Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ПОЯСНЮВАЛЬНА ЗАПИСКА

Дипломної роботи

магістра

(назва освітньо-кваліфікаційного рівня)

галузь знань *12 Інформаційні технології*
(шифр і назва галузі знань)

спеціальність *125 Кібербезпека*
(код і назва спеціальності)

освітній ступень *магістр*
(назва освітньої програми)

освітньо-наукова програма *кібербезпека*

на тему: «Методи захисту від ін'єкційних атак»

Виконавець: студент ІІ курсу, групи КБм-21

Федорчук Олексій Вячеславович
(підпис) (прізвище ім'я по-батькові)

	Прізвище, ініціали	**Оцінка**	**Підпис**
Науковий керівник	Бабенко Т. В.		
Рецензент	Ткач В. М.		
Нормоконтроль	Даков С. Ю.		

Київ 2022

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
завідувач кафедри кібербезпеки
та захисту інформації
_____ Н.В. Лукова-Чуйко
«__» _____ 2021 р.

ЗАВДАННЯ
на виконання дипломної роботи

спеціальності *125 Кібербезпека*
(код і назва спеціальності)

студенту КБм-21 (група) Федорчуку Олексію Вячеславовичу (прізвище ім'я по-батькові)

Тема дипломної роботи Методи захисту від ін'єкційних атак

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол № 5 від 29.10.2021

2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень	Процес захисту від ін'єкційних атак на веб-ресурси
Предмет досліджень	Методи захисту від ін'єкційних атак на веб-ресурси
Мета	Розробка методу захисту від ін'єкційних атак на веб-ресурси.
Вихідні дані для проведення роботи	Методи захисту від ін'єкційних атак для веб-ресурсів

3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна	У роботі запропоновано методи захисту від ін’єкційних атак, що раніше не використовувались
Практична цінність	Запропоновані методи захисту від ін’єкційних атак можуть бути використані , як елементи в системах управління інформаційною безпекою.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Робота виконана у повному обсязі відповідно до теми.

5. ЕТАПИ ВИКОНАННЯ РОБОТИ

Найменування етапів робіт	**Строки виконання робіт (початок-кінець)**
Розробка плану для досягнення мети роботи	29.10.2021 – 23.01.2022
Аналіз літературних джерел	24.01.2022 – 14.02.2022
Розробка методу захисту від ін’єкційних атак	15.02.2022 – 24.04.2022
Оформлення і друк пояснювальної записки	25.04.2022 – 19.05.2022

6. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект	Зменшення збитків через несанкціонований доступ до інформації
Соціальний ефект	Покращення технологій забезпечення захисту інформації.

7. ДОДАТКОВІ ВИМОГИ

Завдання видав _____ _____
(підпис) (прізвище, ініціали)

Завдання прийняв до виконання _____ _____
(підпис) (прізвище, ініціали)

Дата видачі завдання: _____
Термін подання дипломної роботи до ЕК: _____

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Методи захисту від ін'єкційних атак»: 63 сторінки, 7 рисунків та 11 таблиць, 35 літературних джерел.

Об’єкт дослідження – Процес захисту від ін’єкційних атак.

Мета роботи – Розробка методу захисту від ін’єкційних атак.

Методи дослідження – синтез та аналіз методів захисту від ін’єкційних атак, імітаційне моделювання атак з використанням ін’єкційних атак.

Робота складається з трьох частин: дослідження теоретичного матеріалу що до запобігання ін’єкційним атакам, проведення аналізу вже існуючої системи та проведення експерименту у лабораторних умовах. На основі отриманих результатів була створена методика захисту від ін’єкційних атак, яка може бути застосована як основа при проєктувані систем запобігання ін'єкційним атакам.

Наукова новизна: У роботі створені нетипові методи захисту від ін’єкційних атак, які можуть бути застосовані як основа при проєктувані систем захисту від ін'єкційних атак.

Актуальність теми: Із року в рік публікується звіт OWASP Top 10 в якому перелічені основні проблеми, пов'язані з безпекою веб-застосунків. Він регулярно оновлюється, щоб постійно відображати десять найбільш серйозних ризиків, з якими стикаються організації. І постійним членом цієї десятки є ін’єкційні атаки.

Тому актуальним є дослідження технології проведення ін’єкційних атак та відповідно методів захисту від них. Побудова моделі захисту від ін’єкційних атак дозволить мінімізувати шкідливий вплив спотворених запитів, а також визначити основні напрямки захисту в залежності від ймовірності атаки та важливості інформації, що зберігається на певному ресурсі.

Ключові слова: інформаційна безпека, SQL Injection (SQLIA), HTTP-запити, бази даних, модель захисту, ідентифікація, авторизація, автентифікація.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

SQL	–	structured query language
SQLIA	–	SQL-injection
(D)DoS	–	(Distributed) Denial-of-Service
LDAP	–	Lightweight Directory Access Protocol
API	–	Application Programming Interface
XSS	–	Cross-Site Scripting
HTTP	–	HyperText Transfer Protoco
БД	–	База даних
IT	–	Information Technology
ПЗ	–	Програмне забезпечення
ЦОД	–	Центр обробки даних
ІТЗ	–	Інформаційно-технічні засоби

ЗМІСТ

РЕФЕРАТ 4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ 5
ЗМІСТ 6
ВСТУП 8
РОЗДІЛ 1 ОПИС ПРОБЛЕМАТИКИ ДОСЛІДЖЕННЯ 10
1.1 Методи реалізації ін'єкційних атак 10
1.1.1 Методи реалізації SQL-ін'єкції 11
1.1.2 Методи реалізації кодової атаки 15
1.2 Моделі ідентифікації ін'єкційних атак 16
1.2.1 Модель ідентифікації кодової атаки 17
1.2.2 Модель ідентифікації та запобігання SQL-ін’єкції 20
1.3 Постановка задачі дослідження 24
Висновки до першого розділу 26
РОЗДІЛ 2 РОЗРОБКА МОДЕЛІ ЗАХИСТУ ВІД ІН’ЄКЦІЙНИХ АТАК 28
2.1 Вибір та обґрунтування параметрів розроблюваної моделі 28
2.2 Синтез моделі захисту від ін’єкційних атак 32
2.3 Перевірка моделі на адекватність 37
Висновки до другого розділу 40
РОЗДІЛ 3 ПРАКТИЧНЕ ЗАСТОСУВАННЯ РОЗРОБЛЕНОЇ МОДЕЛІ 42
3.1 Методика проведення експериментальних досліджень 42
3.2 Дослідження характеристик моделі захисту від ін’єкційних атак 48
3.3 Розробка методики захисту від ін’єкційних атак 50

Висновки до третього розділу 56

ВИСНОВКИ 57

ВИКОРИСТАНІ ДЖЕРЕЛА 60

ДОДАТОК А 64

ВСТУП

Актуальність теми. В час широкого використання дистанційного доступу до інформаційних ресурсів, окрім полегшення шляхів отримання інформації та скорочення часу на її обробку, збільшився рівень ризику несанкціонованого доступу до інформації з обмеженим доступом. Зловмисники використовують для цього ряд засобів, і одним з таких методів є ін’єкційна атака на інформаційний ресурс, тобто впровадження шкідливого коду на сервер чи робочу станцію користувача через зовнішні запити з метою подальшого виведення з ладу інформаційної системи чи витоку даних, що містяться в ній і не призначені для стороннього використання.

Тому актуальним є дослідження технології проведення ін’єкційних атак та відповідно методів захисту від них. Побудова моделі захисту від ін’єкційних атак дозволить мінімізувати шкідливий вплив спотворених запитів, а також визначити основні напрямки захисту в залежності від ймовірності атаки та важливості інформації, що зберігається на певному ресурсі.

Мета роботи полягає в розробці моделі захисту від ін’єкційних атак на інформаційні ресурси.

Об’єктом дослідження є методи боротьби з ін’єкційними атаками.

Предметом дослідження є методика оцінки ступеню захищеності інформаційних ресурсів від ін’єкційних атак.

Для досягнення мети роботи необхідно виконати такі завдання:

- розглянути поняття ін’єкційних атак і методи захисту від них;
- провести огляд існуючих моделей захисту від атак;
- здійснити постановку задачі дослідження;
- провести вибір та обґрунтування параметрів розроблюваної моделі;
- здійснити синтез моделі захисту від ін’єкційних атак;
- провести перевірку моделі на адекватність;
- розробити методику проведення експериментального дослідження;

- розробити методики захисту від ін’єкційних атак

Методологічна основа та методи дослідження: основою роботи послужили загальнонаукові та спеціальні методи дослідження, зокрема, історичний, аналітичний, аналізу та синтезу, порівняння й інші. Історичний метод було застосовано для вивчення сутності ін’єкційних атак та методик аналізу ризиків їх виникнення.

Інформаційною базою дослідження є наукові розробки, публікації вітчизняних та закордонних вчених, офіційні матеріали та звітні дані. Крім того були використані наукові здобутки із загальної теорії управління, автоматизованих систем управління, моделювання складних систем, захисту інформації, теорії прийняття рішень, інформаційної безпеки.

РОЗДІЛ 1
ОПИС ПРОБЛЕМАТИКИ ДОСЛІДЖЕННЯ

1.1 Методи реалізації ін'єкційних атак

Для дослідження методів реалізації ін’єкційних атак необхідно спочатку розглянути сутність даного поняття. Ін’єкційна атака – це процес, за допомогою якого зловмисник отримує доступ до операційної системи або заражає веб-додаток шкідливим кодом, щоб отримати особисту інформацію або зламати систему. Зловмисник змушує операційну систему думати, що команда була ініційована користувачем, і операційна система обробляє несанкціоновану команду.

Ін’єкційні атаки є однією з найстаріших і найнебезпечніших кібератак через спосіб їх проведення. Зловмисник може завантажити будь-яку необхідну йому інформацію з пристрою кінцевого користувача шляхом отримання доступу через ін’єкцію.

Розрізняють наступні методи реалізації ін’єкційних атак [7]:

– Кодова ін’єкція
– SQL ін’єкція
– Командна ін’єкція
– Міжсайтовий сценарій
– XPath ін’єкція
– Ін’єкція пошти команди
– Ін’єкція CRLF
– Ін’єкція заголовка хоста
– Ін’єкція LDAP

Найбільш розповсюдженими з них є SQL ін’єкції, тому вважаємо за доцільне розглянути їх механізми детальніше. Метою атаки з використанням SQL-ін'єкцій є отримання несанкціонованого доступу до конфіденційної інформації. За останні роки відбулось велика кількість гучних випадків витоку конфіденційної інформації

саме з застосуванням SQL-ін'єкцій. Більшість таких випадків призводили репутаційних, а інколи й значних фінансових збитків. Також використовуючи подібні атаки зловмисник має змогу створити «канал постійного доступу» до системи, що може мати довгострокові ризики безпеки.

Найбільш популярні приклади впровадження SQL- ін'єкцій [9]:

– отримання інформації, що виходить за рівень доступу користувача, при цьому запит SQL змінюється так щоб отримати додаткові результати;

– втручання у логіку застосунку, де змінюється запит, щоб заважати логіці застосунку;

– UNION атаки, які дають змогу отримати дані з різних таблиць бази даних ресурсу;

– вивчення технічних особливостей бази даних, задля пошуку и подальшого використання вразливостей;

– сліпа SQL- ін'єкція, запит без отримання відповіді від застосунку.

Тому даний вид атаки входить до переліку найнебезпечніших загроз кібербезпеці.

1.1.1 Методи реалізації SQL-ін'єкції

Веб-додатки часто вразливі до атак, які дозволяють зловмисникам легко отримати доступ до бази даних програми. Атака з ін’єкцією SQL відбувається, коли зловмисник використовує спеціально створене поле для створення та надсилання запиту веб-програмі, яка веде себе не так, як задумав розробник програмного забезпечення. Тому доцільно розглянути цей тип ін’єкції коду.

Відомо, що атаки SQL Injection (SQLIA) є однією з найпоширеніших загроз безпеки для баз даних. SQLIA — це клас атак введення коду, який не використовує перевірку користувача. Фактично, зловмисники можуть отримувати несанкціонований доступ шляхом дописування частини до кінцевого рядка запиту, який запускається в базах даних. Фінансові веб-додатки та системи захисту інформації також можуть стати жертвами цієї вразливості, оскільки,

використовуючи цю вразливість, зловмисники можуть поставити під загрозу їхню цілісність, захищеність та конфіденційність. Тому розробники вибрали деякі методи захисного шифрування, щоб закрити цю вразливість, але цього недостатньо.

Розглянемо 3 основні типи атак, що застосовують техніку SQL-ін'єкції, оскільки інші типи не є надто ефективними або ж є підвидами досліджуваних видів.

1. Внутрішньосмугова SQL-ін’єкція (класична атака). Цей тип є найпоширенішим, ін'єкція, в основному, відбувається, коли потенційний зловмисник може використати той самий канал зв'язку для проведення атаки та наступного збору результатів.

При цьому внутрішньосмугові SQL-ін'єкції поділяються на два різновиди:

– SQL-ін'єкція на основі помилок. Заснована на повідомленні про помилку, що видається сервером баз даних, для отримання інформації про її структуру;

– використання SQL з урахуванням об'єднання. Заснована на використанні оператора SQL UNION для об'єднання результатів двох або більше операторів SELECT в один результат, який потім повертається як HTTP відповідь [1].

2. Інференційна SQL-ін'єкція (сліпа або Blind SQLi). При атаках з використанням сліпої SQL-ін'єкції зловмисник не може побачити результат своєї атаки, оскільки дані не передаються через веб-програму.

Тому вона також називається Blind SQLi. Інференціальні SQL-ін'єкції бувають двох типів:

– бульова сліпа SQLi. Заснована на надсиланні SQL-запиту до бази даних, що має на меті змусити програму повернути необхідний результат в залежності від того, чи повертає запит результат «TRUE» або «FALSE»;

– сліпа SQLi, що базується на часі. Заснована на надсиланні відповідного SQL-запиту до БД, що призводить до затримки відповіді. Власне, час відповіді вкаже зловмиснику, чи є результатом запиту «TRUE» або «FALSE»;

3. Позасмугова SQL-ін'єкція. Впровадження SQL-ін'єкції відбувається, коли зловмисник не може використовувати один і той же канал для запуску атаки та збору результатів.

Позасмугові методи дають зловмиснику можливість використати альтернативу логічним методам SQL-ін'єкції, заснованим на часі, особливо якщо відповіді сервера не дуже стабільні, роблячи захист, що базується на часі відповіді, ненадійним [1].

Ці методи залежать від здатності сервера БД робити DNS (Domain Name Server) або HTTP запити для доставки необхідних даних хакеру.

В основному SQL-ін'єкції складаються з трьох етапів [2]:

– відправлення шкідливого HTTP-запиту у додаток;

– створення SQL-виписки;

– SQL-запит до БД на зворотному шляху;

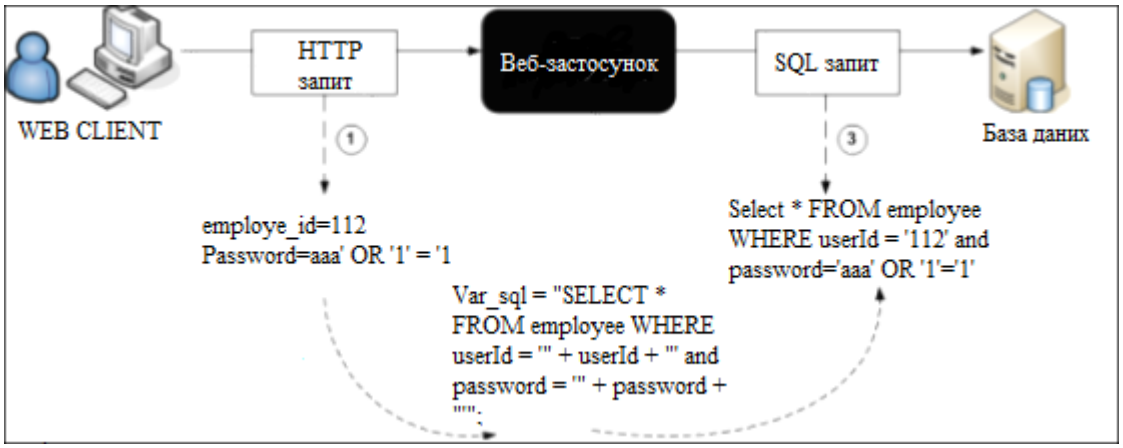

Рисунок 1.1 – Приклад атаки SQL-ін’єкції [7]

Розглянемо основні вразливості, які доволі часто існують у веб-додатках та експлуатуються при SQL-ін'єкції:

– недійсний вхід. Найбільш поширена вразливість для SQL-ін’єкцій. У веб-додатку є деякі параметри, які зазвичай використовуються у запитах SQL. Якщо їх не контролювати, використання цих параметрів може бути використано в атаках SQL-ін’єкцій;

– привілеї доступу. Зазвичай в базі даних вони визначаються як правила для визначення того, який суб'єкт бази даних має право отримати доступ до якого саме

об'єкту та яку саме операцію асоціюють з певним користувачем що надає можливість виконання їх на об'єктах;

– неконтрольований змінний розмір. Використання даних більшого розміру, ніж необхідно (замість поля для даних певного типу та розміру надається універсальне поле тощо), дозволяє зловмисникам модифікувати данні або сфабриковувати SQL-запити;

– повідомлення про помилку. Генерація заднім числом або сфабрикування повідомлень про помилки зі сторони серверу можуть бути отримані на стороні клієнта. Ці повідомлення корисні при розробці але можуть бути використані для атак зловмисників.

– динамічний SQL-запит. Часто для функціонування системи використовуються скрипти, які мають право вносити зміни і, перехоплюючи інформацію що надає користувач, таку як ім'я або пароль, можуть складати запити «WHERE» в операторі запиту і проводити ін’єкції;

– змінний тип даних. Регламентовані зміні повинні мати встановлений тип даних, щоб не допускати введення даних які можуть нашкодити роботі системи;

– клієнтський контроль. Якщо верифікація вводу виконується лише зі сторони клієнта, то зловмисник може її обійти завдяки кросплатформенних сценаріїв на проміжних етапах;

– SQL-процедури. Це твердження, які зберігаються в БД на базі мови SQL. Основною проблемою використання цих процедур є то, що вони можуть виконуватись зловмисником за для внесення змін у базу даних, операційну систему або навість інші компоненти мережі;

– множинні вирази. Якщо база даних підтримує функцію UNION, то зловмисник має більше шансів успішної атаки;

– суб-запити. Підтримка режиму суб-запиту - вразливість для СУБД при розгляді SQL-ін'єкції, адже саме у формі такого під запиту зловмисник може вбудувати шкідливий код.

1.1.2 Методи реалізації кодової атаки

Ін’єкція може призвести до втрати даних, тривалої відсутності відповіді сервера або відмови у доступі. У деяких випадках метою кодової атаки може бути захоплення управління операційною системою користувача.

Деякі типи введення коду є помилками інтерпретації, які лише надають особливе значення для користувача. Подібні помилки інтерпретації існують і за межами інформаційних технологій, наприклад, у повсякденній мові власні імена інколи неможливо відрізнити від звичайних слів. Аналогічно, з деякими типами введення коду неможливо відрізнити введення коду користувачем від регламентованих системних команд. В деяких операційних системах стоїть захист від випадкового введення, наприклад, заборона створювати вручну папки з іменами системних ресурсів тощо. Проте для розробників програмного забезпечення обійти цю заборону не є проблемою.

Технології введення коду популярні в системах злому для отримання інформації, розширення можливостей або несанкціонованого доступу до системи.

Код ін’єкції можна використовувати зловмисно для кількох цілей, зокрема:

– Швидко змінювати окремі значення в базі даних за допомогою ін'єкції SQL. Наслідки цього можуть варіюватися від відхилення веб-додатків до серйозного порушення конфіденційних даних;

– встановлення шкідливого програмного забезпечення або виконання шкідливого коду на сервері шляхом вставки коду сценарію на стороні сервера (наприклад, PHP або ASP);

– Привілейована ескалація root через використання вразливості Shell в rootuid-бінарних системах UNIX або локальних системах, які використовують службу в Windows;

– атака користувачів Інтернету за допомогою HTML / Script Injection.

Ін'єкційний код можна використовувати з добрими намірами; наприклад, зміна або коригування поведінки програми чи системи шляхом введення коду

може «змусити» систему виконувати певну роботу без злих намірів. Введення коду може, наприклад, спровокувати наступні дії:

– Вивести новий стовпець, який не має відображатися.

– Встановити новий спосіб фільтрації, упорядкування або групування даних використовуючи поля інформації котрі не повинні відображатися;

– Додавання спеціальних частин, які можна використовувати для підключення до онлайн-ресурсів в офлайн-програмі.

Деякі користувачі можуть санкціонувати виконання певного шкідливого коду, оскільки вхідні дані, які вони надають програмі, не візуалізуються в повній мірі або ж користувач не має уявлення про формат повідомлення, наприклад:

– Те, що користувач може вважати дійсним введенням, може містити додатково маркерні символи або рядки, які розробник зарезервував для спеціального значення (спец символи, які не відображаються при друкуванні, проте мають значення при зчитуванні, ідентифікації та порівнянні текстів);

– Користувач може надіслати як вхідний файл неправильно згенерований файл, який обробляється в одній програмі, але не дійсний для системи-отримувача.

Іншим уразливим застосуванням коду може бути виявлення найбільш помилкових дефектів для усунення цих недоліків. Цей метод ще відомий як тест на проникнення, або тест «білого капелюха». Даний тест є робочою процедурою, коли хакер-експерт перевіряє розроблене програмне забезпечення на вразливості, тобто здійснює санкціоновані спроби зламу. Тому якщо підозра на ін’єкцію не пов’язана з санкціонованими робочими процедурами, слід вжити заходів щодо протидії.

1.2 Моделі ідентифікації ін'єкційних атак

Ін’єкційні атаки найкраще відслідковувати та блокувати досить рано, перш ніж зловмисник зможе повністю захопити систему. Тому розглянемо окремі моделі, які дозволять ідентифікувати ін’єкційні атаки.

Найефективніший спосіб виявлення вразливостей із застосуванням ін’єкцій – це запровадити у локальній мережі автоматизований веб-сканер уразливостей.

Можна також вручну виявити наявність загрози за допомогою тесту на проникнення, але це займе більше часу та ресурсів.

Використання автоматичного сканера швидше реагує на сигнали загроз і допомагає ініціювати захисну відповідь для протидії кібератакам.

1.2.1 Модель ідентифікації кодової атаки

Ідентифікація кодової атаки може відбуватись через аналіз змін трафіку та відхилень від базового режиму використання. Технічна реалізація подібних рішень передбачає використання деяких додаткових пристроїв. Це насамперед система детекції атак у вхідному трафіку та система яка буде відфільтровувати трафік. Під час атаки ці системи повинні затримати «інфікований трафік», не даючи йому потрапити до внутрішньої мережі. Для цього система повинна виконувати певні послідовні дії. Аналогічним чином моніторинг виконується і при спробах DDos-атак [5]:

1. фіксація факту проникнення у систему через нестандартну динаміку трафіку на окремих вузлах;

2. виявлення джерел атак (веб-додаток, сервер, пошта тощо);

3. припинення трафіку, що надходить від джерел атаки, якщо це відбувається на боці зловмисника, а не всередині локальної мережі;

4. перевірка успішності протистояння атаці, чи відновився типовий трафік.

Використання ентропії мережевого трафіку для виявлення атак ґрунтується на порівнянні ентропії трафіку, усередненої за короткий проміжок часу (локальна міра невизначеності) з відповідним показником за тривалий період часу (міра глобальної невизначеності), розрахованою без атаки на мережевий сервіс. У випадку, якщо локальний захід значно відрізняється від відповідного глобального, ймовірність мережевих атак значно зростає [6].

Існуючі системи виявлення умовно розділяються на: виявлення аномалій і виявлення ознак. Головним недоліком систем на виявленні ознак є, те що вони базуються на виявленні вже відомих типів атак. Однак перелік можливих загроз і їх

методів передачі постійно збільшується і такі системи швидко застарівають. Системи виявлення аномалій базуються на припущені, що система повинна працювати певним чином, наприклад, що повинна зберігатись статична однорідність трафіку. Однак такі системи повинні переналаштовуватись при будь-якій вагомій зміні структури трафіку. Найкращим із можливих рішень є комбінування цих систем.

Пропонується побудувати систему захисту на основі наступних елементів [7]:

– агенти відстеження;
– заходи попередньої обробки та зберігання;
– інформацію про операції, що буде описувати систему;
– аналітичні компоненти для виявлення загроз;
– регламент заходів проти атак.

Основою побудови подібних систем є визначення необхідного математичного забезпечення на кожному етапі роботи [7]:

1. Контроль трафіку. Під цим розуміється перехоплення трафіку з подальшою його оцінкою. Для цього створюється алгоритм визначення того скільки пакетів буде захоплено так як часто. Перехоплення трафіку не повинно проходити через рівні проміжки часу, щоб уникнути створення «сліпих зон». А також не повинно проходити занадто часто, щоб не сповільнювати трафік.

2. Попередня обробка перехоплених пакетів, оцінка загроз, збереження інформації про випадок атаки. На цьому етапі необхідно контролювати витрати ресурсів системи та їх продуктивне використання, тому рекомендуються прості та адаптивні засоби ідентифікації та вимкнення окремих елементів мережі, де відбулась ін’єкція.

3. Аналіз даних при завантаженні, виявлення атак, оцінка загроз. Після того як інформація була збережена у окремому репозиторії необхідно провести аналіз ризиків. Для цього доцільно використовувати алгоритми багатоканального моніторингу та розрахунку ковзного середнього щодо трафіку по окремих вузлах.

4. Фоновий аналіз даних для встановлення спроб сканування, атак погіршення якості та пульсуючих атак. Його необхідно проводити за певним графіком.

5. Виявлення атаки. Якщо при перевірці порогових на попередніх етапах були виявлені значення котрі перевищують норму , або ж виявлена нестандартна активність, існує можлива загроза нападу.

6. Оцінка ризику, вибір моделі, верифікація, пошук стратегії. При виявленні атаки необхідно приймати рішення щодо вживання заходів. Заходи можуть відрізнятись в залежності від типу атаки і її характеристик. На вибір стратегії повинні впливати результати моделювання взаємодії між агентами захисту та агентами нападу. Вивчення аналітичних моделей дозволяє підвищити ефективність контрзаходів і купірувати можливі наслідки [7].

Щоб уникнути проблем із кодуванням, варто використовувати безпечну обробку введення-виводу, наприклад:

– API захищені від усіх введених символів при правильному використанні. Параметричні запити (також відомі як «складні запити», «пов'язані змінні») дозволяють інтерпретувати дані користувача за межами рядка. Крім того, критерії API та подібні API відхиляються від концепції командного рядка, яку необхідно створити та інтерпретувати;

– здійснення мовного розподілу через систему статичного типу;

– Перевірка введення, наприклад білий список лише відомих значень, може бути виконана на стороні клієнта, наприклад за допомогою JavaScript, або на стороні сервера, що є більш безпечним;

– кодування введення, наприклад уникнення небезпечних символів. Наприклад, PHP використовує функцію htmlspecialchars (), щоб уникнути спеціальних символів для безпечного відображення тексту в HTML, і mysqli :: real_escape_string () для виділення

1.2.2 Модель ідентифікації та запобігання SQL-ін’єкції

Щоб запобігти SQL-ін’єкції, як рішення часто пропонують захисне кодування, але це дуже складно реалізовується. Розробники не лише намагаються вбудувати певні елементи керування у вихідний код, але атаки все ще надають нові способи обійти ці елементи керування. Враховуючи новітні та найкращі методи оборонного кодування, важко слідувати розробникам. З іншого боку, впровадження найкращих практик захисного кодування дуже складне для «рядового» виконавця і вимагає спеціальних навичок. Ці міркування виправдовують необхідність вирішення проблеми ін'єкції SQL іншими методами.

SQLIA — це хакерська технологія, до якої зловмисник додає SQL-запити за допомогою полів введення веб-додатків або параметрів доступу до прихованих ресурсів. Відсутність перевірки вхідних даних у веб-додатках робить атаку на 100% ефективною.

Доцільно використовувати для ідентифікації такі дані, що міститимуться в SQL-запиті і захищатимуть від SQL-ін’єкцій:

– вихідне кодування, тобто запобігання атак на введення HTML-коду (XSS) на відвідувачів сайту;

– HttpOnly — це прапор для файлів cookie HTTP, який у разі встановлення запобігає взаємодії клієнтського сценарію з файлами cookie, запобігаючи таким чином деяким атакам XSS;

– модульне відокремлення корпусу від ядра;

За допомогою SQL-ін’єкції можуть використовуватись параметризовані запити, збережені процедури, введення в білий список тощо, щоб допомогти пом’якшити проблеми з ін’єкцією коду. Тому модель ідентифікації може базуватись на контролі внесення змін до «білого списку» та несанкціонованого доступу до інших реєстрів.

Щоб уникнути подібних атак, необхідно максимально обмежити програмний доступ до даних сервера, тобто розробляти та розгортати програми, які працюють лише з параметризованими запитами. Таким чином, у разі атак відповідних

програм, які не мають достатніх прав доступу до вихідних таблиць, виключається можливість отримання зловмисником незаконного доступу до локальних даних або баз даних. Тому розглянемо найкращі відомі способи захисту від атак SQL-ін’єкції, виділимо рекомендовані заходи безпеки при розробці компонентів бази даних та створимо загальні рекомендації щодо побудови веб-систем із використанням серверів баз даних [1].

Перший спосіб пов'язаний з необхідністю фільтрації даних, що надходять на сервер: спеціальні символи повинні бути перевірені, а числова інформація має бути звірена з введеним типом. Крім того, необхідно обмежити введення (наприклад, кількість інформації, введеної після перевірки на сервері; запити, що перевищують зазначену кількість, відхиляються).

Крім того, безпека самого процесу конфіденційності є важливим аспектом захисту від таких атак. Наприклад, використовувана база даних не повинна містити такі дані у вигляді простого тексту або таблиць (паролі повинні бути хешовані, а також містити випадково згенерований рядок, доданий перед шифруванням тощо) [2].

Другим способом забезпечення безпеки є використання параметричних запитів серверами баз даних. Загалом параметричні запити — це метод передачі даних, у якому зовнішні параметри передаються на сервер незалежно від запитів SQL. У більшості мов програмування реалізація цих функцій вже передбачена [2]:

1. Delphi - властивість TQuery.Params;
2. Java - клас PraparedStatement;
3. C # - властивість SqlCommand.Parameters;
4. PHP - властивість MySQLi.

Третій спосіб — максимально обмежити відображення повідомлень про помилки користувача (відображаються загальні повідомлення про помилки, які можливі для всіх збоїв). Однак на стороні сервера всі невдалі запити необхідно відстежувати, щоб у разі атаки їх можна було переглянути та проаналізувати (аудит інцидентів).

Періодичне тестування та моніторинг також можна вважати досить ефективними методами захисту від ін'єкції SQL. Однак найкращий спосіб перевірити – спробувати ввести свій код у SQL. Існує багато сканерів для таких атак, які знаходять вразливості, а також тестують різні типи атак.

Деякі науковці пропонують математичний спосіб ідентифікації атак SQL-ін’єкцій за допомогою обмеженої знизу функції, що залежить від вхідного рядка. Для побудови такої функції використано символи і ключові слова, які часто зустрічаються в побудові атак зловмисників. За допомогою запропонованого методу можна виявляти атаки ін’єкцій SQL, використовуючи один символ. Даний метод виявлення з використанням набору численних символів дозволяє більш точно визначити вразливість виду SQL-ін’єкції. У запропонованому методі створюється набір символів, що поєднується як з атакою, так і з нормальними запитами, з раніше відомим порогом, використовуючи приблизні дані атак та нормальних запитів. Згідно з експериментами з штучними даними набір містить пробіл, крапку з комою і праву дужку, що найбільше підходить для виявлення атаки чи нормального запиту [3].

Перевірка даних у запитах за допомогою простого обробника досить проста. Існує кілька способів відокремити дані від команд і запитів:

– Використовуйте безпечний API, який або виключає використання інтерпретатора, або надає параметризований інтерфейс.

– Перевірка даних з використанням білих списків. Звичайно, цей метод не забезпечить повного захисту, оскільки багато програм використовують спеціальні символи (наприклад, у текстових областях або API для мобільних додатків).

– Ви також повинні ввести спеціальний символовий щит для інших динамічних запитів, використовуючи відповідний синтаксис. Елементи структури SQL, такі як імена таблиць або стовпців, не можуть бути захищені, тому найменування з іменами користувачів небезпечні. Це поширена проблема з платформами звітності.

– Використовуйте елементи керування SQL у запитах, щоб запобігти витоку даних.

Також можна використовувати різноманітні додатки, які відслідковують несанкціоновані введення чи іншу підозрілу активність. Розглянемо деякі з них:

WAVES — це технологія Blackbox для тестування веб-додатків на можливість впровадження SQL. Інструмент визначає всі точки веб-додатків, які можна використовувати для впровадження SQL-ін'єкції.

JDBC-Checker може бути використаний для запобігання атак на невідповідність типів у динамічно згенерованому ланцюжку запитів.

Запити SQL Check і SQL Guard повинні перевіряютися під час виконання за моделлю, вираженою як граматика, яка приймає лише юридичні запити. SQL Guard аналізує структуру запиту до та після додавання введених даних користувача на основі моделі.

AMNESIA поєднує у собі моніторинг продуктивності та статичний аналіз. На статичній фазі він моделює різні типи запитів, які програма може легально створювати в будь-якій точці доступу до БД. Перехоплення запитів відбувається перед відправкою в БД. Перевірка відбувається з використанням статично побудованих моделей динамічної фазі.

WebSSARI має за основу статистичний аналіз при перевірці потоків перешкод. При цьому використовується очищений впускний отвір, який пройшов через певний набір фільтрів. Обмеження цього підходу є достатньою явним, чутлива функція не може бути точно виражена, через це деякі фільтри пропускаються.

SecuriFly намагається вилікувати рядки запиту, які були згенеровано за допомогою введених вами даних, але, на жаль, ін'єкцію в числові поля не можна зупинити за допомогою цього підходу. Найбільшою проблемою використання цього підходу є складність визначення джерел даних.

IDS має за основу систему виявлення SQL-ін'єкції на основі методів навчання комп'ютера. Технологія створює моделі для поширених запитів, а потім під час виконання запиту, які не відповідають моделі, вони будуть ідентифіковані як атаки. Цей інструмент ефективний для виявлення атак, але серйозно залежить від навчання.

Ще одним підходом з цієї ж категорії є SQL-IDS, який створює специфікації для с для веб-додатків, що описують передбачувану структуру SQL-запитів, створених програмою, а також автоматичному моніторингу виконання цих запитів SQL на предмет порушень цих специфікацій.

Таким чином, існує ряд методів ідентифікації загрози, проте вони розрізненими і не систематизованими, тому дана проблема потребує більш детального дослідження.

1.3 Постановка задачі дослідження

Підсумовуючи вищевикладене, можна стверджувати, що ін’єкційні атаки є досить розповсюдженим типом кіберзагроз, і вони є більш шкідливими та витонченими, ніж фішинг, ДДоС-атаки чи фізичне викрадення носіїв інформації. Ін’єкційні атаки не тільки дозволяють проникнення у мережеві ресурси користувача з метою викрадення, знищення чи викривлення, а і дозволяють закріпитись в операційній системі користувача задля подальшого витоку інформації та контролю, а в особливо серйозних випадках і до передачі шкідливих запитів на інші об’єкти. Тому задачею дослідження є синтез універсальних методів захисту від ін’єкційних атак різних видів, яка має враховувати існуючі механізми ін’єкції, реалізовувати заходи щодо «закриття дір», тобто нейтралізації вразливостей в існуючій системі, а також ідентифікувати спроби несанкціонованого доступу через ін’єкцію, і вживати заходів, адекватних загрозі.

Синтезована методика має відповідати ряду критеріїв, таких як універсальність щодо різних видів операційних систем та атак на них, масштабованість під потреби користувача, легкість у використанні та ефективність протидії ін’єкційним атакам на різних рівнях.

Тому перед постає задача створити методику виявлення ін’єкцій у різних середовищах, використання якої дозволяє формалізувати роботу по протидії атаці на функціональному рівні, рівні процесів та потоків даних. Крім того необхідно

сформувати та обґрунтувати методи використання програмного забезпечення необхідних для її реалізації.

Структура моделі матиме такий вигляд (рис. 1.1):

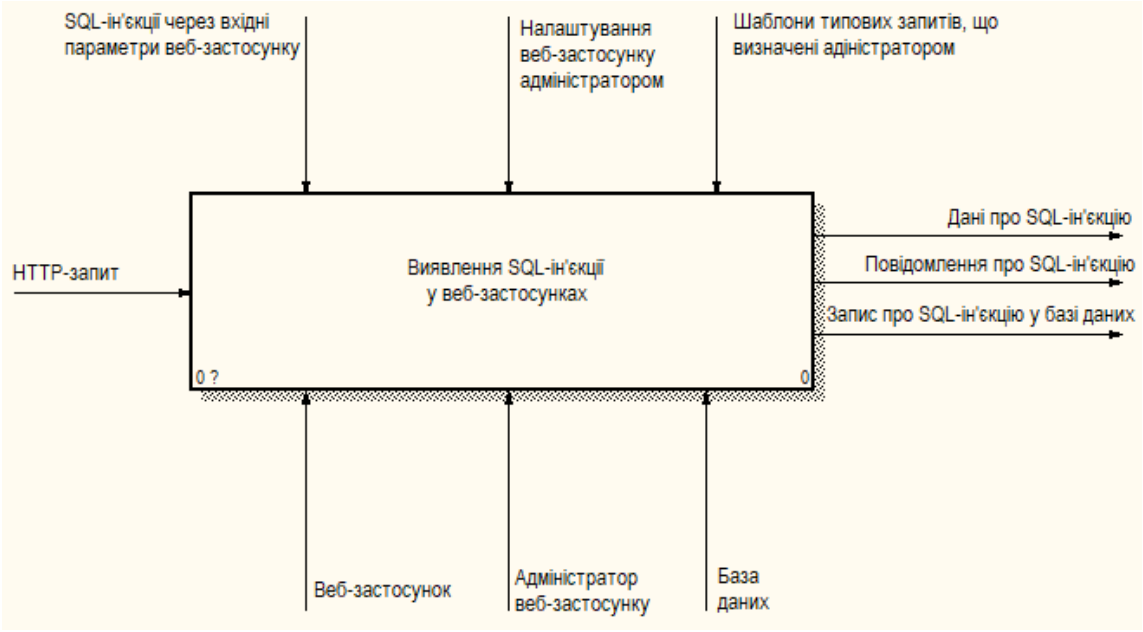

Рисунок 1.2 – Модель виявлення ін’єкційних атак на веб-застосунки

Вхідними є "HTTP-запити" - набір запитів, надісланих веб-додатку. Механізми програмного забезпечення виявлення ін’єкцій SQL відображено на діаграмі структури моделі:

– «веб-застосунок» - програмне забезпечення, що виконує функції виявлення ін'єкцій SQL, а також аналізує використання довірених мережевих ресурсів і, у разі виявлення будь-яких підозрілих або просто незвичайних подій, здатне виконати кілька незалежних дій для виявлення, ідентифікації та усунення їх причин;

– «адміністратор веб-застосунку» - механізм, що відповідає за моніторинг роботи програмного забезпечення та його взаємодії з системою;

– «база даних» - механізм, що відповідає за зберігання та отримання даних.

Вихідними даними є оброблена та структурована інформація щодо ін’єкційної атаки:

– «дані про SQL-ін`єкцію» – форматовані дані, отримані аналітичною підсистемою;

– «Повідомлення про SQL-ін'єкцію» – текстове повідомлення, яке було надіслано підсистемою сповіщень про ін'єкцію SQL із готових "даних ін'єкції SQL", отриманих від аналітичної підсистеми;

– «Запис про SQL-ін'єкцію в базу даних» – записи, створені підсистемою для зберігання інформації в базі даних.

Структура аналізу HTTP-запитів показана на рис. 1.2.

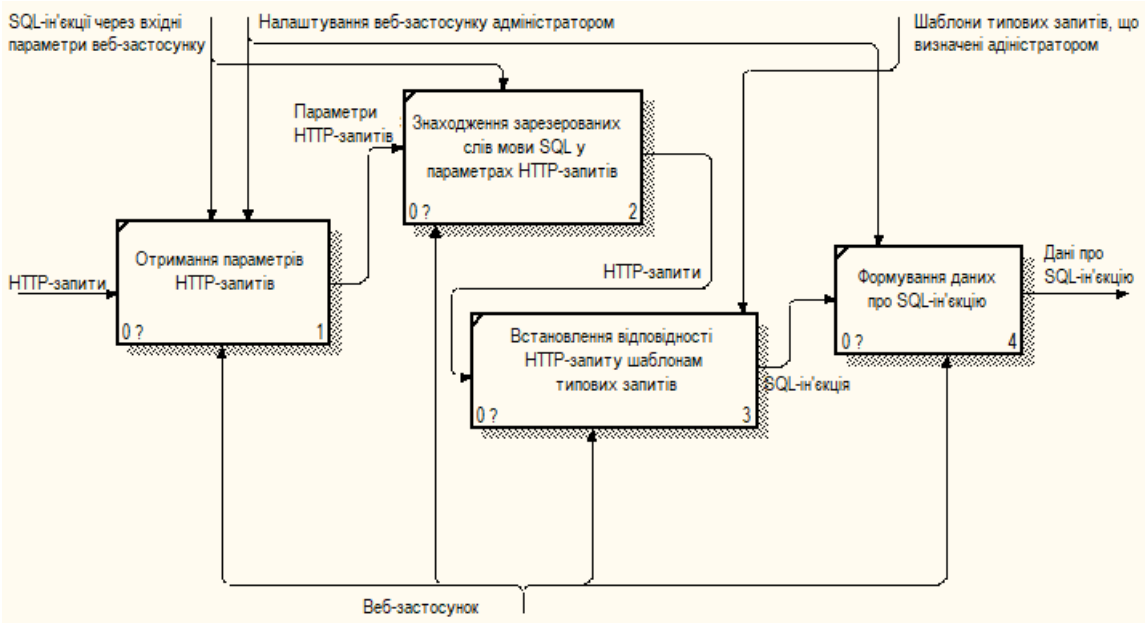

Рисунок 1.3 – Структурна схема аналізу HTTP-запитів

Дана модель передбачатиме збір інформації про можливі ін’єкції у вхідних запитах і відповідну обробку даних запитів з відсіканням ін’єкційних складових. На виході отримуємо інформацію про ін’єкцію, її структуру та розташування в запиті для формування додаткових правил відхилення запитів, що міститимуть аналогічні ін’єкційні елементи

Висновки до першого розділу

Таким чином, було розглянуто поняття ін’єкційних атак і методи захисту від них. Серед усіх типів ін’єкційних атак виокремлено найбільш небезпечні, серед яких особливу увагу приділено SQL-інєкціям; розглянуто методи та моделі ідентифікації таких атак в залежності від виду атаки. На основі проведеного

дослідження стало зрозуміло, що більшість рекомендацій, сформованих науковцями з цього приводу, є розрізненими та не дозволяють ефективно протидіяти ін’єкційним атакам, тому це дозволило сформулювати та конкретизувати задачі дослідження щодо синтезу більш універсальних моделей захисту від ін’єкційних атак.

Крім того, здійснено постановку задачі дослідження. Задачею дослідження є синтез універсальних методів захисту від ін’єкційних атак різних видів, які мають враховувати існуючі механізми ін’єкції, реалізовувати заходи щодо «закриття дір», тобто нейтралізації вразливостей в існуючій системі, а також ідентифікувати спроби несанкціонованого доступу через ін’єкцію, і вживати заходів, адекватних загрозі. Ця модель передбачатиме збір інформації про можливі ін’єкції у вхідних запитах і відповідну обробку даних запитів з відсіканням ін’єкційних складових. На виході отримуємо інформацію про ін’єкцію, її структуру та розташування в запиті для формування додаткових правил відхилення запитів, що міститимуть аналогічні ін’єкційні елементи.

РОЗДІЛ 2
РОЗРОБКА МОДЕЛІ ЗАХИСТУ ВІД ІН'ЄКЦІЙНИХ АТАК

2.1 Вибір та обґрунтування параметрів розроблюваної моделі

Існує 6 основних кроків, необхідних для ефективного управління ризиками втрат від наслідків ін'єкційних атак.

1) Методи оцінки ризику ін'єкційних атак.

Слід розпочати з визначення правил, за якими компанія буде управляти ризиками в майбутньому, оскільки різні рівні, частини організації, а також її структурні підрозділи повинні оцінювати ризики в один спосіб. Тому це необхідно визначити, якісна чи кількісна оцінка ризику буде, які шкали використовуватимуться для якісної оцінки, яким буде прийнятний рівень ризику ін'єкційних атак тощо.

2) Запровадження оцінки ризику ін'єкційних атак.

Оскільки оцінка ризику інформаційної безпеки виконується для активів, пов'язаних із заходами обробки інформації, спочатку визначаємо та перераховуємо всі активи, які можуть постраждати від ін'єкційних атак, а також усі потенційні загрози та вразливості, оцінюємо вплив та ймовірність для кожного набору активів – загроза – вразливість і, нарешті, розраховуємо рівень ризику ін'єкційних атак.

3) Впровадження обробки ризиків ін'єкційних атак.

Не всі виявлені ризики є однаково важливими, тому необхідно виділити найважливіші з них, вплив яких може завдати значної шкоди діяльності системи. Отже, варто зосередитися на цих неприйнятних ризиках, тобто небезпечною є не сама ін'єкційна атака, а втрати, які настають внаслідок неї.

4) Звіт про оцінку ризиків системи захисту від ін'єкційних атак.

На цьому етапі створюється звіт, що документує всі попередні кроки. Він може знадобитися як для аудиту, так і для контролю та перевірки власних результатів.

5) Положення щодо застосування системи захисту від ін’єкційних атак.

Цей документ фактично показує профіль безпеки конкретної системи – за результатами управління ризиками слід перерахувати всі впроваджені інструменти управління безпекою, обґрунтувати їх доцільність та описати процес впровадження. Цей запис також дуже важливий, оскільки використовується як основний документ під час перевірки працездатності моделі захисту.

6) План управління ризиками ін’єкційних атак.

Розробка цього плану полягає в чіткому визначенні того, хто буде впроваджувати кожен із інструментів управління безпекою, в який період, за яких умов тощо. Тобто в залежності від визначеного рівня ризику в поточний момент виконується різний алгоритм реагування на інциденти.

Основними загрозами інформаційної безпеки внаслідок ін’єкційних атак для системи є:

1. Несанкціонований доступ. В наш час з комп`ютерними технологіями так чи інакше пов’язана більша частина інформації. Зазвичай інформація передається по підприємству через телекомунікаційні системи. Здебільшого така інформація є конфіденційною. Порівнюючи загрози несанкціонованого доступу до конфіденційної інформації в електронних системах обробки даних з ручними системами, нескладно дійти висновку, що використання автоматизованих систем більша кількість інформації може бути перетворена у більш зручну технологічну форму без несанкціонованого доступу шляхом ін’єкційної атаки. Уразливими місцями є: термінали; АРМи; сервери та робочі станції

2. Відмова в обслуговуванні внаслідок ін’єкційної атаки. Ін`єкція збільшує навантаження на систему, генеруючи додаткові запити, тим самим збільшуючи можливу кількість запитів, які перевищують пропускну спроможність системи. Логічним наслідком є втрати від порушення роботи системи внаслідок ін’єкційних атак можуть далеко перевищити витрати на відновлення системи. Уразливими місцями є: сервери та робочі станції

3. Повномасштабна кібератака. SQL-ін’єкції можуть мати у якості наслідку більш серйозні проблеми для системи, ніж сповільнення її працездатності чи

відсутності стабільного доступу. До можливих наслідків належать: викрадення, внесення змін та знищення інформації внаслідок несанкціонованої модифікації запиту; пошкодження інформації внаслідок отримання несанкціонованого доступу до неї зловмисниками. Внаслідок чого можливі як репутаційні втрати, так і фінансові. Уразливими місцями є: термінали, банкомати, сервери та робочі станції.

З точки зору системного підходу до захисту інформації, необхідно створити певні умови. Захист інформації повинен бути [24]:

– неперервним. Будь-який ресурс системи що підлягає захисту може бути атакована у будь-який час, тому захист повинен бути безперервним;

– плановим. Спеціальна служба повинна розробляти детальні плани захисту від ін'єкційних атак у сферах її компетенцій, при цьому вона повинна враховувати загальну мету об'єкту управління;

– конкретним. Захисту підлягають тільки ті інформаційні ресурси, які об'єктивно можуть підпасти під дію ін'єкційної атаки, втрата яких може завдати об'єкту управління шкоди;

– цілеспрямованим. Захищаються повинні тільки ті, для яких існують ризики, всі інші ресурси виводяться з загального доступу через HTTP-запити;

– надійним. Методи і форми захисту від ін'єкційних атак надійно перекривати будь які можливі шляхи несанкціонованого доступу до таємниць, які вони оберігають, при чому незалежно від форми в якій вони представленні і того де вони представленні;

– універсальним. Не залежно від каналу або способу доступу до інформації, необхідно здійснювати ефективні і достатні дії, при чому незалежно від форми в якій інформація представлена і того де вона представлена;

– комплексним. Недопустимо застосовування тільки окремих форм або технічних засобі. Для захисту інформаційних ресурсів повинні застосовуватись усі види і форми захисту від ін'єкційних атак у повному обсязі.

До системи безпеки інформації ставлять такі вимоги [9]:

– чіткість розподілення прав та повноважень, які надаються користувачам для доступу до певних видів інформації;

– надання користувачам мінімальних, для виконання їх роботи, повноважень;

– зведення до мінімуму кількості спільних для кількох користувачів засобів захисту;

– врахування того що спроби і навіть випадки несанкціонованого доступу до конфіденційної інформації будуть траплятися;

– забезпечення оцінки міри конфіденційної інформації;

– забезпечення контролю цілісності засобів захисту і миттєвого реагування на їх несправність.

Організацію захисту інформації в автоматизованих системах можна умовно поділити на три рівні. За основу поділу визначено такий критерій, як середовище, в якому знаходиться інформація:

– соціальне середовище (окрема людина, спільноти людей, держава);

– інженерно-технологічне (машинне, апаратно-програмне, автоматичне) середовище;

– людино/машинне (автоматизоване) середовище.

Важливим елементом організації інформаційної безпеки – захисту інформації – є поділ заходів на групи щодо протидії. В теорії і практиці майже однозначно виділяють три такі групи [3]:

– активні засоби захисту (розвідка, дезінформація, зашумлення тощо);

– пасивні засоби захисту (встановлення екранів несанкціонованого витоку інформації);

– комплекс засобів захисту (органічне поєднання вищевказаних груп).

Для реалізації наведених вище заходів захисту доцільно використовувати універсальні механізми захисту інформації [9].

До числа таких механізмів відносяться:

– ідентифікація, аутентифікація і авторизація суб’єктів;

– контроль доступу;

– реєстрація й аналіз подій в системі;

– контроль цілісності ресурсів системи.

Недоліки зазначених методик полягають або в їх громіздкості, тобто надмірній кількості операцій по дослідженню ймовірності ризиків, або у недостатності оцінювання, якщо методика розглядає тільки певну категорію ризиків, а не комплекс всіх можливих загроз.

Оцінка рівня безпеки інформаційних ресурсів є одним з найважливіших етапів проектування системи захисту інформаційної системи від ін'єкційних атак, оскільки вона дозволяє оцінити фактичний рівень захищеності інформації і необхідні фінансові витрати на створення системи захисту. В якості базових параметрів моделі можуть бути використані:

– величина ймовірності порушення безпеки інформації за фіксований час;

– характеристики ймовірності ін'єкційної атаки (щільність розподілу ймовірності, характеристичні функції, інтегральні показники тощо);

– ймовірний час безпечного функціонування інформаційного ресурсу при заданій довірчій вірогідності;

– потік ін'єкційних атак, що оцінюється математичним очікуванням інтервалу між сусідніми порушеннями і щільністю розподілу;

– функція відновлення інформації в часі, що оцінюється аналогічним чином.

Оскільки параметрів може бути дуже багато, виберемо з цих факторів три основні, які зможуть бути враховані в моделі захисту.

2.2 Синтез моделі захисту від ін'єкційних атак

Оскільки найбільш релевантну інформацію для моделі можемо отримати за допомогою імітаційного моделювання ін'єкційних атак, то у якості параметрів для оцінки рівня захисту від ін'єкційних атак візьмемо три основні показники – ймовірність ін'єкційної атаки у визначений проміжок часу, яка визначається за допомогою імітаційної моделі, час реакції після виявлення ін'єкції в запиті, а також рівень важливості ресурсу для працездатності системи. Кожен з показників складає безпосередній вплив на захищеність системи від ін'єкційних атак, яку беремо за

результативну ознаку. Таким чином, потрібно дослідити зв'язок між наступними параметрами:

Незалежні змінні:

X1 – ймовірність ін’єкційної атаки в HTTP-запиті за фіксований час, %

X2 – час реакції на отримання інфікованого запиту, с

X3 – важливість працездатності вузла, на який здійснювалась атака, %

Залежна змінна:

Y – рівень захищеності від ін’єкційної атаки, %.

Для побудови моделі та оцінки її параметрів було використано табличний процесор Excel.

MS Excel – одна із найпопулярніших програм електронних таблиць. В Excel є низка статистичних функцій та інструменти з надбудови «Пакет аналізу».

Таким чином, можемо використовувати дану програму для оцінки параметрів багатофакторної моделі.

Для отримання вхідних значень рівня захисту від ін’єкційних атак було проведено імітаційне моделювання на базі СМО (системи масового обслуговування). Систему було розгорнуто на підприємстві "Центр транспортної логістики" публічного акціонерного товариства "Українська залізниця" під час проходження науково-дослідної практики. При цьому використовувався методологічний підхід аналогічний використаному у попередніх дослідженнях Коваленко О.В. [16]. За результатами моделювання були отримані результати, які наведені у таблиці 2.1.

Таблиця 2.1

Вхідні умови дослідження рівня захисту від ін’єкційних атак

Порядковий номер модельованої атаки	Показники рівня захисту			Результат ступеню рівня захисту від ін’єкційних атак
	X1	X2	X3	
1	28	50	56	52
2	36	52	54	56

3	30	54	40	50
4	39	58	56	64
5	38	54	44	56

В результаті розраховані коефіцієнти кореляції між показниками ймовірності атаки та її наслідків і показником якості захисту від атак (табл. 2.2).

Таблиця 2.2

Результати розрахунків коефіцієнтів кореляції Пірсона r_{xy} між факторними ознаками *x* та *y*

Зв'язок факторних ознак	Коефіцієнт кореляції Пірсона r_{xy}
x_1-*y*: ймовірність ін’єкційної атаки – захист від атаки	0,52
x_2-*y*: час реакції – захист від атаки	0,74
x_3-*y*: важливість вузла – захист від атаки	0,83

Аналіз отриманих даних показав (див. табл. 2.1), що чисельне значення коефіцієнта кореляції 0,52 між частотою атак і рівнем захисту виявляє відносно слабкий їх кореляційний зв'язок, тобто захист має спрацьовувати на кожну атаку, незалежно від високої чи низької ймовірності її настання.

Тоді як рівень важливості має найбільший коефіцієнт кореляції (див. табл. 2.2), бо якщо атака здійснювалась на менш важливий вузол, то її наслідки будуть незначними, наприклад, уповільниться обмін даними між внутрішніми ресурсами, а якщо атака здійснювалась на вузол, від якого залежить працездатність всього ресурсу, наслідки атаки будуть більш критичними і тому рівень захисту потрібен вищий.

Час реакції також має досить важливе значення, адже ін’єкційна атака може мати на меті й уповільнення роботи сервера з запитами, тому швидкість реагування на атаку та її знешкодження мають досить високе значення.

Аналіз кореляційних зв’язків між кількома ознаками (спільним впливом показників ін’єкційних атак на результати успішності захисту) показав у таблиці 2.3 тісноту зв’язків між ними.

Таблиця 2.3

Результати розрахунків сукупних коефіцієнтів множинної кореляції між факторними ознаками x_1, x_2, x_3 та y

Зв'язок факторних ознак	Коефіцієнт множинної кореляції
x_1-x_2-y: ймовірність атаки та час реакції на неї – рівень захисту	0,89
x_1-x_3-y: ймовірність атаки на важливий вузол – рівень захисту	0,93
x_1-x_2-x_3-y: якість реагування на атаки – рівень захисту	0,97

Як можна побачити з приведених вище даних, коефіцієнт множинної кореляції рівня захисту з показниками реагування на атаки (береться до уваги саме одночасний вплив усіх перерахованих факторів) дещо збільшується.

А тепер повернемося до багатофакторної регресійної моделі, приведеної на початку, та підставимо до формули отримані факторні ознаки та обчислені коєфіцієнти кореляції, в результаті чого отримаємо, що якість захисту від ін’єкційних атак усіх перелічених видів має вплив на кінцевий результат рівний 0.92, іншими словами 92% успіху відбиття атаки складає якість системи своєчасного реагування та виявлення ін’єкції.

Тобто, беручи до уваги значення коефіцієнту кореляції Пірсона, можна зробити висновок про сильну залежність розглядуваних понять [4]. Тобто в даній залежності нехтування обраними параметрами недопустимо.

Отже, можемо побудувати багатофакторну модель оцінки якості захисту від ін’єкційної атаки на основі трьох незалежних ознак.

Множинна регресія це статистичний метод аналізу зв'язку між залежною змінною y і множиною змінних (x1, x2,…xn), що слугує для вибору

незалежних змінних у порядку їхньої значимості. Цей метод реалізує в найбільш простому варіанті структурну ідентифікацію, оскільки з деякого заданої множини регресивних залежностей вибирається єдина, яка найкраще відповідає прогнозованим даним.

Для побудови багатофакторної регресійної моделі потрібно:

1) Ідентифікувати змінні моделі

Загальний вигляд математичної моделі:

$$Y = f(X_1, X_2, X_3) \quad (2.1)$$

Y – рівень захищеності (залежна змінна), X_1 – ймовірність ін'єкційної атаки протягом певного часу (незалежна пояснювальна змінна), X_2 – час реакції після атаки (незалежна пояснювальна змінна), X_3 – важливість ресурсу, у який здійснювалась ін'єкція (незалежна пояснювальна змінна)

2) Специфікувати модель

Специфікація моделі — це аналітична форма математичної багатофакторної яка базується на досліджуваних чинників. За основу беруться ймовірнісні характеристики, для яких є притаманні стохастичні залишки моделі.

Ми будуємо лінійну модель залежності значення Y

$$Y = a_o + a_1X_1 + a_2X_2 + a_3X_3 \quad (2.2)$$

3) Оператор оцінювання параметрів моделі за 1МНК має вигляд

$$A = (X'X)^{-1}X'Y \quad (2.3)$$

Звідси: a0 = 0,967, a1 = 0,640, a2 = 0,343, a3 = 0,287.

$$\hat{Y} = 0{,}967 + 0{,}640X_1 + 0{,}343X_2 + 0{,}287X_3 \quad (2.4)$$

- вигляд математичної за МНК.

Можемо зробити висновок, що коли за всіх однакових умов незалежна змінна X1 – ймовірність атаки збільшується на один відсоток, то залежна змінна Y – рівень захисту від атаки також повинна збільшитись на 0,64%. Відповідно, за цих однакових умов, якщо незалежна змінна X2 – час реакції після атаки збільшується на 1 с, то успішність рівня захисту також повинна бути збільшена на 0,343%. Якщо X3 – важливість вузла / ресурса занять збільшиться на один відсоток, то рівень його захисту також збільшується на 0,287%.

Кореляційна матриця має вигляд:

	Y	*X1*	*X2*	*X3*
Y	*1*			
X1	*0,837121*	*1*		
X2	*0,741215*	*0,692106*	*1*	
X3	*0,522913*	*0,122239*	*-0,09009*	*1*

(2.5)

Тобто найбільш тісний зв'язок є між У та x_1 і x_2. З x_3 зв'язок менш вагомий.

Таким чином, було отримано багатофакторну модель залежності потрібного рівня захисту від атак від конкретних характеристик кожного ресурсу, на який може бути здійснено атаку.

2.3 Перевірка моделі на адекватність

Визначаємо матрицю коваріацій, стандартні похибки та даємо інтервальну оцінку параметрам моделі.

Знаходимо $\hat{Y}$ за формулою

$$\hat{Y} = 0{,}967 + 0{,}640X_1 + 0{,}343X_2 + 0{,}287X_3 \quad (2.6)$$

Обчислимо дисперсію похибки з урахуванням числа ступенів свободи Du = sum $\hat{u}^2/(n-m)$

Дисперсія без урахування числа ступенів свободи Du1= sum $\hat{u}^2/n$

Стандартне відхилення похибки від дисперсії з урахуванням ступенів свободи Su – корінь квадратний з Du, Su = 2,5435, Du = 6,4691, Du1=1,2938

Будуємо матрицю коваріації.

Коваріаційна матриця має вигляд:

Cov (Â)

	a^0	a^1	a^2	a^3
a^0	23,04	17,68	9,44	16,8
a^1	17,68	19,36	8,08	16,8
a^2	9,44	8,08	7,04	-1,6
a^3	16,8	3,6	-1,6	44,8

(2.7)

де – A0, a1, a2, a3 – оцінки параметрів моделі.

У головній діагоналі матриці коваріації знаходяться дисперсії оцінок параметрів моделі.

Знайдемо стандартну похибку Sa^j

A0= 1,80;

A1= 0,40;

A2= 0,195;

A3= 0,097.

A0 – нестійка оцінка параметрів, тобто статистично незначуща.

Оцінюємо достовірність моделей, використавши:

- коєфіцієнти детермінації і кореляції;
- критерій Фішера (F- критерій);

Будуємо статистику в Excel і отримуємо оцінку даних:

M(u)=0

$$R^2=(Dy-Du/Dy) \quad (2.8)$$

– коефіцієнт детермінації

Далі шукаємо дисперсії по у

Dy – дисперсія з урахуванням числа ступенів свободи

$$Dy = sum\ (yi-yser)^2/\ n-1 = 6{,}063 \quad (2.9)$$

Dy1- дисперсія без урахування числа ступенів свободи

$$Dy1 = sum\ (yi-yser)^2/\ n = 5{,}760 \quad (2.10)$$

Dregr – дисперсія регресії для F-критерію

$$Dregr = sum\ (y^i-yser)^2/\ m-1 = 28{,}8 \quad (2.11)$$

R^2 = 0,95 – коефіцієнт детермінації

Це вказує на високий рівень достовірності. Коефіцієнт детермінації з урахуванням числа ступенів свободи показує, що на 95% варіація у визначається варіацією х. А лише 5 %- інші фактори.

Отже, залежність між х та у дуже висока.

F-критерій розраховується від оцінки статистичної значущості математичної моделі в цілому.

$$F = Б_{\chi}^2/Б_{\gamma}^2 \quad (2.12)$$

Також висуваємо 2 гіпотези: Н0 – математична модель статистично недостовірна і Н1 – математична модель статистично достовірна

F факт. > Fтабл. – гіпотеза Н1;

F факт. < Fтабл – гіпотеза Н0;

F факт = 4,75;

Ступінь свободи 1 = m-1 =3;

Ступінь свободи 2 = n-m = 16;

F табл. = 3, 2388.

F табл менше, ніж F факт, що означає, що математична модель статистично достовірна в цілому.

Отже, розроблена багатофакторна модель може використовуватись для оцінки ступені захищеності від ін’єкційних атак.

Висновки до другого розділу

Таким чином, було обрано та обґрунтовано розроблено параметри для моделі оцінки потреби у рівні захисту.

Незалежні змінні:

X1 – ймовірність ін’єкційної атаки в HTTP-запиті за фіксований час, %

X2 – час реакції на отримання інфікованого запиту, с

X3 – важливість працездатності вузла, на який здійснювалась атака, %

Залежна змінна:

Y – рівень захищеності від ін’єкційної атаки, %.

А також, сама модель оцінки потреби у рівні захисту для кожного вузла/ресурсу в залежності від трьох основних параметрів – ймовірності атаки на даний вузол, часу реакції вузла на спробу атаки і ступеню важливості даного ресурсу для функціонування всієї системи, на яку здійснюються атаки. Дана модель є лінійною і показує потребу у рівні захисту від атак для кожного об’єкта системи, тобто налаштовуваний параметр виявлення ін’єкції змінюється для кожного окремого об’єкта в залежності від наслідків, до яких може призвести атака на цей об’єкт. Дана модель дозволяє динамічно реагувати на можливі атаки і зменшувати навантаження на систему захисту в цілому, розмежовуючи рівень потреби у захисті для кожного елемента системи окремо. Практична реалізації моделі захисту від ін’єкційних атак може бути виконана як у статичному форматі,

тобто налаштування проводяться за результатами разової оцінки вручну, чи у динамічному автоматичному форматі в залежності від кількості об’єктів, що підлягають захисту, а також рівня загроз для кожного з них.

РОЗДІЛ 3
ПРАКТИЧНЕ ЗАСТОСУВАННЯ РОЗРОБЛЕНОЇ МОДЕЛІ

3.1 Методика проведення експериментальних досліджень

Експериментальне дослідження проводилось шляхом імітації ін'єкційної атаки і використання засобів її ідентифікації та протидії. Спочатку розраховується рівень потреби у захисті за побудованою моделлю. Якщо рівень нижче 50%, то активного захисту не потрібно. На рівні від 50 до 60% потрібно контролювати окремі параметри, які можуть вказувати на можливість ін'єкції. Якщо рівень вище 60%, то потрібно ввімкнути на зазначених ресурсах моніторинг вхідних запитів на вимогу. Якщо рівень вище 80%, доцільно встановити більш спеціалізоване програмне забезпечення, яке буде аналізувати кожен запит.

Розглянемо, як проводиться моніторинг запитів у випадку рівня небезпеки вище 60%.

Збір статистичної інформації проводиться за допомогою програми-аналізатора трафіку. Базова інформація про проведення експерименту наведена в табл. 3.1.

Таблиця 3.1

Параметри моделювання SQL-атак

№	кількість пакетів	Час передачі пакетів, хв	
		Звичайний режим	Режим атаки
1	50000	15	20
2	100000	30	35
3	150000	60	65
4	200000	75	86
5	250000	84	98

Для проведення експериментального моделювання SQL-атак була створена розподілена мережа і використовувалися програми netmap, nping та WPEPro. Під

час дослідження на мережеві сервіси впливають SQL-атак різного рівня інтенсивності – в HTTP-запити вбудовуються ін'єкційні елементи, і система повинна на основі побудованої моделі визначити ступінь ризикованості того чи іншого запиту і відповідним чином відреагувати – відхилити запит при високому рівні ризику наслідків атаки, відправити його на більш детальний аналіз на предмет вмісту ін'єкційної складової чи пропустити, оцінивши запит як безпечний.

З метою досягнення максимального наближення до реальних умов функціонування мережевих сервісів, при проведені експерименту трафік складався, як з «інфікованих» запитів так і реального мережевого трафіку.

Після цього було проведено розрахунок ентропії для кожної послідовності з чого була отримана можливість визначити чи відбувається атака (табл. 3.2).

Таблиця 3.2

Значення ентропії для послідовностей пакетів

№	Значення ентропії H без атаки, біт	Значення ентропії H* при загрозі ін'єкційної атаки, біт
1	3,39	1,88
2	3,32	1,28
3	3,28	1,51
4	3,25	1,71
5	3,21	1,93

Таким чином, за допомогою обчислення ентропії пакету HTTP-зпитів можна вловити початок ін'єкційної атаки і своєчасно запобігти її наслідкам. Тобто було доведено, що інформаційний аналіз трафіку в режимі реального часу може бути використаний для ідентифікації аномальної активності і надходження запитів сумнівного вмісту. Зокрема, значення ентропії трафіку є чутливими до SQL-атак, із розвитком ін'єкційної атаки її значення зменшуються майже на 70%, оскільки запити є більш довгими за рахунок додаткових SQL-рядків. При цьому необхідно відзначити: на результати обчислень не повинен впливти розмір пакету. У проведеному експерименті на ентропію впливає тільки факт надходження пакету з

попередньо визначеними параметрами, тобто зі збільшеним розміром чи вмістом рядків з ключовими словами. Відповідно, обчислення ентропії трафіку для визначення початку ін'єкційної атаки можна застосовувати наприклад у системах виявлення вторгнень IDS, системах підтримки прийняття рішень, системах управління інформаційною безпекою підприємства тощо.

Для захисту від атак на середньому рівні можливо використовувати схеми проходження запитів тільки визначеної довжини. Цього можна досягти використанням спеціалізованих програм з врахуванням топології мережі та шляхів проходження запитів.

Створення спеціалізованих програм аналізу та синтезу топологій окремих мережевих ресурсів насамперед вимагає розроблення додаткових математичних моделей цих структур, які мають не тільки адекватно описували систему «запит - сервер», але й просто та ефективно представлялися для подальшої роботи з ними, а також розроблення відповідних обчислювальних методів роботи з підозрілими запитами. Для цього можна вдатися до застосування графів як математичної моделі топології. Для роботи з графами створено велика кількість методів виявлення шляхів між вершинами, виявлення циклів та контурів, дерев і таке інше.

Крім того існує потреба в всеохоплюючому підході до розробки матричних методів аналізу та синтезу топологій проходження HTTP-запитів. Отже, науково-технічна проблема системного аналізу та синтезу топологій комунікаційних систем та створення обчислювальних методів опрацювання матричних моделей топологій для побудови ефективних програмних засобів проектування таких систем є актуальною.

При розгляданні яким чином повинна бути представлена вищезазначена двоелементна система, таблицю можна представити у вигляді біграфу, один сегмент якої – користувачі, інший – надіслані ними запити. Ці сегменти з'єднуються один з одним зв'язки із значенням «підозрілий запит». Відповідно пропонується сформувати матрицю цих зв'язків.

Аналіз графів можна проводити в цілому або для окремого ресурсу з пошуком оптимальної моделі проходження запитів дозволеної довжини. Розглянемо

приклад аналізу графу для пошуку оптимального шляху інформації. Нехай існує потреба проходження запитів до ресурсу (1) за участю декількох користувачів, деякі з яких надсилають запити, що перевищують середню довжину. Цифри над стрілками позначають кількість можливих шляхів проходження запитів.

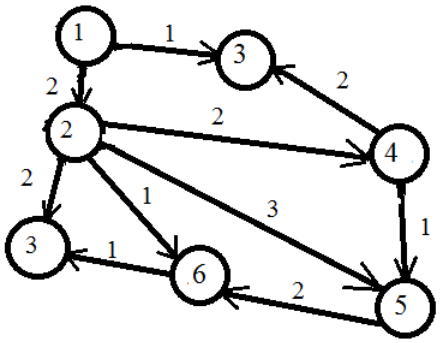

Рисунок 3.2 – Граф надходження запитів від різних користувачів

Розрахуємо показник складності для даного графу Зобразимо даний граф у вигляді ієрархічного графа:

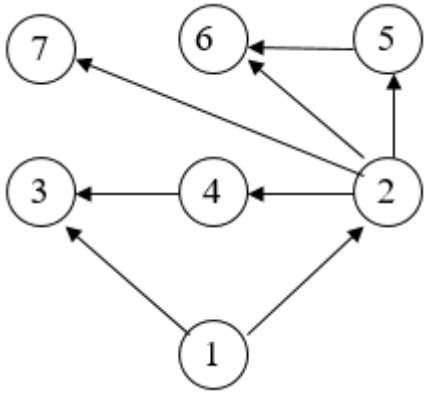

Рисунок 3.3 – Ієрархічний граф комунікаційної мережі

Отриманий трирівневий граф перетворюємо в еквівалентний, що не містить суміжних вершин на одному рівні

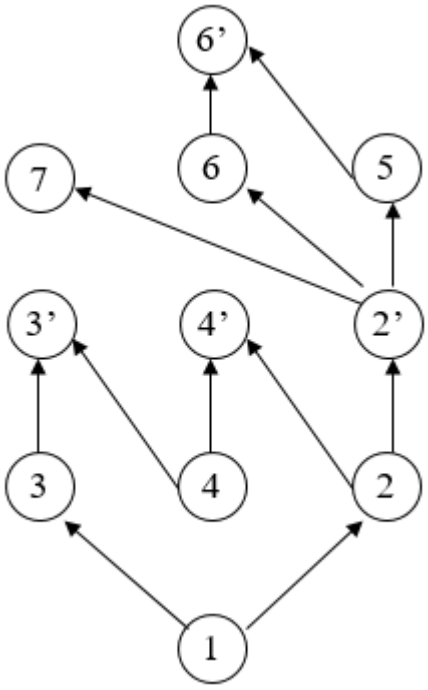

Рисунок 3.4. – Еквівалентний граф проходження запитів по мережі

Отриманий граф описується такими матрицями інциденцій:

$$W_1 = \begin{array}{c|ccc} & 2 & 3 & 4 \\ 1 & 1 & 1 & 0 \end{array} \quad (3.1)$$

$$W_2 = \begin{array}{c|ccc} & 2` & 3` & 4` \\ 2 & 1 & 0 & 1 \\ 3 & 0 & 1 & 0 \\ 4 & 0 & 1 & 1 \end{array} \quad (3.2)$$

$$W_3 = \begin{array}{c|ccc} & 5 & 6 & 7 \\ 2` & 1 & 1 & 1 \\ 3` & 0 & 0 & 0 \\ 4` & 0 & 0 & 0 \end{array} \quad (3.3)$$

$$W_4 = \begin{array}{c|c} & 6` \\ 5 & 1 \\ 6 & 1 \\ 7 & 0 \end{array} \quad (3.4)$$

$$W = W_1W_2W_3W_4 = \begin{array}{c|ccc} & 2 & 3 & 4 \\ \hline 1 & 1 & 1 & 0 \end{array} * \begin{array}{c|ccc} & 2` & 3` & 4` \\ \hline 2 & 1 & 0 & 1 \\ 3 & 0 & 1 & 0 \\ 4 & 0 & 1 & 1 \end{array} W_3W_4 =$$

$$= \begin{array}{c|ccc} & 2` & 3` & 4` \\ \hline 1 & 1 & 1 & 1 \end{array} * \begin{array}{c|ccc} & 5 & 6 & 7 \\ \hline 2` & 1 & 1 & 1 \\ 3` & 0 & 0 & 0 \\ 4` & 0 & 0 & 0 \end{array} W_4 = \begin{array}{c|ccc} & 5 & 6 & 7 \\ \hline 1 & 1 & 1 & 1 \end{array} * \quad (3.5)$$

$$\begin{array}{c|c} & 6` \\ \hline 5 & 1 \\ 6 & 1 \\ 7 & 0 \end{array} = 2$$

Для графа на рис. 3.3 знайдемо максимальний шлях з вершини a_1 у вершину a_7.

Для вершини a_1 приймаємо

$$q_s^{макс}(a_1a_1) = 0. \quad (3.6)$$

$$a_2, a_3 : q_s^{макс}(a_1a_2) = 2,\ q_s^{макс}(a_1a_3) = макс(1, 2+2+2) = 6 \quad (3.7)$$

$$a_4 : q_s^{макс}(a_1a_4) = 2+2 = 4. \quad (3.8)$$

$$a_5 : q_s^{макс}(a_1a_5) = макс(2+3, 2+2+1) = 5. \quad (3.9)$$

$$Д\ a_6 : q_s^{макс}(a_1a_6) = макс(2+1, 5+2) = 7. \quad (3.10)$$

$$a_7 : q_s^{макс}(a_1a_7) = макс(2+2, 7+1) = 8. \quad (3.11)$$

Значення функції на максимальному шляху дорівнює восьми пунктам, а сам шлях складається з пунктів 1-2-5-6-7, тобто оптимальний шлях передачі запита від користувача 1 до ресурсу 7 за заданими критеріями проходить через вузли 2, 5 і 6. Інші вузли в цей момент відсіюють запити, які перевищують задану довжину.

Також можна подати інформацію про наявність чи відсутність сумнівних запитів через побудову біографу (двокомпонентного графу), тоді можливо наочно бачити не тільки кількість проблемних запитів, а й їх якість, тобто які саме проблеми були вирішені автоматично, а які – ні.

Отже, дана модель дозволяє відслідковувати довжину запитів і вирішувати конфлікти у випадку виявлення перевищення окремих запитів від різних користувачів. Таким чином, це дозволяє запобігти втраті інформаційних ресурсів, оскільки ведеться облік всіх запитів, з врахуванням дати проходження сумнівного запиту та користувача, який ці зміни вніс, щоб за потреби визначити ризиковість користування інформаційним ресурсом з боку окремих користувачів

3.2 Дослідження характеристик моделі захисту від ін’єкційних атак

Для використання моделі можливо введення додаткових параметрів у процес ідентифікації загрози ін’єкційної атаки і врахування їх впливу на підсумкові дії по захисту від атаки.

Серед параметрів, які застосовувались до вибору середовища проведення експерименту, були визначені наступні параметри:

X1 – ймовірність ін’єкційної атаки в HTTP-запиті за фіксований час, %

X2 – час реакції на отримання інфікованого запиту, с

X3 – важливість працездатності вузла, на який здійснювалась атака, %

Кожен з показників складає безпосередній вплив на захищеність системи від ін’єкційних атак, яку беремо за результативну ознаку.

Таблиця 3.3

Умови проведення експериментального дослідження

Показники рівня захисту			Результат ступеню рівня захисту від ін’єкційних атак
X1	X2	X3	
37	59	66	64
38	57	67	64

Отже, за допомогою моделі можливо оцінити рівень захисту від атаки під час експерименту і через заздалегідь відомі параметри:

$$\hat{Y}_{КЗ} = 0{,}967 + 0{,}640 \cdot 37 + 0{,}343 \cdot 59 + 0{,}287 \cdot 66 = 63{,}9\% \quad (3.12)$$

$$\hat{Y}_{ЕЗ} = 0{,}967 + 0{,}640 \cdot 38 + 0{,}343 \cdot 57 + 0{,}287 \cdot 67 = 64{,}1\% \quad (3.13)$$

Отже, оцінка за моделлю співпадає з фактичним значенням з точністю до 0,1%, що підтверджує ефективність моделі і дозволяє її застосовувати в подальшому для прогнозування рівня захисту від атак.

Тобто, в даній моделі розраховується ймовірність того, що інформаційний ресурс не зможе протистояти діям агенту загрози. Значення розраховується на основі показників можливості загрози і сили захисних засобів. Результатом виконання даного кроку є значення уразливості, яке розраховується за допомогою матриці визначення уразливості, поданої в табл. 3.4 [32].

Таблиця 3.4

Матриця визначення ймовірності вразливості

Вразливість						
Можливість загрози	**ДВ**	ДВ	ДВ	ДВ	В	ДВ
	В	ДВ	ДВ	В	С	Н
	С	ДВ	В	С	Н	ДН
	Н	В	С	Н	ДН	ДН
	ДН	С	Н	ДН	ДН	ДН
		ДН	**Н**	**С**	**В**	**ДВ**
	Сила захисних засобів					

Також враховується імовірна частота ін’єкційної атаки за певний період часу. Значення розраховується на основі частоти подій ризику та вразливості, отриманих під час аудиту інформаційного ресурсу. Результатом цього кроку є значення частоти подій, що призводять до втрат, яке розраховується за допомогою матриці частоти подій втрат, наведеної у табл. 3.5 [32].

Таблиця 3.5

Матриця визначення частоти подій, що призводять до негативних наслідків

Частота подій, що приводить до негативних наслідків						
Частота подій реалізації загрози	**ДВ**	С	В	ДВ	ДВ	ДВ
	В	Н	С	В	В	В
	С	ДН	Н	С	С	С
	Н	ДН	ДН	Н	Н	Н
	ДН	ДН	ДН	ДН	ДН	ДН
		ДН	**Н**	**С**	**В**	**ДВ**
	Вразливість					

Після цього відбувається встановлення походження ризику ін’єкційної атаки, чітке формулювання критеріїв моніторингу та оцінка величини потенційних збитків у випадку успішності атаки.

Таблиця 3.6

Матриця визначення величини ризику

Ризик						
Вірогідна величина втрати	Критична	В	В	К	К	К
	Висока	С	В	В	К	К
	Значна	С	С	В	В	К
	Середня	Н	С	С	В	В
	Низька	Н	Н	С	С	С
	Дуже низька	Н	Н	С	С	С
		ДН	Н	С	В	ДВ
	Частота подій, що приводить до втрат					

3.3 Розробка методики захисту від ін’єкційних атак

Як було зазначено вище існуючі системи умовно розділяються на: виявлення аномалій і виявлення ознак. Головним недоліком систем на виявленні ознак є, те що вони базуються на виявленні вже відомих типів атак. Однак перелік можливих

загроз і їх методів передачі постійно збільшується і такі системи швидко застарівають. Системи виявлення аномалій базуються на припущені, що система повинна працювати певним чином, наприклад, що повинна зберігатись статична однорідність трафіку. Пропонується побудувати систему захисту на основі розкритих вище елементів:

– агенти нагляду;

– засоби попередньої обробки та зберігання;

– репозиторій для зберігання інформації про транзакції, що описує роботу системи;

– репозиторій з аналітичними компонентами для виявлення загроз та ознак зловмисної діяльності;

– контратакуючі агенти.

Важливим елементом побудови такої системи є визначення відповідного математичного забезпечення для кожного етапу роботи [14]:

1. Стеження за трафіком.

2. Попередня обробка перехоплених пакетів, оцінка загроз, збереження інформації про випадок атаки.

3. Аналіз даних при завантаженні, виявлення атак, оцінка загроз.

4. Фоновий аналіз даних для встановлення спроб сканування, атак погіршення якості, пульсуючих атак.

5. Прийняття рішення про виявлення атаки.

6. Оцінка загрози, вибір моделі, її верифікація та пошук стратегії.

Для створеня стратегії протидії необхідно провести оцінювання параметрів динамічної моделі. До цього процесу входить:

1. Аналіз динаміки.

2. Виявлення кількості зловмисників та потужностей нападу.

3. Аналіз загрози.

4. Визначення методів протидії та створення прогнозів що до наслідків.

5. Застосування методів протидії та порівняння наслідників з прогнозами.

Після застосування протидії системою необхідно провести оцінку ефективності стратегії. При продовженні атаки стратегія підлягає переробці [14].

Усі заходи захисту від ін’єкційних атак можна поділити двома категоріями:

– пасивні та активні;

– превентивні та реакційні.

Отже, спочатку розраховується рівень потреби у захисті за побудованою вище моделлю оцінки потреби у рівні захисту. Якщо рівень нижче 50%, то активного захисту не потрібно. На рівні від 50 до 60% потрібно контролювати окремі параметри, які можуть вказувати на можливість ін’єкції. Якщо рівень вище 60%, то потрібно ввімкнути на зазначених ресурсах моніторинг вхідних запитів. Якщо рівень вище 80%, доцільно встановити більш спеціалізоване програмне забезпечення, яке буде аналізувати кожен запит.

Розглянемо, як проводиться моніторинг запитів у випадку рівня небезпеки вище 60%.

На рис. 3.1 наведена блок схема виявлення та зупинення SQL -атаки.

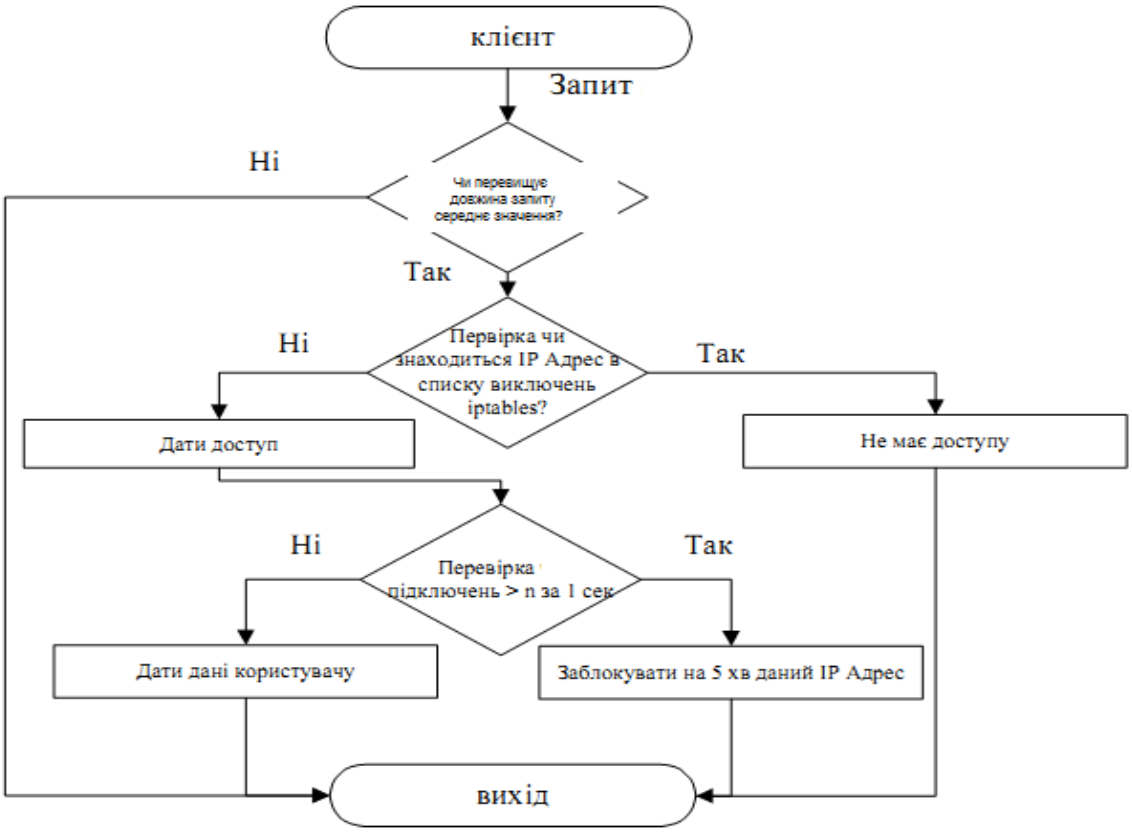

Рисунок 3.1 – Алгоритм виявлення та протидії SQL -атакам [17]

З самого початку перевіряється чи йде збільшення тривалості з’єднань до сервера, ніж це відбувається зазвичай, або збільшення довжини HTTP-запита. Для цього доцільно використовувати експериментально перевірену вище методику детекції ін’єкційних атак.

Якщо ж все-таки існує перевищення, то це можна вважати підставо для подальшої перевірки чи на існування атаки на сервер і чи містять HTTP-запити ін’єкцію. Перш за все перевіряється відхилення від довжини і чи знаходяться IP адреси підозрілих запитів у списку виключень. Після цього підраховується довжина запитів з кожного IP адресу до сервера, що підлягає захисту. Якщо отримане значення більше, ніж n, тоді можна вважати, що йде ін’єкційна атака. Щоб припинити надходження запитів з шкідливим кодом, варто заблокувати підозрілі IP-адреси на термін від 5 хв. У цей час перевірено структуру запитів і зупинено ін’єкційну атаку блокуванням HTTP-запитів з даних адрес. Дана схема є дієвою для всіх типів атак. Однак для кожного типу атаки може відрізнятися число n, від якого вираховується перевищення часу проходження запиту чи його довжини [17].

Аналогічним чином можна перевіряти час проходження запиту, який також відрізнятиметься від стандартного, оскільки містить додатковий код.

Коли є підозра, що йде SQL-атака, усі дані про з’єднання варто записуємо в окремий лог. В ньому необхідно виокремити час і вхідна IP-адреса, щоб надалі перевіряти дані адреси. Після чого необхіодно провести аналіз даних і вилучити адреси з яких, кількість запитів з яких перевищує задане число. Потім виявлені адреси блокуються в iptables.

Крім того, можна подати інформацію про наявність чи відсутність сумнівних запитів через побудову біографу (двокомпонентного графу), тоді можливо наочно бачити не тільки кількість проблемних запитів, а й їх якість, тобто які саме проблеми були вирішені автоматично, а які – ні.

Також у випадку виявлення частих ін’єкційних атак на окремий ресурс можна скористатись засобами інженерно-технічного захисту.

Для різних об’єктів захисту ставляться різні цілі і задачі для виконання яких приймаються різні заходи, тому необхідно розглянути деяку систему класифікації.

Наприклад, засоби ІТЗ можна розглядати за об’єктами на які здійснюється вплив. Для цього вони застосовуються для захисту користувачів, ресурсів, фінансів, інформації тощо.

Існує багато різноманітних характеристик, що можуть слугувати для кваліфікації ІТЗ:

– об’єкти впливу;
– характер заходів;
– способи реалізації;
– масштабом охоплення;
– засоби зловмисників.

Очевидно, що такий поділ засобів захисту інформації має досить умовний характер, практиці вони здебільшого реалізуються в комплексі.

Після цього проводиться підрахунок вірогідної величини втрат внаслідок атаки.

Оцінити величину втрати в найгіршому випадку можливо, виконуючи наступні три пункти:

– визначити дію загрози, що скоріш за все призведе до результату найгіршого випадку;
– визначити величину для кожної форми втрати, пов’язаної з тією дією загрози;
– підсумувати величини кожної форми втрат.

Оцінити ймовірну величину втрат можна наступним чином:

– встановити найбільш вірогідну дію(ї) множини загроз;
– визначити величину втрат для кожної форми втрат;
– підсумувати величини.

Саме на цьому етапі відбувається зіставлення якісних і кількісних показників.

Після цього відбувається встановлення походження ризику ін’єкційної атаки, чітке формулювання критеріїв моніторингу та оцінка величини потенційних збитків у випадку успішності атаки.

Фактично на даному кроці підсумовуються усі отримані на попередніх кроках дані. На основі отриманих даних встановлюються ймовірна частота ін’єкційних атак та ймовірна величина майбутніх втрат.

Добре сформульовані параметри ризику забезпечать осіб, які приймають рішення щодо засобів захисту від ін’єкційних атак, принаймні двома ключовими елементами інформації:

– підрахована частота подій, що призводить до втрат;

– підрахована вірогідна величина втрат.

Ця інформація може бути виражена у різній формі. У більшості випадків бажано також надавати підрахований потенціал щодо втрат найвищого рівня для того, щоб особа, яка приймає рішення, була проінформована щодо сценарію втрат у найгіршому випадку.

Таблиця 3.7

Схема оцінки вірогідної величини втрат у разі дії загроз

	Види втрат					
Дія загрози	Продуктивність	Реакція	Заміна	Штрафи,рішення	Рекламакомпанії	Репутація
Отримання НСД						
Збій в роботі системи						
Розголошення інформації						
Модифікація ресурсу						
Відмова в доступі						

Тобто на цьому етапі мають бути чітко визначені наступні величини: частота подій, що приводить до втрат; вірогідні втрати; втрати в найгіршому випадку.

За допомогою табл.3.7 встановлюється величина ризику, а її значення розшифровується за допомогою таблиці 3.8 [14].

Таблиця 3.8

Значення рівня ризику втрати інформаційних ресурсів

Значення	Рівень ризику
К	Критичний (більше 80;)
В	Високий (більше 60%)
С	Середній (більше 50%)
Н	Низький (менше 50%)

В окремих випадках, для уточнення оцінки інформаційних ризиків, може бути застосований підхід, заснований на переході від якісної до кількісної оцінки інформаційних ризиків.

Висновки до третього розділу

Таким чином, були проведені експериментальні дослідження та дослідження характеристик моделі захисту від ін’єкційних атак. А також, була розроблена методика захисту від ін’єкційних атак, яка складається з моделі оцінки потреби у рівні захисту на основі якої може застосовуватись обґрунтована експериментальним дослідженням методика детекції ін’єкційних атак, алгоритму дій при виявленні атаки та запропонована методика оцінки ризиків від атак.

Ця методика може бути застосована, як основа при проектуванні систем захисту від ін’єкційних атак. Вона є універсальною щодо різних видів операційних систем та атак на них, може легко масштабуватись під потреби користувача, легка у використанні та ефективна у протидії ін’єкційним атакам на різних рівнях.

ВИСНОВКИ

Роботу присвячено дослідженню методик та моделей захисту інформаційних ресурсів від ін’єкційних атак в телекомунікаційних мережах. В роботі було поставлено та виконано наступні завдання:

– розглянуто поняття ін’єкційних атак і методи захисту від них. Серед усіх типів ін’єкційних атак виокремлено найбільш небезпечні, серед яких особливу увагу приділено SQL-інєкціям. Ін’єкційні атаки не тільки дозволяють проникнення у мережеві ресурси користувача з метою викрадення, знищення чи викривлення, а і дозволяють закріпитись в операційній системі користувача задля подальшого витоку інформації та контролю, а в особливо серйозних випадках і до передачі шкідливих запитів на інші об’єкти;

– проведено огляд існуючих моделей захисту від атак. Вони базуються на ідентифікації атаки та виборі методу захисту програмним, апаратним чи іншим засобом. Ідентифікація ін’єкційних атак відбувається шляхом аналізу вхідних запитів на довжину, вміст певних елементів, ключові слова чи символи та інші критерії оцінки, які дозволяють виявити підозрілі запити і більш детально проаналізувати їх за допомогою спеціального програмного забезпечення;

– здійснено постановку задачі дослідження. Задачею дослідження є синтез універсальних методів захисту від ін’єкційних атак різних видів, які мають враховувати існуючі механізми ін’єкції, реалізовувати заходи щодо «закриття дір», тобто нейтралізації вразливостей в існуючій системі, а також ідентифікувати спроби несанкціонованого доступу через ін’єкцію, і вживати заходів, адекватних загрозі. Таким чином, були розглянуті основні типи ін`єкційних атак, загрози, які вони створюють, а також методи ідентифікації таких атак в залежності від виду атаки. На основі проведеного дослідження стало зрозуміло, що більшість рекомендацій, сформованих науковцями з цього приводу, є розрізненими та не дозволяють ефективно протидіяти ін’єкційним атакам, тому це дозволило

сформулювати та конкретизувати задачі дослідження щодо синтезу більш універсальних моделей захисту від ін’єкційних атак;

– проведено вибір та обґрунтування параметрів розроблюваної моделі. Було обрано три основні фактори, які було використано в подальшій розробці моделі захисту від ін’єкційних атак;

– здійснено синтез моделі захисту від ін’єкційних атак. Було отримано три факторну модель та розрахували її кореляційні характеристики, що свідчать про відсутність автокореляції між обраними параметрами. Таким чином, було отримано багатофакторну модель залежності потрібного рівня захисту від конкретних характеристик кожного ресурсу, на який може бути здійснено атаку;

– проведено перевірку моделі на адекватність. В результаті роботи було розроблено модель оцінки потреби у рівні захисту для кожного вузла/ресурсу в залежності від трьох основних параметрів – ймовірності атаки на даний вузол, часу реакції на атаку і ступеню важливості даного ресурсу для функціонування всієї системи, на яку здійснюються атаки. Дана модель є лінійною і показує потребу у рівні захисту від атак для кожного об’єкта системи, тобто налаштовуваний параметр виявлення ін’єкції змінюється для кожного окремого об’єкта в залежності від наслідків, до яких може призвести атака на цей об’єкт. Дана модель дозволяє динамічно реагувати на можливі атаки і зменшувати навантаження на систему захисту в цілому, розмежовуючи рівень потреби у захисті для кожного елемента системи окремо. Практична реалізації моделі захисту від ін’єкційних атак може бути виконана як у статичному форматі, тобто налаштування проводяться за результатами разової оцінки вручну, чи у динамічному автоматичному форматі в залежності від кількості об’єктів, що підлягають захисту, а також рівня загроз для кожного з них;

– розроблено методику проведення експериментального дослідження. Вона полягала в формуванні пакетів запитів, які містять ін’єкцію та засобів реагування системи на загрозу. Під час експерименту виявлено додатковий чинник, як ентропія, який також може бути додано до моделі;

–виконано дослідження характеристик моделі захисту від ін’єкційних атак.

Таким чином, була розроблена методика захисту від ін’єкційних атак, яка складається з моделі оцінки потреби у рівні захисту на основі якої може застосовуватись обґрунтована експериментальним дослідженням методика детекції ін’єкційних атак, алгоритму дій при виявленні атаки та запропонована методика оцінки ризиків від атак.

Ця методика може бути застосована, як основа при проектуванні систем захисту від ін’єкційних атак. Вона є універсальною щодо різних видів операційних систем та атак на них, може легко масштабуватись під потреби користувача, легка у використанні та ефективна у протидії ін’єкційним атакам на різних рівнях. Таким чином було виконано всі поставленні завдання.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Ільницький А. Ю. Основи захисту інформації від несанкціонованого доступу / Д. Ю. Ільницький, В. А. Саницький, В. В. Порошев та ін. – К. : Національна академія внутрішніх справ України, 2002. – 208 с.

2. Інформаційна безпека (соціально-правові аспекти) : підруч. / за заг. ред. Є. Д. Скулиша. – К. : КНТ, 2010. – 776 с.

3. Інформаційна безпека (соціально-правові аспекти): підруч. / [В. В. Остроухов, В. М. Петрик, М. М. Присяжнюк та ін.] ; за заг. ред. Є. Д. Скулиша. – К. : КНТ, 2010. – 776 с.

4. Кошель А. О. Поняття ризику та його види при використанні земельних ресурсів у ринкових умовах [Електронний ресурс] / А.О. Кошель. – Режим доступу: http://www.nbuv.gov.ua/portal/Chem_Biol/Vldau/APK/2010_1/files/ 10kalimc.pdf.

5. Липінська Є. І. Зарубіжний досвід захисту інформації в сфері підприємництва та його використання в Україні / Є. І. Липінська // Порівняльно-аналітичне право. – 2017. – № 5. – С. 148–150.

6. Лук'янова В. В. Економічний ризик: навч. посіб. / В. В. Лук'янова, Т. В. Головач. – К.: Академвидав, 2007. – 464 с.

7. Ляшенко О. М. Кадрова безпека у системі економічної безпеки підприємства / О. М.Ляшенко, Я. М. Криль // Економіка. Менеджмент. Підприємництво : зб. наук. праць. – 2013. – № 25 (2). – 220 с.

8. Маковецький О. М. Підходи до удосконалення методики оцінки ефективності комплексної системи захисту інформації / О. М. Маковецький, І. Р. Мальцева, Н. А. Паламарчук, Ю. О. Черниш, О. В. Шемендюк // Сучасні інформаційні технології у сфері безпеки та оборони. – 2016. – № 2 (26). – С. 54–58.

9. Менеджмент інформаційної безпеки: підруч.: у 2 ч. / А. К. Гринь, О. Д. Довгань, В. І. Журавель та ін.; за заг. ред. Є. Д. Скулиша. – К. : Наук.–вид. Центр НА СБ України, 2013. – Ч.1. – 456 с.; Ч.2. – 604 с.

10. Мороз Е. С. Методы противодействия сетевым атакам / Е. С. Мороз, В. О. Хорошко, Е. Е. Смычков // Збірник наукових праць. – Севастополь, СНУЯЕтаІІ, 2007. – Т. 18 (№ 5). – С. 180-187.

11. Невойт Я. В. Метод оцінювання стану захищеності інформаційних ресурсів на основі дослідження джерел загроз інформаційній безпеці : дис. канд. техн. наук. спец. 21.05.01/ Я. В. Невойт ; К. – ДУТ, 2016. – 110 с.

12. Нестеренко М.М., Романов А.О. Аналіз методів захисту серверів від розподілених TCP SYN-FLOOD атак / М.М. Нестеренко, А.О. Романов // 2016. – [Електронний ресурс]. – Режим доступу: conferenc.its.kpi.ua/proc/article/download/73108/68432

13. Андон П. І. Атаки на відмову в мережі Інтернет : опис проблеми та підходів до її вирішення / П. І. Андон, О. П. Ігнатенко. – К. : Ін-т ПС, 2008. – 52 с.

14. Андон П. І., Ігнатенко О.П. Протидія атакам на відмову в мережі інтернет: концепція підходу. *Проблеми програмування.* 2008. № 2-3. С. 564-574.

15. Антонюк П. Є. Класифікація ймовірних способів вчинення атак на інформацію як напрям протидії комп'ютерній злочинності. URL: http://www.nbuv.gov.ua/portal/Soc_Gum/bozk/19text/g1927.htm. (дата звернення 22.04.2022)

16. Коваленко О. В. Моделі та методи розроблення безпечного програмного забезпечення комп'ютерних систем [Електронний ресурс]. – Режим доступу : https://er.chdtu.edu.ua/handle/ChSTU/1595 (дата звернення 22.04.2022)

17. Багнюк Н. В. Види DDoS-атак та алгоритм виявлення DDoS-атак типу Flood-атак. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*. 2015. № 18. С. 6-12.

18. Гарасимчук О. І. Оцінка ефективності систем захисту інформації / О. І. Гарасимчук, Ю. М. Костів // Вісник КНУ імені Михайла Остроградського. – 2016. – № 1. – С. 16–20.

19. Гнатюк С. Є. Математичні моделі оцінки та прогнозування надійності програмно-керованих засобів захисту інформації в системі урядового зв'язку / С. Є.

Гнатюк // Ukrainian Information Security Research Journal. – 2016. – № 2. – С. 150–156.

20. Грищук Р. В. Атаки на інформацію в інформаційно-комунікаційних системах / Р. В. Грищук // Сучасна спеціальна техніка. – 2011. – № 1(24).– С. 61-66.

21. Державна служба спеціального зв'язку та захисту інформації України. Офіційний сайт. [Електронний ресурс]. – Режим доступу : https://cip.gov.ua/ (дата звернення 26.4.2022).

22. Діордіца І. Система забезпечення кібербезпеки: сутність та призначення / І. Діордіца // Інформаційне право. – 2017. – № 7. – С. 109–116.

23. Єрмошин В. В. Методика оцінки інформаційних ризиків системи управління інформаційною безпекою / В. В. Єрмошин, В. О. Хорошко, М. В. Капустян // Сучасний захист інформації. – 2010. – №3. – С. 95–104.

24. Перевалова Л. В. Захист конфіденційної інформації: проблеми та шляхи вирішення / Л. В. Перевалова, С. В. Кваша / Вісник Національного тех.-нічного університету «Харківський політехнічний інститут». Збірник науко-вих. праць. Тематичний випуск: Актуальні проблеми розвитку українського суспільства. – Харків : НТУ «ХПІ», 2011. – № 30. – 179 с.

25. Проблеми впровадження сучасних стандартів інформаційної безпеки в умовах становлення національної системи кібербезпеки України [Електронний ресурс]. – Режим доступу : http://www.niss.gov.ua /content/articles/files/1_cPPP-standarts_27-04_Gn_var_FIN-732b6.pdf

26. Сенейко Ю. В. Сучасні підходи до трактування категорії «ризик». Регіональна економіка. 2006. № 1. С . 206-211.

27. Ткачук Т. Ю. Забезпечення інформаційної безпеки: досвід окремих країн східної Європи / Т. Ю. Ткачук // Інформація і право. – 2017. – № 4 (23). – С. 62–72.

28. Ткачук Т. Ю. Сучасні загрози інформаційній безпеці держави: теоретико-правовий аналіз / Т. Ю. Ткачук // Інформаційне право. – 2017. – № 10. – С. 182–186.

29. Чунарьова А. Система управління інформаційною безпекою на базі міжнародних стандартів серії ISO / А. Чунарьова, А. Чунарьов // Правове,

нормативне та метрологічне забезпечення системи захисту інформації в Україні, – 2012. – № 2. – С. 48–52.

30. Шатун В. Т. Інформаційна безпека – невід'ємна складова національної безпеки України / В. Т. Шатун, О. В. Гладун // Наукові праці. Державне управління. – 2016. – Вип. 255, Т. 267. – С. 174–180.

31. Шпінталь М. Я. Методи захисту робочих станцій від DDoS-атак / М. Я. Шпінталь, Н. М. Орловський // ACIT'2014. – Тернопіль, 2014. – С. 230–231.

32. Factor Analysis of Information Risk (FAIR) [Електронний ресурс]. – Режим доступу : http://www.riskmanagementinsight.com/.

33. Feinstein L., Schnackenberg D., Balupari R., Kindred D. Statistical Approaches to DDoS Attack Detection and Response. // DARPA Information Survivability Conference and Exposition

34. ISACA RAM [Електронний ресурс]. – Режим доступу : https://www.isaca.org/Pages/default.aspx?cid=1002083&Appeal=SEM&gclid=CMD70-aXs8oCFcLVcgoduzIAmg/(дата звернення 26.4.2022).

35. NIST SpecialPublication 800-30 Risk Management Guide for Information Technology Systems [Електронний ресурс]. – Режим доступу : http://www.nist.gov (дата звернення 26.4.2022).

ДОДАТОК А

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИПЛОМНОЇ РОБОТИ МАГІСТРА

1. Analysis of the main methods of countering injection attacks / O. Fedorchuk, T. Babenko / Information Technology and Implementation (Satellite) – 2021 – 3 с.
2.