Міністерство освіти і науки України
Одеський національний політехнічний університет
Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій
Кафедра кібербезпеки та програмного забезпечення

Єржов Євгеній Олександрович,
студент групи РЗ-151

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Розробка системи виявлення кібервразливості інтернет-представництв підприємств, яка базується на аналізі звітів web-серверів

Спеціальність:
125 Кібербезпека

Спеціалізація, освітня програма:
Кібербезпека

Керівник:
Стопакевич О. А.,
к.т.н., доцент

Одеса – 2020

Міністерство освіти і науки України
Одеський національний політехнічний університет
Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій
Кафедра кібербезпеки та програмного забезпечення
Рівень вищої освіти другий (магістерський)
Спеціальність 125 – Кібербезпека
Освітня програма – Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри КБПЗ

_____
д.т.н.,проф. А.А.Кобозєва
____ _____202_р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Єржову Євгенію Олександровичу

1. Тема роботи: *Розробка системи виявлення кібервразливості інтернет-представництв підприємств, яка базується на аналізі звітів web-серверів.* керівник роботи *Стопакевич О. А., к.т.н., доцент,*

затверджені наказом ректора ОНПУ від „___”_____ 20__ р. №_____ .

2. Зміст роботи*: дослідження ефективності виявлення атак на веб-представництва на основі звітів веб-серверів, створення методики ефективного виявлення вразливостей веб-представництв на основі поєднання існуючих підходів та власного метода, розробка програмного продукту, охорона праці*.

3. Перелік ілюстративного матеріалу: *існуючі конкурентні програми для виявлення вразливостей, особливості кібервразливостей, ілюстрація роботи програмного інтерфейсу, результати роботи розробленого*

5. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
Охорона праці	Ярова І.А., к.т.н., доцент	Завдання видав	Завдання прийняв

6. Дата видачі завдання "_____"_____20___ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання	Примітка
1	*Аналіз джерел за темою випускної кваліфікаційної роботи*	*15-09-2019*	*виконано*
2	*Дослідження існуючих методів виявлення шкідливого програмного забезпечення*	*09-10-2019*	*виконано*
3	*Розробка власного методу детектування шкідливих програм*	*22-10-2020*	*виконано*
4	*Розробка методики ефективного виявлення шкідливих програм*	*03-11-2020*	*виконано*
5	*Реалізація програмного коду та інтерфейсу програми*	*15-11-2020*	*виконано*
6	*Підготовка тексту роботи*	*23-11-2020*	*виконано*
7	*Підготовка презентації та доповіді*	*30-11-2020*	*виконано*
8	*Попередній захист*	*02-12-2020*	*виконано*
9	*Нормоконтроль, рецензування*	*15-12-2020*	*виконано*
10	*Занесення роботи в електронний архів*	*24-12-2020*	*виконано*
11	*Допуск до захисту у завідувача кафедри*	*24-12-2020*	*виконано*

Здобувач вищої освіти _____ *Єржов Є. О.*

Керівник роботи _____ Стопакевич О. А.

ЗАВДАННЯ

на розробку розділу “Охорона праці та безпека в надзвичайних ситуаціях”

Єржову Євгенію Олександровичу, група РЗ-151

Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій

Кафедра кібербезпеки та програмного забезпечення

Тема роботи: *Дослідження ефективності звуколізоляції*

Зміст розділу:

1 Аналіз умов праці і вибір заходів і засобів захисту від небезпечних і шкідливих виробничих факторів.

2 Аналіз техногенних небезпек і вибір заходів і засобів забезпечення безпеки у надзвичайних ситуаціях.

3 Розрахунок ефективності звукоізоляції.

Керівник роботи	Консультант з охорони праці
_____ (_____)	_____ (_____)
«____» _____ 2020 р.	«____» _____ 2020 р.

АНОТАЦІЯ

Кваліфікаційна робота на тему «Розробка системи виявлення кібервразливості інтернет-представництв підприємств, яка базується на аналізі звітів web-серверів» на здобуття другого (магістерського) рівня вищої освіти за спеціальністю 125 – Кібербезпека, освітня програма: Кібербезпека, містить 24 рисунки, 4 таблиці, 1 додаток, 25 літературних джерел за переліком посилань. Робота виконана на 60 сторінках загального тексту і 50 сторінках основного тексту.

Метою роботи є підвищення ефективності виявляння шкідливого програмного забезпечення на основі методів евристичного аналізу.

Основними етапами виконання роботи є: аналіз існуючих аналогів аналізаторів логів, проектування свого програмного продукту, створення веб-сторінки на базі Wordpress та розробка програми зчитування та аналізу логів на основі звітів веб-сервера.

В роботі досліджені існуючі аналоги програмного забезпечення, розроблено метод аналізу та форматування звітів веб-серверів та пошуку їх вразливостей. Проведена успішна програмна реалізація запропонованої методики.

Результат виконання кваліфікаційної роботи – розроблено власну методику ефективного пошуку вразливостей веб-представництв на основі аналізу звітів веб-серверів.

ANNOTATION

Qualification work "Research of heuristic analysis methods of malware" for obtaining the second (master) level of higher education in the specialty 125 - Cybersecurity, educational program: Cybersecurity, contains 26 figures, 5 tables, 1 attachment, 25 literature sources in the list of references. The thesis consists of 60 pages of general text and 50 pages of the main text.

The aim of the work is to increase the efficiency of malware detection based on heuristic analysis methods.

The main stages of the work: analysis of existing analogs of log analyzers, development of a new software product, creation of web pages based on the Wordpress and development of applications for analyzing logs of web server.

The existing analogs of software are investigated in the work, the log analysis method is developed, the own technique of effective detection of vulnerabilities in veb server is created. The software implementation of the proposed technique was successfully performed.

ЗМІСТ

ВСТУП....8
1 ОГЛЯД ІСНУЮЧИХ АНАЛІЗАТОРІВ ЛОГІВ....10
1.2 Опис стеку технологій для розробки системи....13
2 ПРОЕКТУВАННЯ СИСТЕМИ....19
2.1 Мета та задачі ІС....19
2.2 Типи користувачів....20
2.3 Функціональні вимоги....21
2.4 Нефункціональні вимоги....22
2.5 Ідентифікація архетипу ІС....23
2.7 Уявлення слоїв ІС (Design View)....25
3 РОЗРОБКА ТА МОДЕЛЮВАННЯ ПРОГРАМНОЪ СИСТЕМИ....28
3.1. Віртуальна машина....28
3.2. Операційна система....28
3.3 Стек технологій LAMP....29
3.4 Створення веб-сайту....29
3.5. Застосування утіліти Apache bench для тестування сервера навантаженням....30
3.6. Застосування сканера вразливостей Arachni для перевірки сайту....30
3.7. Розроблена програма для аналізу логів веб-сервера Apache....31
3.8 Керування програмним кодом ІС....33
3.9 Розрахунок метрик програмного коду ІС....34
4 ОХОРОНА ПРАЦІ .... **Ошибка! Закладка не определена.**
ВИСНОВКИ.... **Ошибка! Закладка не определена.**

ВСТУП

Актуальність теми. При виявленні кіберзагроз найважливіше знати не процедури вияву і навіть не ТТП зловмисника, а хто саме стоїть за кіберзагрозою. Тактика, техніка та процедури зловмисників постійно розвиваються. Проте джерела кіберзагроз залишаються й надалі однаковими. Людський фактор ніхто не відміняв – завжди будуть ті, хто потрапить у пастку зловмсника. Проте зайвжди є той, й кому це не руку. Він і являє собою реальне джерело кіберзагроз.

Сьогодні найбільшу загрозу інформаційній безпеці для підприємств представляють національні держави, організовані кіберзлочинці і суб'єкти кібершпіонажу. Багато організацій мають труднощі з виявленням цих загроз через їх прихований характер, велику кількість використованих ресурсів і навми. Для підприємств ці більш витончені, організовані і постійні суб'єкти загроз видно тільки по цифровим слідах, які вони залишають після себе.

Веб-сайти підприємств та організацій кожного дня можуть потрапити під DDOS-атаку, яка зупинить на якійсь час повноцінне функціонування інтернет-представництва.

Таким чином задача дослідження та розробки методів виявлення вразливостей інтернет-представництв є актуальною.

Мета і завдання дослідження. Метою роботи є розробка компактного програмного застосування, яке зможе на основі логів веб-сервера виявити втручання зловмисників.

Досягнення поставленої мети потребує розв’язання таких задач:

– проаналізувати наявні аналоги аналізатора логів;

– дослідити ефективність існуючих методів і набір функцій для виявлення вразливостей;

– розробити власний продукт, який буде більш зручним за наявні аналоги вірешення проблеми;

– протестувати розроблене програмне застосування;

Об’єкт дослідження – процес виявлення вразливостей на основі аналізу логів веб-сервера інтернет-представництв.

Предмет дослідження – методи виявлення атак на веб-сервер.

Методи досліджень. Для вирішення задач використовувалися чисельні методи, окремі положення теорії машинного навчання. Оцінка отриманих результатів проводилася з використанням методів теорії ймовірностей і математичної статистики.

Наукова новизна одержаних результатів. Розроблено компактние програмне застосування, яке вирішує проблему виявлення атаки на веб-сервер більше ефективно за наявні аналоги, при цьому потребуючи менше ресурсів комп’ютеру.

Практичне значення отриманих результатів. На основі розробленої методики аналізу та обробки логів веб-серверів, розроблений продукт, який швидко та ефективно виявляє атаку на веб-сервер, потребуючи при цьому мало ресурсів.

1 ОГЛЯД ІСНУЮЧИХ АНАЛІЗАТОРІВ ЛОГІВ

1.1 Порівняння аналогів

Перед початком робочого процесу треба провести дослідження предметної області та знайти існуючи аналоги майбутньої системи. Це дозволить зрозуміти, які потреби до продукту існують на ринку, та як можна зробити здатний до конкуренції продукт.

Оскільки запланована система використовує досить сучасні принципи, треба порівняти заплановану систему як з існуючими, перевіреними часом комерційними продуктами, так і з експерементальними open-source проектами.

В результаті пошуку аналогів було знайдено три системи: Kiwi Syslog Server, Graylog Open Source та SolarWinds Papertrail

Оскільки наша система буде відкритою до модифікації, порівняння фізичних характеристик проводитися не буде. Основний упор буде зроблений на порівняння принципів і програмну складову.

Сервер SolarWinds® Kiwi Syslog® - це безкоштовний інструмент, який можете вдовольнити потреби організацій, які шукають рішення для централізованого управління SNMP-пастками та повідомленнями системних логів. (рис. 1.1.)

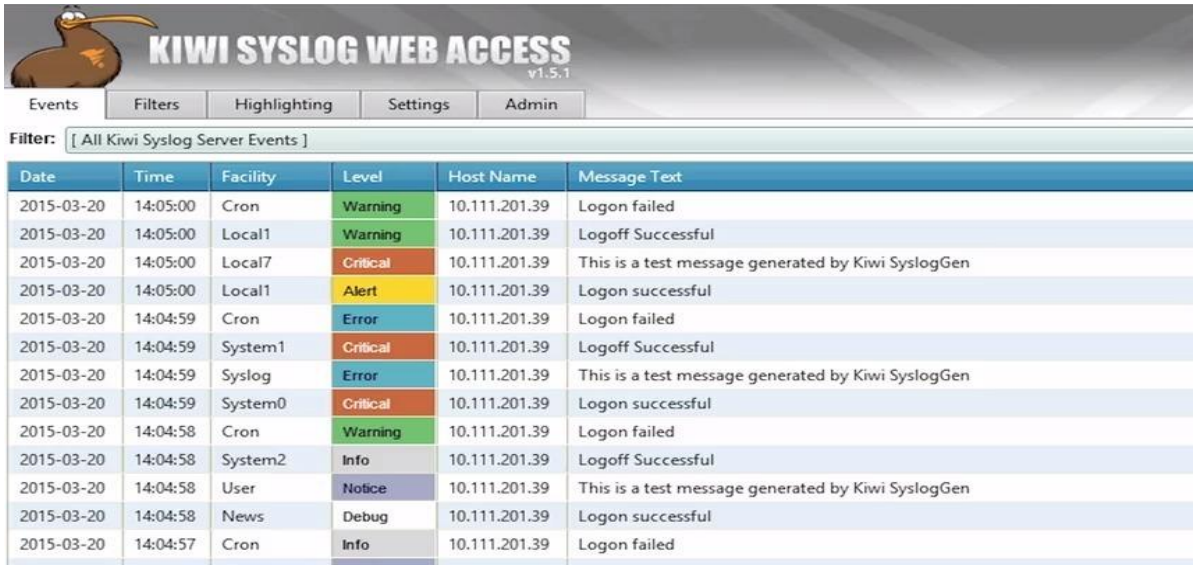


Рисунок 1.1 – Kiwi syslog server

Однак є обмеження – відстежувати логи можна лише з п'яти джерел. Інструмент дозволяє відстежувати мережеві пристрої, такі як маршрутизатори, брандмауери та робочі станції у вашій організації. Ви можете відстежувати у реальному часі всі повідомлення,сповіщення про перевищення трафіка та звіти з консолі. Крім того, платна версія інструменту дозволяє архівувати ваші логи або зберігати їх у базу даних.

Graylog представляє собою безкоштовну версію інструменту управління журналами, який збирає логи з вашого середовища за допомогою Sidecar-патерну.

Це дуже масштабний інструмент із простим користувальницьким інтерфейсом із функціями візуального аналізу журналу. Інструмент дозволяє швидко оглядати масивні частини логів за допомогою багатопотокових вузлів пошуку. Також можна налаштувати попередження та сповіщення, щоб стан справ у вашому середовищі був завжди відомий. Проте, функція співвідношення журналів з різних джерел у безкоштовній версії відсутня. Відсутність журналів аудиту у версії з відкритим кодом також може бути неприємним відкриттям для якихось організацій.

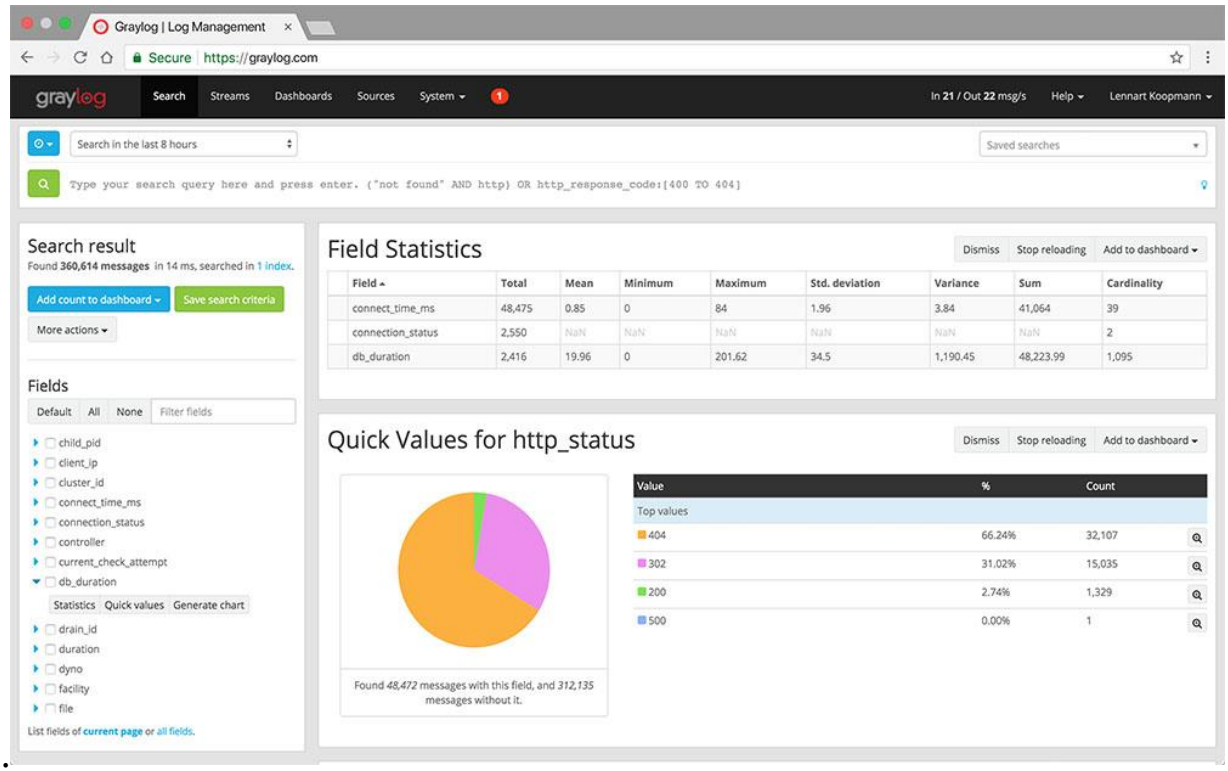


Рисунок 1.2 – Скріншот програми Graylog

Solarwinds Papertrail – це досить непоганий продукт для моніторингу журналів вашого сервера і відправки тривожних повідомлень по електронній пошті, щоб попередити вас або вашу команду, якщо в ваших журналах з'являться які-небудь аномалії. Ви можете налаштувати, як Papertrail буде себе поводити, зчитуючи ваші логи, наприклад ваш додаток може виводити повідомлення з попередженнями або помилками, коли вони будуть, а Papertrail буде стежити за тим, щоб кожен раз, коли це відбувається, ви отримували інформацію по електронній пошті. Також додаток може перехоплювати логи вашого веб-сервера.

Переваги: можливість налаштовувати сповіщення, оновлення подій та пошуку можна переглядати в режимі реального часу, зручна інсталяція

Недоліки:

– Відсутність пошукових шаблонів

– Маленький об’єм безкоштовного трафіку

– Відсутність швидкої інтеграції з відомими брендами веб-серверів (AWS, Azure, Apache)

– Відсутність розширеного парсингу (можливий лише текстовий пошук)

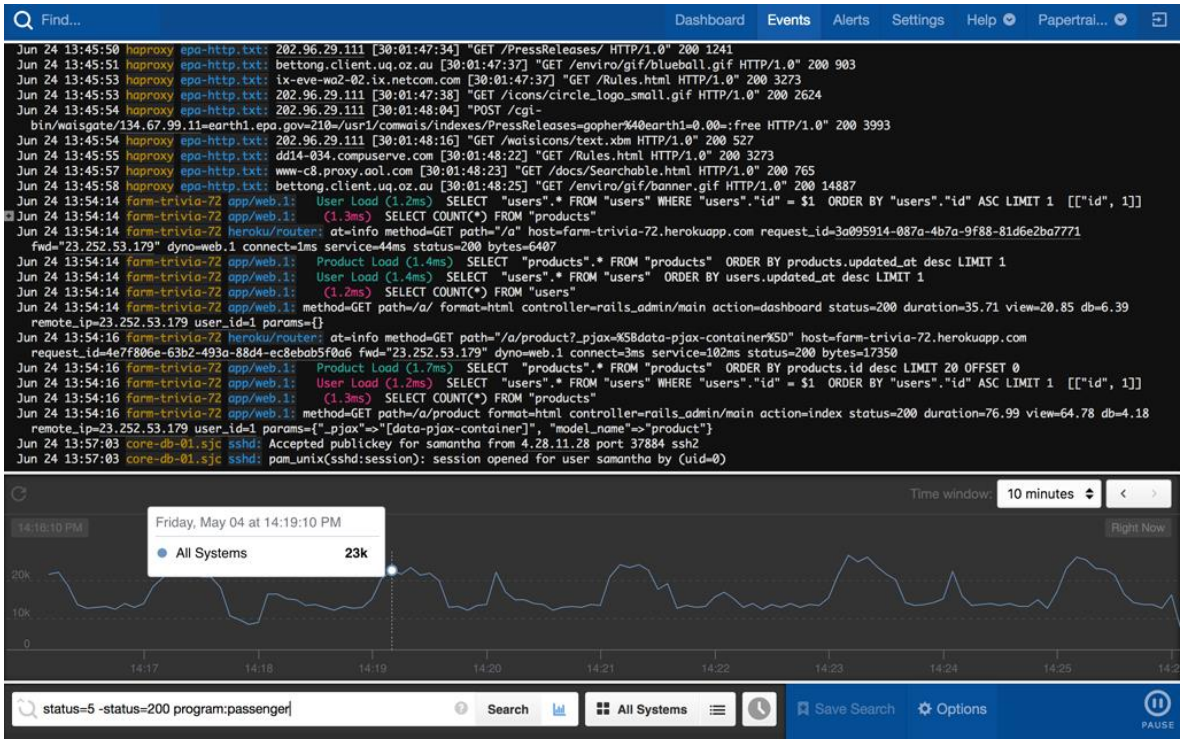


Рисунок 1.3 – Інтерфейс застосування Solarwinds Papertrail

В результаті дослідження декількох аналогів розробляємої системи можна зробити висновок, що всі представлені варіанти дуже відрізняються один від одного.

Для більш наглядного порівняння створимо таблицю 1.1 в якій відзначимо основні схожості та розбіжності систем.

Таблиця 1.1 – Результати порівнянь аналогів

Ознаки	Назва системи			
	Kiwi Syslog Server	Graylog Open Source	SolarWinds Papertrail	Власна система
Кросплатформовість	+	+	+	+
Можливість віддалено читати логи	-	-	-	+
Користування без реєстрації	-	+	-	+
Обмеження безкоштовного трафіку	+	-	+	+
Можливість розширенного парсингу логів	+	+	-	+
Можливість налаштування повідомлень про аномальну активність	-	-	-	+
Обмеження безкоштовного трафіку	-	-	50 МБ/місяць	-
Оптимізація під Apache	-	+	-	+
Можливість використання для комерційних проектів	-	+	-	+
Гнучка модифікація функцій	-	+	+	+

Таким чином розробляєма система є продовженням ідей кожної з приведених систем, об'єднуючи відкритість до модифікацій некомерційних і функціонал комерційних проектів, а також застосовуючи сучасні алгоритми і принципи для збільшення якості обробки інформації на виході системи.

1.2 Опис стеку технологій для розробки системи

Опис стеку технологій для розробки системи дано в табл. 1.2.

В результати проведення порівняння аналогів були розглянуті три системи,

що мають в собі схожий функціонал. На підставі їх переваг і недоліків було розроблено бачення системи, що дозволяє забезпечити всі необхідні функції, для розробки

Таблиця 1.2 – Стек технологій

Тип	Назва	Основні переваги	Призначення
Операційна система	Linux	Безкоштовність; потребує мало ресурсів; Захищеність; Зручна інсталяція; Зручна командна строка; Великий вибір графічних оточень; Продумана файлова система; Підтримка великої кількості архітектур; Відсутність збору даних	Linux - це операційна система, ядро якої поширюється на безкоштовній основі. Вона складається з ядра системи і набору невеликих програм, що взаємодіють з цим ядром.
IDE	Visual studio code	Великий вибір налаштувань (як всієї програми, так і інтерфейсу); розширювана бібліотека доповнень і готових рішень; мультифункціональність (редактор підтримує майже всі мови, які використовуються для створення додатків); простота і гнучкість.	Visual Studio Code – являє собою сучасне і зручне інтегроване середовище розробки програмного забезпечення від Microsoft, написане на основі платформи Electron і NodeJS.
Веб-сервер	Apache	Безкоштовний навіть для використання у комерційних цілях; гнучкий; надійний; кросплатформовість; оптимізований під Wordpress.	Веб-сервер Apache обробляє запити і обслуговує веб-ресурси через HTTP, отже застосування є для всіх у вільному доступі по звичайному URL.
СУБД	Mysql	Записи фіксованої і змінної довжини; Гнучка система привілеїв і паролів; Інтерфейс з мовами C і Perl, PHP; Швидка робота; Безкоштовна;	MySQL - вільна реляційна система управління базами даних.

Мова	PHP	Висока швидкість роботи; Висока ефективність ресурсів; Простий синтаксис; Відмінна сумісність і портативність.	PHP - скриптова мова, зазвичай застосовується для розробки веб-сторінок.

Також були розглянуті основні технології, які будуть потрібні на етапі реалізації системи. Всі технології є безкоштовними та розповсюджуються за відкритими, вільними ліцензіями, а тому їх використання не накладає ніяких обмежень.

1.3. Обгрунтування вибору технологій

При виборі стеку технологій – LAMP (Linux, Apache, Mysql, Php) чи WAMP (Windows, Apache, Mysql. Php), перш за все треба вирішити, яка операційна система буде більш зручноую і ефективною при користуванні.

1.3.1. Windows

Ціна: Microsoft Windows зазвичай коштує від 99,00 до 199,00 доларів США за кожну ліцензовану копію. Спочатку Windows 10 позіціонувався як безкоштовне оновлення для тодішніх власників Windows 7 або Windows 8.1, за умови що вони були оновлені до 29 липня 2016 року, але ця пропозиція більше не доступна.

Зручність користування: Windows - одна з найпростіших у використанні настільних операційних систем. Однією з основних його характеристик є зручність користування та невеликі потреби у системних ресурсах. Для звичайного користувача простота системи є скоріше плюсом. Однак більш досвідчені користувачі можуть бути розчаровані надмірним спрощенням системи.

Надійність: незважаючи на те, що за останні роки Microsoft Windows значно покращила надійність, вона вважається менш надійною, ніж Linux. За рахунок надмірної зручности для широкого кола користувачів система стає досить уразливою і нестабільною.

Програмне забезпечення: Windows має найбільшу кількість користувачів настільних комп'ютерів і, отже, найбільший вибыр програмного забезпечення. Він також значно переважає за кількістю відеоігор.

Безпека: за останні роки Microsoft значно вдосконалила безпеку Windows. Але оскільки ОС має найбільшу базу користувачів, переважно не досить досвідчених,

вона стає основною мішенню для зловмисників. Отже, Windows скоріше за всі інші ОС стане жертвою вірусів та шкідливих програм.

1.3.2. Linux

Ціна: ядро Linux, а також утиліти та бібліотеки GNU, якы наявны у більшості дистрибутивів, є повністю безкоштовними та відкритими. Ви можете завантажувати та встановлювати дистрибутиви GNU / Linux без покупки. Деякі компанії пропонують платну підтримку своїх дистрибутивів Linux, але базове програмне забезпечення залишається все одно безкоштовним.

Зручність користування: останні дистрибутиви Linux простіші у використанні, ніж попередні варіанти. Деякі дистрибутиви Linux мають графічний інтерфейс, подібний до Windows, що дозволяє зручно користуватися звичайному користувачу комп'ютера. Дистрибутиви графічного інтерфейсу Linux більш зручні для користувача і не містить купи непотрібніх функцій яких, як відомо, досить багато у Windows. Прикладами простіших у використанні дистрибутивів є Ubuntu та Linux Mint.

Надійність: ця ОС, як відомо, надійна та безпечна. У ній увага приділяється переважно управлінню процесами, безпеці системи та тривалості роботи. В Linux зазвичай значно менше багів, аніж у інших ОС.

Програмне забезпечення: Для Linux доступні тисячі програм і багато з них дуже легко можна встановити - і все безкоштовно. Крім того, багато програм Windows можуть працювати на Linux, використовуючи рівні сумісності, такі як WINE. Також Linux підтримує ширший спектр некомерційного програмного забезпечення, ніж Windows.

Безпека: це надзвичайно безпечна операційна система. Хоча атаки на неї й проводяться час від часу, його вихідний код відкритий і доступний для перегляду будь-яким користувачам, що полегшує виявлення та виправлення вразливостей

Отже, підсумовуючи це все, я обрав саме стек LAMP. Для чого може знадобитися LAMP?

– В разі потреби середи для веб-розробки і тестування додатків, які написані для LAMP. Як для власних додатків, так і для будь-яких необхідних Вам CMS. Ви можете працювати на Windows і на Ubuntu;

– якщо Вам необхідно зробити продуктивний веб-сервер для хостингу своїх проектів на віртуальному VPS сервері або на виділеному сервері;

– якщо Вам потрібен сервер для систем контролю версій;

– для самонавчання адмініструванню;

– в економічних цілях, для створення власного сервера.

Установка LAMP (Linux + Apache + MySQL + PHP / Perl / Python) є досить широко розповсюдженим варіантом налаштування серверів з Ubuntu. Є велика кількість додатків, які мають відкритий вихідний код і написані з використанням стека технологій LAMP. Популярні програми на LAMP: wiki енциклопедії, системи управління вмістом (CMS) і керуючі додатки, наприклад, phpMyAdmin.

Важлива перевага LAMP це досить великий вибір альтернативних баз даних, web серверів і мов сценаріїв. Актуальною заміною для MySQL служать PostgreSQL і SQLite. Python, Perl і Ruby можна замінити PHP. А Nginx, Cherokee і Lighttpd альтернатива Apache.

Для швидкого встановлення LAMP використовується tasksel. Tasksel - інструмент Debian / Ubuntu, який встановлює кілька залежних пакетів в вашу систему в якості єдиного завдання.

2 ПРОЕКТУВАННЯ СИСТЕМИ

2.1 Мета та задачі ІС

Розробляєма інформаційна система – це застосування, основною функцією якого є зчитування, обробка та аналіз логів веб-сервера Apache.

Мета ІС: інформаційна система повинна зчитувати логи веб-сервера Apache, вміти аналізувати, форматувати та виводити наявні вразливості.

Цільова аудиторія: розробники некомерційних або комерційних веб-додатків, якім потрібно у короткі сроки перевірити лог веб-сервера або великий масив логів (у випадку великих компаній)

Головними функціями застосевання є:

– Конфігурація системи зчитування логів веб-сервера Apache;

– Обробка наявних логів;

– Форматування логів та зберігання їх у вигляді структури данних;

– збереження результатів спостереження.

Інформаційні потоки ІС зображені на рисунку 2.1.

Можна побачити, що на виході користувач отримує два типи інформаційних одиниць, тобто результати зчитування логів про наявні вразливості та структуру даних з інформацією про роботу сервера.

Рисунок 2.1 – Інформаційні потоки розробляємої ІС.

Результати інформації у вигляді даних про наявні вразливості разом із структурованими логами утворюють сутність «Знаходження».

Знаходження повинен мати наступні атрибути: IP-адреси відвідувачів; мітка часу; тип запиту; статус запиту; розмір запиту; список підозрілих IP-адрес;

Список підозрілих IP-адрес – це IP-адреси, з яких здійснюється кількість запитів більша за гранично допустиму.

2.2 Типи користувачів

В розробляємій інформаційній системі можливий лише один тип користувачів – це оператор системи.

Повний функціонал програми доступний адміністратору та авторизованим користувачам, якщо користувач не має аккаунту, йому необхідно буде зареєструватися, однак він може швидко перевірити збережені логи при потребі. Інформаційна система має певні функції, які на мові UML вказуються на діаграмі варіантів використання. Найголовніші варіанти використання наведені в табл. 2.1.

Таблиця 2.1 – Визначені варіанти використання

Код	Основний актор	Найменування
A1	Оператор системи	Запустити середовище виконання програми
A2	Оператор системи	Знайти потрібний лог на носії інформації
A3	Оператор системи	Виставити чергу логів для обробки
A4	Оператор системи	Запустити зчитування логу
A5	Оператор системи	Вивести підозрілі IP-адреси
A6	Оператор системи	Зберегти лог у вигляді структури даних
A7	Оператор системи	Провести пошук даних необхідних IP-адрес
A8	Оператор системи	Відсортувати запити за часом
A9	Оператор системи	Зберегти структуру даних як текстовий файл

2.3 Функціональні вимоги

Вимога 1 – налаштування зчитування логів

F1.1 Функція налаштування системи зчитування логів

FR1.1.1 Оператор системи має можливість обрати класи об’єктів, інформація про виявлення яких повинна бути збереженаю. Назви класів повинні бути розміщені у файлі формату .txt на носії, що підтримує інтерфейс USB мінімальної версії 2.0. Файл повинен бути розміщений у корені носія. Назва файлу має бути “Record.txt”.

FR1.1.2 Обмеження: назви класів повинні міститися у списку підтримуваних класів, що можно знайти у документації до пристрою. Класи, що не містяться у списку підтримуваних класів, будуть проігноровані.

FR1.1.3 Обмеження: якщо внутішній або зовнішній носій відсутній, зчитування логів проводитися не буде.

FR1.1.4 Обмеження: якщо не знайдено жодного підтримуваного класу, зчитування логів проводитися не буде.

F1.2 Функція управління статусом обробки логів

FR1.2.1 Оператор системи має можливість вмикати та вимикати обробку, відповідно починаючи чи припиняючи роботу застосунку. Аналіз логів запускається автоматично, ПЗ додано до списку автозавантаження системи.

FR1.2.2 При наявності підключеного зовнішнього носія що підтримує інтерфейс USB мінімальної версії 2.0, на носії буде автоматично створена папка «LogRecords». У створену папку будут зберігатися вихідні текстові файли.

Вимога 2 – отримання інформації про типи підтримуваних логів

F2.1 Функція отримання даних веб-сервера

FR2.1.1 Оператор системи має можливість отримати типи підтримуваних логів, а саме: їх назву та інформацію про їх призначення.

FR2.1.2 При відсутності підтримки логу, буде відображено відповідне повідомлення.

F2.2 Функція отримання файлу структури даних

FR3.2.1 Оператор системи має можливість отримати файл структури даних за

будь-який період або за увесь час. Початок та кінець періоду зазначаються розміром логу.

FR2.2.2 Після натискання на кнопку «Запуск», користувач побачить детальну інформацію про кожну IP-адресу, з якої робилися запити.

F2.3 Функція збереження списку знаходжень

FR2.3.1 Оператор системи має можливість зберегти отриманий раніше список оброблених запитів як txt-файл. Назвою txt-файлу має бути зазначення періоду створення списку. Для повного списку відповідна помітка «Complete log».

FR2.3.2 Обмеження: якщо в сесії немає попередніх запросів на отримання списку знаходжень, кнопка збереження txt-файлу має бути вимкнена.

Вимога 3 – проведення тестового зчитування логу

F3.1 Функція проведення тестового розпізнвання

FR3.1.1 Оператор системи має можливість протестувати якість розпізнавання аномальної активності, для цього можна використати на веб-сервері сканер вразливостей, н. У разі успішного розпізнавання буде відображено кількість знайдених об’єктів кожного підтримуваного класу та вихідне зображення з нанесеними обмежувальними паралелепіпедами.

2.4 Нефункціональні вимоги

NF1. Застосування повинно працювати незалежно від наявності підключення до мережі интернет.

NF2. Застосування повинно бути мати максимально автономним, тобто здатним продовжувати роботу в разі помилок, втрати з'єднання до мережі Інтернет або закінчення місця на накопичувачі.

NF3. Застосовання має швидко працювати та мати зручний інтерфейс користувача.

NF4. Серверна частина застосовання має працювати завдяки будь-якому актуальному веб-браузеру.

Вимоги визначають можливості застосування, отже на підставі списку вимог можна побудувати діаграму варіантів використання (рис 2.2) ІС.

Для детальнішого розгляду процесу є можливість побудови додактові, більш детальніші диаграми використання. Наприклад, побудуємо діаграму прецедентів для блоку отримання інформації з пристрою (рис. 2.3).

Ця діаграма надасть нам більше розуміння про можливі варіанті поведінки користувача у обраному сценарії. Аналогічні діаграми можут бути запрошено для будь-якого процесу.

2.5 Ідентифікація архетипу ІС

Інформаційна система має клієнт-серверну структуру.

Рисунок 2.2 – Діаграма варіантів використання ІС

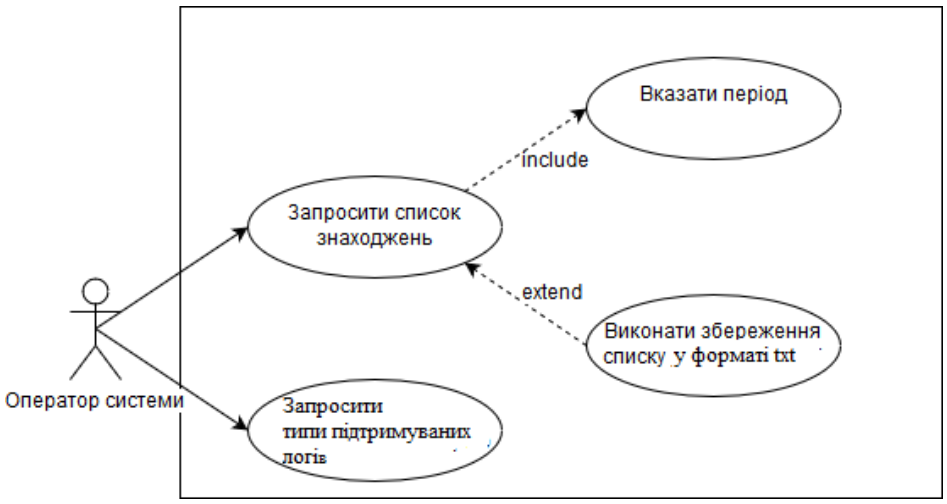


Рисунок 2.3 – Діаграма прецедентів для блоку отримання інформації з логів

2.6 Логічне уявлення про ІС (Logical View)

Для уявлення майбутньої системи, побудуємо логічну діаграму зв’язку компонентів системи (рис. 2.7). Модель складається з оператора системи, що взаємодіє з програмою; програми, що зчитує логи та відображає результати; серверу, який відповідає на запити, керує процесом роботи веб-сайту та зберігає результат до бази даних; веб-сайту, на якому ми проводимо тестування; та бази даних. Тобто на цій діаграмі можна побачити взаємодію користувача із розробленою ІС та взаємодію розробленої ІС із логами сервера.

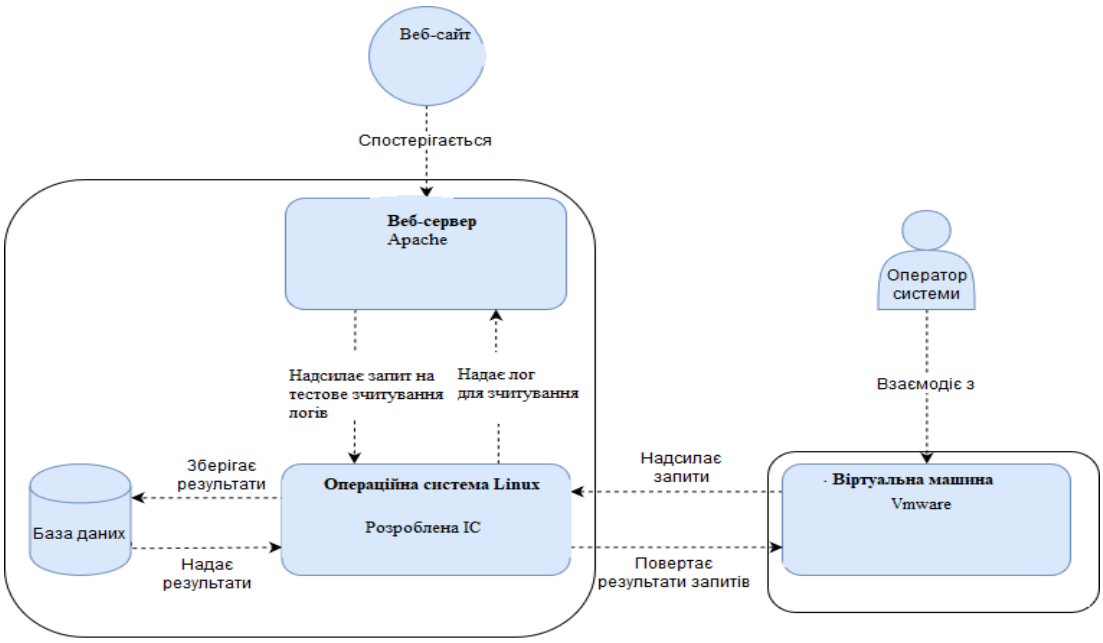


Рисунок 2.7 – Схема логіки роботи системи та зв’язку компонентів

2.7 Уявлення слоїв IC (Design View)

Серверна частина поділена на дві частини: безпосередньо серверний процес та робочий процес розпізнавання Intel NCS.

Серверний процес складається з наступних рівнів (рис 2.10):

– Application Layer (містить основні сервіси та функції серверу; саме ці компоненти створюють основну бізнес-логіку застосування;);

– Domain Layer (сутності та сховища; основні бізнес-об’єкти застосування);

– Communication Layer (сервіси для взаємодії застосування з зовнішнями компонентами підсистеми).

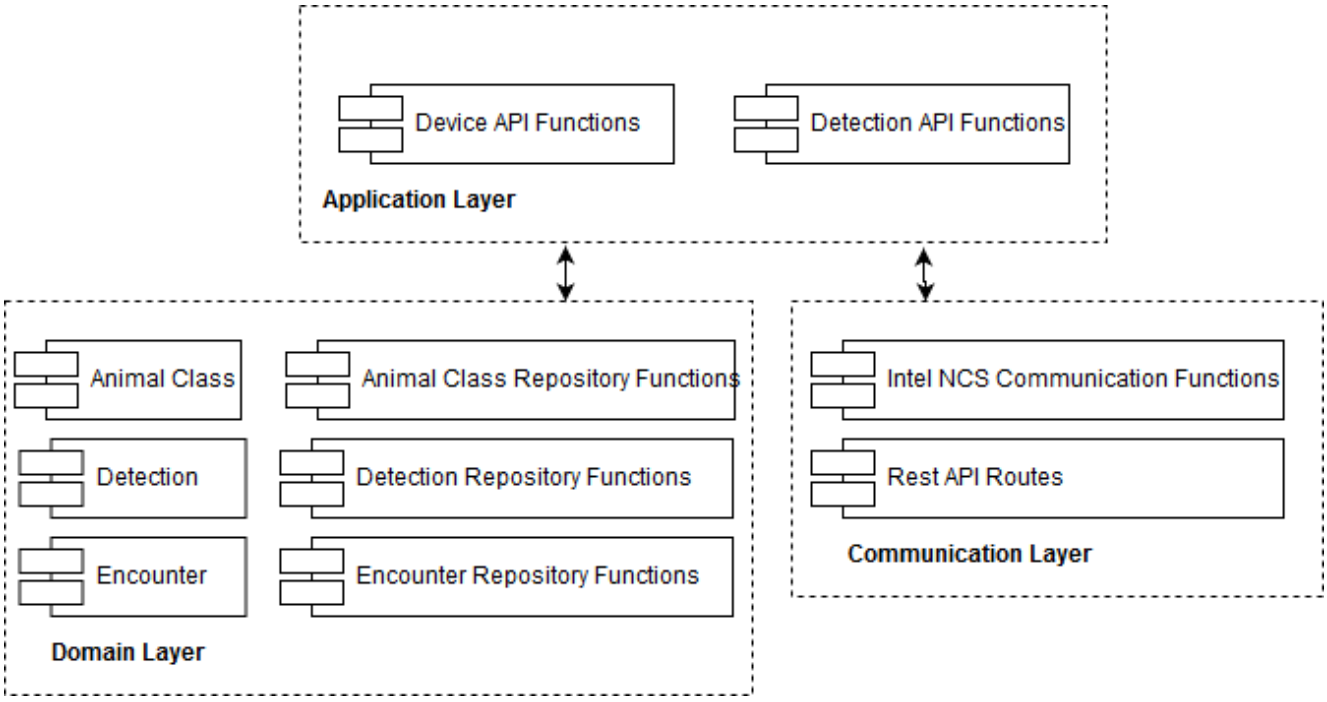
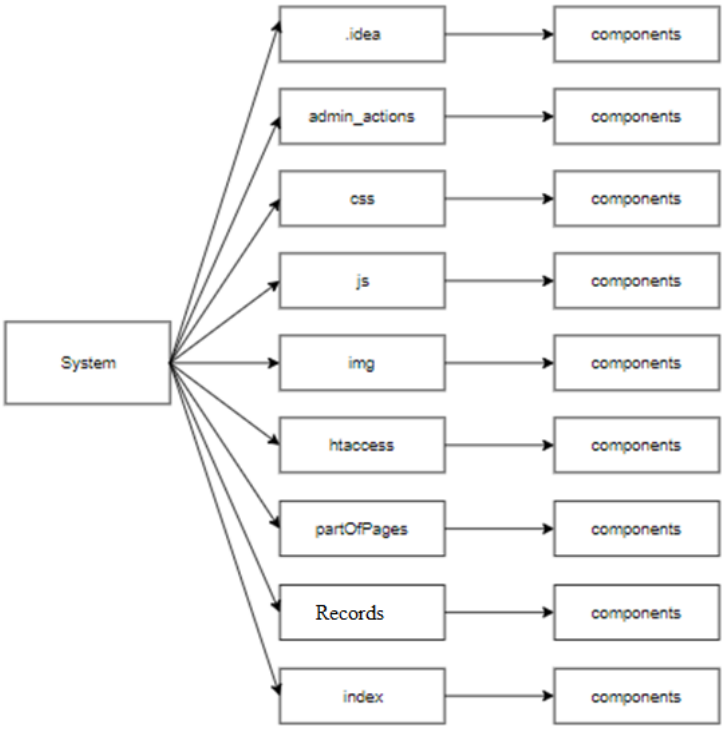


Рисунок 2.8 – Діаграма слоїв IC (Design View), серверна частина

Було проведено проектування програмного забезпечення клієнт-серверного веб-застосування, де були сформовані функціональні та нефункціональні вимоги, побудована діаграма прецедентів, яка ілюструє ці вимоги.

Розглянуті питання операційного представлення, був описаний стек технологій, що задіяний у розробці даної інформаційної системи.

2.8 Уявлення про структуру проекту IC



2.9 Уявлення про структуру проекту ІС

2.11 Уявлення комунікацій ІС (Communication View)

Для зрозумілого уявлення комунікацій, приведемо діаграму комунікацій ІС (рис 2.19)

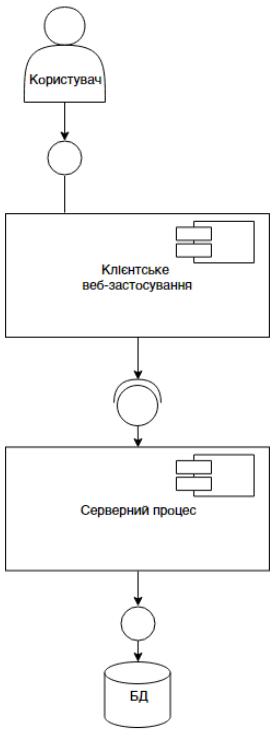


Рисунок 2.10 – Діаграма комунікації ІС (Communication View)

2.12 Уявлення про класи ІС

Для кращого розуміння зв’язку між компонентами необхідно представити загальну діаграму класів(рис 3.3).

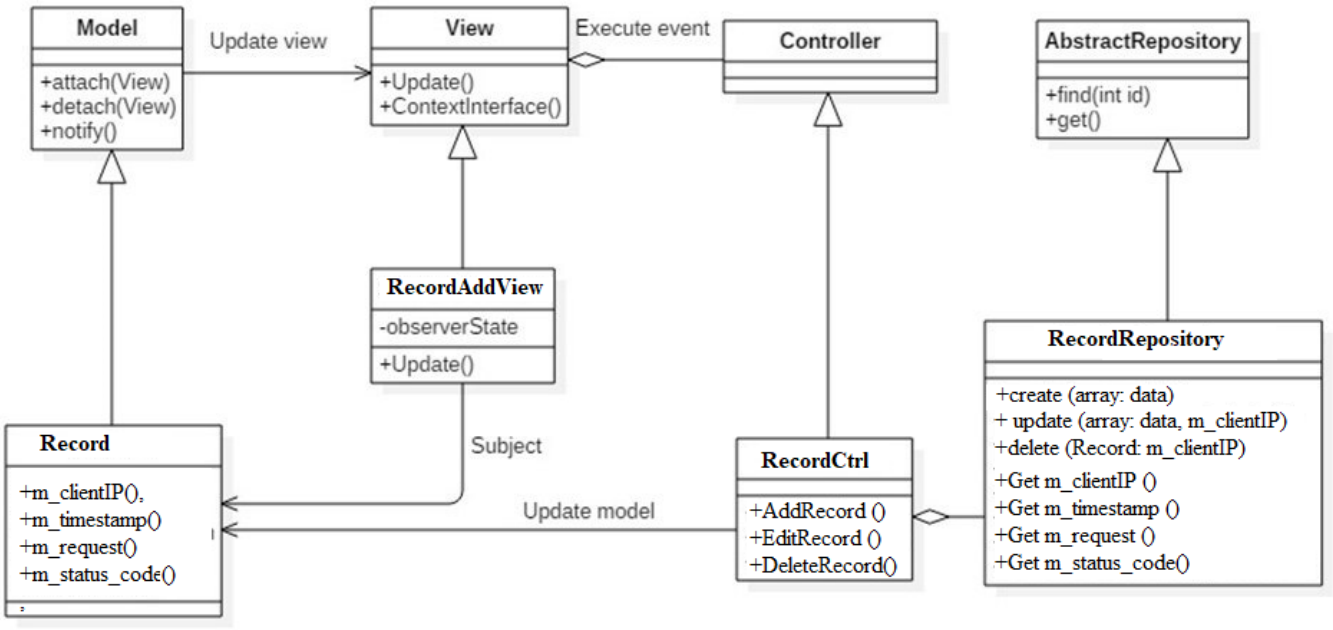


Рисунок 2.11 – Діаграма класів

3 РОЗРОБКА ТА МОДЕЛЮВАННЯ ПРОГРАМНОЪ СИСТЕМИ

3.1. Віртуальна машина

Було встановлено ПЗ VMware Workstation. Воно надає можливість користувачу встановити одну чи декілька віртуальних машин на один фізичний комп'ютер і запускати їх паралельно з ним. Кожна віртуальна машина може виконувати свою операційну систему, включаючи Microsoft Windows, Linux, BSD, і MS-DOS. VMware Workstation розроблена і продається компанією VMware, підрозділом EMC Corporation.

3.2. Операційна система

Для розробки була використана операційна система Linux Ubuntu. Головні особливості дистрибутива Ubuntu:

– стабільність роботи - систему часто використовують на дуже навантажених серверах. ОС не вимагає частих перезавантажень комп'ютера, навіть у випадку оновлень, установки або видалення програм.

– безпечна система, що не потребує антивірусів. Щоб спіймати вірус, треба прикласти дуже багато зусиль, якщо Ви використовуєте цей дистрибутив.

– безкоштовна ОС - інсталяція відбувається в кілька кліків, в мережі повно версій для безкоштовного завантаження, можна використовувати на безлічі комп'ютерів одночасно.

– приємний і зрозумілий інтерфейс - велика кількість безкоштовних тим і ефектів, на будь-який смак і колір.

– стабільність роботи забезпечується великою командою підтримки, яка швидко знаходить і виправляє баги.

– передбачуваність системи - кожен новий реліз виходить з періодичністю в 6 місяців, користувачі завжди мають доступ до свіжих версій ОС.

– відмінно уживається на одному ПК з іншими ОС, наприклад, Windows.

3.3 Стек технологій LAMP

Цей набір технологій встановлюють на сервер для відображення динамічних веб-сайтів і веб-додатків. Він дозволяє встановити на ваш сервер переважну більшість типів веб-сайтів і мережевого програмного.

Комбінація LAMP - це: серверна ОС Linux; веб-сервер Apache, для якого створено безліч додаткових модулів, що вирішують проблему спільної роботи веб-сервера і сценаріїв, написаних на різних мовах програмування; серверна мова динамічних сценаріїв PHP (або Perl); СУБД MySQL, яка демонструє відмінну швидкість виконання SQL-запитів і ідеально підходить для малих і середніх проектів.

3.4 Створення веб-сайту

Веб-сайт був створений на базі Wordpress.

WordPress - система керування змісту сайта з відкритим вихідним кодомю. Сфера застосування - від блогів до досить складних ресурсів з новинами. Вбудована система «тем» і «плагінів» та вдала архітектура дозволяють конструювати проекти широкої функціональної складності.

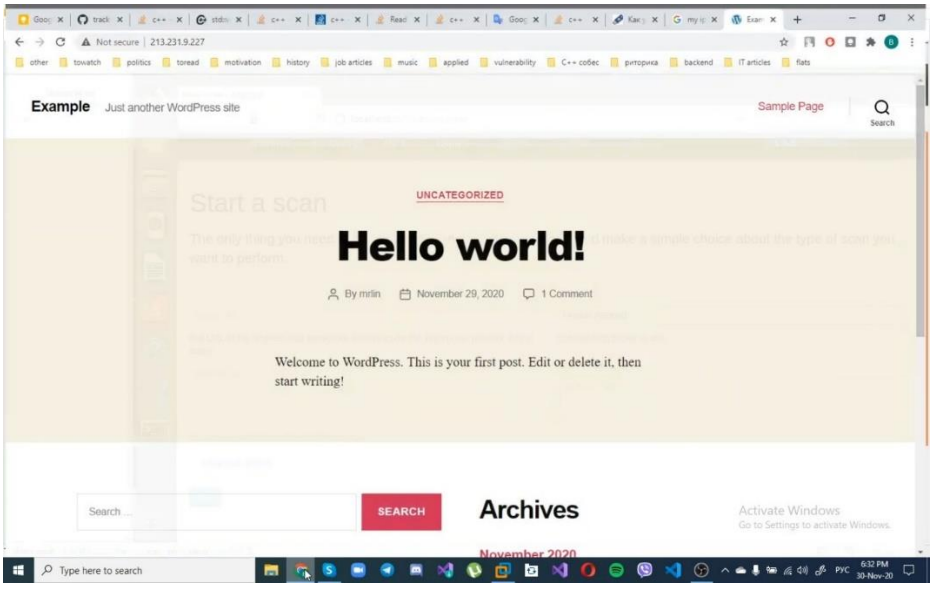


Рисунок 3.1 – Сайт для тестування розробленого ПЗ

3.5. Застосування утіліти Apache bench для тестування сервера навантаженням

ApacheBench - однопотокова програма для командної строки, що використовується для вимірювання продуктивності HTTP веб-серверів. Спочатку розроблена для тестування HTTP-сервера Apache, але в цілому підходить для тестування будь-якого веб-сервера.

Важливо: Apache Bench слід застосовувати тільки на тих серверах, якими ви володієте або на які у вас є явний дозвіл на тестування. У деяких країнах використання Apache Bench на несанкціонованих сайтах може вважатися злочином і карається законом.

```
Time taken for tests: 1.889 seconds
Complete requests: 100
Failed requests: 0
Write errors: 0
Total transferred: 1003100 bytes
HTML transferred: 949000 bytes
Requests per second: 52.94 [#/sec] (mean)
Time per request: 188.883 [ms] (mean)
Time per request: 18.888 [ms] (mean, across all concurrent requests)
Transfer rate: 518.62 [Kbytes/sec] received

Connection Times (ms)
min mean[+/-sd] median max
Connect: 57 59 1.7 59 64
Processing: 117 126 7.5 124 162
Waiting: 57 62 7.0 60 98
Total: 175 186 8.0 184 224

Percentage of the requests served within a certain time (ms)
50% 184
66% 186
75% 187
80% 188
90% 192
95% 203
98% 216
99% 224
100% 224 (longest request)
```

Рисунок 3.2 – Приклад логу Apache bench

3.6. Застосування сканера вразливостей Arachni для перевірки сайту

Це потужний безкоштовний інструмент для тестування захищеності веб-додатків і пошуку вразливостей. Має графічний інтерфейс і величезну функціональність, про яку більш докладно можна почитати на офіційному сайті.

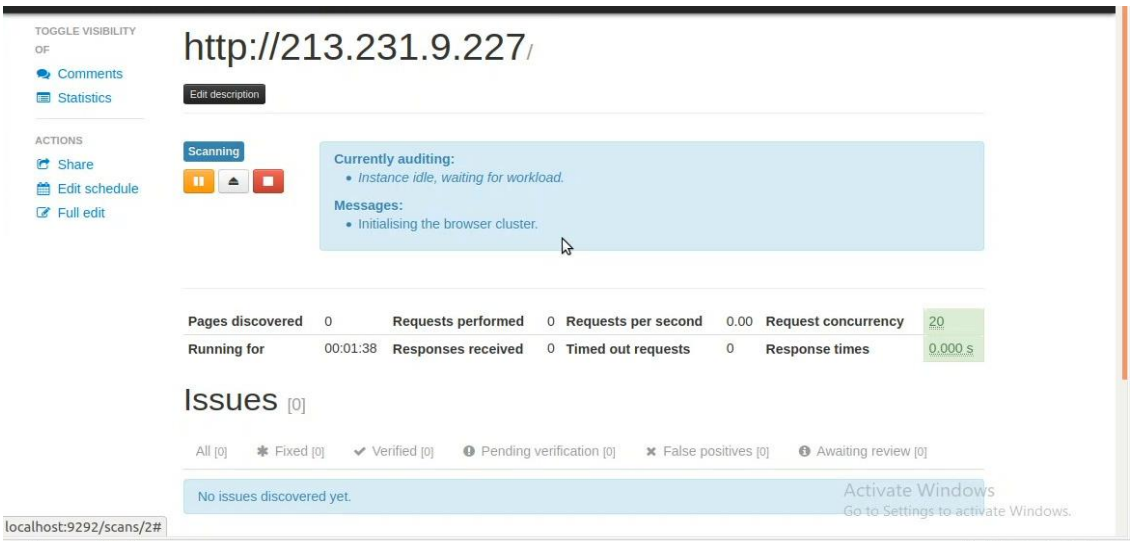


Рисунок 3.3 – Перевірка сайту сканером вразливостей

3.7. Розроблена програма для аналізу логів веб-сервера Apache

Перш за все, треба компілювати програму за допомогою команди make.

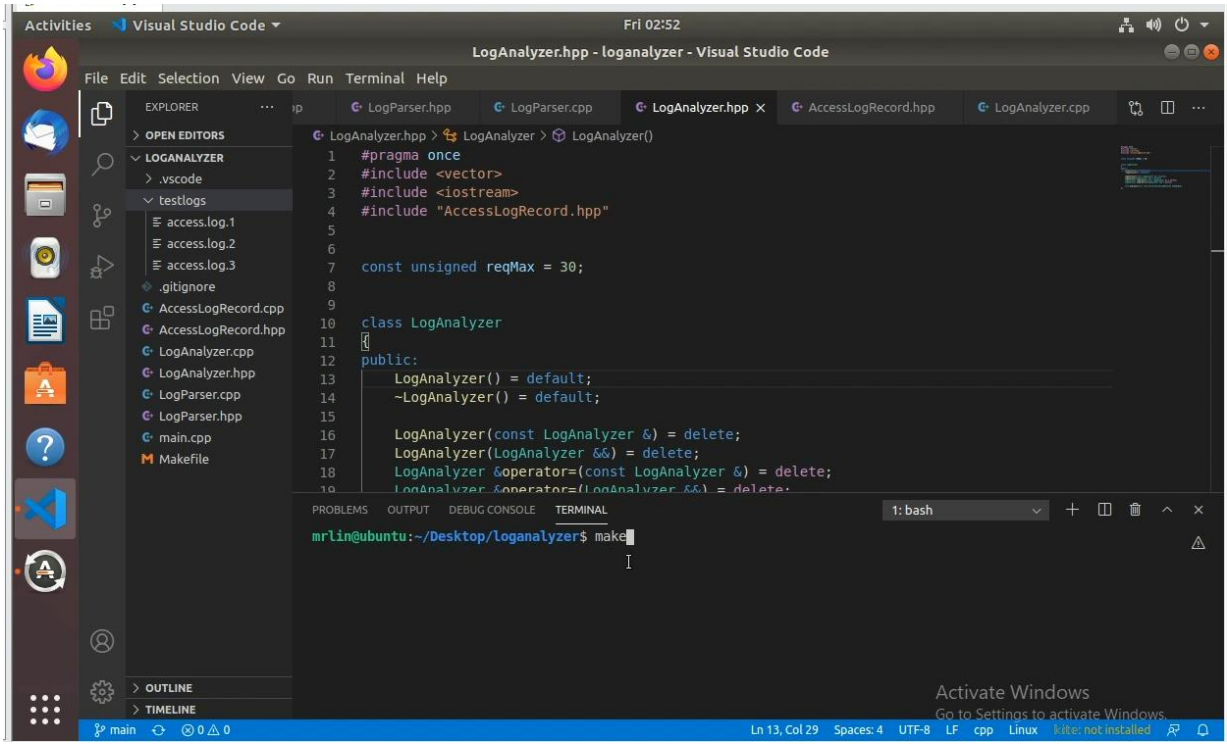


Рисунок 3.4 – Компіляція програми

Для тестування роботи програми наявні три логи в лівій частині на панелі управління, перший – до застосування сканеру вразливостей, а другий та третій – після – з певним проміжком часу між ними.

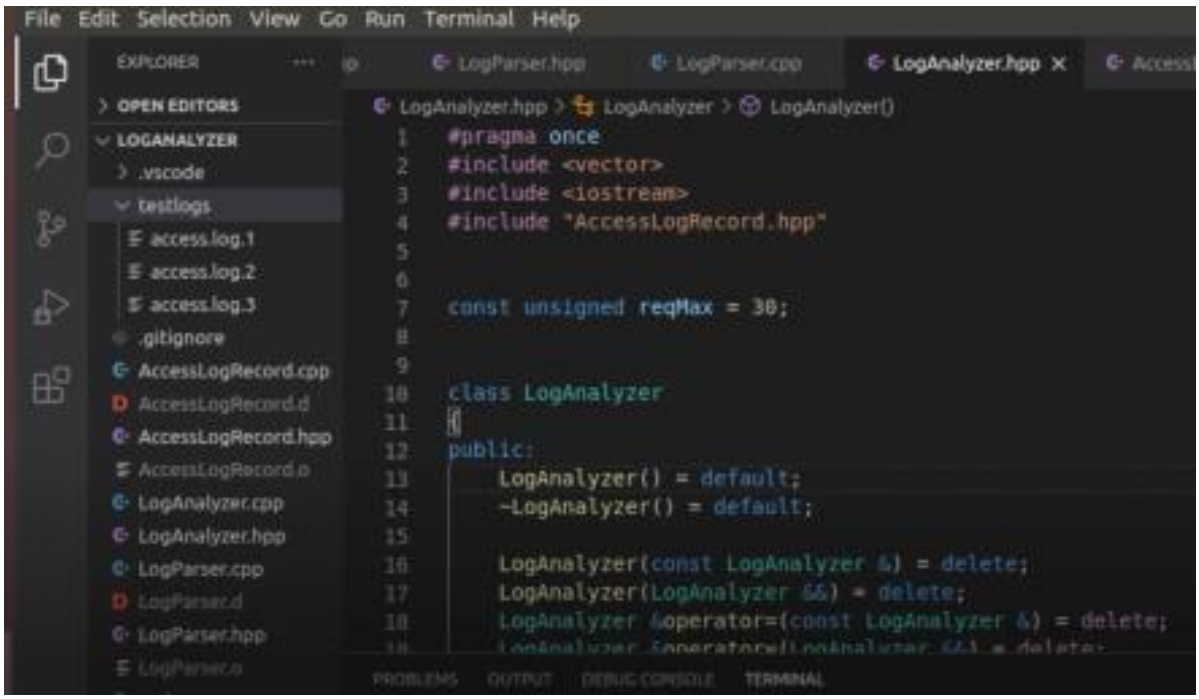


Рисунок 3. 5 – Наявні логи для перевірки працездатності

Якщо просканувати перший лог веб-сервера, програма відповідає, що підозрілої активності не виявлено. (Рис. 3.5.)

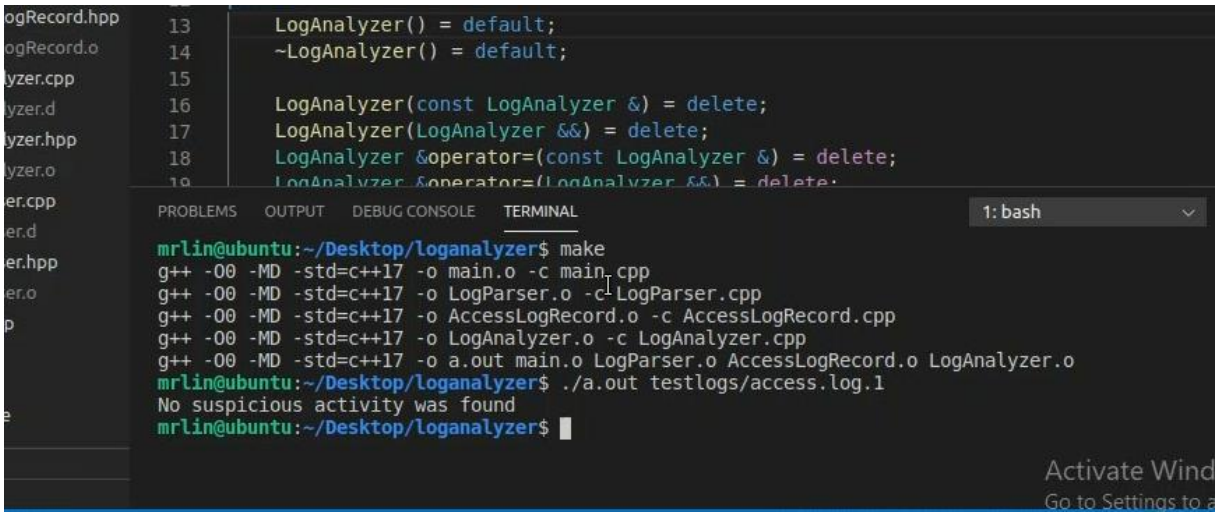


Рисунок 3. 6 – Результат перевірки першого логу.

Тепер спробуємо просканувати лог веб-сервера після застосування сканеру вразливостей. (Рис. 3.7)

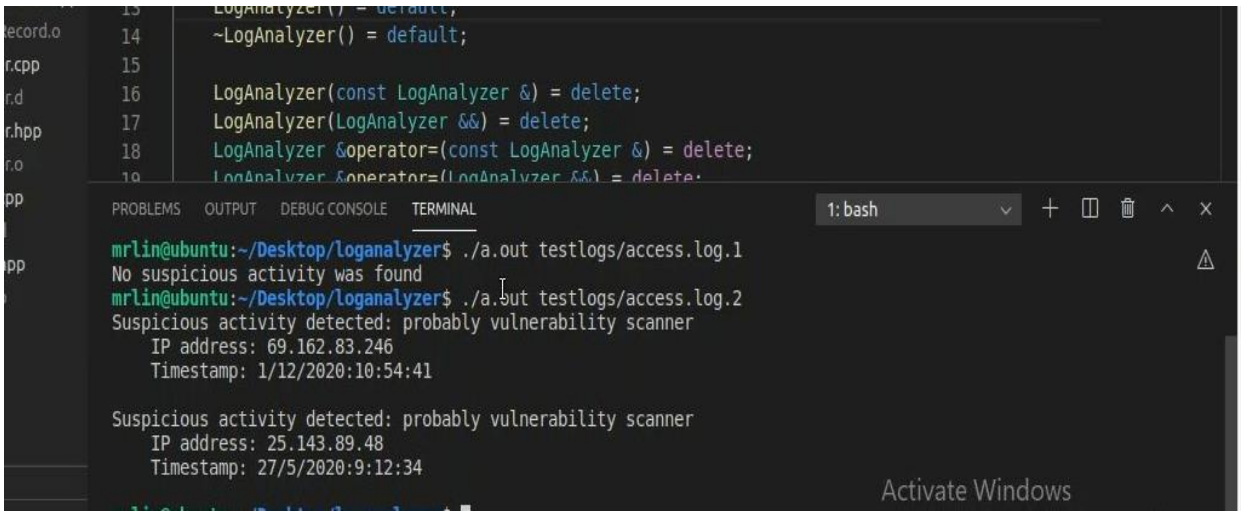


Рисунок 3.7 – Результат перевірки логу після застосування сканера вразливостей

Як ми бачимо, застосування вивело IP-адреси та час, коли ймовірно на сервер відбувалась атака.

3.8 Керування програмним кодом ІС

Управління програмним кодом виконується за допомогою системи контролю версій Git та його веб-оболонки GitHub, що дозволяє зберігати та вести історію написання програмного продукту з можливістю відкату чи перегляду попередніх змін.

Вихідний код системи розміщен у публічному репозиторії.

Тип ліцензії: MIT License, що робить це застосовання відкритим до копіювання та модифікацій.

Після завершення розробки були розраховані метрики, що демонструють кількісну складову реалізації системи.

Таблиця 3.1 – Метрики системи контролю версій Git

Метрика	Значення
Кількість комітів у master branch	36
Загальна кількість бранчів	4
Кількість закритих PR	2

3.9 Розрахунок метрик програмного коду ІС

Для розрахунку метрик коду представимо їх у відповідній таблиці (табл. 3.2).

Таблиця 3.2 – Метрика програмного коду проекту

Метрика	Одиниця вимірювання
Загальна кількість рядків коду в проекті ІС	938
Середня кількість рядків коду у класі	87
Максимальная кількість рядків коду у класі	224
Середня кількість рядків коду у методі	26
Максимальна кількість рядків у одному методі	58
Максимальна глибина дерева спадкування	1
Средня цикломатична складність	1.33
Максимальна цикломатична складність	1.5
Коментування коду, %	20
Покриття коду тестами, %	40

3.10 Розробка тест-плану

Тестування інформаційної системи це етап, що проводиться для покращення якості програмного продукту. В даній систему треба провести 2 типи тестування:

- функціональне тестування;
- тестування обсягом.

Функціональне тестування – це проходження тих функціональних вимог, що ставились перед системою. Функціональне тестування відобразить, наскільки користувацький інтерфейс коректно працює та задовольняє необхідним вимогам.

Для тестування об’ємів будуть використовуватись інструменти розробника, що доступні у браузері Google Chrome 74.0.

При тестуванні на якість тестів буде впливати версія браузера, швидкість Інтернету.

3.11 Контрольний список з якості реалізації ІС

Було створено контрольний список та відображено у відповідній таблиці (табл. 3.3).

Таблиця 3.3 – Контрольний список з якості реалізації ІС

Твердження	Відповідь	Коментарі
Використання Dependency Injection	Так	
Використання логування (logging)	Ні	
Використання модульного тестування	Ні	
Захист від SQL ін’єкцій	Так	
Захист від Javascript/XSS ін’єкцій	Ні	
Використання валідації даних у всіх полях вводу для користувацьких інтерфейсів	Так	
Використання інсталяторів/інтернет магазинів	Ні	
Використання засобів синхронізації даних у випадку багатопоточного застосування	Ні	
Підтримка глобалізації СППР (інтернаціоналізація, локалізація)	Так	
Відсутність зашитих в програмний код конфігураційних параметрів застосування	Ні	
Використання UI патернів при розробці UI	Ні	
Врахування coding style guide lines для обраного язика програмування	Ні	
Врахування рекомендованих guide lines при розробці користувацького інтерфейсу для тієї чи іншої ОС	Ні	
Підтримка локалізації та глобалізації СППР	Так	

3.12 Проведення тестування юзабіліті

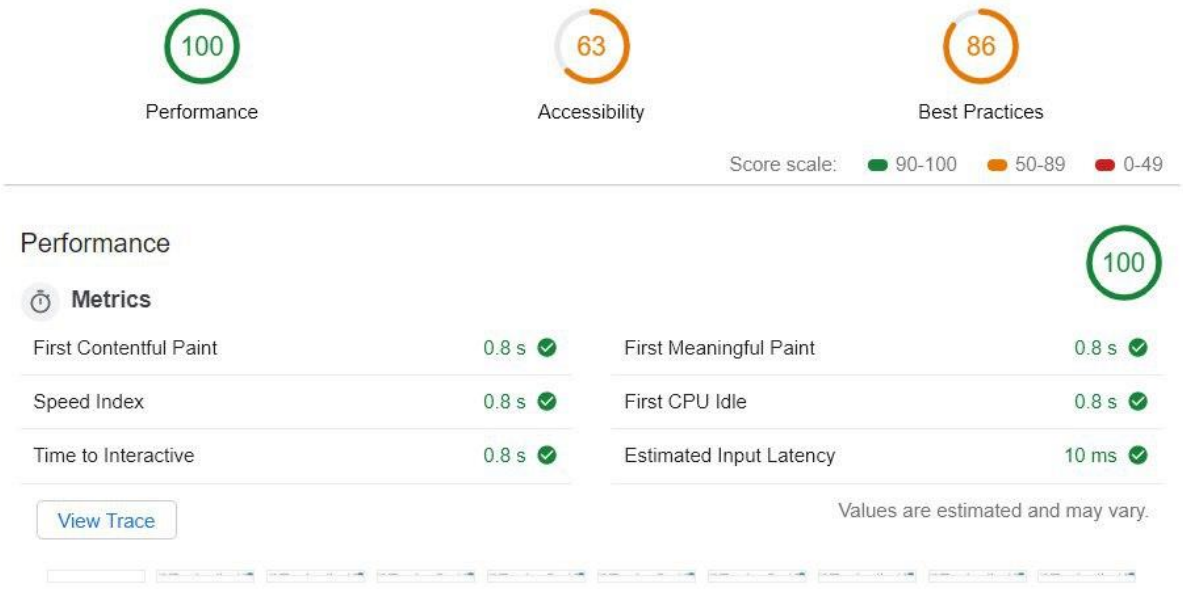


Рисунок 3.8– Результати тестування юзабіліті

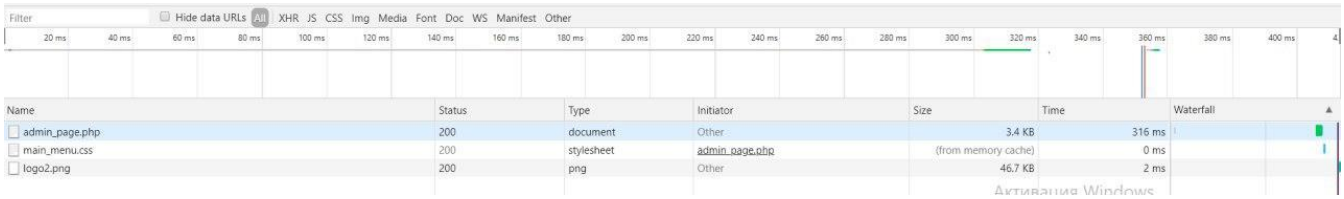


Рисунок 3.9 – Результати тестування при запуску сторінки

За результатами тестування можна зазначити, що всі показники знаходяться у нормі.

3.13. Атрибути якості

Для підтримки якості системи було виявлено основні види небезпечності з OWASP:

- Ін’єкції (Injections) - найнебезпечніша вразливість, що дозволяє зловмисникові отримати доступ до бази даних і можливість читати / змінювати / видаляти інформацію, яка для нього не призначена;
- Недоліки системи аутентифікції та зберігання сесій (Broken Authentical and Session Management);
- Міжсайтовий скриптинг (Cross Site Scripting) – помилка валідації

користувацьких даних;

- Небезпечні прямі ссилки на обэ’єкти (Insecure Direct Object References) – недостатня перевірка користувацьких даних;
- Незахищеність критичних даних (Sensitive Data Exposure) – передача даних тільки за протоколом HTTP.

Система працює відповідно атрибутам якості.

ПЕРЕЛІК ПОСИЛАНЬ

1. McDaniel A. Perl and Apache; - Москва: Астрель, АСТ, СпецЛит, 2010. 448с.

2. Ellis E. S. Caught in an Apache raid. Москва: Обновление, 2011. 346 с.

3.Kabir M. J Apache Server Administrator Handbook. Москва: Дрофа, 2009. 548с.

4. Pliny G. Earle Myths and tales from the White Mountain Apache. Москва: Книга по Требованию, 2011. 429 с.

5. Apache Server Bible., Москва: ИПБ-БИНФА, 2010. 624 с.

6. Rusel J. Apache Cassandra; Москва: Книга по Требованию, 2012. 134 с.

7. MySQL руководство администратора. Москва: Вильямс, 2005. 621 с.

8. Конверс Т. PHP 5 и MySQL. Библия для пользователя / Конверс, др. Т. и. - М.: Вильямс, 2006. - 426 с.

9. Яргер Р.Дж. MySQL и mSQL: Базы данных для небольших предприятий, а также Интернета. СПб: Символ-Плюс, 2015. 560 с.

10. Гаевский А.Ю., Романовский В.А. Самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript. Москва: Триумф, 2014. 464 с.

11. Дронов В. PHP 5/6, MySQL 5/6 и Dreamweaver CS4. Разработка интерактивных Web-сайтов. Москва: БХВ, 2009. 544 с

12. Ullman L. PHP and MySQL for Dynamic Web Sites: Visual QuickPro Guide. Москва, 2009. 799 с.

13. Diseno L.M. Web Dreamweaver. Москва, 2013. 126 с.

14. Thomson L. PHP and MySQL Web Development. Москва, 2011. 773 с.

15. Wandschneider M. Core Web Application Development with PHP and MySQL (Core). Москва, 2010. 912 с.

16. Schuchmann M. Dynamische Webseiten: Einstieg in HTML, PHP und MySQL. Москва, 2012. 124 с.

17. ДБН В.1.1-7:2016 Пожежна безпека об`єктів будівництва. Загальні вимоги.

18. Кондратьев А.И., Местечкина Н.М. Охрана труда в строительстве. Москва: Высш. шк., 1990.

19. Орлов Г.Г. Инженерные решения по охране труда в строительстве: Справочник. Москва: Стройиздат, 1985.

20. ДСТУ Б В.2.5-38:2008 Інженерне обладнання будинків і споруд. Улаштування блискавкозахисту будівель і споруд (IEC 62305:2006, NEQ)

21. ДБН В.2.5-74:2013. Водопостачання. Зовнішні мережі та споруди. Основні положення проектування.