Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Програмний засіб захисту web-сервера від DoS-атак»

Виконав: студент 2 курсу, групи КІ-21мз
напряму підготовки (спеціальності)
123 — «Комп'ютерна інженерія»
_____ Савчук О. М.

Керівник: к.т.н., професор каф. ОТ
_____ Азарова А. О.
«10» 06 2023 р.

Опонент: д.т.н., професор каф. МБІС
_____ Яремчук Ю. Є.
«12» 06 2023 р.

Допущено до захисту
Завідувач кафедри ОТ
д.т.н., проф. Азаров О. Д.
«14» 06 2023 р.

Вінниця ВНТУ — 2023

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Галузь знань — Інформаційні технології
Освітньо-кваліфікаційний рівень — магістр
Спеціальність — 123 «Комп'ютерна інженерія»
Освітньо-професійна програма — Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри
обчислювальної техніки
проф., д.т.н. О.Д. Азаров

«20» 03 2023 р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Савчуку Олексію Миколайовичу

1 Тема роботи «Програмний засіб захисту web-сервера від DoS-атак» Керівник роботи Азарова Анжеліка Олексіївна, к.т.н., професор, затверджені наказом вищого навчального закладу від 20.03.2023 р. № 68

2 Строк подання студентом роботи 12.06.2023 р.

3 Вихідні дані до роботи — дані про існуючі різновиди захисту web-сервера від DoS-атак.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, аналіз об'єкта захисту та обґрунтування вимог до системи захисту, аналіз DDoS-атак та розробки методу захисту, проектування структури системи захисту від DDoS-атак, висновки, перелік джерел посилання.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): архітектура DDoS мережі, схема протидії DDoS атакам, діаграма конфігурації web-сервера Against DDoS-демоном, виявлення атак на веб-сервер, діаграма потоку даних.

6 Консультанти розділів роботи представлені в таблиці 1.

Таблиця 1 — Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1,2,3	Захарченко С.М., к.т.н., професор		
4	Нікіфорова Л.О., к.е.н., доцент		

7 Дата видачі завдання 23.03.2023 р.

8 Календарний план наведено в таблиці 2.

Таблиця 2 – Календарний план

№	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	Примітк
1	Постановка задачі роботи	25.03.2023	вик.
2	Аналіз об'єкта захисту та обгрунтування вимог до системи захисту	27.03-09.04	вик.
3	Класифікація та огляд різновидів DoS-атак	10.04-18.04	вик.
4	Порівняння існуючих методів захисту мережевих ресурсів	19.04-27.04	вик.
5	Проектування структури системи захисту від DDos-атак	28.04-12.05	вик.
6	Підготовка матеріалів та опис розробки	13.05-17.05	вик.
7	Оформлення пояснювальної записки та ілюстративного матеріалу	18.05-31.05	вик.
8	Аналіз виконання роботи, висновки, додатки	01.06-05.06	вик.
9	Перевірка якості виконання магістерської роботи та усунення недоліків	06.06.2023	вик.

Студент Савчук О.М.

Керівник роботи Азарова А.О.

АНОТАЦІЯ

УДК 004.056

Савчук О.М. Програмний засіб захисту web-сервера від DoS-атак. Магістерська кваліфікаційна робота зі спеціальності 123 — комп'ютерна інженерія, освітня програма комп'ютерна інженерія. Вінниця: ВНТУ, 2023, 98 с.

На укр.мові. Бібліогр.: 16 назв, рис. 9, табл. 7.

Дана магістерська кваліфікаційна робота присвячена дослідженню та реалізації програмного засобу захисту web-сервера від DoS-атак.

При виконанні роботи проведений аналіз об'єкта захисту та обґрунтування вимог до системи захисту, виконаний аналіз DDoS-атак та розглянуто існуючі методи захисту від них.

В результаті роботи розроблений програмний засіб захисту web-сервера від DoS-атак з використанням web-серверів Nginx, який виконує роль frontend серверу та Apache, який виконує роль backend серверу.

Ключові слова: DoS, DDoS-атаки, комп'ютерні мережі, баєсівський класифікатор, захист комп'ютерних мереж.

ANNOTATION

Savchuk O.M. Software for web server protection against DoS attacks. Master's qualification thesis on specialty 123 — computer engineering, educational program computer engineering. Vinnytsia: VNTU, 2023, 98 p.

In Ukrainian. Bibliogr .: 16 titles, fig. 9, table. 7.

This master's thesis is devoted to the research and implementation of a software tool for web server protection against DoS attacks.

During the performance of the work, an analysis of the object of protection and substantiation of the requirements for the protection system was carried out, an analysis of DDoS attacks was performed and existing methods of protection against them were considered.

As a result of the work, a software tool for web server protection against DoS attacks was developed using Nginx web servers, which acts as a frontend server and Apache, which acts as a backend server.

Keywords: DoS, DDoS-attacks, computer networks, Bayesian classifier, protection of computer networks.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

АС — Автоматизована система

БД — База даних

ІБ — Інформаційна безпека

ІР — Інформаційні ресурси

ІС — Інформаційна система

КЗЗ — Комплекс засобів захисту

КСЗІ — Комплексна система захисту інформації

ОС — Операційна система

ПЗ — Програмне забезпечення

СЗІ — Система захисту інформації

СКБД — Система керування базами даних

DDoS-атака — Distributed Denial of Service attack, розподілена атака на відмову в обслуговуванні

DNS — Domain Name System

DoS — Denial of Service, відмова в обслуговуванні

HTTP — HyperText Transfer Protocol

HTTPS — HyperText Transfer Protocol Secure

ВСТУП

Актуальністю теми захисту web-серверів від атак є бурхливий розвиток комунікаційних і обчислювальних технологій, широке використання мережі Інтернет породжує значні потоки інформації, що передається комп'ютерними мережами. Через всесвітню мережу Internet здійснюється оплата послуг, оперативний електронний зв`язок, доступ до публічних інформаційних ресурсів, тощо. Саме тому проблема захисту від кіберзлочинців з кожним роком набуває все більшої актуальності. Враховуючи той факт, що доступ до більшості інформаційних ресурсів здійснюється через протоколи HTTP та HTTPS, тема магістерської кваліфікаційної роботи, присвячена захисту web-сервісу від DoS та DDoS атак є вкрай актуальною. [1]

Метою та задачею дослідження є підвищення захищеності інформаційних ресурсів web-серверів від атак відмови в обслуговуванні за рахунок багатопараметричної класифікації запитів, що надходять на сервер. Для досягнення поставленої мети слід розв'язати такі задачі:

— проаналізувати відомі атаки відмови в обслуговуванні;

— проаналізувати існуючі методи захисту від атак відмови в обслуговуванні;

— визначити перелік атрибутів даних, за якими буде здійснюватись класифікація;

— запропонувати спосіб класифікації трафіку з використанням критерію Баєса;

— запропонувати систему захисту від атак відмови в обслуговуванні;

Об`єктом досліджень є процес здійснення DoS атаки.

Предметом досліджень є методи виявлення атак у відмові в обслуговуванні.

Наукова новизна одержаних результатів полягає у вдосконалені методу ідентифікації атак відмови в обслуговуванні за рахунок багатопараметричної класифікації вхідних запитів, що дозволило підвищити точність розпізнавання хибних запитів на сервер.

Практична цінність полягає в тому, що:

— розроблено структуру системи захисту від атак відмови в обслуговуванні, що містить додатковий web-сервер, який виконує функцію посередника;

— розроблено програмне забезпечення для реалізації запропонованого методу захисту.

Апробація результатів магістерської роботи опублікована на LII Науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії (2023).

Матеріали роботи доповідались та опубліковувались [2]:

Савчук О.М. використання баєсівського класифікатора для ідентифікації DDos-атак в комп’ютерних мережах. / О.М. Савчук, С.М. Захарченко. // Тези доповіді. LII Науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії (2023): веб-сайт. URL: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2023/paper/view/17440 (дата звернення: 02.06.2023)

1 АНАЛІЗ О’БЄКТА ЗАХИСТУ ТА ОБҐРУНТУВАННЯ ВИМОГ ДО СИСТЕМИ ЗАХИСТУ

1.1 Структура корпоративного порталу

У сучасному світі ресурси Інтернету використовуються в різних сферах людської діяльності, але особливе значення Інтернет представляє для сучасного бізнесу, перед яким відкриваються нові можливості:

— надання інформації про товари та послуги клієнтам партнерам 24 години на добу;

— інформування клієнтів про нові товари, знижки та новини;

— підвищення якості обслуговування клієнтів;

— продаж товару безпосередньо з сайту;

— зміцнення і та розвиток корпоративного бренду;

— пошук нових партнерів і вербування нових співробітників;

— тестування та проведення маркетингових досліджень.

Також Інтернет — це не тільки інструмент для вирішення бізнес-задач та ефективний канал просування ваших товарів та послуг, це нове поле діяльності, де ви можете створити для свого бренду історію. Але його віддача буде відчутною тільки при серйозному підході до використання його можливостей:

— розміщення банерної та контекстної реклами на порталі;

— розміщення розширеної інформації про компанію;

— розміщення статей та новин на правах реклами;

— реєстрація компанії в ряді тематичних каталогів;

— просування продукції компанії в мережі Інтернет;

— розміщення рекламних акцій для користувачів порталу.

Корпоративний портал — це продукт або набір продуктів, який базується на певній інфраструктурі (до неї зазвичай відносяться сервери баз даних та сервери додатків).[3] У складі даного корпоративного порталу умовно можна виділити три основні функціональні складові:

а) складова базової інфраструктури, яка відповідає за основні сервіси, зокрема система безпеки, управління транзакціями та управління продуктами. Вона складається зі сервера баз даних, сервера додатків і web-сервер;

б) складова інтегрування додатків, яка відповідає за обмін даними між порталом та всіма застосуваннями, які існують в компанії, такими як СКБД, CRM та ERP-системи;

в) складова інтерфейсів, яка включає інтерфейси для передачі даних між ІС бізнес-партнерів, методи керування інформаційним наповненням та засоби для роботи з бездротовими та мобільними пристроями. Також до цієї складової відносяться візуальні та не візуальні компоненти порталів.

Архітектура порталу відкрита, що дає можливість розширювати його функціональність за допомогою сторонніх компонентів. Зазвичай ці компоненти включають засоби управління інформаційним наповненням, які зазвичай надаються виробниками порталів або входять до їх складу.

Функціональна архітектура порталу, зображена на рисунку 1.1, включає доступ до різноманітних інформаційних джерел, які індексуються та зберігаються у власній базі метаданих. Для користувачів доступні два режими роботи, один з яких дозволяє налаштувати портал відповідно до їхніх потреб.

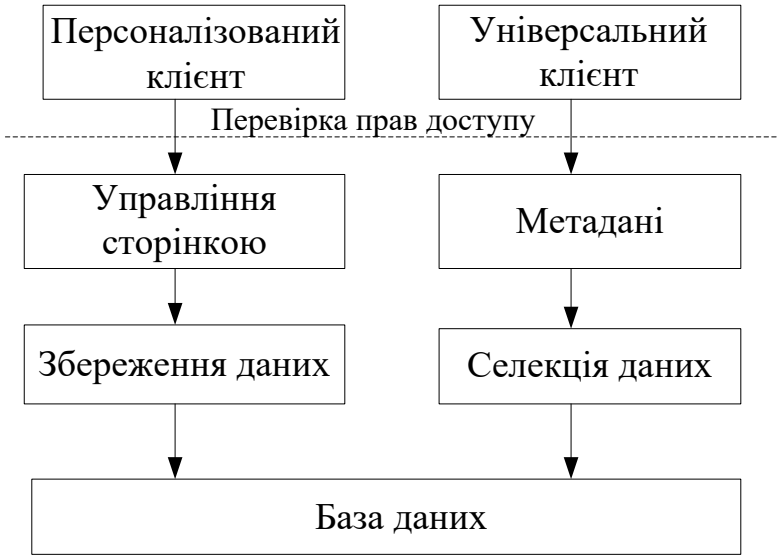

Рисунок 1.1 — Логічна структура порталу

Функціональна архітектура підтримується логічною, яка може включає наступний набір компонентів. [4]

Клієнт. Він є звичайним навігатором, який надає користувачу доступ до різних сторінок у форматі HTML. Головна особливість полягає у тому, що домашня сторінка може бути налаштована під конкретних користувачів, забезпечуючи доступ до звітів, документів та інших даних, які є цікавими та доступними для них. Крім зазначеної систематики, сторінка може містити пошуковий механізм за ключовими словами та інші типові навігаційні інструменти.

Забезпечення безпеки доступу через Інтернет. Даний компонент є важливою складовою частиною порталу. Цей аспект включає застосування стандартних захисних екранів, які забезпечують засоби кодування, аутентифікації і керування сесіями. Ці засоби дозволяють забезпечити конфіденційність і цілісність даних, а також забезпечити контроль доступу до різних ресурсів порталу через ідентифікацію та перевірку прав доступу користувачів.

Репозиторій. Він є однією з ключових складових корпоративного порталу, де зберігаються метадані про різноманітні інформаційні об'єкти, робочі групи, користувачів та інформаційні канали. Метадані, пов'язані з кожним об'єктом, включають його тип, приналежність до певного розділу, формат зберігання та місцезнаходження.

1.2 Структура публічного web-порталу

В Україні та по всьому світу, спостерігається зростання практичного використання комерційними та державними організаціями публічних web-порталів, які підключені до Інтернету. Такі портали можуть використовуватися для різноманітних завдань, включаючи рекламу діяльності компанії в мережі Інтернет. Цьому сприяє наявність готових промислових рішень на вітчизняному ринку інформаційних технологій, що дозволяють побудувати повноцінні web-портали. До таких рішень належать продукти, такі як «Internet Information Services» від компанії Microsoft, «WebSphere» від компанії IBM та «Sun ONE Portal» від компанії Sun Microsystems.

Базова архітектура web-порталу, зазвичай складається з компонентів, описаних далі.

Публічні web-сервери надають користувачам мережі Інтернет можливість отримувати доступ до ресурсів порталу.

Кеш-сервери використовуються для тимчасового зберігання копій ресурсів, до яких користувачі вже мали доступ. При спробі отримати ресурс з web-порталу, перевіряється, чи наявний цей ресурс у кеш-серверах. Якщо ресурс знаходиться в кеші, він відразу надсилається користувачеві, інакше запит передається публічним web-серверам для отримання необхідних даних.

На серверах додатків установлено програмне забезпечення, яке керує інформаційним змістом web-порталу. Вони виконують функції додавання, редагування та видалення контенту, створення структури та категоризації матеріалів порталу.

Сервери баз даних, що дозволяють зберігати інформаційні ресурси (ІР) web-порталу централізовано.

Комунікаційне устаткування, яке забезпечує стабільну взаємодію та роботу між різними серверами web-порталу.

Зазвичай, сервери web-порталів розташовуються в Інтернет-провайдерів, які забезпечують достатню пропускну здатність для підключення серверів порталу до мережі Інтернету.[5]

Управління web-порталом в цьому випадку здійснюється віддалено за допомогою мережі Інтернет з АРМ (автоматизованих робочих місць) адміністраторів.

Зважаючи, що публічні ресурси web-порталу доступні будь-якому користувачеві Інтернету, вони можуть стати мішенню для атак з боку зловмисників. В останні роки спостерігається значний приріст інформаційних атак, більшість з яких спрямовані на загальнодоступні ресурси, включаючи web-портали. Ці атаки можуть спрямовуватись на порушення доступності, цілісності або конфіденційності ІР, які зберігаються на серверах web-порталу.

1.3 Комплексна система захисту web-порталу

Для забезпечення безпеки web-порталу рекомендується використовувати комплексний підхід, який поєднує технічні й організаційні заходи захисту. Технічні заходи забезпечуються за допомогою апаратних, програмних або програмно-апаратних засобів, які виконують визначені завдання і цілі зазначених нормативно-правових документів.[6] Організаційні заходи включають розробку і впровадження нормативно-правових документів, зокрема інструкції для персоналу щодо роботи з автоматизованою системою порталу, політики і концепції забезпечення інформаційної безпеки web-порталу, та інші відповідні документи. Комплексний підхід передбачає об'єднання технічних та організаційних засобів web-порталу в інтегрований комплекс, який включає підсистеми контролю цілісності, антивірусного захисту, розмежування доступу, аналізу захищеності, підсистему управління, виявлення вторгнень, а також криптографічного захисту інформації. Далі описані можливості даних підсистем, їх класифікацію та особливості застосування для забезпечення захисту web-порталу.

Інформація web-сторінки поділяється на дві категорії:

— загальнодоступна інформація;

— технологічна інформація.

До загальнодоступної інформації відноситься публічно оголошувана інформація, користуватися якою можуть будь-які фізичні або юридичні особи (користувачі інформаційних ресурсів), що мають доступ до мережі Інтернет.

До технологічної інформації web-сторінки відноситься технологічна інформація комплексної системи захисту інформації (КСЗІ) та технологічна інформація на рахунок способів управління та адміністрування обчислювальною системою АС і засобами обробки інформації, до яких відносяться дані про мережеві адреси, персональні ідентифікатори, імена, паролі користувачів, інформація журналів реєстрації дій користувачів, їхні права доступу до об’єктів та повноваження, інша інфор-

мація баз даних захисту, встановлені робочі параметри окремих механізмів або засобів захисту, інформація про профілі обладнання та режими його функціонування, робочі параметри функціонального програмного забезпечення тощо.

Технологічна інформація призначена для використання тільки уповноваженими користувачами з числа співробітників та персоналу, що забезпечує функціонування АС.

Способи і методи обробки інформації web-сторінки (зберігання, супроводження, передачі, введення, актуалізації та використання інформації) визначають технології оброблення інформації.

Технологічні особливості роботи користувачів із загальнодоступною інформацією web-сторінки визначаються особливостями системного та функціонального програмного забезпечення, зокрема браузерів, які ними використовуються.

Технологічні особливості роботи користувачів інших категорій визначаються, крім того, архітектурою АС, способами оброблення та передавання інформації між компонентами АС і способами здійснення доступу до неї.

Можливі наступні способи здійснення доступу до технологічної інформації та передавання даних для актуалізації загальнодоступної інформації:

— з робочої станції, розміщеної на тій самій території, що і web-сервер (установи-власника web-сторінки або оператора) або з терміналу web-сервера;

— з робочої станції, яка розміщена на території установи-власника web-сторінки, до web-сервера, що розміщений на території оператора, з використанням мереж передачі даних.

КСЗІ повинна гарантувати захист цілісності та доступності загальнодоступної інформації, розміщеної на web-сторінці та повинна забезпечувати конфіденційність і цілісність технологічної інформації, що стосується web-сторінки.

Технологія опрацювання інформації повинна відповідати політиці безпеки автоматичних систем, що забезпечує функціонування web-сторінки.

Для виконання вимог по забезпеченню цілісності інформації web-сторінки, а також цілісності та конфіденційності технологічної інформації, необхідно використовувати технології, які забезпечують контрольований та санкціонований доступ до інформації та запобігають несанкціонованій чи неконтрольованій модифікації.

Технологія для оброблення інформації має виявляти спроби несанкціонованого доступу до інформаційних ресурсів web-сторінки та пов'язаних з нею процесів і забезпечувати реєстрацію в системному журналі відповідних подій згідно з політикою безпеки.

Засоби комплексної системи захисту інформації (КСЗІ) повинні блокувати доступ до web-сторінки для користувачів, які порушили відповідні правила розмежування доступу протягом сеансу роботи. [7]

Необхідно реалізувати технологічні процеси, які дозволять створювати резервні копії інформації web-сторінки, а також забезпечити процедури їх відновлення з використанням цих резервних копій.

Необхідно, щоб технологія для оброблення інформації мала здатність аналізувати використання користувачами та процесами обчислювальних ресурсів автоматизованої системи і забезпечувати керування ресурсами. Узагальнена функціонально-логічна структура обчислювальної системи АС включає:

— підсистему обробки інформації;

— підсистему обміну даними;

— підсистему взаємодії з користувачами АС.

Підсистема обробки інформації відповідає за створення, актуалізацію та зберігання інформації на web-сторінці. Вона складається з різних засобів обробки інформації, а також функціонального та системного ПЗ.

До засобів обробки інформації належать web-сервер та необхідна кількість робочих станцій для забезпечення всіх функцій щодо супроводження web-сторінки та захищення інформації.

Підсистема обміну даними відповідає за підготовку та прямий обмін інформацією між автоматизованими системами шляхом імпорту та експорту даних, а також відповідає за внутрішній обмін інформацією між web-сервером і робочими станціями. Це включає процеси встановлення, підтримки і завершення з'єднання.

Підсистема взаємодії з користувачами АС забезпечує надання доступу до загальнодоступної інформації web-сторінки, що представлена у вигляді HTML-документу, відповідаючи на запити користувачів. Це здійснюється з використанням мереж передачі даних і стандартних Інтернет-протоколів.

Підсистема складається з програмно-апаратного комплексу, який дає можливість проводити маршрутизацію запитів користувачів, проводити пошук необхідних інформаційних ресурсів та доступ до цих ресурсів.

Згідно з політиками безпеки, для інформації в автоматизованих системах виконуються заходи захисту інформації (зокрема використовуються стандартні засоби захисту функціонального й системного програмного забезпечення або спеціалізовані засоби), що складають компоненти комплекс засобів захисту. [8]

Встановлення на операційну систему нових (додаткових) компонентів, сервісів, програмного забезпечення (функціонального, системного або їх поєднання) та розміщення інших мережевих ресурсів, що не відносяться до web-сторінки установи, не мають порушувати політику безпеки інформації в автоматизованих системах, що забезпечує функціонування web-сторінки.

До робочих станцій фізичних чи юридичних осіб, які являються користувачами web-сторінки зі загальнодоступною інформацією, а також до їх програмного забезпечення особливі вимоги не висуваються.

За рівнем прав користувачів на рахунок доступу до інформації, складністю та характером робіт, що виконуються під час функціонування АС, користувачі поділяються на такі категорії:

а) користувачі, яким надано повноваження супроводжувати комплексна система захисту інформації та забезпечувати керування АС (адміністратор безпеки, користувачі з обов’язками web-майстрів, співробітники систем захисту інформації, адміністратори мережевого обладнання, адміністратори сервісів, адміністратори

ресурсів DNS, FTP, PROXY, якщо є потреба взаємодії цих ресурсів з web-сторінкою);

б) користувачі, яким доступні повноваження для входу лише загальнодоступної інформації web-сторінки;

в) розробники ПЗ, які розробляють та впроваджують нові функціональних процеси, розробники фізичної структури АС, а також обслуговування вже діючого функціонального програмного забезпечення сервера;

г) технічний обслуговуючий персонал, який відповідає за належні умови функціонування АС, штатну підтримку життєдіяльності фізичного середовища (спеціалісти з ліній зв'язку, електрики, працівники для обслуговування приміщень в будівлі).

Користувачі, яким надано повноваження супроводжувати КСЗІ та забезпечувати управління АС, повинні володіти навичками обслуговування засобів захисту інформації та використання технічних і програмних засобів, що застосовуються ними під час виконання своїх службових і функціональних обов'язків.

Користувачі, що відносяться до категорії «г», повинні мати відповідний рівень кваліфікації для виконання функціональних та службових обов'язків відповідно до установлених режимів експлуатації обладнання та технологічних процесів.

Доступ до інформації web-ресурсу має надаватися користувачам у відповідності до положень політики безпеки інформації, визначеної для АС, що забезпечує функціонування web-сторінки.

Порядок доступу до програмного забезпечення та компонентів АС користувачів різних категорій розробляється СЗІ й затверджується керівником установи.

Обов'язковою є реєстрація в АС користувачів, що належать до категорії «а», чим забезпечується можливість однозначної їх ідентифікації, а також їхніх дій щодо інформації web-сторінки.

Для регламентації доступу користувачів категорії «а» до інформації web-сторінки та встановлення для них правил, системою захисту інформації розробляються та впроваджуються розпорядчі та нормативні документи, які передбачені за планом захисту інформації. [9]

Користувачі з перелічених категорій повинні мати відповідний дозвіл на доступ до даних, які містяться в технічній і програмній документації на автоматизовану систему або її компоненти.

Вимоги до користувачів, яким надається право доступу до загальнодоступної інформації web-сторінки, не висуваються.

Користувачі загальнодоступної інформації отримують доступ до web-сторінки у відповідності до діючих у мережі Інтернет правил та регламенту.

Це дозволяє управляти потоками інформації від пристроїв користувачів до захищених ресурсів web-сторінки.

Політика мінімальної адміністративної цілісності стосується: загальнодоступної інформації web-сторінки; користувачів усіх категорій; файлової системи та функціонального програмного забезпечення, яке використовується для захисту, оновлення загальнодоступної інформації та супроводження web-сторінки; створеної в процесі супроводження web-сторінки технологічної інформації комплексна система захисту інформації та технологічної інформації щодо управління АС.[9]

Комплекс засобів захисту має реалізовувати розмежування доступу на підставі прав доступу користувачів до захищених об’єктів. Розмежування доступу реалізовується на рівні надавання користувачу прав редагувати об’єкт.

Право визначати множину об'єктів автоматизованих систем, цілісність яких забезпечується комплексом засобів захисту, надається адміністратору безпеки.

Комплекс засобів захисту має надавати права адміністратору безпеки для кожного захищеного об’єкта ідентифікувати домен, до якого мають належати ті користувачі чи групи користувачів, які мають право редагувати об’єкт. Тільки адміністратору надається право додавати і вилучати користувачів та об'єкти у конкретних доменах.

Призначення ідентифікаторів доступу процесам та користувачам до захищених об'єктів, а також запити на зміну цих прав мають оброблятися КЗЗ тільки за умови, що вони поступають від адміністратора безпеки. Користувачам забороняється редагувати будь-які захищені об’єкти, якщо вони мають доступ лише до загальнодоступної інформації web-ресурсу.

Адміністратор безпеки отримує право редагувати функціональне програмне забезпечення, що використовується для захисту загальнодоступної інформації, та технологічну інформацію комплексної системи захисту інформації. Користувачам, які мають право управляти автоматизованою системою, дозволяється відповідно до обов’язків, надається можливість редагувати технологічну інформацію та функціональне програмне забезпечення, що використовується для актуалізації загальнодоступної інформації та супроводження web-сторінки.

Права доступу до захищених об’єктів web-сторінки повинні встановлюватися в момент їх створення або ініціалізації. [10]

Ця послуга дозволяє забезпечити захист web-сторінки від несанкціонованої модифікації інформації, яка передається між web-сервером та терміналами при використання технології Т2, під час передачі інформації через середовище без належного захисту. Політика послуги відноситься до всіх об’єктів, які передаються.

КЗЗ повинен забезпечувати контроль за цілісністю даних в повідомленнях, що передаються та повинен мати можливість виявляти факти незаконне видалення або дублювання даних.

КЗЗ повинен реалізовувати реєстрацію подій, через які відбулось порушення цілісності повідомлень чи їх незаконне видалення або дублювання.

Дана послуга дозволяє відмінити окрему операцію чи послідовність операцій, завдяки чому з’являється можливість повернути захищений об'єкт до попереднью визначеного стану.

Політика обмеженого відновлення відноситься до: [11]

— користувачів, яким дається право супроводження комплексної системи захисту інформації та управління АС;

— об’єктів, що зберігають публічну інформацію;

— функціонального ПЗ, що використовується для захисту та оновлення публічної інформації та підтримки web-сторінки;

— створеної в процесі підтримки web-сторінки технологічної інформації комплексної системи захисту інформації та технологічної інформації поо управлінню автоматизованих систем.

Якщо відносно якогось з об’єктів, вище перерахованих категорій, в процесі обробки не передбачається його редагування, політика послуги по відношенню до нього не поширюється.

До складу автоматизованих систем мають входити автоматизовані засоби, що дають змогу співробітнику систем захисту інформації, адміністратору безпеки або користувачу, який має відповідні права управління АС, відновити або скасувати певний набір операцій, які були проведені над захищеним об’єктом web-ресурсу за певний час.

Кожне використання послуги має бути зареєстроване в системному журналі. Скасування операції не повинне видаляти запис про редагування зі журналу, якщо ця операції підлягала реєстрації відносно до політки безпеки. Ця послуга дозволяє управляти використанням послуг та ресурсів.

Політика використання ресурсів, яка створюється КЗЗ, стосується: адміністратора безпеки; користувачів загальнодоступної інформації; користувачів, яким надано права управління автоматизованою системою; системного та функціонального ПЗ; файлової системи; технологічної інформації по управлянню автоматизованою системою; обчислювальних ресурсів АС та окремих периферійних пристроїв. Політика використання ресурсів передбачає можливість встановлення обмежень на їх права доступу та використання.

Обмеження, які відносяться до використання окремим користувачем чи процесом певного обсягу обчислювальних ресурсів автоматизованих систем, встановлюються адміністратором безпеки чи користувачами, у яких є права управління АС. Запити на редагування встановлених обмежень мають оброблятися КЗЗ лише у випадку, якщо вони надходять від вище названих користувачів.

Спроби користувачів перевищити обмеження на використання ресурсів, мають бути записані в системному журналі.

Політика відновлення після збоїв, що реалізується комплексом засобів захисту, стосується:

— функціонального та системного ПЗ;

— засобів управління КСЗІ та засобів захисту інформації;

— засобів керування та адміністрування обчислювальною системою АС.

Вони забезпечують управління та контроль за станом системи, а також забезпечують її повернення у відомий захищений стан після виникнення скасувань або переривання обслуговування. Це включає відновлення системи після помилкових дій користувачів, функціональних недоліків апаратного та програмного забезпечення, які не були передбачені під час проектування, а також управління іншими непередбачуваними ситуаціями

Політика відновлення, яка створюється комплексом засобів захисту, повинна визначати множину типів відмов web-сторінки і призупинки обслуговування, після чого можливе відновлення об’єктів у відомий захищений стан відповідно до політики безпеки. Для кожної окремої відмови мають бути чітко зазначені рівні відмов, за перевищення яких потрібна інсталяція web-сторінки.

Після відмови web-сторінки або переривання обслуговування, комплекс засобів захисту повинен перевести web-сторінку до виду, з якого вивести сторінку в режим штатного функціонування може лише адміністратор безпеки або користувачі, яким надані права управління АС. Для кожного користувача з відповідними правами повинен бути визначений перелік допустимих операцій для повернення автоматизованої системи у захищений стан.

Для повернення АС з режиму, через який погіршуються характеристики обслуговування, в нормальний режим функціонування, мають здійснюватися тільки ручні процедури.

Ця послуга дає можливість розмежувати права користувачів, визначивши категорії користувачів відповідно до певної, підходящої для кожної з категорій ролі.

Вона призначена для зниження ймовірних збитків від помилкових чи навмисних дій користувачів та обмеження управління АС.

Політика розподілу обов’язків, яку створює КЗЗ, відноситься до користувачів усіх категорій, а також повинна визначати такі обов’язкові ролі:[12]

— користувачів, які мають право доступу до певних видів інформації (технологічної, публічної, функціонального та системного ПЗ);

— адміністратора безпеки.

Кількість користувачів, що мають право доступу до технологічної інформації та функціонального й системного ПЗ повинна бути мінімізована та надана тільки тим користувача, яким необхідні такі права доступу для виконання обов'язків, які передбачаються розпорядчою та експлуатаційною документацією на web-ресурс.

Адміністратору безпеки надаються права доступу до всієї інформації web-сторінки. При необхідності роль адміністратора може дублюватися певним працівником служби захисту інформації. Право доступу всіх інших користувачів до доступу інформації надаються безпосередньо адміністратором безпеки. [13]

КЗЗ повинен присвоїти користувачу ідентифікатори, якими однозначно характеризується його роль. Користувач може виконувати певну роботу тільки після автентифікації, яка підтверджує його роль.

2 АНАЛІЗ DDOS-АТАК ТА РОЗРОБКА МЕТОДУ ЗАХИСТУ

2.1 Огляд DoS-атак

DoS-атака — це атака на обчислювальну систему задля виведення її з ладу, для створення умов, при яких легітимні користувачі системи не мають змоги отримати доступ до ресурсів, які надає система, або ж доступ ускладнений. [14]

Можлива реалізація атаки на відмову в обслуговуванні трьома способами:

— надсилати такий об’єм мережевого трафіку, який суттєво перевищує пропускну здатність системи;

— використати вразливості в ПЗ;

— завантажити до максимуму продуктивності критично системні ресурси атакованої системи, зокрема процесорний час та пам'ять.

Подібні атаки неможливо реалізувати, тому що один пристрій не може перевантажити канал сервера, через це використовується розподілена атака.

DDoS-атака являє собою ієрархічну структуру — трирівневу модель, яка складається з: [15]

— консолі управління, головного комп'ютера, яка подає команду розпочати атаку;

— «комп'ютерів-демонів», які, отримавши команду від консолі управління, віддають команду «зомбованим машинам»;

— атакуючих агентів — це керовані заражені пристрої, які відправляють запити на кінцеву ціль.

Загальна схема структури DDoS-атаки зображена на рисунку 2.1. Відслідкувати структуру в зворотному напрямку від кінцевої цілі до консолі управління, з якої організовано атаку та виявити його адресу, майже неможливо. В найкращому результаті, що можна визначити адреси частини агентів, які атакували сервер. В кращому випадку певні заходи приведуть до «комп'ютера-демону». Проте при DDos-атаках що комп’ютери агенти, що «комп’ютери-демони» самі являються жертвами взлому.

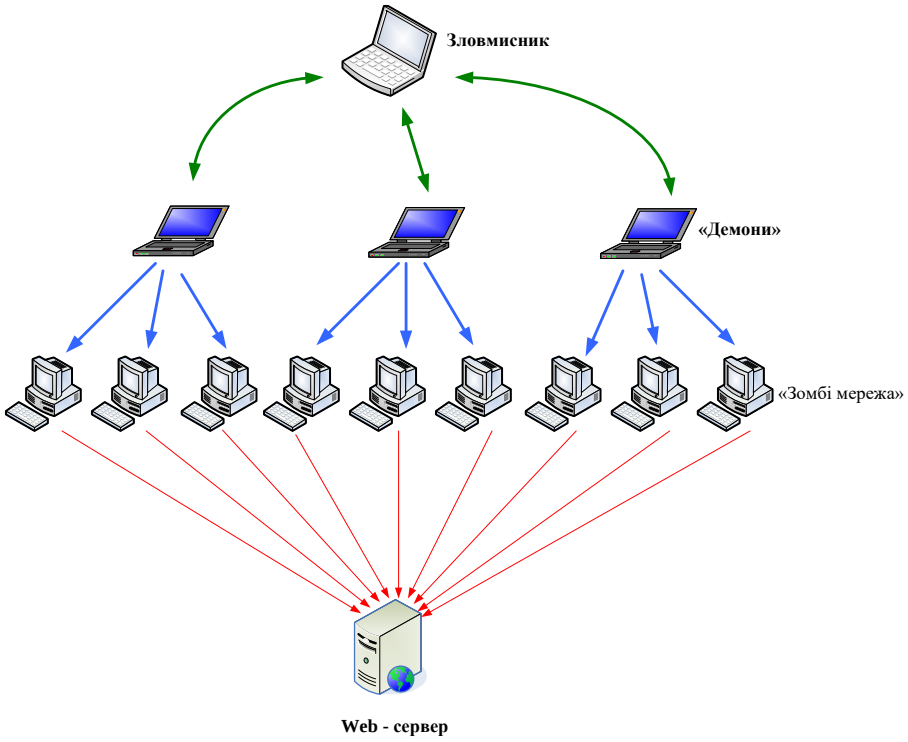

Рисунок.2.1 — Архітектура DDoS-мережі

«Зомбі-мережа» створюється шляхом зараження комп'ютерів (як правило, це домашня машина, підключена до виділеного каналу) троянською програмою. Ця програма потрапляє на комп'ютер користувача, найчастіше, при необережному поводженні з електронною поштою, наприклад, відкриття вкладень в листах, або при відвідуванні зараженого сайту, коли зловмисник може, використовуючи вразливості браузера, програмного забезпечення або операційної системи, встановити на пристрій користувача шкідливе ПЗ. Подібна програма може протягом довгого часу нічим деструктивним себе не проявляти. Часто власник пристрою навіть не підозрює, що його комп’ютер заражений та може повністю контролюватись зовні.

Команда до атаки віддається, наприклад, у чаті. Господар пише фразу, яка містить адресу сайту-жертви. Мережа атакуючих агентів починає працювати. Запити надходять на сервери з багатьох різних точок мережі, йдуть з великою високою частотою, в наслідок чого атакована цій починає не справлятися зі занадто великою кількістю запитів, перестає відповідати на запити легітимних користувачів, після чого зависає.

Найбільші DDoS-мережі: [16]

— Kraken — 400 тисяч комп'ютерів;

— Srizbi — 315 тисяч комп'ютерів;

— Bobax — 185 тисяч комп'ютерів;

— Rustock — 150 тисяч комп'ютерів;

— Storm — 100 тисяч комп'ютерів;

— Psybot — 100 тисяч ADSL-маршрутизаторів, заснованих на Linux;

— Ботнет BBC — 22 тисячі комп'ютерів, експериментальний ботнет, якийстворений компанією BBC.

В 2000-му році виведеними з ладу виявилися web-сайти eBay, CNN, Yahoo та інших великих сайтів (атака принесла сукупні збитки у розмірі $ 1,2 млрд). [17]

2001 рік — DDoS-атака на web-сайт Microsoft (компанія втратила близько $ 500 млн. всього за кілька днів);

2002 рік — атака на кореневі DNS-сервери мережі Інтернет. Під час атаки на певний час були виведені з роботи 7 із 13 серверів; [16]

2003 року — DDoS-атака на LiveJournal.com. Протягом двох днів сайт майже не працював, лише іноді подаючи ознаки життя;

2010 — Google і Adobe піддалися DDoS атаці через вразливості Internet Explorer.

2016 — численні атаки ботнетом Mirai, внаслідок чого впало багато популярних сайтів, зокрема Twitter , Netflix, GitHub, PayPal, HBO, Reddit та Airbnb.

2020 — атака наклієнтів Amazon Web Services, атака відбувалась 3 дні.

2023 — зафіксована рекордна DDoS-атака в 71 мільйон запитів за секунду.

За статистикою Департаменту ІБ компанії ТТК в 2005 році було зафіксовано 26 DDoS-атак, у 2006 році — 53, у 2009 — 114, у 2016 — 36000, у 2018 — 61000, тобто спостерігається суттєве щорічне зростання DDoS-атак. Причому, зростає як частота, так і тривалість атак.

2.2 Класифікація DDoS атак

Виділяється декілька видів DDoS-атак за способом реалізації та результатом атаки.

Перевантаження смуги пропускання каналу зв’язку (bandwidth consumption). Такі атаки базуються на тому, що атакуючий заповнює запитами смугу пропускання каналу. В результаті через забитий канал не можливо відправити на сервер легітимні запити. Інформація зі сервера на певний час стає недоступною для легітимних користувачів.

Нестача ресурсів (resource starvation). Це атаки, які направлені на захоплення критичних ресурсів системи, зокрема процесорний час, пам'ять чи місце на диску. Resource starvation часто дуже схожа на bandwidth consumption: зловмисники відсилають безліч запитів на сервер. Але на цей раз пакети не забивають канал хоста-жертви, а займають, скажімо, весь його процесорний час. Адже на обробку кожного пакету сервер витрачає деякий процесорний час (флопс). Результат — всі інші процеси не можуть виконуватись, а користувачі не можуть отримати доступу до серверів.

Помилки програмування (programming flaw). Атаки спрямовані на «баги», тобто слабкі місця, програмні помилки які реалізовані при розробці випадково, а також недокументовані функції процесорів, ОС, ПЗ і програмованих мікросхем. Знаючи їх вразливості, можна відправити на потрібний сервіс певний пакет, через який виникне певна помилка, переповнення буфера або стека. В результаті цього можливі тяжкі наслідки для всієї системи. [18]

Маршрутизація та DNS. Маючи доступ до маршрутизатора, з’являється можливість редагувати таблицю маршрутизації так, щоб бажаючі перейти на сервер з вказаною IP-адресою, переходили на іншу IP-адресу, або на адресу, яка не існує. Аналогічну атаку можна реалізувати, маючи доступ до DNS-сервера, але для сайтів. Маючи доступ до кешу DNS, його можна редагувати та прив'язати доменне ім'я до іншого IP адресу, в результаті цього користувачі будуть переходити на зовсім інший сервер, а не туди, куди вони мали б потрапити.

Особливістю цих атак є те, що сам атакований сервер продовжує нормально працювати, в той час як його користувачі не можуть на нього потрапити.

Flood. Даний тип атак можна віднести до попередніх DoS, проте він має свої особливості. З певної кількості пристроїв посилають жертві велику кількість запитів. Від такого об’єму жертва не встигає відповідати на всі запити, в результаті чого не відповідає на запити легітимних користувачів, тобто перестає нормально функціонувати. Опис найбільш поширених атак типу Flood наведено далі.[19]

SYN Flood — при даному виді атаки на вузол, що атакується направляється велика кількість SYN-пакетів по протоколу TCP. При цьому на атакованому сервері через короткий час вичерпується кількість відкритих сокетів і сервер не відповідає.

UDP Flood — цей тип флуду атакує не комп'ютер, а його канал зв'язку. Провайдери резонно припускають, що UDP більш пріоритетний, ніж TCP. Великою кількістю UDP-пакетів різного розміру викликається перевантаження каналу зв'язку, і сервер, що працює по протоколу TCP, перестає відповідати.

ICMP Flood або Ping Flood — те ж саме, що SYN Flood тільки пакетами ICMP. Система повинна відповісти на такий пакет, тим самим створюється велика кількість пакетів, які знижують продуктивність каналу.

Identification Flood (Ident Flood) — схожий на ICMP Flood, але відповідь на запит на порт 113 типу identd займає у системи більше часу, тому атака більш ефективна.

DNS Flood — атака спрямована на DNS сервер. На сервер надсилають велику кількість DNS запитів, на які сервер не встигає відповідати, таким чином, на Ваші запити він так само відповісти не зможе. Як наслідок, ви не можете відвідувати Інтернет сайти.

DDoS DNS — атака досить нова, і ще немає «встановленої» назви. По суті, виконується вона так само, що й попередня, з тією лише різницею, що запити надходять з великої кількості машин (попередній тип цього не виключає). Адреса, за якою повинен відповісти DNS-сервер на ці запити, дорівнює адресі самого DNS сервера, тобто його не тільки наповнюють запити DNS, але він же ще й відправляє їх собі ж. Таким чином, прийом більш ефективний, ніж попередній, але й більш складний у реалізації.

Boink (Bonk, Teardrop) — на сервер направляється величезна кількість сильно фрагментованих пакетів, але при цьому фрагменти великого розміру. Для кожного фрагментованого пакета виділяється спеціальний буфер, в який надалі будуть поміщені інші фрагменти, щоб потім скласти їх в одне ціле. Величезна кількість великих фрагментів переповнюють буфера і можуть спровокувати зависання або аварійну зупинку.

HTTP flood (POST, GET) — один з найбільш поширених сьогодні способів «флуду». Базується на нескінченному надсиланні HTTPS-повідомлень на 443-й порт з метою завантажити web-сервер настільки, щоб він не зміг обробляти всі інші запити від легітимних користувачів. Часто метою «флуду» стає не ядро web-сервера, а один з скриптів, які реалізовують ресурсомісткі задачі або працює з БД.

2.3 Методи захисту від DDoS-атак

Боротьба з DDoS-атаками — досить важка справа. Одна з головних причин — дуже важко встановити організатора атаки, а користувачі, з пристроїв яких генерують паразитичний трафік, зазвичай не підозрюють, що їхні комп’ютери стали інструментом в руках злочинців. Також майже неможливо відрізнити шкідливий трафік, тому що такі запити являються такими ж, що й запити від легітимних користувачів, але відправляються в гігантській кількості. [20]

Аналіз дивного мережевого трафіку — це єдиний ефективний метод для виявлення DDoS-атаки. DDoS-атаки в плані захисту — це один з найбільш складних видів атак, тому використання ефективних заходів протидії є досить складним завданням для компаній, робота яких суттєво залежить від мережі Інтернет. DDoS-атаку дуже складно виявити і запобігти, оскільки «шкідливі» пакети не відрізняються від «легітимних». Мережеві пристрої та традиційні технічні рішення для забезпечення безпеки мережевого периметра, такі як міжмережеві екрани, системи виявлення вторгнень (IDS) та маршрутизація в «чорні діри», є важливими складовими загальної стратегії мережевої безпеки, але тільки ці пристрої не забезпечують повного захисту від DDoS-атак.

Процес маршрутизації в «чорні діри» застосовується провайдером послуг для блокування всього трафіку, адресованого на цільовий об'єкт, в якомога швидший термін. «Знятий з маршруту» трафік маршрутизується в «чорну діру» для захисту мережі провайдера та інших його клієнтів. Маршрутизацію в «чорні діри» не можна назвати вдалим рішенням, оскільки разом із зловмисними трафіком атаки відбраковуються і пакети правомірних користувачів. Жертви повністю позбавляються свого трафіку, і хакер святкує перемогу.

Багато хто вважає, що маршрутизатори, на яких застосовуються списки контролю доступу (ACL) для фільтрації «небажаного» трафіка, забезпечують захист від атак DDoS. Дійсно, списки ACL можуть захистити від простих і відомих атак DDoS, наприклад, від ICMP-атак, фільтрація другорядних, невикористовуваних протоколів.

Однак на сьогоднішній день в атаках DDoS, як правило, використовуються коректні діючі протоколи, які необхідні для присутності в мережі Інтернет, і тому фільтрація протоколів стає менш ефективним засобом захисту. Маршрутизатори також можуть блокувати зони з некоректними IP-адресами, однак хакери, щоб їх не виявили, зазвичай підробляють коректні IP-адреси. В цілому, хоча списки ACL на маршрутизаторах служать першою лінією оборони від базових атак, вони не оптимізовані для захисту від далі описаних складних DDoS-атак.[21]

SYN, SYN-ACK, FIN та інші лавинні атаки. Списки ACL не можуть заблокувати атаку SYN з довільним вибором об'єктів спуфінга (проставлення в поле зворотної адреси IP-пакета невірної адреси) або атаки ACK і RST на 443-й порт web-сервера, при яких підроблені IP-адреси джерела постійно змінюються, оскільки для цього треба було б вручну відстежити та ідентифікувати кожне підроблене джерело, а це завдання практично нездійсненне. Єдиний можливий варіант тут полягає в тому, щоб заблокувати весь сервер, а саме в цьому і полягає завдання хакера.

Proxy. Оскільки списки ACL не можуть відрізнити один від одного «правомірні» і «зловмисні» SYN, що надходять з одного джерела IP або Proxy, то, намагаю-

чись зупинити сфокусовану атаку зі спуфінга, вони змушені, за визначенням, блокувати весь трафік клієнтів жертви, що надходить з певного вихідного IP або Proxy (модуля доступу).

DNS або протокол граничного шлюзу (Border Gateway Protocol, BGP). Коли запускаються атаки з довільним вибором об'єктів спуфінга на сервер DNS або на маршрутизатор BGP, списки ACL, як і у випадку з лавинними атаками SYN, не можуть відстежити швидко змінний обсяг трафіку з довільно вибраними об'єктами спуфінга. Крім цього, списки ACL не в змозі відрізнити підроблені адреси від коректних.

Атаки на рівні додатків (клієнтські). Хоча списки ACL теоретично можуть блокувати клієнтські атаки, наприклад, атаки з помилковими з’єднаннями HTTPS і з напіввідкритими з’єднаннями HTTPS (за умови, що є можливість точно ідентифікувати джерело атаки і конкретні не підроблені джерела), користувачам буде потрібно конфігурувати сотні, а в деяких випадках і тисячі списків ACL для кожної потенційної жертви.

Хоча між мережеві екрани грають винятково важливу роль у системі безпеки будь-якої компанії, вони не створені саме як інструмент запобігання атак DDoS. Фактично, у міжмережевих екранів є ряд вихідних властивостей, які не дозволяють їм забезпечити повний захист від найвитонченіших сучасних атак DDoS. Насамперед, це відсутність механізму виявлення аномалій. Міжмережеві екрани в першу чергу призначені для контролю доступу в приватні мережі, і вони відмінно справляються з цим завданням. Один із шляхів виконання цього завдання — відстеження сеансів, які ініційовані зсередини (на «чистій» стороні) і адресовані на зовнішній сервіс, із подальшим прийом тільки особливих відповідей від очікуваних джерел на зовнішній стороні. Однак така схема не діє стосовно до таких сервісів як web, DNS, і до інших сервісів, які повинні бути відкриті для загального доступу, щоб була забезпечена можливість приймати запити. У подібних випадках міжмережеві екрани виконують операцію, яка називається «відкриванням каналу»: вони пропускають трафік HTTPS на IP-адреса web-сервера. Хоча такий підхід і забезпечує деякий захист, оскільки дозволені лише певні протоколи, адресовані на певні адреси,

він не дуже ефективний у боротьбі з атаками DDoS, оскільки хакери можуть без перешкод скористатися «дозволеним» протоколом (в даному випадку HTTPS) для перенесення трафіку атаки. Відсутність можливостей для виявлення аномалій означає, що міжмережеві екрани не можуть розпізнати ситуацію, в якій носієм атаки служать коректні дозволені протоколи.

Також в міжмережевих екранах відсутні ресурси боротьби зі спуфінгом. Якщо виявлена атака DDoS, міжмережеві екрани можуть заблокувати конкретний потік трафіку, пов'язаний з атакою, але не можуть застосувати заходи антиспуфінгу, щоб відокремити хороший, «легітимний» трафік від поганого, а саме ця операція важлива для захисту від атак, в яких використовується великий обсяг підроблених IP-адрес.

Процедури захисту від атак DDoS, ініційовані вручну, можна охарактеризувати словами «занадто мало, занадто пізно». Перша реакція жертви на атаку DDoS, як правило, полягає в тому, що він просить найближчого попереднього провайдера послуг з'єднання (це може бути провайдер Інтернет-послуг, провайдер послуг хостингу або магістральний) спробувати ідентифікувати джерело. Якщо адреси підроблені або їх занадто багато, цей процес може виявитися довгим і важким, і для його реалізації буде необхідно об'єднати зусилля багатьох провайдерів. Хоча джерело, можливо, і буде ідентифіковане, блокування цього джерела виллється в блокування всього трафіку — і поганого, і хорошого.

Фахівці компаній можуть використовувати для протистояння атакам DDoS різні стратегії, зокрема, застосування резервних ресурсів, тобто закупівлю резервної смуги пропускання або резервних мережевих пристроїв, які допоможуть впоратися з будь-яким піковим зростанням попиту. Такий підхід не відрізняється високою рентабельністю, особливо через те, що необхідно вводити резервні мережеві інтерфейси і пристрої. І, незалежно від початкового ефекту, для того щоб здолати ці додаткові потужності хакерам знадобиться лише збільшити масштаби атаки. [22]

2.4 Аналіз архітектури систем захисту від DDoS-атак

Вчасно виявити DDoS-атаку — це і є головна проблема, щоб не довелось боротись з результатом падіння мережевих ресурсів. Один з найбільш ефективний способів встановити факт DDoS-атаки базується на на накопиченні статистичних даних про історію мережевого трафіку. Склавши план нормального стану мережі, можна виявити виникнення певної аномалії в трафіку мережі.

Як джерело даних для збору статистики можна використовувати сам трафік, який надходить в мережу або певну статистичну інформацію про нього. Для збору статистики використовуються або додаткові сенсори, які ставляться на мережу або вже встановлені мережеві елементи, які здатні надати інформацію про мережу. Схема протидії DDoS-атакам наведена на рисунку 2.2. [23]

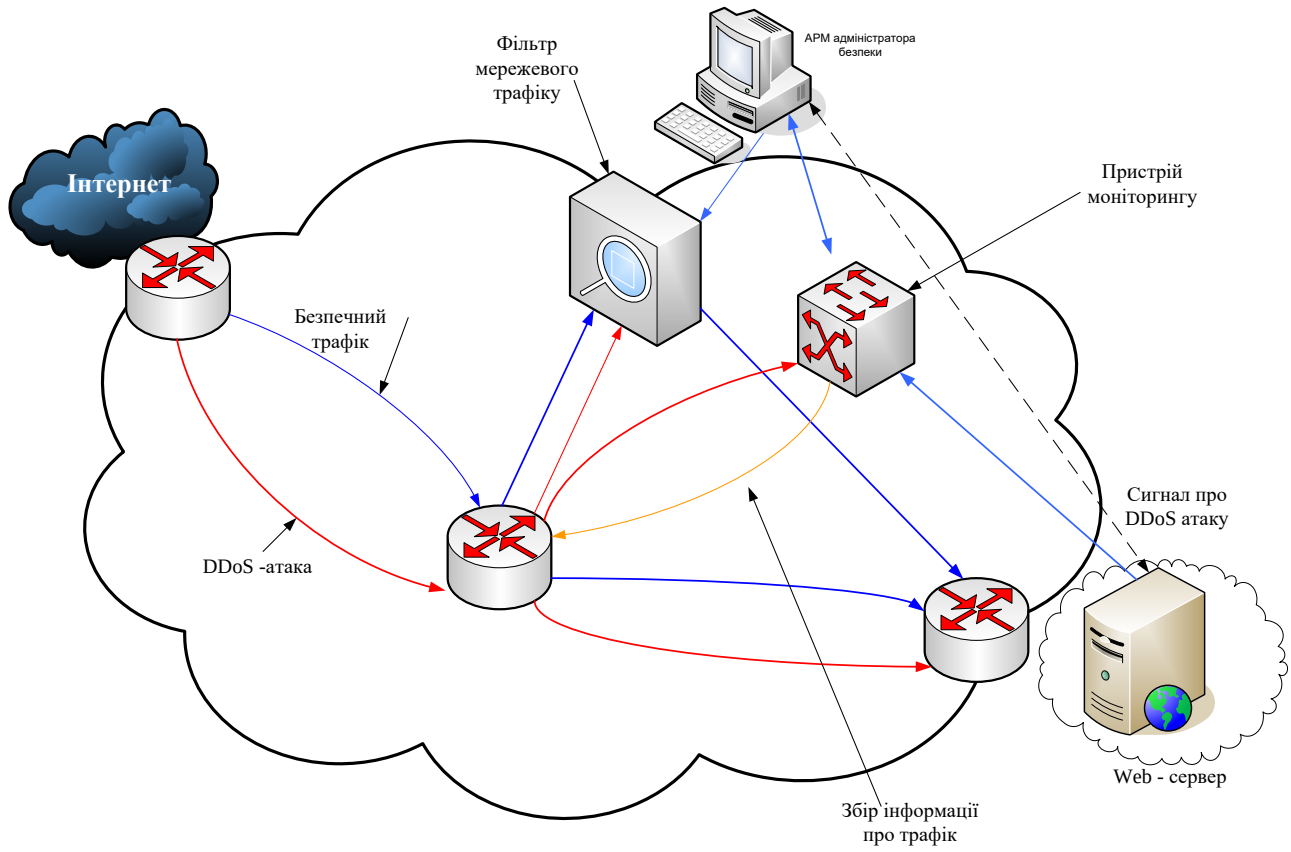

Рисунок 2.2 — Схема протидії DDoS-атакам

У разі отримання такої інформації напряму з маршрутизаторів використовується протокол Netflow. Даний протокол був створений для оптимізації роботи маршрутизаторів, його завдання полягало в тому, щоб не обробляти кожен пакет, а

перенаправляти його якомога швидше, якщо він відповідав вимогам потоку. Протокол виявився неефективним для вирішення основного завдання, але дуже став у нагоді для боротьби з DDoS-атаками і ширше — для аналізу роботи мережі в цілому. Такі можливості протоколу дає закладена в нього можливість формувати таблицю, в якій у динамічному режимі прописуються всі статистичні дані по вхідних потоках і пакетах: звідки прийшов пакет, куди він прямує, який у нього протокол, порт, яка кількість даних передано. Причому є можливість експорту статистичних даних в зовнішні системи для подальшого аналізу.

При штатній роботі, за відсутності DDoS-атак на ресурси, система проходить етап тестування або навчання. Вона визначає і запам'ятовує, який трафік для цього захищеного ресурсу є штатним. Випадки, при якій поточний трафік на даний захищений ресурс суттєво відрізняється від штатного, вважається DDoS-атакою. Слід зауважити, що система розпізнає лише відхилення від трафіку, а встановити причину, або це сплеск легітимних звернень до захищених ресурсів (виклали новий патч, пройшла рекламна кампанія), або це DDoS-атака може виявити тільки власник ресурсу, чи очікувався такий обсяг звернень від легітимних користувачів, чи ні.

Після визначення факту, що є аномалія, відбувається її класифікація і з’ясовується, наскільки аномалія серйозна. Якщо DDoS-атака не загрожує появою проблем у мережі, то краще спостерігати і нічого не робити, так як виникає шанс не пустити на ресурс легітимного користувача.

Технічна реалізація даного рішення передбачає наявність у мережі двох додаткових пристроїв: перший здійснює моніторинг усього вхідного трафіку і виявляє факт нанесення DDoS-атаки; другий фільтрує вхідний трафік мережі, який надходить ззовні.

У нормальному режимі роботи дані пристрої не повинні робити ніякого впливу на поточний трафік. У разі ж атаки пристрій «очищення» затримує трафік, ідентифікований як DDoS-пакети, не допускаючи його попадання у відносно вузькосмугові клієнтські канали і на клієнтські ресурси, тим самим не перериваючи надання клієнту основної послуги.

2.5 Аналіз існуючих систем захисту від DDoS-атак

На ринку існують два основні вендори, які розробляють спеціалізовані системи для захисту від DDoS-атак операторського класу (Arbor Networks та Cisco Systems), а також розробляють продукти, які використовуються для захисту від атак менших обсягів або орієнтованих на окремі ситуації. В залежності від потужності атак і популярності сайту, захист від них може коштувати від тисяч до десятків тисяч доларів щомісячно. Зокрема, обладнання від Cisco Systems, може коштувати $ 40 000.

Технологія Cisco Clean PIPes використовує два модулі: Cisco Guard та Cisco Anomaly Detector та різні системи аналізу статичного мережевого трафіку, які базуються на даних, отриманих із маршрутизаторів та допомогою Cisco Netflow. Модуль Anomaly Detector і системи аналізу статистичного трафіку виконують роль виявлення DDoS-атак, а модуль Cisco Guard виконує функцію протидії виявленої атаки. В порівнянні з іншими способами захисту від DDoS-атак, технологія Cisco Clean PIPes щодо захисту від атак дає можливість не тільки блокувати атакуючі адреси, а розділити легітимний та шкідливий трафік. Захист мережі від DDoS-атак гарантує три основні функції, які пропонує технологія. [24]

Виявлення. Під час роботи мережі в нормальних умовах, відбувається вивчення стандартних характеристик трафіку, на основі яких визначаються аномалії в мережевому трафіку (відхилення від стандартних характеристик), які інформують про виявлення аномалії. Будь-яке відхилення в мережевому трафіку, який перевищує певну межу, викликає спрацьовування сигналу тривоги.

Придушення. Це процес «чистки» трафіку (визначення аномалій, анти-спуфінгу, перевірка пакетів та «очищення» нелегітимних: відкидання зловмисного трафіку та дозвіл проходження легітимного до кінцевих користувачів).

Зміна маршруту трафіку та його повернення. У межах зміни маршруту трафіку на маршрутизатор в основній мережі, направляється перелік дій про зміну маршруту проходження нелегітимного трафіку (пакети зі зміненим адресою, великий потік зі встановленим бітом SYN та інші) так, щоб трафік проходив через пристрій

з модулем Cisco Guard. Після очищення нелегітимного трафіку, пристрій Cisco Guard повертає такий трафік назад у мережу. Схема роботи компонентів Cisco Clean PIPes наведена на рисунку 2.3.

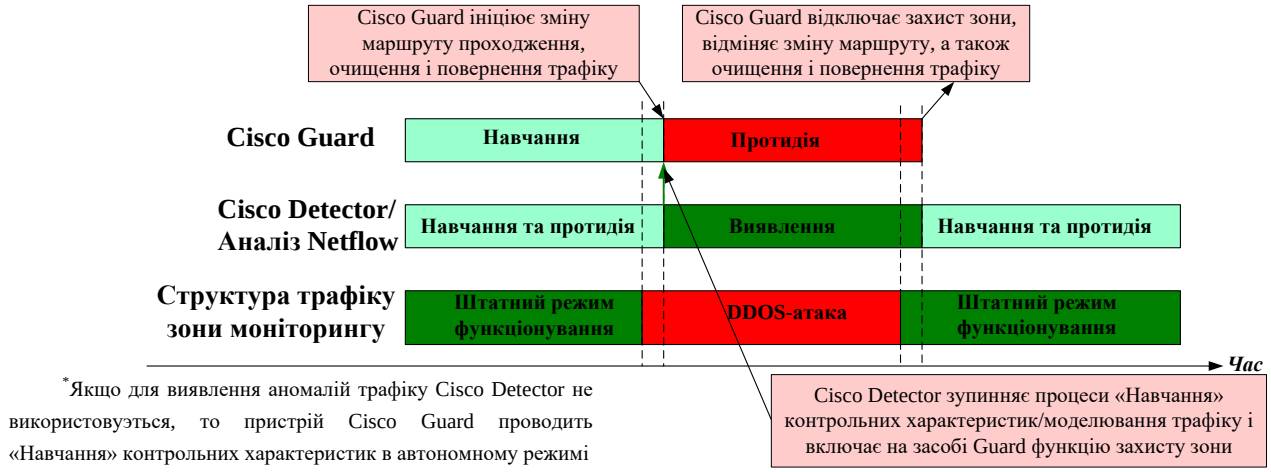

Рисунок 2.3 — Схема роботи компонентів Cisco Clean PIPes [24]

Cisco Clean PIPes — це комплексна глобальна технологія, яка складається з системи виявлення аномалій та зупинки DDoS-атак. В порівнянні з існуючими способами захисту від атак, технологія Cisco Clean PIPes може дуже точно відрізняти легітимний від нелегітимного трафіку, який прямує на важливий сервер, додаток чи хост.[22] Під час реалізації, дана технологія розгортається поруч з мережевими пристроями та легко масштабується, за рахунок цього зникають будь-які точки можливого збою та зберігається надійність і швидкодія існуючих наявних маршрутизаторів та комутаторів. Основний недолік технології — висока ціна. Лише дуже великі компанії можуть використовувати високовартісну установку обладнання від Cisco Systems.

2.6 Розробка методу захисту

Ados використовує основний робочий процес під назвою ados_daemon, який отримує доступ до пакетів TCP протокоу, які надсилаються клієнтами по протоколу HTTPS. Цей процес обробляє пакети через TCP/IP стек в ядрі системи і приймає рішення про те, чи треба їх пропустити або відкинути. Визначення долі пакета залежить від результатів пошуку його джерела в списку IP-адрес, який формується

під час роботи Ados або завантажується при запуску демона. Всього підтримується три таких списки IP-адрес:

— ban list (банліст, чорний список) — це IP-адреси, пакети з цього списку блокуються. Даний список має головний пріоритет;

— white list (білий список) — це IP-адреси, які не записуються в банліст незалежно від характеру і частоти надходження з них HTTPS-запитів;

— track list (трекліст) — це IP-адреси, з яких зареєстровані HTTPS-запити, але рішення по яких ще не винесено.

Кожному списку запитів присвоюється певний ваговий коефіцієнт, який встановлюється в конфігурації Ados. Цей коефіцієнт вираховується на основі навантаження на web-сервер, що створюється запитом даного типу, а також характеру запитів, які здійснюють легальні відвідувачі порівняно з ботами. Конфігурація також враховує роботу на сервері скриптів та інші фактори. [25]

Після того, як перший запит з будь-якої IP-адреси був зареєстрований, інформація про цю IP-адресу та кількість звернень кожного типу, виконаних з неї, заносяться до трек-списку і зберігаються в ньому протягом певного періоду часу, відомого як період аналізу. Кожен наступний запит з цієї IP-адреси, який зареєстрований протягом аналізу, збільшує лічильник звернень відповідного класу. Після закінчення аналізу вираховується середня частота запитів шляхом розрахунку відношення загальної кількості запитів до тривалості періоду.

$$rate = \frac{(C_1 \cdot W_1 + C_2 \cdot W_2 + \ldots + C_n \cdot W_n)}{trackperiod},$$

де C_i — кількість запитів одного класу;

W_i — ваговий коефіцієнт цього типу, який зрівнюється з межею спрацьовування flt_rate_threshold, який задається при конфігурації.

Після перевищення межі, IP-адреса переміщується в ban list, якщо IP-адреси немає у white list.Після переміщення IP в бан-лист паралельно з поточною обробкою пакетів запускається завдання зворотного резолвінгу (якщо резолвер не відключений в конфігурації). Якщо знайти ім'я хоста для цієї IP-адреси вдається, і воно відноситься до списку «білих» доменів, то IP-адреса переноситься в білий список і більше не буде блокуватись.

Ados — система порівняно недорога, і має універсальний метод боротьби з атаками DDoS.

Існує 2 види захисту — засобами web-сервера (наприклад, Apache або Nginx) і засобами додаткових модулів (наприклад, засобами php, perl, bash). Хороший результат може дати комбінація цих способів.

Перед початком атаки, боти «розганяються», збільшуючи потік пакетів на атаковану машину. Важливо зловити момент і почати активні дії.

Для початку потрібно підрахувати кількість процесів Apache і кількість з’єднань на 443-й порт:

Ps aux | grep HTTPSd | wc-l

Netstat-na | grep ": 443 \" | wc-l

Числа, які у кілька разів більші за середньостатистичні, потребують провести більш детальне вивчення. Далі слід переглянути список IP-адрес, з яких йдуть запити на підключення:

Netstat-na | grep ": 443 \" | sort | uniq-c | sort-nr | less

Стовідсотково ідентифікувати DDoS-атаку нереально, можна тільки підтвердити свої здогадки про наявність такої, якщо одна адреса повторюється в списку надто багато разів. Додатковим підтвердженням буде детальна перевірка пакетів за допомогою команди tcpdump:

Tcpdump-n-i eth0-w attack.log dst port 443 and host IP_серверу

Команда збереже в файл attack.log всі запити на 443 порт. Показником буде служити великий потік одноманітних пакетів від різних IP-адрес, спрямованих на один сервіс (наприклад, корінь web-серверу або певний cgi-скрипт).

Tcpdump-nr attack.log | awk '{print $ 3}' | grep-oE '[0-9] {1,} \. [0-9] {1,} \. [0-9] {1, } \. [0-9] {1,} '| sort | uniq-c | sort-rn | head -20

Буде виведено 2 стовпці: в першому кількість підключень, у другому IP-адресу. Чим більше підключень для однієї IP-адреси тим з більшою ймовірністю, що це бот. [7]

Фільтрація пакетів по URL за допомогою розширення iptables. У iptables реалізовано розширення string, яке дозволяє виконувати фільтрацію пакетів, грунтуючись на аналізі вмісту області даних пакета.

Якщо атака йде на одну сторінку сайту, наприклад, надсилаються запити виду "GET / dir / site.php HTTPS/1.0", то за цим критерієм можна відкидати всі запити, що йдуть на цю сторінку:

Iptables -I INPUT -p tcp - dport 443-m string - string "GET / dir / site.php HTTPS/1.0" - algo kmp-j DROP

Таким чином, сайт захищений від DDoS-атаки і зберігає працездатність, нехай і ціною тимчасового відключення сторінки.

Якщо сайт орієнтований на цілком конкретну мовну групу, наприклад на українськомовних користувачів Інтернету, атаку на нього можна значно послабити, якщо сервер перестане обробляти запити з «небажаних» країн.

Фільтрувати трафік по країнах можна за допомогою iptables та розширення geoip.

Приклад користування:

Iptables -A INPUT - dport 25-m geoIP - src-cc UA-j ACCEPT

Iptables -A INPUT - dport 25-j LOG - log-prefix "Blocked smtp from not UA:"

Iptables -A INPUT - dport 25-j REJECT

Часто для захисту від не інтенсивних атак на сайт, використовується модуль apache mod_evasive. Цей модуль при перевищенні певного числа запитів до сайту з однієї IP-адреси, тимчасово блокує атакуючого, видаючи у відповідь на запити помилку 403 Forbidden. При незначних атаках на сайт це дозволяє знизити навантаження.

При більш серйозних атаках такий метод захисту не врятує, тому що навіть на запити зі «заблокованої» адреси web-сервера, необхідно генерувати відповідь (403 Forbidden) на кожний запит, що при великій кількості запитів приводить до швидкої витрати ресурсів.

Mod_evasive може викликати зовнішній скрипт. Таким чином можна організувати блокування IP-адреси атакуючого на рівні фаєрволу.

Слабке місце ботів у тому, що вони не розрізняють «cookie» (невеликий фрагмент даних, створений web-сервером і зберігається на комп'ютері користувача у вигляді файлу, який web-клієнт кожного разу відправляє web-серверу в HTTPS-запиті при спробі доступу до сторінки певного сайту). На цій особливості і засновані деякі системи захисту. Ґрунтуючись на тому, чи передав відвідувач «cookie» web-серверу чи ні, можна розділити трафік на легальний і «сміттєвий».

Спочатку потрібно приховати основний web-сервер, наприклад призначивши на 8888 порт, а на його місце поставити швидкий Nginx, і сконфігурувати сторінку-заглушку. Як приклад підходить стандартна конфігурація nginx, єдине, що потрібно зробити — додати «cookie», дописавши в конфігураційному файлі nginx:

add_header Set-Cookie "type = this-is-not-bot";

На самій сторінці-заглушки можна написати повідомлення, наприклад «Вибачте, сайт під DDoS-атакою, увімкніть «cookie» в браузері і натисніть на посилання для переходу». [12]

Коли відвідувач відкриє таку сторінку, то при наступному запиті він передасть «cookie», на основі якого можна направити трафік до прихованого web-серверу:

Iptables -t nat -A PREROUTING -p tcp - dport 443-m string - string "this is--not-bot" - algo kmp -j REDIRECT - to-ports 8888

У Nginx реалізований модуль ngx_http_limit_zone_module, який дозволяє обмежувати кількість з'єднань для заданої сесії або, як окремий випадок, з однієї IP-адреси.

Можна зробити припущення про те, що легітимні користувачі роблять не більше 2-х одночасних запитів до однієї сторінки сайту. Вважати клієнтів, які відкрили більше 3х одночасних з'єднань атакуючими ботами і забороняти їх доступ через визначення IP-адреси на фаєрволі.

Для реалізації пошуку ботів треба створити спеціальну обробку запитів в Nginx:

```
error_log / var / log / nginx / error.log;
<...>
location = / {
limit_conn one 3;
root / var / www / domain.ua;
}
```

IP-адреси, з яких було відкрито більше 3х одночасних підключень, будуть записані в error.log з повідомленням limiting connections by zone. На основі логу помилок можна побудувати blacklist IP атакуючого ботнету. На рисунку 2.4 наведена схема, яка реалізує таку фільтрацію адрес.

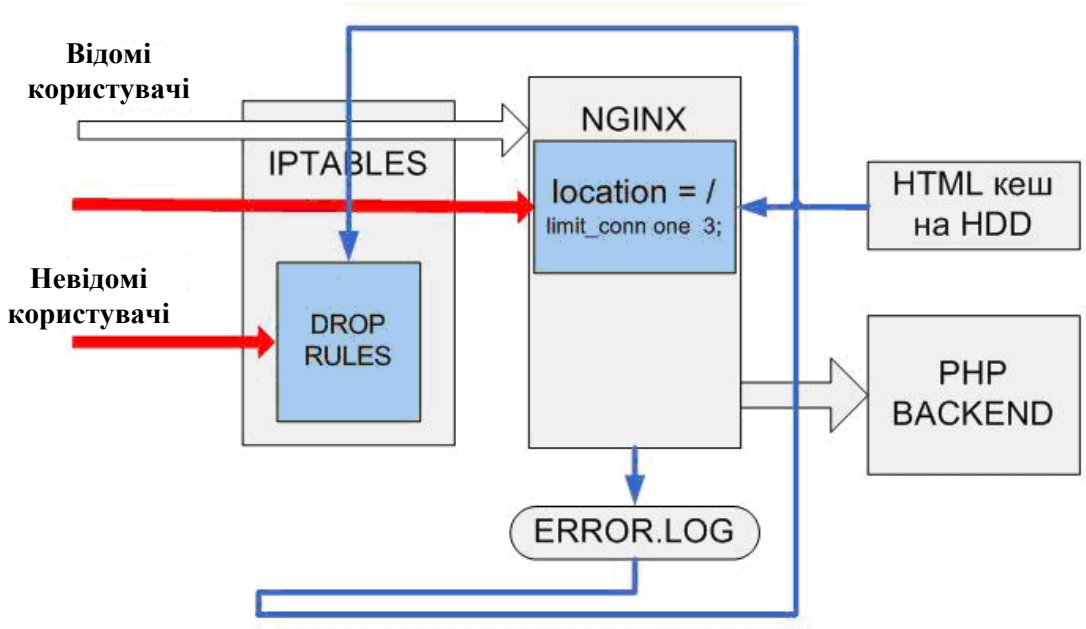

Рисунок 2.4 — Схема фільтрації адрес, що перевищують ліміт з'єднань [26]

Фільтрацію пакетів по URL можна реалізувати засобами Nginx. Наприклад, за критерієм POST index.php? Action = login з порожнім посиланням (URL джерела запиту) можна реалізувати так:

```
Set $ add 1;
location / index.php
{
limit_except GET POST
{
deny all;
}
Set $ ban "";
If ($ HTTPS_referer= "") {set $ ban $ ban $ add;}
If ($ request_method = POST) {set $ ban $ ban $ add;
}
If ($ query_string = "action = login") {set $ ban $ ban $ add;}
If $ an 111)
{
access_log/var/log/nginx/ban IP;
return 404;
}
proxy_pass HTTPS: //127.0.0.1:8888;
}
```

3 ПРОЕКТУВАННЯ СТРУКТУРИ СИСТЕМИ ЗАХИСТУ ВІД DDOS-АТАК

3.1 Вибір класифікатора трафіку

Баєсівський класифікатор є методом класифікації, який використовує теорему Баєса для визначення ймовірності того, що спостереження (елемента вибірки) до одного з класів C при тому, що значеннях залежних змінних набувають $P(C|F_1, \ldots, F_n)$. Це означає, що за умови, що значення змінних відомі, баєсівський класифікатор визначає ймовірність приналежності спостереження до певного класу C. Якщо ця ймовірність дорівнює 1, класифікатор повідомляє, що спостереження належить до цього класу. У випадках, коли спостереження може належати до різних класів з різною ймовірністю, результатом роботи класифікатора являється вектор, в якому кожний компонент представляє ймовірність приналежності до відповідного класу. [27]

Ідеальний баєсівський класифікатор можна вважати оптимальним в певному сенсі. Його результат не може бути покращений, тому що в усіх варіантах, коли існує однозначна відповідь, він її надасть. В тих же випадках, коли відповідь неоднозначна, результат класифікації кількісно відображає ступінь цієї неоднозначності.

Однак, основним недоліком ідеального баєсівського класифікатора є необхідність мати вибірку, яка містить всі можливі комбінації змінних — а розмір такої вибірки експоненціально зростає зі зростанням числа.

Для подолання описаної проблеми у практичні реалізації використовується наївний баєсівський класифікатор. Даний класифікатор базується на припущенні про незалежність змінних, тобто припущенні про те, що

$$\forall i, j P\left(F_i|C, F_j\right) = P(F_i|C).$$

Використання цього класифікатора дозволяє уникнути вивчення взаємодії всіх можливих поєднань змінних і зосередитись тільки на впливі кожної змінної,

незалежно до приналежності об'єкта до одного з класів. Згідно теоремі Баєса, в цьому випадку визначається наступна формула:

$$P(C|F_1 \dots F_n) = const * P(C) * P(F_1|C) * \dots * P(F_n|C).$$

Перевагою наївного баєсівського класифікатора є скорочення вимог до розміру вибірки з експоненціальних до лінійних. Однак, недоліком наївного баєсівського класифікатора є те, що він припускає незалежність змінних, і тому точність моделі може знижуватися, коли це припущення не виконується. В таких випадках обчислені ймовірності можуть бути неточними, і їх сума може не дорівнювати одиниці, що потребує нормування результатів. Проте незначні відхилення від припущення про незалежність змінних зазвичай мають незначний вплив на точність наївного баєсівського класифікатора. Навіть у випадках, коли змінні взаємодіють між собою, класифікатор все ще може корелювати з істинною приналежністю об'єктів до класів. При цьому переваги використання наївного баєсівського класифікатора, такі як простота, масштабованість та швидкість роботи зазвичай переважують недоліки.

Безліч пар «запит-клас» $X \times Y$ є ймовірнісним множиною з невідомою ймовірнісної мірою P. Існує кінцева вибірка спостережень для навчання: $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$, яка згенерована згідно ймовірнісною мірою P. Потрібно побудувати функцію a: X → Y, здатну класифікувати довільний запит $x \in X$.

Комерційне рішення від Cisco для захисту від атак DDoS універсально і гарантовано фільтрує всі види DDoS-атак, але висока ціна обмежує можливість купувати його. Перевага його відносно інших систем полягає в тому, що ця система навчається на нормальному трафіку, на підставі якого визначаються аномалії структури трафіку при атаці.

Мережеві адміністратори придумали багато способів захисту, але всі вони не універсальні і не вирішуються відразу всіх завдань при виявленні потужної DDoS-атаки, а саме:

— зберегти працездатність сервісу;

— відсіяти максимальну кількість ботів і мінімум легітимних користувачів;

— не допустити блокування адрес пошукових систем.

Тому прийнято рішення розробити систему, яка буде враховувати всі переваги даних методів, а саме система буде:

— навчатися на «хорошому» трафіку;

— враховувати з якої IP-адреси йде запит;

— враховувати кількість запитів з однієї IP-адреси;

— враховувати країну, з якої йде запит;

— враховувати метод запиту;

— враховувати адресу підмережі;

— сторінку, яка запитується з даної IP-адреси;

— сторінку входу на сайт;

— тип браузера;

— тип операційної системи.

Задача класифікації запитів полягає у визначенні, до якого класу віднести запит:

— клас DDoS-ботів;

— клас легітимних користувачів.

Основна мета — класифікувати нові запити по мірі їх надходження, тобто потрібно вирішити до якого класу вони належать, використовуючи інформацію про належність до класу легітимних користувачів вже наявних запитів. Вибірка запитів, здійснюватиметься з журналів доступу web-сервера Apache (apache.log).

Розглядається умовна ймовірність приналежності запиту класу, при тому, що він має ознаки $F_1, F_2, \dots F_n$: $P(C|F_1, \dots F_n)$.

За теоремою Баєса:[28]

$$P(C|F_1, \dots F_n) = \frac{P(C) * P(F_1, \dots F_n|C)}{P(F_1, \dots F_n)}.$$

При визначення умовної ймовірності:

$$P(C|F_1, \dots F_n) = P(C) * P(F_1, \dots F_n|C) = \mathrm{P(C)} * \mathrm{P}(F_1|C) * \mathrm{P}(F_2, \dots F_n|C, F_1)$$
$$= \mathrm{P(C)} * \mathrm{P}(F_1|C) * \mathrm{P}(F_2|C) * P(F_3, \dots F_n|C, F_1, F_2).$$

Відповідно до «наївного» баєсівського підходу передбачається, що події незалежні для будь-яких $i \neq j$:

$$P(F_i, C) = P(F_i, C).$$

Відповідно, ймовірність віднесення запиту до одного з класів, можна обчислити за формулою:

$$P(C|F_1, \dots F_n) = P(C) * P(F_1|C) * P(F_2|C) * \dots * P(F_n|C) = P(C) * \prod_{i=1}^{n} P(F_i|C).$$

Відповідно до «наївної» баєсівського алгоритму, ймовірність приналежності запиту класу визначається за формулою:

$$P(D|C) = \prod_{i=1} P(w_i|C),$$

де w_i— це атрибути заголовку HTTPS-запиту, а саме:

— IP-адреса, з якої здійснюється вхід на сайт (IP);
— сторінка, яка запитується з даної IP-адреси (url);
— сторінка входу на сайт (referer);
— тип браузера (user_agent);
— тип операційної системи (os);
— країна, з якої йде запит (country);
— метод запиту (method).

Класифікація запитів відбувається за двома класами — DDoS-боти (C) і легітимні користувачі ($\overline{C}$), тому відповідно до формули Баєса утворюються два вирази [4]:

$$P(C|D) = \frac{P(C)}{P(D)} \prod_{i=1}^{n} P(w_i|C),$$

$$P(\bar{C}|D) = \frac{P(\bar{C})}{P(D)} \prod_{i=1}^{n} P(w_i|\bar{C}).$$

Розділивши один вираз на інший, в результаті буде:

$$\frac{P(C|D)}{P(\bar{C}|D)} = \frac{P(C) \prod_{i=1}^{n} P(w_i|C)}{P(\bar{C}) \prod_{i=1}^{n} P(w_i|\bar{C})}.$$

Взявши логарифм всіх цих ступенів, виходить такий вираз:

$$\ln \frac{P(C|D)}{P(\bar{C}|D)} = \ln \frac{P(C)}{P(\bar{C})} + \sum_{\text{ш}} \ln \frac{P(w_i|C)}{P(w_i|\bar{C})}.$$

В результаті, запит може бути класифікований наступним чином: це DDoS-бот, якщо $\ln \frac{P(C|D)}{P(\bar{C}|D)} > 0$, в іншому випадку це легітимний користувач.

Безумовну ймовірність справедливості гіпотези, тобто наскільки ймовірна причина взагалі, можна обчислити за формулою:

$$P(C) = \frac{k - k_n}{k_n},$$

де k — кількість запитів до web-серверу в поточний момент;

k_n— кількість запитів до web-серверу в середньому.

Умовну ймовірність, тобто наскільки ймовірна причина виявилася з урахуванням даних подій, можна обчислити за далі наведеними формулами.

Ймовірність, що дану сторінку запитує легітимний користувач:

$$P(w_i = url|\bar{C}) = \frac{m-m_n}{m_n},$$

де m — кількість запитів на дану сторінку;

m_n — загальне число запитів з навчальної вибірки.

Можливість, що з даної IP-адреси звертається легітимний користувач:

$$P(w_i = ip|\bar{C}) = \begin{cases} t, \text{якщо з даної IP адреси не надходили запити під час атаки} \\ \frac{k-k_n}{k_n}, \text{якщо запити з даної IP адреси надходили вперше} \\ \text{якщо запити з даної IP адреси не надходили} \\ \text{під час атаки} \begin{cases} 0.1, \text{якщо } \frac{m}{m_n} \geq 0.05 \\ 0.5, \text{якщо } \frac{m}{m_n} < 0.05 \end{cases} \end{cases}$$

де t — задається користувачем в конфігураційному файлі;

k — кількість запитів до web-сервера в поточний момент;

k_n — кількість запитів до web-сервера в середньому;

m_n — кількість запитів під час атаки на web-сервер;

m — кількість запитів з даної IP-адреси під час атаки на web-сервер.

Можливість, що з запитуваної сторінки звертається легітимний користувач:

$$P(w_i = referer|\bar{C}) = \frac{r-r_n}{r_n},$$

де r — кількість запитів з цієї сторінки;

r_n — загальне число запитів з навчальної вибірки.

Ймовірність, що даний метод запиту (GET, POST) в HTTPS-заголовку запитує легітимний користувачів:

$$P(w_i = method|\bar{C}) = \frac{d-d_n}{d_n},$$

де d — кількість запитів цього методу,

d_n — загальне число запитів з навчальної вибірки.

3.2 Проектування системи класифікації запитів

При великій кількості запитів по HTTPS протоколу, web-сервер Apache не встигає коректно обробляти запити і може вийти з ладу. Для отримання додаткових ресурсів для обробки більшої кількості запитів, останнім часом широко використовується дворівнева архітектура обробки запитів з двома web-серверами: frontend і backend, де як frontend сервера використовується Nginx — HTTPS-сервер і IMAP/POP3 проксі-сервер. На даний момент nginx працює на великій кількості високонавантажених сайтів.[28]

Існують три моделі роботи сервера, детальний опис яких наведено нижче.

Послідовна. Сервер відкриває прослуховування сокету і чекає, коли з'явиться з'єднання (під час очікування він перебуває в заблокованому стані). Коли приходить з'єднання, сервер обробляє його в тому ж контексті, закриває з'єднання і знову чекає з'єднання. Очевидно, це далеко не найкращий спосіб, особливо коли робота з клієнтом ведеться досить довго і підключень багато. Крім того, у послідовній моделі є ще багато недоліків (наприклад, неможливість використання декількох процесорів), і в реальних умовах вона практично не використовується.

Багатопроцесорна (багатопотокова). Сервер відкриває прослуховування сокетів. Коли приходить з'єднання, він приймає його, після чого створює (або бере з пулу заздалегідь створених) новий процес або потік, який може як завгодно довго працювати зі з'єднанням, а після закінчення роботи завершитися або повернутися в пул. Головний потік тим часом готовий прийняти нове з'єднання. Це найбільш

популярна модель, тому що вона відносно просто реалізується, дозволяє виконувати складні і довгі обчислення для кожного клієнта і використовувати всі доступні процесори. Приклад її використання — web-сервер Apache. Однак у цього підходу є й недоліки: при великій кількості одночасних підключень створюється дуже багато потоків (або, що ще гірше, процесів), і операційна система витрачає багато ресурсів на переключення контексту. Особливо погано, коли клієнти дуже повільно приймають контент. Виходять сотні потоків або процесів, зайнятих тільки відправленням даних повільним клієнтам, що створює додаткове навантаження на планувальник ОС, збільшує число переривань і споживає досить багато пам'яті.

Неблоковані сокети чи кінцевий автомат. Сервер працює в рамках одного потоку, але використовує неблоковані сокети і механізм поллінгу (періодичне опитування сервера). Тобто сервер на кожній ітерації нескінченного циклу вибирає з усіх сокетів той, що готовий для прийому/відправки даних за допомогою виклику select(). Після того, як сокет обраний, сервер відправляє на нього дані або читає їх, але не чекає підтвердження, а переходить в початковий стан і чекає події на іншому сокеті або ж обробляє наступний, в якому подія відбулась під час обробки попереднього. Дана модель дуже ефективно використовує процесор і пам'ять, але досить складна в реалізації. Крім того, в рамках цієї моделі обробка події на сокеті повинна відбуватися дуже швидко — інакше в черзі буде накопичуватися багато подій, в результаті чого вона переповниться. Саме за такою моделлю працює Nginx. Крім того, він дозволяє запускати кілька робочих процесів, тобто може використовувати кілька процесорів.

Таким чином, перед web-сервером Apache розміщується легкий і швидкий web-сервер Nginx якому дається можливість обслуговувати запити до статичних файлів, а запити до динамічних файлів проксуються до головного сервера. При такому рішенні сервер Apache не створює додаткових процесів для обробки статичних сторінок та файлів і віддає результати обробки динамічних запитів frontend-серверу дуже швидко, що дозволяє йому звільнити ресурси для використання в об-

робці інших запитів. Frontend-сервер може чекати скільки завгодно довго, поки клієнт забере свою «відповідь» і завершить з'єднання, а backend-сервер не витрачає ресурси для цього. [29]

Діаграма конфігурації web-сервера зображена на рисунку 3.1

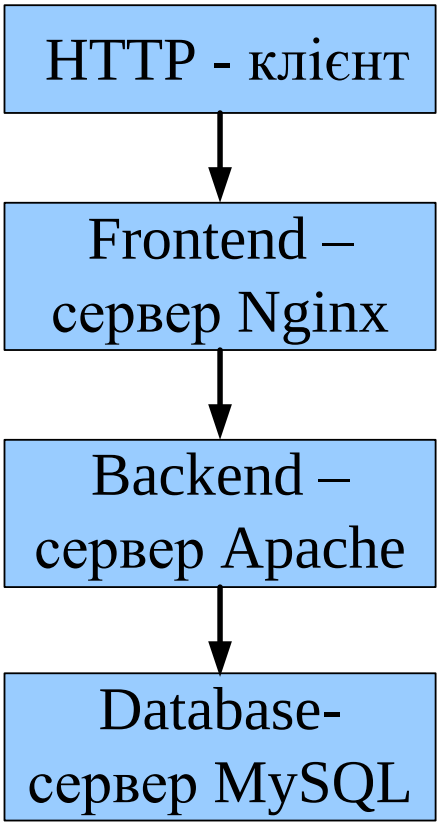

Рисунок 3.1 — Діаграма конфігурації web-сервера

Завдяки дворівневій архітектурі, істотно менше витрачається ресурсів операційної системи та пам'яті, завдяки чому web-сервер може обробити більше звернень. Але зі збільшенням потужності атаки створюється безліч потоків чи процесів на Apache, які не встигають генерувати контент і віддавати його Nginx, тому між frontend-сервером і backend-сервером розміщується програмв, яка буде класифікувати запити, що надходять на web-сервер Nginx і відкидати DDoS-ботів, не пропускаючи їх надалі до Apache.

При зверненні користувача запит надходить на frontend-сервер, і там або обробляється і повертається відповідь користувачеві, або передається далі в систему — на програму класифікації запитів (againstDDoS-daemon) і на web-сервер Apache.

Програма оцінює запит у відповідності з класифікацією по інтелектуальному алгоритму, що враховує запити користувачів до атаки на web-сервер, і залежно від результату класифікації забороняє обробку запитів від даного користувача за допомогою брандмауера, або не перешкоджає нормальній роботі серверів. Модернізована діаграма конфігурації наведена на рисунку 3.2.

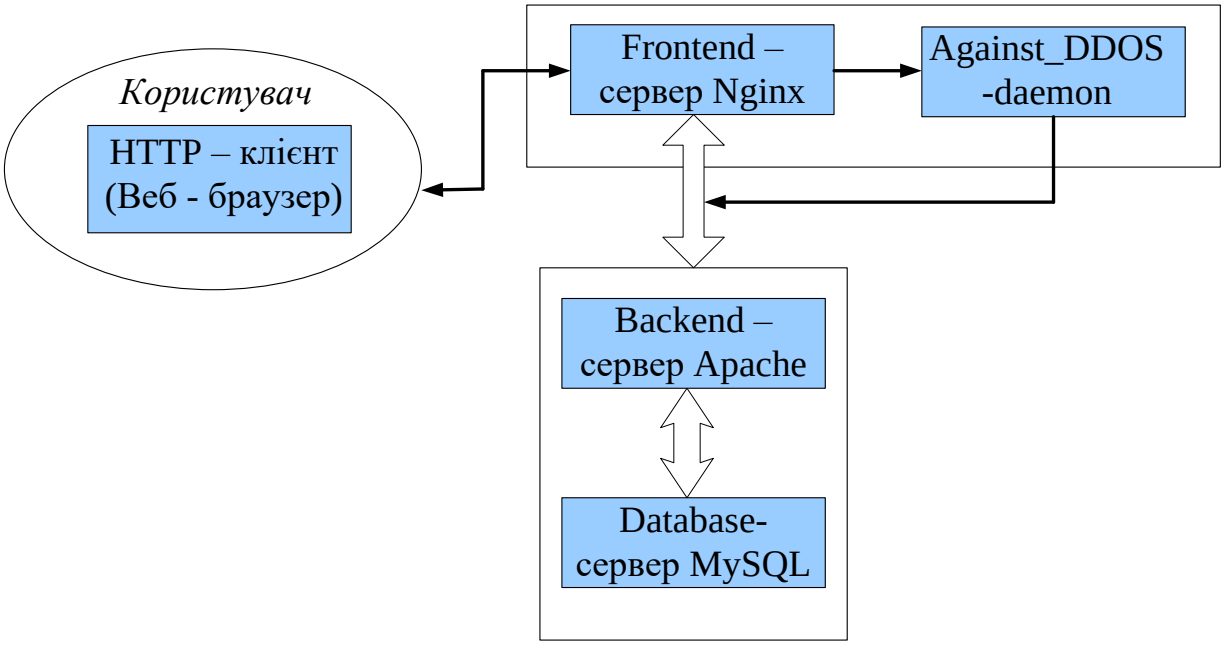

Рисунок 3.2 — Діаграма конфігурації web-сервера з AgainstDDoS-демоном

3.3 Конфігурація web-серверу

Основне завдання — налаштувати Nginx на прослуховування запитів зі зовнішніх адрес на 443-му порту, обробка статичних запитів і перенаправлення інших запитів на web-сервер Apache на локальну адресу і порт 442 (192.168.0.1:442).

Установка необхідного програмного забезпечення:

> Apt-get install nginx apache2

Також необхідно встановити пакет libapache2-mod-rpaf, щоб у apache-log записувалися реальні IP-адреси користувача, а не адреса frontend-сервера.

> Apt-get install libapache2-mod-rpaf

Конфігураційний файл Nginx (/ etc / nginx / nginx.conf) залишається в первинному вигляді (логи запитів записуються в / var / log / nginx / access.log).

Далі створюються настройки для підтримуваних ресурсів в / etc / nginx / sites-available / default:

```
server {
  listen 443;
  server_name domain.ua;
  access_log / var / logs / nginx-access.log;
  location /
  {
    proxy_pass HTTPS://192.168.0.1:442 /;
#переадресація запитів на apache
    proxy_redirect off;
    proxy_set_header Host $ host;
#ці установки необхідні, щоб зі скриптів було видно реальні IP-адреси користувача
    proxy_set_header X-Real-IP $ remote_addr;
    proxy_set_header X-Forwarded-For
    $ proxy_add_x_forwarded_for;
        client_max_body_size 10m;
        client_body_buffer_size 128k;
        proxy_connect_timeout 90;
        proxy_send_timeout 90;
        proxy_read_timeout 90;
        proxy_buffer_size 4k;
        proxy_buffers 4 32k;
        proxy_busy_buffers_size 64k;
        proxy_temp_file_write_size 64k;
  }
}
```

Далі налаштовується apache-сервер на прослуховування 442-го порту з внутрішнього інтерфейсу, відредагувавши у файлі / etc/apache2/ports.conf наступні команди:

NameVirtualHost *: 442

Listen 192.168.0.1:442

І необхідно підправити налаштування хостів apache в / etc/apache2/sites-available/default, замінивши 443-й порт на 442:

<VirtualHost *:442>

Таким чином потрапити на Apache тепер можна тільки через Nginx, тобто зовні Apache недоступний.

Навчання класифікатора запитів проводиться за логами web-сервера Apache. Для збільшення швидкодії роботи програми класифікації запитів, при вибірці з логів, записуватимуться access log в СКБД MySQL. Для цього треба встановити модуль mysql libapache2-mod-log-sql-mysql.

Apt-get install apache2 mysql-server mysql-client libapache2-mod-log-sql-mysql

Створення бази даних:

Mysql-uroot-p-hlocalhost

Create database apachelogs;

Редагування конфігураційного файлу Apache:

Vi / etc/apache2/sites-available/default

LogSQLLoginInfo mysql: / / loguser: loguser_rootroot @ localhost / apachelogs

LogSQLCreateTables on

#чи створювати таблицю для зберігання логів

LogSQLDBParam socketfile / var / run / mysqld / mysqld.sock

#сокет MySQL

LogSQLTransferLogFormat AbHhmRSsTUuvI

#що писати, а що ні в таблицю з логами

<VirtualHost *:442>

ServerAdmin webmaster @ localhost

```
DocumentRoot / var / www /
<Directory /var/www/>
    Options Indexes MultiViews
    AllowOverride None
    Order allow, deny
    allow from all
</ Directory>
LogSQLTransferLogTable web1_access_log
#ім'я таблиці, в яку писати логи
</ VirtualHost>
LogSQLLoginInfo mysql: //loguser: loguser_rootroot @ localhost / apachelogs
#директива містить параметри підключення до бази даних
LogSQLCreateTables on
#вказує модулю mod_log_sql створювати таблиці, якщо вони не існують
LogSQLDBParam socketfile / var / run / mysqld / mysqld.sock
#визначає сокет MySQL
LogSQLTransferLogFormat AHhmRprSstUuv
#директива визначає які колонки створювати в таблиці
LogSQLTransferLogTable web1_access_log
```

У таблиці 3.1 наведено список полів з властивостями, які будуть використовуватись для аналізу запитів. [30]

Визначається ім'я таблиці, директива прописується у віртуальному хості конфігурації — таким чином, можна створити різні таблиці в журналі доступу для кожного віртуального хоста.

Таблиця 3.1 — Список полів та властивостей, які використовуються при аналізі запитів

№	Визначення	Імя поля	Тип поля	Приклад
1	User agent	agent	varchar(255)	Mozilla/5.0(compat; MSIE 6.0;Windows 10)
2	HTTPS протокол	request_protocol	varchar(255)	HTTPS/1.1

№	Визначення	Імя поля	Тип поля	Приклад
3	Ім’я віддаленого хоста	remote_host	varchar(255)	wind.flash.com
4	Метод запиту	request_method	varchar(255)	GET
5	PID процесу	child_pid	Smallint unsigned	3215

Продовження таблиці 3.1

№	Визначення	Імя поля	Тип поля	Приклад
6	HTTPS порт	server_port	Smallint unsigned	443
7	Реферер	referer	varchar(255)	HTTPS://www.testlinksforyou.com/page.html
8	Повний запит	request_line	varchar(255)	GET /foo.htm HTTPS/1.1
9	Час	time_stamp	int unsigned	1005598029
10	Статус запиту	status	Smallint unsigned	404
11	Час виконання запиту	request_time	char(28)	[25/May/2023:15:01:26 +3000]
12	Короткий запит	request_uri	varchar(255)	/books-cycroad.html
13	Логін (з процесу авторизації)	remote_user	varchar(255)	oleksii
13	Віртуальний хост	virtual_host	varchar(255)	user-agents.net

Після внесення всіх змін в конфігураційні файли, потрібно перезапустити Apache:

/ Etc/init.d/apache2 reload

Для перевірки створення таблиці, проводиться вхід на сервер MySQL:

Mysql-uroot-hlocalhost-prootroot

mysql> use apachelogs;

mysql> show tables;

+---------------------+

| Tables_in_apachelogs |

+---------------------+

| Web1_access_log |

+---------------------+

1 row in set (0.00 sec)

Обираються для прикладу всі колонки таблиці, де «Час» дорівнює «1005598029»:

mysql> SELECT * FROM web1_access_log WHERE time_stamp = 1005598029;

agent | request_protocol | remote_host | request_method | child_pid | server_port | referrer | request_line | time_stamp | status | request_time | request_uri | remote_user | virtual_host

Mozilla/5.0 (compat; MSIE 6.0; Windows 10) | HTTPS/1.1 | 192.168.1.15 | GET | 51 | NULL | - | NULL | 1005598029| 200 | [25/May /2023: 15:01:26 +0300] | / mysql_test.php / | NULL | debian.localdomain | 1 rows in set (0.00 sec)

Так як вибірка буде часто здійснюватися по полях «Час» та «Ім'я віддаленого хоста», то первинний ключ привласнюється даними стовпцям:

mysql> ALTER TABLE web1_access_log ADD INDEX (time_stamp, remote_host)

Отже, приведена вище конфігурація зв’язків web-сервера Nginx з Apache така, що Nginx тепер виступає як проксі-сервер, весь трафік який йде на Apache, спочатку проходить через frontend-сервер. Web-сервери Nginx і Apache встановлені на двох різних машинах. Програма класифікації запитів встановлюється на frontend-сервері і звертається до бази даних з логами Apache на другу машину. Таким чином, система буде працювати до тих пір, поки буде витримувати навантаження web-сервер Nginx, з чим він справляється при правильному налаштуванні відмінно. В даний час практично не проводиться DDoS-атак такої потужності, здатні забити пам'ять і вивести з ладу даний проксі-сервер.

3.4 Виявлення атаки на web-сервер

Виявлення атаки відбувається на основі аналізу статистичних даних, отриманих при функціонуванні web-сервера в штатному режимі.

До складу системи виявлення атаки входять наступні файли:

— netstatone.sh — програма рахує кількість запитів на 443-й порт і записує результат у файл netstat.txt;

— netstat.txt — текстовий файл, в якому містяться результати виконання програми netstatone.sh;

— mean.pl — програма рахує середнє значення чисел у файлі netstat.txt і записує результат в meanscores.txt;

— meanscores.txt — текстовий файл, в якому містяться результати виконання програми mean.pl;

— netstat.sh — програма рахує кількість запитів на 443-й порт і порівнює це значення зі середнім, якщо воно перевищує середнє в *k* раз, програма запускає daemon.pl (програма класифікації запитів).

Схема виявлення атак на web-сервер наведена на рисунку 3.3.

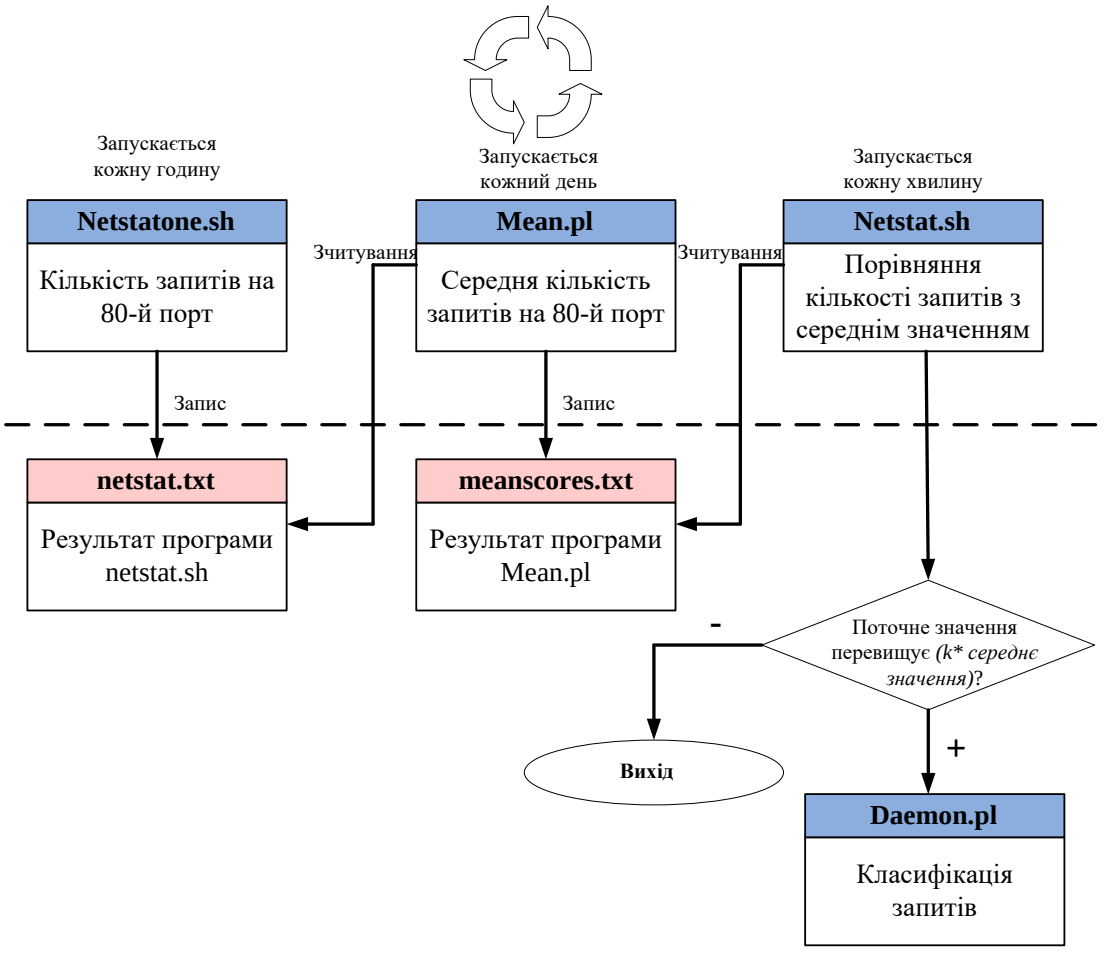

Рисунок 3.3 — Виявлення атак на web-сервер

Програми netstatone.sh і netstat.sh реалізовані на мові програмування Bash. Кількість запитів на 443-й порт рахується за допомогою команди netstat, яка показує стан усіх активних сокетів:

Netstat -na | grep ".443" | wc -l

Програма mean.pl написана на мові PERL. Програма відкриває netstat.txt і вважає середнє значення чисел у файлі, результат виконання при кожному запуску перезаписується в файл meanscores.txt. Число, яке зберігається в meanscores.txt, є головним критерієм прийняття рішення про запуск класифікатора запитів.

За допомогою системи виконання програм (crontab) відбувається регулярний запуск програм netstatone.sh, mean.pl і netstatone.sh, коли система працює в штатному режимі.

Сайти в різний час доби відвідує різна кількість клієнтів. Для більш об'єктивного підрахунку середнього значення прийнято запускати програму netstatone.sh кожні 5 годин. Mean.pl буде переглядати файл netstat.txt раз на добу для підрахунку середнього значення. А програма netstat.sh буде виконуватися кожну хвилину і при необхідності запускати daemon.pl.

```
# crontab -e
SHELL = /bin/bash
PATH = /usr/local/sbin: /usr/local/bin: /sbi: /bin: /usr/sbin: /usr/bin
HOME = /home/root1/dyplom/
MAILTO = oleksii@ht-systems.ua
#m h dom mon dow command — формат дати та часу
* * * * * $ HOME / netstat.sh
#запускати кожну хвилину
* * / 5 * * * $ HOME /netstat.sh>> $HOME /netstat.txt 2> & 1
#запускати раз в п'ять годин
02 * * * $HOME /netstat.sh>> $HOME /meanscores.txt 2> & 1
#запускати о 02 годині 00 хвилин
```

У момент атаки на web-сервер процес функціонування системи складається з далі перелічених етапів.

Клієнт ініціює запит до web-сервера. IP-адреса HTTPS-заголовка, що надходить на 443-й порт, перевіряється в правилах Netfilter, якщо така адреса існує в правилах на відкидання, то пакет видаляється фаєрволом, якщо IP-адреси немає в правилах, пакет проходить далі до web-серверу Nginx.

Nginx приймає з'єднання від клієнта і читає від нього весь запит. Якщо запитується статичний контент, Nginx сам обробляє запит і віддає запитувану сторінки клієнту.

Якщо запитується динамічна сторінка (наприклад, запит до php-скрипту), Nginx відкриває з'єднання до Apache, далі останній виконує свою роботу (генерує динамічний контент), після чого віддає свою відповідь Nginx, який його буферизує в пам'яті або тимчасовому файлі, тим часом, Apache звільняє ресурси.

Далі Nginx повільно віддає контент клієнту, витрачаючи при цьому на порядки менше ресурсів, ніж Apache.

Браузер клієнта отримує відповідь, закриває з'єднання з сервером і відображає запитувану сторінку. Nginx записує всі звернення до сайту в журнал доступу (access.log). Запущена програма класифікації запитів чекає нового запису в журнал, отримуючи запис він зрівнюється із запитами з журналу доступу web-сервера Apache (вибираються запити до початку атаки).

Якщо запит «схожий» на запити до сервера до атаки, то він пропускається, якщо «не схожий», то IP-адреса з заголовка запиту передається міжмережевому екрану на видалення. Схема функціонування системи представлена на рисунку 3.5.

Початком атаки на web-сервер вважається момент запуску програми класифікації запитів.

Ймовірність того, що з даної операційної системи (os) і типом браузера (user_agent) звертається легітимний користувач є константа і визначається користувачем в конфігураційному файлі.

Таке рішення обґрунтовується тим, що ці атрибути не залежать від кількості легітимних користувачів, що використовують операційну систему і браузер, а має

залежати виключно від надійності і наявності вразливостей в операційній системі та браузері.

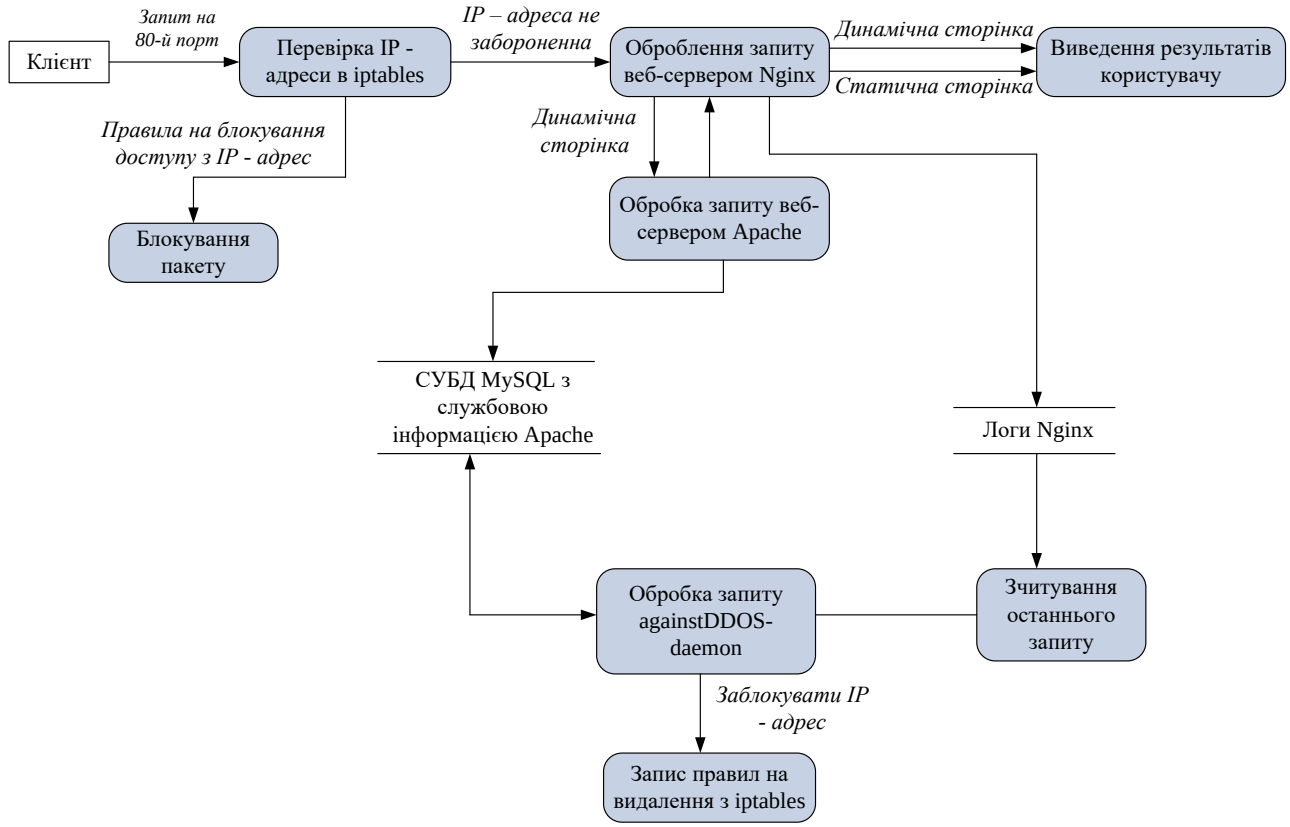

Рисунок 3.5 — Діаграма потоку даних

Програма класифікації запитів складається з 4-х файлів:

— config.pl — у ньому міститься конфігураційна інформація (сервер, логін і пароль до БД з логами Apache, шляхи до файлів системи та зовнішніх програм: iptables та ряд констант);

— daemon.pl — програма класифікації запитів;

— ipddos.txt — текстовий файл, що містить заблоковані IP-адреси;

— control.sh — програма, що перевіряє, чи запущений демон daemon.pl, якщо запущений — повторно не запускається.

Програма реалізована на мові програмування PERL, що дозволяє системі бути кросплатформенною, так як дана мова вбудована в більшість серверних систем. Крім того, PERL володіє наступними перевагами:[31]

— PERL інтерпретована мова, а отже невеликі програми виконуються дуже швидко;

— володіє багатими можливостями для роботи з текстом, який реалізований за допомогою регулярних виразів;

— простотою використання.

У програмі daemon.pl реалізований наступний алгоритм:

— очікується запис нового рядка в access.log web-сервера Nginx;

— рядок зчитується і кожному атрибуту рядка присвоюється змінна;

— перевіряється чи міститься IP-адреса у файлі ipddos.txt, якщо вона є у файлі, то запит не обробляється;

— відбувається підключення до бази даних, яка містить логи web-сервера Apache та створює вибірку тих значень, які містяться в запиті;

— проводиться підрахунок ймовірностей класифікації атрибутів запиту, як «легітимних» запитів;

— проводиться підрахунок ймовірностей класифікації атрибутів запиту, як DDoS-ботів запитів;

— проводить підрахунок повної ймовірності класифікації запиту за теоремою Баєса;

— якщо запит класифікується як DDoS-бот, то IP-адреса із заголовка запиту записується в файл ipddos.txt і передається файерволу на блокування:

/ Sbin / iptables -a INPUT -i eth1 -p tcp-dport 443 - source $ ip-j DROP ;

— повертається на обробку нового запиту (початок алгоритму).

В результаті виконання даного алгоритму при атаці типу «HTTPS-flood», досягається захист системи від перевантаження запитами завдяки поділу по Баєса на користувачів з легітимними та нелегітимними запитами, після чого небажаних запитів фільтруються.

У рамках даної роботи були розглянуті методи DDoS-атак. Були проаналізовані основні підходи до захисту від атак DDoS в цілому і від атак типу «HTTPS-flood». Зокрема, зроблено огляд існуючих комерційний рішень та рішень з відкритим вихідним кодом.

В якості практичної частини реалізована система захисту від розподілених атак на web-сервер, що відповідає всім поставленим вимогам. На підставі теоретичних досліджень розроблені оптимальні формули підрахунку ймовірностей віднесення атрибутів запитів до одного з двох класів, які застосовуються в «наївному» алгоритмі Баєса для класифікації запитів.

Варто відзначити, що на ринку вільно поширюваного програмного забезпечення немає аналогів розробленій системі. Існуючі рішення класифікують запити, ґрунтуючись на одному критерії «благонадійності» запиту, а система класифікації запитів до web-серверу, розроблена в даній роботі, аналізує запит відразу за такими атрибутами:

— IP-адреса ініціатора;
— кількість запитів з однієї IP-адреси;
— країна, з якої йде запит;
— метод запиту;
— запитувана сторінка;
— сторінка входу на сайт;
— тип браузера;
— тип операційної системи.

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАСОБУ ЗАХИСТУ WEB-СЕРВЕРА ВІД DOS-АТАК

Дана магістерська робота відноситься до прикладної науково-технічної роботи. Планується виведення науково-технічної розробки на ринок із залученням потенційним інвестора.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного та технологічного аудиту є оцінювання комерційного потенціалу методу та програмного засобу захисту web-сервера від DoS-атак.

Для проведення технологічного аудиту було залучено 3-х незалежних експертів: Кузьменко Вадим Петрович, Потапенко Андрій Іванович та Бевз Ірина Володимирівна. Для проведення технологічного аудиту використано таблицю 4.1, в якій за п’ятибальною шкалою, використовуючи 12 критеріїв, здійснено оцінку комерційного потенціалу. [32]

Таблиця 4.1 — Оцінювання комерційного потенціалу розробки

Критерії оцінювання та бали (за 5-бальною шкалою)					
Критерій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження таблиці 4.1

Критерій	0	1	2	3	4
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження таблиці 4.1

Критерій	0	1	2	3	4
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

В таблиці 4.2 наведені рівні комерційного потенціалу розробки.

Таблиця 4.2 — Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

В таблиці 4.3 наведено результати оцінювання експертами комерційного потенціалу розробки.

Середньоарифметична сума балів, розрахована на основі висновків експертів склала 33 бали, що згідно таблиці 4.2 вважається, що рівень комерційного потенціалу проведених досліджень є вище середнього.

Таблиця 4.3 — Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	Кузьменко Вадим Петрович	Потапенко Андрій Іванович	Бевз Ірина Володимирівна
	Бали, виставлені експертами:		
1	2	2	2
2	3	4	2
3	2	2	1
4	2	5	3
5	4	2	3
6	1	1	3
7	2	2	4
8	3	3	3
9	4	1	4
10	4	3	4
11	4	3	2
12	4	4	2
Сума балів	$СБ_1$=35	$СБ_2$=32	$СБ_3$=33
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_1^3 СБ_i}{3} = \frac{35+32+33}{3} = 33{,}3$		

Реалізація програмного засобу можлива на підприємствах, ресурси яких знаходяться в мережі Інтернет та які потребують захисту від DoS-атак. Дане рішення дозволяє ефективно захищати сервіси від атак ботів. За рахунок зручного масштабування, зберігається ефективність захисту при збільшенні мережі.

Порівняємо нову розробку з аналогами, які існують на ринку. В якості аналогічних засобів захисту обрано технологію Cisco Clean PIPes та Ados. Cisco Clean PIPes дуже ефективна технологія, яка реалізовується на апаратному та програмному рівні, проте дуже високої вартості. Ados реалізовується на програмному рівні, має низьку ціну, але малоефективний при сучасних DDoS-атаках.

У розробці дана проблема вирішується за допомогою розробки власного алгоритму програмних засобів. Для покращення захисту програмний засіб реалізований на двох різних серверах. Вартість їх реалізації суттєво нижча в порівнянні з технологією від Cisco, але не суттєво поступається по якості захисту.

4.2 Прогнозування витрат на виконання науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи групуються за такими статтями: витрати на оплату праці, витрати на соціальні заходи, матеріали, паливо та енергія для науково-виробничих цілей, витрати на службові відрядження, програмне забезпечення для наукових робіт, накладні витрати та інші витрати.

4.2.1 Витрати на оплату праці

Основна заробітна плата розробників, що працюють над проектом, визначена у формулі:

$$З_о = \sum_{i=1}^{k} \frac{М_{ni} \cdot t_i}{Т_р}, \quad (4.1)$$

де k — кількість посад дослідників, залучених до процесу досліджень;

$М_{ni}$ — місячний посадовий оклад конкретного дослідника, грн;

$Т_р$ — середня кількість робочих днів в місяці, Тр=21…23 дні. Обрано 22 дні;

t_i — кількість днів роботи конкретного дослідника, дні.

Для розробки програмного засобу захисту web-сервера від DoS-атак необхідно залучити програміста з посадовим окладом 10000 грн та системного адміністратора з окладом 12000 грн. Кількість робочих днів у місяці складає 22, а кількість робочих днів програміста складає 16. Сумарні розрахунки зведено до таблиця 4.4.

Таблиця 4.4 — Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	15000	682	22	15000

Продовження таблиці 4.4

Програміст	10000	455	16	7280
Системний адміністратор	12000	545	22	12000
Всього				34280

4.2.2 Відрахування на соціальні заходи

Додаткова винагорода ($З_{дод.}$) усіх розробників та працівників, які брали участь у цьому етапі роботи, обчислюється як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод.} = (З_о + З_р) \cdot \frac{Н_{дод.}}{100\%}, \quad (4.2)$$

де $Н_{дод.}$ — норма нарахування додаткової заробітної плати.

$$З_{дод} = 0{,}1 * 34280 = 3428 \text{ грн.}$$

Нарахування на заробітну плату $Н_{ЗП}$ дослідників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою (4.3):

$$Н_{ЗП} = (З_о + З_д) * \frac{\beta}{100} \text{ грн}, \quad (4.3)$$

де $З_о$ — основна заробітна плата розробників, грн.;

$З_д$ — додаткова заробітна плата всіх розробників та робітників, грн.;

β — ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % .

Дана діяльність відноситься до бюджетної сфери, тому ставка єдиного внеску на загальнообов'язкове державне соціальне страхування буде складати 22%, тоді:

$$Н_{ЗП} = (34280 + 3428) \cdot \frac{22\%}{100\%} = 8295{,}76 \text{ грн.}$$

4.2.3 Сировина та матеріали

Витрати на матеріали М та комплектуючі К, що були використані під час виконання даного етапу роботи, розраховуються по кожному виду матеріалів за формулою:

$$М = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j - \sum_{j=1}^{n} В_j \cdot Ц_{вj}, \quad (4.4)$$

де $Н_j$ — кількість матеріалу j-го виду, шт.;

n — кількість видів матеріалу;

$Ц_j$ — ціна матеріалу j-го виду, грн;

$К_j$ — коефіцієнт транспортних витрат, $К_j$ = (1,1…1,15). Обираємо $К_j$ 1,15;

$В_j$ — маса відходів j-го найменування, кг;

$Ц_{вj}$ — вартість відходів j-го найменування, грн/кг.;

Результати розрахунків занесено до таблиці 4.5.

4.2.4 Програмне забезпечення для наукових робіт

Програмне забезпечення для наукової роботи включає витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, необхідного для проведення дослідження.

$$В_{прг} = \sum_{i=1}^{k} Ц_{іпрг} \cdot С_{прг.і} \cdot К_{і,} \quad (4.5)$$

де Ціпрг — ціна придбання одиниці програмного засобу цього виду, грн;

Спрг.і — кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

Кі — коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, (Кі = 1,10…1,12);

k — кількість найменувань програмних засобів.

Таблиця 4.5 — Витрати на матеріали

Найменування матеріалу	Ціна за одиницю, грн.	Витрачено	Вартість витраченого матеріалу, грн.
Папір	130	1	150
Флеш накопичувач	220	1	200
Всього			350
З урахуванням коефіцієнта транспортування			402,5

Для створення програмного засобу використовується мова програмування Perl, програмне забезпечення якої є безкоштовним та використовується програмна оболонка Bash, яка також у безкоштовному доступі. На двох ноутбуках використовується операційна система Ubuntu, яка теж поширюється безкоштовно

Таблиця 4.6 — Витрати на придбання програмних засобів по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість
Операційна система Windows 10	1	5300	5300
Оренда сервера	2	7000	14000
Всього			19300

4.2.5 Амортизація обладнання, програмних засобів та приміщень

Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час виконання даного етапу роботи.

Дані відрахування розраховують по кожному виду обладнання, приміщенням тощо.

$$А_{обл} = \frac{Ц_б}{Т_в} \cdot \frac{t_{вик}}{12}, \quad (4.6)$$

де Цб – балансова вартість даного виду обладнання (приміщень), грн.;

Тв – час користування;

tвик – термін використання обладнання (приміщень), цілі місяці.

Таблиця 4.7 — Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, місяців.	Амортизаційні відрахування, грн
Ноутбук Alienware	27000	3	1	750
Ноутбук Asus	20000	3	1	555,55
Ноутбук Xiaomi	26000	2	1	1083,33

Продовження таблиці 4.7

Приміщення	600000	20	1	2500
Меблі	40000	10	1	333,33
Всього				5222,21

Згідно пункту 137.3.3 Податкового кодексу амортизація нараховується на основні засоби вартістю понад 20000 грн. В нашому випадку для написання магістерської роботи використовувався персональний комп'ютер вартістю 27000+20000+26000= 73000 грн.

4.2.6 Паливо та енергія для науково-виробничих цілей

До статті «Паливо та енергія для науково-виробничих цілей» відносяться витрати на всі види палива й енергії, що безпосередньо використовуються з технологічною метою на проведення досліджень.

Витрати на силову електроенергію ($В_е$) розраховують за формулою:

$$В_е = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_е \cdot К_{впі}}{\eta_i}, \qquad (4.7)$$

де W_{yi} — встановлена потужність обладнання на певному етапі розробки, кВт;

t_i — тривалість роботи обладнання на етапі дослідження, год;

$Ц_е$ — вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), $Ц_е = 6{,}2$ грн/кВт;

$К_{впі}$ — коефіцієнт, що враховує використання потужності, $К_{впі} < 1$; обрано $К_{впі} = 0{,}7$;

η_i — коефіцієнт корисної дії обладнання, $\eta_i < 1$; обрано $К_{впі} = 0{,}8$.

Проведені розрахунки зведено у таблиці 4.8.

Таблиця 4.8 — Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Ноутбук Alienware	0,24	176	229,15
Ноутбук Asus	0,16	176	152,77
Ноутбук Xiaomi	0,2	128	138,88
Всього			520,8

4.2.7 Службові відрядження

Витрати на службові відрядження, витрати на роботи, які виконують сторонні підприємства, установи, організації та інші витрати в нашому досліджені не враховуються оскільки їх не було.

4.2.8 Накладні (загальновиробничі) витрати

Накладні (загальновиробничі) витрати $В_{нзв}$охоплюють: витрати на управління організацією, оплата службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо. Накладні (загальновиробничі) витрати $В_{нзв}$можна прийняти як (100…150)% від суми основної заробітної плати розробників та робітників, які виконували дану МКНР, тобто:

$$В_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.8)$$

де $Н_{нзв}$ – норма нарахування за статтею «Інші витрати».

$$В_{нзв} = 34280 \cdot \frac{50}{100\%} = 17140 грн.$$

Сума всіх попередніх статей витрат дає витрати, які безпосередньо стосуються даного розділу МКНР:

$$В = 34280 + 3428 + 8295{,}76 + 402{,}5 + 19300 + 5222{,}21 + 520{,}8 +$$

$+17140\text{грн} = 88589{,}27\text{грн}.$

Прогнозування загальних втрат ЗВ на виконання та впровадження результатів виконаної МКНР здійснюється за формулою:

$$ЗВ = \frac{В_{заг}}{\eta}, \qquad (4.9)$$

де η — коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи. Під час він дослідження дорівнює 0,5.

$$ЗВ = \frac{88589{,}27}{0{,}5} = 177178{,}54 \text{ грн}$$

4.3 Розрахунок економічної ефективності науково-технічної розробки

При розробці програмного засобу можливе збільшення чистого прибутку для потенційного інвестора $\Delta П_i$ у кожному із років, на протязі яких очікується отримання результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховується за формулою:

$$\Delta П_i = (\pm \Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \qquad (4.10)$$

де $\pm\Delta Ц_o$— зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році;

N — основний кількісний показник, який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної розробки;

$Ц_o$ — основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році, $Ц_o = Ц_б \pm \Delta Ц_o$;

$Ц_б$ — основний якісний показник, який визначає ціну реалізації існуючої (базової) науково-технічної розробки у році до впровадження результатів;

ΔN — зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році;

λ — коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2021 році ставка податку на додану вартість становить 20%, а коефіцієнт λ=0,8333;

ρ — коефіцієнт, який враховує рентабельність інноваційного продукту (послуги);

ϑ — ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2021 році ϑ=18%.

При впровадженні результатів наукової розробки покращується якість програмного засобу для захисту мережі від DoS-атак. Кількість одиниць реалізованої продукції також збільшується: протягом першого року на 200 шт., протягом другого року – на 300 шт., протягом третього року на 400 шт. Реалізація продукції до впровадження розробки складала 0 шт. Базова ціна складає 5000 грн. Збільшення ціни за рахунок підвищення якості складає 1000 грн. Розрахуємо прибуток, яке отримає підприємство протягом трьох років.

Прогнозується щорічний приріст прибутку компанії при впровадження результатів науково-дослідницьких розробок. Збільшення чистого прибутку підприємства $\Delta П_i$ за перший рік складе:

$$\Delta П_1 = [1000 \cdot 0 + (5000 + 1000) \cdot 200] \cdot 0,8333 \cdot 0,26 \cdot \left(1 - \frac{18\%}{100\%}\right) = 213191,47 \text{ грн.}$$

Збільшення чистого прибутку компанії $\Delta П_i$ на другий рік (порівняно з базовим, тобто роком, що передує впровадженню результатів наукових досліджень) складе:

$$\Delta П_2 = [1000 \cdot 0 + (5000 + 1000) \cdot (200 + 300)] \cdot 0,8333 \cdot 0,26 \cdot \left(1 - \frac{18\%}{100\%}\right) = 532978,68 \text{ грн.}$$

Збільшення чистого прибутку підприємства $\Delta П_i$ на третій рік складе:

$$\Delta П_3 = [1000 \cdot 0 + (5000 + 1000) \cdot (200 + 300 + 400)] \cdot 0,8333 \cdot 0,26 \cdot \left(1 - \frac{18\%}{100\%}\right) = 959361,62 \text{ грн.}$$

Далі розраховується вартість збільшення усього чистого прибутку ПП, який може отримати потенційний інвестор при впровадженні та комерціалізації науково-дослідницької розробки:

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \qquad (4.11)$$

де $\Delta П_i$— збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-дослідницької розробки, грн;

T — період часу, протягом якого очікується отримання позитивних ре-зультатів від впровадження та комерціалізації науково-дослідницької розробки, роки;

τ — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ = 0,05…0,15. Обирається τ 0,1;

t — період часу (в роках) від початку впровадження науково-технічної розробки до отримання інвестором додаткових чистих прибутків в поточному році.

$$ПП = \frac{213191,47}{(1+0,1)^1} + \frac{532978,68}{(1+0,1)^2} + \frac{959361,62}{(1+0,1)^3} = 193810,43 + 440478,25 + 720782,59 = 1355071,27 \text{ грн.}$$

Далі розраховується величина початкових інвестицій PV, які потенційний інвестор вкладає для введення та комерціалізації науково-дослідницької розробки. Для цього використовується формула:

$$PV = k_{інв} \cdot ЗВ, \quad (4.12)$$

де $k_{інв}$— коефіцієнт, що враховує витрати інвестора на впровадження нако-во-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{інв}$=2...5, але може бути і більшим. Обираємо даний коефіцієнт 2;

ЗВ — загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 \cdot 177178{,}54 = 354357{,}08 \text{ грн}$$

Тоді абсолютний економічний ефект $E_{абс}$ або чистий приведений дохід потенційного інвестора при впровадженні та комерціалізації науково-дослідницької розробки становитиме:

$$E_{абс} = ПП - PV, \quad (4.13)$$

де ПП — приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн;

PV — теперішня вартість початкових інвестицій, грн.

$$E_{абс} = 1355071{,}27 - 354357{,}08 = 1000714{,}19 \text{ (грн.)}$$

Внутрішня економічна дохідність інвестицій $E_в$, які може вкласти потенційний інвестор у встановлення та комерціалізацію науково-дослідницької розробки, розраховується за формулою:

$$E_в = \sqrt[Т_ж]{1 + \frac{E_{абс}}{PV}} - 1, \qquad (4.14)$$

де $E_{абс}$ — абсолютний економічний ефект вкладених інвестицій, грн;

PV — теперішня вартість початкових інвестицій, грн;

$Т_ж$ — життєвий цикл науково-дослідницької розробки, тобто час з початку розробки до закінчення отримання позитивних результатів від її впровадження, роки.

$$E_в = \sqrt[з]{1 + \frac{1000714,19}{354357,08}} - 1 = 0,56$$

Далі розраховується бар'єрна ставка дисконтування τ_{min}, тобто мінімальна внутрішня економічна дохідність інвестицій, нижче якої кошти у впровадження науково-дослідницької розробки та її комерціалізацію не будуть вкладати.

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$ визначається за формулою:

$$\tau_{мін} = d + f, \qquad (4.15)$$

де d — середньозважена ставка за депозитними операціями в комерційних банках; в 2021 році в Україні d=0,09...0,12. Обирається $d = 0,1$;

f — показник, що характеризує ризикованість вкладення інвестицій; за-звичай величина f=0,05...0,5, але може бути і значно вищою. Обирається 0,25.

$$\tau_{мін} = d + f = 0,1 + 0,25 = 0,35$$

Величина Ев>$\tau_{мін}$. Отже інвестор може бути зацікавлений у фінансуванні цього дослідження.

Далі розраховується період окупності інвестицій Ток, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в}, \qquad (4.16)$$

де Ев — внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = \frac{1}{0{,}56} = 1{,}79 \text{ роки}$$

В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний засіб за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 1,79 роки.

ВИСНОВКИ

В даній магістерській кваліфікаційній роботі проаналізовано сучасні різновиди DDoS атак і показано, що атаки типу Flood є найпоширенішими і найнебезпечнішими.

Проведено аналітичний огляд апаратних рішень протидії DDoS атакам, зокрема комерційного рішення від Cisco для захисту від DDoS-атак. Показано, що це рішення універсальне та гарантовано фільтрує всі види DDoS-атак, але має високу вартість.

Проаналізовано сучасні програмні способи захисту, але всі вони не універсальні і не вирішуються відразу всіх завдань при відображенні потужної DDoS-атаки, а саме: збереження працездатності сервісу, відкидання максимальної кількість ботів і мінімуму легітимних користувачів, недопускання блокування адрес пошукових систем;

Запропоновано структуру захисту на основі web-серверів Nginx та Apache так, що Nginx тепер виступає як проксі-сервер, весь трафік, який йде на Apache проходить через frontend-сервер. Web-сервери Nginx та Apache встановлені на двох різних машинах.

Запропоновано здійснювати класифікацію трафіку по основних показниках з використанням критерію Баєса, що дозволило підвищити ефективність фільтрації хибних запитів, зберігаючи при цьому високий рівень доступності системи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Орленко В. С. «Методи оцінки та підвищення захищеності інформаційних ресурсів систем спеціального призначення» Обл.№ 1195дск, 24.03.2009 Автореферат на здобуття наукового ступення кандидата технічних наук, Державний університет інформаційно-комунікаційних технологій, Київ-2009

2. Савчук О.М. використання баєсівського класифікатора для ідентифікації DDos-атак в комп’ютерних мережах. / О.М. Савчук, С.М. Захарченко. // Тези доповіді. LII Науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії (2023): веб-сайт. URL: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2023/paper/view/17440 (дата звернення: 02.06.2023)

3. Типова структура корпоративного сайту: веб-сайт. URL: https://www.centum-d.com/tipova-struktura-korporativnogo-sajtu/ (дата звернення: 02.06.2023)

4. Моделі і методи проектування інформаційних систем: веб-сайт. URL: https://elearning.sumdu.edu.ua/free_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20160217112601/170352/index.html (дата звернення: 02.06.2023)

5. D.E. Bell Secure Computer Systems: Mathematical foundations and model / D.E. Bell, L.J. La Padula. – Report ESD – TR – 73 – 278, Mitre Corp., Bedford, Mass, Nov. 1973

6. Комплексні системи захисту інформації: проектування: веб-сайт. URL: www.academia.edu/40525032/Комплексні_системи_захисту_інформації_проектування_впровадження_супровід (дата звернення: 02.06.2023)

7. Що таке комплексна система захисту інформації: веб-сайт. URL: http://altersign.com.ua/korysna-informacija/pobudova-kszi/shcho-take-kompleksna-systema-zahystu-informaciji-kszi (дата звернення: 02.06.2023)

8. Політика безпеки інформації в захищених автоматизованих системах: веб-сайт. URL: https://ekmair.ukma.edu.ua/server/api/core/bitstreams/50763dd1-6bf8-4c7d-96a2-947a38487b09/content (дата звернення: 02.06.2023)

9. НД ТЗІ 2.5-010-03 Вимоги до захисту інформації web-сторінки від несанкціонованого доступу

10. Вимоги до реалізації функціональних послуг безпеки інформації: веб-сайт. URL: https://studfile.net/preview/16436163/page:7/ (дата звернення: 02.06.2023)

11. Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу: веб-сайт. URL: tzi.ua/assets/files/НД-ТЗІ-2.5-005--99.pdf (дата звернення: 02.06.2023)

12. Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу: веб-сайт. URL: tzi.ua/assets/files/НД-ТЗІ-2.5-004-99.pdf (дата звернення: 02.06.2023)

13. Reddy Y. The DARPA Initiative in Concurent Engineering, Concurent Engineering Res earch in Review, Reddy Y., Wood R., Cleetus Y. vol. 1, 2007.

14. Що таке DoS-атака: веб-сайт. URL: https://pingvin.pro/gadgets/article-gadget/shho-take-dos-ataka.html (дата звернення: 02.06.2023)

15. Як працює DDos-атака та як від неї захиститися? Погляд зсередини: веб-сайт. URL: https://www.imena.ua/blog/how-works-ddos/ (дата звернення: 02.06.2023)

16. Найбільші DDoS-мережі: веб-сайт. URL: https://studwood.net/1654203/informatika/naybilshi_ddos_merezhi (дата звернення: 02.06.2023)

17. 10 найрезонансніших DDoS-атак: веб-сайт. URL: https://news.dtkt.ua/society/community/16491-ddostukatisia-do-nebes-10-nairezonansnisix-ddos-atak (дата звернення: 02.06.2023)

18. Поняття DDoS-атак та їх класифікація: веб-сайт. URL: https://dspace.univd.edu.ua/items/db0873d5-c349-4778-b38d-f3a8f6cad272 (дата звернення: 02.06.2023)

19. Кобозева А.А. Аналіз захищеності інформаційних систем / Кобозева А.А., Мачалін І.О., Хорошко В.О. – Київ. Вид. ДУІКТ. – 2010. – 316 с.

20. DDoS атаки і захист від них: веб-сайт. URL: https://www.ukraine.com.ua/uk/blog/hosting_ukraine/ddos-ataki-i-zashchita-ot-nih.html (дата звернення: 02.06.2023)

21. Використання ACL API для створення списків контролю доступами: веб-сайт. URL: https://internetdevels.ua/blog/using-acl-api-to-create-access-control-lists (дата звернення: 02.06.2023)

22. Системи захисту від DDos атак: веб-сайт. URL: https://omnilink.ua/sistemi-zahistu-vid-ddos-atak/ (дата звернення: 02.06.2023)

23. Найкращі методи запобігання та захисту від DDos-атак: веб-сайт. URL: https://iitd.com.ua/news/najkrashhi-metodi-zapobigannja-ta-zahistu-vid-ddos-atak/ (дата звернення: 02.06.2023)

24. DDoS Protection Solution Enabling “Clean Pipes”: веб-сайт. URL: www.cisco.com/c/dam/assets/cdc_content_elements/networking_solutions/service_provider/ddos_protection_sol/ddos_protection.pdf (дата звернення: 02.06.2023)

25. DDoS-атаки й методи боротьби з ними: веб-сайт. URL: http://elar.khmnu.edu.ua/jspui/bitstream/123456789/1449/1/Chern09.pdf (дата звернення: 02.06.2023)

26. А. В. Жилін, О. М. Шаповал, О. А. Успенський технології захисту інформації в інформаційно-телекомунікаційних системах, навчальний посібник: веб-сайт. URL: https://ela.kpi.ua/bitstream/123456789/45723/1/NP_TZI_ITS.pdf (дата звернення: 02.06.2023)

27. Математична модель оцінювання захисту web-сайтів: веб-сайт. URL: https://openarchive.nure.ua/server/api/core/bitstreams/12dbbaca-2a7a-420e-83e1-36f6dd0957bc/content (дата звернення: 02.06.2023)

28. Ю. В. Коваль, А. Б. Ставровський, Інформаційні мережі, навчальний посібник: веб-сайт. URL: http://csc.knu.ua/media/filer_public/b7/26/b7263e88-6e95-44a9-9b50-ae36400badcc/infonets-006.pdf (дата звернення: 02.06.2023)

29. Rachel McCollin, DDoS Attacks Explained: Causes, Effects, and How to Protect Your Site: веб-сайт. URL: https://kinsta.com/blog/what-is-a-ddos-attack/ (дата звернення: 02.06.2023)

30. User Agents: веб-сайт. URL: https://user-agents.net/ (дата звернення: 02.06.2023)

31. Perl Language Server: веб-сайт. URL: https://github.com/FractalBoy/perl-language-server (дата звернення: 02.06.2023)

32. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2023. – 42 с.

ДОДАТОК А

Міністерство освіти та науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп’ютерної інженерії

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ ВНТУ

д.т.н., проф.

_____Азаров О.Д.

“__” _____ 2023 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської дипломної роботи

«Програмний засіб захисту web-сервера від DoS-атак»

08-23.МКР.003.00.000 ТЗ

Науковий керівник к.т.н., професор

_____ _____ Азарова А.О.

Студент групи КІ-21мз

_____ Савчук О.М.

Вінниця 2023

1 Найменування та область застосування

Робоча назва проекту «Програмний засіб захисту web-сервера від DoS-атак», розробляється для захисту мережевих ресурсів від DoS-атак.

2 Мета та призначення розробки

Експлуатаційне призначення розробки — для захисту мережевих ресурсів від DoS-атак.

3 Основи для розробки

Основою для розробки є дисципліни Комп'ютерні мережі, Основи мультимедіа та безпеки в комп`ютерних мережах, Корпоративні і загальнодоступні мережі, Програмування.

4 Етапи МКР та очікувані результати

Робота виконується в шість етапів, що наведені в таблиці А.1.

Таблиця А.1 — Етапи виконання роботи

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз завдання. Вступ	25.03.23	26.03.23	Вступ
2	Аналіз об'єкта захисту та обґрунтування вимог до системи захисту	27.03.23	09.04.23	Розділ 1
3	Класифікація та огляд різновидів DoS-атак	10.04.23	19.04.23	Розділ 2
4	Порівняння існуючих методів захисту мережевих ресурсів	19.04.23	27.04.23	Розділ 2, Розділ 3
5	проектування структури системи захисту від DDos-атак	28.04.23	12.05.23	Розділ 3
6	Підготовка матеріалів та опис розробки, Оформлення пояснювальної записки та ілюстративного матеріалу	13.05.23	31.05.23	Пояснювальна записка
7	Аналіз виконання роботи, висновки, додатки	01.06.23	06.06.23	Висновки, додатки, презентація

5 Матеріали, що подаються до захисту МКР

Пояснювальна записка МКР, графічні та ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відгук наукового керівника, рецензія опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, довідка про відповідність оформлення МКР діючим вимогам.

6 Порядок контролю виконання та захисту МКР

Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора.

7 Вимоги до оформлення МКР

7.1 При оформлюванні МКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— ГОСТ 2.104-2006 «єдина система конструкторської документації. Основні написи»;

— методичні вказівки до виконання магістерських кваліфікаційних робіт зі спеціальності 123 — «Комп’ютерна інженерія»;

— документи на які посилаються у вище вказаних.

8.2 Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21»

ДОДАТОК Б

Архітектура DDoS мережі

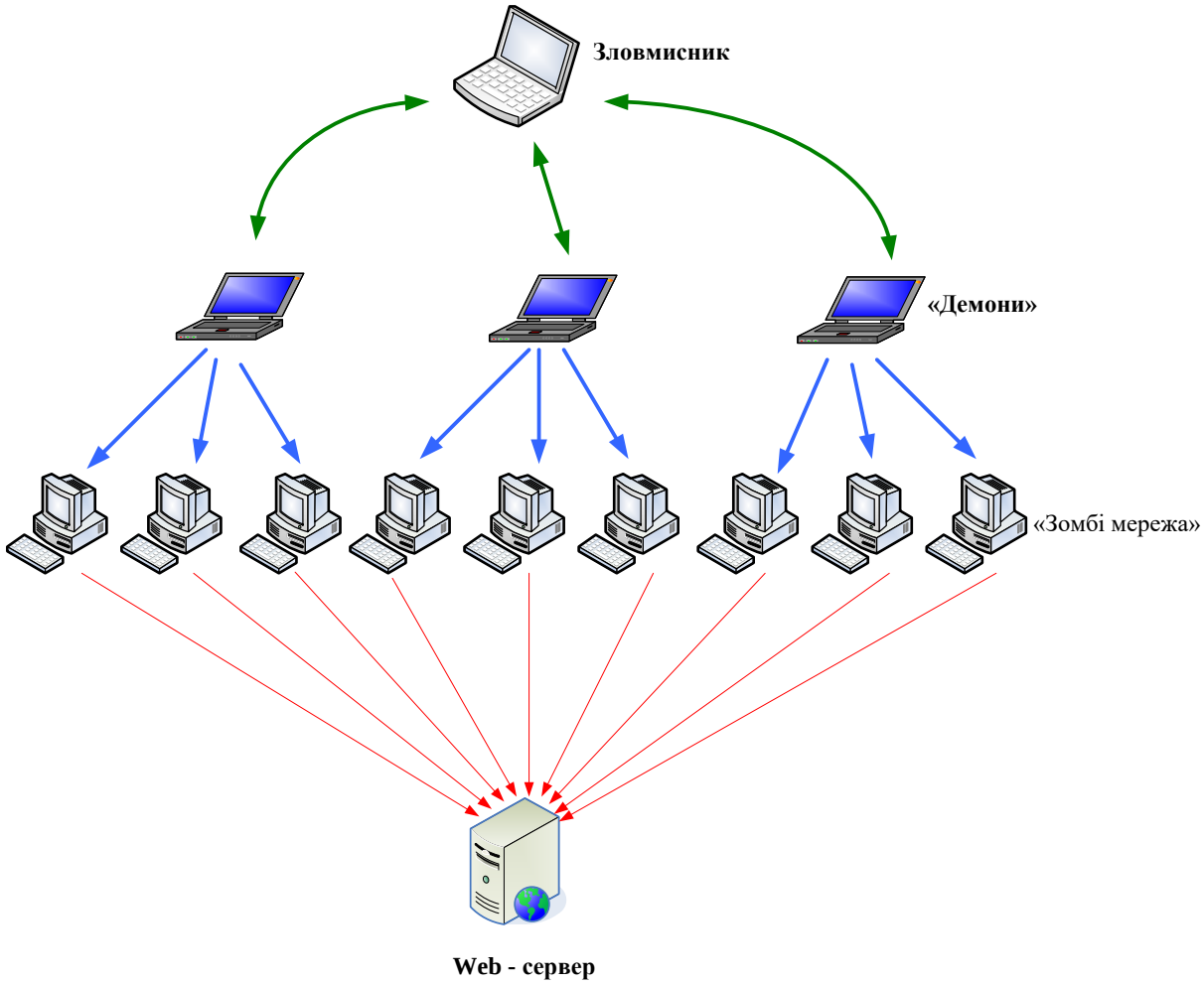

Рисунок Б.1 — Зображення архітектури DDoS мережі

ДОДАТОК В

Схема протидії DDoS атакам

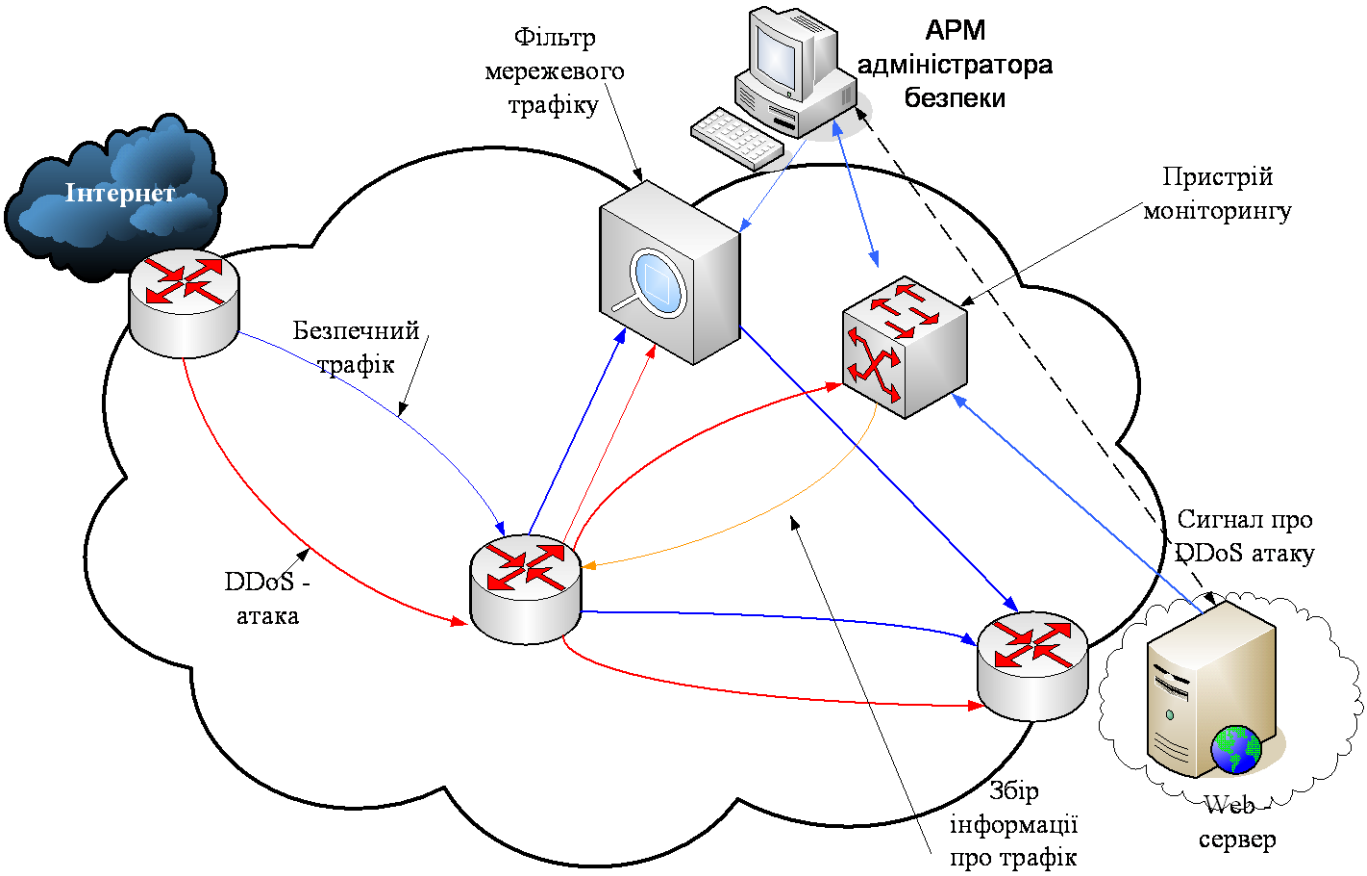

Рисунок В.1 — Схема протидії DDoS атакам

ДОДАТОК Г

Діаграма web-сервера Against DDoS-демоном

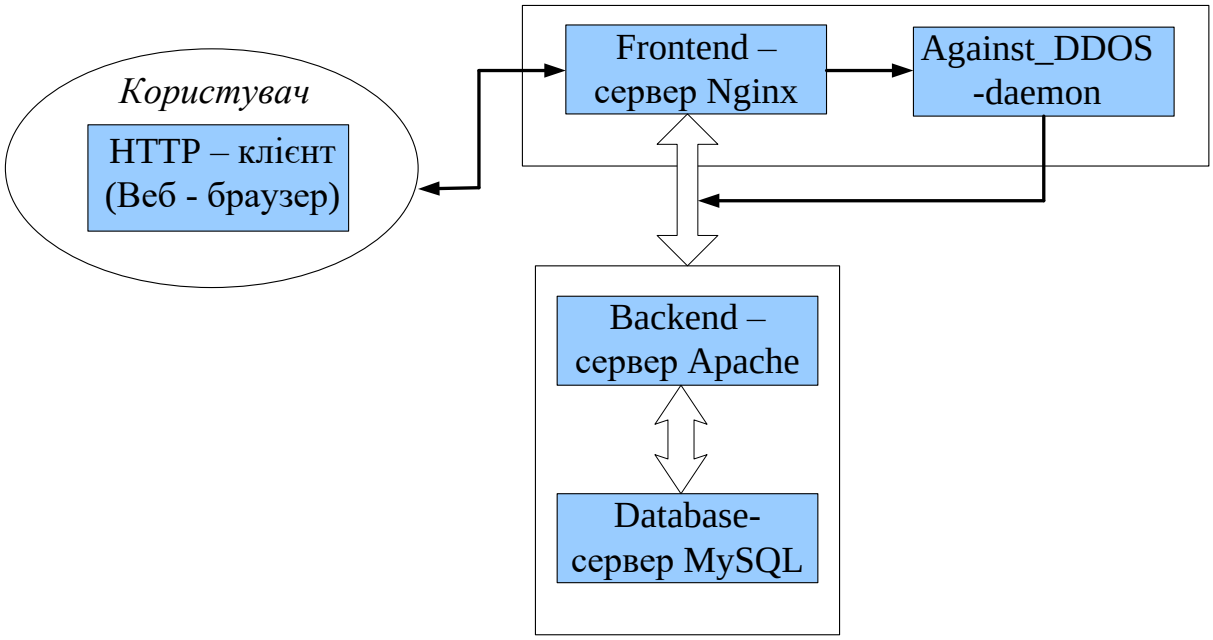

Рисунок Г.1 — Діаграма конфігурації веб сервера Against DDoS-демоном

ДОДАТОК Д

Виявлення атак на web-сервер

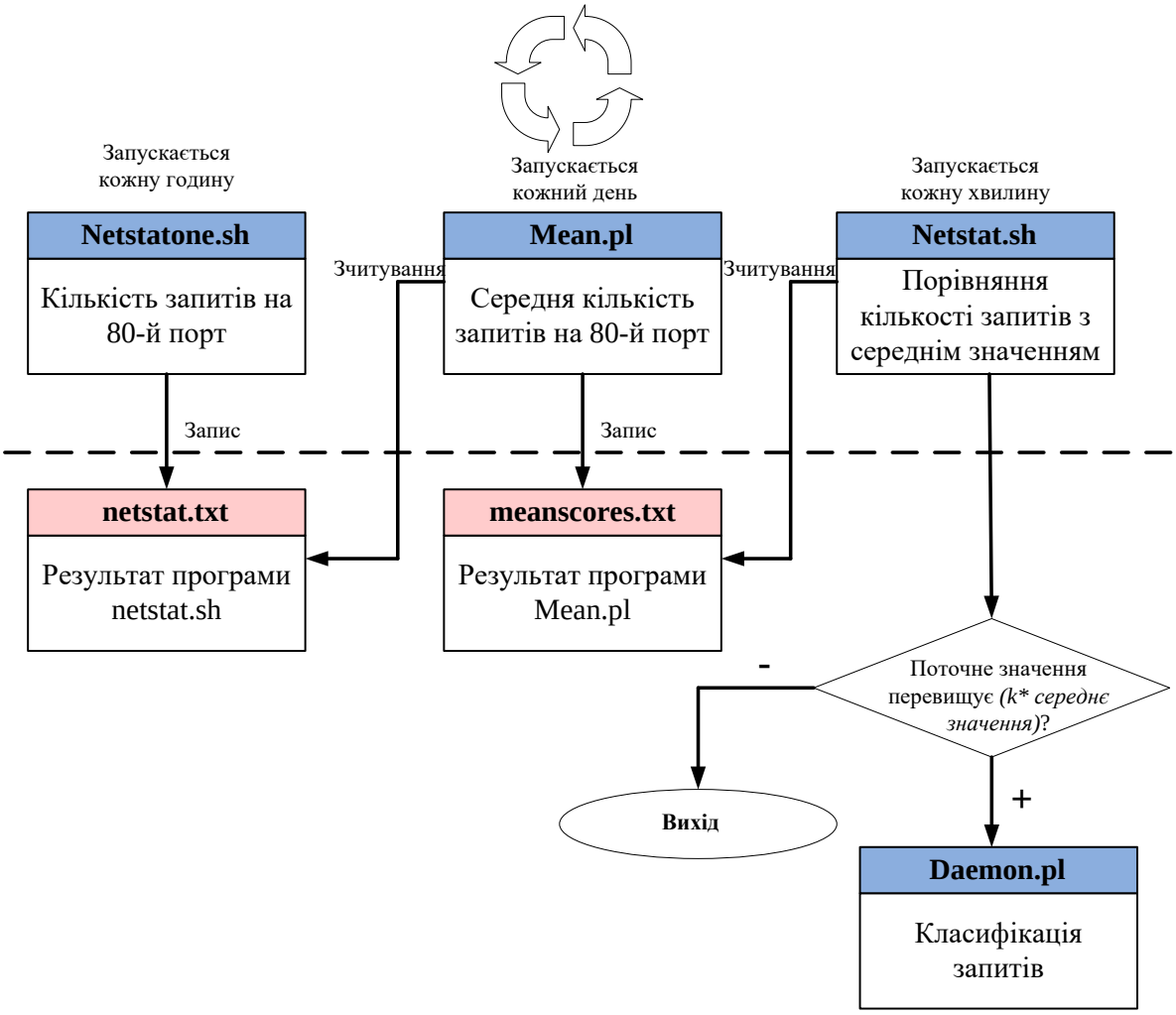

Рисунок Д.1 — Схема виявлення атак на web-сервер

ДОДАТОК Е

Діаграма потоку даних

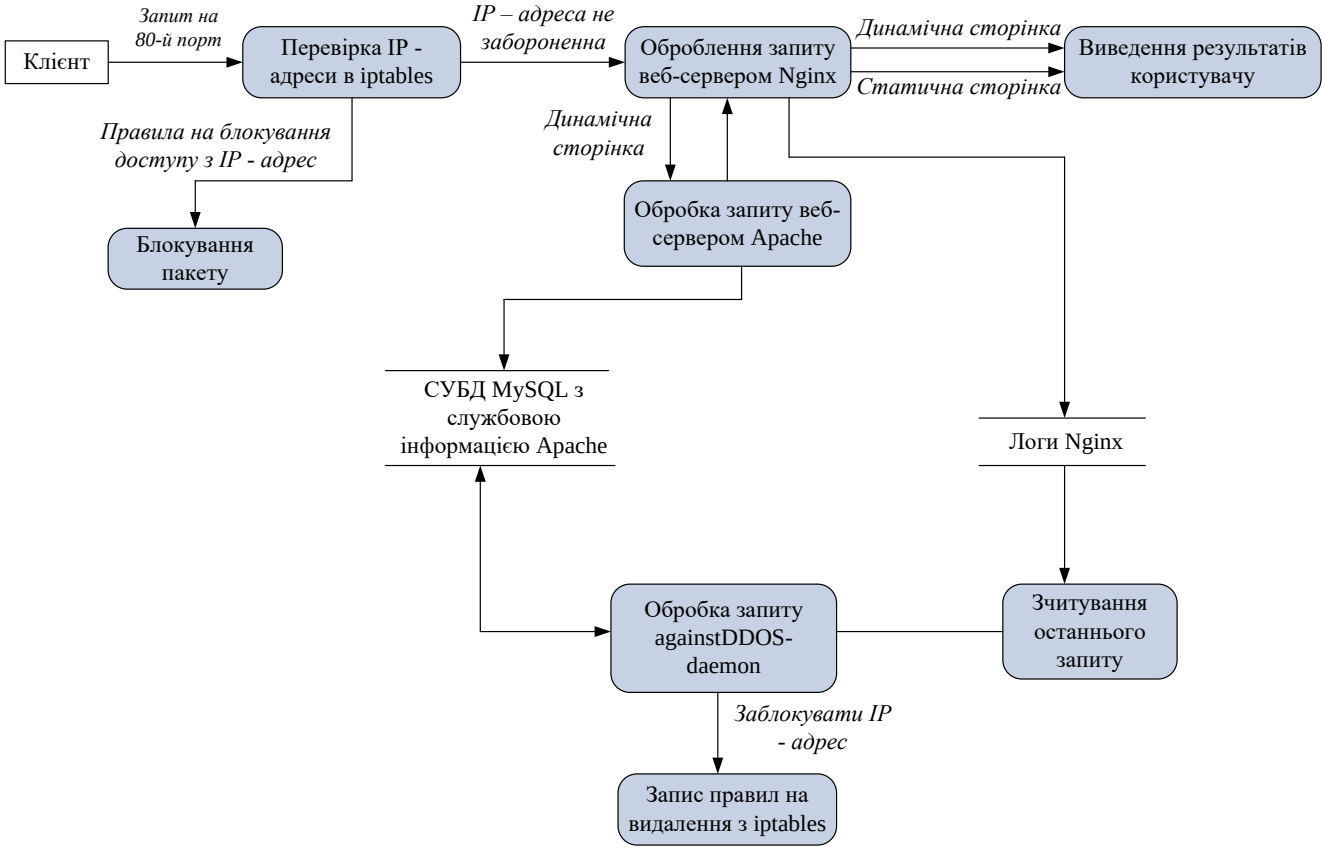

Рисунок Е.1 — Діаграма потоку даних у програмному засобі

ДОДАТОК Ж

Фрагмент лістингу комп’ютерної програми

```
error_log / var / log / nginx / error.log;
location = / {
limit_conn one 3;
root / var / www / domain.ua;
}

{
limit_except GET POST
{
deny all;
}
Set $ ban "";
If ($ HTTPS_referer= "") {set $ ban $ ban $ add;}
If ($ request_method = POST) {set $ ban $ ban $ add;
}
If ($ query_string = "action = login") {set $ ban $ ban $ add;}
If $ an  111)
{
access_log/var/log/nginx/ban IP;
return 404;
}
proxy_pass HTTPS: //127.0.0.1:8080;
}

server {
  listen 443;
  server_name domain.ua;
  access_log / var / logs / nginx-access.log;
  location /
  {
    proxy_pass HTTPS://192.168.0.1:442 /;
    proxy_redirect off;
```

```
    proxy_set_header Host $ host;
    proxy_set_header X-Real-IP $ remote_addr;
    proxy_set_header X-Forwarded-For
    $ proxy_add_x_forwarded_for;
            client_max_body_size 10m;
            client_body_buffer_size 128k;
            proxy_connect_timeout 90;
            proxy_send_timeout 90;
            proxy_read_timeout 90;
            proxy_buffer_size 4k;
            proxy_buffers 4 32k;
            proxy_busy_buffers_size 64k;
            proxy_temp_file_write_size 64k;
  }
}

NameVirtualHost *: 442
Listen 192.168.0.1:442
LogSQLLoginInfo mysql: / / loguser: loguser_rootroot @ localhost / apachelogs
LogSQLCreateTables on
LogSQLDBParam socketfile / var / run / mysqld / mysqld.sock
LogSQLTransferLogFormat AbHhmRSsTUuvI
<VirtualHost *:442>
      ServerAdmin webmaster @ localhost
      DocumentRoot / var / www /
      <Directory /var/www/>
            Options Indexes MultiViews
            AllowOverride None
            Order allow, deny
            allow from all
      </ Directory>
LogSQLTransferLogTable web1_access_log
</ VirtualHost>
LogSQLLoginInfo mysql: //loguser: loguser_rootroot @ localhost / apachelogs
LogSQLCreateTables on
```

LogSQLDBParam socketfile / var / run / mysqld / mysqld.sock

LogSQLTransferLogFormat AHhmRprSstUuv

LogSQLTransferLogTable web1_access_log

mysql> SELECT * FROM web1_access_log WHERE time_stamp = 1265021581;

agent | request_protocol | remote_host | request_method | child_pid | server_port | referrer | request_line | time_stamp | status | request_time | request_uri | remote_user | virtual_host

Mozilla/5.0 (Windows; U; Windows NT 5.1) | HTTPS/1.0 | 192.168.1.15 | GET | 51 | NULL | - | NULL | 1265021581 | 200 | [18/Mar/2023: 13:53:00 +0200] | / mysql_test.php / | NULL | debian.localdomain | 1 rows in set (0.00 sec)

mysql> ALTER TABLE web1_access_log ADD INDEX (time_stamp, remote_host)

netstat-na | grep ".443" | wc-l

SHELL = / bin / bash

PATH =/usr/local/sbin: /usr/local/bin: /sbi: /bin: /usr/sbin:
/usr/bin

HOME = /home/root1/diplom /

MAILTO = alina@ht-systems.UA

* * / 5 * * * $ HOME /netstat.sh>> $HOME /netstat.txt 2> & 1

02 * * * $HOME /netstat.sh>> $HOME /meanscores.txt 2> & 1

/ Sbin / iptables-A INPUT-i eth1-p tcp-dport 443 - source $ ip-j DROP

ДОДАТОК И

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Програмний засіб захисту web-сервера від DoS-атак

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра обчислювальної техніки
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 90,6% Схожість 9,4%

Аналіз звіту подібності (відмітити потрібне):

- ✓ Запозичення, виявлення у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Захарченко С.М.
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ Савчук О.М.
(підпис) (прізвище, ініціали)

Керівник роботи _____ Азарова А.О.
(підпис) (прізвище, ініціали)