НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп‘ютерних систем

«До захисту допущено»

Завідувач кафедри

_____ Віталій Романкевич

(підпис) (ініціали, прізвище)

“___” червня 2021 р.

Дипломний проект

на здобуття ступеня бакалавра

з напряму підготовки 6.050102 «Комп’ютерна інженерія» на тему:

«Комп'ютерна система протидії кібератакам на вебсервер»

Виконав:
студент IV курсу, групи КВ-74

Козлюк Павло Андрійович _____ _____
(прізвище, ім’я, по батькові) (підпис)

Керівник проф. каф. СПіСКС, д.т.н. проф Терейковський І. А. _ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант з нормоконтролю, доц.каф.СПСКС, к.т.н. Клятченко Я.М. _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цьому дипломному проекті немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2021 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп‘ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп’ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Віталій РОМАНКЕВИЧ
(підпис) (ініціали, прізвище)
«___» червня 2021 р.

ЗАВДАННЯ
на дипломний проект студента
Козлюк Павло Андрійович
(прізвище, ім’я, по батькові)

1. Тема проекту Комп'ютерна система протидії кібератакам на вебсервер____ ,
керівник проекту проф. каф. СПіСКС, д.т.н. проф Терейковський І. А. _____ ,
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «__» _____ 2021р. №_____

2. Термін подання студентом проекту _____

3. Вихідні дані до проекту : література по системам протидії кібератакам, література про написання WAF для вебсерверів .

4. Зміст пояснювальної записки: аналіз існуючих програмних рішень та обґрунтування теми дипломного проекту; обгрунтування вибору засобів реалізації; розроблення програмних модулів; аналіз розробленої системи та висновки.

5. Перелік графічного матеріалу (із зазначенням обов’язкових креслеників, плакатів, презентацій тощо) :
- Схема виявлення аномальних запитів.
- Схема роботи Web-сервера.
- Схема перевірки запитів.
- Схема роботи користувача з графічним.

6. Консультанти розділів проекту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М., к. т. н., доцент каф. СП і СКС.		

7. Дата видачі завдання 30 жовтня 2020р._____

Календарний план

№ з/п	Назва етапів виконання дипломного прокту	Термін виконання етапів	Примітка
1.	Видача завдання на дипломне проектування	30.10.2020	
2.	Вивчення літератури за тематикою проекту	17.12.2020	
3.	Розроблення та узгодження технічного завдання	20.01.2021	
4.	Аналіз існуючих рішень	12.02.2021	
5.	Розробка архітектури графічного інтерфейсу	28.03.2021	
6.	Тестування швидкісті обробки даних	24.04.2021	
7.	Підготовка матеріалів графічної частини проекту	03.05.2021	
8.	Оформлення документації дипломного проекту	08.05.2021	

Студент _____ Козлюк П. А.
(підпис) (ініціали, прізвище)

Керівник проекту _____ Терейковський І. А.
(підпис) (ініціали, прізвище)

* Консультантом не може бути зазначено керівника дипломного проекту.

АНОТАЦІЯ

Дипломний проєкт включає пояснювальну записку (55 сторінки, 4 додатки, 31 рисунок).

Об’єкт розробки – створення комп’ютерної системи для забезпечення безпеки вебсайту від кібератак на нього, що дозволяє уникати некоректної роботу сервісу та збереження конфіденційності даних за будь-яких умов.

В роботі проведено аналіз існуючих методів захисту вебсерверу, та враховуючи всі недоліки сформульовано вимоги до комп’ютерної системи захисту від кібератак.

Web Application Firewall розроблено у вигляді модуля Apache, який знаходиться перед перед вебсервером (попередньо перевіряє вхідні дані). WAF призначений для зупинки шкідливих запитів від відомих атак, таких як SQL Injection, XSS атаки і від невідомих атак шляхом вивчення легітимного трафіку. Дана програма реалізована мовою C.

Ключові слова:

КОМП’ЮТЕРНА СИСТЕМА , ВЕБСАЙТ, КІБЕРАТАКА, WEB APPLICATION FIREWALL, APACHE, SQL INJECTION, XSS.

ABSTRACT

Diploma project includes an explanatory note (55 pages, 4 supplements, 31 figures).

The object of development is the creation of a computer system to ensure the security of the website from cyberattacks on it, which avoids the incorrect operation of the service and maintaining the confidentiality of data under any circumstances.

The paper analyzes the existing methods of web server protection, and taking into account all the shortcomings, formulates requirements for a computer system to protect against cyberattacks.

Web Application Firewall is designed as an Apache module, which is located in front of the web server (pre-checks the input). WAF is designed to stop malicious queries from known attacks, such as SQL Injection, XSS attacks and from unknown attacks by studying legitimate traffic. This program is implemented in C.

Keywords:

COMPUTER SYSTEM, WEBSITE, CYBERATACK, WEB APPLICATION FIREWALL, APACHE, SQL INJECTION, XSS.

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	А4	ІАЛЦ.045490.002 ТЗ	Комп'ютерна система	4		
			протидії кібератакам			
			на вебсервер.			
			Технічне завдання.			
	А4	ІАЛЦ.045490.003 ТП	Комп'ютерна система	1		
			протидії кібератакам			
			на вебсервер.			
			Відомість технічного			
			проєкту.			
	А4	ІАЛЦ.045490.004 ПЗ	Комп'ютерна система	55		
			протидії кібератакам			
			на вебсервер.			
			Пояснювальна записка.			
	А4	ІАЛЦ.045490.005 Д1	Структурна схема	1		
			виявлення аномальних			
			запитів			
	А4	ІАЛЦ.045490.006 Д2	Структурна схема			
			перевірки запитів			
			з сигнатурами.			

Змін.	Арк.	№ докум.	Підпис	Дата	ІАЛЦ.045490.003 ТП		
Розробив		Козлюк П. А.			Комп'ютерна система протидії кібератакам на вебсервер **Відомість технічного проєкту**	Літ.	Аркуш 1 / Аркушів 2
Перевірив		Терейковський І.А					
Консульт.						КПІ ім. Ігоря Сікорського, ФПМ КВ-74	
Н. контроль		Клятченко Я.М.					
Зав. каф.		Романкевич В.О.					

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	А4	ІАЛЦ.045490.007 Д3	Схема	1		
			послідовності роботи			
			системи.			
	А4	ІАЛЦ.045490.008 Д4	Схема роботи	1		
			Web-сервера.			
		Диск CD-ROM	Текст пояснювальної	1		
			записки.			
			Графічний матеріал.			

Змін.	Арк.	№ докум.	Підпис	Дата	ДП.045490.003 ТП	Арк. 2

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Комп'ютерна система протидії кібератакам на вебсервер».

Галузь застосування: забезпечення безпеки вебсайту кібератак на нього шляхом застосування комп’ютерної системи.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломне проектування на здобуття першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп’ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є створення комп’ютерної системи що буде захищати вебсервер від кібератак.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- сумісність з операційними системами: Windows, Linux.
- можливість запобігти несанкіонованому доступу;

- захит даних;
- коректність роботи вебсерверу при кібератаці;
- наявність надійної системи;
- наявність зручної системи сповіщень;

5.2. Вимоги до апаратного забезпечення

- Процесор: 4-ядерний,Intel CORE i-3;
- Оперативна пам’ять: 4 Гб;
- Місце на жорсткому диску 1гб;
- Доступ до інтернету

5.3. Вимоги до програмного та апаратного забезпечення користувача

- Операційна система Windows, Linux.

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів
1.	Вивчення літератури за тематикою проекту	15.04.2021
2.	Розроблення та узгодження технічного завдання	23.04.2021
3.	Аналіз існуючих рішень	02.05.2021
4.	Підготовка матеріалів першого розділу дипломного проекту	06.05.2021
5.	Підготовка матеріалів другого розділу дипломного проекту	12.05.2021
6.	Підготовка графічної частини дипломного проекту	18.05.2021
7.	Оформлення документації дипломного проекту	20.05.2021
8.	Попередній огляд матеріалів диплому на кафедрі	26.05.2021

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

БД – база даних.

Веббраузер – програмне забезпечення для під'єднаного до мережі Інтернет пристрою, що дає можливість взаємодії з даними на гіпертекстовій веб-сторінці.

ООП – об’єктно-орієнтоване програмування.

ОС – операційна система.

ПЗ – програмне забезпечення.

ПК – персональний комп’ютер.

Плагін або плаґін (plug-in — підключати) — додаток, програмний модуль, що підключається до основної програми, призначений для розширення або використання її можливостей. Плагіни зазвичай виконуються у вигляді динамічних бібліотек.

СУБД – система управління базами даних.

Скрипт — це програма, яка автоматизує деяке завдання, яке без сценарію користувач робив би вручну, використовуючи інтерфейс програми.

Сні́фер (sniff — нюхати) — програма або програмно-апаратний пристрій, призначений для перехоплення і подальшого аналізу мережного трафіку.

Спам — масове розсилання кореспонденції рекламного чи іншого характеру людям, які не висловили бажання її одержувати. Передусім термін «спам» стосується рекламних електронних листів.

Троянський кінь — різновид шкідницького програмного забезпечення.

Черв’як — це шкідлива програма, яка може подолати всі три етапи розповсюдження самостійно.

Тег - асоційоване ключове слово, що відноситься до частини інформації. Такі метадані допомагають описати ці частини інформації і швидко знаходити їх через пошуковий запит.

Тест-кейс – це тестовий артефакт, суть якого полягає у виконанні

певної кількості дій і / або умов, необхідних для перевірки певної функціональності програмної системи що розробляється.

Фреймворк - програмна платформа, яка визначає структуру програмної системи; програмне забезпечення, що полегшує розробку і об'єднання різних компонентів великого програмного проекту.

CAPTCHA (completely automated public turing test to tell computers and humans apart) — повністю автоматизований публічний тест Тюринга, який використовується для того, щоб визначити, хто використовує систему — людина чи комп'ютер.

CLI (Command-line interface- Інтерфейс командного рядка) — текстовий інтерфейс користувача, в якому інструкції можна дати тільки введенням із клавіатури текстових рядків.

CSRF(Cross-Site Request Forgery) — тип веб-атаки, що призводить до виконання певних дій від імені користувача на веб-сторінці де останній аутентифікований.

Hooking — технологія, що дозволяє змінити стандартну поведінку тих чи інших компонентів інформаційної системи.

HTML – HyperText Markup Language (мова розмітки гіпертексту).

HTTP – HyperText Transfer Protocol (протокол передачі гіпертексту).

IDE (Integrated Development Environment) — Інтегроване середовище розробки. Комплексне програмне рішення для розробки програмного забезпечення.

JavaScript (JS) — динамічна, об'єктно-орієнтована прототипна мова програмування.

Makefile — це файл, що містить набір інструкцій використовуваних утилітою make в інструментарії автоматизації збірки.

PDF(Portable Document Format) — формат файлу, створений і підтримуваний компанією Adobe Systems, для представлення двовимірних документів у незалежному від пристрою виведення та роздільної здатності

вигляді.

Qt (cute — к'ют) — крос-платформовий інструментарій розробки програмного забезпечення мовою програмування C++.

SQL (Structured query language) — мова структурованих запитів, декларативна мова програмування для взаємодії користувача з базами даних.

URL – Uniform Resource Locator (Уніфікований локатор ресурсів) – унікальна адреса сайту.

US-CERT – підрозділ Національного управління кібербезпеки Міністерства внутрішньої безпеки США

WEB – система доступу до пов'язаних між собою документів на різних комп'ютерах, підключених до Інтернету.

Wi-Fi - технологія бездротової локальної мережі

ВСТУП

Ніхто в Інтернеті не застрахований від ризиків безпеки. В сьогоднішній гонці за створенням передових бізнес-рішень, веб-додатки розробляються і впроваджуються з мінімальним увагою до загроз.

Не дивно, що кількість корпоративних веб-сайтів, вразливих для злому, зростає швидкими темпами. Відомі сайти з таких галузей, як державне управління, фінансові послуги, роздрібна торгівля і охорона здоров'я, піддаються злому щодня. Наслідки порушення безпеки руйнівні: шкода авторитету, втрата доходів, юридична відповідальність, а також втрата лояльності клієнтів [1].

За деякими оцінками, щодня зламується від 30 000 до 50 000 вебсайтів. Ці цифри зростають з кожним днем, і важливість безпеки вебсайту стрімко збільшується. Безпека в сучасному світі стає все більш важливою з кожним днем, і зараз вкрай важливо захистити свій сайт і конфіденційність інформації.

А під час пандемії COVID-19 актуальність питання кібербезпеки тільки зросла, так як і значно збільшився онлайн користувачів.

Інша причина, по якій безпека вебсерверів повинна викликати занепокоєння, - це постійно зростаюче число вразливостей, що потрапляють в виробничий код.

2020 роки став четвертим рекордним роком за кількістю вразливостей, зареєстрованих в базі даних вразливостей US-CERT.

У цьому дипломному проєкті буде розглянуто найактуальніші проблеми в захисті вебсерверів та розроблено власну комп’ютерну систему протидії кібератакам.

1. АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ

1.1. Огляд функціонування вебсерверу та дослідження області кібербезпеки

Вебсервер - сервер, що приймає HTTP-запити від клієнтів, зазвичай веб-браузерів, і видає їм HTTP-відповіді (Рис 1.1), як правило, разом з HTML-сторінкою, зображенням, файлом, медіа-потоком або іншими даними [2].

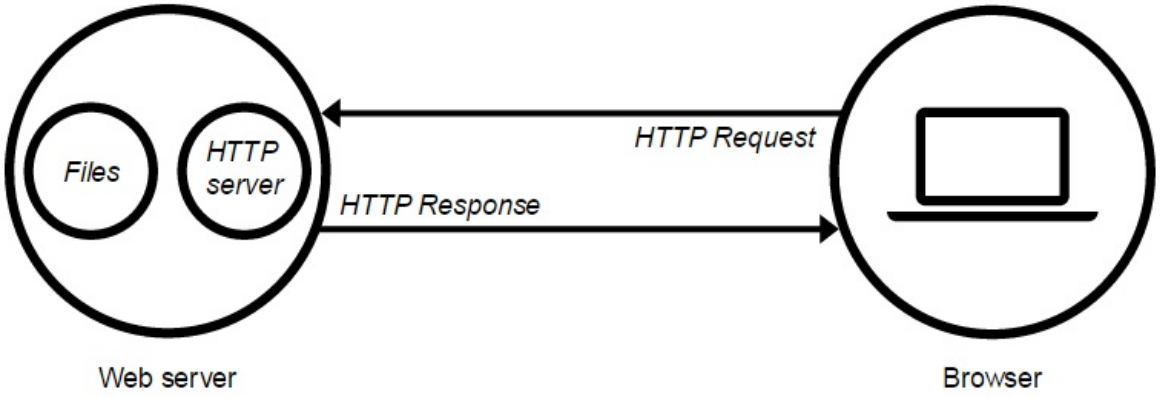


Рисунок 1.1 – Взаємодія браузера та вебсервера

З точки зору програмного забезпечення, вебсервер включає деяку кількість компонентів, відповідальних за доступ користувача до файлів на сервері (наприклад, сервера HTTP). Сервер HTTP - це програмне забезпечення, яке може зрозуміти веб-адреси (URL) та HTTP. Сервер зберігає веб-сторінки та надає їх клієнтам на основі запитів, оброблених через HTTP, який у всій архітектурі веб-сторінки відіграють важливу роль як протокол для передачі інформації від клієнта на сервер (і навпаки) [3].

Природно, що на мережу легко нападають хакери, які атакують вебсервери різними способами. Ось чому будь-які уразливості програм, баз даних, операційних систем або мереж призведуть до атак на вебсервер. Атаки на WEB-сервер означають порушення нормальної функції вузла, видалення або зміна його вмісту або отримання привілейованого доступу до комп'ютера.

З огляду на різноманітні технології, що використовуються в компоненті вебсервера, необхідно проаналізувати існуючі мережеві атаки на вебсервер та способи їх захисту.

Кіберпростір - це область, що виникла в результаті об'єднання комп'ютерів і телекомунікаційних мереж по всьому світу. Вона використовується для зберігання, зміни і обміну даними через мережеві і пов'язані з ними фізичні структури в глобальному масштабі, незалежно від фізичної географії [4].

Основною проблемою кіберпростору є безпека. Кіберпростір в значній мірі залежить від інформатики і телекомунікацій практично для кожної послуги та діяльності. Тому постійні погрози для життя громадян, їх діяльності і державних систем являють собою величезну проблему для ефективного функціонування кіберпростору [5].

Наприклад, споживачі цифрових технологій все більше залежать від Інтернету в плані комунікації: люди використовують електронну пошту, стільниковий телефон і текстові повідомлення, щоб залишатися на зв'язку з друзями і сім'єю по всьому світу. Але через постійні погрози, таких як злом і крадіжку даних, ця свобода спілкування обмежується.

І це не тільки спілкування. Є ще кілька аспектів нашого життя, які все більше залежать від Інтернету, включаючи фінанси (банківські рахунки, кредити, електронні чеки), освіту (візуальні класи, дослідження, онлайн-табелі), медицину (медичні записи, медичне обладнання) і державне управління (соціальне забезпечення, записи про народження / смерті, податкові записи). Оскільки всі ці види діяльності залежать від безпечного кіберпростору, до будь-яких погроз його безпеки слід ставитися дуже серйозно.

Найпростіший спосіб поглянути на це - уявити, скільки інформації ви зберігаєте в своєму комп'ютері або в чужій системі. Якщо ви недавно брали кредит у банку, то ця інформація напевно зберігається в комп'ютерній

системі. Якщо ви недавно проходили лікування в лікарні, то ж саме відноситься і до вас. Там, де ви працюєте або вчитеся, ваша особиста інформація також буде зберігатися на комп'ютері в цілях ідентифікації. Кіберзлочинці становлять загрозу для всієї цієї інформації. При будь-якій атаці злочинці можуть вкрасти всі ці дані і робити з ними все, що їм заманеться [6].

Це підводить нас до питання про важливість кібербезпеки, яка полягає в забезпеченні конфіденційності, цілісності та доступності інформації в Інтернеті. Вона визначається як "всі дії і операції, спрямовані на зниження і запобігання кіберзагроз та вразливостей". І це те, що повинно бути головним пріоритетом для кожної людини, що живе в сучасному технологічно залежному суспільстві [7].

1.2. Основні види кібератак

Головне правило при створенні мережевих ресурсів - ніколи не довіряйте даним, введеним користувачем.

В даний час існують такі атаки (Рис 1.2): mailbombing, переповнення буфера, використання шкідливого ПЗ (вірусів, сніфферів, троянських коней, черв'яків і т.д.), мережева розвідка, IP-спуфинг, man-in-the-middle, ін'єкція (SQL-ін'єкція, PHP-ін'єкція, CSRF-атаки, міжсайтовий скриптинг, XPath ін'єкція), - і DDoS- атаки, phishing-атаки, брутфорс-атаки [7].

Рисунок 1.2 – Види кібератак на вебсервер

Mailbombing.

Вважається найстарішою методом атак, хоча суть його проста і примітивна: велика кількість поштових повідомлень унеможливлюють роботу з поштовими скриньками, а іноді і з цілими поштовими серверами. Для цієї мети було розроблено безліч програм, і навіть недосвідчений користувач міг зробити атаку, вказавши лише e-mail жертви, текст повідомлення, і кількість необхідних повідомлень. Цю атаку складно запобігти, так як навіть поштові фільтри провайдерів не можуть визначити реального відправника спаму. Провайдер може обмежити кількість листів від одного відправника, але адреса відправника і тема часто генеруються випадковим чином. Зазвичай до таких атак досвідчені зловмисники вдаються вкрай рідко [8].

Шкідливе ПЗ

Шкідливе ПЗ- це тип програми, що може виконувати різні шкідливі завдання. Деякі види шкідливого ПЗ призначені для створення постійного доступу до мережі, деякі - для шпигунства за користувачем з метою отримання облікових даних або інших цінних відомостей, а деякі - просто для порушення роботи. Деякі види шкідливих програм призначені для вимагання у жертви. Мабуть, найвідомішою формою шкідливого ПЗ є програма-вимагач - вона призначена для шифрування файлів жертви і подальшого вимагання викупу для отримання ключа дешифрування [8].

SQL ін'єкція - один із найпоширеніших способів вторгнення у веб-ресурси (програми, що використовують дані) (Рис 1.3). Його мета - ввести сторонній, чужий код SQL у запит.

Впровадження SQL запитів, в залежності від виду використовуваної СУБД і умов, дає можливість хакерам виконати сторонній запит до бази даних (наприклад, видалити, прочитати, додати або змінити дані з будь-яких таблиць бази), отримати можливість працювати з локальними файлами і виконання своїх команд на сервері, що атакується. Здебільшого, такий вид

нападу можна бачити на прикладі сайтів, де використовуються параметри командного рядка (змінні URL), щоб побудувати SQL-запити до баз даних, без відповідної перевірки [7].

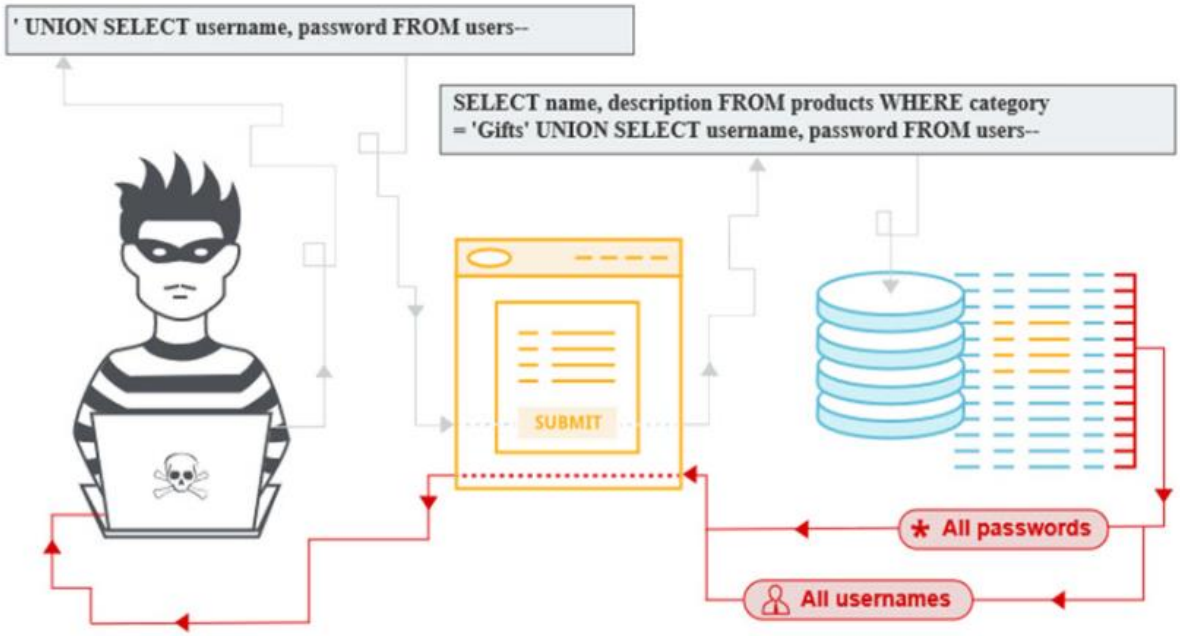


Рисунок 1.3 – Схематичне зображення роботи SQL ін'єкції

Щоб уникнути цієї ситуації, потрібно відфільтрувати лапки та інші спеціальні символи, що може порушити логіку вашого запиту. Також, коли у вас є число, обов'язково явно приводити його до числового типу. Деякі фахівці радять застосовувати для цього спеціальні конструктори SQL-запитів, які самі забезпечують необхідний поділ запиту і даних [9].

Ін'єкція PHP - це також спосіб зламати веб-сайт, написаний на PHP. Основною ідеєю є реалізація розробленого сценарію в коді на стороні сервера ресурсів, що призводить до виконання сторонніх команд. Відомо - в багатьох розповсюджених движках та на форумах, які працюють на PHP (найчастіше це застарілі версії), є не зовсім продумані модулі та конструкції з уразливими місцями. Хакери шукають ці вразливості, аналізують їх і використовують [9].

XPath-ін'єкція - вид вразливостей, який полягає у впровадженні XPath-виразів в оригінальний запит до бази даних XML. XML (eXtensible Markup Language) - це мова розмітки, за допомогою якої створюються XML документи.

CSRF - вид атак на відвідувачів веб-сайтів, який використовує недоліки протоколу HTTP. Міжсайтова підробка запитів (CSRF) - це вразливість веб-безпеки, яка дозволяє зловмисникові спонукати користувача виконати дії, які він не збирався виконувати. Вона дозволяє зловмисникові частково обійти політику однакового походження, яка розроблена для того, щоб різні веб-сайти не заважали один одному. При успішній атаці CSRF зловмисник змушує користувача-жертву ненавмисно виконати будь-яку дію. Наприклад, це може бути зміна адреси електронної пошти в обліковому записі, зміна пароля або переказ коштів. Залежно від характеру дії зловмисник може отримати повний контроль над обліковим записом користувача. Якщо зламаний користувач має привілейовану роль в додатку, то зловмисник може отримати повний контроль над усіма даними і функціями програми. Більшість сайтів, які використовують стандартну архітектуру, уразливі «за замовчуванням» до цих атак [10].

DOS.

Відокремлено в ряду загроз безпеки стоять вразливості DOS - Denial Of Service (відмова в обслуговуванні) (Рис 1.4). Як правило, до цього класу нападів належать події, описувані в новинах "Хакери атакували сайт X, порушивши його роботу. Сайт не працював протягом Y годин". Тобто це саме "атака", а не "злом". На сервер виробляються запити, які він не може (не встигає) обробити, в результаті чого він не встигає обробити запити звичайних відвідувачів і виглядає для них як непрацюючий.

Ці атаки не мають на меті викрасти дані з бази, але можуть допомогти почати інші види атак, тобто звільнити шлях. Наприклад, деякі програми через помилки в своєму коді можуть викликати виняткові ситуації, і при

відключенні сервісів здатні виконувати код, наданий зловмисником. Або атаки лавинного типу, коли сервер не може обробити величезну кількість вхідних пакетів [10].

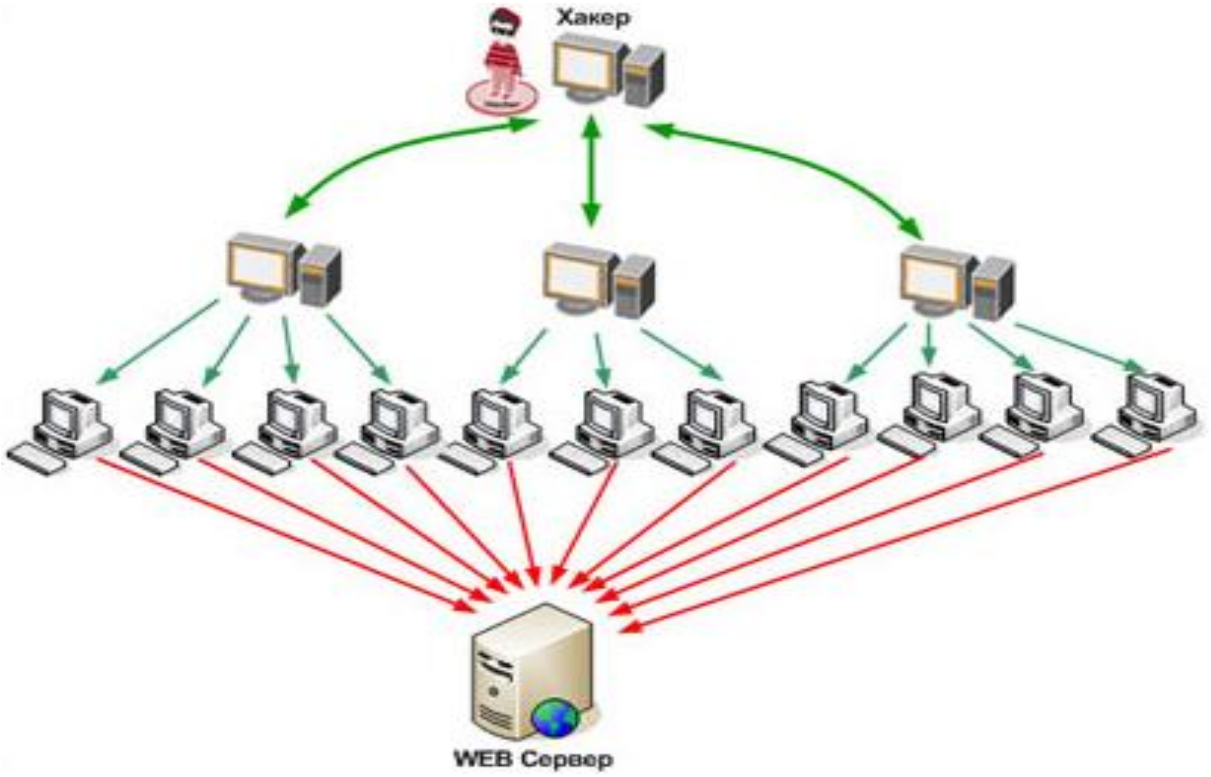


Рисунок 1.4 – Схематичне зображення DDOS атаки

Проти DOS атаки неможливо захиститися на 100%, але можна зробити обмеження на кількість спроб логіна з однієї IP-адреси в деяку кількість часу. Наприклад - не більше 5 в 10 хвилин. При вичерпанні показувати повідомлення "почекайте" або пропонувати ввести CAPTCHA. Деякі системи просять ввести CAPTCHA взагалі при кожній спробі логіна [11].

Phishing атаки.

Фішингова атака - це практика надсилання електронних листів, які видаються листами з надійних джерел з метою отримати особисту інформацію або вплинути на користувачів, щоб вони щось зробили. Вона поєднує в собі соціальну інженерію і технічні прийоми. Це може бути

вкладення в лист, яке завантажує шкідливе ПЗ на ваш комп'ютер. Це також може бути посилання на незаконний веб-сайт, який може обманом змусити вас завантажити шкідливі програми та передати свої особисті дані. Spear phishing - це дуже цілеспрямований вид фішингу. Зловмисники витрачають час на вивчення цілей і створюють повідомлення, які є персональними і актуальними. Тому Spear phishing дуже важко розпізнати і ще важче захиститися від нього [12]. Один з найпростіших способів, за допомогою якого хакер може провести атаку spear phishing, - це підміна електронної пошти, коли інформація в розділі "Від" електронного листа підробляється, створюючи враження, що лист приходить від знайомого вам людини, наприклад, від вашого керівництва або компанії-партнера. Ще один прийом, який шахраї використовують для додання достовірності своїй історії, - це клонування веб-сайтів: вони копіюють законні веб-сайти, щоб обдурити вас і змусити ввести персональну інформацію, або облікові дані для входу в систему [13].

XSS-атака.

Атака міжсайтового скриптингу (XSS) відбувається, коли на сайті є вразливість, що дозволяє впроваджувати скрипти (Рис 1.5). Зловмисники використовують такі уразливості і впроваджують шкідливі JS скрипти в базу даних сайту. Коли користувач згодом запитує ці дані, веб-браузер користувача виконує шкідливий скрипт JS. Це дозволить зловмисникові вкрасти файли cookie браузера для перехоплення сесії. Потім хакери можуть використовувати інформацію про сесії для використання додаткових вразливостей, можливо, отримати мережеву інформацію та контролювати комп'ютер користувача. Це особливо важливо для корпоративного середовища, оскільки одна XSS-атака може скомпрометувати всю мережу [14].

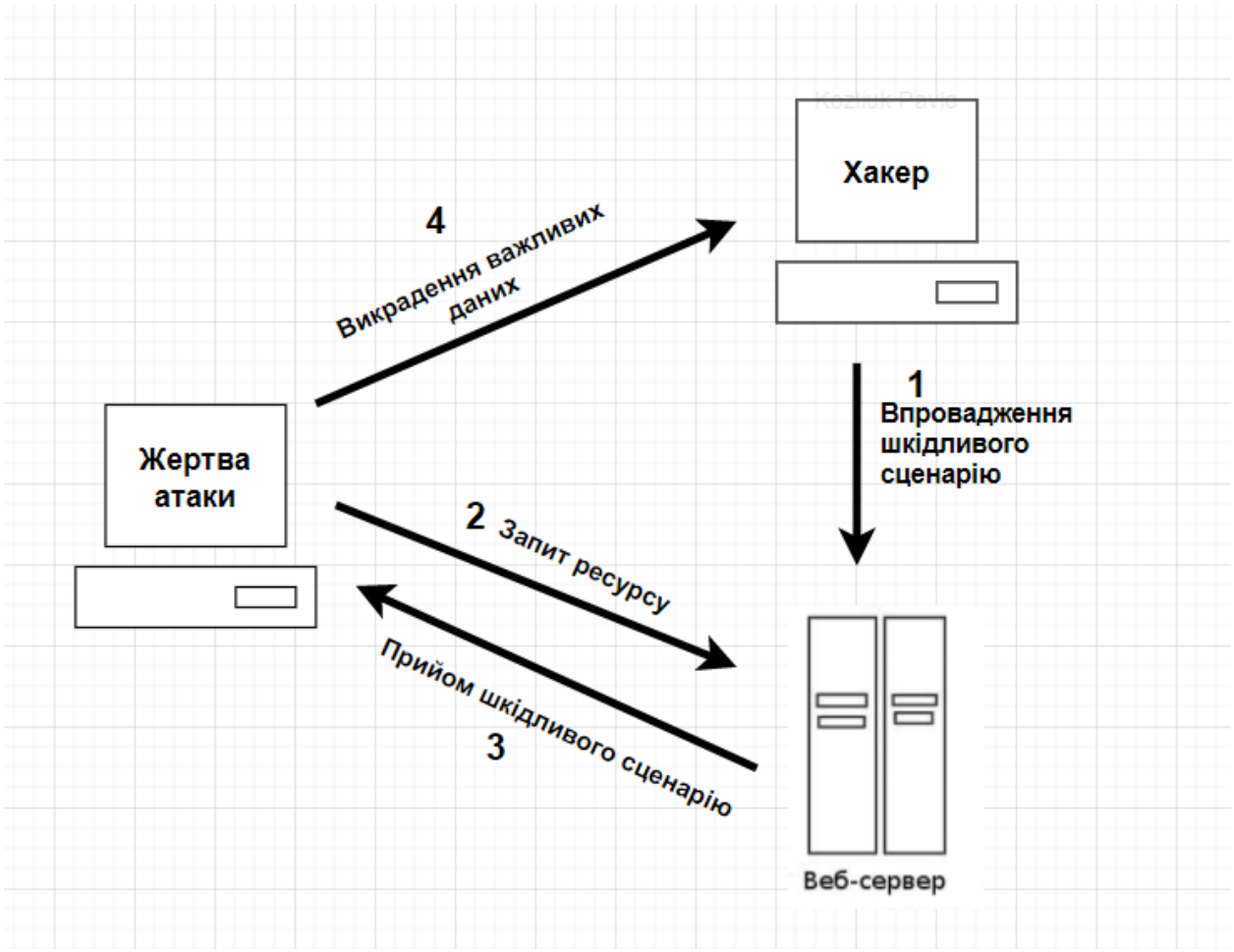


Рисунок 1.5 – Схематичне зображення XSS атаки

Щоб не стати жертвою атаки XSS слід дотримуватися наступних правил безпеки:

Головне правило для розробників - використання будь-якого фільтра.

Фільтрувати всі вкладені конструкції.

Шифрування. При створенні фільтра слід обов'язково враховувати ризик кодування атак. Є маса програм-кодувальників, за допомогою яких можна зашифрувати будь-яку атаку так, що не один фільтр «не побачить» її.

Застосування тегів. Є одне вразливе місце, пов'язане з тегами url, bb, img, мають безліч параметрів, включаючи lowsrc і dynsrc, що містять javacsript. Дані теги слід фільтрувати [15].

Брутфорс.

Атака методом перебору, іноді звана атакою по паролю, є однією з найпростіших форм веб-атак. Хакер просто багаторазово пробує різні комбінації імен користувачів і паролів, поки не увійде до облікового запису користувача. Звичайно, одного комп'ютера потрібні були б роки, щоб перебрати всі комбінації. Але коли хакери отримують контроль над декількома комп'ютерами або розробляють потужний програмний обчислювальний механізм, все може стати дуже просто.

Брутфорс - один з найбільш популярних методів злому паролів облікових записів онлайн-банків, платіжних системах або на веб-сайтах. Але з ростом довжини пароля цей метод стає незручним через тривалість витраченого часу на перебір всіх можливих варіантів [16].

1.3. Огляд видів захисту вебсерверу та аналогів протидії кібератакам

Захист від злому серверів — це комплекс заходів, в тому числі постійний моніторинг роботи сервера і вдосконалення його захисту. Захистити сервер назавжди практично нереально тому, що постійно з’являються нові можливості обходу антивірусних систем.

1.3.1. Загальні рекомендації

В Інтернеті є багато інформації про різні типи хакерських атак на веб-сайтах, але давайте підсумуємо, пояснимо та визначимо методи боротьби з ними та припинення їх дії.

Що повинен робити користувач:

- Регулярно оновлювати операційну систему.
- Потрібно постійно стежити за відповідними списками розсилки і веб-сайтами на предмет оголошень, пов'язаних з безпекою.
- Налаштувати операційну систему відповідно до передових методів роботи з системою.

Це включає, але не обмежується наступним:

Включити необхідні служби і додатки; відключити всі інші.

Створити облікові записи користувачів відповідно до принципу найменших привілеїв.

Встановити паролі для всіх облікових записів відповідно до правил паролів.

Видалити або вимкнути непотрібні облікові записи за замовчуванням.

Змінити всі паролі за замовчуванням, встановлені прикладним програмним забезпеченням [17].

• Налаштувати вебсервер відповідно до рекомендованих виробником передовими методами. Вебсервери повинні бути налаштовані таким чином, щоб заборонити доступ до файлів, які не призначені для публічного використання.

• Відокремити вміст вебсервера і відповідні підкаталоги від каталогів операційної системи і додатків.

• Регулярно робити резервні копії веб-вмісту і час від часу резервне копіювання конфігурації операційної системи і додатків.

• Використовуйте технології веб-аутентифікації і шифрування, такі як SSL / TLS, в залежності від характеру даних вебсервера.

• Встановити методологію внутрішнього контролю змін.

1.3.2. Додатки і фреймворки

GlassWire - монітор мережевої безпеки. Його головна особливість - відображення активності на графіку (Рис 1.6). Це допомагає швидко визначати незвичайна поведінка програм і вживати заходів щодо їх блокування. Якщо нова служба або додаток підключається до мережі, ви отримуєте повідомлення. Це дозволяє виключити можливість несанкціонованого доступу. Якщо ви ігноруєте попередження, з'єднання буде додано до списку довірених [18]. В налаштуваннях можна виставити більш

сувору політику дозволів, тоді будь-яке нове з'єднання буде блокуватися. Потім ви самі вирішуєте, чи варто додавати його в список дозволених.

Переваги додатку:

- Зручний для сприйняття графік мережевої активності, на якому кидаються в очі підозрілі сплески.
- Оповіщення про всі незвичайні зміни в роботі системи.
- Контроль інших комп'ютерів всередині мережі та отримання повідомлень про підключення до Wi-Fi нових пристроїв.

мінуси

- Велика частина корисних можливостей доступна тільки після оплати ліцензії.
- Підписка на повну функціональність брандмауера коштує від 39 доларів в рік на один пристрій.

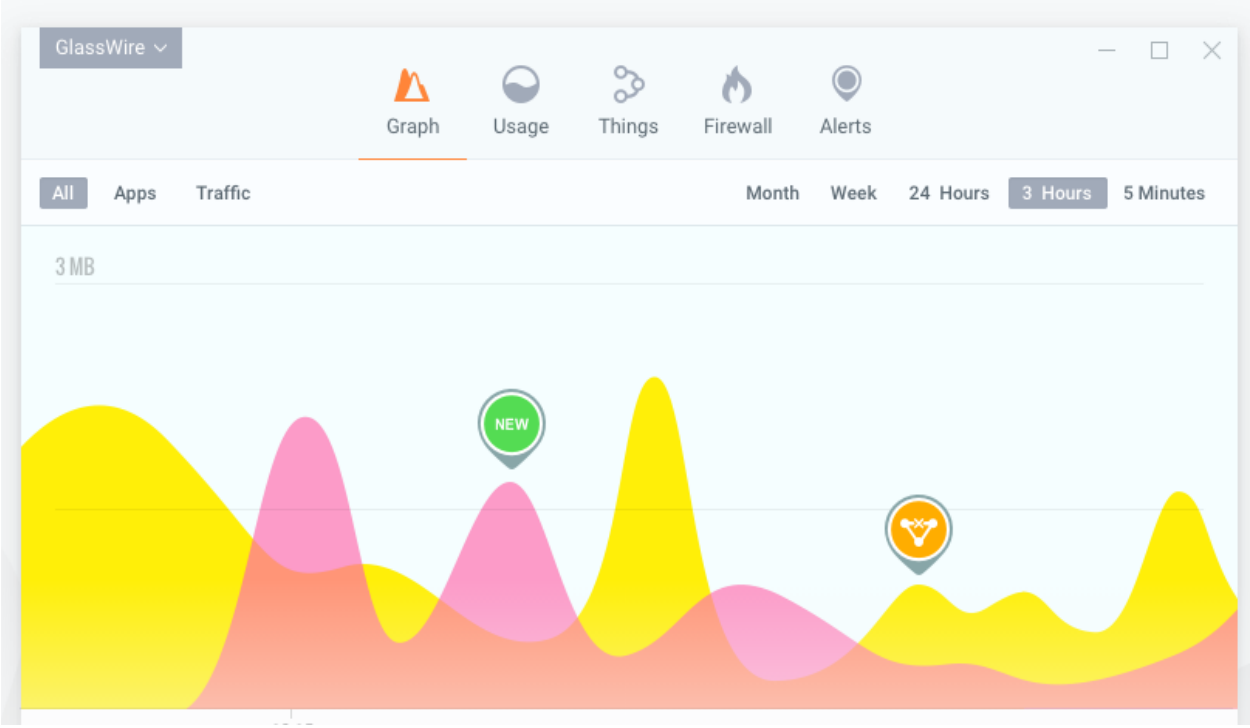


Рисунок 1.6 – Графік активності служб і програм

AVG Internet Security Файрвол від AVG входить до складу антивірусного програмного забезпечення AVG Internet Security (Рис 1.7). Він доступний в повній версії пакету. При установці безкоштовного антивіруса брандмауера немає - тільки інструменти сканування ПК і захисту від небезпечних посилань і вкладень. Брандмауер пропонує стандартний набір функцій, що дозволяє захиститися від несанкціонованих підключень до комп'ютера і обмежити доступ додатків в інтернет.

Захист від зловмисників в AVG Internet Security

Переваги:

- Захист від вірусів, здирників, шпигунів, шкідливих програм.
- Блокування небезпечних посилань і вкладень.
- Автоматичне оновлення.
- Блокування спроб стеження через веб-камеру.
- Захист від несанкціонованих підключень.
- Тонкі налаштування дозволів для додатків.
- Захист від фішингових сайтів.

Недоліки:

- За деякі функції антивірус просить гроші навіть після установки повного пакета.
- Нав'язлива реклама платних поліпшень.
- безкоштовна пробна версія тільки на 30 днів.

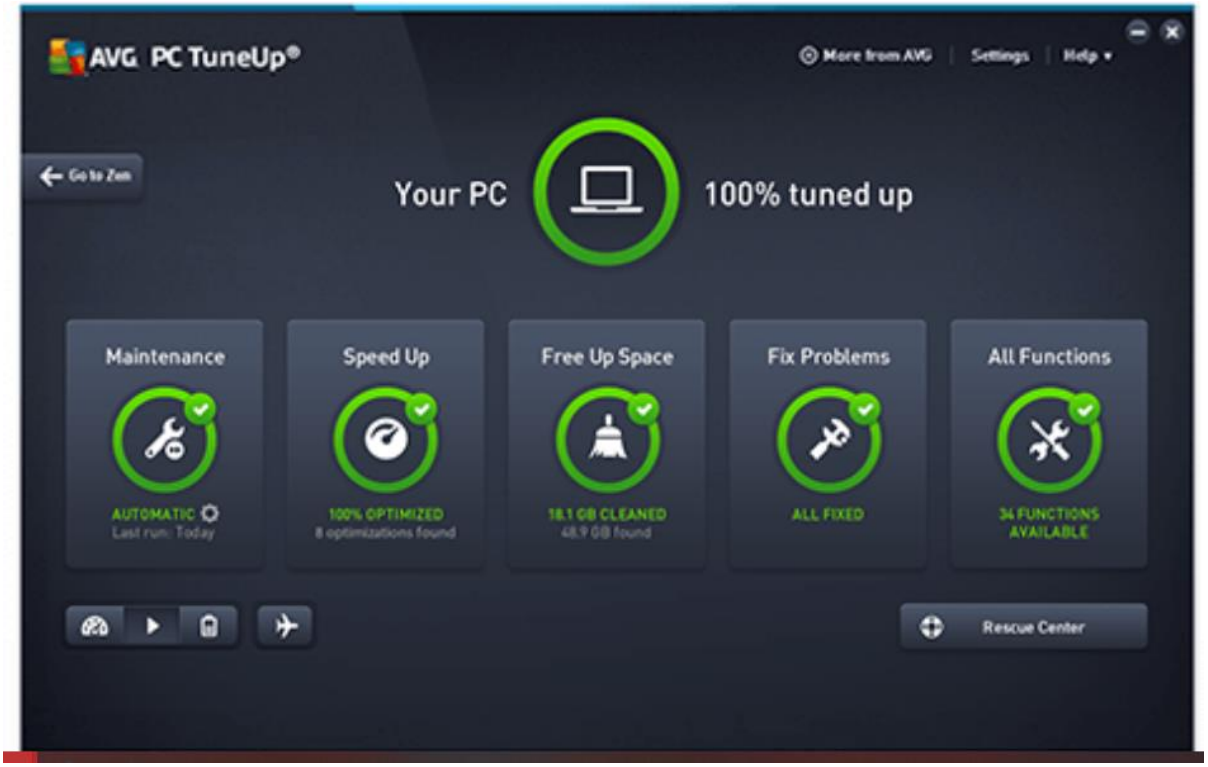


Рисунок 1.7 – AVG Internet Security

UpGuard Web Scan - це зовнішній інструмент оцінки ризиків, який використовує загальнодоступну інформацію для оцінки (Рис 1.8).

Рисунок 1.8 – Зображення робочого середовища UpGuard Web Scan

1.4. Висновки до розділу та обгрунтування теми дипломного проєкту

Проаналізовано найпоширеніші веб-атаки, та досліджено вразливості, які можуть бути застосовані для проведення даних атак. Розглянуто веб-атаки та слабкі місця, що можуть використовуватись для виконання цих атак. Також, було підкреслено на тому, що слід бути досить делікатним до деталей вебсерверу, оскільки кожна маленька вразливість може привести до великих збитків. Визначено, що навіть одна не вірна команда, що використовується в коді певного вебсерверу може звести всю безпеку ресурсу нанівець, тому слід бути обережним.

Абсолютно безпечних систем не існує, можна лише знизити ймовірність взлому. Також, очевидно, що сфера кіберзахисту вебсерверів є дуже широкою і постійно розробляються нові продукти для захисту від атак.

В результаті дипломного проєкту буде розроблено та впроваджено Web Application Firewall у вигляді модуля Apache, що буде знаходитись перед вебсервером та перевірятиме вхідні дані. WAF призначений для уникнення проходу шкідливих запитів від SQL ін’єкцій, XSS атак та інших атак вивчаючи вхідний трафік. За статистикою 84% Інтернет-ресурсів добре захищені від даних атак. Інші 16% не в змозі ефективно протистояти їм. Усунення цього грубого недоліку вимагає від власників сайтів додаткових капіталовкладень в безпеку, на що більшість з них не готові [19]. Тому буде розроблено безкоштовний продукт що ефективно вирішуватиме проблему SQL ін’єкцій та XSS атак.

2. ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1. Вибір мови програмування

Враховуючи результати аналізу в першому розділі дипломного проєкту та аналізу існуючих рішень, було вирішено розробляти WAF (Web Application Firewall) у вигляді модуля Apache. WAF розроблена для протидії шкідливим запитам SQL ін'єкцій та XSS атак.

До мови програмування для розроблення серверної частини було висунуто наступні вимоги:

- Простота використання
- Гнучкість
- Широкий функціонал
- Мова що не навантажуватиме систему
- Прості математичні обчислення виконуються максимально швидко

Мова C

C - одна з найстаріших мов, яка не може бути замінена ніякою іншою мовою, оскільки саме C є ядром кожної машини. У всього є свої сильні сторони і обмеження, які роблять його унікальним [20].

Переваги мови програмування C:

- Мова C є потужною мовою, оскільки вона містить безліч типів даних і операторів, що надає велику платформу для виконання всіх видів операцій.
- Дуже гнучка, або, можна сказати, машинно-незалежна, що допомагає виконувати код на будь-якій машині без внесення будь-яких змін або всього лише невеликого корегування коду.
- Містить безліч вбудованих функцій, які допомагають при побудові програми.
- Здатність до саморозширення. Можна додавати свої власні

функції в стандартну бібліотеку С і спрощувати код.

- Структурована мова програмування. Це означає, що питання або складні проблеми діляться на більш дрібні компоненти чи процедури. Така модульна структура допомагає легше і простіше проводити тестування і підтримку продукту.
- Це мова програмування середнього рівня, тобто вона підтримує як високорівневе, так і низькорівневе програмування.
- Використання алгоритмів і структур даних в мові С зробило обчислення в програмах дуже швидкими і плавними. Таким чином, мова С можна використовувати в складних обчисленнях і операціях.
- Динамічний розподіл пам'яті.
- Системне програмування.

Недоліки мови програмування С:

- Концепція ООП. С не підтримує концепцію ООП (успадкування, поліморфізм, інкапсуляція, абстракція, приховування даних). С просто має процедурний підхід до програмування.
- У мові програмування С помилки або недоліки не виявляються
- після кожного рядка коду. Замість цього компілятор показує всі помилки після написання програми. Це робить перевірку коду дуже складною у великих програмах.
- У мові С не реалізована концепція просторів імен.
- Відсутність обробки винятків. Під час компіляції коду можуть виникати різні аномалії і помилки.
- У мові С немає ні конструктора, ні деструктора.
- Низький рівень абстракції.

Python

Python - це проста, універсальна і повноцінна мова програмування.

Переваги Python:

- Підвищена продуктивність
- Мова Python - інтерпретована мова, що означає, що Python безпосередньо виконує код рядок за рядком. У разі будь-якої помилки вона зупиняє подальше виконання та повідомляє про виниклу помилку.
- Він автоматично привласнює тип даних під час виконання. Програмісту не потрібно турбуватися про оголошення змінних і їх типів даних.
- Стандартна бібліотека Python величезна, ви можете знайти практично всі функції, необхідні для більшості завдань [21].

Недоліки Python:

- Повільна швидкість. Послідовне виконання коду часто призводить до повільного виконання.
- Мова програмування Python використовує великий обсяг пам'яті.
- Програмувати на Python досить легко, але коли якщо взаємодіяти з базою даних, він відстає в продуктивності.
- Помилки під час виконання. Python - динамічно типізована мова, тому тип даних змінної може змінитися в будь-який момент. Змінна, яка містить ціле число, може стати рядком і це призведе до помилок [21].

PHP

Назва PHP розшифровується як Hypertext Preprocessor і є мовою сценаріїв на стороні сервера, це означає що написані на ньому програми працюють на вебсерверах і не залежать від інтернет-браузера.

Синтаксис мови PHP аналогічний мови C. PHP широко використовується для розробки веб-додатків і став одним з основних мов для розробників нових додатків. Однак з роками область його застосування змінилася, і сьогодні мова кодування PHP вважається одним з найпростіших і популярних інструментів програмування для веб-розробки завдяки своїм численним перевагам, які і є основною метою даного тексту. PHP широко використовується для розробки веб-додатків і інших додатків у всіх областях.

Нижче перераховані деякі технології, на основі яких розробляються PHP-додатки:

— Системи управління контентом.

— Веб-додатки та розробка сайтів.

— Веб-сайти та додатки для електронної комерції.

— Аналітика та подання даних.

— Обробка зображень.

— Додатки на основі графічного дизайну інтерфейсу [8].

Переваги PHP:

- З відкритим вихідним кодом. Розробляється і підтримується великою групою розробників. Це допомагає створити спільноту підтримки і безліч бібліотек розширень.
- Швидкість. PHP відносно швидкий, тому що не використовує багато системних ресурсів.
- Простота у використанні: Він використовує синтаксис, схожий на C, тому тим, хто знайомий з C, буде дуже легко освоїти його і легко створювати скрипти для веб-сайтів.
- Потужна бібліотечна підтримка: Можна легко знайти необхідні функціональні модулі, такі як PDF, graph і т.д.
- Вбудовані модулі підключення до баз даних: Можна легко

підключатися до баз даних за допомогою PHP [22].

Недоліки PHP:

- Не підходить для великих додатків: Буде важко використовувати його для програмування масштабних додатків. Оскільки мова програмування не є високомодульною, великі додатки, створені на цій мові, буде важко підтримувати.
- Неявне перетворення типів може привести до несподіваних помилок.
- Фреймворк має поганий метод обробки помилок.
- Технологія не здатна підтримувати велику кількість додатків. Нею дуже складно управляти.
- Вона не дасть продуктивності, наприклад, мов C або C ++. Оскільки це скриптова мова і інтерпретується, вона буде трохи повільніша, ніж оптимізовані програми на C [23].

Безпосереднє порівняння характеристик

Серед оглянутих мов програмування для розробки проекту найбільше підходить мова С. Ця мова програмування є дуже гнучкою та містить в своєму арсеналі широкий спектр вбудованих функцій. В С хороший розподіл динамічної пам’яті, що дозволить використовувати кінцевий продукт на комп’ютерах з низькими системними характеристиками. Також, всю програму можна зручно поділити на окремі модулі що додає до вигоди написання системи та подальшого вдосконалення продукту.

Основною причиною відмовитися від мови Python стала низька швидкість роботи та потреба в великій кількості пам’яті, що звичайно ж навантажуватиме ПК. Такий продукт як WAF не повинен впливати на швидкість роботи комп’ютера і сповільнювати обробку запитів клієнта.

PHP хоч і досить швидкий та все ж поступається мові С. Відлякує від себе і складність написання самого коду, в ході розробки часто виникають помилки які потребують час для їх вирішення.

2.2. Вибір середовища розроблення програми

Visual Studio Code - це редактор коду розроблений компанією Microsoft. Visual Studio Code - це "вільний редактор, який допомагає програмісту писати код, допомагає в налагодженні і коригує код за допомогою методу intelli-sense".

Його можливості дозволяють користувачеві модифікувати редактор відповідно до потреб, що означає, що користувач може завантажувати бібліотеки з Інтернету і інтегрувати їх в код відповідно до своїх вимог [24].

Серед переваг середовища [23]:

- Дуже зручне управління вікнами з вкладками, що дозволяє найкращим чином розділити простір екрану.
- Чистий і дружній інтерфейс.
- Вбудований термінал.
- Інструменти налагодження візуально привабливі.
- Функціональність програми може бути розширена за допомогою безлічі різних розширень / плагінів.
- Досить швидкий і простий у використанні.

Недоліки Visual Studio Code:

- Деякі плагіни / розширення, які доступні, можуть викликати проблеми зі стабільністю роботи, особливо якщо вони встановлені разом.
- Час від часу додаток споживає більше ресурсів, ніж необхідно.

CLion - це інтелектуальна, крос-платформна IDE, яку можна використовувати для розробки додатків на декількох мовах програмування. При цьому система орієнтована в основному на C + і C ++. Вона пропонує підтримку основних технологій, а інші функціональні можливості можна отримати з вбудованих інструментів і розширень. CLion полегшує розробку

чистого коду за допомогою миттєвого аналізу коду, автоматичного завершення рядків, форматування коду і інтелектуального редактора коду. Всі ці функції дозволяють підвищити продуктивність роботи. Інтеграція з Valgrind Memcheck дозволяє програмістам перевіряти код та виявляти проблеми з пам'яттю, забезпечуючи бездоганну роботу програми, що розробляється [25].

Переваги середовища:

- Зручний інтерфейс користувача.
- Підтримка VIM для швидкого написання коду.
- Інтелектуальне завершення коду.
- Крос-платформна підтримка.
- Функція поділу вікон.
- Підтримує як C, так і C ++.
- Надає безкоштовну студентську ліцензію.

Недоліки CLion:

- Не має вбудованого компілятора.
- Погано розвинена підтримка Makefiles [26].

Також часто зручно використовувати такий текстовий редактор як Code Writer.

Це безкоштовний редактор тексту і коду з більш ніж 20 підтримуваними типами файлів і активної підсвічуванням синтаксису, яка оновлюється по мірі редагування документів. Використовую його як заміну Блокноту і іншим додаткам для швидкого редагування. Додаток швидкий і плавний. Code Writer ідеально підходить для редагування будь-яких текстових файлів, проведення оглядів коду або презентацій з прикладами коду. Текстовий редактор дозволяє редагувати документи практично в повноекранному режимі, при цьому хромований текст зникає на задньому плані. Складна бічна панель містить такі функції, як провідник документів, пошук,

розширені команди редагування, налаштування і допомогу. Спливаюча палітра команд забезпечує швидкий доступ з клавіатури практично до всіх команд програми. Можна обрати тему редактора і інші параметри, бажаний розмір шрифту, коротше кажучи- створити ідеальне середовище редагування [27].

2.3. Додаткові технології

Joomla

Для перевірки роботи системи протидії потрібно створити сайт, для цього використовувалася платформа побудови веб-сайті та додатків Joomla . Це система управління контентом (CMS), яка підключає сайт до бази даних MySQLi, MySQL або PostgreSQL, щоб полегшити управління та доставку контенту як для менеджера сайту, так і для відвідувачів [28].

Joomla - це повністю безкоштовне рішення доступне кожному, хто бажає створювати динамічні і надійні сайти для різних цілей. Вона здатна вирішувати найрізноманітніші завдання - від корпоративних сайтів і блогів до соціальних мереж і електронної комерції. Joomla підтримується великою і активною спільнотою розробників. Завдяки тисячам унікальних і корисних розширень і шаблонів, в даний час це одна з найбільших платформ для створення сайтів. Широкий набір інтегрованих технологій Joomla дозволяє впроваджувати інновації, що виходять далеко за рамки простого веб-сайту [29].

WAF розроблялася в якості модуля Apache.

Вебсервер Apache – це популярне програмне забезпечення розроблене і підтримується компанією Apache Software в США. Багато серверів по всьому світу управляються вебсервером Apache. Причина цього - швидкість і безпеку, які забезпечує ця компанія [27].

2.4. Висновки до розділу

В цьому розділі було розглянуто основні засоби та технології для реалізації системи. Було проведено аналіз серед мов програмування та описано переваги інших інструментів в порівнянні з аналогами
Для розробки системи було обрано мову C, так як вона є максимально зручною для виконання завдання проєкту, містить вбудовані функції, спрощує код, структура мови є простою та зрозумілою і при цих характеристиках система працює швидко та без загромадження пам’яті.

Проєкт розроблявся в таких середовищах як Visual Studio Code, CLion та Code Writer, де в основним критерієм вибору була зручність використання та низька навантаженість на ПК.

Для проєктування сайту оптимальним варіантом була платформа Joomla, так як вона є дуже зручною та сучасною.

3. РОЗРОБЛЕННЯ ПРОГРАМНИХ МОДУЛІВ

3.1 Опис системних файлів

Вся програма складається з 8 файлів (Рис 3.1):

два файли Header

mod_waf.h

waf_file_operations.h

та шість основних С модулів.

anomaly_detection_phase.c – основний файл для виявлення аномалій до якого підключені інші модулі

mod_waf.c – модуль перевірки сигнатур

waf_detection_mode.c – файл режиму виявлення

waf_generate_profile.c – створення профілю нормальності

waf_parse_signature.c – модуль синтаксичного розбору запиту

waf_train_mode.c – файл для режиму навчання WAF

Така структура дозволяє оперативно керувати додатком та швидко вносити зміни в відповідні модулі, також розробник який перший раз бачить дану програму зможе відразу зорієнтуватися в її будові.

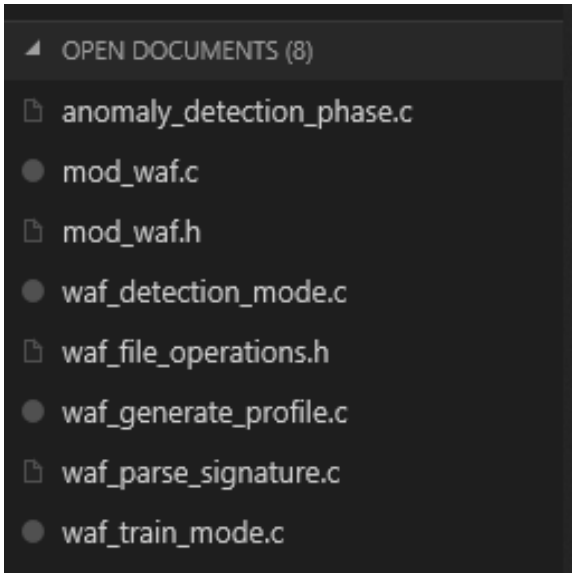


Рисунок 3.1 – Файли WAF

3.2 Високорівневий дизайн

Запити, направлені на сервер, спочатку проходять через брандмауер веб-додатків. WAF перевіряє запити на наявність шкідливого вмісту та відхиляє запит, якщо воно виявлено. WAF допускає прохід до серверу тільки легітимному трафіку. Виявлення шкідливого вмісту в запиті складається з двох частин:

3.2.1. Сигнатура

WAF сканує запит і перевіряє наявність атак на основі правил, визначених у списку сигнатур (Рис 3.2). Сховище сигнатур містить правила для перевірки WAF в закодованому вигляді, проти поширених атак, таких як XSS, SQL injection і т.д. Якщо система виявляє бідь-які нові види атак то можна додати сигнатуру в сховище для подальшого виявлення подібних запитів. Тобто, список сигнатур можна поповнювати та розширювати стек перевірки інформації що надходить. В WAF що розроблено впроваджено перевірку таких символів: апостроф, вирази ‘select’, ‘union all’ та ‘javascript’, знак більше та менше, послідовність двох крапок та слеш, а також інших виразів що можуть бути частиною небезпечного запиту.

Приклади сигнатур системи:

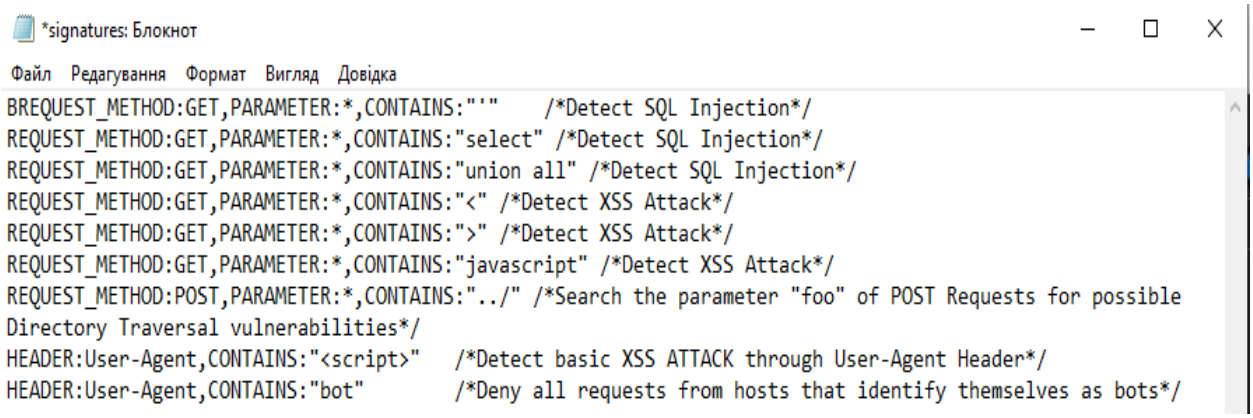


Рисунок 3.2 – сигнатури WAF

3.2.2 Виявлення аномалій

При виявленні аномалій система спочатку вивчає, як виглядає легітимний трафік. Грунтуючись на цій зібраній інформації, WAF робить розрахунки про дозволені запити і продовжує фільтрувати інформацію, яка сильно відхиляється від норми. Виявлення аномалій можна розділити на дві фази: фазу навчання і фазу виявлення.

На етапі навчання WAF не зупиняє жоден запит, дозволяючи всім їм потрапити на сервер. Метою цієї фази є збір інформації. Дані отримані в ході навчання використовуються для розрахунку наступних параметрів:

- Максимальна кількість параметрів для всіх запитів на всіх сторінках.
- Максимальна кількість символів для кожного запиту.
- Середня довжина кожного конкретного параметра для кожного запиту.
- Набори символів (допустимі символи) кожного конкретного параметра для кожного запиту.

На етапі виявлення загроз WAF відстежує вхідні запити на основі обчислень.

Запити, що перевищують максимум за кількістю параметрів на всіх сторінках або для однієї сторінки, не пропускаються.

Якщо довжина параметрів виходить за межі діапазону середнього значення на ’’+- 3’’ то запит також відхиляється системою.

Запити де параметри містять символи, що не зустрічаються в допустимому наборі, також ігноруються.

3.3 Низькорівневий дизайн

Брандмауер вебсерверів реалізований у вигляді модуля Apache. Apache має модульний дизайн, що дозволяє нам писати явні функціональні блоки для сервера. Новий, окремий модуль повинен бути підключений до сервера в відповідному місці технологічного процесу. Функція hook (Рис 3.3) надається Apache та використовується для реалізації цього завдання. Оскільки

потрібно, щоб модуль WAF викликався на початку кожного запиту процесу, оголошується хук за допомогою APR_HOOK_FIRST.

```
static void waf_register_hooks(apr_pool_t *p)
{
    ap_hook_post_read_request(waf_handler, NULL, NULL, APR_HOOK_FIRST);
}
```

Рисунок 3.3 – Оголошення hook

Web application firewall складається з двох частин фільтрації: перевірка сигнатур і виявлення аномалій.

3.3.1. Перевірка сигнатур (mod_waf.c, waf_parse_signature.c)

Міжсайтовий скриптинг (XSS) - це тип ін'єкційних атак, коли шкідливі скрипти впроваджуються в доброякісні веб-сайти. Аналогічно, SQL-ін'єкція - це тип атаки, коли SQL-запити вводяться в поле вводу для виконання. Ці атаки дуже поширені, і сигнатури є хорошим способом захисту від них. У WAF зберігається запис сигнатури, яка містить правила обробки атак в закодованому форматі. WAF перевіряє наявність цих форматів у переданому URL і фільтрує відповідним чином. Щоб перевірити наявність правильних підписів, модуль спочатку зчитує закодовані правила з файлу сигнатур. Потім WAF переходить до перевірки параметрів заголовка і методу (GET або POST), який був переданий (Рис 3.4).

```
// examine GET parameters to check characteristic of malliciousness
if(gNum >0 && strcmp(r->method, "GET")==0){
    if(examineGetParams(r, gList, gNum)==MALICIOUS){
        ap_set_content_type(r, "text/html");
        ap_rprintf(r,"<body bgcolor=\"#B0E0E6\"><H2><center>Request is blocked
        return DONE;
    }
}
// examine POST parameters to check characteristic of malliciousness
if(pNum > 0 && strcmp(r->method, "POST")==0){
    if(examinePostParams(r, pList, pNum)==MALICIOUS){
        ap_set_content_type(r, "text/html");
        ap_rprintf(r,"<body bgcolor=\"#B0E0E6\"><H2><center>Request is blocked
        return DONE;
    }
```

Рисунок 3.4 -- Функція вивчення параметрів GET та POST

Якщо буде виявлено, що будь-які з цих параметрів містять шкідливі сигнатури, запит негайно відкидається без подальших дій. Будь-які нові сигнатури можуть буті додані в файл сигнатур у відповідному кодованому форматі, який буде використовувати система на випадок повторних запитів.

```
int i = 0;
for(i = 0; i < getParamsArr->nelts; i++) {

    if(0 == isMalicious(r, getParams[i].key, getParams[i].val, getSigs, getSigsCount)){
        return MALICIOUS;
    }
}
return NON_MALICIOUS;
```

Рисунок 3.5 -- Функція перевірки параметру запиту

У вище представленій функції для кожного параметра GET-запиту перевіряється, чи є параметр запиту шкідливим, тобто визначеним у SignatureList, якщо параметр є зловмисним, повертається MALICIOUS, інакше повернеться NON_MALICIOUS (Рис 3.5).

3.3.2. Знаходження аномалій (anomaly_detection_phase.c, waf_train_mode.c, waf_generate_profile.c, waf_detection_mode.c)

Перевірка наявності сигнатур може захистити тільки від обмеженої кількості атак, тих що присутні в попередніх записах. Для атак з невідомими сигнатурами система аналізує санкціонований трафік і потім фільтрує ті запити, які в значній мірі відхиляються від дозволених. Виявлення відхилень ділиться на три режими, а саме: режим навчання, режим генерації нормальності, режим виявлення.

Режим навчання (anomaly_detection_phase.c, waf_train_mode.c)

У режимі навчання WAF нічого не фільтрує. В цьому режимі дозволені всі запити. Передбачається, що всі запити в цьому режимі є легітимними. Мета тут полягає в тому, щоб зібрати інформацію про запити. Вся інформація для конкретної сторінки, тобто URL і максимальна кількість параметрів, а також вся інформація для конкретного параметра, тобто URL і кількість введених символів, збирається і зберігається у файловому сховищі (Рис 3.6). Дані, записані в файл, знаходяться в зашифрованому форматі для того щоб уникнути їх несанкціонованого використання, якщо раптом якось потраплять до рук зловмисників.

```
void update_param_file
void waf_expand_page_struct
void waf_expand_param_struct
void record_page_file
void record_param_file
void add_char_set
void update_param_file
void update_page_file
```

Рисунок 3.6 – Функції модуля навчання

Режим генерації профілю нормальності (anomaly_detection_phase.c, waf_generate_profile.c)

Зібрана інформація витягується з файлового сховища для обчислення наступних параметрів:

Максимальна кількість параметрів для всіх сторінок

Максимальна кількість параметрів для кожної сторінки

Середня довжина значень

Стандартне відхилення значень параметрів

Допустимий набір символів

Розраховані параметри знову поміщаються в файлове сховище (Рис 3.7).

```
 while (r > 0) {
    memcpy(&temp, buffer, sizeof(int));
record_buf = malloc(temp);
if (record_buf == NULL) {
    break;
}
if (temp == 0) {
    break;
}
fseek(param_file, -sizeof(int), SEEK_CUR);
r = fread(record_buf,1, temp, param_file);

param_struct = malloc(temp + 1);
memset(param_struct, 0, temp+1);

if (param_struct == NULL) {
    break;
}
memcpy(param_struct, record_buf, temp);
param_struct->avg_param = param_struct->param_sum/param_struct->param_count;
param_struct->sd_param = sqrt((param_struct->sd_sum/param_struct->param_count)-
(param_struct->avg_param * param_struct->avg_param));
fseek(param_file, -temp, SEEK_CUR);
ret = fwrite(param_struct, 1, temp, param_file);

if (ret == -1) {
    break;
}
```

Рисунок 3.7 – Обчислення параметрів запиту

Режим виявлення (anomaly_detection_phase.c, waf_detection_mode.c)

У режимі виявлення кожен запит ретельно перевіряється з використанням параметрів, створених в попередньому режимі.

Якщо для будь-якого запиту кількість параметрів перевищує загальний максимальний параметрів, то його буде відхилено.

Якщо для будь-якого запиту кількість параметрів знаходиться в межах загального ліміту, але перевищує певну кількість максимальних параметрів для даного url, то він також відхиляється системою.

Для кожного параметра кожного запиту, що мають значення довжини близько середнього значення ’’ + - 3 ’’, пропускаються далі. Потім, набір символів кожного параметра для кожного запиту також перевіряється, і запити, що містять символи, які не присутні в наборі допустимих символів, відхиляються системою.

Нижче наведені різні прапори, які повертаються на основі фільтрації при виявленні відхилень від стандартних записів.

- EXCEED_ALL_MAX_PARAM: Повертається, коли кількість параметрів перевищує загальну кількість на всіх сторінках. Запит відхиляється системою (Рис 3.8).

```
if (ret == 0){
    if(max_param_all >= max_param){
        ret = PASS_DETECTION;
    }else{
        ret = EXCEED_ALL_MAX_PARAM;
    }
}
```

Рисунок 3.8 – Функція перевірки максимальної кількості параметрів на всіх сторінках

- EXCEED_MAX_PARAM: Повертається, коли кількість параметрів перевищує загально допустиму кількість для даної сторінки. Запит

відхиляється системою (Рис 3.9).

- MAX_PARAM_PASS: Повертається, якщо кількість параметрів знаходиться в межах допустимого. Запит пропускається для подальшої перевірки.

```
if(page_struct->max_param >=max_param){
    ret = MAX_PARAM_PASS;
} else {
    ret = EXCEED_MAX_PARAM;
}
break;
```

Рисунок 3.9 – Функція перевірки допустимої довжини параметра

- PARAM_LEN_ILLEGAL: Повертається, якщо довжина значення параметра знаходиться за допустимими межами, прийнятих для нормального розподілу. Запит відхиляється.
- UNKNOWN_PARAM: Повертається, якщо запит містить параметр, що не зустрічався за час режиму навчання (Рис 3.10). Запит відхиляється.

```
if (ret == 0) {
    ret = UNKNOWN_PARAM;
}
```

Рисунок 3.10 – Функція виявлення нового запиту

- CONTAINS_NO_SEEN_CHAR: Повертається, якщо якесь з значень параметра містить недопустимий символ. Запит відхиляється.
- PASS_DETECTION: Повертається, коли всі критерії виконані (Рис 3.11). Тобто всі перевірки пройдено і відповідно запит пропускається та потрапляє на веб-сервер.

```
if (strcmp(uri, uri_name) == 0 && strcmp(param, param_name) == 0){
    if(len > (param_struct->avg_param + 3*param_struct->sd_param) ||
     len <(param_struct->avg_param - 3*param_struct->sd_param)){
        ret = PARAM_LEN_ILLEGAL;
    }else if(!check_char_set(charset, charset_name)){
        ret = CONTAINS_NO_SEEN_CHAR;
    }else {
        ret  = PASS_DETECTION;
    }
    break;
}
```

Рисунок 3.11 – Функція перевірки допустимих меж довжини параметра та валідність символів що містить запит

3.4. Висновки до розділу

В даному розділі було представлено опис програмного забезпечення яке було розроблене в ході виконання дипломного проєкту. Показано основні системні файли що містять програмний код.

Представлено функціональні вимоги комп’ютерної системи, всі вони були реалізовані.

Програма поділяється на дві головні частини: перша відповідає за виявлення SQL- ін’єкцій та XSS атак, а інша за знаходження аномальних запитів на вебсервер, що можуть представляти небезпеку. Але обидві ці частини взаємопов’язані між собою і частково залежать одна від одної.

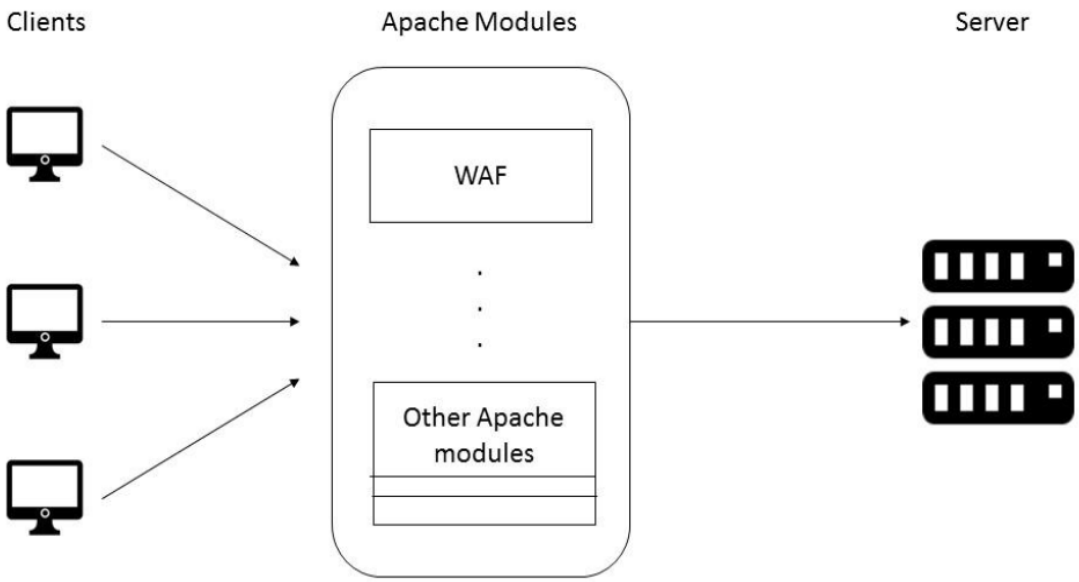


Рисунок 3.12 – WAF в клієнт-серверній архітектурі

4. АНАЛІЗ РОЗРОБЛЕНОЇ СИСТЕМИ

4.1. Особливості тестування

До тестування системи було висунуто наступні вимоги :

- Встановити додатки для реалізації тестування
- Створити тест кейси для перевірки
- Перевірити правильність відповіді системи для всіх можливих вхідних даних
- Перевірити практичність використання фаєрволу
- Перевірити сумісність з програмним забезпеченням

4.1.1. Розгортання додаткових технологій

Для початку було встановлено Apache тому, що розроблена система повина працювати як його модуль. Apache - найпопулярніший веб-сервер, що обслуговує більше половини активних сайтів в Інтернеті і відповідає потребам великих і малих проектів. Тому його використання доцільне. Всі компоненти веб-сервера Apache є портативними.

Інсталюємо сервер (Рис 4.1):

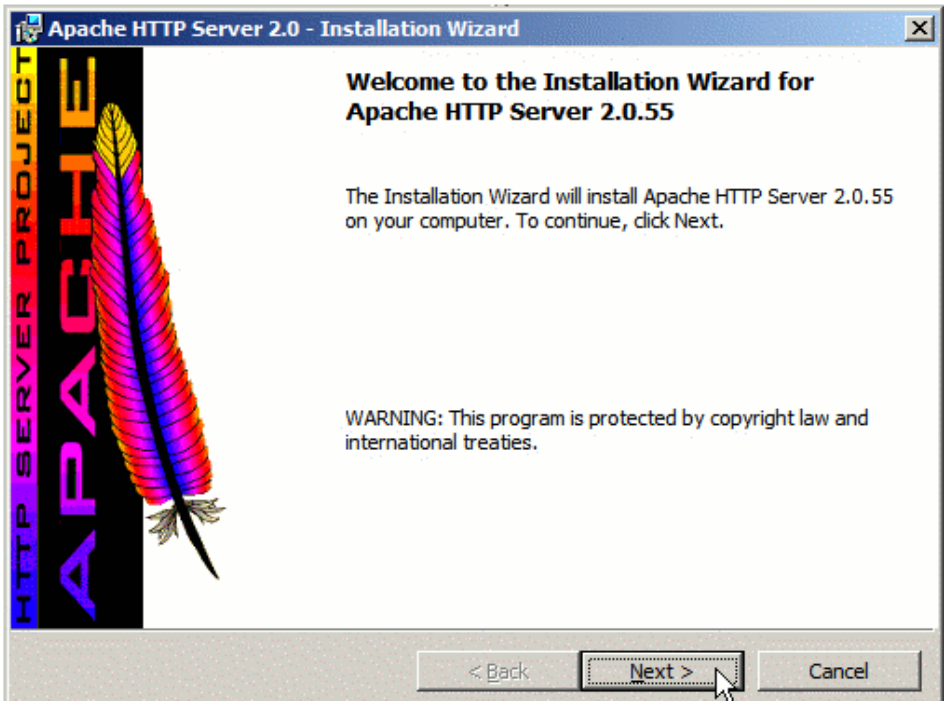


Рисунок 4.1 – Скріншот встановлення Apache

Після налаштувань можна вважати, що сервер встановлений і готовий до роботи (Рис 4.2).

Для перевірки працездатності, вводимо в браузері http: // localhost /

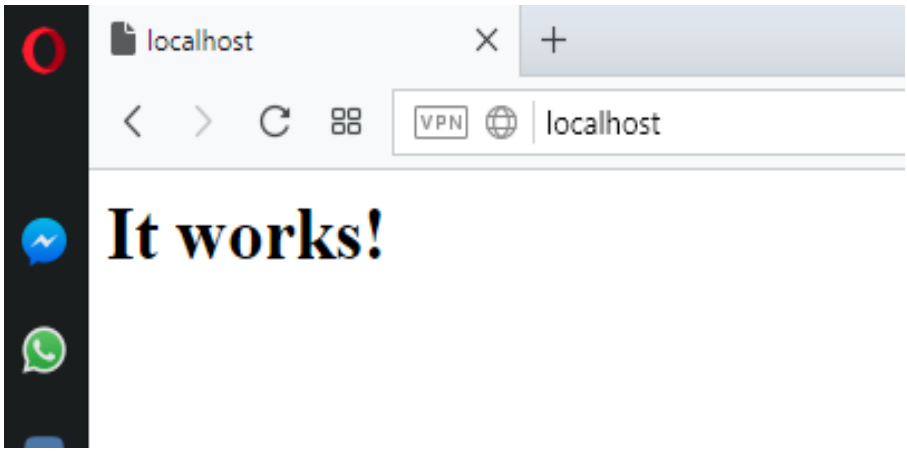


Рисунок 4.2 – Скріншот роботи сервера

А також додатково встановлюємо Joomla, додаток для створення зразку сайту, який розміщений на сервері Apache, розташований на локальній машині (Рис 4.3).

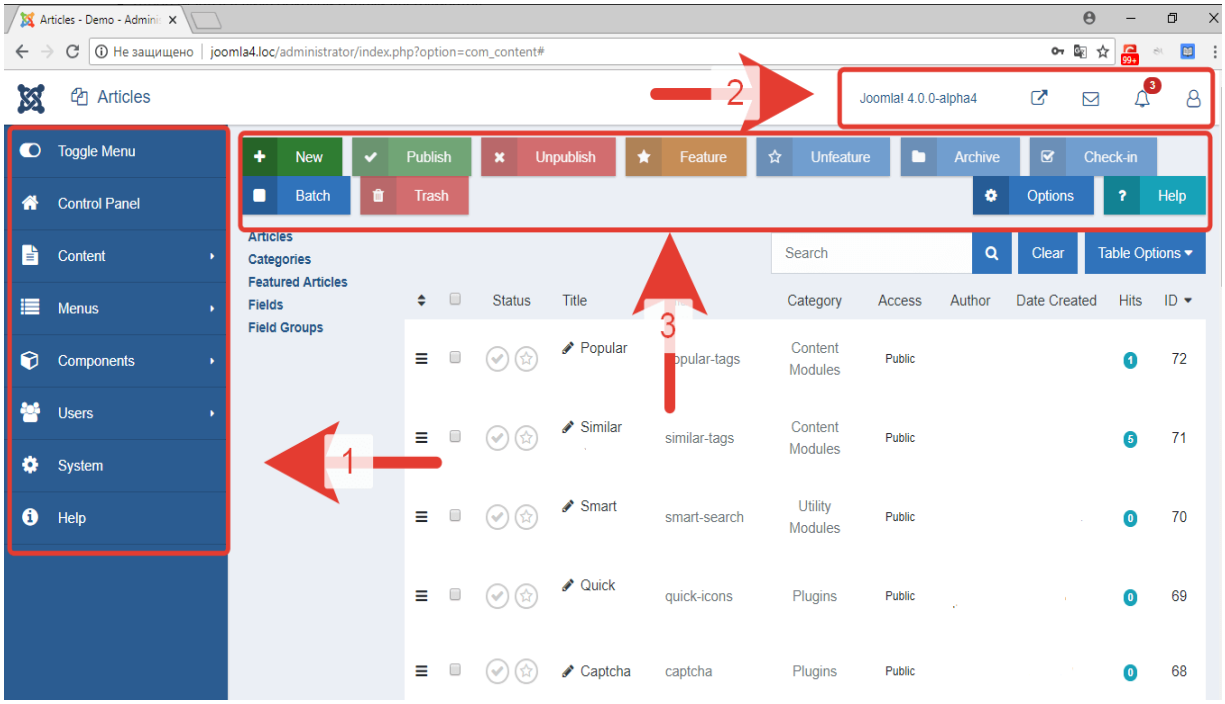


Рисунок 4.3 – Скріншот інтерфейсу панелі керування CMS

Після встановлення та входу розпочинаємо роботу з CMS.

Було розроблено різні форми для введення запитів: числові, буквенні, вибір з доступних варіантів ()Рис 4.4. Тобто, тепер все повністю готове для проведення тестування.

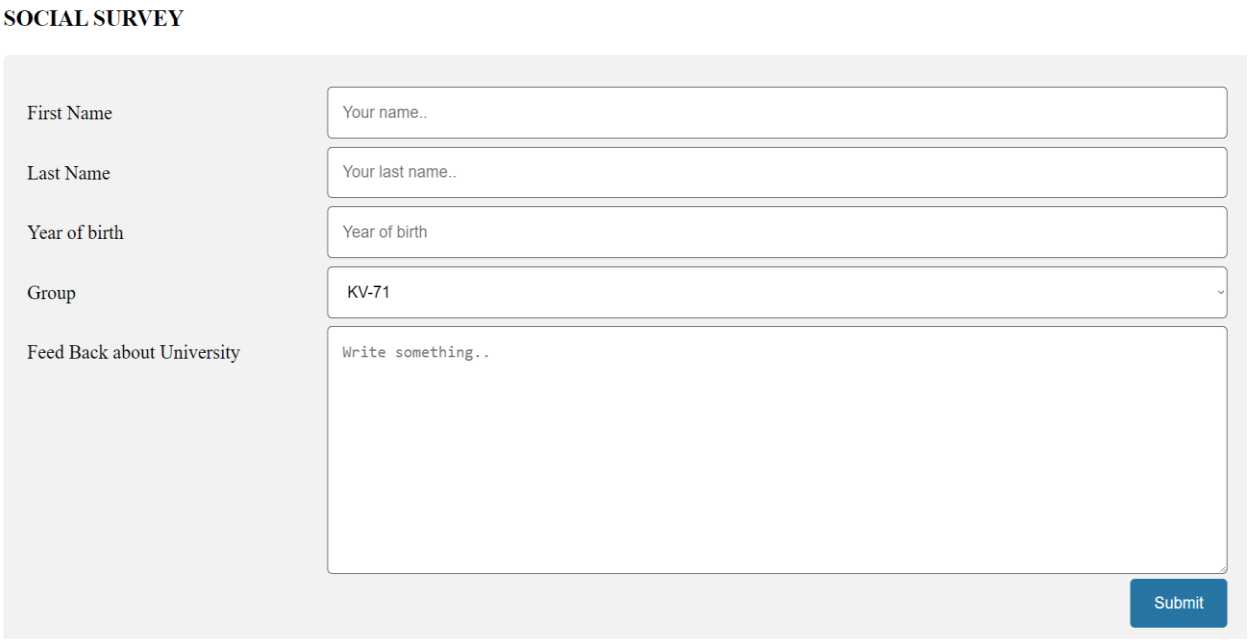


Рисунок 4.4 – Інтерфейс створеної форми для тестування

На сайті передбачена можливість заповнення деяких полів та відповідно відправлення GET і POST запитів на інші сторінки. Всі запити спочатку направляються в WAF, перш ніж потрапляють на сервер для отримання фактичного вмісту, присутнього на цих сторінках.

4.1.2. Написання тест кейсів

Для ідеальної перевірки попередньо були написані документи для окремих функцій.

Тест-кейс №1

ID:0

Заголовок: Кейс перевірки можливості проходження SQL- ін'єкції.

Кроки для відтворення:

- Відкрити форму на створеному сайті.
- В рядок ‘’ім’я’’ або ‘’прізвище’’ ввести запит для бази даних.
- Інші рядки заповнити коректно.
- Натиснути клавішу відправлення.
- Перевірити співпадіння з очікуваним результатом.

Очікуваний результат:

WAF блокує запит.

Тест-кейс №2

ID:1

Заголовок: Кейс перевірки можливості реалізації XSS атаки.

Кроки для відтворення:

- Відкрити форму на створеному сайті.
- В будь-який рядок форми ввести скрипт: <script>alert(123)</script>.
- Натиснути клавішу відправлення.
- Перевірити співпадіння фактичного результату з очікуваним.

Очікуваний результат:

WAF блокує запит.

Тест-кейс №3

ID:2

Заголовок: Кейс перевірки режиму навчання системи.

Кроки для відтворення:

- Виконати попереднє навчання програми для рядку ‘’Рік народження’’.
- Відкрити форму на створеному сайті.
- В рядок ‘’Рік народження’’ ввести буквені символи.
- Інші рядки заповнити коректно.
- Натиснути клавішу відправлення.

- Перевірити відповідність з очікуваним результатом.

Очікуваний результат:

Відхилення через недозволенний набір символів.

4.1.3. Тестування системи

Виконання першого тест-кейсу:

Перевірка сигнатури.

Виявлення сигнатури SQL-ін'єкції із запиту. Спроба виконати запит "Select" (Рис 4.5) з текстового поля.

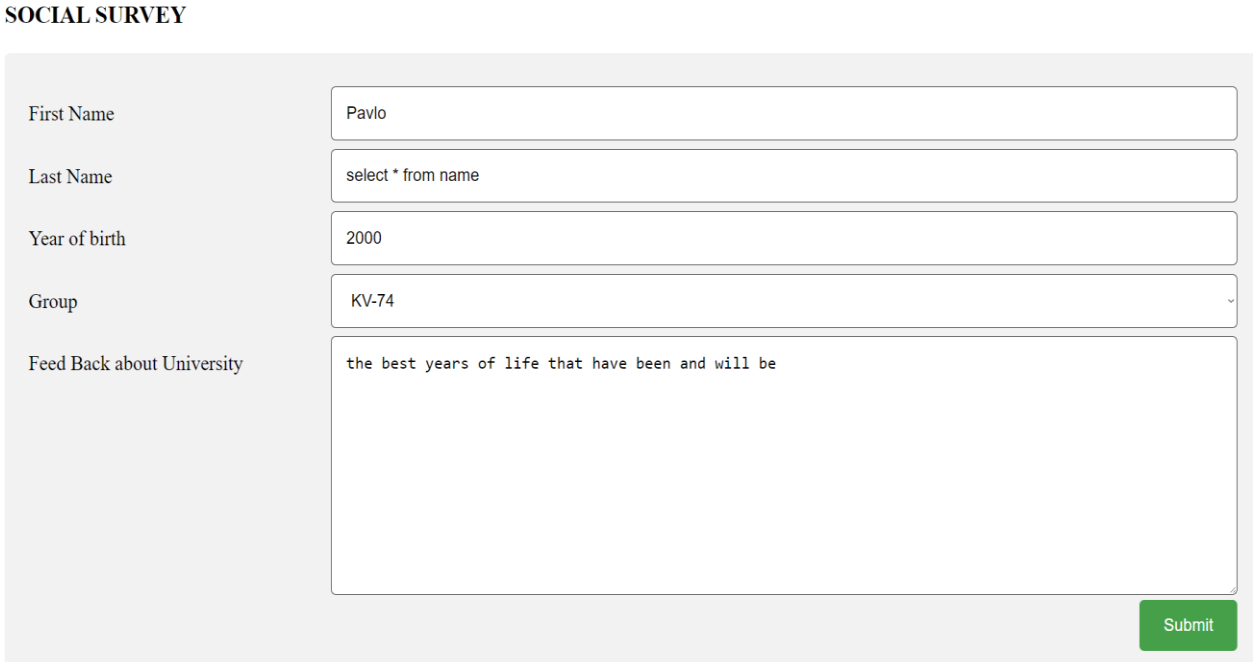


Рисунок 4.5 – Запит з SQL ін’єкцією

Атака з ін’єкцією запиту SQL відхилена (Рис 4.6).

Рисунок 4.6 – Результат спроби введення SQL ін’єкції

Перевірка XSS-атакою (Рис 4.7), за допомогою другого тест-кейсу

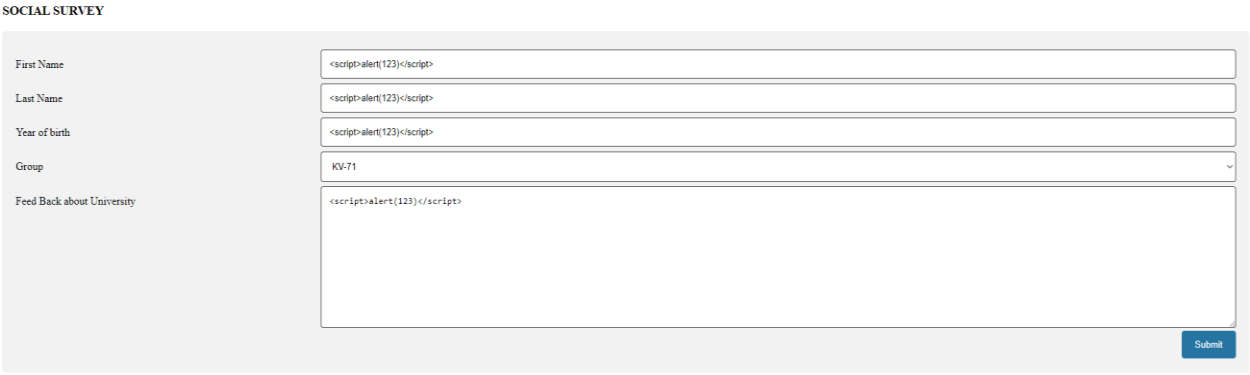


Рисунок 4.7 – Запит з скриптами XSS атаки

В результаті запит не було прийнято, і надалі нічого не відбувалося, також в деяких випадках (при різному наборі символів) можлива поява повідомлення про використання недопустимих символів

Вигляд повідомлення яке отримують сайти, що вразливі (Рис 4.8) до даного типу атак:

Рисунок 4.8 – Повідомлення якщо сайт є вразливим до XSS атак

Перевірка режиму навчання системи

В режимі навчання набір символів параметру ’’Рік народження’’ є числовим (Рис 4.9)

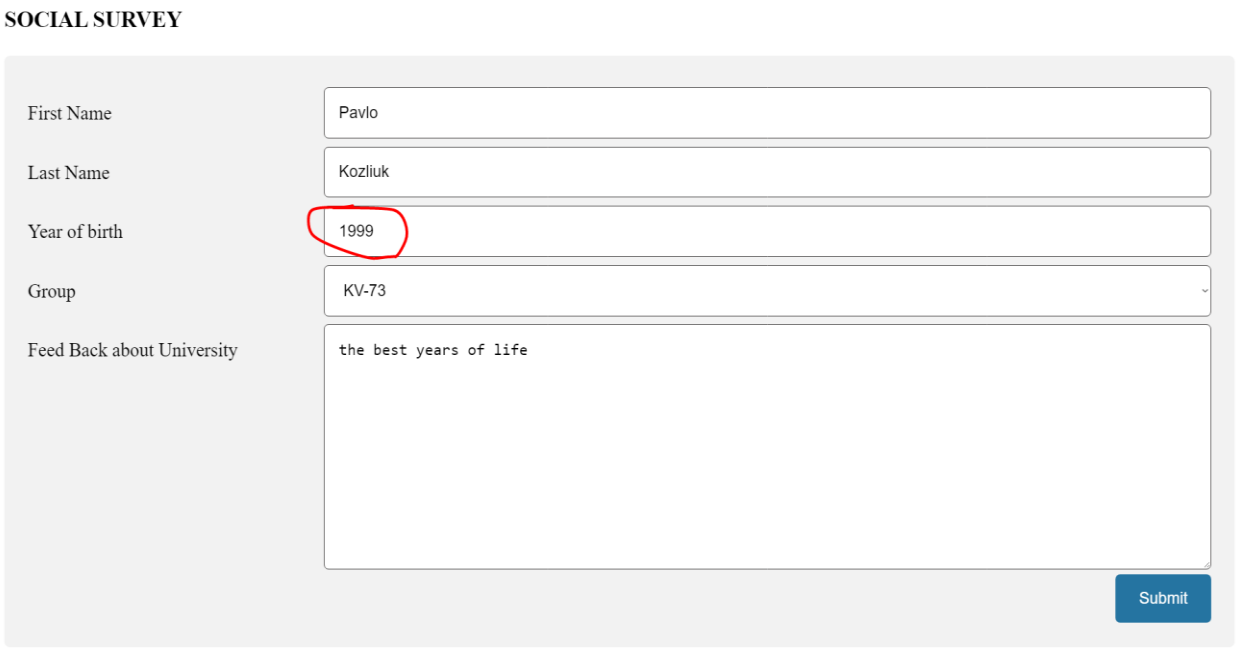


Рисунок 4.9 – Навчання системи

Режим виявлення, використання третього тестового випадку:

Під час тестування в полі ’’Рік народження’’ вказуються літери (Рис 4.10)

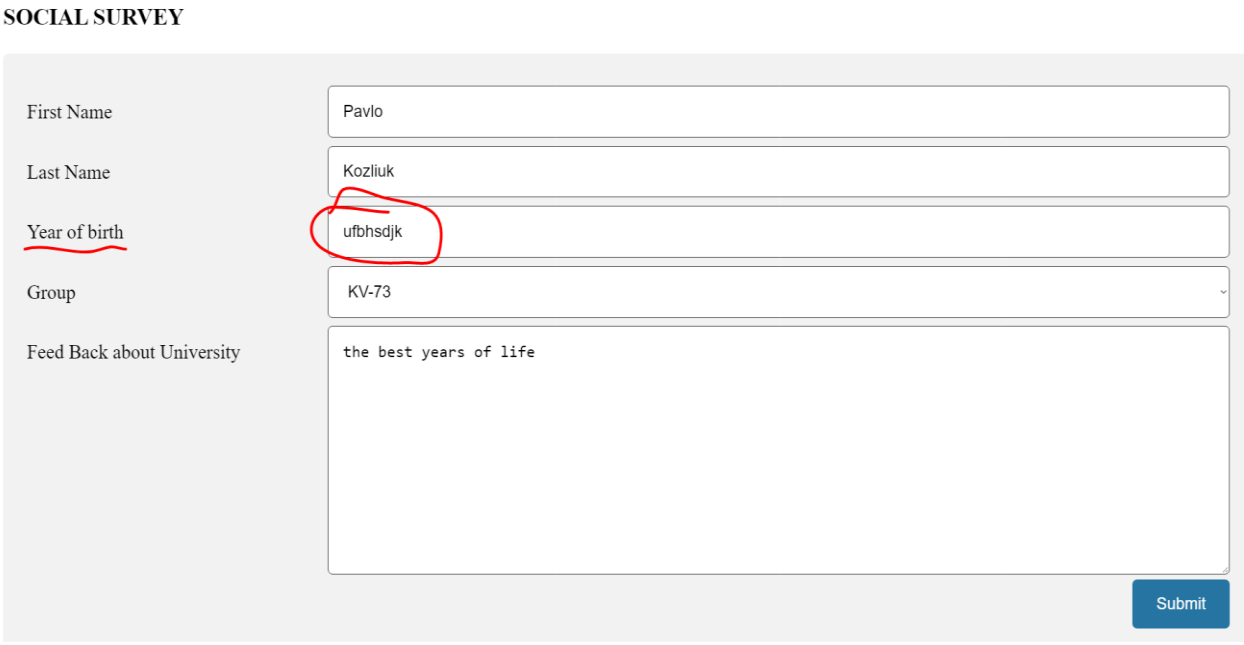
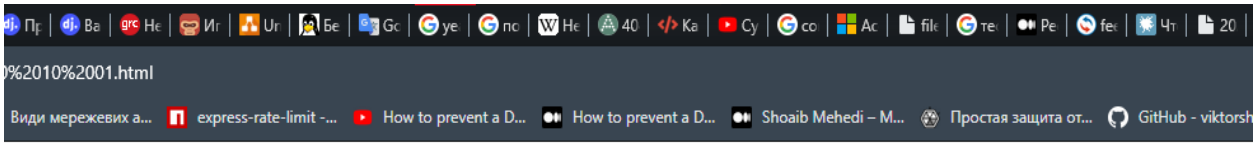


Рисунок 4.10 – Введення букв і числовий формат

Запит відхилено (Рис 4.11)WAF через недопустимий набір символів



REQUEST IS BLOKED BY WAF BECAUSE THE REQUEST CONTAINS ILLEGAL CHARACTER SET

Рисунок 4.11 – Результат перевірки третього кейсу

4.2. Порівняння з аналогами

Розроблений брандмауер для комп'ютера створений спеціально для фільтрації пакетів вхідних даних. Головна задача – це захист мережі від різних загроз.

Переваги розробленої WAF над існуючими аналогами:

- Контроль вхідного трафіку.
- Висока швидкість роботи.
- Простота в інсталяції.
- Висока пропускна здатність.
- Низькі вимоги до параметрів комп'ютера.
- Простота використання

4.3. Рекомендації для подальшого вдосконалення

Проект з розроблення WAF можна продовжувати шляфом додавання додаткових фільтрів або перевірки самої концепції поведінки клієнта.

Наприклад додати модуль що відповідатиме за блокування IP- мереж від яких іде надто багато однотипних запитів, тим самим програма буде допомагати уникати DDOS атак.

Також, можна обмежити тривалість запитів клієнта до певного часу, незалежно від того, що саме він робить. В результаті час здійснення запиту обмежується як для реального користувача, так і для скрипта. Тобто впровадити таку собі інкрементну затримку для регулювання запитів.

Це внесе ще декілька особливостей до системи.

- Обмеженян частоти звернень може контролювати величину вхідного і вихідного трафіку.
- Таким чином можна обмежити кількість запитів на сервер від одного окремого користувача.
- Це допомагає захиститися від DOS/DDOS і брутфорс атак.

4.4. Висновки до розділу

В даному розділі було описано послідовність дій для того щоб протестувати написану програму. Встановлено Apache та підключено розроблений проєкт як модуль серверу. Також за допомогою додатку Joomla створено веб сторінку (форму) через яку і були направлені різні види атак, і таким чином перевірено правильність роботи розробленого додатку Web Application Security

Тепер можна стверджувати що система відповідає всім вимогам які були попередньо обгрунтовані та поставлені як завдання на дипломний проєкт.

Також розкрито можливі сценарії подальшого розроблення системи, вдосконалення та поліпшення.

ВИСНОВКИ

Метою даного дипломного проекту було створення зручної комп'ютерної системи захисту вебсерверу від кібератак.

Використання розробленого продукту забезпечить надійний захист комп’ютера під час користування веб-сервісами.

Аналіз актуальної ситуації та існуючих програмних рішень в даній сфері, попередньо виконаний в дипломному проєкті, показав важливість розробки. В процесі розроблення були використані відповідні технології, а саме:

- Мова програмування C для реалізації перевірки вхідних даних на сервер.
- Система керування вмістом Joomla для створення вебсайту та можливості практичної перевірти роботи сервісу.
- Сервер Apache для можливості підключення модулю що розроблявся в ході всього дипломного проєкту.

Розроблене програмне забезпечення:

- Дозволяє уникнути атак на сервер.
- Зберігає конфіденційність даних.
- Забезпечує захист комп’ютера.

Розроблення виконано у повному обсязі, відповідає всім вимогам, описаним в документі, тестування продукту виконано з усіма необхідними критеріями та правилами. Також надано рекомендації щодо подальшого вдосконалення та розширення функціоналу розроблюваної комп’ютерної системи.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Захист WEB-додатків: [Веб-сайт]. URL: https://octavacapital.ua/zahyst-web-dodatkiv-chomu-ce-aktualno/ (дата звернення: 26.05.2021).
2. Что такое веб-сервер - Изучение веб-разработки: [Веб-сайт]. URL: https://developer.mozilla.org/ru/docs/Learn/Common_questions/What_is_a_web_server. (дата звернення: 26.05.2021).
3. Apache: [Веб-сайт]. URL: https://betterstudio.com/blog/what-is-apache/ (дата звернення: 28.05.2021).
4. Most Common Types of Cyber Attacks: [Веб-сайт]. URL: https://www.lepide.com/blog/the-15-most-common-types-of-cyber-attacks/. (дата звернення: 18.05.2021).
5. Most Common Types of Cyberattacks on Web Applications in 2020: [Веб-сайт]. URL: https://www.pentasecurity.com/blog/top-7-common-types-cyberattacks-web-applications/. (дата звернення: 29.05.2021).
6. ТЕСТУВАННЯ БЕЗПЕКИ: SQL-ІН’ЄКЦІЇ: [Веб-сайт]. URL: https://training.qatestlab.com/blog/technical-articles/security-testing-sql-injection/. (дата звернення: 31.05.2021).
7. Сетевые угрозы: [Веб-сайт]. URL: https://datbaze.ru/internet/setevyie-ugrozyi.html. (дата звернення: 02.06.2021).
8. Advantages and Disadvantages of C Programming – Discover the Secrets of C: [Веб-сайт]. URL: https://data-flair.training/blogs/advantages-and-disadvantages-of-c/. (дата звернення: 01.06.2021).

9. Python Advantages and Disadvantages – Step in the right direction: [Веб-сайт]. URL: https://techvidvan.com/tutorials/python-advantages-and-disadvantages/. (дата звернення: 01.06.2021).

10. PHP: [Веб-сайт]. URL: https://www.phpbabu.com/advantages-and-disadvantages-of-php/. (дата звернення: 31.05.2021).

11. Чому її обирають: 7 переваг мови програмування PHP: [Веб-сайт]. URL: https://blog.ithillel.ua/ua/articles/chomu-vybyraiut-7-perevah-movy-prohramuvannia-php. (дата звернення: 01.06.2021).

12. Microsoft Visual Studio Code Reviews: [Веб-сайт]. URL: https://www.trustradius.com/products/microsoft-visual-studio-code/reviews?qs=pros-and-cons. (дата звернення: 01.06.2021).

13. What is Visual Studio Code?: [Веб-сайт]. URL: https://www.educba.com/what-is-visual-studio-code/. (дата звернення: 30.05.2021).

14. CLion Reviews : [Веб-сайт]. URL: https://www.trustradius.com/products/clion/reviews?qs=pros-and-cons. (дата звернення: 01.06.2021).

15. Joomla!: [Веб-сайт]. URL: http://whichcmstochoose.com/joomla.html. (дата звернення: 01.06.2021).

16. СЕРВЕРНОЕ WEB-ПРОГРАММИРОВАНИЕ: [Веб-сайт]. URL: https://studref.com/642155/informatika/servernoe_programmirovanie. (дата звернення: 02.06.2021).

17. Кіберпростір: [Веб-сайт]. URL: https://uk.wikipedia.org/wiki/Кіберпростір. (дата звернення: 02.06.2021).

18. «АКТУАЛЬНІ ПРОБЛЕМИ КІБЕРБЕЗПЕКИ»: [Веб-сайт]. URL: http://www.dut.edu.ua/uploads/p_1739_27992763.pdf. (дата звернення: 01.06.2021).

19. Mailbombing: [Веб-сайт]. URL: https://uk.wikipedia.org/wiki/Хакерська_атака. (дата звернення: 02.06.2021).

20. ЗАХИСТ ВІД DDOS-АТАК: [Веб-сайт]. URL: https://iitd.com.ua/zashchita-ot-ddos-atak/. (дата звернення: 02.06.2021).

21. Что такое XSS атака и защититься от неё?: [Веб-сайт]. URL: https://datbaze.ru/article/xss-ataka.html. (дата звернення: 02.06.2021).

22. Glasswire: [Веб-сайт]. URL: https://www.glasswire.com (дата звернення: 02.06.2021).

23. Проверка сайта на вирус: Проверка сайта на вирусы: [Веб-сайт]. URL: https://iapple-59.ru/raznoe/proverka-sajta-na-virus-proverka-sajta-na-virusy.html. (дата звернення: 02.06.2021).

24. Використання XSS уразливостей по максимуму. Easy Hack: Як добути дані через Cross Site Scripting Inclusion Що таке xss вразливість: [Веб-сайт]. URL: https://remzhuk.ru/uk/ispolzovanie-xss-uyazvimostei-po-maksimumu-easy-hack-kak-dobyt-dannye-cherez-cross-site/. (дата звернення: 02.06.2021).

25. Code Writer: [Веб-сайт]. URL: https://www.microsoft.com/en-us/p/code-writer/9wzdncrfhzdt?activetab=pivot:overviewtab. (дата звернення: 02.06.2021).

26. An IDE for C++ programming: [Веб-сайт]. URL: https://clion.en.softonic.com. (дата звернення: 02.06.2021).

27. Joomla!: [Веб-сайт]. URL: https://joomla.en.softonic.com. (дата звернення: 02.06.2021).

28. Види мережевих атак. Способи їх виявлення: [Веб-сайт]. URL: https://holodoks.blogspot.com/2017/12/blog-post.html. (дата звернення: 02.06.2021).

29. WHY WEB APPLICATION SECURITY IS IMPORTANT IN EVERY STEP OF WEB APPLICATION DEVELOPMENT: [Веб-сайт]. URL: https://www.ingeniumweb.com/blog/post/why-web-application-security-is-important-in-every-step-of-web-application-development/2432/. (дата звернення: 02.06.2021).

ДОДАТКИ