експлойтів: база даних містить інформацію про вразливості, які вже використовувалися для атак. Я не використовую цю базу даних у своїй роботі, оскільки вона містить лише інформацію про вразливості, для яких існують експлойти;

– MFSA: опубліковано інформацію про вразливості в продуктах Mozilla. У своєму дослідженні я не розглядаю цю базу даних, оскільки її вразливість впливає лише на продукти Mozilla і не охоплює всі служби.

2.1.2 Написання програмного продукту

Для початку роботи програми були підключені такі бібліотеки

```
#include <ctime>
#include <math.h>
#include "stdlib.h"
#include "myNeuro.h"
#include <chrono>
#include <ctime>
#include <msclr\marshal_cppstd.h>
```

Наступним кроком є ініціалізація форми

```
/// <summary>
     /// Сводка для MyForm
     /// </summary>
     public ref class MyForm : public
System::Windows::Forms::Form
     {
     public:
          MyForm(void)
          {
               InitializeComponent();
               //
               //TODO: добавьте код конструктора
               //
```

```
        }

    protected:
        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        ~MyForm()
        {
            if (components)
            {
                delete components;
            }
        }
```

Для обробки події на натискання кнопки та виведення інформації про зміст вказаного веб-сайту була написана наступна частина коду:

```
    string ans = "Initializing QUICK SCAN..." +
msclr::interop::marshal_as<std::string>(Environment::NewLine);

            string str = to_string(now->tm_year + 1900) + '-' +
to_string(now->tm_mon + 1) + '-' + to_string(now->tm_mday) + ' ' +
to_string(now->tm_hour) + ':' + to_string(now->tm_min) + ':' +
to_string(now->tm_sec) + " (UTC) Quick scan has been started for " +
msclr::interop::marshal_as<std::string>(textBox1->Text);
            ans += str;
            ans +=
msclr::interop::marshal_as<std::string>(Environment::NewLine);
            ans += "-----------------------------------------
----------------------------";
            ans +=
msclr::interop::marshal_as<std::string>(Environment::NewLine);
            ans += "+ Server: gws";
            ans +=
msclr::interop::marshal_as<std::string>(Environment::NewLine);
```

```
                ans += "+ X-XSS-Protection header has been set to
disable XSS Protection. There is unlikely to be a good reason for
this.";
                ans +=
msclr::interop::marshal_as<std::string>(Environment::NewLine);
                ans += "+ Cookie CONSENT created without the httponly
flag";
                ans +=
msclr::interop::marshal_as<std::string>(Environment::NewLine);
                ans += "----------------------------------------------
---------------------------- -";
                ans +=
msclr::interop::marshal_as<std::string>(Environment::NewLine);
                ans += "+Scan terminated : "+ to_string(vs.size())+ "
error(s) reported on remote host" +
msclr::interop::marshal_as<std::string>(Environment::NewLine);
                time_t t2 = std::time(0);
                std::tm* end = std::localtime(&t2);
                ans += "+ End Time : ";
                ans += to_string(end->tm_year + 1900) + '-' +
to_string(end->tm_mon + 1) + '-' + to_string(end->tm_mday) + ' ' +
to_string(end->tm_hour) + ':' + to_string(end->tm_min) + ':' +
to_string(end->tm_sec) + " (UTC) Quick scan has been started for " +
msclr::interop::marshal_as<std::string>(textBox1->Text);
                ans += "----------------------------------------------
---------------------------- -" +
msclr::interop::marshal_as<std::string>(Environment::NewLine);
                ans += "+1 host(s) tested" +
msclr::interop::marshal_as<std::string>(Environment::NewLine);



                textBox2->Text = gcnew System::String(ans.c_str());
```

Для аналізу файлів та виведення інформації було створено окремий клас для навчання нейронної мережі

```
myNeuro::myNeuro()
{
    inputNeurons = 100;
    outputNeurons = 2;
    nlCount = 4;
    list = (nnLay*)malloc((nlCount) * sizeof(nnLay));

    inputs = (float*)malloc((inputNeurons) * sizeof(float));
    targets = (float*)malloc((outputNeurons) * sizeof(float));

    list[0].setIO(100, 20);
    list[1].setIO(20, 6);
    list[2].setIO(6, 3);
    list[3].setIO(3, 2);

}
```

2.2 Опис функціоналу програми

Для запуску програми необхідно перейти вказаним шляхом (рис. 2.1) і запустивши site.exe

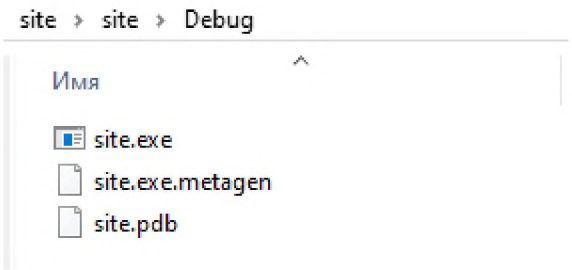

Рисунок 2.1 - Шлях до файлу

Джерело: власне

Перед нами з'явиться наступне вікно (рис. 2.2)

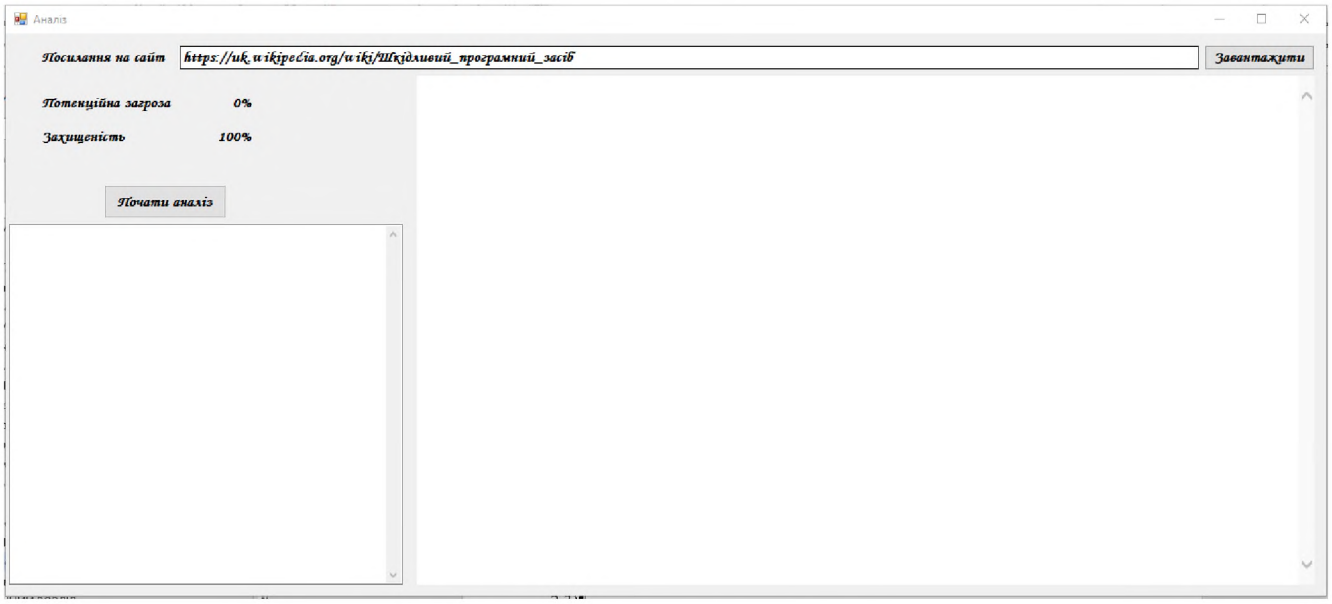

Рисунок 2.2 - стартове вікно

Джерело: власне

У верхній частині програми є можливість введення посилання на файл (рис. 2.3)

Рисунок 2.3 - поле для введення шляху

Джерело: власний

Після натискання на кнопку «Завантажити» відображається зміст вказаної сторінки (рис. 2.4), що знаходяться у вказаному шляху

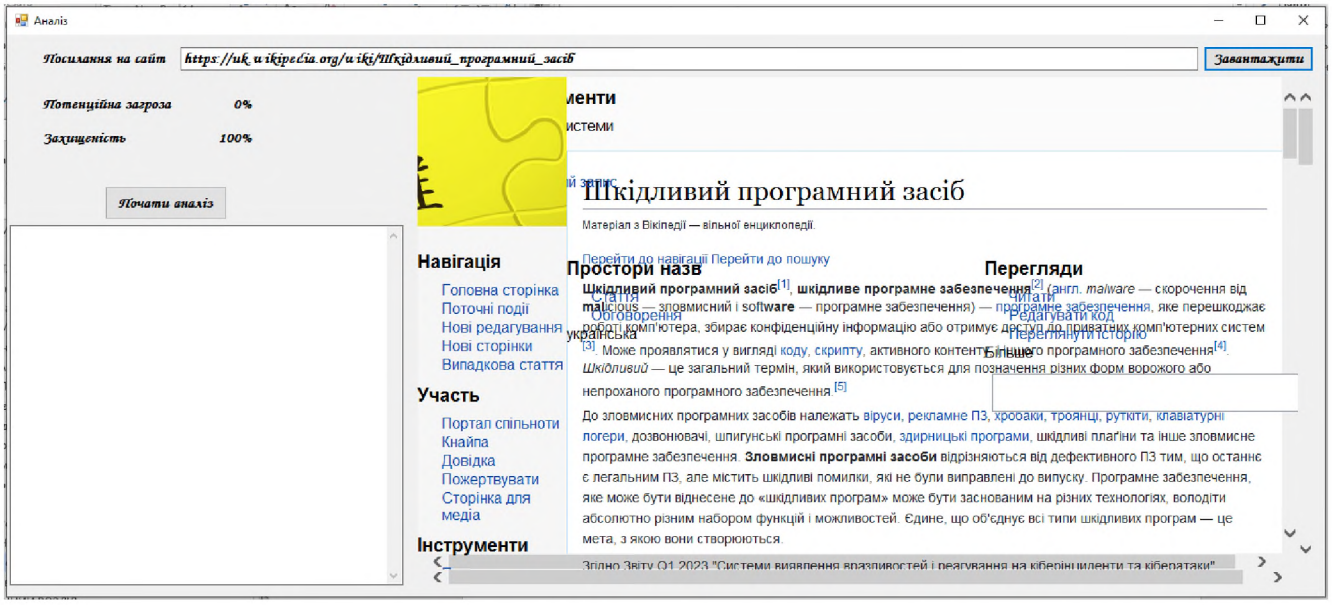

Рисунок 2.4 - Зміст сторінки

Джерело: власне

Для початку сканування на можливі загрози достатньо натиснути кнопку "Почати аналіз". Результати аналізу представлено на рис. 2.5.

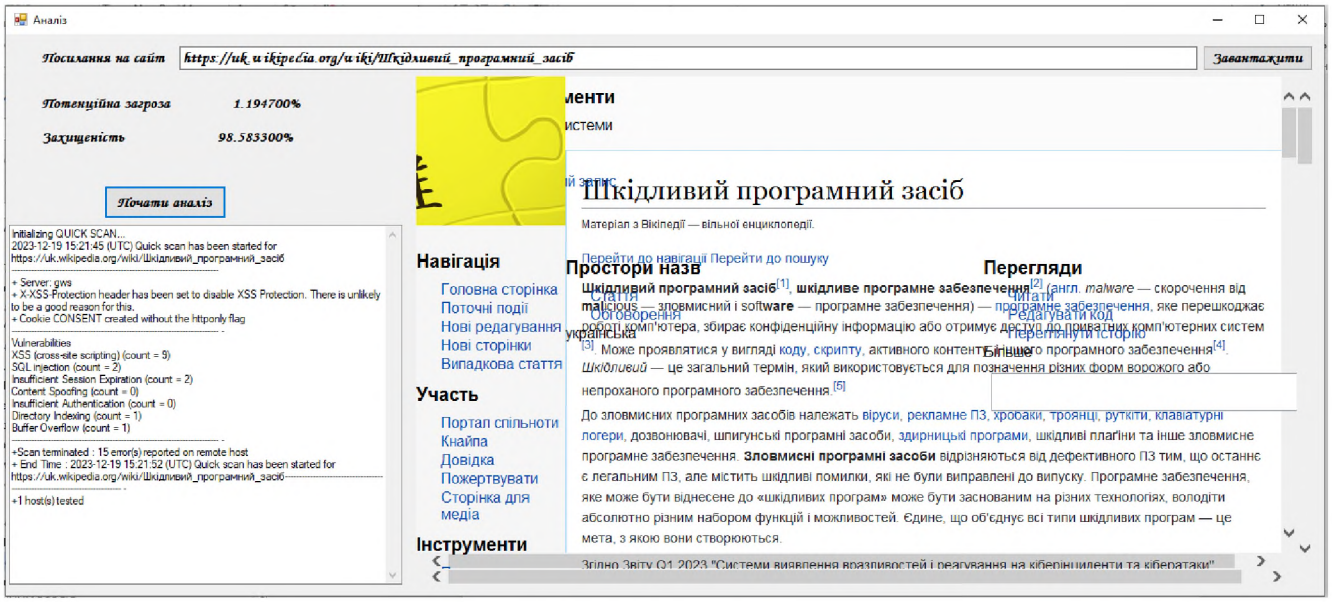

Рисунок 2.5 – результати аналізу.

Джерело: власне

2.3. Перевірка захисту

При тестуванні роботи програми на захист веб-сайтів, помилки, що виникають, відразу ж виправлялися. Нижче наведено кілька прикладів під час тестування (рис. 2.6-2.7).

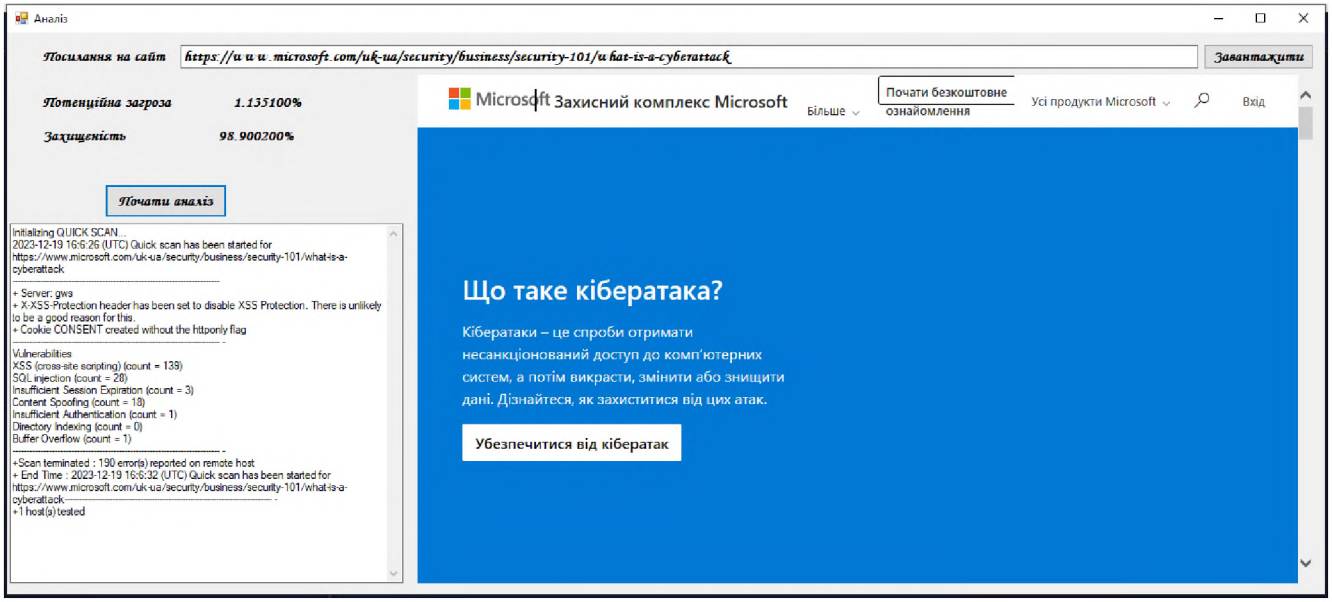

Рисунок 2.6 - тестування програми

Джерело: власне

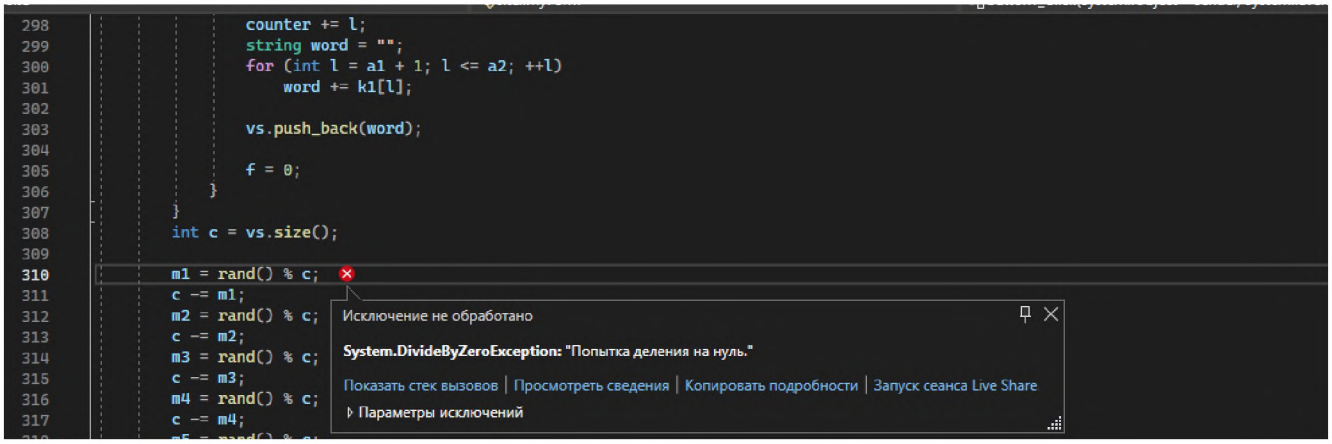

Рисунок 2.7 - тестування програми

Джерело: власне

3 ЕКОНОМІЧНИЙ РОЗДІЛ

3.1 Вибір та обґрунтування методики розрахунку економічної ефективності

Розробка нового програмного забезпечення передбачає розрахунок економічного ефекту від програмного продукту. Нова технологія повинна бути вигідна не тільки для розробника, але і для споживача, так як придбавши її, він сподівається зменшити витрати часу, ресурсів і т.п. Крім того, нова розробка має бути вигідною і для сторонньої особи – інвестора (якщо така існує), який сподівається не лише повернути вкладені кошти, а й отримати прибуток. Отже, необхідно оцінити економічний ефект всім сторін. Для цього в економічній частині слід розрахувати:

– скласти кошторис витрат на розробку програмного продукту;

– розрахувати експлуатаційні витрати, пов'язані з використанням нового програмного продукту;

– розрахувати обсяг роботи, який може бути виконаний із застосуванням нового програмного продукту;

– розрахувати річний економічний ефект від запровадження нового програмного продукту;

– розрахувати термін окупності витрат.

1. Основна заробітна плата розробників, що розраховується за формулою (3.1):

$$З_о = \frac{М}{Т_р} \times t \quad [грн.], \qquad (3.1)$$

де $М$ - місячний посадовий оклад конкретного розробника, руб.,

$Т_р$ - Число робочих днів у місяці (22),

t – кількість днів роботи розробника.

Розрахунки зведено до таблиці 3.1.

Таблиця 3.1

Назва посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	22000	500	30	30000
Інженер	11000	200	30	15000
Загалом				45000

2. Розрахунок додаткової заробітної плати.

Розраховується як 10-15% від основної зарплати розробників.

3. Розрахунок витрат на соціальні потреби.

Ставка нарахування єдиного соціального внеску у 2023 році становить 22% від фонду оплати праці за рік.

ЄСВ нараховується тільки на максимальну базу нарахування; якщо розмір заробітної плати перевищує цей розмір, то ЄСВ сплачується лише від максимальної бази, встановленої законом. Максимальна величина бази для нарахування єдиної соціальної допомоги, що дорівнює 25 розмірам прожиткового мінімуму для працездатних осіб, встановленого законом, на яку нараховується єдина соціальна допомога в розрахунку на місяць.

У разі якщо база нарахування єдиного соціального внеску є меншою від розміру мінімальної заробітної плати, встановленої законом на місяць, за який отримано дохід, сума єдиного внеску обчислюється виходячи з розміру мінімальної заробітної плати.

4. Сума амортизаційних відрахувань за умови рівномірного перерахування амортизації основних засобів на вартість виконаних робіт розраховується за формулою (3.2):

$$A = \frac{B_{поч.} - B_{л.}}{T_{к.в.} \times 12} [грн], \quad (3.2)$$

де $B_{поч.}$ - балансова вартість обладнання, грн.,

$B_{л}$ – ліквідаційна вартість обладнання, грн.

$T_{к.в.}$ – термін корисного використання обладнання, цілих місяців.

t – кількість місяців роботи обладнання.

5. Вартість матеріалів, використаних для розробки програмного продукту (додатку).

Проведені розрахунки зведені в таблицю 3.2.

Назва матеріалу, марка, тип, сорт	Ціна, грн	Ліквідна, грн	Витрачено	Вартість витрачених матеріалів, грн
Комп'ютер	25000	10000	3	5000
Периферійне обладнання	8000	1200	3	2266.6
Загалом				7266.6

6. Витрати на силову електроенергію розраховуються за формулою (3.3):

$$В_с = В \times П \times Ф[грн.] \quad (3.3)$$

де $В$ – вартість 1 кВт-години електроенергії, (тариф на електроенергію, що враховується станом на 2023 рік).

$П$ - встановлена потужність комп'ютера та інших пристроїв $П = 0{,}5кВт$;

$Ф$ - Фактична кількість годин роботи комп'ютера, $Ф = 126$ годин.

7. Інші витрати - приймаємо як 80-150% від основної заробітної плати розробників.

8. Сума всіх попередніх витрат дає загальні витрати на розробку лабораторних робіт

3.2 Розрахунок показників економічної ефективності

Розрахунок експлуатаційних витрат програмного продукту зводимо до таблиці 3.3.

Таблиця 3.3 Річна експлуатаційна витрата програмного продукту

	Статьи расходов	Сума, грн.
1	Заробітна плата обслуговуючого персоналу, руб	90000
2	Додаткова заробітна плата обслуговуючого персоналу	9000
3	Нарахування на заробітну плату	21780
4	Витрати електроенергію, крб.	410
5	Амортизаційні відрахування, руб.	7266
6	Витрати ремонт комп'ютерної техніки, руб.	3000
7	Інші витрати	13145
	Разом	144601

1. Розрахунки зарплати. Заробітна плата обслуговуючого персоналу з розрахуємо за формулою (3.4):

(3.4)

де, 12 – число місяців;

- Місячний посадовий оклад конкретного інженерно-технічного працівника гр.

– частка часу, який витрачає працівник виконання робіт із застосуванням програми – 0,5 (50%).

2. Додаткова заробітна плата обслуговуючого персоналу.

Розраховується як 10-12% від основної заробітної плати обслуговуючого персоналу.

3. Нарахування на зарплатню. Нарахування на заробітну плату у 2023 році становлять 22% від суми основної та додаткової заробітної плати розробника, тобто .

4. Витрати електроенергію. При живленні з мережі витрати на електроенергію розраховуються за такою формулою (3.5):

(3.5)

де - вартість 1 кВт-години електроенергії; (тариф згідно з інформацією станом на 2023 рік).

- Потужність комп'ютера разом з принтером та іншими приладами - 0,5 кВт;

Ф – фактична кількість годин роботи комп'ютера разом із принтером та іншими приладами на рік – 1900,9 годин;

- Коефіцієнт використання потужності;

- Частка часу, яке витрачає працівник на виконання конкретних робіт із застосуванням даної програми в загальному часі роботи - 0,5.

5. Амортизаційні відрахування. Амортизаційні відрахування розраховуємо за формулою (3.6):

$$A = \frac{B_{поч} - B_{л}}{T_{кв}} [руб], \quad (3.6)$$

де, $B_{поч}$ - балансова вартість персонального комп'ютера, руб.

$B_{л}$ - Ліквідаційна вартість, руб.

$T_{кв}$ - Термін корисного використання, цілі місяці

6. Витрати з ремонту комп'ютерної техніки. Витрати на ремонт комп'ютерної техніки можна розрахувати за формулою (3.7):

$$Р = [(0,04 \times 0,1) \times Ц + З_{р}] \times \beta \frac{руб.}{год} \quad (3.7)$$

де $Ц$ - балансова вартість апаратури, руб.;

$З_{р}$ - Заробітна плата окремо найманих робітників, зайнятих проведенням ремонтних робіт, руб.;

β - Частину часу, витрачається працівником на виконання конкретних робіт із застосуванням даного програмного продукту в загальному часі своєї роботи.

4-10% - від вартості обладнання складають матеріальні витрати на ремонт комп'ютерної техніки.

6. Інші витрати. Інші витрати приймаємо як 6-10% загальної суми попередніх витрат.

7. Експлуатаційні витрати під час використання програмного продукту. Сума витрат за всіма попередніми статтями і дає величину експлуатаційних витрат під час використання програмного продукту.

Обсяги робіт можна розрахувати за формулами:

$$Q_1 = \frac{F \cdot 60 \cdot \beta}{t_1} \tag{3.8}$$

і

$$Q_2 = \frac{F \cdot 60 \cdot \beta}{t_2} \tag{3.9}$$

де Q_1 - обсяг робіт при застосуванні старого програмного продукту, умовних одиниць, штук тощо;

Q_2 - Обсяг робіт при застосуванні нового програмного продукту, умовних одиниць, штук тощо;

F – ефективний фонд часу роботи протягом року $F = 1600$г.;

β – доля часу, витрачена працівником виконання конкретних робіт із застосуванням даного програмного продукту, у час своєї роботи;

t_i - Час виконання конкретної функції, хвилин.

Початкові дані:

F - Ефективний фонд часу роботи за годину;

β - Частка часу, яку витрачає працівник на виконання конкретних робіт із застосуванням даного програмного продукту, у загальному часі своєї роботи $\beta = 0{,}5$;

t_1 - Час виконання конкретної функції старого програмного продукту $t_1 = 10$хв.;

$t_2 = 4$хв. – час виконання конкретної функції нового програмного продукту.

$$Q_1 = \frac{1600 \cdot 60 \cdot 0{,}5}{10} = 4800$$

$$Q_2 = \frac{1600 \cdot 60 \cdot 0{,}5}{4} = 12000$$

Отже, видно, що застосування нового програмного продукту збільшує продуктивність і під час певної функції в $12000 / 4800 = 2{,}5$ раз.

$$\Delta E = \left(\frac{E_1}{Q_1} - \frac{E_2}{Q_2} \right) \cdot Q_2 \qquad (3.10)$$

де E_1 - Експлуатаційні витрати при використанні старого програмного продукту;

E_2 - Експлуатаційні витрати при використанні нового програмного продукту;

Q_1 - Обсяг робіт, що виконується за рік при застосуванні старого програмного продукту (умовні одиниці, кількість функцій і т.п.);

Q_2 - Обсяг робіт, що виконується через рік при застосуванні нового програмного продукту.

Термін окупності капіталовкладень розраховується за такою формулою:

$$T_o = \frac{B}{\Delta E} \text{ років} \qquad (3.11)$$

де B – загальна сума капіталовкладень;

ΔE - Річний економічний ефект від впровадження нового програмного продукту.

ВИСНОВКИ

В результаті написання дипломної роботи проведено дослідження та проектування автоматизованої системи тестування безпеки від кібератак. Встановлено, що тестування використовується веб-розробниками та адміністраторами безпеки для тестування та вимірювання рівня безпеки веб-додатку з використанням ручних та автоматизованих методів тестування безпеки. Основна мета тестування безпеки веб-застосунків – виявити будь-які вразливості або загрози, які можуть поставити під загрозу безпеку або цілісність веб-програми.

У ході написання роботи встановлено, що існуючі системи автоматизованого тестування недостатньо ефективні та функціональні для виявлення сучасних загроз та уразливостей, саме тому у роботі проведено розробку власної системи тестування. Аналіз сайту в системі відбувається за допомогою методів нейронної мережі. Текст сторінки записується в рядок, а потім через виклик методу класу відбувається аналіз даного рядка, після аналізу вказується можливий відсоток шкідливого контенту. Цей підхід є ефективним та функціональним, що дозволяє підвищити інформаційну безпеку веб-сайту.

У результаті написання роботи було виконано такі завдання:

1. Виконано огляд літератури на тему дослідження. Досліджено проблеми захисту веб-сайтів та визначено актуальність автоматизованого тестування;

2. Проведено огляд засобів захисту веб-сайтів та розглянуто системи автоматизованого тестування;

3. Виконано розробку та дослідження системи автоматизованого тестування безпеки веб-сайтів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Безпека веб-додатків: від вразливості до моніторингу [Електронний ресурс]: Режим доступу: https://habr.com/en/company/pentestit/blog/526878/ (дата звернення: 17.12.2023).

2. О.Бондаренко, І.Ушкалено. Безпека web-додатків: Актуальні проблеми та їх аналіз. Формування ринкової економіки Україні. 2017. Вип. Тридцять восьмий С. 28-36.

3. Третій Бабаш, А.В. Інформаційна безпека: Лабораторний практикум/О.В. Бабаш, Є.К. Баранова, Ю.М. Мельників. - М.: КноРус, 2019. - 432 с.

4. Web Application Security Consortium [Електронний ресурс]: Режим доступу: https://www. academia. Edu /11623665/ Web_ Application_ Security_ Consortium_ Threat_ Classification_WASC-TC_and_ISECOM_ Open_ Source_ Security_ Testing_ Methodology_Manual_OSSTMM_ (дата звернення: 17.12.2023).

5. Гришина, Н.В. Інформаційна безпеку підприємства: Навчальний посібник / Н.В. Гришин. - М.: Форум, 2018. - 118 с.

6. 2021 CWE Top 25 Most Dangerous Software Weaknesses [Електронний ресурс]: Режим доступу: http://cwe.mitre.org/ top25/archive/2021/ 2021_cwe_top25.html (дата звернення: 17.12.2023).

7. OWASP Top 10:2021 [Електронний ресурс]: Режим доступу: https://owasp.org/Top10/ (дата звернення: 15.11.2021).

8. OWASP Top 10:2021 List [Електронний ресурс]: Режим доступу: https://owasp.org/Top10/0x00-notice/ (дата звернення: 17.12.2023).

9. OWASP Top 10: 10 найнебезпечніших вразливостей [Електронний ресурс]: Режим доступу: https://cisoclub.ru/owasp-top-10-10-samyh-opasnyh-uyazvimostej/ (дата звернення: 17.12.2023).

10. Десятий Партико, Т.Л. Інформаційна безпека: Навчальний посібник/Т.Л. Партико, І.І. Попов. - М.: Форум, 2018. - 88 с.

11. Бенкен, Олена AJAX. Програмування для Інтернету (CD-ROM) / Олена Бенкен, Геннадій Самков. - М.: БХВ-Петербург, 2017. - 464 с.

12. Буділов, В. Основи програмування для Інтернету. Самовчитель/В. Будилів. - М.: БХВ-Петербург, 2016. - 634 с.

13. Баранова, Є.К. Навчальний посібник/Є.К. Баранова, А.В. Бабаш. - М.: Piop, 2017. - 400 с.

14. Heady R., Luger G., Maccabe A. та Servilla M. Network Level Intrusion Detection System. Computer Science Department, University of New Mexico, Tech. Rep. TR-90 1990. 21 p. URL: https://www.osti.gov/servlets/purl/425295 (дата звернення: 17.12.2023)

15. Anderson J. P. Computer Security Threat Monitoring and Surveillance. James P Anderson Co, Форт Washington, Pennsylvania, Tech. Rep., 1980. 56 p. URL: https://csrc.nist.gov/csrc/media/publications/conference- paper/1998/10/08/proceedings-of-the-21st-nissc-1998/documents/early-cs- papers/ande80. pdf (дата звернення: 18.12.2023)

16. Chandola V., Banerjee A., Кумар V. Anomaly Detection: A Survey. ACM Computing Surveys. 2009. Vol. 41, no. 3. P. 15-38.

17. Ghorbani A., Lu W., Tavallaee M. Network Intrusion Detection and Prevention: Concepts and Techniques. New York: Springer-verlag, 2009. 216 p.

18. Ning P., Jajodia S. Intrusion Detection Techniques. HBidgoli (Ed.), The Internet Encyclopedia, 2003. URL: https://doi.org/10.1002/047148296X.tie097 (дата звернення:18.12.2023)

ДОДАТОК А. Відомість матеріалів кваліфікаційної роботи

№	Формат	Найменування	Кількість листів	Примітки
Документація				
1.	А4	Реферат		
2.	А4	Список умовних скорочень		
3.	А4	Зміст		
4.	А4	Вступ		
5.	А4	Стан питання. Постановка задачі		
6.	А4	Спеціальна частина		
7.	А4	Економічний розділ		
8.	А4	Висновки		
9.	А4	Перелік посилань		
10.	А4	Додаток А		
11.	А4	Додаток Б		

ДОДАТОК Б. ВІДОМІСТЬ ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ

1 Титульна сторінка.doc

2 Завдання.doc

3 Реферат.doc

4 Список умовних скорочень.doc

5 Зміст.doc

6 Вступ.doc

7 Розділ 1.doc

8 Розділ 2.doc

9 Розділ 3.doc

10 Висновки.doc

11 Перелік посилань.doc

12 Додаток А.doc

13 Додаток Б.doc

Презентація.ppt