КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА
ФАКУЛЬТЕТ РАДІОФІЗИКИ, ЕЛЕКТРОНІКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ
Кафедра радіотехніки та радіоелектронних систем

«На правах рукопису»

Робота допущена до захисту в ЕК
рішенням кафедри радіотехніки та радіоелектронних систем
від _____ 2026 року, протокол № _____.
В. о. завідувача кафедри, канд. фіз.-мат. наук, доцент
_____Ігор БЕХ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА
на тему:
«РОЗРОБЛЕННЯ ТА ДОСЛІДЖЕННЯ МАТЕМАТИЧНОЇ МОДЕЛІ ВИЯВЛЕННЯ DDoS-АТАК У ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМАХ І МЕРЕЖАХ»

Виконав:
студент 2-го курсу магістратури
денної форми здобуття освіти
спеціальності 172 «Електронні комунікації та радіотехніка»
освітньої програми «Інформаційна безпека
телекомунікаційних систем і мереж»
Казаченко Владислав Ігорович _____

Науковий керівник:
Д-р фіз.-мат. наук, професор
Дехтярук Леонід Васильович _____

Рецензент:
Лоза В. М. _____

Засвідчую, що у цій кваліфікаційній роботі
немає запозичень з праць інших авторів без
відповідних посилань
Студент _____Владислав КАЗАЧЕНКО

Київ – 2026

АНОТАЦІЯ

Цей магістерський проект розглядає проблему виявлення DDoS-атак у телекомунікаційних системах та мережах. Проведено аналіз сучасних підходів до організації мережевої інфраструктури та класифікації розподілених атак типу «відмова в обслуговуванні». У дослідженні розглянуто існуючі методи виявлення DDoS-атак, включаючи статистичні, ентропійні та машинне навчання.

На основі теорії черг розроблено математичну модель мережевого трафіку, яка дозволяє виявляти аномальні стани системи. Запропоновано інтегральний коефіцієнт аномалії, що враховує інтенсивність трафіку, кількість джерел, частку SYN-пакетів та ентропію.

Реалізовано програмний інструмент моделювання для дослідження поведінки мережі за різних умов експлуатації. Експериментальне моделювання підтверджує ефективність запропонованої моделі для виявлення DDoS-атак та аномальних навантажень.

Ключові слова: DDoS-атака, мережевий трафік, математична модель, теорія черг, ентропія, інформаційна безпека, телекомунікаційні систем.

ABSTRACT

This master's thesis project addresses the problem of detecting DDoS attacks in telecommunication systems and networks. An analysis of modern approaches to network infrastructure organization and the classification of distributed denial-of-service attacks is conducted. The study examines existing methods for detecting DDoS attacks, including statistical, entropy-based, and machine learning approaches.

Based on queueing theory, a mathematical model of network traffic is developed, which enables the detection of abnormal system states. An integral anomaly coefficient is proposed, taking into account traffic intensity, the number of sources, the proportion of SYN packets, and entropy.

A software simulation tool is implemented to study network behavior under different operating conditions. Experimental modeling confirms the effectiveness of the proposed model for detecting DDoS attacks and abnormal loads.

Keywords: DDoS attack, network traffic, mathematical model, queueing theory, entropy, information security, telecommunication systems.

СПИСОК ВИКОРИСТАНИХ СКОРЕЧЕНЬ

ACL - Access Control List (список контролю доступу);

BGP - Border Gateway Protocol (протокол граничного шлюзу);

DDoS - Distributed Denial of Service (розподілена атака відмови в обслуговуванні);

DNS - Domain Name System (система доменних імен);

DoS - Denial of Service (атака відмови в обслуговуванні);

HTTP - HyperText Transfer Protocol (протокол передачі гіпертексту);

HTTPS - HyperText Transfer Protocol Secure (захищений протокол передачі гіпертексту);

IDS - Intrusion Detection System (система виявлення вторгнень);

IoT - Internet of Things (Інтернет речей);

IP - Internet Protocol (інтернет-протокол);

IPFIX - Internet Protocol Flow Information Export (експорт інформації про потоки інтернет-протоколу);

IPS - Intrusion Prevention System (система запобігання вторгненням);

NetFlow - протокол збору статистики мережевого трафіку;

NFV - Network Functions Virtualization (віртуалізація мережевих функцій);

NTP - Network Time Protocol (протокол мережевого часу);

OSPF - Open Shortest Path First (протокол найкоротшого шляху спочатку);

OSI - Open Systems Interconnection (модель взаємодії відкритих систем);

QoS - Quality of Service (якість обслуговування);

SDN - Software-Defined Networking (програмно-керовані мережі);

TCP - Transmission Control Protocol (протокол керування передачею);

UDP - User Datagram Protocol (протокол користувацьких датаграм).

ЗМІСТ

ВСТУП

Сучасний розвиток інформаційного суспільства та стрімке зростання обсягів цифрових комунікацій зумовлюють підвищення вимог до надійності та захищеності телекомунікаційних систем і мереж. Розподілені атаки на відмову в обслуговуванні (Distributed Denial of Service, DDoS) є однією з найбільш руйнівних загроз для сучасної мережевої інфраструктури, здатних паралізувати роботу критично важливих сервісів та завдати значних фінансових і репутаційних збитків організаціям будь-якого масштабу.

Актуальність дослідження обумовлена кількома ключовими факторами. По-перше, глобальна частота DDoS-атак невпинно зростає: за даними провідних компаній у сфері кібербезпеки, лише протягом останніх років кількість зафіксованих інцидентів збільшилась на десятки відсотків щорічно. По-друге, сучасні атаки стають дедалі більш складними, багатовекторними та важко-виявлюваними завдяки застосуванню технік маскування під легітимний трафік. По-третє, традиційні сигнатурні методи захисту виявляються неефективними проти нових та модифікованих векторів атак, що вимагає розробки принципово нових математичних моделей виявлення загроз. Особливої гостроти проблема набуває в контексті масового впровадження технологій IoT (Internet of Things), хмарних обчислень та мереж п’ятого покоління (5G), які суттєво розширюють поверхню атаки та надають зловмисникам нові ресурси для організації масштабних DDoS-кампаній через формування ботнет-мереж з мільйонів підключених пристроїв. Мережевий хробак Mirai та його похідні наочно продемонстрували катастрофічний потенціал IoT-ботнетів, здатних генерувати трафік на рівні сотень гігабіт на секунду.

Метою кваліфікаційної роботи магістра є розробка та дослідження математичної моделі виявлення DDoS-атак у телекомунікаційних системах і мережах, що забезпечує ефективне та своєчасне виявлення аномалій мережевого трафіку з мінімальним рівнем хибнопозитивних спрацювань.

Об’єктом дослідження є процеси виявлення та класифікації аномалій у мережевому трафіку телекомунікаційних систем і мереж в умовах проведення DDoS-атак різних типів і векторів впливу.

Предметом дослідження є математичні методи, статистичні моделі та алгоритми машинного навчання, що застосовуються для виявлення DDoS-атак шляхом аналізу характеристик мережевих потоків та поведінкових профілів трафіку.

Для досягнення поставленої мети необхідно вирішити такі завдання: по-перше, проаналізувати структуру та принципи функціонування сучасних телекомунікаційних систем і мереж; по-друге, систематизувати та класифікувати основні типи і механізми реалізації DDoS-атак; по-третє, здійснити порівняльний аналіз існуючих методів виявлення атак у мережевому трафіку; по-четверте, розробити математичну модель виявлення DDoS-атак на основі статистичного аналізу та ентропійних методів; по-п’яте, провести дослідження ефективності запропонованої моделі та обґрунтувати її практичну застосовність.

Методи дослідження: системний аналіз та синтез при вивченні структури DDoS-атак; теорія ймовірностей та математична статистика при формалізації моделей трафіку; теорія інформації (апарат ентропії Шеннона) для виявлення аномалій; методи машинного навчання (класифікація, кластеризація) для побудови детектора атак; імітаційне моделювання для верифікації запропонованих рішень.

Практична значущість роботи полягає в тому, що розроблена математична модель може бути використана як основа для побудови реальних систем виявлення DDoS-атак (IDS/IPS) у телекомунікаційних мережах операторів зв’язку, корпоративних мережах та центрах обробки даних. Запропоновані алгоритми характеризуються низькою обчислювальною складністю, що дозволяє їх реалізацію в умовах жорстких вимог до затримок обробки трафіку в реальному часі.

РОЗДІЛ 1 Теоретичні основи функціонування телекомунікаційних мереж та DDoS-атак

1.1. Структура та принципи функціонування телекомунікаційних систем і мереж

Розуміння природи DDoS-атак та механізмів їх впливу неможливе без ґрунтовного аналізу архітектури сучасних телекомунікаційних систем і мереж. Дослідження структурних особливостей, протокольного стеку та принципів маршрутизації трафіку формує необхідну теоретичну базу для побудови ефективних моделей виявлення аномалій. Перший розділ кваліфікаційної роботи присвячений систематизації теоретичних знань у цих взаємопов'язаних предметних областях. У розділі послідовно розглядаються: структура та принципи функціонування телекомунікаційних систем, класифікація і механізми реалізації DDoS-атак, а також існуючі методи виявлення атак у мережевому трафіку. Така логічна послідовність дозволяє сформувати цілісне уявлення про проблематику захисту мережевої інфраструктури від розподілених атак на відмову в обслуговуванні.

Телекомунікаційна система являє собою сукупність технічних засобів, програмного забезпечення та організаційних заходів, що забезпечують передачу, комутацію та обробку інформації між просторово розподіленими абонентами. У найширшому розумінні телекомунікаційна система охоплює всі компоненти, необхідні для встановлення, підтримання та завершення сеансів зв'язку - від фізичного середовища передачі до прикладних протоколів кінцевого рівня. Розвиток телекомунікаційних систем пройшов кілька ключових етапів: від аналогових комутованих телефонних мереж загального користування (PSTN) через цифрові мережі з інтеграцією служб (ISDN) до сучасних пакетних IP-мереж, що забезпечують конвергентне передавання голосу, відео та даних в єдиній транспортній інфраструктурі [1].

Сучасні телекомунікаційні мережі є складними гетерогенними системами, що інтегрують різнорідні технології передачі даних - від оптоволоконних магістралей до бездротових мереж останнього покоління. Гетерогенність

мережевого середовища зумовлена різноманітністю завдань, які вирішують телекомунікаційні системи: забезпечення широкосмугового доступу в Інтернет для індивідуальних абонентів, підключення корпоративних клієнтів до глобальної мережі, організація резервованих каналів зв'язку для критичної інфраструктури, надання хмарних і мультимедійних сервісів. Кожна із цих задач висуває специфічні вимоги до параметрів мережі та зумовлює застосування відповідних технологічних рішень. Архітектура сучасних телекомунікаційних мереж базується на ієрархічному принципі організації, що передбачає виділення кількох рівнів: базового рівня доступу, рівня агрегації та рівня ядра мережі. Така структура забезпечує масштабованість, відмовостійкість і ефективне управління трафіком у мережах провайдерів і великих корпоративних операторів.

Рівень доступу є точкою підключення кінцевих пристроїв - комп'ютерів, смартфонів, IoT-пристроїв, IP-камер - до мережевої інфраструктури. На цьому рівні застосовуються технології DSL, GPON, кабельне телебачення стандарту DOCSIS, бездротовий доступ стандартів Wi-Fi та LTE/5G. Рівень доступу характеризується найвищою щільністю вузлів та відносно обмеженою пропускною здатністю каналів. Саме на цьому рівні концентрується значна частина пристроїв, що стають частиною ботнетів та беруть участь у DDoS-атаках - насамперед незахищені IoT-пристрої з вразливим вбудованим програмним забезпеченням.

Рівень агрегації виконує функцію концентрації трафіку від рівня доступу та його передачі на рівень ядра. На цьому рівні реалізуються функції маршрутизації між підмережами, фільтрації трафіку за допомогою списків контролю доступу (ACL), сегментації мережі через VLAN, а також застосовуються перші засоби контролю якості обслуговування. Обладнання рівня агрегації - як правило, комутатори третього рівня та маршрутизатори середнього класу - обробляє агрегований трафік від тисяч кінцевих абонентів, що ставить підвищені вимоги до його продуктивності.

Рівень ядра мережі є найвищим ієрархічним рівнем і забезпечує високошвидкісну транзитну передачу трафіку між вузлами агрегації та

зовнішніми мережами. Ядро мережі будується на базі високопродуктивних маршрутизаторів операторського класу та оптоволоконних каналів із пропускною здатністю від 10 Гбіт/с до 100 Гбіт/с і вище. Відмовостійкість ядра мережі забезпечується дублюванням обладнання та надлишковими топологіями зв'язку - повнозв'язною або кільцевою - що мінімізує час перемикання при відмові окремих вузлів чи каналів [2]. Окремим важливим компонентом архітектури телекомунікаційних мереж провайдерів є точки обміну трафіком (Internet Exchange Points, IXP) - фізичні інфраструктури, де автономні системи різних операторів обмінюються трафіком безпосередньо, минаючи транзитних провайдерів. Великі IXP, такі як DE-CIX у Франкфурті або LINX у Лондоні, обробляють трафік порядку кількох терабіт на секунду і відіграють критичну роль у глобальному транспортуванні інтернет-трафіку.

Функціонування телекомунікаційних мереж підпорядковане стандартизованій еталонній моделі OSI (Open Systems Interconnection), що визначає сім рівнів взаємодії відкритих систем: фізичний, канальний, мережевий, транспортний, сеансовий, представницький та прикладний. Кожен рівень виконує строго визначені функції та взаємодіє з суміжними рівнями через стандартизовані інтерфейси, що забезпечує сумісність обладнання різних виробників.

Фізичний рівень (Layer 1) відповідає за передачу бітового потоку через фізичне середовище: електричні сигнали в мідних кабелях, світлові імпульси в оптоволокні, радіохвилі в бездротових мережах. Канальний рівень (Layer 2) забезпечує адресацію вузлів у межах одного сегмента мережі за допомогою MAC-адрес та реалізує протоколи управління доступом до середовища (Ethernet, Wi-Fi). Мережевий рівень (Layer 3) відповідає за логічну адресацію та маршрутизацію пакетів між мережами - ключові функції, що реалізуються протоколом IP. Транспортний рівень (Layer 4) забезпечує наскрізну передачу даних між прикладними процесами з необхідним рівнем надійності. Рівні 5–7 (сеансовий, представницький та прикладний) відповідають за управління сеансами зв'язку, представлення та кодування даних, а також реалізацію

прикладних протоколів - HTTP, DNS, SMTP, FTP тощо. На практиці домінуючим є стек протоколів TCP/IP, що об’єднує функції рівнів OSI в чотири рівні: канального доступу, міжмережевий (Internet), транспортний та прикладний. Стек TCP/IP є основою функціонування глобальної мережі Інтернет та переважної більшості корпоративних і телекомунікаційних мереж. Розуміння особливостей кожного рівня цього стеку є принципово важливим для аналізу DDoS-атак, оскільки різні типи атак спрямовані на різні рівні стеку й експлуатують специфічні особливості відповідних протоколів.

Ключовим протоколом мережевого рівня сучасних IP-мереж є протокол IPv4/IPv6, що забезпечує логічну адресацію вузлів та маршрутизацію пакетів через гетерогенне мережеве середовище. Протокол IPv4, стандартизований у 1981 році в RFC 791, досі залишається домінуючим у більшості мережевих інфраструктур, хоча простір 32-бітних адрес де-факто вичерпаний ще в 2011 році. IPv4-заголовок містить поля, що безпосередньо впливають на безпеку мережі: поле Time to Live (TTL), що обмежує час життя пакета в мережі та запобігає їх нескінченній циркуляції; поле протоколу, що визначає протокол верхнього рівня; поля адрес джерела та призначення, що можуть бути підроблені зловмисниками (IP spoofing) при організації атак. Протокол IPv6, що використовує 128-бітну адресацію, поступово набирає поширення у глобальній мережі та телекомунікаційних інфраструктурах. Окрім розширеного адресного простору, IPv6 вносить принципові зміни в архітектуру протоколу: обов’язкова підтримка IPsec, механізм автоматичної конфігурації адрес (SLAAC), відмова від широкомовних пакетів на користь мультикасту [3]. Втім, перехід на IPv6 породжує нові виклики для безпеки: подвійна стекова конфігурація (dual-stack) збільшує поверхню атаки, а менший досвід адміністраторів у роботі з IPv6 підвищує ймовірність помилок конфігурування. Зокрема, атаки на протокол виявлення сусідів NDP (Neighbor Discovery Protocol), що замінює ARP у середовищі IPv6, стають новим вектором мережевих атак.

Маршрутизація здійснюється на основі таблиць маршрутизації, що формуються протоколами динамічного маршрутизації - OSPF, BGP, EIGRP.

Протокол OSPF (Open Shortest Path First) є протоколом внутрішньої маршрутизації (IGP), що реалізує алгоритм стану каналу Дейкстри для побудови дерева найкоротших шляхів у межах автономної системи. OSPF підтримує ієрархічну організацію мережі через поняття зон (areas), що знижує навантаження на маршрутизатори при обміні маршрутною інформацією у великих мережах.

Протокол BGP (Border Gateway Protocol) відіграє особливу роль, забезпечуючи обмін маршрутною інформацією між автономними системами на глобальному рівні Інтернету. BGP є єдиним протоколом зовнішньої маршрутизації (EGP), що використовується в Інтернеті, та формує так звану «тканину» глобальної маршрутизації. Вразливості протоколу BGP - зокрема відсутність автентифікації маршрутних оголошень у базовій специфікації - можуть бути використані для організації атак типу BGP hijacking, коли зловмисна автономна система оголошує чужі префікси IP-адрес та перехоплює трафік, призначений іншим організаціям. Такі інциденти неодноразово траплялися і завдавали значної шкоди глобальній мережевій зв’язності.

Для протидії BGP-атакам розроблені механізми RPKI (Resource Public Key Infrastructure) та BGPsec, що забезпечують криптографічну верифікацію маршрутних оголошень [4]. Впровадження цих механізмів залишається важливим завданням для глобального інтернет-спільноти, оскільки підміна маршрутів є одним із засобів організації великомасштабних DDoS-атак через перенаправлення трафіку через підконтрольну зловмиснику інфраструктуру.

Транспортний рівень забезпечує наскрізну передачу даних між прикладними процесами та представлений двома ключовими протоколами: TCP (Transmission Control Protocol) і UDP (User Datagram Protocol). TCP реалізує надійну орієнтовану на з’єднання передачу з механізмами контролю потоку і перевантаження, тоді як UDP забезпечує ненадійну передачу без встановлення з’єднання з мінімальними накладними витратами. Особливості механізмів цих протоколів активно експлуатуються зловмисниками при організації DDoS-атак.

Процедура встановлення TCP-з’єднання реалізується через тристороннє рукостискання (three-way handshake): клієнт надсилає SYN-пакет, сервер відповідає SYN-ACK, клієнт завершує встановлення з’єднання пакетом ACK. Механізм збереження напіввідкритих з’єднань у черзі очікування (backlog queue) на стороні сервера став основою для атак типу SYN flood - зловмисник надсилає велику кількість SYN-пакетів із підробленими адресами джерела, переповнюючи чергу очікування та виснажуючи ресурси сервера. Для боротьби з цим типом атак розроблені механізми SYN cookies, що дозволяють серверу не зберігати стан напіввідкритих з’єднань. Протокол UDP, позбавлений механізмів встановлення з’єднання та контролю доставки, є ідеальним інструментом для організації amplification-атак. Зловмисник надсилає UDP-запит із підробленою адресою джерела (IP-адрес жертви) на сервери протоколів, що генерують значно більшу відповідь, ніж запит - DNS, NTP, SSDP, Memcached. Відповіді сервера спрямовуються на IP-адресу жертви, забезпечуючи багаторазове підсилення трафіку атаки. Коефіцієнт підсилення (amplification factor) може досягати 50–70 разів для DNS і сотень разів для протоколу Memcached, що робить ці атаки надзвичайно ефективними з точки зору зловмисника [5].

Інфраструктуру сучасних телекомунікаційних мереж формують різнорідні активні компоненти: маршрутизатори, комутатори, міжмережеві екрани (firewall), балансувальники навантаження, системи виявлення та запобігання вторгненням (IDS/IPS), а також спеціалізовані пристрої очищення трафіку (scrubbing centers). Кожен з цих елементів виконує специфічні функції забезпечення продуктивності, безпеки та керованості мережі. Маршрутизатори операторського класу є ключовими елементами транспортної інфраструктури та забезпечують пересилання пакетів зі швидкостями в десятки та сотні гігабіт на секунду. Сучасні маршрутизатори реалізують апаратне прискорення пересилання пакетів через спеціалізовані мікросхеми ASIC та мережеві процесори, що дозволяє обробляти трафік на лінійній швидкості. Програмна частина маршрутизатора - операційна система та протокольний стек - відповідає за реалізацію протоколів маршрутизації, управління та конфігурування.

Міжмережеві екрани (firewall) реалізують політику безпеки мережі через фільтрацію трафіку на основі заданих правил. Еволюція міжмережевих екранів пройшла шлях від простих пакетних фільтрів через екрани з відстеженням стану з'єднань (stateful inspection) до сучасних міжмережевих екранів наступного покоління (NGFW), що виконують глибоку інспекцію пакетів (DPI) та здатні ідентифікувати прикладні протоколи незалежно від використовуваних портів. NGFW-рішення інтегрують функції IPS, антивірусної перевірки, URL-фільтрації та аналізу SSL-трафіку, формуючи комплексний периметр безпеки.

Балансувальники навантаження розподіляють вхідні запити між кількома серверами з метою рівномірного завантаження ресурсів та забезпечення відмовостійкості. Алгоритми балансування - Round Robin, Least Connections, IP Hash, Resource Based - визначають логіку розподілу сесій між серверами пулу. Балансувальники навантаження відіграють важливу роль у забезпеченні стійкості до DDoS-атак, розподіляючи навантаження та приховуючи реальні IP-адреси серверів за єдиною зовнішньою адресою. Центри очищення трафіку (scrubbing centers) є спеціалізованими інфраструктурними вузлами, що забезпечують фільтрацію шкідливого трафіку при DDoS-атаках. Трафік, що надходить на IP-адресу атакованого ресурсу, перенаправляється (через BGP або DNS) на scrubbing center, де проходить багаторівневу очистку від атакуючого трафіку, після чого легітимний трафік доставляється до захищеного ресурсу. Провідні постачальники послуг захисту від DDoS - Cloudflare, Akamai, Radware - розгортають scrubbing centers у ключових точках присутності по всьому світу, забезпечуючи ємність поглинання атак у сотні гігабіт на секунду [6].

Концепція програмно-визначених мереж (SDN, Software-Defined Networking) та віртуалізації мережевих функцій (NFV, Network Functions Virtualization) революціонізує архітектуру сучасних телекомунікаційних систем. Відокремлення площини управління від площини передачі даних дозволяє централізовано керувати мережевою інфраструктурою через програмні інтерфейси, забезпечуючи гнучку реконфігурацію потоків трафіку у відповідь на виявлені аномалії в режимі реального часу. В архітектурі SDN виділяються три

рівні: рівень інфраструктури (data plane), рівень управління (control plane) та рівень застосунків (application plane). Рівень інфраструктури представлений мережевими комутаторами та маршрутизаторами з підтримкою протоколу OpenFlow, що відповідають за пересилання пакетів відповідно до правил, завантажених контролером. SDN-контролер (рівень управління) реалізує централізовану логіку маршрутизації та безпеки, маючи глобальний огляд стану мережі. Рівень застосунків забезпечує взаємодію між додатками для управління мережею та контролером через програмні інтерфейси (northbound API).

Застосування SDN для захисту від DDoS-атак відкриває нові можливості: централізований контролер здатний миттєво змінювати правила пересилання трафіку на всіх вузлах мережі, перенаправляючи атакуючий трафік на засоби очищення або блокуючи його ще на граничних вузлах. Дослідження у цій галузі демонструють ефективність SDN-підходу для реалізації адаптивних систем захисту з часом реагування, що суттєво перевищує можливості традиційних апаратних рішень.

NFV передбачає реалізацію мережевих функцій - маршрутизації, фільтрації, балансування навантаження, NAT - у вигляді програмних компонентів (VNF, Virtual Network Functions), що виконуються на стандартних серверах загального призначення. Це знижує залежність від спеціалізованого апаратного забезпечення, прискорює розгортання нових сервісів та спрощує масштабування мережевих функцій безпеки у відповідь на зростання інтенсивності атак. Якість обслуговування (QoS, Quality of Service) є критичним параметром телекомунікаційних мереж, що визначає набір характеристик передачі трафіку: пропускну здатність, затримку, джиттер та рівень втрати пакетів. Механізми QoS реалізуються через пріоритизацію черг, управління перевантаженням та формування трафіку. DDoS-атаки безпосередньо впливають на показники QoS, деградуючи якість обслуговування легітимних користувачів аж до повної недоступності сервісів.

Пропускна здатність (bandwidth) визначає максимальний обсяг даних, що може бути переданий через канал зв’язку за одиницю часу. Volumetric DDoS-

атаки безпосередньо вичерпують пропускну здатність каналів, генеруючи трафік, що перевищує їхню ємність [7]. Затримка (latency) і джиттер (jitter) - дисперсія затримки - є критичними параметрами для додатків реального часу: голосового зв'язку (VoIP), відеоконференцій, онлайн-ігор. Зростання затримки та джиттеру при DDoS-атаках призводить до деградації якості голосових та відеосервісів ще до повного вичерпання пропускної здатності. Рівень втрати пакетів є ще одним показником, що суттєво зростає в умовах DDoS-атаки: перевантажені вузли відкидають пакети, що не вміщуються в чергах, що призводить до повторних передач та зростання ефективного навантаження.

Диференційована служба та інтегрована служба є двома основними архітектурами QoS в IP-мережах. DiffServ забезпечує пріоритизацію трафіку на основі значення поля DSCP у заголовку IP-пакета, дозволяючи виділяти привілейоване обслуговування для критичного трафіку навіть в умовах перевантаження мережі. Коректно налаштовані механізми QoS можуть частково пом'якшити вплив DDoS-атак на критичні сервіси, гарантуючи їм мінімальну смугу пропускання навіть при насиченні каналу атакуючим трафіком. Моніторинг та управління телекомунікаційними мережами здійснюється за допомогою систем мережевого управління (NMS) і операційних систем підтримки (OSS), що реалізують стандарти SNMP, NetFlow/IPFIX та NETCONF/YANG. Ці системи збирають статистику роботи мережевих елементів, що є важливим джерелом даних для виявлення аномалій трафіку та ідентифікації DDoS-атак на основі аналізу відхилень від нормального профілю навантаження.

Протокол SNMP (Simple Network Management Protocol) є базовим стандартом управління мережевим обладнанням, що дозволяє збирати статистику інтерфейсів, стан маршрутних таблиць та безліч інших операційних параметрів. SNMP-пастки (traps) забезпечують проактивне повідомлення NMS про значущі події - відмови обладнання, перевищення порогових значень навантаження. Однак SNMP має суттєві обмеження для цілей виявлення DDoS: відносно низька частота опитування (polling interval) та агрегований характер

лічильників не дозволяють виявляти короткочасні імпульси атак. Протоколи NetFlow (Cisco) та його стандартизований варіант IPFIX дозволяють збирати детальну статистику мережевих потоків: IP-адреси та порти джерела та призначення, протокол транспортного рівня, кількість переданих байтів та пакетів, часові мітки початку та закінчення потоку. Аналіз даних NetFlow/IPFIX є одним із найбільш ефективних підходів до виявлення DDoS-атак у мережах операторського рівня, оскільки дозволяє виявляти аномальні патерни трафіку без необхідності аналізувати вміст пакетів. Стандарт NETCONF/YANG визначає сучасний програмний інтерфейс управління мережевим обладнанням, що приходить на зміну застарілому SNMP у сучасних SDN-архітектурах. YANG є мовою моделювання даних, що описує конфігурацію та операційний стан мережевих пристроїв, а NETCONF забезпечує транспортний рівень для операцій читання, запису та видалення конфігурацій. Програмне управління мережевим обладнанням через NETCONF/YANG відкриває можливості для автоматизованого реагування на DDoS-атаки шляхом динамічної зміни конфігурації ACL та маршрутних політик.

Безпека телекомунікаційних мереж забезпечується комплексом організаційно-технічних заходів, що включає сегментацію мережі, фільтрацію трафіку, шифрування даних та системи автентифікації. Захист від DDoS-атак потребує окремої уваги завдяки специфіці цього типу загроз: на відміну від більшості кібератак, DDoS використовує розподілені ресурси та легітимні мережеві протоколи, що ускладнює ефективне розмежування атакуючого та легітимного трафіку засобами традиційного периметрового захисту.

Концепція «глибокого захисту» (defense in depth) передбачає розгортання засобів захисту на кількох рівнях мережевої архітектури - від граничних маршрутизаторів до серверних платформ. На рівні провайдера застосовуються такі механізми, як Remotely Triggered Black Hole (RTBH) - оголошення маршруту до атакованої IP-адреси з наступним хопом null0, що призводить до відкидання всього трафіку; та Flowspec (RFC 5575) - розповсюдження через BGP детальних правил фільтрації трафіку на основі кількох полів заголовка пакета. Ці механізми

дозволяють оперативно блокувати атакуючий трафік ще у мережі провайдера, не допускаючи його до інфраструктури клієнта. Зростання масштабів DDoS-атак - рекордні атаки вимірюються вже терабітами трафіку на секунду - вимагає залучення ресурсів хмарних провайдерів та спеціалізованих сервісів захисту для поглинання надлишкового трафіку [8]. Це зумовлює активний розвиток гібридних архітектур захисту, де локальні засоби протидії поєднуються з можливостями хмарного «scrubbing» великих обсягів трафіку, що в сумі формує багаторівневий захист телекомунікаційної інфраструктури від сучасних розподілених атак на відмову в обслуговуванні.

1.2. Основні типи та механізми реалізації DDoS-атак

DDoS-атаки (Distributed Denial of Service) являють собою координовані дії з використанням розподіленої мережі атакуючих вузлів (ботів), спрямовані на вичерпання обчислювальних, мережевих або програмних ресурсів цільового об’єкта з метою унеможливлення надання послуг легітимним користувачам. На відміну від звичайних DoS-атак, що генеруються з одного джерела і порівняно легко блокуються фільтрацією за IP-адресою, розподілений характер DDoS суттєво ускладнює їх виявлення та нейтралізацію традиційними засобами захисту периметра. Тисячі або мільйони атакуючих вузлів, розподілених по всьому світу, роблять блокування за географічною ознакою або адресою джерела практично неможливим без одночасного відсікання значної частини легітимного трафіку. Еволюція DDoS-атак нерозривно пов’язана з розвитком інтернет-інфраструктури та зростанням кількості підключених пристроїв. Перші масштабні DDoS-атаки з’явилися наприкінці 1990-х - початку 2000-х років і були орієнтовані переважно на флуд мережевого рівня. З часом атаки суттєво ускладнилися: зросли їх обсяги, з’явилися нові вектори на рівні застосунків, сформувався комерційний ринок DDoS-послуг («DDoS-as-a-Service»), де будь-яка особа може замовити атаку за відносно невелику суму без жодних технічних знань. Сучасні атаки нерідко є багатовекторними, одночасно задіюючи кілька типів DDoS для обходу засобів захисту.

Мотивація організаторів DDoS-атак є різноманітною. Серед основних мотивів виділяють: конкурентну боротьбу (атаки на бізнес-конкурентів), здирництво (вимога викупу під загрозою атаки або під час активної атаки), хактивізм (атаки на ресурси організацій чи державних структур із метою висловлення протесту), кібервійна (атаки, що організовуються або підтримуються державними структурами для дестабілізації критичної інфраструктури противника), а також демонстрацію можливостей або банальне хуліганство. Незалежно від мотивації наслідки DDoS-атак для жертви є однотипними: фінансові збитки від простою сервісу, репутаційні втрати та витрати на відновлення і посилення захисту.

Класифікація DDoS-атак за об’єктом впливу виділяє три основні категорії. Volumetric-атаки (об’ємні) спрямовані на перевантаження пропускної здатності мережевого каналу жертви шляхом генерації надмірного обсягу трафіку. Protocol-атаки використовують вразливості та особливості мережевих протоколів для вичерпання стану з’єднань і ресурсів мережевого обладнання. Application Layer-атаки (атаки рівня застосунків) імітують легітимні запити до веб-сервісів, виснажуючи обчислювальні ресурси сервера та бази даних. Volumetric-атаки вимірюються в гігабітах або терабітах трафіку на секунду (Гбіт/с, Тбіт/с) і спрямовані на насичення каналів зв’язку жертви або її провайдера. Ключова особливість цих атак полягає в тому, що жертва стає недоступною ще до досягнення власне серверної інфраструктури - канал зв’язку виявляється переповненим на рівні провайдера або точки обміну трафіком. Рекордні зафіксовані обсяги volumetric-атак вимірюються вже терабітами: у 2018 році GitHub зазнав атаки потужністю 1,35 Тбіт/с, а вже у 2020 році Amazon Web Services відбила атаку обсягом 2,3 Тбіт/с [9]. Захист від volumetric-атак потребує залучення ресурсів великих хмарних провайдерів або спеціалізованих «scrubbing»-сервісів із розподіленою мережею точок очищення трафіку.

Protocol-атаки вимірюються в пакетах за секунду (PPS) або у мільйонах запитів за секунду (MRPS) і завдають удару по обчислювальним ресурсам мережевого обладнання - маршрутизаторів, міжмережевих екранів,

балансувальників навантаження. Ці пристрої мають обмежену таблицю станів з'єднань (state table), і при її переповненні вони перестають обробляти нові легітимні з'єднання. Protocol-атаки є небезпечними навіть при відносно невеликому обсязі трафіку: сотні тисяч пакетів за секунду можуть вивести з ладу апаратний міжмережевий екран, не насичуючи при цьому мережевого каналу. Application Layer-атаки (рівень L7) вимірюються в запитах за секунду (RPS) і є найскладнішими для виявлення, оскільки кожен окремий запит синтаксично та семантично відповідає легітимному. Виявлення таких атак потребує глибокого поведінкового аналізу та розуміння «нормального» профілю навантаження сервісу.

Додатковою класифікацією є розподіл DDoS-атак за вектором: прямі атаки, де атакуючі вузли безпосередньо надсилають трафік жертві; атаки відбиття (reflection), де зловмисник використовує сторонні сервери-посередники для перенаправлення трафіку до жертви; та атаки підсилення (amplification), що є різновидом атак відбиття з коефіцієнтом підсилення, що багаторазово збільшує обсяг трафіку відносно початково надісланого зловмисником [10].

З точки зору архітектури організації, DDoS-атаки реалізуються через ботнет-мережі - розподілені системи заражених шкідливим програмним забезпеченням пристроїв (ботів), що перебувають під централізованим управлінням зловмисника. Сучасні ботнети можуть налічувати мільйони пристроїв, включаючи персональні комп'ютери, смартфони, сервери та IoT-пристрої. Управління ботнетом здійснюється через командно-контрольну інфраструктуру (C2/C&C), що може використовувати IRC, HTTP, P2P або інші протоколи комунікації. Класична централізована архітектура ботнету передбачає ієрархічну структуру: оператор ботнету (botmaster) → C2-сервери → боти. Команди на проведення атаки розповсюджуються від botmaster через C2-сервери до ботів, які безпосередньо генерують атакуючий трафік. Перевагою такої архітектури є простота управління, проте вона має критичний недолік - вразливість до виведення з ладу C2-серверів правоохоронними органами або провайдерами, що може паралізувати весь ботнет.

Децентралізована P2P-архітектура ботнету позбавлена цього недоліку: кожен бот одночасно є і клієнтом, і сервером, отримуючи та ретранслюючи команди через одноранговий протокол. Ботнети на основі P2P (наприклад, Zeus, Kelihos) є значно більш стійкими до дії правоохоронних органів, оскільки не мають єдиної точки відмови. Виявлення та нейтралізація P2P-ботнетів вимагає складніших методів - наприклад, sinkhol ing (підміна DNS-записів для перехоплення трафіку ботнету) та злому протоколу (reverse engineering C2-протоколу для надсилання команд самознищення ботам). Окремою категорією є ботнети на базі IoT-пристроїв. Поява шкідливого програмного забезпечення Mirai у 2016 році відкрила нову сторінку в еволюції ботнетів: мільйони незахищених IP-камер, маршрутизаторів та інших IoT-пристроїв із стандартними заводськими паролями були заражені та використані для проведення рекордних на той час DDoS-атак, зокрема атаки на провайдера Dyn, що призвела до недоступності великої кількості популярних інтернет-сервісів. Вихідний код Mirai було оприлюднено, що спровокувало появу численних його модифікацій та наступників - Hajime, Satori, Okiru. IoT-ботнети є особливо небезпечними через колосальну кількість потенційно вразливих пристроїв, відсутність вбудованих механізмів безпеки та складність оновлення прошивки власниками.

SYN Flood є одним з найпоширеніших механізмів protocol-атак, що експлуатує механізм тристороннього рукостискання протоколу TCP. Зловмисник надсилає масу SYN-пакетів із підробленими IP-адресами джерела, змушуючи сервер виділяти ресурси для незавершених з’єднань (half-open connections) і заповнювати чергу очікування SYN до переповнення. Коли черга вичерпана, сервер більше не може прийняти нові легітимні з’єднання, що призводить до відмови в обслуговуванні.

Детальніше механізм атаки розгортається таким чином: при отриманні SYN-пакета сервер виділяє структуру даних у пам’яті для відстеження стану напіввідкритого з’єднання, надсилає у відповідь SYN-ACK пакет та запускає таймер очікування ACK-підтвердження. Оскільки адреса джерела підроблена, SYN-ACK відповідь надходить до несправжнього хоста, що ніколи не надішле

ACK. Сервер утримує запис про напіввідкрите з'єднання до спливу тайм-ауту (зазвичай 75 секунд), після чого видаляє його. При масованому надходженні SYN-пакетів таблиця напіввідкритих з'єднань (backlog queue) переповнюється значно швидше, ніж встигають спливати тайм-аути [11]. Для протидії SYN Flood розроблені механізми SYN Cookies - детерміністичний алгоритм генерації початкового порядкового номера TCP (ISN) на основі хешу адрес та портів обох сторін і секретного ключа сервера. При включених SYN Cookies сервер не зберігає стану напіввідкритого з'єднання після надсилання SYN-ACK: коли легітимний клієнт надішле ACK з правильним підтвердженням, сервер зможе відтворити всю необхідну інформацію про з'єднання з ACK-пакета без звернення до збереженого стану. Це дозволяє серверу витримувати масовані SYN Flood атаки без переповнення черги, проте SYN Cookies мають обмеження - зокрема, не підтримують деякі TCP-опції.

Додаткові засоби протидії SYN Flood включають: обмеження частоти SYN-пакетів (SYN rate limiting) на маршрутизаторах та міжмережевих екранах, зменшення тайм-ауту напіввідкритих з'єднань, збільшення розміру черги backlog, а також застосування апаратних рішень TCP offload, що виконують перевірку рукостискання на виділеному процесорі.

DNS Amplification - потужний тип атаки підсилення (amplification), що використовує відкриті DNS-резолвери для ампліфікації атакуючого трафіку. Зловмисник надсилає DNS-запити з підробленою IP-адресою жертви, а DNS-сервери повертають значно більші відповіді безпосередньо жертві. Коефіцієнт ампліфікації може сягати 50–70 разів, що дозволяє атакуючому з обмеженою пропускною здатністю генерувати колосальний трафік у напрямку цілі.

Механізм DNS Amplification будується на кількох ключових особливостях DNS-протоколу. По-перше, DNS працює переважно на UDP, що дозволяє підроблювати адресу джерела без необхідності проходити TCP-рукостискання. По-друге, DNS-запит є коротким (зазвичай 20–60 байтів), тоді як відповідь, особливо для записів типу ANY або DNSSEC, може досягати 3000–4000 байтів і більше [12]. По-третє, Інтернет все ще містить мільйони відкритих DNS-

резолверів, що обробляють запити від будь-яких IP-адрес, а не лише від своїх клієнтів. Для протидії DNS Amplification рекомендується закрити відкриті рекурсивні резолвери, впровадити BCP38 (фільтрацію вихідного трафіку з підробленими адресами) на рівні провайдерів, а також застосовувати DNS Response Rate Limiting (RRL). Аналогічний принцип застосовується в NTP Amplification атаках, що використовують команду monlist протоколу Network Time Protocol. Ця команда повертає список останніх 600 хостів, що зверталися до NTP-сервера, генеруючи відповідь у сотні разів більшу за запит - коефіцієнт ампліфікації може досягати 556. Після масового використання цього вектора в атаках 2013–2014 років команда monlist була відключена в більшості сучасних реалізацій NTP.

Рекордним за коефіцієнтом підсилення є вектор Memcached Amplification, що використовує неправильно сконфігуровані сервери Memcached - системи кешування даних - доступні з Інтернету через UDP. Коефіцієнт підсилення для цього вектора може досягати 50 000, тобто запит у 15 байтів може спровокувати відповідь обсягом до 750 кілобайтів. Саме цей вектор був використаний в атаці на GitHub у 2018 році, що встановила рекорд за обсягом трафіку. Усунення вразливості полягає в закритті UDP-доступу до Memcached із публічного Інтернету та переконфігурації сервера для прослуховування лише на локальних інтерфейсах. SSDP Amplification використовує протокол Simple Service Discovery Protocol - частину стандарту UPnP, - що реалізований у мільйонах домашніх маршрутизаторів і SmartTV-пристроїв. Запит SSDP M-SEARCH генерує відповідь у 30–40 разів більшу, а сукупна кількість публічно доступних SSDP-пристроїв дозволяє сформувати атакуючий трафік у десятки гігабіт за секунду.

HTTP Flood є представником application layer атак (L7), що завдає удару по рівню веб-застосунків. На відміну від volumetric-атак, HTTP Flood не потребує великого об’єму трафіку - достатньо великої кількості легітимних HTTP GET або POST запитів до ресурсомістких сторінок або API-ендпоінтів. Оскільки кожен запит є синтаксично коректним, фільтрація таких атак на основі сигнатур є

надзвичайно складним завданням, яке потребує глибокого поведінкового аналізу.

HTTP GET Flood орієнтований на запити до сторінок, що вимагають значних обчислювальних ресурсів для формування відповіді: динамічні сторінки з великою кількістю запитів до бази даних, сторінки пошуку з не оптимізованими SQL-запитами, ресурсомісткі API-ендпоінти. Зловмисники цілеспрямовано вибирають «найдорожчі» з точки зору серверних ресурсів запити: наприклад, запити до пошукового движка з широкими параметрами фільтрації або запити до API агрегації даних. Навіть кілька тисяч таких запитів на секунду можуть паралізувати сервер, не насичуючи при цьому мережевий канал.

HTTP POST Flood підходить для атак на ресурси, що обробляють завантаження файлів або великих форм даних. Надсилання HTTP POST-запитів із великими тілами змушує сервер витрачати ресурси не лише на обробку запиту, але й на прийом даних. Особливо ефективними є POST-атаки проти ресурсів, що виконують обробку завантаженого вмісту - наприклад, зміну розміру зображень або конвертацію документів. Для протидії HTTP Flood застосовуються: CAPTCHA-механізми, що відсіюють автоматизовані запити; перевірка JavaScript challenge (клієнти, що не підтримують JavaScript, не є легітимними браузерами); аналіз поведінкових характеристик - швидкість кліків, рух миші, послідовність запитів; обмеження частоти запитів (rate limiting) на рівні IP-адреси або сесії; а також системи репутаційного аналізу IP-адрес, що ідентифікують відомих ботів та зловмисні мережі.

Атаки типу Slowloris реалізують унікальний підхід до відмови в обслуговуванні через утримання максимальної кількості з’єднань із веб-сервером у відкритому, але неповністю переданому стані. Зловмисник встановлює HTTP-з’єднання та надсилає заголовки запиту вкрай повільно, не завершуючи їх, що змушує сервер тримати з’єднання відкритим у нескінченному очікуванні. При вичерпанні пулу доступних з’єднань сервер стає недоступним для нових клієнтів, тоді як атаку складно виявити через мінімальний об’єм генерованого трафіку.

Технічно Slowloris надсилає HTTP-заголовки по одному рядку з великими інтервалами - наприклад, один рядок заголовка кожні 10–15 секунд. Веб-сервер, очікуючи завершення запиту, не закриває з'єднання, адже воно технічно є активним. Зловмиснику достатньо підтримувати кілька сотень або тисяч таких «млявих» з'єднань одночасно, щоб вичерпати пул потоків веб-сервера (Apache, наприклад, за замовчуванням підтримує 150 одночасних з'єднань). При цьому генерований трафік є мінімальним і практично непомітним на тлі фонового навантаження.

Схожий принцип реалізують атаки Slow POST (також відомі як RUDY - R-U-Dead-Yet?): зловмисник відправляє HTTP POST-запит із великим значенням Content-Length, але передає тіло запиту по кілька байтів із великими інтервалами. Сервер, очікуючи завершення передачі тіла запиту, утримує ресурси виділеного потоку, що призводить до поступового виснаження пулу з'єднань. Загальна категорія «low-and-slow» атак є особливо небезпечною тому, що традиційні засоби захисту, орієнтовані на виявлення аномально великого обсягу трафіку, не здатні їх ідентифікувати [13]. Захист від цих атак включає: зменшення тайм-аутів утримання з'єднань на рівні веб-сервера; застосування зворотних проксі-серверів (Nginx, HAProxy), що буферизують HTTP-запити та передають їх бекенду лише після повного отримання; обмеження кількості одночасних з'єднань від однієї IP-адреси; а також використання CDN, що поглинає ці атаки на граничних вузлах.

Сучасна тенденція у розвитку DDoS-атак - перехід до багатовекторних (multi-vector) кампаній, що одночасно або послідовно задіюють кілька типів атак. Наприклад, зловмисник може поєднати volumetric UDP flood для насичення каналу, SYN Flood для виснаження таблиці станів міжмережевого екрану та HTTP Flood для перевантаження веб-сервера. Мета багатовекторної атаки - одночасно задіяти кілька рівнів захисту жертви та змусити команду реагування розпорошувати зусилля між кількома фронтами.

Багатовекторні атаки є значно складнішими для нейтралізації: засоби захисту, що ефективно відбивають один тип атаки, можуть виявитися

неефективними проти іншого. Наприклад, scrubbing-центр, що блокує volumetric-трафік, може пропустити низькошвидкісну L7-атаку, якщо не має відповідних механізмів поведінкового аналізу. Ефективний захист від багатовекторних DDoS вимагає комплексного підходу з використанням рішень, що забезпечують видимість та захист на всіх рівнях мережевого стеку - від мережевого до прикладного.

Нижче наведена таблиця систематизує основні механізми реалізації DDoS-атак, таблиця 1.1.

Таблиця 1.1

Механізми реалізації DDoS-атак та їх характеристики

Тип атаки	**Механізм реалізації**	**Цільовий ресурс**	**Приклади інструментів**
Volumetric	Генерація надлишкового трафіку для перевантаження каналу передачі даних	Пропускна здатність мережі	LOIC, HOIC, Mirai Botnet
Protocol	Експлуатація вразливостей мережевих протоколів для виснаження серверних ресурсів	Стан з’єднань, ресурси брандмауера	SYN Flood, Ping of Death, Smurf

Продовження таблиці 1.1.

Application Layer	Атаки на рівні додатків, що імітують легітимні запити користувачів	CPU, RAM сервера, БД	Slowloris, RUDY, HTTP Flood
Amplification	Використання відкритих сервісів для ампліфікації трафіку у напрямку жертви	Пропускна здатність, DNS/NTP сервери	DNS Amplification, NTP Amp, SSDP
Reflection	Підробка IP-адреси жертви для перенаправлення відповідей сторонніх вузлів	Пропускна здатність мережі	UDP Reflection, ICMP Reflection
Slowloris	Утримання відкритих з'єднань через повільне відправлення HTTP-заголовків	Пул з'єднань веб-сервера	Slowloris, PyLoris, Goloris
Zero-day DDoS	Використання раніше невідомих вразливостей у протоколах або ПЗ	Будь-який компонент системи	Спеціалізовані exploit-інструменти
Botnet-based	Координована атака через мережу заражених пристроїв (ботів)	Вся інфраструктура жертви	Mirai, Zeus, Necurs Botnet
Multi-vector	Одночасне використання кількох типів атак для ускладнення захисту	Кілька рівнів інфраструктури	Комбіновані APT-кампанії

Багатовекторні (multi-vector) DDoS-атаки представляють найбільш складний і небезпечний клас загроз, що одночасно або послідовно комбінують кілька типів атак проти різних рівнів інфраструктури жертви [14]. Такі атаки значно ускладнюють побудову ефективної системи захисту, оскільки протидія одному вектору не забезпечує захисту від інших. Дослідження показують, що частка багатовекторних атак у загальній кількості DDoS-інцидентів неухильно зростає з року в рік.

Критичним компонентом сучасних DDoS-атак є техніки ухилення від виявлення (evasion techniques), що включають ротацію IP-адрес атакуючих вузлів, мімікрію під легітимний трафік, використання зашифрованих HTTPS-з'єднань та динамічну зміну характеристик атаки у відповідь на дії засобів захисту. Застосування цих технік суттєво знижує ефективність традиційних сигнатурних систем виявлення і обумовлює необхідність розробки адаптивних математичних моделей, здатних виявляти аномалії навіть в умовах активного маскування атаки, рис.1.1.

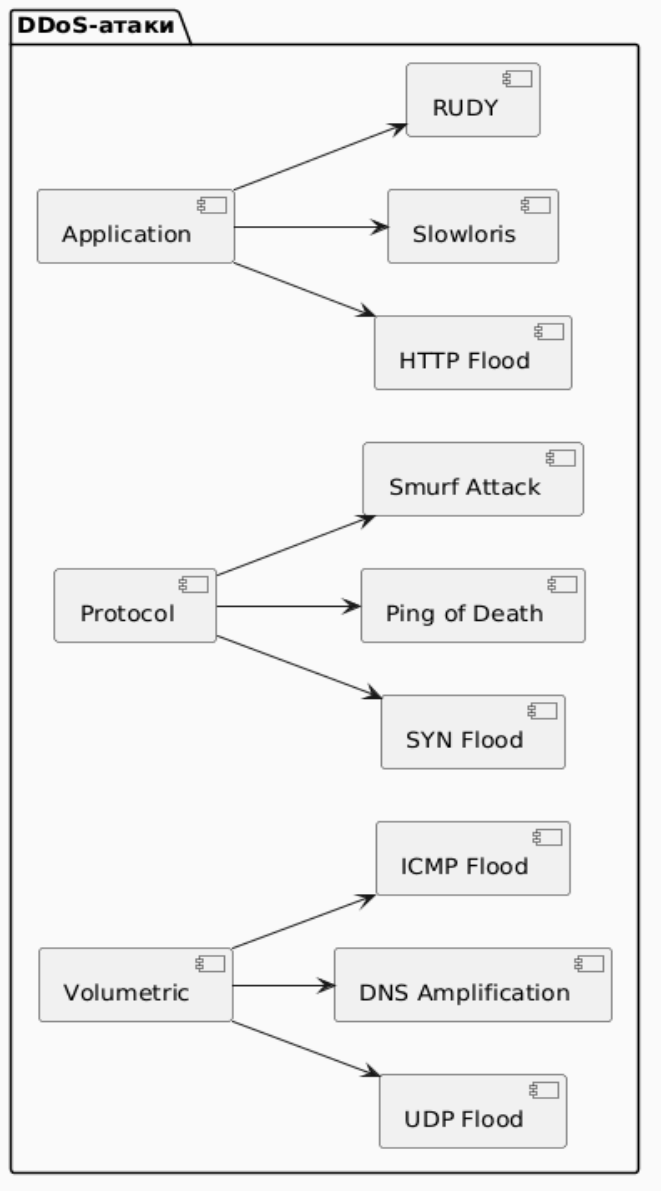

Рис. 1.1. Класифікація типів DDoS-атак [8].

Після представленої класифікації DDoS-атак стає очевидним, що кожен рівень інфраструктури має власні вразливості, які можуть бути використані

зловмисниками. Комбінування різних типів атак дозволяє значно підвищити їхню ефективність та ускладнити процес виявлення. Зокрема, одночасне застосування протокольних і об’ємних атак може призводити до перевантаження як мережевих ресурсів, так і серверних додатків. Це створює ситуацію, коли традиційні засоби фільтрації трафіку не здатні адекватно реагувати на загрозу. Важливим аспектом є також те, що атакуючі можуть адаптувати свою поведінку в режимі реального часу, змінюючи інтенсивність і характер трафіку. У таких умовах статичні правила захисту втрачають ефективність і потребують доповнення інтелектуальними механізмами аналізу. Особливу роль відіграють методи машинного навчання, які дозволяють виявляти приховані закономірності у мережевому трафіку. Вони забезпечують можливість своєчасного реагування навіть на нові, раніше невідомі типи атак. Крім того, використання поведінкових моделей дозволяє відрізняти легітимних користувачів від бот-мереж. Таким чином, комплексний підхід до аналізу та захисту від DDoS-атак є ключовою умовою забезпечення стійкості сучасних інформаційних систем.

1.3. Методи та підходи до виявлення DDoS-атак у мережевому трафіку

Виявлення DDoS-атак у мережевому трафіку є фундаментальним завданням мережевої безпеки, що потребує застосування різноманітних математичних та аналітичних методів. Ефективна система виявлення атак повинна задовольняти суперечливим вимогам: забезпечувати максимально можливу точність виявлення при мінімальному рівні хибнопозитивних спрацювань, обробляти трафік в режимі реального часу при масштабах сучасних телекомунікаційних мереж і демонструвати стійкість до технік ухилення.

Статистичний аналіз трафіку є базовим підходом до виявлення аномалій, що ґрунтується на порівнянні поточних характеристик трафіку з еталонним профілем нормальної поведінки мережі. Метод включає відстеження статистичних метрик: середнього значення, дисперсії, стандартного відхилення та квантилів розподілу ключових показників трафіку - кількості пакетів, байтів, нових з’єднань та унікальних IP-адрес за одиницю часу. Перевищення заданих

порогових значень сигналізує про потенційну атаку. Метод ентропійного аналізу базується на теорії інформації Шеннона і вимірює ступінь невизначеності (різноманітності) розподілу атрибутів мережевих пакетів - IP-адрес джерела і призначення, номерів портів та ідентифікаторів протоколів. У стані нормальної роботи мережі ці розподіли демонструють відносно стабільну ентропію. При DDoS-атаці ентропія IP-адрес джерела різко змінюється: при volumetric-атаках вона суттєво зростає через спуфінг, а при цільових атаках з вузького пулу джерел - знижується. Ентропійний аналіз є обчислювально ефективним та добре масштабованим методом.

Методи машинного навчання набувають дедалі більшого поширення у системах виявлення DDoS-атак завдяки здатності автоматично виявляти складні нелінійні залежності в даних без явного програмування правил. Серед застосовуваних алгоритмів виокремлюють методи навчання з учителем (Random Forest, Support Vector Machine, нейронні мережі) для класифікації відомих типів атак, а також методи навчання без учителя (K-Means, Autoencoder, Isolation Forest) для виявлення невідомих аномалій. Методи глибокого навчання, зокрема рекурентні нейронні мережі (LSTM), демонструють особливу ефективність при аналізі часових рядів мережевого трафіку.

Аналіз мережевих потоків на основі протоколів NetFlow та IPFIX забезпечує масштабований підхід до моніторингу трафіку у великих телекомунікаційних мережах без необхідності глибокої інспекції кожного пакету. Мережеві потоки агрегують метадані сотень чи тисяч пакетів одного з’єднання, значно скорочуючи обсяг даних, що аналізуються. Аналіз розподілу довжин потоків, кількості пакетів у потоці та часових інтервалів між потоками дозволяє ефективно виявляти більшість типів DDoS-атак при збереженні масштабованості системи. Глибока інспекція пакетів (DPI, Deep Packet Inspection) реалізує найбільш деталізований аналіз мережевого трафіку через повне дослідження вмісту пакетів на всіх рівнях стеку протоколів, включаючи корисне навантаження (payload). DPI дозволяє виявляти атаки рівня застосунків, що неможливо ідентифікувати за метаданими потоків. Основним обмеженням

методу є висока обчислювальна складність, що ускладнює застосування у лінійному режимі на каналах з пропускною здатністю понад десятки гігабіт на секунду.

Гібридні підходи, що поєднують переваги різних методів виявлення, забезпечують найбільш збалансоване співвідношення точності, продуктивності та стійкості до ухилення. Типова гібридна архітектура включає: перший рівень - швидкий ентропійний або статистичний аналіз для фільтрації явних аномалій; другий рівень - класифікатор машинного навчання для ідентифікації типу атаки; третій рівень - DPI-аналіз підозрілих потоків для підтвердження та детального дослідження. Такий каскадний підхід дозволяє оптимально розподілити обчислювальні ресурси, таблиця 1.2.

Таблиця 1.2.

Порівняльна характеристика методів виявлення DDoS-атак

Метод виявлення	**Принцип роботи**	**Переваги**	**Недоліки**
Статистичний аналіз	Порівняння поточних метрик трафіку з еталонними базовими показниками	Простота реалізації, низькі обчислювальні витрати	Висока частота хибнопозитивних спрацювань при нерегулярному трафіку
Сигнатурний аналіз	Зіставлення пакетів із базою відомих шаблонів атак	Висока точність для відомих атак, мінімум хибних спрацювань	Неефективний проти нових і модифікованих атак (zero-day)
Машинне навчання	Побудова класифікаційних моделей на основі навчальних наборів даних	Адаптивність, виявлення аномалій без сигнатур	Потребує великих датасетів, схильний до adversarial attacks

Продовження таблиці 1.2.

Ентропійний аналіз	Вимірювання ентропії розподілу IP-адрес, портів, протоколів у потоці	Ефективне виявлення volumetric і spoofed атак	Складність налаштування порогів, чутливість до легітимних флуктуацій
Аналіз потоків (NetFlow)	Агрегація та аналіз метаданих мережевих потоків без глибокої інспекції пакетів	Масштабованість, незалежність від шифрування	Відсутність вмісту пакетів знижує деталізацію аналізу
Глибока інспекція пакетів (DPI)	Аналіз вмісту пакетів на всіх рівнях моделі OSI	Найвища деталізація, виявлення атак рівня застосунків	Великі обчислювальні витрати, проблеми конфіденційності
Honeypot-системи	Розгортання пасток для збору інформації про атаки і атакуючих	Збір реальних даних про нові вектори атак	Не забезпечує захист в реальному часі, вимагає ізоляції
UEBA-аналіз	Профілювання поведінки користувачів та аналіз відхилень від норми	Виявлення інсайдерських загроз і складних атак	Висока складність впровадження, потребує тривалого навчання моделі

Представлена у таблиці 1.2 порівняльна характеристика методів виявлення DDoS-атак демонструє, що жоден із підходів не є універсальним рішенням для забезпечення повного захисту мережевої інфраструктури. Кожен із розглянутих методів має власні сильні сторони, які ефективно проявляються лише в певних умовах, а також обмеження, що знижують їхню результативність у випадку складних або комбінованих атак. Зокрема, традиційні підходи, такі як статистичний та сигнатурний аналіз, характеризуються відносною простотою реалізації, проте поступаються сучасним методам за рівнем адаптивності. Методи, засновані на машинному навчанні та поведінковому аналізі, відкривають нові можливості для виявлення невідомих та багатовекторних атак, однак потребують значних обчислювальних ресурсів і якісних навчальних вибірок. Водночас ентропійний аналіз та аналіз потоків забезпечують ефективний компроміс між швидкістю обробки та точністю виявлення, що

робить їх придатними для використання в умовах високонавантажених мереж. Глибока інспекція пакетів, незважаючи на свою високу точність, обмежена через ресурсоємність та питання конфіденційності, що особливо актуально в контексті сучасних вимог до захисту даних.

Окрему нішу займають honeypot-системи та UEBA-підходи, які більше орієнтовані на аналітику, дослідження та виявлення складних сценаріїв атак, включаючи внутрішні загрози. Їх використання доцільне як доповнення до основних механізмів захисту, а не як самостійне рішення. Таким чином, ефективна система виявлення DDoS-атак повинна базуватися на комплексному підході, що поєднує декілька методів із урахуванням специфіки мережі та потенційних загроз. Інтеграція різних технологій дозволяє компенсувати недоліки окремих підходів та забезпечити більш стійке і адаптивне реагування на сучасні кіберзагрози. Це, у свою чергу, підкреслює необхідність подальших досліджень у напрямку розробки гібридних моделей виявлення атак, які поєднують точність, швидкодію та здатність до самонавчання.

РОЗДІЛ 2 Розробка математичної моделі виявлення DDoS-атак

2.1. Обґрунтування вибору математичного апарату для моделювання мережевого трафіка

Сучасні телекомунікаційні системи характеризуються високою складністю структури та динамічністю процесів, що відбуваються в них, що обумовлює необхідність використання адекватного математичного апарату для їх дослідження. Мережевий трафік є випадковим процесом, який формується під впливом великої кількості незалежних та залежних факторів, включаючи поведінку користувачів, характеристики протоколів передачі даних, топологію мережі та зовнішні впливи. При цьому слід зазначити, що сучасні мережі є гетерогенними системами, в яких одночасно функціонують різноманітні типи трафіку - голосовий, відеопотоки, веб-запити, транзакційні дані - кожен з яких має власні статистичні закономірності. У зв’язку з цим для моделювання таких процесів доцільно застосовувати апарат теорії ймовірностей та математичної статистики. Використання детермінованих моделей у даному випадку є недостатнім, оскільки вони не дозволяють адекватно описати випадковий характер надходження пакетів у мережу. Крім того, детерміновані підходи не здатні відображати нелінійні ефекти та непередбачувані стрибки навантаження, що є притаманними для реальних мереж. Саме тому стохастичні моделі є базовим інструментом аналізу мережевого трафіку. Вони дозволяють враховувати випадковість процесів та оцінювати їх характеристики у середньому або з певною ймовірністю. Це особливо важливо при дослідженні аномальних явищ, таких як DDoS-атаки. Таким чином, вибір стохастичного підходу є обґрунтованим як з теоретичної, так і з практичної точки зору.

Однією з ключових характеристик мережевого трафіку є його інтенсивність, яка визначається кількістю пакетів, що надходять у систему за одиницю часу. Цей параметр є випадковою величиною та може змінюватися у широких межах залежно від режиму роботи мережі. Для його опису доцільно використовувати випадкові процеси, зокрема пуассонівські потоки. Пуассонівський процес широко застосовується у теорії телекомунікацій для

моделювання надходження заявок у системи обслуговування. Він характеризується незалежністю подій та стаціонарністю, що у багатьох випадках є прийнятним наближенням реального трафіку. Математично пуассонівський процес визначається єдиним параметром - інтенсивністю λ, яка відповідає середній кількості подій за одиницю часу. Така простота параметризації є значною перевагою при побудові аналітичних моделей. Використання цього підходу дозволяє отримати аналітичні залежності для оцінки навантаження на мережу. Крім того, він значно спрощує математичний аналіз. Слід, однак, зазначити, що реальний трафік може відхилятися від пуассонівської моделі, особливо у випадках самоподібного або фрактального характеру трафіку, що характерно для сучасних широкосмугових мереж. Тому у практичних застосуваннях пуассонівська модель використовується як базовий орієнтир, від якого відраховуються відхилення. Це робить його зручним інструментом для побудови моделей виявлення атак. Таким чином, пуассонівські процеси є важливою складовою математичного апарату дослідження мережевого трафіку.

Іншою важливою характеристикою є розподіл джерел трафіку, який може бути описаний за допомогою ентропійних показників. Ентропія дозволяє оцінити рівень невизначеності або різноманітності джерел трафіку. У нормальному режимі роботи мережі розподіл джерел є відносно стабільним, тоді як під час DDoS-атаки спостерігається різке збільшення кількості унікальних IP-адрес. Це призводить до зміни ентропії, що може бути використано як індикатор аномалії. З математичної точки зору ентропія Шеннона визначається як
$H = \sum p_i \cdot \log_2(p_i)$, де p_i - відносна частота появи і-го джерела. При рівномірному розподілі джерел ентропія досягає максимуму, що відповідає ситуації масованої атаки з великої кількості рівноімовірних вузлів. При концентрації трафіку з обмеженої кількості джерел ентропія знижується. Використання інформаційно-теоретичних показників дозволяє доповнити класичні статистичні методи. Це підвищує точність виявлення атак. Крім того, ентропійний аналіз є нечутливим до окремих шумових коливань. Це робить його ефективним у складних умовах. Таким чином, ентропія є важливим елементом математичної моделі.

Важливу роль у моделюванні відіграють також нормалізовані показники, які дозволяють порівнювати різні характеристики трафіку. Оскільки різні параметри мають різні одиниці вимірювання та масштаби, їх безпосереднє порівняння є некоректним. Наприклад, інтенсивність трафіку вимірюється у пакетах за секунду, тоді як ентропія є безрозмірною величиною в бітах. Для вирішення цієї проблеми застосовується нормалізація, яка полягає у приведенні значень до безрозмірної форми. Найпростішим методом є мінімакс-нормалізація, при якій значення параметра масштабується до діапазону (0, 1) відносно його мінімального та максимального значень у вибірці. Альтернативним підходом є z-нормалізація, яка враховує статистичний розподіл параметра через його середнє та стандартне відхилення. Це дозволяє об'єднувати різні показники в єдину інтегральну характеристику. Такий підхід широко використовується у системах прийняття рішень. Він дозволяє формувати узагальнений критерій оцінки стану системи. Це особливо важливо для задач виявлення аномалій. Таким чином, нормалізація є необхідним етапом побудови математичної моделі.

Для прийняття рішення про наявність атаки доцільно використовувати інтегральні критерії, що враховують декілька параметрів одночасно. Одним із таких підходів є використання зваженої суми показників. У цьому випадку кожному параметру призначається ваговий коефіцієнт, що відображає його важливість. Це дозволяє гнучко налаштовувати модель відповідно до умов експлуатації. Наприклад, у деяких мережах більш важливим є контроль інтенсивності трафіку, тоді як в інших - кількість джерел. Визначення вагових коефіцієнтів може здійснюватись як евристично на основі експертних знань, так і автоматично за допомогою методів машинного навчання. У другому випадку ваги оптимізуються на навчальній вибірці, що містить зразки нормального та аномального трафіку. Використання вагових коефіцієнтів дозволяє врахувати ці особливості. Це робить модель адаптивною. Крім того, інтегральний критерій порівнюється з заздалегідь визначеним порогом: перевищення порогу свідчить про аномалію. Правильний вибір порогу є критичним для балансу між

чутливістю та специфічністю системи виявлення. Таким чином, інтегральні критерії є ефективним інструментом виявлення DDoS-атак .

Окрім цього, важливо враховувати часову динаміку процесів, що відбуваються у мережі. Для цього використовується розбиття часу на інтервали спостереження. У кожному інтервалі обчислюються характеристики трафіку. Це дозволяє відслідковувати зміни у поведінці системи. Вибір тривалості інтервалу спостереження є важливим параметром: занадто короткі інтервали призводять до шумових коливань показників, тоді як занадто довгі знижують оперативність виявлення атаки. Зазвичай рекомендується використовувати інтервали від одиниць до десятків секунд залежно від специфіки мережі. Аналіз часових рядів дозволяє виявляти тренди та різкі зміни. Саме такі зміни є характерними для атак. Для підвищення точності аналізу можуть використовуватись ковзні середні та методи згладжування, які дозволяють відфільтрувати шум і виділити стійкі тенденції. Таким чином, часовий аналіз є важливим компонентом математичної моделі.

Суттєвим доповненням до описаного підходу є застосування методів статистичного тестування для підтвердження або спростування гіпотези про наявність аномалії. Зокрема, можуть застосовуватись критерій χ^2 для перевірки відповідності розподілу трафіку очікуваному, а також непараметричні критерії, наприклад критерій Колмогорова–Смирнова. Ці методи дозволяють оцінити статистичну значущість виявлених відхилень. Крім того, байєсівський підхід дозволяє враховувати апріорну інформацію про ймовірність атак у певні часові проміжки, наприклад у пікові години навантаження або в передсвяткові дні. Поєднання частотних та байєсівських методів дозволяє суттєво підвищити якість класифікації трафіку. Таким чином, статистичне тестування є невід’ємною частиною комплексного математичного апарату.

Таким чином, обраний математичний апарат базується на поєднанні теорії ймовірностей, математичної статистики, інформаційної теорії та методів нормалізації. Це дозволяє комплексно описати поведінку мережевого трафіку. Використання такого підходу забезпечує достатню точність та гнучкість моделі.

Крім того, він є достатньо простим для реалізації у програмному забезпеченні. Важливо підкреслити, що запропонована комбінація методів є масштабованою: вона може бути розширена за рахунок включення додаткових параметрів трафіку або більш складних статистичних моделей без принципової зміни архітектури системи виявлення. Це забезпечує довгострокову придатність підходу в умовах постійного розвитку телекомунікаційних технологій та еволюції методів кібератак. Це робить його практично придатним для використання у системах виявлення DDoS-атак.

2.2. Побудова моделі мережевого трафіка на основі теорії масового обслуговування

Теорія масового обслуговування є одним із фундаментальних напрямів прикладної математики, що широко використовується для аналізу та моделювання телекомунікаційних систем. У межах цієї теорії мережевий трафік розглядається як потік заявок, що надходять до системи обслуговування, де відбувається їх обробка. Такий підхід дозволяє формалізувати процес передачі пакетів у мережі та застосувати строгі математичні методи для оцінки її характеристик. Кожен пакет або запит у мережі інтерпретується як заявка, яка потребує обслуговування сервером або мережевим вузлом. При цьому важливими параметрами є інтенсивність надходження заявок та швидкість їх обслуговування. Варто підкреслити, що апарат теорії масового обслуговування дозволяє не лише оцінювати поточний стан системи, але й прогнозувати її поведінку при зміні вхідних умов, що є особливо цінним при проектуванні захисних механізмів. Теорія масового обслуговування дозволяє визначити ймовірність перевантаження системи, час очікування заявок та інші ключові показники. Це робить її ефективним інструментом для аналізу DDoS-атак. Таким чином, використання даної теорії є обґрунтованим для побудови моделі мережевого трафіку.

Основною моделлю, що використовується для опису мережевого трафіку, є система типу M/M/1 або її узагальнення. У цій моделі перша літера «M» означає, що процес надходження заявок описується пуассонівським потоком,

друга «M» - що час обслуговування має експоненційний розподіл, а «1» - що система має один канал обслуговування. Така модель є достатньо простою, але водночас дозволяє отримати важливі аналітичні результати. Експоненційний розподіл часу обслуговування є математично зручним, оскільки він задовольняє властивість відсутності пам'яті, що суттєво спрощує аналіз стаціонарних характеристик системи. У реальних мережах часто використовуються багатоканальні системи, однак базова модель M/M/1 є зручною для початкового аналізу. Вона дозволяє визначити основні залежності між параметрами системи. Крім того, ця модель є добре дослідженою у науковій літературі, що дозволяє спиратися на значний масив теоретичних результатів. Це дозволяє використовувати вже відомі результати для побудови власної моделі. Таким чином, модель M/M/1 є базовою для дослідження мережевого трафіку.

Ключовими параметрами моделі є інтенсивність надходження заявок λ та інтенсивність обслуговування μ. Інтенсивність λ визначає середню кількість пакетів, що надходять у систему за одиницю часу. Інтенсивність μ характеризує здатність системи обробляти ці пакети. Обидва параметри є визначальними для поведінки системи та можуть змінюватися у широких межах залежно від умов функціонування мережі. Важливим показником є коефіцієнт завантаження системи, який визначається як відношення λ до μ. Якщо цей коефіцієнт менший за одиницю, система працює стабільно. У протилежному випадку система переходить у стан перевантаження, при якому черга заявок необмежено зростає. Це є ключовим моментом для виявлення DDoS-атак. Оскільки під час атаки значення λ різко зростає, система не встигає обробляти всі запити. Таким чином, коефіцієнт завантаження є важливим індикатором стану мережі.

Для формалізації цього показника використовується співвідношення $\rho = \lambda/\mu$, де ρ - коефіцієнт завантаження системи. Якщо $\rho < 1$, система знаходиться у стійкому режимі, при якому черга має скінченну середню довжину. Якщо $\rho \geq 1$, система стає нестійкою, що призводить до накопичення черги заявок. У контексті телекомунікацій це означає перевантаження мережі. Саме така ситуація характерна для DDoS-атак. При цьому слід зауважити, що навіть

значення ρ, наближені до одиниці знизу, є небезпечними: при $\rho \to 1$ середня кількість заявок у черзі та час очікування прямують до нескінченності, що практично означає деградацію обслуговування задовго до досягнення граничного значення. Таким чином, аналіз цього коефіцієнта дозволяє визначити не лише критичні, але й передкритичні стани системи.

Важливим показником у теорії масового обслуговування є також середня кількість заявок у системі. Цей показник дозволяє оцінити рівень завантаженості мережі. У моделі M/M/1 він визначається аналітично через параметри λ та μ за формулою $L = \rho/(1 - \rho)$, де L - середня кількість заявок у системі. Зі зростанням інтенсивності трафіку кількість заявок у системі також зростає. Це призводить до збільшення часу обробки та затримок у мережі. У випадку DDoS-атаки цей процес набуває лавиноподібного характеру: навіть незначне перевищення порогового значення λ призводить до катастрофічного зростання черги. Крім того, середня кількість заявок є безпосередньо пов'язаною з іншими показниками через закон Літтла, згідно з яким $L = \lambda \cdot W$, де W - середній час перебування заявки у системі. Таким чином, аналіз цього показника дозволяє виявити аномальні ситуації.

Ще одним важливим параметром є середній час перебування заявки у системі. Він включає як час обслуговування, так і час очікування у черзі. У нормальному режимі цей час є відносно невеликим і визначається формулою $W = 1/(\mu - \lambda)$. Однак при перевантаженні системи він різко зростає - знаменник $\mu - \lambda$ наближається до нуля, що спричиняє необмежене зростання часу очікування. Це призводить до деградації якості обслуговування. У телекомунікаційних системах це проявляється у вигляді затримок, втрат пакетів та зниження пропускної здатності. Затримки у доставці пакетів безпосередньо впливають на якість надання послуг реального часу, таких як голосовий зв'язок і відеоконференції. Таким чином, цей показник також може бути використаний для виявлення атак .

Для більш точного опису мережевого трафіку можуть використовуватися розширені моделі, такі як M/M/n або M/G/1. У моделі M/M/n враховується

наявність кількох каналів обслуговування. Це більш точно відображає реальні мережеві системи, де існує кілька серверів або процесорів. У моделі M/G/1 допускається довільний розподіл часу обслуговування, що дозволяє врахувати більш складні характеристики трафіку, зокрема пакетну передачу або нерівномірність навантаження. Разом із тим аналіз таких моделей є значно складнішим і потребує чисельних методів або апроксимацій. Тому для цілей кваліфікаційної роботи доцільно використовувати спрощену модель, яка дозволяє зберегти баланс між точністю та складністю аналізу.

У контексті розробленої програмної системи модель масового обслуговування використовується для інтерпретації параметра інтенсивності трафіку. Зокрема, показник Packets/sec у програмі відповідає інтенсивності λ. Базове значення цього параметра відповідає нормальному режиму роботи системи, при якому $\rho < 1$. При моделюванні атаки цей параметр значно зростає, що призводить до перевищення критичного порогу завантаження. Таким чином, програмна реалізація безпосередньо відображає теоретичну модель, а зміна значення λ у симуляторі відтворює реальний ефект атаки на мережеву систему обслуговування.

Крім того, у моделі враховується кількість унікальних джерел трафіку. У теорії масового обслуговування це може інтерпретуватися як кількість незалежних потоків заявок, кожен з яких має власну інтенсивність надходження. За властивістю суперпозиції пуассонівських потоків сукупний трафік від N незалежних джерел з однаковою інтенсивністю λ_0 описується пуассонівським потоком з інтенсивністю $\lambda = N \cdot \lambda_0$. Під час DDoS-атаки кількість таких потоків різко зростає, що призводить до пропорційного зростання сукупного навантаження. Це ускладнює обробку трафіку та підвищує навантаження на систему. Таким чином, цей параметр є додатковим індикатором аномальної активності.

Також у моделі враховується частка SYN-пакетів, яка є важливим показником для виявлення атак типу SYN flood. У нормальному режимі ця частка є відносно стабільною і відповідає природному розподілу типів пакетів у

мережевому трафіку. Однак під час атаки вона різко зростає, що пов’язано з великою кількістю незавершених з’єднань. З точки зору теорії масового обслуговування, незавершені з’єднання можна розглядати як заявки, що займають ресурси системи, але не звільняють їх у нормальний термін, що суттєво знижує ефективну пропускну здатність. Таким чином, аналіз цього параметра дозволяє виявити специфічні типи атак на рівні протоколу TCP.

Ентропія джерел трафіку використовується для оцінки розподілу IP-адрес у потоці заявок. У нормальному режимі вона має певний характерний рівень, що відображає типову різноманітність мережевих вузлів, які звертаються до системи. Під час атаки ентропія може суттєво змінюватися залежно від характеру атаки: при розподіленій атаці з великої кількості різних адрес вона зростає, тоді як при атаці з обмеженого числа джерел - знижується. Це дозволяє виявляти як централізовані, так і розподілені атаки за допомогою єдиного показника. Таким чином, цей параметр доповнює інші характеристики моделі та підвищує її діагностичну здатність.

Важливим аспектом є використання часових інтервалів для аналізу трафіку. У кожному інтервалі обчислюються значення параметрів λ, ρ та похідних показників. Це дозволяє відслідковувати зміни у динаміці системи та виявляти перехідні процеси. Аналіз часових змін дозволяє виявляти різкі стрибки навантаження, характерні для початкової фази DDoS-атаки. Крім того, порівняння характеристик суміжних інтервалів дозволяє оцінювати швидкість зростання навантаження, що є важливим для своєчасного реагування. Саме такі стрибки є характерними для DDoS-атак і відрізняють їх від природних піків трафіку, які зазвичай мають більш плавний характер.

Таким чином, побудована модель мережевого трафіку базується на принципах теорії масового обслуговування та доповнюється статистичними показниками. Вона дозволяє описати як нормальний, так і аномальний режим роботи мережі через зміну ключових параметрів λ, μ та похідних від них величин. Використання цієї моделі забезпечує можливість ефективного виявлення DDoS-атак завдяки чіткій математичній інтерпретації аномалій. Крім того, вона є

достатньо простою для реалізації у програмному забезпеченні без залучення складних чисельних методів. Запропонована модель є також розширюваною: за необхідності вона може бути доповнена більш складними підмоделями для окремих компонентів системи, що робить її перспективною основою для подальших досліджень у галузі захисту телекомунікаційних мереж.

2.3. Формування критеріїв виявлення аномального навантаження та DDoS-атак

Формування критеріїв виявлення аномального навантаження та DDoS-атак є одним із центральних завдань побудови системи захисту телекомунікаційної інфраструктури. Критерій виявлення являє собою формалізоване правило або сукупність правил, що дозволяють розмежувати легітимний мережевий трафік від атакуючого на основі аналізу певних характеристик мережевих потоків. Якість сформованих критеріїв визначає два ключові показники ефективності системи виявлення: частоту хибних спрацьовувань - ситуацій, коли легітимний трафік класифікується як атака, та частоту пропусків атак - ситуацій, коли реальна атака не виявляється системою. Мінімізація обох показників є суперечливим завданням: зниження порогу виявлення зменшує кількість пропущених атак, але одночасно збільшує кількість хибних спрацьовувань, і навпаки. Підхід до формування критеріїв виявлення принципово відрізняється залежно від обраної парадигми детектування. Сигнатурний підхід ґрунтується на порівнянні характеристик трафіку з базою відомих шаблонів атак. Аномалійний підхід базується на побудові моделі нормального трафіку та виявленні відхилень від неї. Поведінковий підхід аналізує патерни взаємодії між вузлами мережі в динаміці. Гібридні підходи поєднують кілька парадигм для досягнення вищої точності при збереженні прийнятної обчислювальної складності. Вибір парадигми безпосередньо визначає характер критеріїв, що формуються, та механізми їх застосування в реальному часі.

Статистичні критерії є фундаментальним інструментом виявлення аномального навантаження та базуються на порівнянні поточних характеристик трафіку з їхніми еталонними значеннями, отриманими в результаті аналізу

нормального функціонування мережі. Ефективність статистичних критеріїв визначається якістю побудованого базового профілю трафіку та адекватністю обраних порогових значень.

Критерій перевищення порогу інтенсивності трафіку є найпростішим і найбільш широко застосовуваним статистичним критерієм. Він визначається як умова:

$$I(t) > I_{\text{базова}} \times k,$$

де *I(t)* - поточна інтенсивність трафіку в момент часу *t* (в пакетах за секунду або бітах за секунду), $I_{\text{базова}}$ - базова інтенсивність для відповідного часового інтервалу доби та дня тижня, *k* - коефіцієнт чутливості, що визначає допустиме відхилення. Базова інтенсивність формується як ковзне середнє значення за аналогічні часові інтервали попередніх тижнів з урахуванням сезонних коливань. Перевагою цього критерію є висока швидкість обчислення та інтерпретованість; недоліком - нездатність виявляти low-and-slow атаки, що не перевищують порогового значення інтенсивності.

Критерій аномальної ентропії адрес джерела базується на властивості розподілу IP-адрес джерела в трафіку. При нормальному функціонуванні мережі трафік надходить від відносно рівномірно розподіленої множини адрес джерела, що характеризується певним рівнем ентропії Шеннона. Під час volumetric-атак, що використовують підроблені адреси джерела, розподіл адрес стає більш рівномірним (ентропія зростає) або навпаки концентрується навколо обмеженої кількості адрес (ентропія знижується) залежно від типу атаки. Ентропійний критерій обчислюється як:

$$H(t) = -\Sigma\, p\_i \times log_2(p\_i),$$

де *p_i* - частка трафіку від *i*-ї унікальної IP-адреси джерела у вікні спостереження. Значне відхилення *H(t)* від базового значення *H_базова* є ознакою аномалії. Цей

критерій є ефективним для виявлення розподілених атак із рандомізованими адресами джерела, однак чутливий до вибору розміру часового вікна спостереження. Критерій аномального співвідношення SYN/ACK-пакетів є специфічним критерієм для виявлення SYN Flood атак. При нормальному функціонуванні мережі кожному SYN-пакету відповідає приблизно один ACK-пакет завершення рукостискання. При SYN Flood кількість SYN-пакетів різко зростає без відповідного зростання кількості ACK-пакетів, оскільки рукостискання не завершуються. Формально критерій визначається як:

$$R_{SYN(t)} = N_{SYN(t)} * N_{ACK(t)} > R_{\text{порог}},$$

де $N_{SYN(t)}$ та $N_{ACK(t)}$ - кількість SYN та ACK-пакетів у вікні спостереження відповідно. Базове значення R_SYN для нормального трафіку становить приблизно 1.0–1.2; значення, що перевищують 2.0–3.0, є характерними ознаками SYN Flood. Критерій є вузькоспеціалізованим, однак надзвичайно точним саме для свого класу атак і практично не дає хибних спрацьовувань.

Критерій аномальної швидкості створення нових потоків (flow creation rate) базується на тому факті, що під час DDoS-атак зі значною кількістю унікальних адрес джерела швидкість появи нових мережевих потоків різко зростає. У системах аналізу NetFlow/IPFIX цей критерій реалізується як перевищення порогу кількості нових записів у таблиці потоків за одиницю часу. Цей критерій є ефективним для виявлення розподілених атак навіть при відносно невеликій інтенсивності трафіку від кожного окремого бота.

Аналіз характеристик окремих пакетів та їхніх сукупностей дозволяє формувати критерії виявлення, що не залежать від абсолютних значень інтенсивності трафіку і тому є ефективними проти атак, що маскуються під нормальне навантаження.

Критерій аномального розподілу розмірів пакетів використовує той факт, що різні типи DDoS-атак характеризуються специфічними розподілами довжини пакетів. UDP Flood, що використовує пакети фіксованого розміру для

максимізації пропускної здатності, формує вироджений розподіл з концентрацією навколо одного або кількох фіксованих значень. DNS Amplification характеризується аномально великою часткою UDP-пакетів розміром 512–4096 байтів. Відхилення розподілу розмірів пакетів від базового, виміряне за допомогою статистик Колмогорова-Смирнова або відстані Хеллінгера, слугує критерієм виявлення. Перевага цього критерію - незалежність від абсолютної інтенсивності трафіку; недолік - необхідність підтримання актуального базового розподілу для кожного типу сервісу.

Критерій аномальної частки фрагментованих пакетів застосовується для виявлення атак, що використовують навмисну фрагментацію IP-пакетів для обходу засобів захисту. Аномально висока частка фрагментованих пакетів або пакетів із встановленим прапором More Fragments (MF) є ознакою потенційної атаки. Цей критерій формально задається як:

$$F(t) = N_{frag(t)} * N_{total(t)} > F_{\text{порог}},$$

де $N_{frag(t)}$ - кількість фрагментованих пакетів, $N_{total(t)}$ - загальна кількість пакетів у вікні спостереження. Базове значення *F* для типового інтернет-трафіку є досить низьким (менше (1–2)%), тому навіть незначне перевищення порогу є значущим сигналом. Критерій аномальних значень TTL базується на аналізі поля Time to Live заголовка IP. В нормальному трафіку розподіл значень TTL є відносно стабільним і відображає топологію мережі та кількість хопів між відправниками та одержувачами. При атаках із підробленими адресами джерела зловмисник нерідко встановлює фіксоване значення TTL (наприклад, 64 або 128), що призводить до аномальної концентрації певних значень у розподілі TTL. Ентропійний аналіз розподілу TTL є ефективним додатковим критерієм виявлення.

Поведінкові критерії аналізують патерни взаємодії між вузлами мережі в часовій динаміці та дозволяють виявляти атаки, що не виявляються

статистичними критеріями через відсутність явних аномалій в окремих характеристиках трафіку.

Критерій аномальної кількості унікальних адрес призначення від одного джерела (fan-out) є ефективним для виявлення сканування та розвідки, що передує DDoS-атаці. Легітимний хост у типових умовах взаємодіє з обмеженою кількістю адрес призначення, тоді як заражений бот, що виконує розвідку перед атакою, генерує запити до великої кількості унікальних адрес. Формально:

$$FO(h,t) = |\{dst_{ip}: flow(src = h, dst = dst_{ip}) \in W(t)\}| > FO_{\text{порог}},$$

де *FO(h,t)* - кількість унікальних адрес призначення для хоста *h* у часовому вікні *W(t)*. Порогове значення $FO_{\text{порог}}$визначається індивідуально для кожного типу хостів на основі аналізу базової поведінки. Критерій асиметрії трафіку (traffic asymmetry) аналізує співвідношення вхідного та вихідного трафіку для кожного вузла мережі. При нормальній взаємодії трафік є відносно симетричним: запити та відповіді мають порівнянні розміри. Під час об'ємних DDoS-атак трафік стає різко асиметричним - вхідний трафік у рази перевищує вихідний. Критерій асиметрії:

$$AS(v,t) = \frac{I_{\text{вхідний}(v,t)}}{I_{\text{вихідний}(v,t)}} > AS_{\text{порог}}.$$

Цей критерій є особливо ефективним на рівні аналізу трафіку конкретних серверів та сервісів.

Критерій кореляції між вузлами ботнету використовує той факт, що боти одного ботнету отримують команди одночасно і тому починають генерувати атакуючий трафік синхронно. Виявлення груп хостів, що одночасно починають надсилати трафік до однієї адреси призначення, є ознакою координованої DDoS-атаки. Цей критерій реалізується через кластерний аналіз часових рядів

активності хостів і є одним із найбільш складних у обчислювальному відношенні.

Критерій аномальної тривалості сесій застосовується для виявлення low-and-slow атак типу Slowloris. При нормальній роботі HTTP-сесії завершуються протягом секунд або десятків секунд. Аномально велика кількість сесій із тривалістю, що суттєво перевищує базовий розподіл, є ознакою атаки. Формально:

$$LS(t) = |\{session: duration(session) > T\_\text{порог}\}|/N\_sessions(t) > LS\_\text{порог},$$

де $T\_порог$ - порогова тривалість сесії (наприклад, 120 секунд для HTTP). Цей критерій є відносно нечутливим до обсягів трафіку і здатний виявляти атаки, що генерують мінімальний трафік.

Застосування методів машинного навчання дозволяє формувати критерії виявлення, що автоматично адаптуються до змін нормального трафіку та здатні виявляти нові типи атак без явного задання правил. Критерії на основі автокодувальників (autoencoders) формуються шляхом навчання нейромережевої моделі відновлення нормального трафіку. Після навчання на даних нормального трафіку автокодувальник характеризується низькою помилкою відновлення для легітимного трафіку та високою - для аномального. Критерій виявлення:

$$RE(x) = ||x - decoder(encoder(x))||^2 > RE_{\text{порог}},$$

де x - вектор ознак поточного часового вікна трафіку, $RE_{\text{порог}}$- поріг, що визначається за відсотком аномалій, допустимих у базовому трафіку. Перевагою автокодувальників є здатність виявляти невідомі раніше типи атак; недоліком - складність інтерпретації результатів та чутливість до якості навчальних даних. Критерії на основі методу опорних векторів (One-Class SVM) будуються шляхом навчання моделі, що описує «нормальну» область у просторі ознак трафіку та

класифікує всі відхилення від неї як аномалії. One-Class SVM є ефективним для виявлення аномалій при відсутності або обмеженій кількості прикладів атак у навчальних даних. Критерій визначається знаком рішення функції SVM: від'ємне значення відповідає аномалії.

Критерії на основі Random Forest та градієнтного бустингу є ефективними при наявності збалансованих навчальних наборів із прикладами як нормального трафіку, так і різних типів атак. Класифікатори ансамблевих методів демонструють високу точність розмежування та хорошу стійкість до перенавчання. Важливою перевагою Random Forest є можливість аналізу значущості ознак (feature importance), що дозволяє ідентифікувати найбільш інформативні характеристики трафіку для конкретного набору атак і використати цю інформацію для спрощення критеріїв виявлення.

Коректне визначення порогових значень для кожного з сформованих критеріїв є не менш важливим завданням, ніж вибір самих критеріїв. Статично визначені порогові значення є неефективними через значну добову та тижневу циклічність мережевого трафіку: поріг, адекватний для нічного часу, призведе до масових хибних спрацьовувань у пікові години, і навпаки. Адаптивне налаштування порогів реалізується через механізм ковзного базового профілю: порогові значення обчислюються динамічно на основі статистики трафіку за аналогічні часові інтервали попередніх *N* тижнів. При цьому базове значення *B(t)* для моменту часу *t* обчислюється як:

$$B(t) = \mu(t) + k \times \sigma(t),$$

де $\mu(t)$ та $\sigma(t)$ - середнє значення та стандартне відхилення відповідної характеристики для часового інтервалу, що відповідає *t* у попередніх *N* тижнях, *k* - коефіцієнт чутливості, що визначає компроміс між чутливістю та специфічністю критерію. Типові значення *k* варіюються від 2 до 5 залежно від вимог до системи виявлення та критичності захищуваного ресурсу. Механізм захисту від «отруєння» базового профілю є важливим елементом адаптивної

системи порогів. Якщо атака триває тривалий час, вона може поступово включитися в базовий профіль, знижуючи чутливість критеріїв. Для протидії цьому явищу застосовуються: верхнє обмеження значень, що включаються до профілю; вагові коефіцієнти, що знижують вплив аномальних значень; та ручна верифікація базового профілю після виявлених інцидентів.

На практиці жоден окремий критерій не забезпечує прийнятного рівня точності виявлення для всіх типів DDoS-атак. Ефективна система виявлення будується на агрегації множини часткових критеріїв у комплексний показник аномальності.

Метод зваженого голосування передбачає присвоєння кожному критерію вагового коефіцієнта, що відображає його відносну значущість та достовірність для конкретного типу трафіку. Комплексний показник аномальності обчислюється як:

$$A(t) = \Sigma\, w_i \times c_{i(t)},$$

де $c_{i(t)}$- бінарне або безперервне значення *i*-го критерію, *w_i* - ваговий коефіцієнт. Рішення про виявлення атаки приймається при *A(t) > A_поріг*. Вагові коефіцієнти налаштовуються на основі аналізу ефективності окремих критеріїв на навчальній вибірці.

Метод каскадної класифікації організовує критерії у ієрархічну структуру: швидкі та прості критерії першого рівня відбирають підозрілий трафік для детальнішого аналізу складнішими критеріями другого рівня. Такий підхід дозволяє знизити обчислювальне навантаження системи виявлення, застосовуючи ресурсомісткі критерії лише до відфільтрованого підозрілого трафіку. Таким чином, формування критеріїв виявлення аномального навантаження та DDoS-атак є багатовимірним завданням, що потребує врахування типів атак, характеристик захищуваної інфраструктури, обчислювальних обмежень системи виявлення та вимог до балансу між чутливістю та специфічністю. Комплексний підхід, що поєднує статистичні,

поведінкові критерії та методи машинного навчання з адаптивним налаштуванням порогів, забезпечує найвищу ефективність виявлення при прийнятному рівні хибних спрацьовувань.

РОЗДІЛ 3 Програмна реалізація та дослідження математичної моделі

3.1. Розробка програмного засобу моделювання мережевого трафіка

У межах виконання кваліфікаційної роботи було розроблено програмний засіб моделювання мережевого трафіку та виявлення DDoS-атак у телекомунікаційних системах і мережах. Програмна реалізація виконана у вигляді desktop-застосунку для операційної системи Windows із використанням мови програмування C# та технології WPF. Вибір такого середовища розробки обумовлений можливістю створення зручного графічного інтерфейсу користувача, простотою інтеграції математичних обчислень і наочним відображенням результатів моделювання. Розроблена програма дозволяє не лише виконувати обчислення, а й демонструвати поведінку системи в різних режимах роботи мережі. Це є особливо важливим для дослідження DDoS-атак, оскільки дає змогу наочно спостерігати за переходом системи від нормального стану до стану аномального перевантаження. Функціонально програмний засіб складається з кількох основних модулів. Перший модуль призначений для генерації модельованого мережевого трафіку. У межах цього модуля передбачено кілька сценаріїв, зокрема нормальний трафік, підозрілу активність, атакуючий трафік та ручове покрокове моделювання. Другий модуль виконує обчислення параметрів математичної моделі. Саме в ньому визначаються основні характеристики мережевого потоку: інтенсивність трафіку, кількість унікальних джерел, частка SYN-пакетів, ентропія джерел і підсумковий коефіцієнт аномальності. Третій модуль реалізує механізм прийняття рішення щодо поточного стану мережі. Результатом його роботи є віднесення поточного інтервалу часу до одного з класів: норма, підозріла активність або DDoS-атака. Четвертий модуль забезпечує візуалізацію результатів у вигляді графіків, таблиці спостережень і журналу подій.

Інтерфейс програми побудовано таким чином, щоб користувач міг швидко змінювати параметри моделювання та одразу спостерігати за результатами. У лівій частині вікна розміщено блок керування сценаріями трафіку. Тут користувач може вибирати тип трафіку, запускати або зупиняти моделювання,

виконувати окремий крок аналізу, скидати результати або використовувати заздалегідь визначені пресети. Центральна частина інтерфейсу призначена для введення параметрів математичної моделі. У ній задаються вагові коефіцієнти W1, W2, W3, W4, пороги виявлення T1 і T2, а також базові значення показників нормального стану мережі. Права частина вікна використовується для виведення поточних числових характеристик та журналу подій. Нижня область містить графіки та таблицю спостережень.

Основою математичної частини програмної реалізації є інтегральний показник аномальності A, який обчислюється як зважена сума нормалізованих параметрів трафіку. У програмі враховуються чотири основні показники: Packets/sec, Unique IP, SYN ratio та Entropy. Кожен із них нормалізується відносно базового значення нормального режиму. Після цього значення множаться на відповідні вагові коефіцієнти, а отримані результати підсумовуються. Такий підхід дозволяє звести різнорідні характеристики до єдиного числового критерію. Саме на його основі виконується подальша класифікація стану мережі. Прийняття рішення в програмі здійснюється за пороговим принципом. Якщо значення коефіцієнта аномальності A є меншим за T1, система вважається такою, що працює у нормальному режимі. Якщо значення A перевищує T1, але залишається меншим за T2, фіксується підозріла активність. Якщо ж значення A дорівнює або перевищує T2, система класифікує поточний стан як DDoS-атаку. Така схема є достатньо простою для реалізації, але водночас наочною та ефективною для демонстрації принципу роботи математичної моделі. Її перевагою є можливість гнучкого налаштування чутливості системи.

Окрему увагу в програмній реалізації приділено візуалізації. Графік інтенсивності трафіку дозволяє спостерігати зміну Packets/sec у часі. Графік коефіцієнта аномальності A відображає момент наближення до критичних порогів і перехід до небезпечного стану. Таблиця спостережень зберігає значення всіх параметрів для кожного інтервалу моделювання. Журнал подій

фіксує основні дії користувача та зміни стану системи. Такий підхід забезпечує не лише функціональність, а й зручність для подальшого аналізу результатів.

Для збереження результатів у програмі реалізовано експорт у формат CSV. Це дає змогу використовувати отримані дані в табличних редакторах для побудови додаткових графіків, проведення статистичного аналізу та оформлення результатів у тексті кваліфікаційної роботи. Наявність механізму експорту підвищує практичну цінність розробленого програмного засобу. У цілому створений програмний засіб реалізує всі основні етапи дослідження: генерацію трафіку, обчислення параметрів моделі, класифікацію режиму роботи мережі, візуалізацію результатів і збереження даних. Це дозволяє розглядати його як завершений демонстраційно-дослідницький комплекс для вивчення DDoS-атак, рис.3.1.

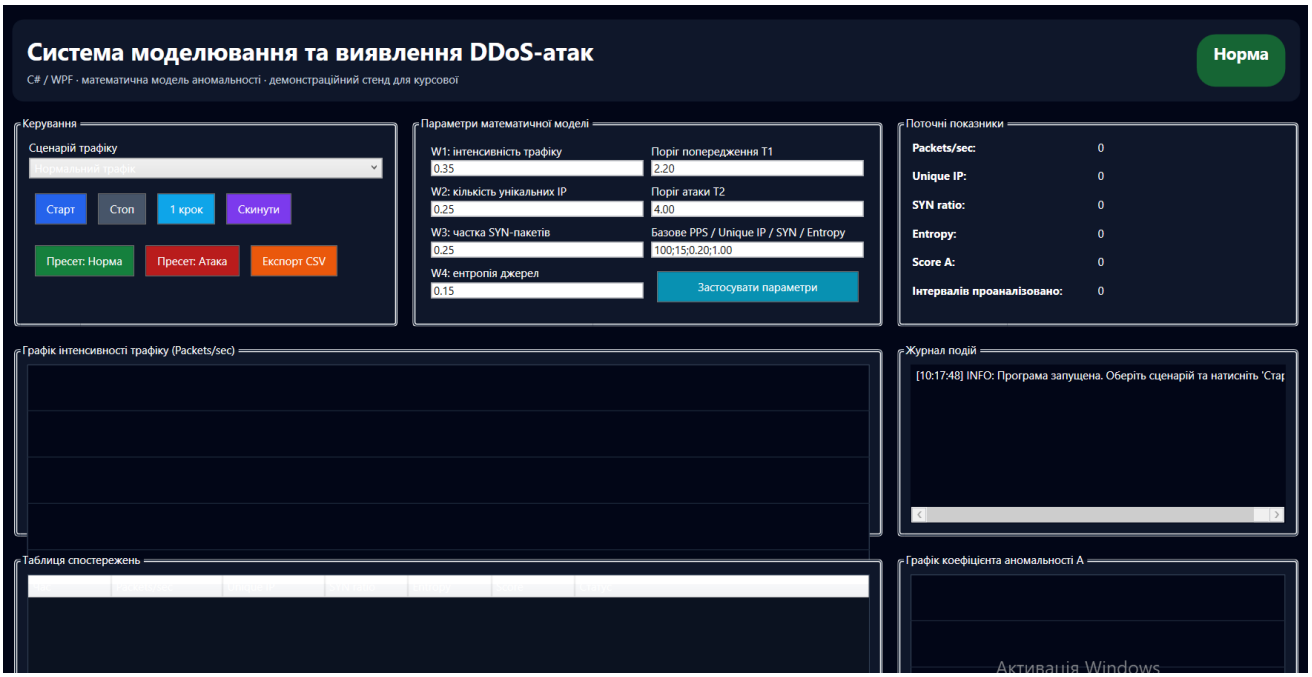

Рис. 3.1 - Головне вікно програми

Окремо винесено блок параметрів математичної моделі, який призначений для налаштування вагових коефіцієнтів вхідних показників (інтенсивність трафіку, кількість унікальних IP-адрес, частка SYN-пакетів, ентропія джерел), а також порогових значень для виявлення попередження та атаки. Така структуризація інтерфейсу забезпечує зручність керування моделлю та дозволяє

оперативно змінювати її параметри без втручання в основну логіку роботи системи, рис.3.2.

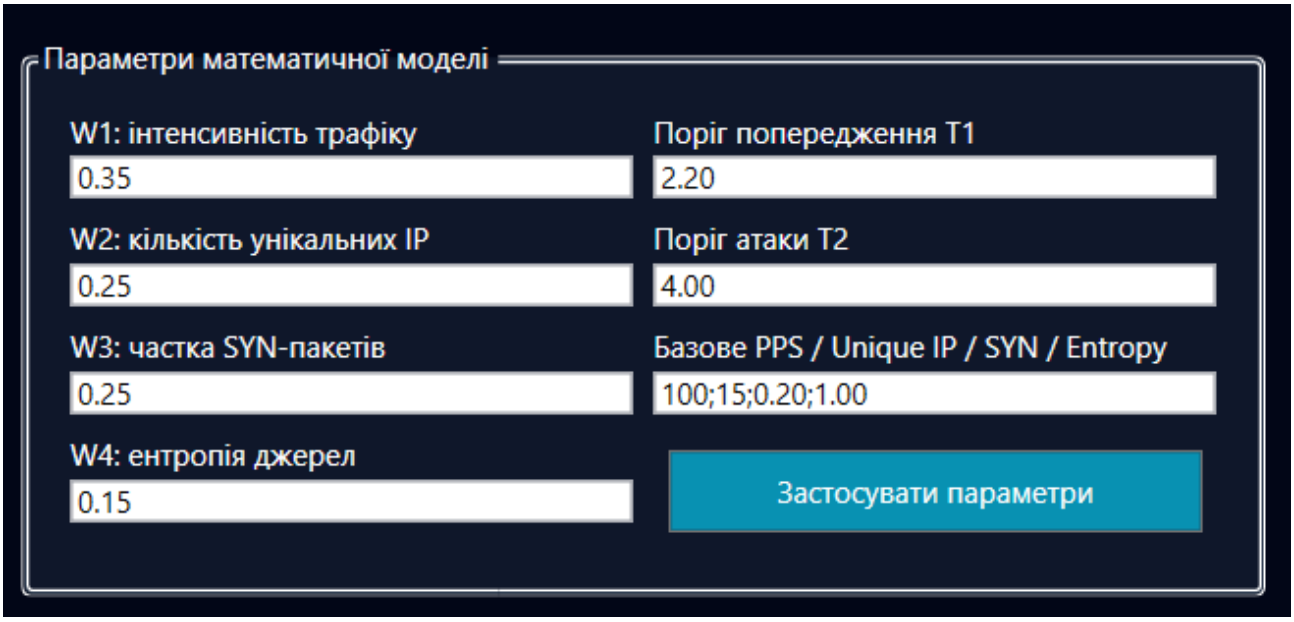

Рис. 3.2 - Блок параметрів математичної моделі

Окремо винесено блок журналу подій, який призначений для фіксації та відображення всіх суттєвих змін у роботі системи. У даному блоці зберігається інформація про виявлені аномалії, спрацювання порогових значень, попередження та зафіксовані атаки, а також час їх виникнення. Це дозволяє здійснювати подальший аналіз роботи моделі, відстежувати динаміку подій і підвищувати ефективність системи шляхом коригування її параметрів, рис.3.3.

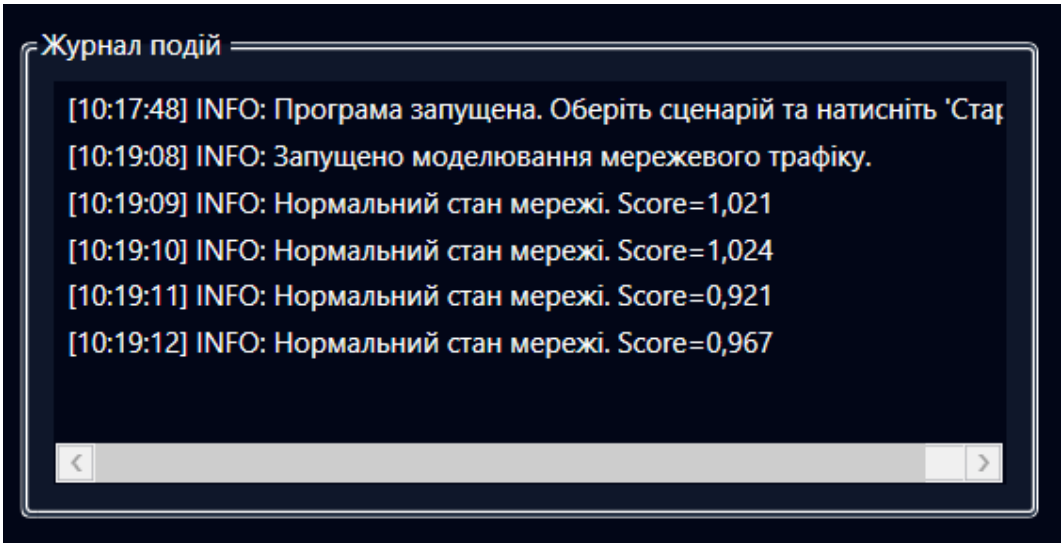

Рис. 3.3 - Журнал подій

Фрагмент програмного коду обчислення коефіцієнта аномальності

```
        snapshot.Score = Math.Round(
    W1 * ppsPart +
    W2 * uniquePart +
    W3 * synPart +
    W4 * entropyPart,
    3);

if (snapshot.Score >= AttackThreshold)
{
    snapshot.Status = "DDoS-атака";
}
else if (snapshot.Score >= WarningThreshold)
{
    snapshot.Status = "Підозріла активність";
}
else
{
    snapshot.Status = "Норма";
}
```

Таким чином, як бачимо, у процесі розробки було створено зручний та інтуїтивно зрозумілий графічний інтерфейс користувача, який забезпечує ефективну взаємодію з програмним засобом. Інтерфейс дозволяє користувачу легко керувати процесом моделювання, змінювати параметри математичної моделі та оперативно отримувати результати аналізу. Завдяки наочному відображенню даних у вигляді графіків, таблиць і журналу подій забезпечується можливість швидкої оцінки стану мережі та виявлення аномальних ситуацій.

Реалізований програмний засіб поєднує у собі функції моделювання, аналізу та візуалізації, що робить його зручним інструментом для дослідження DDoS-атак. У цілому розроблена система відповідає поставленим вимогам та може бути використана як демонстраційний і дослідницький засіб для аналізу мережевого трафіку.

3.2. Проведення експериментального моделювання різних режимів роботи мережі

Для перевірки працездатності програмного засобу та оцінки адекватності математичної моделі було проведено серію експериментів у різних режимах роботи мережі. Основною метою експериментального моделювання було визначення реакції системи на зміну параметрів трафіку, а також перевірка здатності моделі своєчасно виявляти аномальні стани та DDoS-атаки. Дослідження проводилося в умовах імітаційного моделювання, де значення параметрів змінювалися відповідно до обраного сценарію. У межах дослідження було розглянуто три основні режими роботи. Перший режим відповідав нормальному функціонуванню мережі. У цьому випадку інтенсивність трафіку залишалася в межах базових значень, кількість унікальних IP-адрес була невеликою, частка SYN-пакетів не виходила за типові межі, а коефіцієнт аномальності залишався нижчим за поріг T1. Такий режим використано як еталонний для подальшого порівняння.

Другий режим моделював стан підозрілої активності. Для нього характерним було помірне зростання інтенсивності трафіку та числа джерел. У цьому випадку значення коефіцієнта A перевищувало поріг попередження T1, але не досягало рівня T2. Це дозволяло системі ідентифікувати проміжний стан, коли навантаження вже є нетиповим, однак ще не свідчить про повноцінну DDoS-атаку. Даний режим є важливим, оскільки в реальних умовах саме раннє виявлення підозрілої активності може дозволити запобігти переходу до критичного стану.

Третій режим відповідав сценарію DDoS-атаки. У ньому значення Packets/sec суттєво зростало, кількість унікальних IP-адрес різко збільшувалася,

частка SYN-пакетів ставала аномально високою, а ентропія джерел змінювалася відповідно до розподілу джерел атаки. У результаті цього коефіцієнт аномальності перевищував поріг T2, а система класифікувала стан мережі як DDoS-атаку. Саме цей режим став основним у дослідженні ефективності моделі.

Експериментальне моделювання доцільно проводити шляхом поетапного запуску кожного сценарію з фіксацією результатів у таблиці. Для кожного режиму бажано записати середні значення Packets/sec, Unique IP, SYN ratio, Entropy та Score A. На основі цих даних можна побудувати порівняльну таблицю, яка наочно продемонструє зміну параметрів при переході від нормального стану до атаки.

Експеримент 1 - Нормальний режим. Вибрано зліва в меню Нормальний трафік, і програма попрацювала 15–20 кроків, рис.3.4.

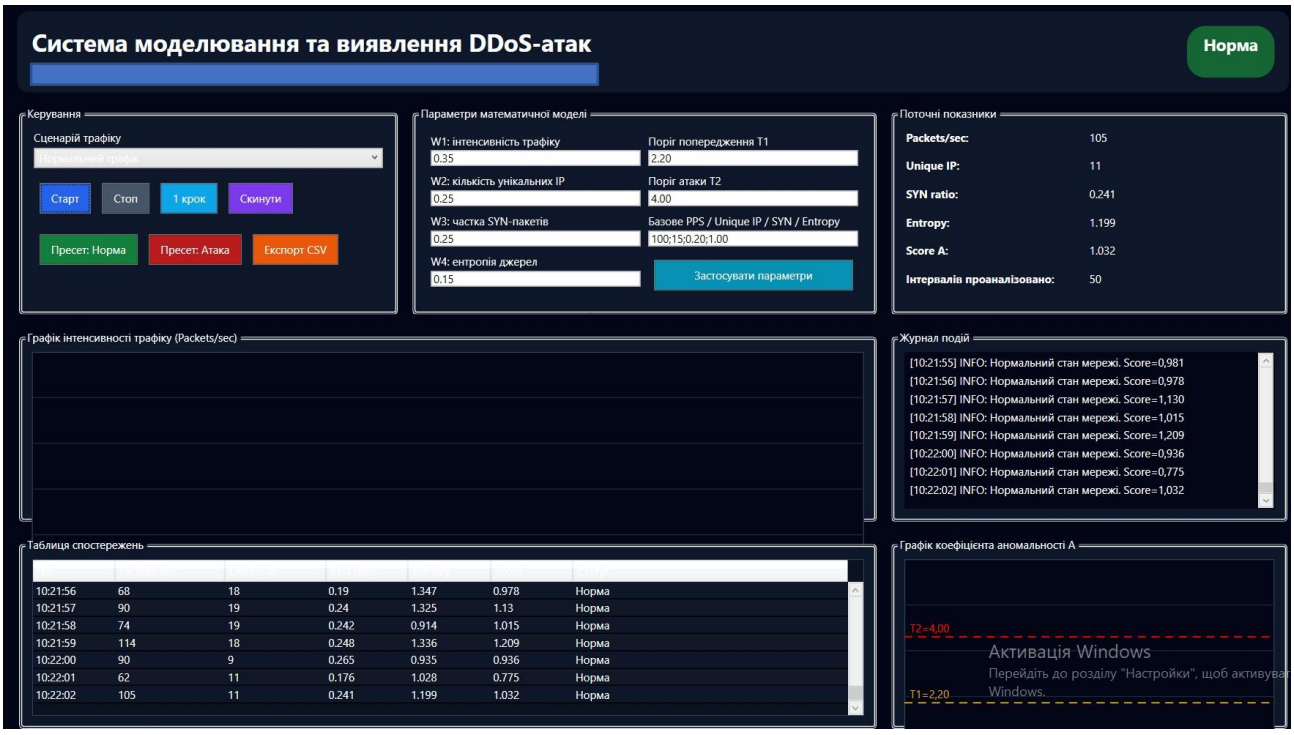

Рис. 3.4. Результати роботи Нормального трафіку

Експеримент 2 - DDoS-атака. Натиснуто кнопку Атака і зразу бачимо результат роботи програми та її зміни, рис.3.5.

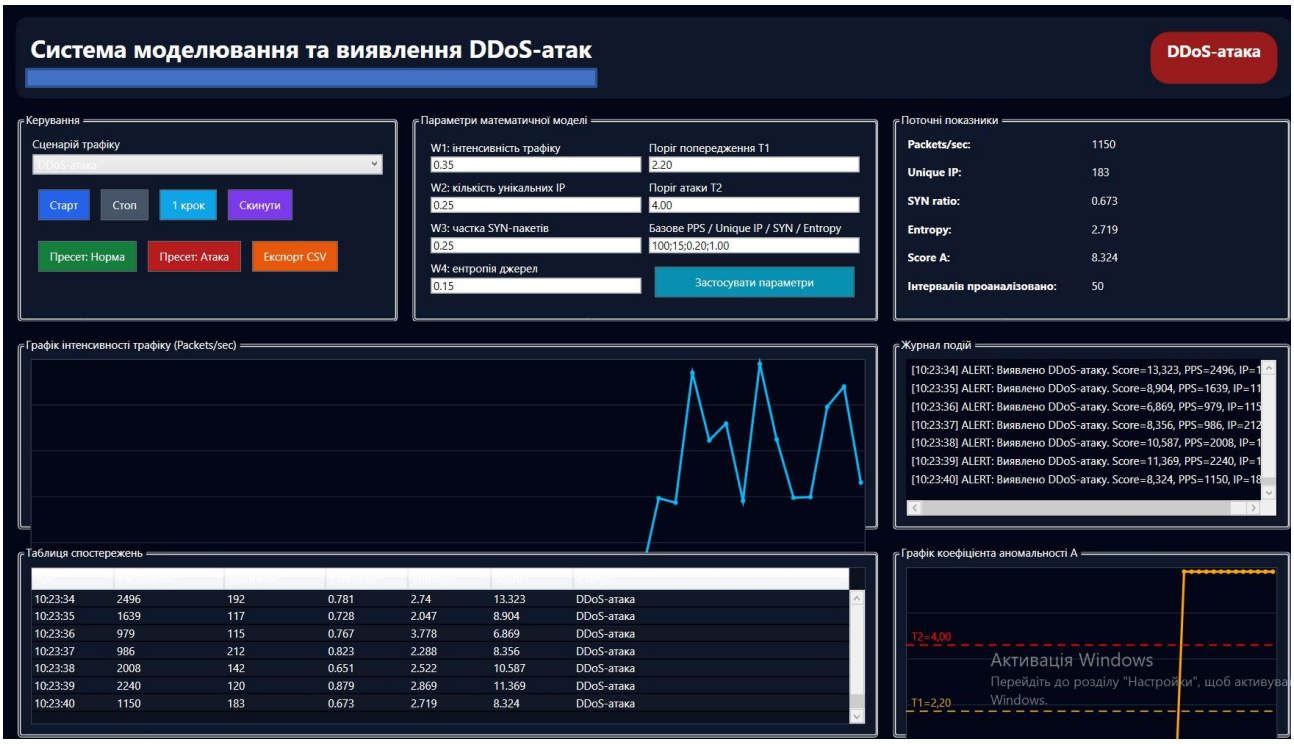

Рис.3.5. DDoS-атака

Для перевірки працездатності розробленого програмного засобу та оцінки ефективності запропонованої математичної моделі було проведено експериментальне моделювання різних режимів роботи мережі. Дослідження виконувалося шляхом запуску програми в кількох сценаріях, зокрема у режимі нормального трафіку, підозрілої активності та DDoS-атаки. У кожному з режимів здійснювалося спостереження за основними параметрами мережевого трафіку, такими як інтенсивність надходження пакетів, кількість унікальних джерел, частка SYN-пакетів, ентропія джерел, а також інтегральний коефіцієнт аномальності. Отримані значення фіксувалися та узагальнювалися для подальшого аналізу. Результати проведеного моделювання наведено в таблиці 3.1.

Таблиця 3.1

Результати моделювання різних режимів роботи мережі

Режим роботи	Packets/sec	Unique IP	SYN ratio	Entropy	Score A	Стан системи
Нормальний трафік	102	14	0.21	0.97	1.18	Норма
Підозріла активність	228	29	0.39	1.32	2.94	Підозріла активність
DDoS-атака	864	96	0.82	2.41	7.36	DDoS-атака

Як видно з наведених у таблиці 3.1 результатів, у нормальному режимі всі параметри знаходяться в межах базових значень, а коефіцієнт аномальності не перевищує порогового значення T1, що підтверджує коректну роботу моделі. У режимі підозрілої активності спостерігається зростання значень основних параметрів, що призводить до перевищення порогу попередження, однак ще не досягає рівня DDoS-атаки. У випадку моделювання DDoS-атаки всі показники різко зростають, що призводить до значного перевищення порогу T2 та однозначної ідентифікації атаки. Це свідчить про здатність моделі ефективно розрізняти різні стани мережі.

3.3. Аналіз результатів моделювання та оцінка ефективності запропонованої моделі

Аналіз результатів проведеного експериментального моделювання показав, що розроблена математична модель є придатною для виявлення аномального навантаження та DDoS-атак у телекомунікаційних системах. У нормальному режимі роботи всі досліджувані показники залишалися в межах базових значень, а інтегральний коефіцієнт аномальності не перевищував порогу попередження. Це свідчить про коректну поведінку моделі в умовах штатного функціонування мережі. У режимі підозрілої активності спостерігалося зростання значень Packets/sec, Unique IP та SYN ratio, що призводило до підвищення значення інтегрального коефіцієнта A. При цьому система не відносила такий стан до атаки одразу, а формувала попередження про підозрілу

активність. Такий результат є позитивним, оскільки вказує на здатність моделі реагувати не лише на критичні, а й на проміжні небезпечні стани.

У режимі DDoS-атаки всі основні параметри змінювалися найбільш суттєво. Значення інтенсивності трафіку та кількості унікальних джерел різко збільшувалися, а частка SYN-пакетів досягала високих значень. Це призводило до впевненого перевищення порога T2. У результаті система стабільно класифікувала стан як DDoS-атаку. Така поведінка свідчить про адекватність обраного математичного апарату та його практичну придатність для задач виявлення атак.

Ефективність запропонованої моделі зумовлена використанням декількох параметрів одночасно. На відміну від підходів, що аналізують лише інтенсивність трафіку, розроблена модель враховує також кількість джерел, SYN ratio та ентропію. Завдяки цьому знижується ризик хибної інтерпретації короткочасних піків трафіку як атаки. Комплексний підхід робить систему більш стійкою до випадкових коливань і підвищує достовірність прийнятого рішення.

До переваг розробленої моделі слід віднести простоту математичної реалізації, наочність, можливість налаштування порогів і вагових коефіцієнтів, а також придатність до програмної реалізації в режимі реального часу. Окрім цього, модель легко масштабувати та адаптувати до інших типів мережевих систем. Вона може бути використана як основа для подальшого ускладнення, наприклад, шляхом інтеграції методів машинного навчання або аналізу реального трафіку. Разом із тим модель має певні обмеження. Вона базується на імітаційному, а не реальному мережевому трафіку. Крім того, її ефективність залежить від правильного вибору базових значень і порогових коефіцієнтів. У майбутньому доцільно провести тестування на реальних мережевих даних і виконати порівняння з іншими методами виявлення аномалій.

Отже, результати моделювання підтверджують, що розроблений програмний засіб та реалізована в ньому математична модель дозволяють ефективно виявляти аномальне навантаження і DDoS-атаки. Це дає підстави вважати поставлену мету курсової роботи досягнутою.

ВИСНОВКИ

У результаті виконання курсової роботи було досліджено проблему виявлення DDoS-атак у телекомунікаційних системах і мережах, яка є однією з актуальних задач сучасної інформаційної безпеки. Проведений аналіз показав, що зростання обсягів мережевого трафіку та ускладнення структури мереж призводять до підвищення вразливості систем до розподілених атак типу відмови в обслуговуванні. У зв’язку з цим виникає необхідність розробки ефективних методів виявлення аномального навантаження на ранніх етапах.

У роботі було обґрунтовано вибір математичного апарату для моделювання мережевого трафіку. Показано, що для опису поведінки мережевих процесів доцільно використовувати методи теорії ймовірностей, математичної статистики та інформаційної теорії. Обґрунтовано застосування стохастичних моделей, зокрема пуассонівських потоків, для опису процесу надходження пакетів у мережі. Також було доведено доцільність використання ентропійних характеристик для аналізу розподілу джерел трафіку.

У межах роботи було побудовано математичну модель мережевого трафіку на основі теорії масового обслуговування. Запропоновано інтерпретацію мережевих пакетів як заявок у системі обслуговування, що дозволило використовувати класичні моделі типу M/M/1 для аналізу навантаження. Введено ключові параметри моделі, зокрема інтенсивність трафіку, кількість джерел, частку SYN-пакетів та ентропію. Показано, що перевищення допустимого рівня навантаження призводить до нестійкого стану системи, що є характерною ознакою DDoS-атаки. На основі проведеного аналізу було сформовано критерій виявлення аномального навантаження, який базується на обчисленні інтегрального коефіцієнта аномальності. Запропонований показник враховує декілька параметрів мережевого трафіку одночасно та обчислюється як зважена сума нормалізованих значень. Для класифікації стану мережі введено два порогових значення, що дозволяють розрізняти нормальний режим,

підозрілу активність і DDoS-атаку. Такий підхід забезпечує гнучкість і адаптивність моделі.

У практичній частині роботи було реалізовано програмний засіб моделювання мережевого трафіку та виявлення DDoS-атак. Програма розроблена у вигляді desktop-застосунку з використанням мови програмування C# та технології WPF. Реалізований інтерфейс забезпечує зручне керування параметрами моделі, запуск різних сценаріїв трафіку та наочне відображення результатів у вигляді графіків, таблиць і журналу подій. Це дозволяє використовувати програму як демонстраційний і дослідницький інструмент.

У процесі експериментального моделювання було досліджено поведінку системи в різних режимах роботи мережі. Отримані результати показали, що у нормальному режимі значення коефіцієнта аномальності залишаються в межах допустимих значень. У режимі підозрілої активності спостерігається поступове зростання показників, що дозволяє виявити потенційно небезпечний стан. У режимі DDoS-атаки всі параметри різко збільшуються, що призводить до перевищення критичного порогу та впевненого виявлення атаки. Це підтверджує адекватність і працездатність запропонованої моделі. Проведений аналіз результатів показав, що використання комплексного підходу, який враховує декілька параметрів трафіку, дозволяє підвищити точність виявлення атак і зменшити кількість хибних спрацьовувань. Запропонована модель є достатньо простою для реалізації, але водночас ефективною для вирішення поставленої задачі. Її перевагою є можливість налаштування вагових коефіцієнтів і порогових значень відповідно до умов конкретної мережі.

Разом із тим слід зазначити, що розроблена модель має певні обмеження, зокрема використання імітаційного трафіку та залежність від заданих параметрів. У подальших дослідженнях доцільно розглянути можливість застосування реальних мережевих даних, а також інтеграції методів машинного навчання для підвищення точності виявлення атак.

Отже, у результаті виконання курсової роботи поставлена мета досягнута: розроблено математичну модель виявлення DDoS-атак та реалізовано

програмний засіб для її дослідження. Отримані результати підтверджують ефективність запропонованого підходу та можливість його практичного застосування у задачах аналізу мережевого трафіку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Рижаков М., Поночовний П. Модель трансформації на основі штучного інтелекту з елементами захисту від DDoS-атак. Прикладні системи управління та інформаційні технології. 2025.

2. Савченко В. А., Степанченко Б. С. Розробка концепції прогнозування часу початку DDoS-атаки на основі дослідження динаміки поведінки еволюційних рівнянь. Телекомунікаційні та інформаційні технології. 2024.

3. Субач І. Ю., Фесьоха В. В. Модель виявлення аномалій в інформаційно-телекомунікаційних мережах органів військового управління на основі нечітких множин. Збірник наукових праць Військового інституту телекомунікацій та інформатизації.-2017-.

4. Vavrychen O., Horodyskyi R., Ploshchyk A. Методи захисту інформації в сучасних засобах радіозв'язку. Наука і техніка сьогодні. 2023.- № 13 (27).

5. Opirskyy I., Bybyk R. Дослідження сучасних методів радіоелектронної боротьби та засобів її протидії. Ukrainian Scientific Journal of Information Security.- 2023,- № 29(2).- С. 88–97.

6. Яковлев М., Волобуєв А., Прібилєв Ю. Математичне моделювання процесів функціонування автоматизованих систем військового радіозв'язку. Збірник наукових праць НАНГУ.- 2024, № 1(43). С.- 130–144.

7. Ye T., Kalyanaraman S., Harrison D. Network management and control using collaborative online simulation. Pennsylvania State University. 2000-. , P. 1–9.

8. Thottan M., Ji C. Using network fault predictions to enable IP traffic management. Journal - of Network and Systems Management.- 2001- Vol. 9, No. 3. P. 327–334.

9. Maxion R., Feather F. Ethernet anomalies in distributed computing environments. IEEE Transactions on Reliability.-1990.- Vol. 39.-P. 433–443.

10. Vigna G., Kemmerer R. NetSTAT: A network-based intrusion detection approach. Annual Computer Security Applications Conference. 1998.

11. Yang J., Ning P., Wang X., Jajodia S. CARDS: A distributed system for detecting coordinated attacks. IFIP International Information Security Conference. 2000-. P. 171–180.

12. Wang H., Zhang D., Shin K. Detecting SYN flooding attacks. IEEE INFOCOM. 2002.

13. Hariharan A., Gupta A., Pal T. Cybersecurity autonomous machine learning platform for anomaly detection. Future of Information and Communication Conference. -2020.- P. 705–720.

14. Siddiqui M. A., Stokes J. W., Seifert C. Detecting cyber attacks using anomaly detection with explanations and expert feedback. IEEE ICASSP.- 2019. - P. 2872–2876.

ДОДАТКИ

Основні класи програми
DetectionService.cs

```
using DdosDetectorApp.Models;

namespace DdosDetectorApp.Services;

public class DetectionService
{
    public int BasePacketsPerSecond { get; set; } = 100;
    public int BaseUniqueIpCount { get; set; } = 15;
    public double BaseSynRatio { get; set; } = 0.20;
    public double BaseEntropy { get; set; } = 1.00;

    public double W1 { get; set; } = 0.35;
    public double W2 { get; set; } = 0.25;
    public double W3 { get; set; } = 0.25;
    public double W4 { get; set; } = 0.15;

    public double WarningThreshold { get; set; } = 2.20;
    public double AttackThreshold { get; set; } = 4.00;

    public TrafficSnapshot Analyze(TrafficSnapshot snapshot)
    {
        var ppsPart = snapshot.PacketsPerSecond / (double)BasePacketsPerSecond;
        var uniquePart = snapshot.UniqueIpCount / (double)BaseUniqueIpCount;
        var synPart = snapshot.SynRatio / BaseSynRatio;
        var entropyPart = snapshot.SourceEntropy / BaseEntropy;

        snapshot.Score = Math.Round(
            W1 * ppsPart +
            W2 * uniquePart +
            W3 * synPart +
            W4 * entropyPart,
            3);

        if (snapshot.Score >= AttackThreshold)
        {
```

```
            snapshot.Status = "DDoS-атака";
        }
        else if (snapshot.Score >= WarningThreshold)
        {
            snapshot.Status = "Підозріла активність";
        }
        else
        {
            snapshot.Status = "Норма";
        }

        return snapshot;
    }
}
```

TrafficSnapshot.cs

```
namespace DdosDetectorApp.Models;

public class TrafficSnapshot
{
    public DateTime Timestamp { get; set; }
    public int PacketsPerSecond { get; set; }
    public int UniqueIpCount { get; set; }
    public double SynRatio { get; set; }
    public double SourceEntropy { get; set; }
    public double Score { get; set; }
    public string Status { get; set; } = "Норма";
}
```

Логіка роботи програми

```
using System.Collections.ObjectModel;
using System.Globalization;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media;
using System.Windows.Shapes;
using System.Windows.Threading;
using DdosDetectorApp.Models;
using DdosDetectorApp.Services;
using Microsoft.Win32;

namespace DdosDetectorApp;

public partial class MainWindow : Window
{
    private readonly DispatcherTimer _timer;
    private readonly TrafficSimulator _simulator = new();
    private readonly DetectionService _detector = new();
    private readonly ObservableCollection<TrafficSnapshot> _snapshots = new();
    private readonly ObservableCollection<EventLogEntry> _events = new();

    public MainWindow()
```

```
    {
        InitializeComponent();

        _timer = new DispatcherTimer
        {
            Interval = TimeSpan.FromSeconds(1)
        };
        _timer.Tick += (_, _) => RunSimulationStep();

        SnapshotsDataGrid.ItemsSource = _snapshots;
        EventLogListBox.ItemsSource = _events;

        AddEvent("INFO", "Програма запущена. Оберіть сценарій та натисніть
‘Старт’.");
        UpdateStatusBadge("Норма");
        SizeChanged += (_, _) => RedrawCharts();
    }

    private void StartButton_Click(object sender, RoutedEventArgs e)
    {
        ApplyModelParameters();
        _timer.Start();
        AddEvent("INFO", "Запущено моделювання мережевого трафіку.");
    }

    private void StopButton_Click(object sender, RoutedEventArgs e)
    {
        _timer.Stop();
        AddEvent("INFO", "Моделювання зупинено.");
    }

    private void StepButton_Click(object sender, RoutedEventArgs e)
    {
        ApplyModelParameters();
        RunSimulationStep();
    }

    private void ResetButton_Click(object sender, RoutedEventArgs e)
    {
```

```
        _timer.Stop();
        _snapshots.Clear();
        _events.Clear();
        _simulator.Reset();
        ClearMetrics();
        RedrawCharts();
        AddEvent("INFO", "Дані очищено. Система готова до нового
експерименту.");
        UpdateStatusBadge("Норма");
    }

    private void ApplyModelButton_Click(object sender, RoutedEventArgs e)
    {
        ApplyModelParameters();
        AddEvent("INFO", "Параметри математичної моделі оновлено.");
    }

    private void NormalPresetButton_Click(object sender, RoutedEventArgs e)
    {
        ScenarioComboBox.SelectedIndex = 0;
        AddEvent("INFO", "Обрано пресет нормального трафіку.");
    }

    private void AttackPresetButton_Click(object sender, RoutedEventArgs e)
    {
        ScenarioComboBox.SelectedIndex = 2;
        AddEvent("WARN", "Обрано пресет DDoS-атаки.");
    }

    private void ExportButton_Click(object sender, RoutedEventArgs e)
    {
        var dialog = new SaveFileDialog
        {
            Filter = "CSV files (*.csv)|*.csv",
            FileName = $"ddos-results-{DateTime.Now:yyyyMMdd-HHmmss}.csv"
        };

        if (dialog.ShowDialog() == true)
        {
```

```
            CsvExportService.Export(dialog.FileName, _snapshots);
            AddEvent("INFO", $"Результати експортовано у файл:
{dialog.FileName}");
        }
    }

    private void RunSimulationStep()
    {
        var scenario = GetSelectedScenario();
        var snapshot = _simulator.Next(scenario);
        snapshot = _detector.Analyze(snapshot);

        _snapshots.Add(snapshot);
        if (_snapshots.Count > 50)
        {
            _snapshots.RemoveAt(0);
        }

        UpdateMetrics(snapshot);
        UpdateStatusBadge(snapshot.Status);
        WriteDecisionLog(snapshot);
        SnapshotsDataGrid.ScrollIntoView(snapshot);
        RedrawCharts();
    }

    private void ApplyModelParameters()
    {
        try
        {
            _detector.W1 = ParseDouble(W1Box.Text);
            _detector.W2 = ParseDouble(W2Box.Text);
            _detector.W3 = ParseDouble(W3Box.Text);
            _detector.W4 = ParseDouble(W4Box.Text);
            _detector.WarningThreshold = ParseDouble(WarningBox.Text);
            _detector.AttackThreshold = ParseDouble(AttackBox.Text);

            var parts = BaselineBox.Text.Split(';');
            if (parts.Length != 4)
            {
```

```
                throw new InvalidOperationException("Базова лінія має бути у форматі:
100;15;0.20;1.00");
            }

            _detector.BasePacketsPerSecond = int.Parse(parts[0]);
            _detector.BaseUniqueIpCount = int.Parse(parts[1]);
            _detector.BaseSynRatio = ParseDouble(parts[2]);
            _detector.BaseEntropy = ParseDouble(parts[3]);
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Помилка введення параметрів: {ex.Message}",
"Помилка", MessageBoxButton.OK, MessageBoxImage.Error);
        }
    }

    private static double ParseDouble(string text) => double.Parse(text.Replace(‘,’,
‘.’), CultureInfo.InvariantCulture);

    private TrafficScenario GetSelectedScenario()
    {
        if (ScenarioComboBox.SelectedItem is ComboBoxItem item && item.Tag is
string tag)
        {
            return Enum.Parse<TrafficScenario>(tag);
        }

        return TrafficScenario.Normal;
    }

    private void UpdateMetrics(TrafficSnapshot snapshot)
    {
        PpsText.Text = snapshot.PacketsPerSecond.ToString();
        UniqueIpText.Text = snapshot.UniqueIpCount.ToString();
        SynText.Text = snapshot.SynRatio.ToString("0.000",
CultureInfo.InvariantCulture);
        EntropyText.Text = snapshot.SourceEntropy.ToString("0.000",
CultureInfo.InvariantCulture);
```

```
        ScoreText.Text = snapshot.Score.ToString("0.000",
CultureInfo.InvariantCulture);
        CountText.Text = _snapshots.Count.ToString();
    }

    private void ClearMetrics()
    {
        PpsText.Text = "0";
        UniqueIpText.Text = "0";
        SynText.Text = "0";
        EntropyText.Text = "0";
        ScoreText.Text = "0";
        CountText.Text = "0";
    }

    private void WriteDecisionLog(TrafficSnapshot snapshot)
    {
        switch (snapshot.Status)
        {
            case "DDoS-атака":
                AddEvent("ALERT", $"Виявлено DDoS-атаку.
Score={snapshot.Score:0.000}, PPS={snapshot.PacketsPerSecond},
IP={snapshot.UniqueIpCount}");
                break;
            case "Підозріла активність":
                AddEvent("WARN", $"Зафіксовано аномалію.
Score={snapshot.Score:0.000}");
                break;
            default:
                AddEvent("INFO", $"Нормальний стан мережі.
Score={snapshot.Score:0.000}");
                break;
        }
    }

    private void AddEvent(string level, string message)
    {
        _events.Add(new EventLogEntry
        {
```

```
            Time = DateTime.Now,
            Level = level,
            Message = message
        });

        while (_events.Count > 100)
        {
            _events.RemoveAt(0);
        }

        EventLogListBox.ScrollIntoView(_events.LastOrDefault());
    }

    private void UpdateStatusBadge(string status)
    {
        StatusText.Text = status;
        StatusBadge.Background = status switch
        {
            "DDoS-атака" => new
SolidColorBrush((Color)ColorConverter.ConvertFromString("#991B1B")),
            "Підозріла активність" => new
SolidColorBrush((Color)ColorConverter.ConvertFromString("#92400E")),
            _ => new
SolidColorBrush((Color)ColorConverter.ConvertFromString("#166534"))
        };
    }

    private void RedrawCharts()
    {
        DrawLineChart(TrafficChartCanvas, _snapshots.Select(s =>
(double)s.PacketsPerSecond).ToList(), Brushes.DeepSkyBlue, 2500,
_detector.WarningThreshold, false);
        DrawLineChart(ScoreChartCanvas, _snapshots.Select(s => s.Score).ToList(),
Brushes.Orange, Math.Max(6, _detector.AttackThreshold + 1),
_detector.WarningThreshold, true);
    }

    private void DrawLineChart(Canvas canvas, IList<double> values, Brush
lineBrush, double yMax, double warningLineValue, bool drawAttackLine)
```

```
    {
        canvas.Children.Clear();

        var width = canvas.ActualWidth;
        var height = canvas.ActualHeight;

        if (width < 10 || height < 10)
        {
            width = 600;
            height = 250;
            canvas.Width = width;
            canvas.Height = height;
        }

        var border = new Rectangle
        {
            Width = width,
            Height = height,
            Stroke = new
SolidColorBrush((Color)ColorConverter.ConvertFromString("#334155")),
            StrokeThickness = 1
        };
        canvas.Children.Add(border);

        for (var i = 1; i <= 4; i++)
        {
            var y = height - i * (height / 5.0);
            canvas.Children.Add(new Line
            {
                X1 = 0,
                X2 = width,
                Y1 = y,
                Y2 = y,
                Stroke = new
SolidColorBrush((Color)ColorConverter.ConvertFromString("#1E293B")),
                StrokeThickness = 1
            });
        }
```

```
        if (values.Count == 0)
        {
            return;
        }

        var polyline = new Polyline
        {
            Stroke = lineBrush,
            StrokeThickness = 2.5
        };

        for (var i = 0; i < values.Count; i++)
        {
            var x = values.Count == 1 ? width / 2 : i * (width - 10) / (values.Count - 1.0) +
5;
            var normalized = Math.Clamp(values[i] / yMax, 0, 1);
            var y = height - (normalized * (height - 10)) - 5;
            polyline.Points.Add(new Point(x, y));

            var dot = new Ellipse
            {
                Width = 5,
                Height = 5,
                Fill = lineBrush
            };
            Canvas.SetLeft(dot, x - 2.5);
            Canvas.SetTop(dot, y - 2.5);
            canvas.Children.Add(dot);
        }

        canvas.Children.Add(polyline);

        if (drawAttackLine)
        {
            DrawHorizontalThreshold(canvas, width, height, _detector.AttackThreshold,
yMax, "T2", Brushes.Red);
        }
```

```
        DrawHorizontalThreshold(canvas, width, height, warningLineValue, yMax,
"T1", Brushes.Goldenrod);
    }

    private static void DrawHorizontalThreshold(Canvas canvas, double width, double
height, double value, double yMax, string label, Brush brush)
    {
        var y = height - (Math.Clamp(value / yMax, 0, 1) * (height - 10)) - 5;
        canvas.Children.Add(new Line
        {
            X1 = 0,
            X2 = width,
            Y1 = y,
            Y2 = y,
            Stroke = brush,
            StrokeDashArray = new DoubleCollection { 6, 4 },
            StrokeThickness = 1.5
        });

        var text = new TextBlock
        {
            Text = $"{label}={value:0.00}",
            Foreground = brush,
            Background = new
SolidColorBrush((Color)ColorConverter.ConvertFromString("#020617"))
        };
        Canvas.SetLeft(text, 6);
        Canvas.SetTop(text, Math.Max(0, y - 18));
        canvas.Children.Add(text);
    }
}
```

Інтерфейс користувача