Київський національний торговельно-економічний університет
Кафедра програмної інженерії та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка програмного забезпечення захисту серверу»»

Студента 2 курсу, 5м групи, спеціальності 121 «Інженерія програмного забезпечення», спеціалізації «Інженерія програмного забезпечення» _____ підпис студента — Гасімова Орхана Хемідага огли

Науковий керівник кандидат технічних наук, професор _____ підпис керівника — Пашорін Валерій Іванович

Гарант освітньої програми доктор технічних наук, професор _____ підпис керівника — Криворучко Олена Володимирівна

КИЇВ – 2019

Київський національний торговельно-економічний університет

Факультет ФОАІС Кафедра «Програмної інженерії та кібербезпеки»
Спеціальність 121 «Інженерія програмного забезпечення»
Спеціалізація «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри

_____Криворучко О.В.

«___» _____2019 р.

Завдання
на випускну кваліфікаційну роботу студентові

Гасімову Орхану Хемідага огли
(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи
«Розробка програмного забезпечення захисту серверу»

Затверджена наказом ректора від "7" листопада 2018 р. № 4183

2. Строк здачі студентом закінченої роботи _____
_____

3. Цільова установка та вихідні дані до роботи (проекту)

Мета роботи - *виступає розробка програмного забезпечення для захисту серверів.*

Об'єкт дослідження - *система безпеки інформації.*

Предмет дослідження - *розробки программного забезпечення для захисту серверів.*

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ ЗАХИСТУ СЕРВЕРІВ

1.1. Система захисту серверів
1.2. Аналіз аналогів програмного забезпечення для захисту серверів
1.3. Проблематика та постановка завдання дослідження
1.4. Висновки до розділу 1

РОЗДІЛ 2 МЕТОДОЛОГІЧНІ АСПЕКТИ ЗАХИСТУ СЕРВЕРІВ

2.1. Методологія захисту серверів
2.2. Математичне забезпечення захисту серверів
2.3. Висновки до розділу 2

РОЗДІЛ 3 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗАХИСТУ СЕРВЕРІВ

3.1. Опис програмного забезпечення та засобів проектування
3.2. Розробка програмного забезпечення для захисту серверів
3.3. Висновки до розділу 3

РОЗДІЛ 4 АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗАХИСТУ СЕРВЕРІВ

4.1 Верифікація результатів
4.2 Висновки до розділу 4

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

ДОДАТКИ

6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	*Вибір теми випускної кваліфікаційної роботи*	*25.09.2018*	
2.	*Розробка та затвердження завдання на проект магістра (Наказ №185 від 07.11.18)*	*07.11.2018*	
3.	*Вступ та перелік літературних джерел*	*28.02.2019*	
4.	*Розробка технічного завдання*	*22.03.2019*	
5.	*Розділ 1.* (*Назва розділу 1*)	*18.04.2019*	
6.	*Розділ 2.* (*Назва розділу 2*)	*24.05.2019*	
7.	*Розділ 3.* (*Назва розділу 3*)	*24.05.2019*	
8.	*Розділ 4.* (*Назва розділу 4, при необхідності, можливе об'єднання розділів 3 ти 4*)	*20.09.2019*	
9.	*Розробка програми та методики тестування*	*18.10.2019*	
10.	*Написання наукової статті*	*27.09.2019*	
11.	*Керівництво користувача*	*23.10.2019*	
12.	*Висновки та пропозиції*	*01.11.2019*	
13.	*Здача випускної кваліфікаційної роботи на кафедру (перша перевірка)*	*13.11.2019*	
14.	*Підготовка автореферату та презентації доповіді*	*14.11.2019*	
15.	*Попередній захист випускної кваліфікаційної роботи*	*14.11.2019 – 15.11.2019*	
16.	*Здача зброшурованої випускної кваліфікаційної роботи*	*25.11.2019*	
17.	*Зовнішнє рецензування випускної кваліфікаційної роботи*	*26.11.2019*	
18.	*Підготовка до публічного захисту випускної кваліфікаційної роботи*	*02.12.2019-04.12.2019*	

7. Дата видачі завдання **«26» вересня 2018 р.**

8. Науковий керівник випускної кваліфікаційної роботи

_____ Пашорин.В.І

(прізвище, ініціали, підпис)

9. Гарант освітньої програми _____ Криворучко О.В__

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент_____ Гасімов О.Х_____

(прізвище, ініціали, підпис)

11. Відгук керівника випускної кваліфікаційної роботи

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Науковий керівник випускної кваліфікаційної роботи_____ Пашорин.В.І
(підпис, дата)

Відмітка про попередній захист _____
(ПІБ, підпис, дата)

12. Висновок про випускну кваліфікаційну роботу

Випускна кваліфікаційна робота студента _____
(прізвище, ініціали)

може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми _____ Криворучко О.В___
(прізвище, ініціали, підпис)

Завідувач кафедри _____ Криворучко О.В____
(підпис, прізвище, ініціали)

АНОТАЦІЯ

Гасімов О.Х Тема роботи: Розробка программного забезпечення для захисту серверів.

Запропоновано комплексне рішення, щодо вдосконалення антивірусного захисту верверів на всіх рівнях. Наведено методи аналізу інформаційного середовища: формування хронологічної послідовності типових впливів базових факторів середовища для їх подальшого аналізу й прогнозу.

У роботі проведено вивчення та розробка методів вдосконалення антивірусного захисту серверів. Отримані результати можуть бути використані у межах організацій для впровадження засобів інформаційної безпеки по етапах.

ANNOTATION

Gasimov O.H Software development for server protection.

A comprehensive solution is proposed to improve antivirus protection for ververials at all levels. Methods of information environment analysis are presented: formation of chronological sequence of typical influences of basic environmental factors for their further analysis and prognosis.

The paper deals with the study and development of methods for improving the antivirus protection of servers. The results obtained can be used by organizations to implement information security by stages.

ТЕХНІЧНЕ ЗАВДАННЯ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Розробка программного забезпечення для захисту серверів.

Галузь застосування: інформаційні технології.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на ВКР, затверджене кафедрою програмної інженерії та інформаційних систем КНТЕУ.

3. ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для використання в якості программного забезпечення для захисту серверів. За допомогою якого користувачі даної розробки отримають підвищення якості сервісу, швидкість реакції на атаки, можливість контролю доступу в автоматичному режимі, спрощення роботи.

4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

Web-ресурс повинен забезпечувати такі основні функції:

1) можливість блокування країн та IP;
2) захист від XSS та MIME;
3) захист від Clickjacking;
4) можливість обробляти запити;
5) безпечне з’єднання.

Мовою реалізації програмного забезпечення для захисту серверів обрано PHP у якості бази даних обрано SQL.

Додаткові вимоги:

1) наявність меню;
2) наявність кнопок;
3) можливість виводу графічного результату;
4) дизайн сторінок з використанням в якості базових білого та синьо-голубого кольорів.

5. ВИМОГИ ДО ПРОЕКТНОЇ ДОКУМЕНТАЦІЇ

У процесі виконання роботи повинна бути розроблена наступна документація:

1) пояснювальна записка;
2) програма та методика тестування;
3) керівництво користувача;
4) креслення:

– «Діаграма діяльності та послідовності. Схема алгоритму»;
– «Синхронізація даних. Схема роботи програмної системи».

6. ЕТАПИ ПРОЕКТУВАННЯ

Вивчення літератури за тематикою роботи.
Розробки та узгодження технічного завдання.
Розробки структури web-ресурсу.
Розробки дизайну сторінок та графічних елементів.
Програмна реалізація web-ресурсу.
Тестування web-ресурсу.
Підготовка матеріалів текстової частини роботи.
Підготовка матеріалів графічної частини роботи.
Оформлення технічної документації роботи.

7. ПОРЯДОК ТЕСТУВАННЯ РОЗРОБКИ

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

ЗМІСТ

ВСТУП

Актуальність дослідження. В наш час масовий вихід бізнесу в Інтернет має величезні обороти: системи онлайн-банкінгу, різні портали (ігрові, медіа, освітні), інтернет-магазини тощо. Збільшення спектру завдань, що вирішуються інформаційними системами (ІС), видів і складності різних алгоритмів обробки інформації; практично кожна організація в наш час має власний сайт і з кожним роком використовує все більше хмарних технологій у своїй IT-інфраструктурі (пошта, управління проектами, месенджери і т. д.) [1].

Незважаючи на те, що розробка систем захисту інформації у web-інфраструктурі ведеться вже досить довгий час, жодне подібне рішення не набуло широкого поширення в силу складнощів налаштування і впровадження систем, малої ефективності алгоритмів виявлення та запобігання атак, відсутності адекватних інструментів щодо усунення наслідків атак, складності їх адміністрування, а також застосування користувальницької документації без спеціальних знань.

Аналіз робіт, що використовуються в обговорюваній області, показує, що дана проблематика вимагає як подальшого розгляду і вивчення побудови надійних математичних моделей, так і створення найбільш ефективних алгоритмів виявлення, запобігання і ліквідації комп'ютерних атак, що показує актуальність роботи в цьому напрямку.

Мета і завдання роботи. Метою роботи виступає розробка програмного забезпечення захисту серверу.

За для досягнення поставленої мети у роботі необхідно виконати низку завдань:

- описати систему захисту серверів;
- проаналізувати аналоги програмного забезпечення для захисту серверів;
- розкрити проблематику та здійснити постановку завдання дослідження;
- описати методологію захисту серверів;
- навести математичне забезпечення захисту серверів;
- описати програмне забезпечення та засоби проектування;
- розробити програмне забезпечення захисту серверу;
- провести тестування програмного забезпечення захисту серверу.

Об'єкт и предмет дослідження. Об'єктом роботи виступають система безпеки інформації.

Предметом є процес розробки програмного забезпечення захисту серверу.

Методи дослідження. Для вирішення поставлених завдань використовувалися як загальнонаукові, так і спеціальні методи наукового пізнання. Системний аналіз, дедуктивний та індуктивний методи використовувалися при визначенні суті понять «Інформаційна безпека». Інформаційну базу дослідження складають чинні вітчизняні законодавчі та нормативно-правові акти, нормативно-методична документація міжнародних установ та організацій у сфері антивірусного захисту, офіційні звіти Міністерства транспорту та зв'язку України, дані звітності та прогнози розвитку інформаційної безпеки.

Наукова новизна роботи полягає в наступному.

Запропоновано комплексне рішення, щодо вдосконалення антивірусного захисту верверів на всіх рівнях. Наведено методи аналізу інформаційного середовища: формування хронологічної послідовності типових впливів базових факторів середовища для їх подальшого аналізу й прогнозу.

Практична значущість результатів. У роботі проведено вивчення та розробка методів вдосконалення антивірусного захисту серверів. Отримані результати можуть бути використані у межах організацій для впровадження засобів інформаційної безпеки по етапах.

Достовірність дослідження. При реалізації проекту було вивчено значне число літературних джерел, посібників і статей, що охоплюють обрану тематику. Повний їх список представлений в кінці проекту.

Структура роботи. Структуру роботу складають вступ, чотири розділи які включають в себе 9 підрозділів, висновки, список використаної літератури яка складає 34 найменування, додатки. Загальний обсяг роботи складає 72 сторінки.

РОЗДІЛ 1
ТЕОРЕТИЧНІ АСПЕКТИ ЗАХИСТУ СЕРВЕРІВ

1.1 Система захисту серверів

Проблема безпеки веб-додатків в останнє десятиліття обговорюється всюди [2-10]. Із зростанням популярності інтернет-сервісів це питання виросло в стійкий тренд, який диктує ринку свої умови. Під його впливом змінюються галузеві стандарти, на ринок виходять нові гравці, а існуючі розширюють компетенції і випускають нові рішення.

На сьогоднішній день висока вартість інформації, що обробляється в процессингу веб-додатків, в сукупності із загрозою злому збільшує ризики інформаційної безпеки компаній. В цих умовах постає закономірне питання: що зробити для захисту веб-додатків?

Контрзаходи можна впроваджувати на двох етапах життя додатку – розробки і експлуатації. На етапі розробки – це різні інструменти тестування безпеки: статичний, динамічний, інтерактивний аналіз. Якщо говорити про безпеку програми, що вже експлуатується, то тут пропонується використовувати накладені засоби захисту – системи запобігання вторгнень, міжмережеві екрани наступного покоління (Next Generation Firewall, скорочено NGFW), а також засоби фільтрації трафіку прикладного рівня, спеціально орієнтовані на веб-додатки (Web Application Firewall, скорочено – WAF). Застосування Web Application Firewall традиційно вважається найбільш ефективним підходом до захисту веб-ресурсів [3]. Одним з основоположних чинників тут служить вузькоспеціалізована розробка.

WAF може бути реалізований як хмарний сервіс, агент на веб-сервері або спеціалізований залізний або віртуальний пристрій. Розвиток ринку WAF поки складається так, що хмарний сервіс затребуваний в середньому і малому бізнесі, а для великого бізнесу зазвичай купується окремий пристрій. WAF як модуль веб-сервера так і залишився, по суті, на початку розробки і більше підходить для початківців, ніж для бізнес-завдань.

Однак незважаючи на націленість виробників WAF на захищеність вузькопрофільної області, кожен з них має своє бачення і підхід до захисту приватних інфраструктур. Тим самим грамотний підбір засобу захисту є вирішальним моментом, адже від того, наскільки добре підходить засіб для конкретної інфраструктури, залежить його кінцева ефективність.

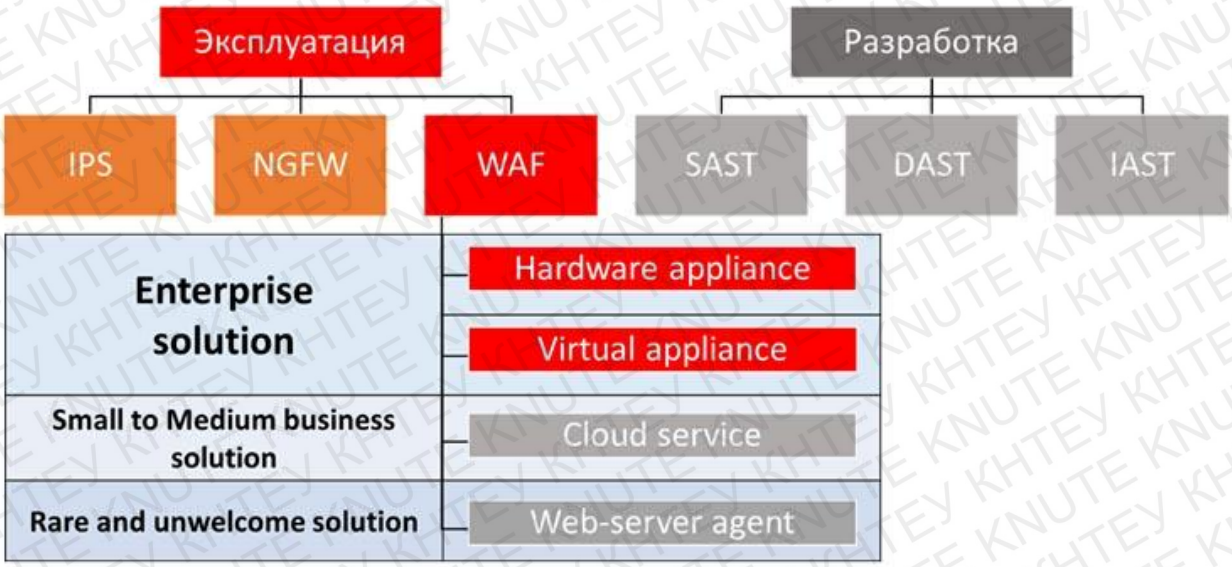

Рис. 1.1. Доступні методи захисту [4]

Класичне розміщення WAF в мережі – в режимі зворотного проксі-сервера, перед захищеними веб-серверами. В залежності від виробника можуть підтримуватися та інші режими роботи – наприклад, прозорий проксі-сервер, міст або навіть пасивний режим, коли продукт працює з реплікацією трафіку.

Після установки WAF і пуску продуктивного трафіку відразу ж починає роботу основний компонент захисту – машинне навчання, в ході якого складається еталонна модель комунікації з об'єктом захисту, і таким чином формується «білий» список допустимих ідентифікаторів доступу. На даний момент у веб-додатках використовуються три типи ідентифікатора доступу:

- HTTP параметри (в уявленнях типу: Raw, XML, JSON);
- ідентифікатор ресурсу (URL, URN);
- ідентифікатор сесії (cookie).

Завдання WAF полягає у визначенні допустимих значень ідентифікаторів для веб-додатку. З певних значень згодом буде складатися еталонна (позитивна) модель. Включення конкретних значень ідентифікатора у модель здійснюється на основі застосування математико-статистичного алгоритму, який за допомогою вибірки продуктивного трафіку оцінює ці значення як допустимі.

Коли всі ресурси веб-додатку додані у позитивну модель, адміністратор системи повинен переконатися у відсутності значущої кількості хибно-позитивних спрацьовувань і переключити систему в режим блокування.

Крім машинного навчання у набір функцій WAF зазвичай входять такі типові механізми захисту:

- валідація протоколу;
- сигнатурний аналіз;
- захист від ін'єкцій і XSS;
- можливість створення власних правил захисту;
- DDoS-захист;
- інтеграція з репутаційними і фрод-сервісами;
- інтеграція з іншими пристроями в ландшафті ІБ компанії [11].

Пріоритетом для виробника WAF є сфокусованість власних дослідницьких центрів на генерації оновлень політик безпеки для своїх

пристроїв з урахуванням актуальних загроз веб-додатків. Так з'являються, наприклад, сигнатури атак, властиві для конкретних веб-фреймворків та систем контролю вмісту або власні механізми захисту від XSS і SQL-ін'єкцій.

1.2 Аналіз аналогів програмного забезпечення захисту серверу

Першим значущим драйвером ринку WAF стало оновлення стандарту безпеки індустрії платіжних карт PCI DSS в частині вимоги 6.6 у 2008 році. Внаслідок поновлення у організацій з'явилася альтернатива при забезпеченні інформаційної безпеки своїх веб-додатків: організація може сама вибирати – або підтримувати бізнес-процеси з аналізу захищеності веб-додатків, або встановлювати WAF. Новий варіант стандарту і визначив перші ключові вимоги для систем захисту веб-додатків [12]. Виконання даних вимог досі актуально для розробників, так як стандарт особливо важливий у фінансовій та банківській сферах.

Загальні вимоги до сучасного Web Application Firewall:

- системні компоненти WAF повинні відповідати вимогам PCI DSS;
- можливість реагування на загрози, описані в OWASP Top Ten;
- інспектування запитів та відповідей у відповідності з політикою безпеки, журналювання подій;
- запобігання витоку даних – інспекція відповідей сервера на наявність критичних даних;
- застосування як позитивної, так і негативної моделі безпеки;
- інспектування всього вмісту веб-сторінок, включаючи HTML, DHTML і CSS, а також нижчих протоколів доставки вмісту (HTTP/HTTPS);
- інспектування повідомлень веб-сервісу, якщо веб-сервіс підключено до інтернету (SOAP, XML);

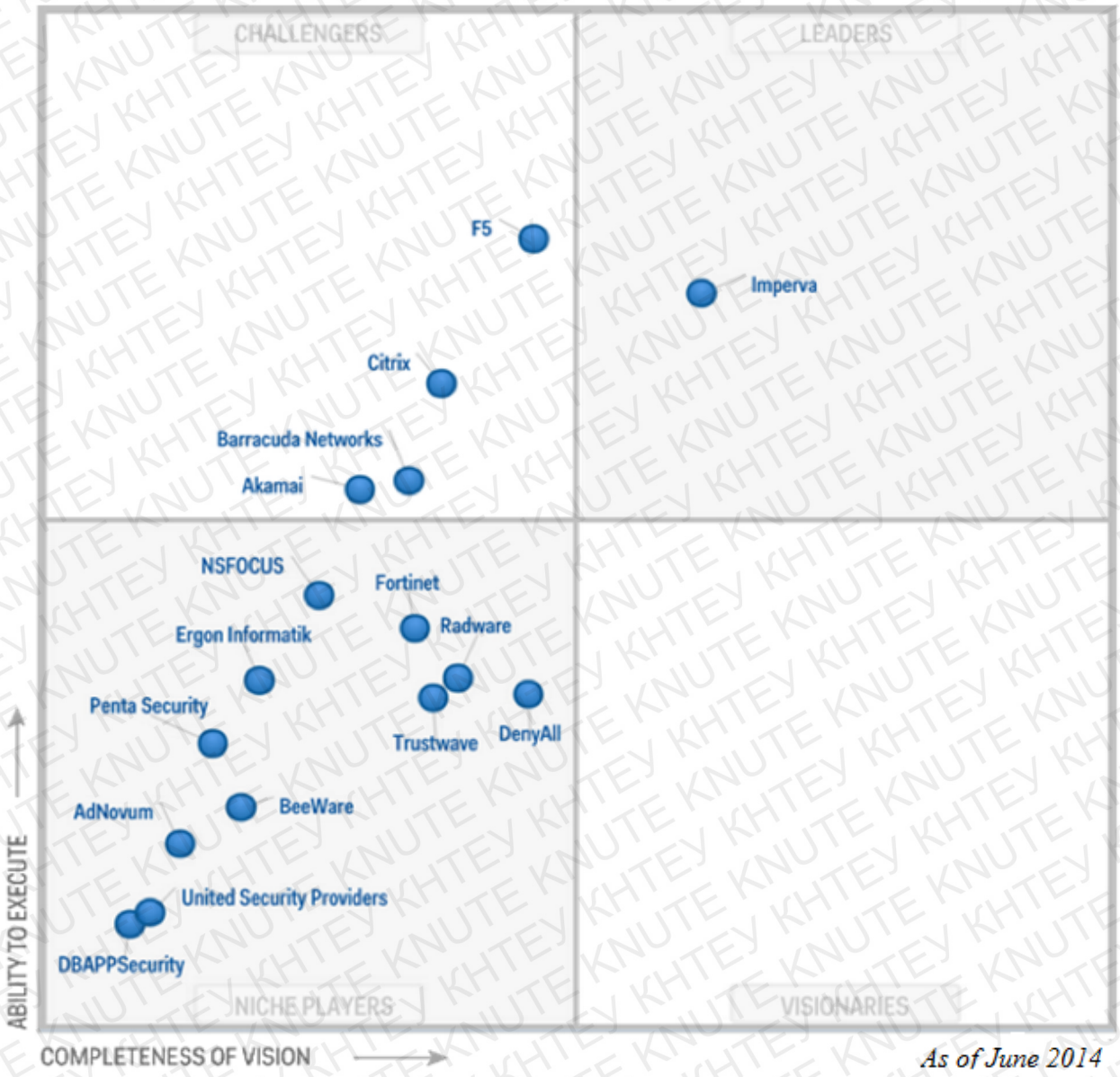

Рис. 1.2. Магічний квадрант Gartner по WAF, 2014 рік

Але незважаючи на настільки значний імпульс до розвитку ринку WAF ще в 2008 році, перший бюлетень компанії Gartner вийшов тільки в 2014 році, чим ознаменував новий виток конкурентної боротьби на ринку. У 2018 році вийшов звіт, який дозволив порівнювати досягнення і позиції компаній відносно один одного у часі. У дослідження Gartner потрапили 16 найбільш значних продуктів розглянутого ринку. Результат дослідження представлений у вигляді так званого магічного квадранта, який є двонаправленою діаграмою. По осі абсцис розташована шкала повноти

бачення, а по осі ординат – шкала можливостей. Сукупність показників визначає позицію в квадранті кожного з виробників щодо інших учасників на ринку [13].

Рис. 1.3. Магічний квадрант Gartner по WAF, 2018 рік [3]

В першу чергу варто звернути увагу на компанії F5 і Imperva, які є історичними лідерами даного ринку і показують кращі результати продажів на глобальному ринку. Далі йдуть Citrix і Barracuda Networks, які найбільш

близькі до лідерів ринку і мають достатній потенціал для майбутнього зростання (який, на жаль, вони не змогли реалізувати в 2014 році, практично не змінюючи позиції, згідно звіту Gartner за 2018 рік).

Великі ресурси компаній Imperva і F5 дозволяють їм першим розробляти нові технології і завойовувати популярність серед професійної спільноти. Нагромаджений ними передовий досвід позитивно впливає на весь ринок WAF в цілому.

1.3 Проблематика та постановка завдання дослідження

Ринок Web Application Firewall незважаючи на свою довголітню історію все ще знаходиться на етапі раннього розвитку. Про це свідчать такі факти, як: велика кількість гравців, у яких не виходить розвинути канали продажів, велика частка зв'язкових продажів, активність невеликих компаній і стартапів [15]. Пов'язано це в першу чергу з тим, що останнє десятиріччя проблематика захисту веб-додатків була повністю віддана розробникам, вважалося, що ніхто крім розробника не зможе усунути ті самі помилки, які і призводять до появи вразливостей.

Такий підхід мав успіх, поки організація обслуговувала тільки один або два веб-додатки і могла закривати витрати щодо забезпечення процесу аналізу вихідного коду для кожного з додатків. Але сьогодні великий бізнес містить десятки і навіть сотні веб-додатків, з яких тільки менша частина є власною розробкою. За таких обставин говорити про повноцінний захист, на жаль, не доводиться.

Незважаючи на масштабність присутніх на ринку програм для захисту серверів варто наголосити на недолік, що об’єднує все системи – це вартість. Всі вони платні. Також кожна окрема програма виконує обмежений перелік завдань та функцій. Тому щоб отримати систему захисту сервера варто здійснити розробку під окремі завдання, також перевагою є безкоштовність рішення.

1.4 Висновки до розділу

Розкрито поняття системи захисту серверів, визначено та проведено аналіз аналогів програмного забезпечення захисту серверу. Сформовано завдання дослідження.

Для захисту серверів використовуються засоби фільтрації трафіку прикладного рівня, спеціально орієнтовані на веб-додатки (Web Application Firewall, скорочено – WAF). Застосування Web Application Firewall традиційно вважається найбільш ефективним підходом до захисту веб-ресурсів. Одним з основоположних чинників тут служить вузькоспеціалізована розробка.

WAF може бути реалізований як хмарний сервіс, агент на веб-сервері або спеціалізований залізний або віртуальний пристрій. Розвиток ринку WAF поки складається так, що хмарний сервіс затребуваний в середньому і малому бізнесі, а для великого бізнесу зазвичай купується окремий пристрій. WAF як модуль веб-сервера так і залишився, по суті, на початку розробки і більше підходить для початківців, ніж для бізнес-завдань.

РОЗДІЛ 2
МЕТОДОЛОГІЧНІ АСПЕКТИ ЗАХИСТУ СЕРВЕРІВ

2.1 Методологія захисту серверів

Для захисту web-додатків необхідно використовувати засоби захисту від атак прикладного рівня, інакше кажучи Web Application Firewall. Використання WAF зарекомендувало себе за весь час від моменту створення даної технології, тому прийнято вважати, що використання WAF вважається найкращим способом захисту від комп'ютерних атак для захисту web-додатків [15].

Один з найважливіших факторів використання WAF-вузькоспеціалізований напрямок розробки:

- модуль CCS аналізує код додатків статичним методом (SAST) і передає гіпотези про уразливості модулю WAF;
- вбудований в WAF динамічний сканер додатків (DAST) відкидає помилкові спрацьовування або підтверджує гіпотези експлойта;
- знайдені вразливості закриваються «віртуальними патчами» до тих пір, поки програмісти готують виправлення;
- модуль WAF безперервно вивчає web-інфраструктуру на основі поведінкової статистики користувачів, динамічного і статичного аналізу, формуючи «норму» роботи;
- виявивши і проаналізувавши аномалії в масиві подій, WAF об'єднує їх в інцидент, визначає вектор атаки і блокує її [16].

«Вірутальні патчі» – технологія, розроблена компанією Wallarm, яка дозволяє закривати вразливість web-додатку на рівні WAF, попереджаючи про це адміністратора системи, поки програмісти не виправлять цю вразливість у вихідному коді програми [7].

Динамічний сканер коду (CCS), що працює в хмарі або ж на сервері системи, самостійно відсилає на ресурс, захищається запити, вивчаючи особливості побудови інфраструктури та web-додатку. З урахуванням виявлених вразливостей автоматично проводиться коригування роботи модуля WAF.

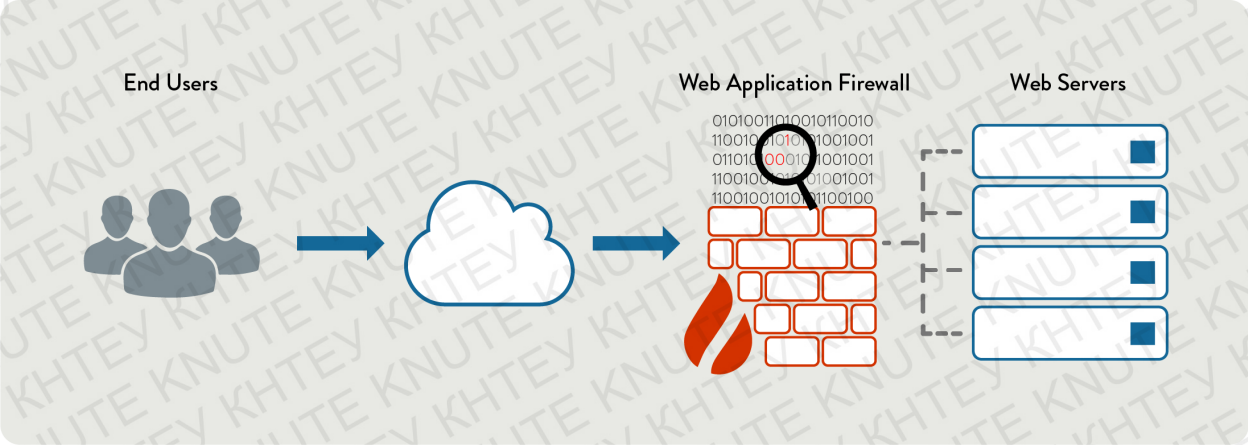

Рис. 2.1. Алгоритм роботи WAF [17]

Компоненти WAF розбиваються на наступні складові:

Проактивний фільтр-блокує більшу частину атак на web-додатку. Може працювати з великим потоком трафіку. Алгоритм самонавчання дозволяє уникнути помилкових спрацьовувань.

Система виявлення вразливостей виявляє існуючу помилку безпеки web-додатку. Клієнт інформується про виявлені уразливості і отримує не тільки опис проблеми, але і докладні рекомендації щодо його усунення.

Система Virtual Patching – захищає додатки сайту від невиправлених вразливостей, виявляючи і блокуючи спроби атак і вторгнень в режимі онлайн, що дозволяє не припиняти роботу сайту (не надається на етапі

тестової експлуатації).

На даній схемі рис. 2.1 бачимо, що WAF має свій власний web-сервер, призначений для взаємодії з користувачем через Особистий кабінет і проксирування трафіку на web-додаток.

Також можна помітити, що оновлення даних сигнатур атак і т. д. проходить через хмару, що дозволяє тримати систему в актуальному стані. Крім цього, система самонавчається, і при недоступності хмари Attack Killer може гарантувати максимальну ефективність при захисті від комп'ютерних атак.

Модуль CCS працює в сукупності з WAF, якщо розглядати їх спільну роботу, то можна сказати, що проактивне сканування WAF працює по моделі «чорного ящика», тобто сканер на стороні сервера намагається провести комп'ютерні атаки з боку зловмисника, якщо відомі способи атаки проходять успішно, то система негайно інформує персонал ІБ. Для усунення вразливостей в коді розробникам web-додатка необхідно час, коли в системі можна буде застосувати «віртуальний патчінг», що дозволяє приховати уразливість на рівні WAF, тобто дану уразливість не можна буде виявити і тим більше проводити атаку [18]. Вразливість в особистому кабінеті буде відображатися до тих пір, поки вона не буде закрита на рівні web-додатку.

Модуль CCS дозволяє сканувати web-додаток за моделлю «білого ящика», що дає можливість виявити більше всіляких загроз і вразливостей. Як правило, частина вразливостей, знайдених у вихідному коді, не зможуть експлуатуватися зловмисником, так як безліч вразливостей можуть бути закриті на рівні інших компонентів інформаційної структури (наприклад, firewall, web-сервер і т. д.). Знайдені уразливості передаються на відпрацювання пасивному сканеру WAF для проведення додаткової перевірки на можливість експлутації уразливості зловмисником. Якщо пасивний сканер зміг провести атаку успішно, то про це інформація передається відповідному персоналу [19].

Всі знайдені уразливості в системі мають докладний опис наслідків, які вони можуть нести рекомендації щодо усунення, а також приклади використання.

2.2 Математичне забезпечення захисту серверів

Розглянемо систему захисту серверів з математичної точки зору. Розділимо процес захисту на етапи.

1. Знаходження всіляких вирівнювань блоків (аналогічно вирівнювання рядків) для двох файлів.

Вирівнювання двох рядків виходить вставкою пробілів між деякими символами (можливо і на кінцях рядків) і розміщенням рядків один над одним так, що б кожен символ одного рядка опинився навпроти такого ж символу іншого рядка [10].

Завдання щодо вирівнювання рядків в загальному випадку має безліч рішень. Для зменшення алгоритмічної складності використовується вирівнювання блоків, оптимальне з точки зору редакційної відстані [10]. Таке вирівнювання може не бути оптимальним у сенсі запропонованої далі міри подібності, однак у більшості випадків дане вирівнювання близьке до оптимального.

В якості критерію близькості блоків у процесі вирівнювання використовується порогова функція: блоки вважаються схожими, якщо різниця між значеннями ентропії блоків менше порога Et_{max} і відношення розмірів блоків лежить в межах допустимого проміжку $[T_{lim}^{-1}; T_{lim}]$

Позначимо безліч схожих пар блоків як $M = \{m_1, m_2, \dots, m_{|M|}\}$, $|M|$ – розмірність множини M. Для кожної пари вирівняних блоків обчислимо ступінь їх відмінності $B_{dif}(m_i)$:

$$B_{dif}(m_i) = \left(1 - \frac{\min(T_1(m_i), T_2(m_i))}{\max(T_1(m_i), T_2(m_i))}\right) \cdot \frac{(Et_1(m_i), Et_2(m_i))}{Et_{max}} \quad (2.1)$$

де $T_1(m_i)$ і $T_2(m_i)$ – розміри, а $Et_1(m_i)$ і $Et_2(m_i)$ – значення ентропії і-ї пари схожих блоків для першого і другого файлу, відповідно.

B_{dif} лежить в діапазоні дійсних чисел [0, 1], де більшому значенню відповідає більша відмінність між блоками.

2. Обчислення ступеня подібності файлів для отриманого вирівнювання на основі виразу:

$$S_{struc}(f_1, f_2) = 1 - \frac{\sum_{i=1}^{|M|} \frac{1-B_{dif}(m_i)}{N_{near}(m_i)+1}}{\max(bc_1, bc_2)} \quad (2.2)$$

де S_{struc} – показник ступеня подібності пари файлів f_1 і f_2 на основі аналізу їх структури (приймає малі значення для однакових файлів і значення, близькі до 1 для різних); bc_1 і bc_2 – кількість блоків у першому і другому файлах відповідно;

$N_{near}(m_i)$ – редакційна відстань пари блоків m_i до найближчої з сусідніх пар (m_{i-1} та m_{i+1}): розраховується, як кількість дій (вставка, заміна, видалення), які потрібно зробити, перш ніж відбудеться вирівнювання наступної (або попередньої) пари.

В якості прикладу розглянемо рядки «abc» і «amnc». Редакційне відстань від першого вирівнювання («a») до наступного («c») становить дві одиниці: заміна елемента «b» на «m», вставка символу «n».

Для експериментальної перевірки розробленого способу порівняння виконуваних файлів було відібрано 1500 шкідливих об'єктів. Далі їх розібрали за групами згідно з вердиктом антивірусу Касперського:

в одну групу віднесені об'єкти, що мають однаковий вердикт. Для більшої точності отримані групи були проаналізовані вручну і з них видалені помилково класифіковані об'єкти. Підсумкова кількість файлів – 1008, груп – 163.

З метою аналізу різних способів порівняння була обрана бікубічна (bcubed) міра якості кластеризації [12]. Її переваги розглянуті в роботі [13].

Для кожного об'єкта розраховуються величини точності (precision) та повноти (recall). Для цього виділяються дві підмножини об'єктів: C_h – всі об'єкти, ступінь подібності яких з об'єктом o менше порога h, і L_h – всі об'єкти того ж класу, що і o:

$$Precision(o,h) = \frac{|C_h \cap L_h|}{|C_h|} \quad (2.3)$$

$$Recall(o,h) = \frac{|C_h \cap L_h|}{|L_h|} \quad (2.4)$$

Графічно поняття однорідності і закінченості представлені на рис. 2.2. Об'єкти, відстань від яких до об'єкта o менше порога h, обведені. Об'єкти одного класу показано однаковими геометричними формами.

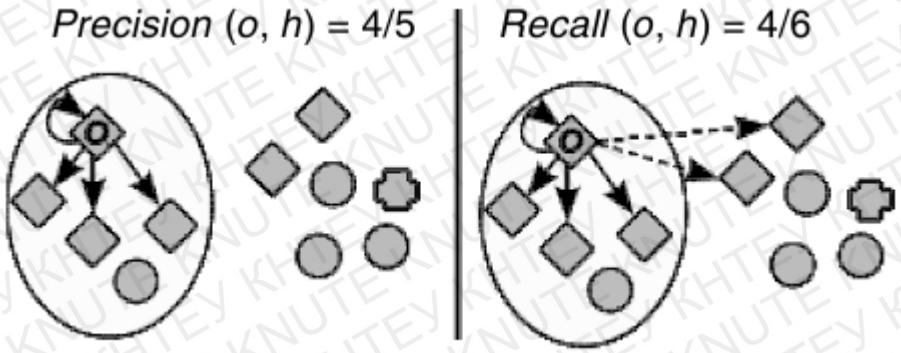

Рисунок 2.3 – Розрахунок однорідності й завершеності для бікубічної міри

Заходи однорідності й завершеності об'єкта o комбінуються в єдину F-міру, яка визначається як зважене середнє гармонійне точності і повноти.

$$F(0,h) = \frac{1}{\alpha \cdot \frac{1}{Precision(o,h)} + (1-\alpha) \cdot \frac{1}{Recall(o,h)}} \qquad (2.5)$$

де коефіцієнт $\alpha \in [0,1]$ дозволяє зважувати співвідношення точності і повноти.

F-міра приймає великі значення, якщо точність і повнота порівняння для даного об'єкта о максимальні, і малі в іншому випадку.

Бікубічний захід для безлічі всіх об'єктів $\{o_1, o_2, ..., o_k\}$ та фіксованого значення порога h розраховується як середнє значення F-заходів кожного з k об'єктів:

$$Q(h) = \frac{\sum_{i=1}^{k} F(o_i,h)}{k} \qquad (2.6)$$

Бікубічна міра приймає великі значення, якщо точність і повнота порівняння об'єктів максимальні, і малі в іншому випадку.

Бікубічний захід може розраховуватися для різних значень граничної відстані. Для порівняння різних функцій відстані для кожної з них визначимо поріг hQ-max, при якому значення бікубічного заходу максимально (Qmax).

Оскільки бікубічний захід являє собою середнє значення для вимірювань F-заходи і-го об'єкта, похибка вимірювань ΔQ для фіксованого порога розраховується наступним чином:

$$\Delta Q(h) = t_{i,k-1} \sqrt{\frac{\sum_{i=1}^{k} \left(F(o_i,h) - Q(h)\right)^2}{k \cdot (k-1)}} \qquad (2.7)$$

де $t_{\gamma,\, k-1}$ – значення коефіцієнта Стьюдента, для k – 1 ступенів свободи та довірчої імовірності $1 - \gamma$.

За допомогою бікубічного заходу порівняємо наступні способи для визначення ступеня подібності файлів:

– спосіб, що використовує контекстно-залежну хеш-функцію [7];

– спосіб, що використовує евклідову відстань між векторами, отриманими згідно роботи [8], і використовує в разі різної розмірності векторів початкові позиції;

– спосіб, що використовує блоки змінної довжини, запропонований в даній роботі.

Для оцінки похибки варто використовувати довірчу ймовірність 0,99.

2.3. Висновки до розділу 2

У межах другого розділу висвітлено методологічні аспекти захисту серверів. Здійснено опис технології захисту та розкрито принципи математичного забезпечення захисту серверів.

Антивірусний механізм використовує велику кількість методів для виявлення шкідливого ПЗ. Антивірусні програми містять в собі велику базу зразків вірусів, які необхідно виявити під час сканування. Кожен зразок може або визначати унікальний шкідливий код або, що є більш поширеним, описувати ціле сімейство вірусів. Основна особливість визначення вірусів шляхом порівнювання зі зразками в тому, що антивірусна програма може визначити тільки добре відомий вірус, в той час як нові загрози можуть бути не виявлені.

РОЗДІЛ 3
ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАХИСТУ СЕРВЕРУ

3.1 Опис програмного забезпечення та засобів проектування

Мовою реалізації програмного забезпечення для захисту серверів обрано php у якості бази даних обрано SQL.

PHP – це широко використовувана мова сценаріїв загального призначення з відкритим вихідним кодом.

Простіше кажучи, PHP – це мова програмування, спеціально розроблена для написання web-додатків (сценаріїв), які працюють на веб-сервері.

Абревіатура PHP означає “Hypertext Preprocessor (Препроцесор Гіпертексту)". Синтаксис мови бере початок від C, Java і Perl. PHP досить простий для вивчення. Перевагою PHP є надання веб-розробникам можливості швидкого створення динамічно генерованих web-сторінок.

У якості засобу реалізації БД була обрана система MS SQL Server.

Microsoft SQL Server– це потужна і надійна система управління даними, що забезпечує безліч функцій, захист даних і високу продуктивність для впроваджених програм-клієнтів, «легких» веб-додатків і локальних сховищ даних. SQL Server призначений для спрощеного розгортання та швидкого створення прототипів; його можна отримати безкоштовно і вільно поширювати разом з додатками.

SQL Server спрощує розгортання, управління і оптимізацію даних підприємства та аналітичних програм. Як платформа керування даними підприємства, він надає єдину консоль управління, яка дозволяє адміністраторам даних, що знаходяться в будь-якому місці організації, відстежувати, керувати та налаштовувати всі бази даних і пов'язані служби по всьому підприємству. Він надає інфраструктуру управління, яка може бути легко запрограмована за допомогою SQL Management Objects (SMO), дозволяючи користувачам переробляти і розширювати їх середу управління і незалежним постачальникам програмних продуктів (ISV) створювати додаткові інструменти і функціональність для подальшого розширення можливостей, що поставляються за замовчуванням.

Розробимо базу даних структура якої представлена на рис. 3.1.

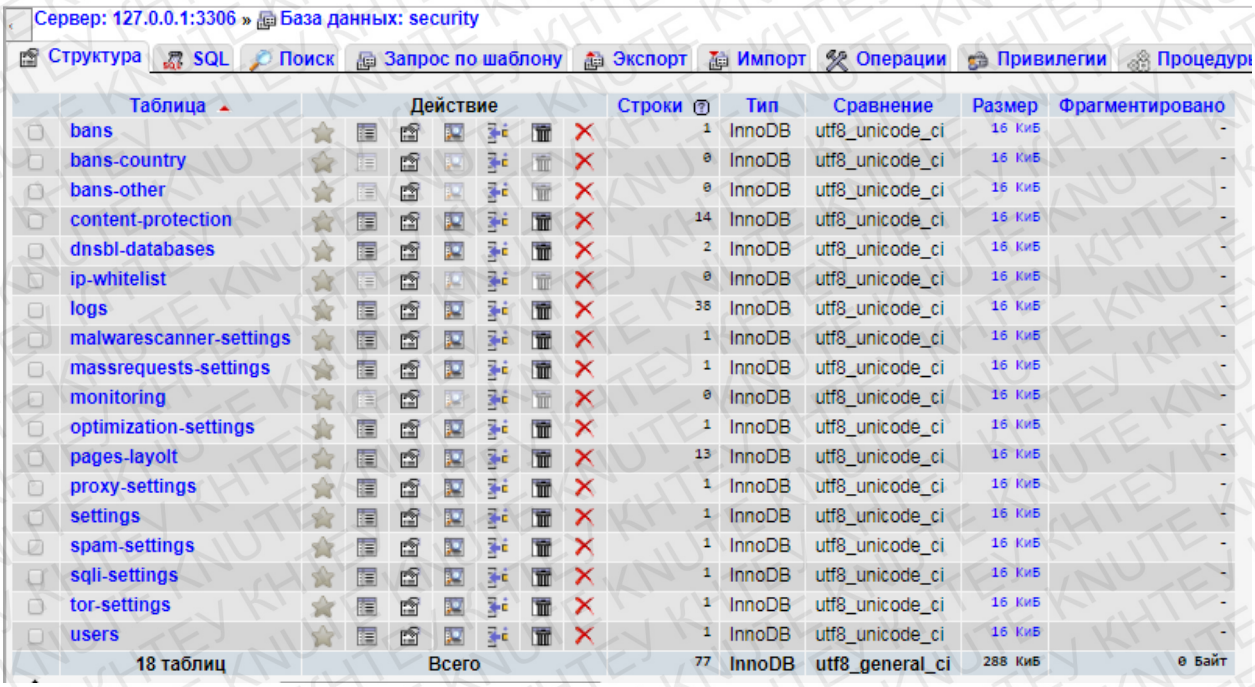

Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
bans		1	InnoDB	utf8_unicode_ci	16 КиБ	-
bans-country		0	InnoDB	utf8_unicode_ci	16 КиБ	-
bans-other		0	InnoDB	utf8_unicode_ci	16 КиБ	-
content-protection		14	InnoDB	utf8_unicode_ci	16 КиБ	-
dnsbl-databases		2	InnoDB	utf8_unicode_ci	16 КиБ	-
ip-whitelist		0	InnoDB	utf8_unicode_ci	16 КиБ	-
logs		38	InnoDB	utf8_unicode_ci	16 КиБ	-
malwarescanner-settings		1	InnoDB	utf8_unicode_ci	16 КиБ	-
massrequests-settings		1	InnoDB	utf8_unicode_ci	16 КиБ	-
monitoring		0	InnoDB	utf8_unicode_ci	16 КиБ	-
optimization-settings		1	InnoDB	utf8_unicode_ci	16 КиБ	-
pages-layolt		13	InnoDB	utf8_unicode_ci	16 КиБ	-
proxy-settings		1	InnoDB	utf8_unicode_ci	16 КиБ	-
settings		1	InnoDB	utf8_unicode_ci	16 КиБ	-
spam-settings		1	InnoDB	utf8_unicode_ci	16 КиБ	-
sqli-settings		1	InnoDB	utf8_unicode_ci	16 КиБ	-
tor-settings		1	InnoDB	utf8_unicode_ci	16 КиБ	-
users		1	InnoDB	utf8_unicode_ci	16 КиБ	-
18 таблиц	Всего	77	InnoDB	utf8_general_ci	288 КиБ	0 Байт

Рис. 3.1. Структура бази даних

Схема роботи системи для захисту серверів складається з кількох етапів, основним з яких є запит до бази даних. SQL є інструментом, призначеним для обробки і читання даних, що містяться в комп'ютерній базі даних. SQL – це скорочена назва структурованої мови запитів (Structured

Query Language). Як випливає з назви, SQL є мовою програмування, яка застосовується для організації взаємодії користувача з базою даних. Насправді SQL працює тільки з базами даних одного певного типу, званих реляційними. На рис. 3.2 зображена схема роботи SQL.

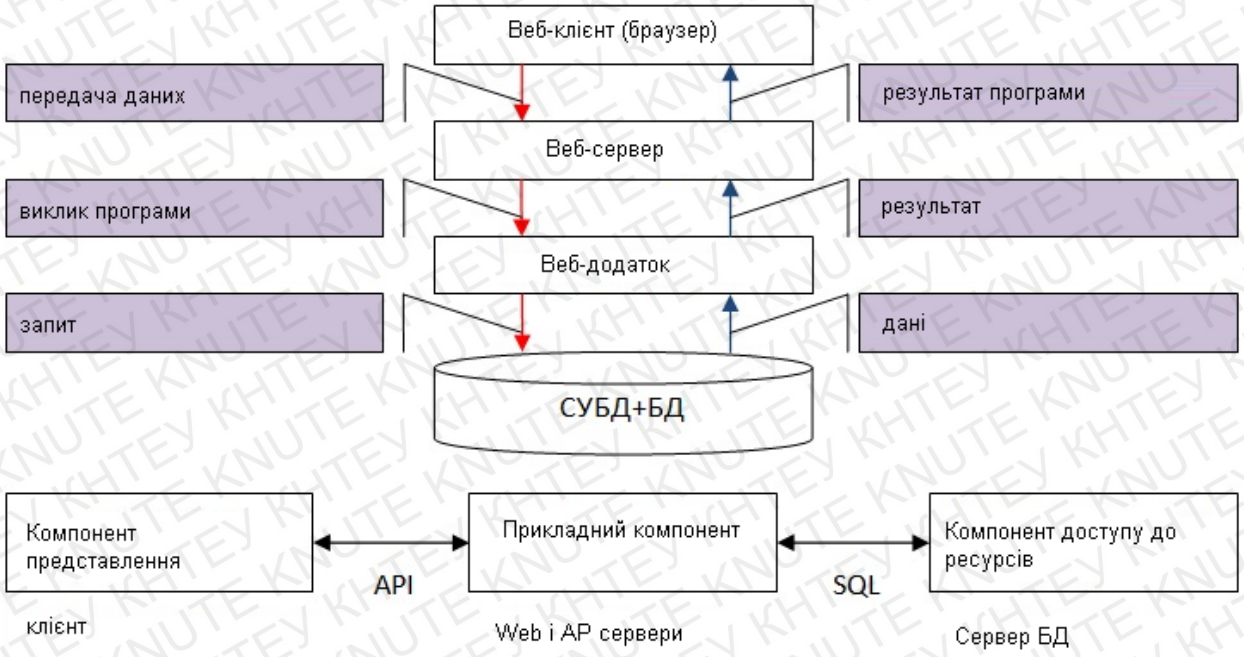

Рис. 3.2. Схема роботи SQL для доступу до БД

Рис. 3.3. Основні компоненти СУБД

Сам по собі SQL не є ні системою управління базами даних, ні окремим програмним продуктом. SQL – це невід'ємна частина СУБД, інструмент, за допомогою якого здійснюється зв'язок користувача з нею. На рис. 3.3. зображена структурна схема типової СУБД, компоненти якої з'єднуються в єдине ціле за допомогою SQL (свого роду "клею").

Ядро бази даних є серцевиною СУБД; воно відповідає за фізичне структурування і запис даних на диск, а також за фізичне читання даних з диска. Крім того, воно приймає SQL-запити від інших компонентів СУБД (таких як генератор форм, генератор звітів або модуль формування інтерактивних запитів), від користувача додатків і навіть від інших обчислювальних систем.

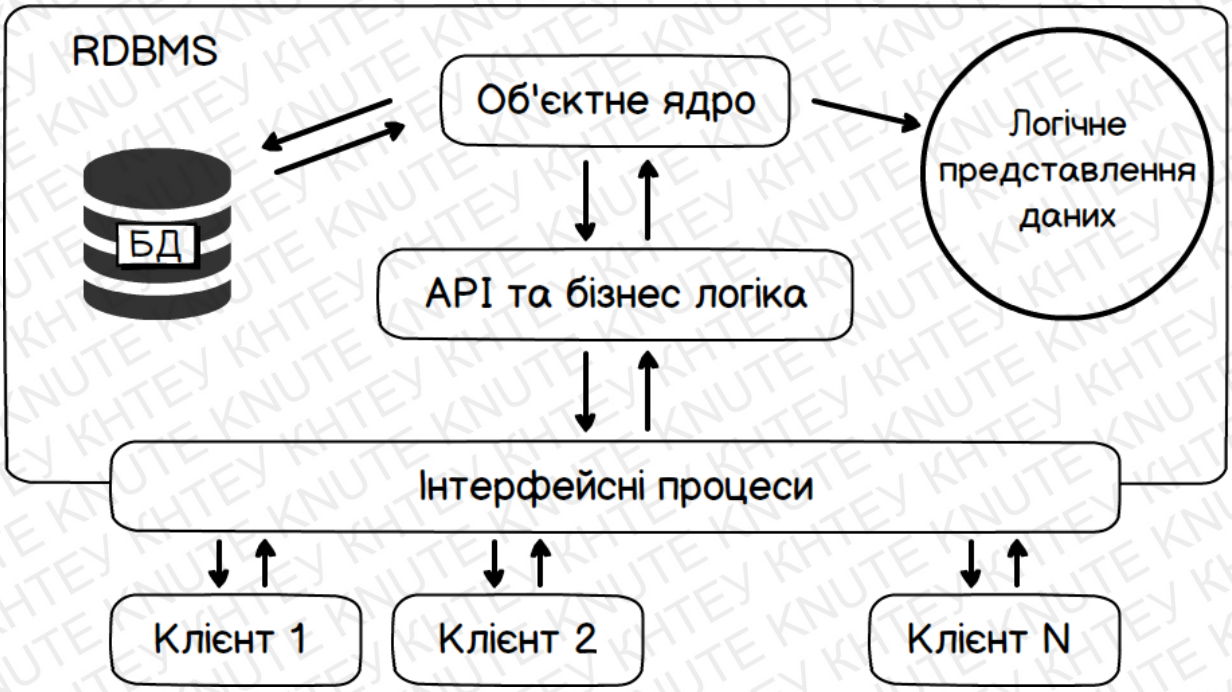

Рис. 3.4. Структурна схема доступу системи до БД

Розглянемо структурну схему доступу до системи СУБД (рис 3.4). На рисунку представлені основні компоненти взаємодії при виконанні доступу системи до бази даних.

3.2 Розробка програмного забезпечення захисту серверу

У процесі роботи з системою працюватимуть два актори (активних об'єкта), один з яких буде займатися роботою з даними адміністратор, а другий – власне користуватися системою, це користувач або власник. Передбачається наявність таких варіантів використання системи: розробка структури сайту, формування наповнення сайту, яка передбачає введення, модифікацію і перегляд даних.

Рис. 3.5. Діаграмма варіантів використання системи захисту серверів

Головними вимогами до системи, що розробляється є скорочення часу розробки структури сайту, формування оптимального інтерфейсу, автоматизації обробки та зберігання даних, роботи сайту.

Для моделювання процесу виконання операцій в мові UML використовуються так звані діаграми діяльності (рис. 3.6). Кожен стан на цій

діаграмі відповідає виконанню деякої елементарної операції, а перехід в наступний стан спрацьовує тільки при завершенні операції в попередньому стані.

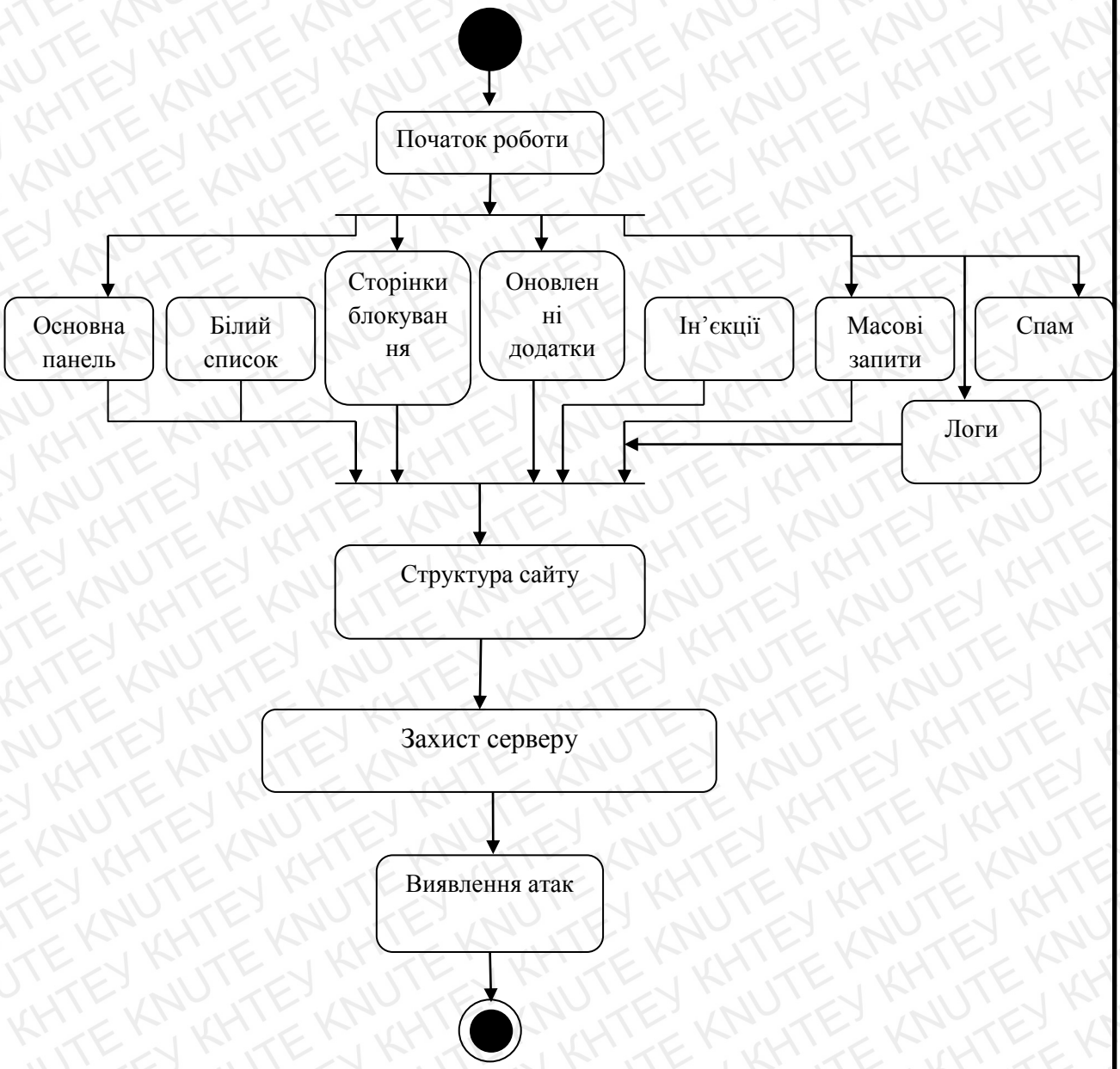

Рис. 3.6. Діаграма діяльності системи захисту серверів

Основою системи захисту серверів передбачено:

- управління безпекою сайту;
- виведення статистики атак;
- блокування країн на сайті;

– блокування IP на сайті;

– управління логами;

– управління доступом до даних.

Побудуємо діаграму прецедентів програмного продукту (рис. 3.7), що розробляється.

Рис. 3.7. Діаграма прецедентів системи захисту серверів

При моделюванні поведінки проектованої або аналізованої системи виникає необхідність не тільки уявити вироблені зміни її станів, а й деталізувати особливості алгоритмічної і логічної реалізації виконуваних системою операцій. Для цієї мети могли використовуватися блок-схеми, які акцентують увагу на послідовності виконання певних дій або елементарних операцій, які в сукупності призводять до отримання бажаного результату. Діаграми діяльності в мові UML використовуються для моделювання процесу виконання операцій. Кожен стан на діаграмі діяльності відповідає

виконанню деякої елементарної операції, а перехід в наступний стан спрацьовує тільки при завершенні цієї операції в попередньому стані.

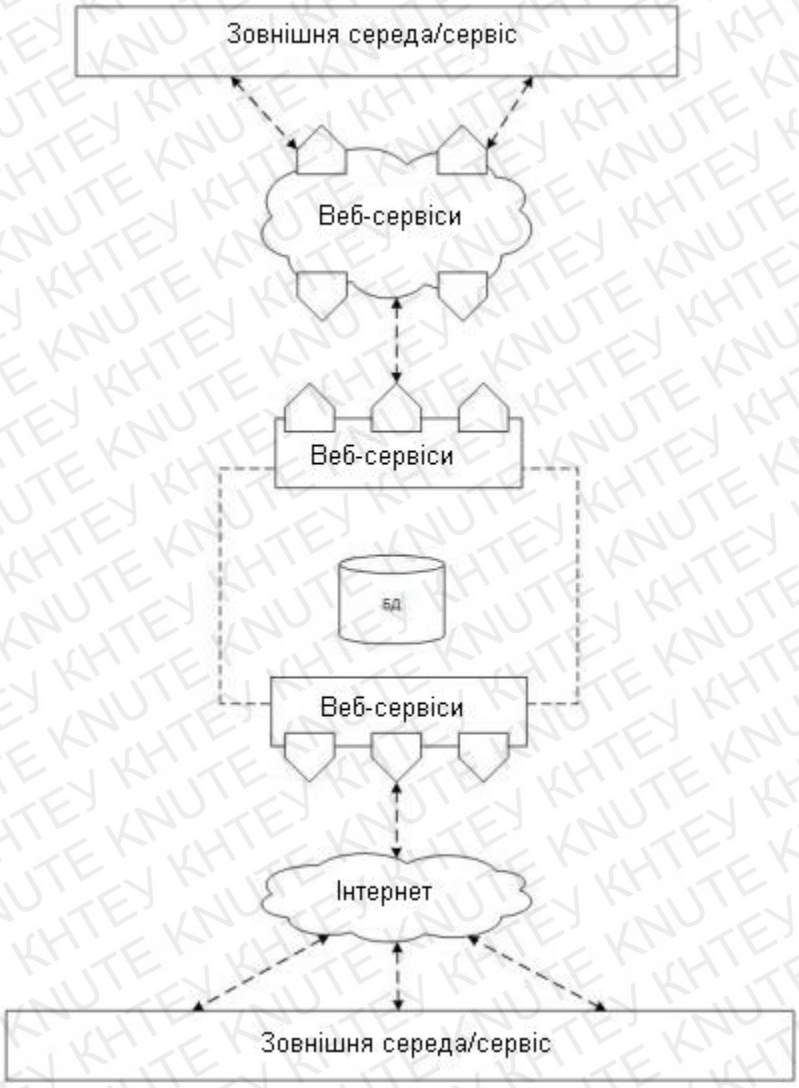

Рис. 3.8. Інформаційна взаємодія із зовнішніми системами і сервісами

Графічно діаграма діяльності представляється у формі графа діяльності, вершинами якого є стани дії, а дугами переходи від одного стану дії до іншого. На діаграмі діяльності відображається логіка або послідовність переходу від однієї діяльності до іншої, при цьому увага фіксується на результаті діяльності. Сам же результат може призвести до зміни стану системи або поверненню деякого значення.

Діаграми діяльності доцільно використовувати для опису складних базових сценаріїв варіантів використання, а також зв'язки між сценаріями, де простежується паралельність виконуваних операцій.

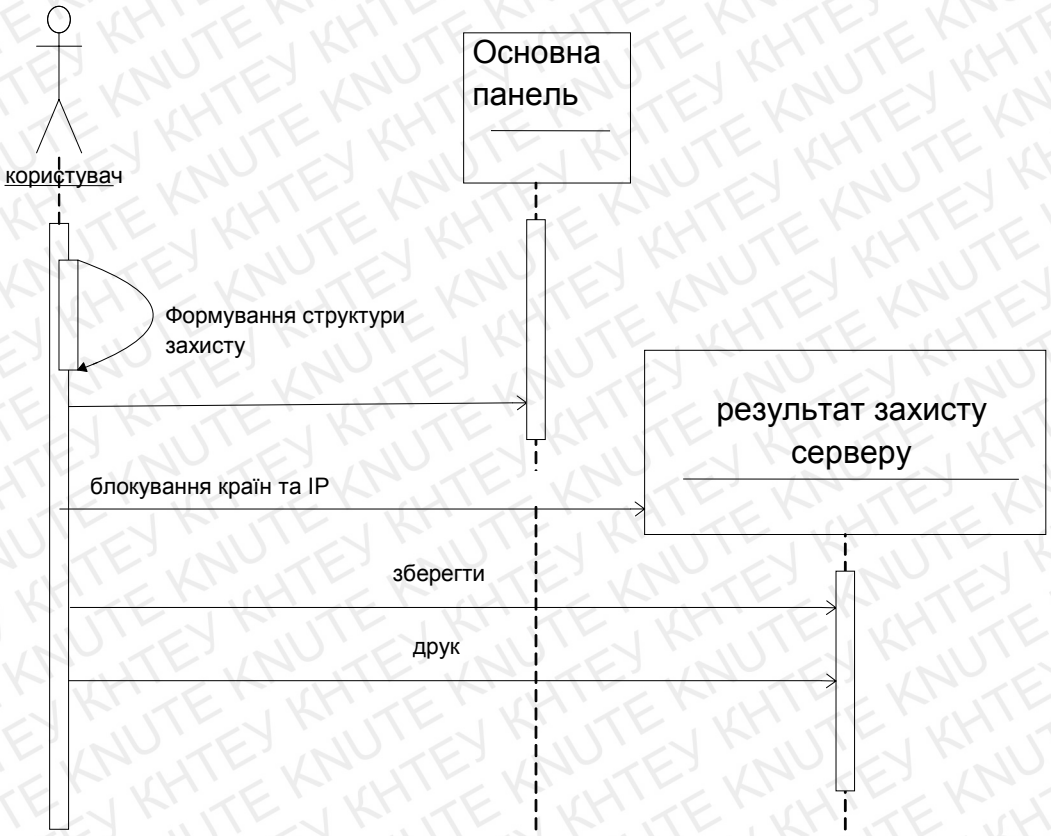

Рис. 3.9. Діаграма послідовності – захист серверів

На рис. 3.9 приведена діаграма діяльності користувача. Відповідно до діаграми станів користувачеві доступні переходи в стани «Основна панель» і «Захист». У стані «Захист» користувачеві надається можливість ввести початкове наповнення, сформувати його.

У кожному з цих станів формується список даних і надається можливість переходу до вибору кожного окремого типу даних.

У мові UML взаємодія елементів розглядається в інформаційному аспекті їх комунікації, тобто взаємодіючі об'єкти обмінюються між собою деякою інформацією. При цьому інформація приймає форму закінчених повідомлень. Іншими словами, хоча повідомлення і має інформаційний зміст,

воно набуває додаткової властивості надавати спрямований вплив на свого одержувача. Для моделювання взаємодії об'єктів в мові UML використовуються відповідні діаграми послідовності. Говорячи про ці діаграми, мають на увазі два аспекти взаємодії. По-перше, взаємодії об'єктів можна розглядати в часі, і тоді для представлення тимчасових особливостей передачі і прийому повідомлень між об'єктами використовується діаграма послідовності. По-друге, можна розглядати структурні особливості взаємодії об'єктів.

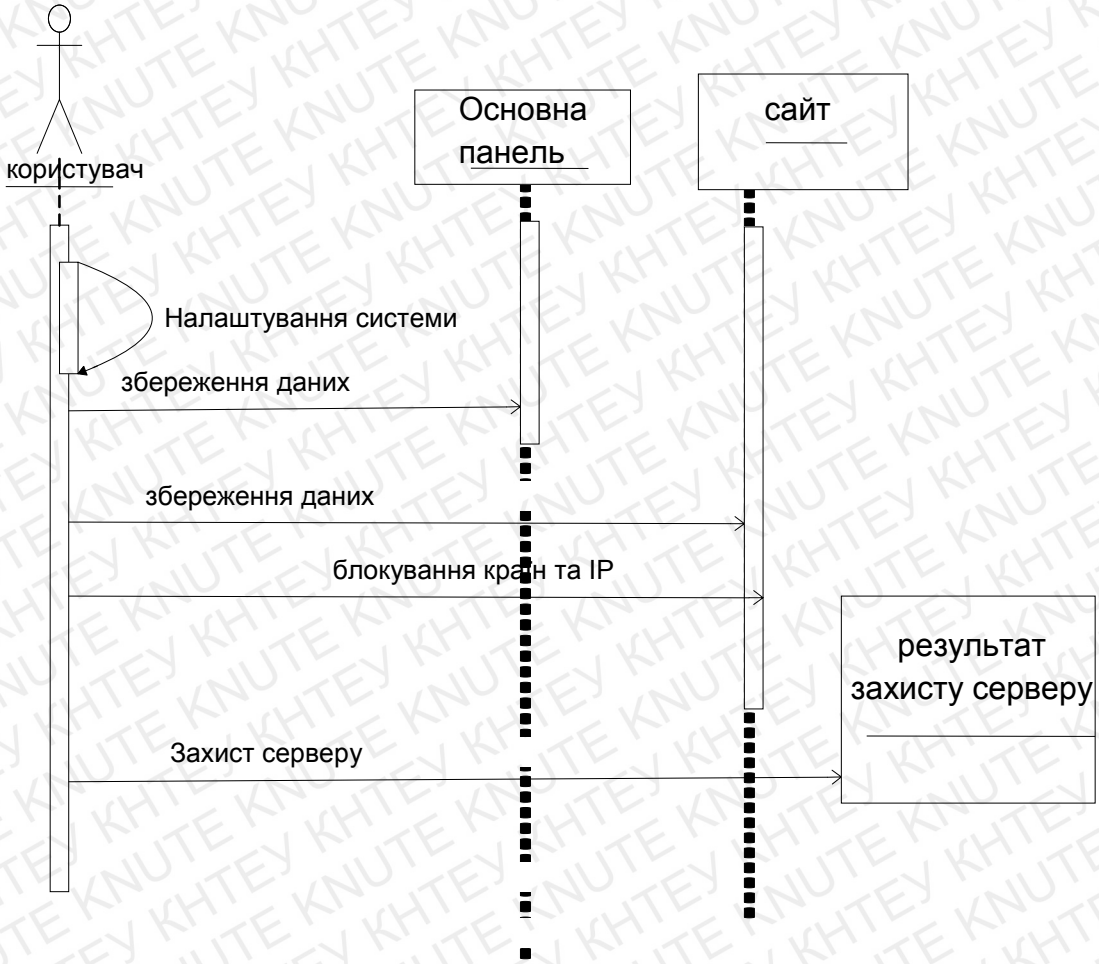

Рис. 3.10. Діаграма послідовності – наповнення значень захисту

Діаграма компонентів, на відміну від раніше розглянутих діаграм, описує особливості фізичного представлення системи.

Вона дозволяє визначити архітектуру системи, що розробляється, встановивши залежності між програмними компонентами, в ролі яких може виступати вихідний і виконуваний код. Основними графічними елементами діаграми компонентів є компоненти, інтерфейси і залежності між ними.

Діаграма компонентів розробляється для наступних цілей:

– візуалізації загальної структури вихідного коду програмної системи;

– специфікації виконуваного варіанту програмної системи;

– забезпечення багаторазового використання окремих фрагментів програмного коду.

У розробці діаграм компонентів беруть участь як системні аналітики та архітектори, так і програмісти. Діаграма компонентів забезпечує узгоджений перехід від логічного представлення до конкретної реалізації проекту у формі програмного коду. Одні компоненти можуть існувати тільки на етапі компіляції програмного коду, інші на етапі його виконання. Діаграма компонентів відображає загальні залежності між компонентами, розглядаючи останні як класифікаторів.

3.3. Висновки до розділу 3

У межах третього розділу здійснено опис програмних засобів для проектування сиситеми захисту серверів, описано структуру бази даних для системи захисту серверів та проведено розробку програмного забезпечення для захисту серверів.

Мовою реалізації програмного забезпечення для захисту серверів обрано php у якості бази даних обрано SQL.

РОЗДІЛ 4
АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗАХИСТУ СЕРВЕРІВ

4.1. Верифікація результатів

Здійснимо аналіз ефективності роботи системи захисту серверів (Security) та порівняємо його з відомими аналогами OVE, CAS, RW та MSAT.

У якості параметрів досліджуються оцінки експертів щодо ефективності та параметри наведені у таблиці 4.1. Далі на основі отриманих даних проводиться математичний аналіз та здійснюється побудова діаграм та графіків для детального розуміння ефективності розробки.

Таблиця 4.1

Параметри звернень

Параметр	Опис	Можливі значення
$Е_ф$	Ефективність	5-надвисока,..., 0-низька
ІН	інцидент	1-інцидент, 0-запит
$О_б$	обсяги даних на захист	10-макс,..., 0-мінім.
$К_д$	категорії даних	1,2,3…n - категорія
$П_п$	пріоритет	1,2,3…n - пріоритет
$Р_{кін}$	кінцевий рівень вирішення	1,2,3,4 - рівні
$С_{звер}$	стан	0-відкритий, 1-закритий
К	коефіцієнт якості захисту	5-надвисокий, ..., 0-низький
$О_{клієн}$	оцінка клієнта	0,1,2,3,4,5

У табл. 4.1 показано параметри звернень, що надходять до захисту серверів, їх опис та можливі значення.

Дані складемо таблицю значень отриманих при дослідження розробленої системи та аналогів, що прийняті до розгляду.

Таблиця 4.2

Значення показників діяльності програмного забезпечення для захисту серверів (Security)

Параметр	Значення									
$E_ф$	5	5	3	5	3	5	4	5	3	5
ІН	1	0	1	1	0	1	1	1	0	1
$O_б$	8	10	9	8	9	10	7	8	9	10
$K_д$	1	2	3	1	3	2	4	1	3	2
$П_п$	2	5	8	8	2	1	5	2	3	5
$Р_{кін}$	1	2	3	1	3	2	4	1	3	2
$С_{звер}$	1	0	1	1	0	1	1	1	0	1
К	5	5	4	5	3	5	5	5	5	4
$О_{клієн}$	5	5	4	4	5	5	5	5	5	5

Таблиця 4.3

Значення показників діяльності OVE

Параметр	Значення									
$E_ф$	2	2	3	2	3	2	3	2	3	2
ІН	0	0	1	0	0	1	1	0	0	1
$O_б$	3	4	3	4	4	2	7	8	3	2
$K_д$	1	2	2	1	3	2	2	1	3	2
$П_п$	2	5	2	2	2	1	5	2	3	2
$Р_{кін}$	1	2	3	1	3	2	2	1	3	2
$С_{звер}$	0	0	1	0	0	1	0	1	0	1
К	2	2	4	2	3	2	2	2	2	4
$О_{клієн}$	2	2	4	4	2	2	2	2	3	3

Таблиця 4.4

Значення показників діяльності системи CAS

Параметр	Значення									
$E_ф$	3	2	3	2	3	2	3	3	3	2
ІН	1	0	1	0	0	1	0	0	0	1
$О_б$	5	4	3	4	4	2	7	2	3	5
$К_д$	2	2	2	1	3	2	2	1	3	2
$П_п$	5	0	2	2	5	1	5	2	3	2
$Р_{кін}$	3	2	3	1	2	2	2	1	2	2
$С_{звер}$	1	0	1	0	1	1	0	1	1	1
К	4	2	4	3	3	2	3	2	4	3
$О_{клієн}$	4	2	4	4	2	2	4	2	3	4

Таблиця 4.5

Значення показників діяльності системи RW

Параметр	Значення									
$E_ф$	4	2	3	2	3	4	3	3	3	4
ІН	1	0	1	0	0	1	1	0	1	1
$О_б$	2	4	3	4	4	6	7	2	3	5
$К_д$	4	2	4	1	3	2	2	1	3	3
$П_п$	5	5	2	2	5	1	2	2	3	2
$Р_{кін}$	2	2	3	1	2	2	2	1	4	4
$С_{звер}$	1	0	0	0	1	1	0	1	1	0
К	3	4	4	3	3	2	3	4	2	3
$О_{клієн}$	2	2	4	4	2	2	3	2	3	2

Таблиця 4.6

Значення показників діяльності системи MSAT

Параметр	Значення									
$E_ф$	4	4	4	5	3	4	4	4	3	4
ІН	1	0	1	1	0	1	1	1	0	1
$О_б$	8	5	9	8	9	5	7	8	9	5
$К_д$	2	2	3	1	3	2	4	1	3	2

$П_п$	3	5	6	6	2	1	5	3	3	5
$Р_{кін}$	1	2	3	1	3	2	4	1	3	2
$С_{звер}$	1	0	1	1	0	1	1	1	0	1
К	4	5	4	5	3	5	5	4	5	4
$О_{клієн}$	4	5	4	4	4	4	4	5	4	4

Відповідно до наведених даних здійснюємо аналіз отриманих даних.

Таким чином, на основі проведеного математичного аналізу ефективності роботи Security та OVE, CAS, RW та MSAT, та використовуючи шкалу Чеддока, можна зробити висновок, що найбільш впливають на ефективність такі чинники:

Для системи Security:

- Ефективність ($Е_ф$),
- Пріоритет ($П_п$)
- Категорія даних ($К_д$)
- Коефіцієнт якості захисту (К)
- Оцінка клієнта ($О_{клієн}$)

Для системи OVE:

- Ефективність ($Е_ф$),
- Обсяги даних ($О_б$)
- Категорія даних ($К_д$)
- Кінцевий рівень вирішення ($Р_{кін}$)

Для системи CAS:

- Обсяги даних ($О_б$)
- Категорія даних ($К_д$)
- Кінцевий рівень вирішення ($Р_{кін}$)

Для системи RW:

- Категорія даних ($К_д$)

Для системи MSAT:

– Інцидент (ІН),
– Пріоритет ($П_п$)
– Категорія даних ($К_д$)
– Обсяги даних ($О_б$)
– Кінцевий рівень вирішення ($Р_{кін}$)
– Стан ($С_{звер}$)
– Коефіцієнт якості захисту (К)
– Оцінка клієнта ($О_{клієн}$)

Далі до розгляду приймаємо дві системи, розроблену у рамках даної роботи Security та найбільшого конкурента MSAT, так як інші програми виявилися не конкурентоспроможними за результатами попереднього аналізу.

На виході цього етапу, згідно до формули:

$$KPI = \{\bigcup_{w=1}^{v} KPI_w\} = \{KPI_1, KPI_2, \dots, KPI_v\} \qquad (4.1)$$

де

$$KPI_w \subseteq KPI, (w = \overline{1, v}) \qquad (4.2)$$

v – кількість ключових показників ефективності.

Для системи Security при w = 5 отримуємо множину ключових показників ефективності KPI :

$$KPI = \{Е_ф, П_п, К_д, К, О_{клієн}\} = \begin{Bmatrix} ефективність, пріоритет, категорія даних, \\ коефіцієнт якості захисту, оцінка клієнта \end{Bmatrix}$$

Для системи MSAT при w = 8 отримуємо множину ключових показників ефективності KPI :

$$KPI = \{ІН, П_п, К_д, О_б, К, О_{клієн}, С_{звер}, Р_{кін}\}$$

$$= \begin{Bmatrix} інцидент, пріоритет, категорія даних, обсяги даних, \\ коефіцієнт якості захисту, оцінка клієнта, стан, кінцевий рівень звернень \end{Bmatrix}$$

Далі проведемо графічне порівняння двох систем за основними показниками для виявлення найбільш ефективної та досконалої системи для захисту серверів.

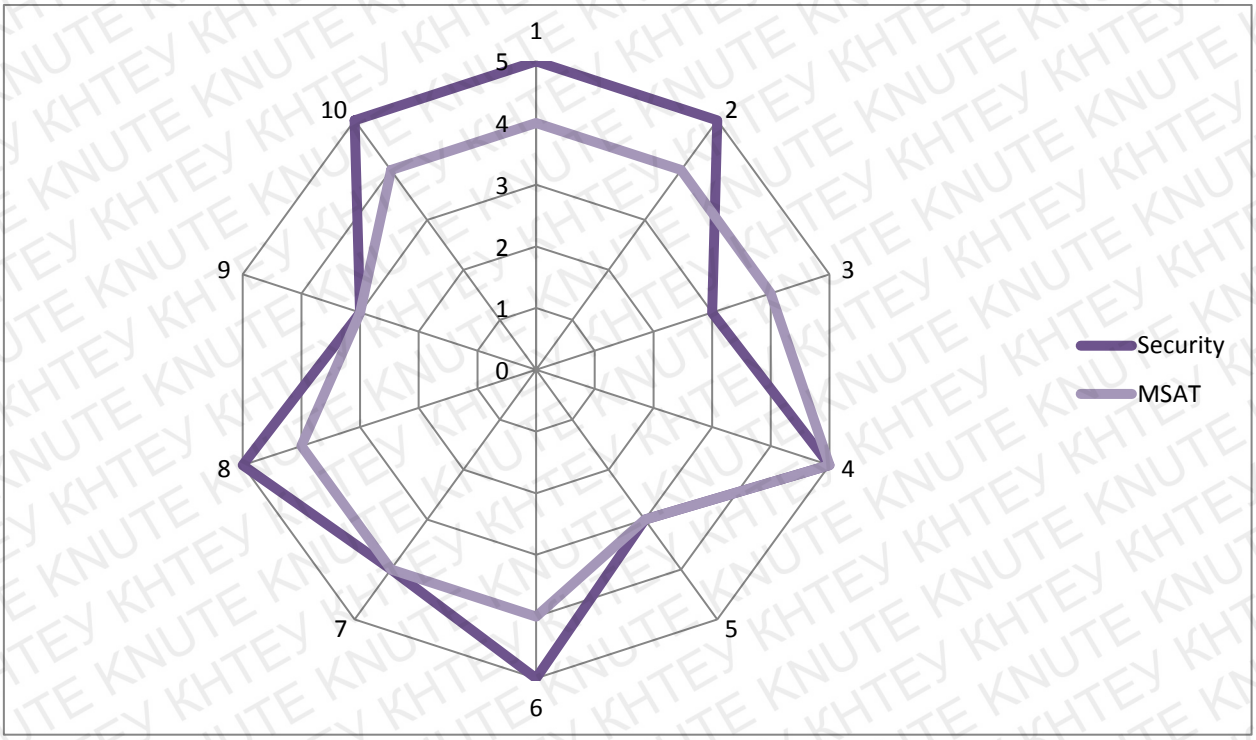

Рис. 4.1. Графік ефективності систем

Графік оцінки клієнта наведемо на рис. 4.2

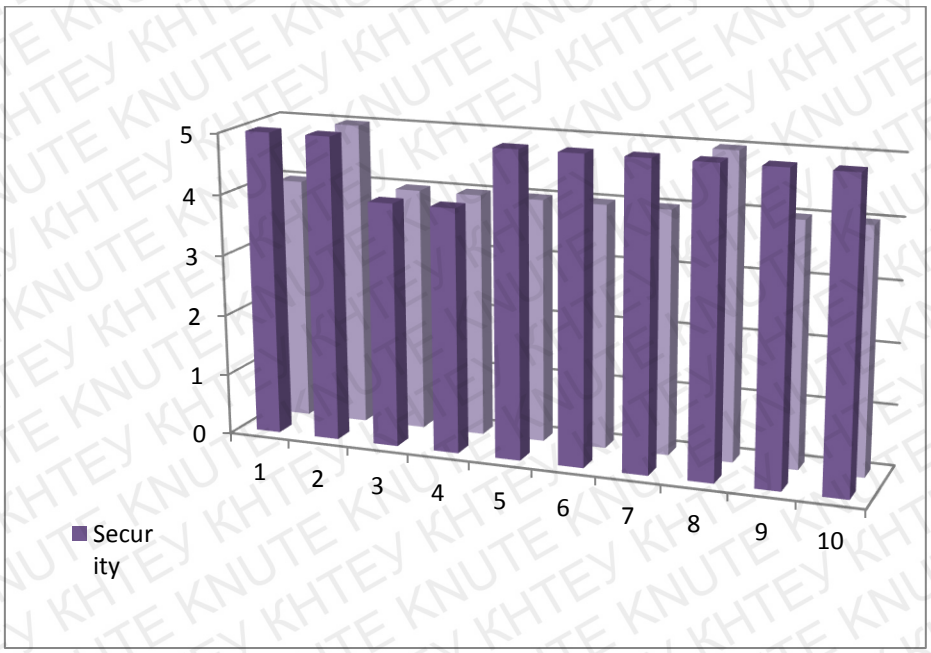

Рис.4.2 . Графік оцінка клієнта систем

Таким чином, згідно до проведеного дослідження варто відзначити, що розроблена системи захисту серверів за всіма показниками перевершує свої конкурентів/аналогів, що говорить про високу якість розробки та можливість впровадження методу в реальну роботу за потребою.

4.2. Висновки до розділу 4

У межах четвертого розділу здійснено тестування розробленого програмного забезпечення для захисту серверів. Під час тестування збоїв та недоліків у роботі програмного забезпечення для захисту серверів не виявлено, що говорить про високу якість розробки та можливість впровадження за потребою.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

У межах даної дипломної роботи здійснено розробку програмного забезпечення для захисту серверів. За допомогою якого користувачі даної розробки отримають підвищення якості сервісу, швидкість реакції на атаки, можливість контролю доступу в автоматичному режимі, спрощення роботи.

Пріоритетом для виробника WAF є сфокусованість власних дослідницьких центрів на генерації оновлень політик безпеки для своїх пристроїв з урахуванням актуальних загроз веб-додатків. Так з'являються, наприклад, сигнатури атак, властиві для конкретних веб-фреймворків та систем контролю вмісту або власні механізми захисту від XSS і SQL-ін'єкцій.

Мовою реалізації програмного забезпечення для захисту серверів обрано php у якості бази даних обрано SQL.

Запропоновано комплексне рішення, щодо вдосконалення антивірусного захисту верверів на всіх рівнях. Наведено методи аналізу інформаційного середовища: формування хронологічної послідовності типових впливів базових факторів середовища для їх подальшого аналізу й прогнозу.

У роботі проведено вивчення та розробка методів вдосконалення антивірусного захисту серверів. Отримані результати можуть бути використані у межах організацій для впровадження засобів інформаційної безпеки по етапах.

Під час тестування збоїв та недоліків у роботі програмного забезпечення для захисту серверів не виявлено, що говорить про високу якість розробки та можливість впровадження за потребою.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Киберугрозы и информационная безопасность в корпоративном секторе: тенденции в мире и в России» [Электронный ресурс] // Лаборатория Касперского: (дата обращения: 03.11.2019).
2. Мельников Д.А. Информационная безопасность открытых систем: учебник. М.: Флинта; Наука, 2013. – 448 с.
3. Результаты глобального исследования по вопросам обеспечения информационной безопасности, перспективы на 2018 год [Электронный ресурс] // PWCURL: (дата обращения: 08.10.2019).
4. Attack Killer [Электронный ресурс]. (дата обращения: 08.11.2019).
5. LMS Industry User Research Report [Электронный ресурс] // Capterra Reseach: (дата обращения: 08.11.2019).
6. Top LMS Software [Электронный ресурс] // Capterra: (дата обращения: 03.11.2019).
7. Microsoft SQL Server 2012. Руководство для начинающих, Душан Петкович, 816 стр, ISBN 978-5-9775-0854-4; БХВ-Петербург 2013.
8. Maedche A., Staab S. Tutorial on Ontologies: Representation, Engineering, Learning and Application / / ISWC'2002. – 234 p.
9. Farquhar A., Fikes R., Rice J. The Ontolingua server: A tool for collaborative ontology construction / / International Journal of Human-Computer Studies, 46 (6). – 1997. – pages 707-728.
10. Musen, M. Domain Ontologies in Software Engineering: Use of Prot? G? with the EON Architecture / / Methods of Inform. in Medicine, 1998. – pages 540-55

11. OKBC: A Programmatic Foundation for Knowledge Base Interoperability. V. Chaudhri, A. Farquhar, R. Fikes P. Karp J. Rice / / Fifteenth National Conf. on Artificial Intelligence. AAAIPres / The MIT Press, Madison, 1998. – P. 600-607.

12. Creating Semantic Web Contents with Prot? G? -2000. N. Noy, M. Sintek, S. Decker, M. Crubezy, R. Fergerson, M. Musen / / IEEE Intelligent Systems, March / April pages 60-71, 2001.

13. OntoEdit: Collaborative ontology development for the Semantic Web. Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, D. Wenke / / In Proc. of the Inter. Semantic Web Conference (ISWC 2002), Sardinia, Italia, June 2002.

14. Bechhofer S., Horrocks I., Goble C., Stevens R. OilEd: A Reason-able Ontology Editor for the Semantic Web / / Joint German / Austrian conf. on Artificial Intelligence (KI'01). Lecture Notes in Artificial Intelligence LNAI 2174, Springer-Verlag, Berlin, pages.396-408, 2001.

15. Domingue J. Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web / / Proc. of the Eleventh Workshop on Knowledge Acquisition, Modeling and Management, KAW'98, Banff, Canada, 1998.

16. Motta E. Reusable Components for Knowledge Modelling / / Ph.D. Thesis. The Open University, 1997.

17. MacGregor R. Inside the LOOM classifier / / SIGART bulletin, Vol.3, No.2, pages 70-76, 1991.

18. Fernandez M, Gomez-Perez A., Pazos J. A Building a Chemical Ontology Using Methondology and the Ontology Design Environment / / IEEE Intelligent Systems, Jan. / Feb. pages 37-46, 1999.

19. Kifer M., Lausen G., Wu J. Logical Foundations of Object-Oriented and Frame-Based Languages / / Journal of the ACM, 1995.

20. Fernandez M., Gomez-Perez A., Juristo N. METHONTOLOGY: From Ontological Art Towards Ontological Engineering / / AAAI-97 Spring Symposium on Ontological Engineering, Stanford University, 1997.

21. Arpirez JC, Corcho O, Fernandez-Lopez M, Gomez-Perez A. WebODE: a scalable workbench for ontological engineering / / First Intern. Conf. Knowledge Capture (KCAP2001). Victoria. Canada, Oct. 2001.

22. The CommonKADS Methodology. G. Schreiber, H. Akkermans, A. Anjewierden, R. Hoog, N. Shadbolt, W. Van de Velde, B. Wielinga / / Knowledge engineering and management. MIT press, Massachusetts, 1999.

23. Franconi E., Ng G. The i.com tool for intelligent conceptual modeling / / 7th Intl. Workshop on Knowledge Representation meets Databases, KRDB'00, Berlin, Germany, 2000.

24. Noy N., Musen M. SMART: Automated Support for Ontology Merging and Alignment / / Stanford Medical Informatics, Stanford Univ., 1999.

25. Noy N., Musen M. The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping / / Stanford Medical Informatics, Stanford Univ., 2003.

26. McGuinness D., Fikes R., Rice J., Wilder S. An environment for merging and testing large ontologies / / In Proc. of the Seventh Int. Conf., KR2000, Morgan Kaufmann Publishers, San Francisco, CA, 2000.

27. Dou D., McDermott D., Qi P., Ontology translation by ontology merging and automated reasoning / / EKAW'02 workshop on Ontologies for Multi-Agent Systems. SigEuenza, Spain, 2002.

28. Chalupsky H, Morph: A translation system for symbolic knowledge / / In: Cohn AG, Giunchiglia F., Selman B. (Eds.), Principles of Knowledge Representation and Reasoning // Proc. of the Seventh Intern. Conf. (KR2000). Morgan Kaufmann Publishers, San Francisco, CA, 2000.

29. Mena E., Illarramendi A., Kashyap V., Sheth A. OBSERVER: An approach for query processing in global information systems based on

interoperation across preexisting ontologies // Distributed and Parallel Databases, An International Journal, Vol.8, No.2 , 2000.

30. Stumme G., Medche A. FCA-Merge: Bottom-up merging of ontologies // 7th Int. Conf. on Artificial Intelligence, (IJCAI'01), Seattle, WA, 2001, P.225-230.

31. Mitra P., Wiederhold G., Decker S. A scalable framework for interoperation of information sources // The 1st Intern. Semantic Web Working Symp., SWWS'01, Stanford University, CA, 2001.

32. Vargas-Vera M., Motta E., Domingue J., Lanzoni M., Stutt A., Ciravegna F. MnM: Ontology-Driven Tool for Semantic Markup // European Conf. on Artificial Intelligence (ECAI 2002). In Proc. of the Workshop Semantic Authoring, Annotation & Knowledge Markup (SAAKM 2002). Lyon France, July 22-23, 2002.

33. Heflin J. Hendler J. A Portrait of the Semantic Web in Action // IEEE Intelligent Systems, March / April, pages 54-59, 2001.

34. Stoffel K., Taylor M., Hendler J. Efficient management of very large ontologies // American Association for Artificial Intelligence Conf., (AAAI97), Menlo Park, CA, pages 442-447, 1997.

ДОДАТКИ

Додаток А

Програма та методика тестування

Проведемо тестування розробленого програмного забезпечення для захисту серверів. Результати тестування наведемо на рис. Головне вікно програми наведено на рис. 1.

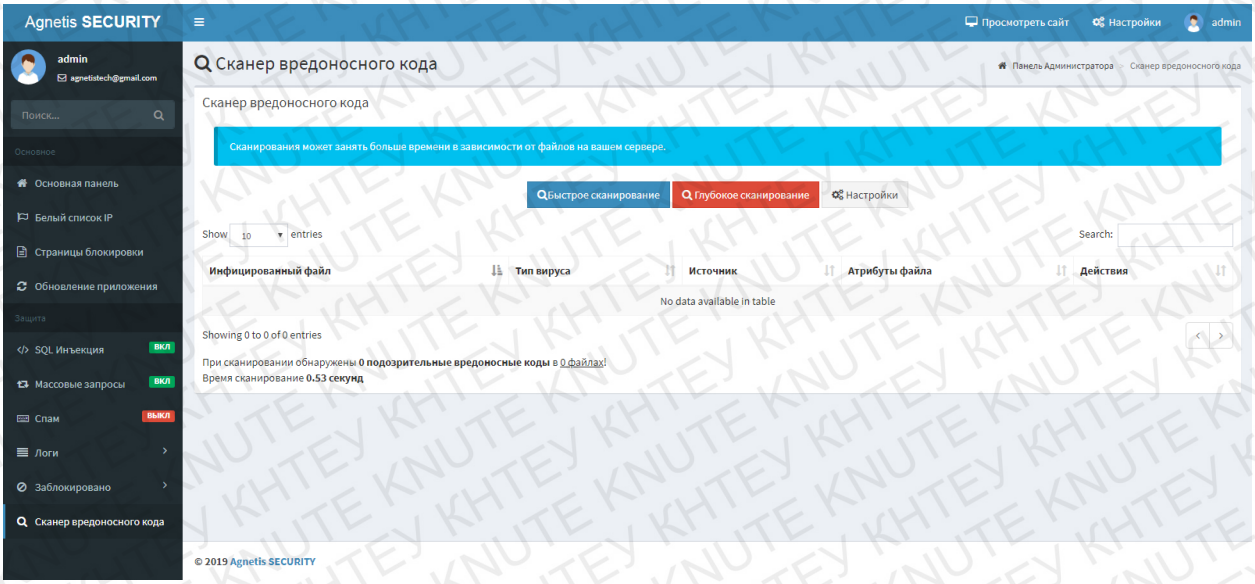

Рис. 1. Головна сторінка програмного забезпечення для захисту серверів

Головна сторінка програмного забезпечення для захисту серверів умовно розділена на два сектори, перший (ліворуч), це панель управління, яка складається з наступних активних посилань:

- основна панель;
- білий список IP;
- сторінки блокування;
- оновлення додатків;
- ін’єкція;
- масові запити;

– спам;

– логи.

Другий сектор – це поле виведення результату.

Налаштування моделів захисту здійснюється при переході на сторінку «Ін’єкції» (рис. 2)

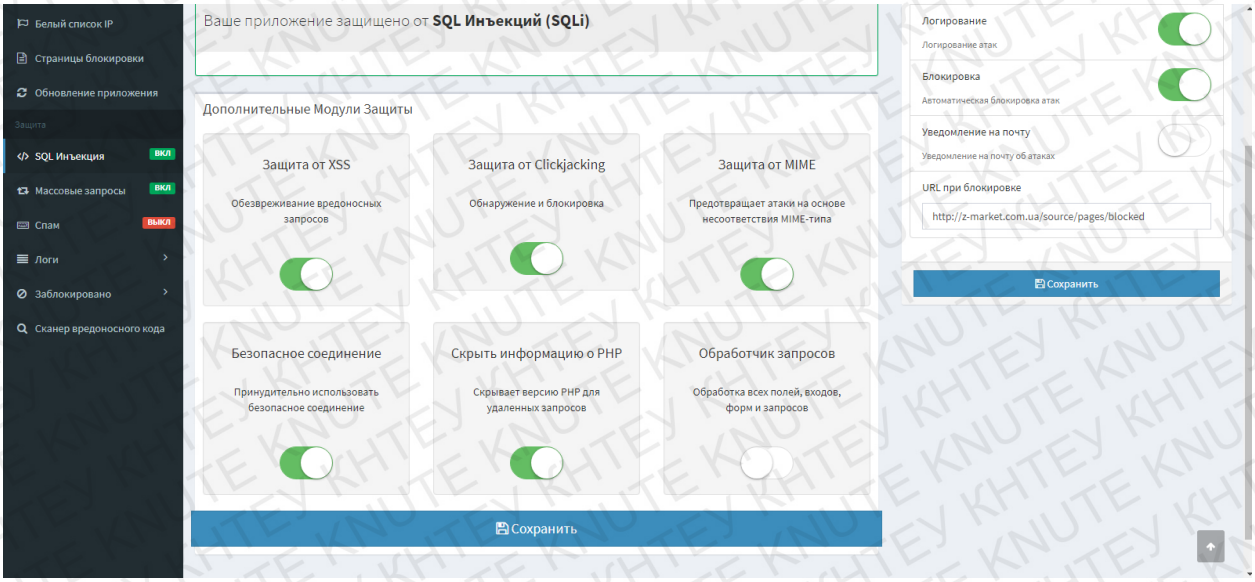

Рис. 2 . Сторінка «Ін’єкції»

На сторінці «Ін’єкції» користувач має можливість встановити параметри захисту за наступними показниками:

– захист від XSS;

– безпечне з’єднання;

– захсит від Clickjacking;

– сховати інформацію про PHP;

– захист від MIME;

– обробник запитів.

Налаштування основних параметрів системи здійснюється на сторінці «Налаштування» (рис. 3)

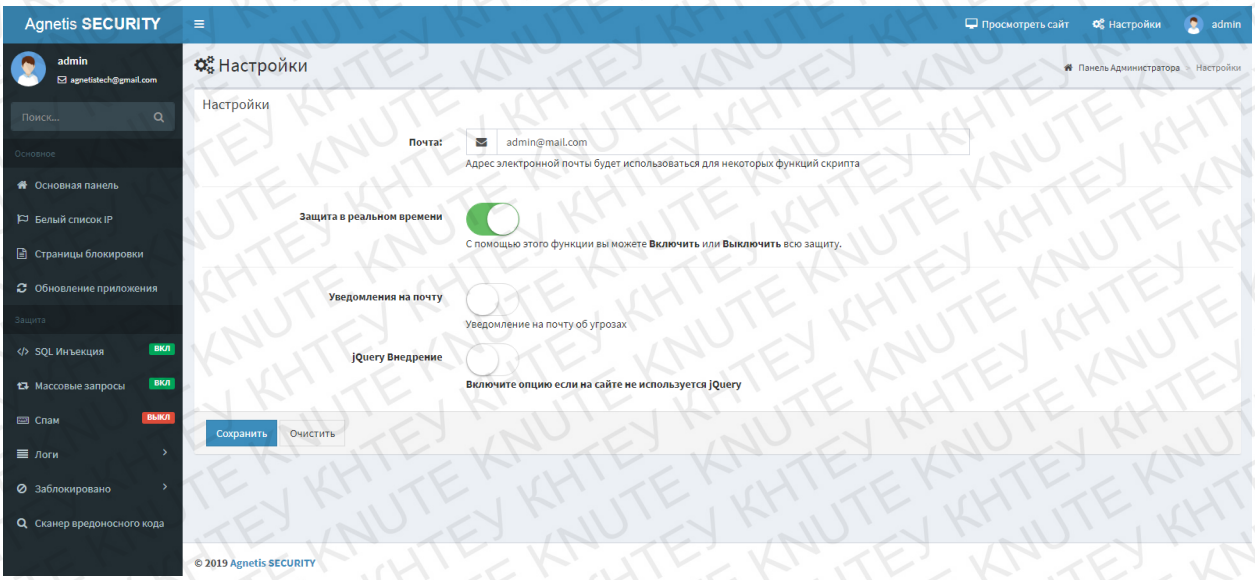

Рис. 3. Сторінка «Налаштування»

Блокування сторінок (рис. 4).

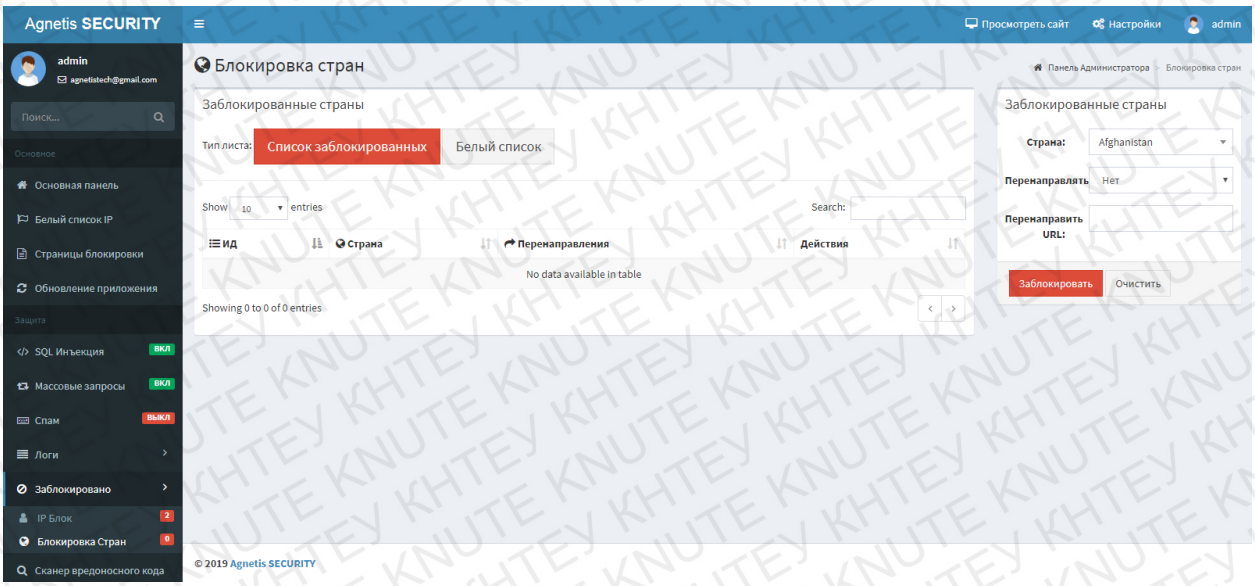

Рис. 4. Сторінка «Блокування країн»

На сторінці «Блокування країн» користувач має можливість здійснити блокування країн, які (на думку користувача) є найбільш небезпечними.

Також сторінкою передбачено виведення «білого» списку, тобто списку країн, які є небезпечними за замовчуванням.

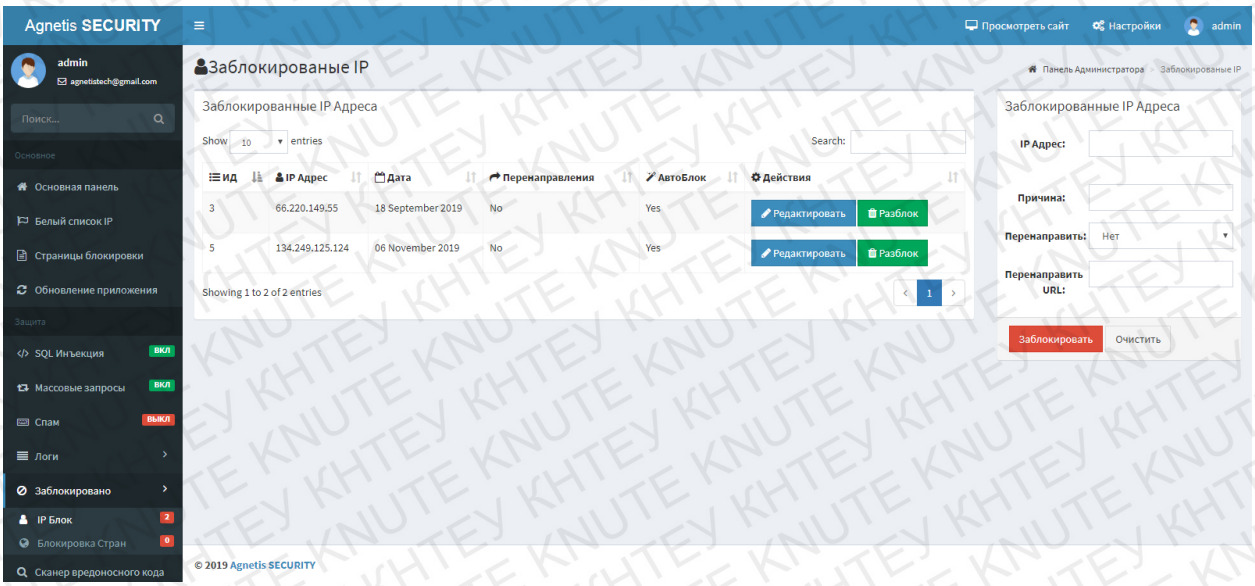

Рис. 5. Сторінка «Блокування IP»

На сторінці «Блокування IP» користувач має можливість здійснити блокування IP, які (на думку користувача) є найбільш небезпечними. Принцип блокування аналогічний попередньому.

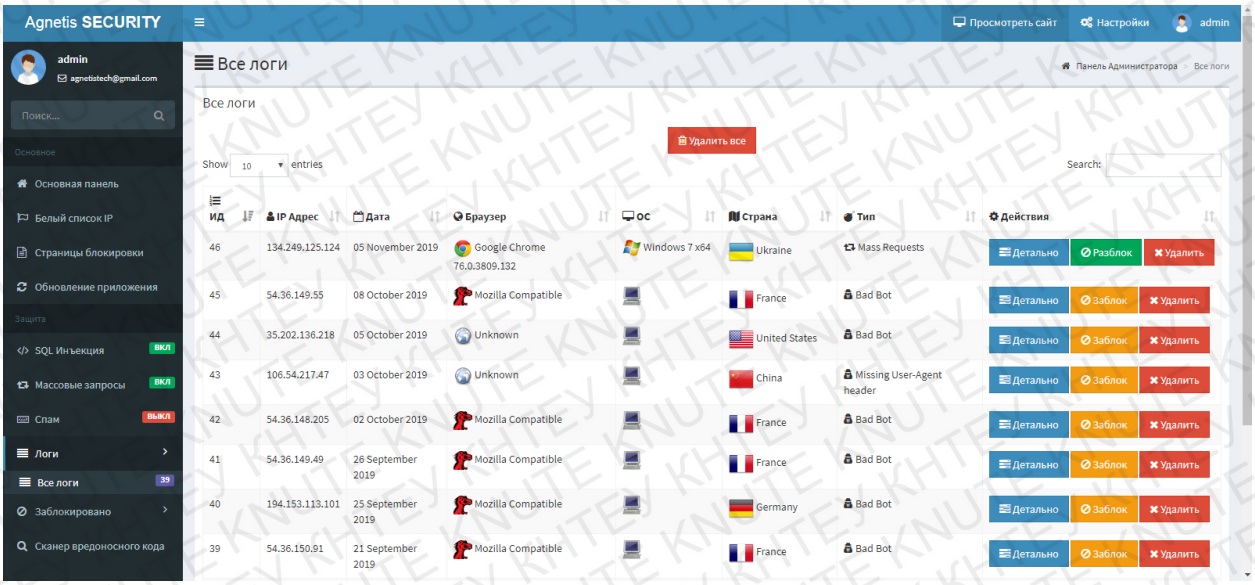

Рис. 6. Сторінка «Всі логи»

На сторінці «Всі логи» користувач має можливість побачити у вигляді таблиці всі логи.

На основній панелі системи виводиться статистика у числовому еквіваленті та у графічному поданні (рис. 7).

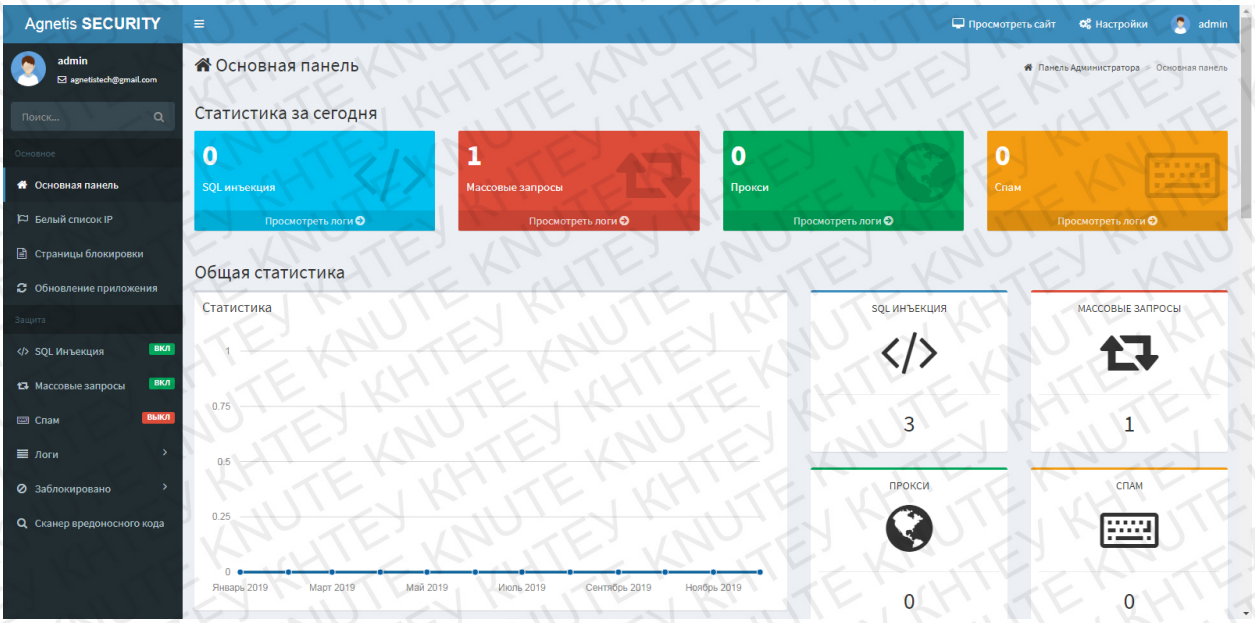

Рис. 7 . Основна панель системи

Під час тестування збоїв та недоліків у роботі програмного забезпечення для захисту серверів не виявлено, що говорить про високу якість розробки та можливість впровадження за потребою.

Додаток Б

Керівництво користувача

1. Загальні дані

Керрівництво користувача призначено для користувачів (операторів), які експлуатують виріб "Засіб захисту серверів". У керівництві містяться відомості, необхідні операторам для роботи з встановленим додатком, в тому числі описані елементи графічного інтерфейсу після його установки виконується на АРМ адміністратора через веб-інтерфейс. В ролі АРМ адміністратора може виступати будь-який персональний комп'ютер, на якому встановлений один з сучасних веб-браузерів (Microsoft Internet Explorer, Google Chrome або Mozilla Firefox).

2. Призначення

" Засіб захисту серверів " є фаєрволом прикладного рівня і призначений для захисту веб-додатків від інтернет-загроз. " Засіб захисту серверів " володіє наступними основними можливостями:

- аналіз трафіку веб-додатків і виявлення атак (вторгнень);
- блокування спроб мережевих атак при роботі з веб-додатками;
- захист веб-додатків від основних типів загроз:
  - ✓ всілякі види ін'єкцій (SQL injection, OS injection, RCE, XPath-injection, XXE);
  - ✓ реалізація атак на зворотний шлях в директоріях і на включення локальних файлів (Directory traversal, Remote / Local FileInclusion);
  - ✓ міжсайтовий скриптинг (XSS);
  - ✓ міжсайтова підробка запитів (CSRF);
  - ✓ реалізація атак, що експлуатують небезпечні налаштування веб-додатку (security misconfiguration);
  - ✓ реалізація переборних атак (bruteforce);
  - ✓ "розумний" DoS (application-level DoS);

- ✓ реалізація атак, що експлуатують недоліки системи аутентифікації (фіксація сесії, крадіжка сесії, відсутність тайм-ауту і т. п.);
- ✓ реалізація атак на механізми авторизації (Insecure Direct Object Ref-erences, Missing Function Level Access Control);
- ✓ автоматизований обхід додатку, масове скачування інформації, використання сканерів вразливостей (web scraping, automation);
- виявлення підозрілої активності користувачів веб-додатків;
- автоматична настройка захисних механізмів " Засіб захисту серверів " під кокретний додаток, що захищається (навчання);
- контроль доступу до функцій і журнал дій користувачів додатків;
- інтеграція зі зовнішніми системами управління подіями інформаціонної безпеки (SIEM) і з системами управління завданнями (issue tracking) .

3. Принципи функціонування

В процесі роботи " Засіб захисту серверів " спирається на моделі нормальної поведінки додатку, що захищається, виявляючи аномалії. Аномалії і інші атрибути (користувач, сесія, дія і ін.) Прив'язуються до HTTP-транзакціях (тут і далі - пара запит-відповідь). Групи транзакцій об'єднуються в події, які мають час життя і свідчать про ті чи інші аномалії в роботі " Засіб захисту серверів ".

4. Адміністрування " Засіб захисту серверів " здійснюється з програми управління (Консолі) через веб-інтерфейс. Для підключення до " Засіб захисту серверів " необхідно запустити веб-браузер і в адресному рядку ввести https: // IP: 8443, де IP-адреса вузла з встановленим " Засіб захисту серверів ". На екрані з'явиться екран із вітанням. Введіть облікові дані і натисніть кнопку "Увійти". За замовчуванням створюється обліковий запис з ім'ям admin, її первісний пароль задається в конфігураційному файлі

/etc/wafui.yml. Після першого входу в обліковий запис admin система зажадає змінити початковий пароль на новий. З'явиться Головне вікно Консолі " Засіб захисту серверів ".

За замовчуванням після входу користувача в систему домашньою сторінкою Консолі є розділ "Основна панель", що відображає зведену інформацію по роботі " Засіб захисту серверів ". Головне вікно програми містить наступні компоненти (рис. 1):

1. Заголовок вікна (верхній рядок), що містить ім'я програмного продукту, а також інформацію про кількість зареєстрованих в " Засіб захисту серверів " додатків, про кількість додатків, що мають в даний момент високий рівень загрози , та загальну кількість правил. Клацанням мишею по синім циф-рам здійснюється перехід до відповідних розділів.

2. Рядок пошуку дозволяє швидко перейти до необхідного параметру.

3. Основне меню Консолі, що містить наступні елементи:

- назва програмного продукту і поточний час;
- список розділів;
- кнопка "Меню", при натисканні якої розгортається основне меню (див. рис. 7). Повторне натискання кнопки дозволяє звернути основне меню;
- кнопки "Налаштування користувача" і "Вихід" .

Смуга прокрутки дозволяє перегортати екран

Лістинг програмного коду. Панель адміністратора

```
<?php
$configfile = 'config.php';
if (!file_exists($configfile)) {
   echo '<meta http-equiv="refresh" content="0; url=install" />';
   exit();
}

include "config.php";

session_start();

if ($client == 'Yes') {
   echo '
<script>
if (window!=window.top) {
 window.location.href = "dashboard";
}
else {
 alert("Direct access is not allowed.");
 window.stop();
}
</script>
';
} else {

   if (isset($_SESSION['sec-username'])) {
      $uname = $_SESSION['sec-username'];
      $table = $prefix . 'users';
```

```
            $suser = mysqli_query($connect, "SELECT * FROM `$table` WHERE
username='$uname'");
            $count = mysqli_num_rows($suser);
            if ($count > 0) {
               echo '<meta http-equiv="refresh" content="0; url=dashboard" />';
               exit;
            }
         }

         $_GET                    =                filter_input_array(INPUT_GET,
FILTER_SANITIZE_STRING);
         $_POST                   =                filter_input_array(INPUT_POST,
FILTER_SANITIZE_STRING);

         $error = "No";
      ?>
      <!DOCTYPE html>
      <html lang="en">
         <head>
            <meta charset="utf-8">
            <meta http-equiv="X-UA-Compatible" content="IE=edge">
            <meta name="viewport" content="width=device-width, initial-
scale=1">
            <META NAME="ROBOTS" CONTENT="NOINDEX,
NOFOLLOW">
            <title>Agnetis SECURITY &rsaquo; Панель Администратора</title>

            <!-- CSS -->
            <link                                                rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Roboto:400,100,300,500">
```

```
        <link rel="stylesheet" href="assets/css/bootstrap.min.css">
        <link rel="stylesheet" href="assets/plugins/font-awesome/css/font-awesome.min.css">
        <link rel="stylesheet" href="assets/css/admin.min.css">

        <!-- Favicon -->
        <link rel="shortcut icon" href="assets/img/favicon.png">
      </head>

      <body class="hold-transition login-page">

    <div class="login-box">
      <div class="login-logo">
        <a href="index"><i class="fa fa-get-pocket"></i> Agnetis <strong>SECURITY</strong></a>
      </div>
      <div class="login-box-body">
        <p class="login-box-msg">Дипломная версия</p>
    <?php
      if (isset($_POST['signin'])) {
        $username = mysqli_real_escape_string($connect, $_POST['username']);
        $password = base64_encode($_POST['password']);
        $table    = $prefix . "users";
        $check    = mysqli_query($connect, "SELECT username, password FROM `$table` WHERE `username`='$username' AND password='$password'");
        if (mysqli_num_rows($check) > 0) {
          $_SESSION['sec-username'] = $username;
          echo '<meta http-equiv="refresh" content="0;url=dashboard">';
        } else {
```

```
            echo '<br />
                <div class="callout callout-danger">
                <i class="fa fa-exclamation-circle"></i> Введеные данные неверны .
            </div>';
            $error = "Yes";
            }
        }
    ?>
            <form action="" method="post">
                <div class="form-group has-feedback <?php
        if ($error == "Yes") {
        echo 'has-error';
        }
    ?>">
                    <input type="username" name="username" class="form-control" placeholder="Логин" <?php
        if ($error == "Yes") {
        echo 'autofocus';
        }
    ?> required>
                    <span class="glyphicon glyphicon-user form-control-feedback"></span>
                </div>
                <div class="form-group has-feedback">
                    <input type="password" name="password" class="form-control" placeholder="Пароль" required>
                                <span class="glyphicon glyphicon-lock form-control-feedback"></span>
                </div>
```

```
            <div class="row">
                <div class="col-xs-12">
                    <button type="submit" name="signin" class="btn btn-primary
btn-block btn-flat btn-lg"><i class="fa fa-sign-in"></i>
     Вход</button>
                </div>
            </div>
        </form>
    </div>
</div>

    <!-- Javascript -->
    <script src="assets/js/jquery-3.1.1.min.js"></script>
    <script src="assets/js/bootstrap.min.js"></script>

  </body>
</html>
<?php
}
?>
```