НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

До захисту допущено
Завідувач кафедри

_________ **Дмитро ЛАНДЕ**
(підпис)

«_____»_____________2022 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Системи, технології та математичні методи кібербезпеки»
спеціальності 125 «Кібербезпека»

на тему: Виявлення повільних DDoS атак у TCP потоках за допомогою методів математичної статистики

Виконав (-ла): здобувач вищої освіти **IV** курсу, групи __ФБ-82___
(шифр групи)

Дяковський Кирило Юрійович ____ ____________
(прізвище, ім'я, по батькові) (підпис)

Керівник к.т.н, доцент, Гальчинський Леонід Юрійович __________________ ____________
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові) (підпис)

Рецензент ____ ____________
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ім'я, по батькові) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Здобувач вищої освіти _____________
(підпис)

Київ – 2022 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 125 «Кібербезпека»
Освітньо-професійна програма «Системи, технології та математичні методи кібербезпеки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_________ Дмитро ЛАНДЕ
(підпис)
«___»____________2022 р.

ЗАВДАННЯ
на дипломну роботу здобувачу вищої освіти

________________Дяковський Кирило Юрійович__________
(прізвище, ім’я, по батькові)

1. Тема роботи Виявлення повільних DDoS атак у TCP потоках за допомогою методів математичної статистики ___________________________________
____,
керівник роботи Гальчинський Леонід Юрійович, к.т.н, доцент _________
____,
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «   » _______ 2022 р. №

2. Термін подання здобувачем вищої освіти роботи 14 червня 2022 р.

3. Вихідні дані до роботи тестовий набір мережевих даних з TCP потокам нормального трафіку та під впливом DDoS атаки ______________________
____
____

4. Зміст роботи Теоретичні відомості про DDoS атаки, розпізнавання повільних DDoS атак у TCP потоках, розробка аналізатора повілних DDoS атак в TCP потоках____

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) _____презентація ____

6. Дата видачі завдання 01.12.2021

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Отримання завдання	01.12.2021	Виконано
2	Огляд та опрацювання інформаційних джерел	20.12.2021 -- 30.01.2022	Виконано
3	Аналіз методів детектування повільних DDoS атак	31.01.2022 -- 12.02.2022	Виконано
4	Пошук тестових даних	13.02.2022 -- 15.02.2022	Виконано
5	Аналіз тестових даних	15.02.2022 – 23.02.2022	Виконано
6	Побудова методології для автоматичного визначення факту атаки	10.04.2022 – 19.04.2022	Виконано
7	Побудова методологій до розпізнавання IP адрес атакуючого	19.04.2022 – 30.04.2022	Виконано
8	Розробка і тестування програми для розпізнавання IP адрес атакуючих	03.05.2022 – 19.05.2022	Виконано
9	Модернізація програми	21.05.2022 – 01.06.2022	Виконано
10	Тестування на наборі даних мережі	01.06.2022 – 03.06-2022	Виконано

Здобувач вищої освіти ______________ __Кирило ДЯКОВСЬКИЙ_
(підпис) (**Власне ім'я, ПРІЗВИЩЕ**)

Керівник роботи ______________ _Леонід ГАЛЬЧИНСЬКИЙ_
(підпис) (**Власне ім'я, ПРІЗВИЩЕ**)

РЕФЕРАТ

Обсяг роботи 69 сторінок, 21 ілюстрація, 1 таблиця, 2 додатка, 19 джерел літератури, 6 джерел зображень. В роботі досліджено методологію виявлення IP адрес, з яких ведеться повільні DDoS атака, серед набору даних мережевого трафіку у вигляді TCP потоків. В роботі оглянуто основні загрози хмарних обчислювань, відомості про DoS та DDoS атаки, типи DoS атаки прикладного рівня, типи повільних DoS атак, методи виявлення повільних DoS атак, способи пом’якшення впливу повільних DoS атак, апарат математичної статистики для аналізу і порівняння множин.

Метою дипломної роботи є розробка методики розпізнавання повільних DDoS атак серед набору мережевого трафаку наданого у вигляді TCP потоків.

Об’єктом дослідження дипломної роботи є способи виявлення повільних DDoS атак

Предметом дослідження дипломної роботи є поведінка повільних DDoS атак в мережі та аналіз значень полів, що відрізняють трафік під впливом атаки від звичайного.

Ключові слова: повільна атаки на відмову в обслуговуванні, математична статистика, TCP потоки, набір даних мережі, DoS, DDoS, розподілена атака на відмову в обслуговуванні, прикладна програма.

ABSTRACT

The volume of work is 69 pages, 21 illustrations, 1 table, 2 appendices, 19 sources of literature, and 6 sources of images. The paper investigates the methodology of detecting IP addresses from which slow DDoS attacks are conducted, among the data set of network traffic in the form of TCP streams. The main threats of cloud computing, information about DoS and DDoS attacks, types of DoS attacks of application level, types of slow DoS attacks, methods of detecting slow DoS attacks, and ways to mitigate the effects of slow DoS attacks, mathematical statistics for analyzing and comparing sets.

The thesis aims to develop a method for recognizing slow DDoS attacks among a set of network traffic provided in the form of TCP streams.

The subject of the thesis is ways to detect slow DDoS attacks.

The subject of the thesis is the behavior of slow DDoS attacks in the network and the analysis of field values that distinguish traffic under the influence of the attack from the usual.

Keywords: slow denial of service attack, mathematical statistics, TCP streams, network data set, DoS, DDoS, distributed denial of service attack, application program.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Ботнет – це пов'язані між собою комп'ютери із певним шкідливим програмним забезпеченням, що створюють мережу, яка керуються зловмисником. Зазвичай це велика кількість комп'ютерів, які використовуються для здійснення атаки без відома користуачів.

Модель OSI – теоретична модель мереді, створена для кращої комунікації мережевих протоколів та пологшення їх розробки. Пропонує рівневий вигляд підхід до мережі. Кожен рівень відповідає за свою частину мережевої взаємодії. Завдяки рівневій структурі спільна робота мережевого обладнання й програмного забезпечення стає набагато простішою, прозорішою й зрозумілішою.
У цієї моделі існує 7 рівнів: фізичний, канальльний, мережевий, транспортний, сеансовий, рівень представлення, прикладний рівень.

HTTP (Hypertext Transfer Protocol) — це прикладний протокол, який працює поверх набору протоколів TCP/IP. Уся всесвітня мережа використовує цей протокол. Є одним з найпопулярніших протоколів рівня додатків, які використовуються в Інтернеті.

Заголовки HTTP – є основною частиною цих запитів і відповідей HTTP, що містять інформацію про клієнтський браузер, запитувану сторінку, сервер тощо.

Браузер — програма на стороні клієнта, що виконує HTTP-запит до сервера, а також приймає, інтерпретує та відображає вміст повідомлення-відповіді у вікні браузера відповідно до типу вмісту.

URL (Uniform Resource Locator) — стандартизована адреса певного ресурсу, що знаходиться в інтернеті: документ, зображення тощо. Описується в RFC 3986. Включає в себе назву протоколу доступу (HTTP, FTP, telnet, gopher та ін.) і, шлях до ресурсу, формат якого залежить від схеми доступу.

TCP — це транспортний протокол, що використовується поверх IP для забезпечення надійної передачі пакетів. TCP містить в собі механізми, що вирішують багато проблем, які виникають при пакетному обміні повідомленнями: пошкодження пакету, втрата пакету, дублікат пакету на порушення черги відправлення.

CDN — це географічно розподілені сервери, які забезпечують швидку доставку ресурсів, необхідних для завантаження Інтернет-контенту.

Словник в Python — це невпорядкований набір едементів, в якому кожному ключу відповідає своє значення.

DHCP — протокол динамічної конфігурації вузла — це стандартний протокол прикладного рівня, який надає комп'ютерам пареметри (наприклад IP адреса або шлюз за замовчуванням), необхідні для роботи в мережі. Для цього комп'ютер звертається відповідно до DHCP-сервера.

ВСТУП

Інтернет сьогодні – це платформа для спілкування, співпраці, навчання, роботи, обміну інформацією та дозвілля. Він має також стратегічне значення для оборони країни, для координації військових, медиків, волонтерів. Критична інфраструктура також залежна від інтернету. Цим процесам допомагають і хмарні обчислення, що мінімізують експлуатаційні витрати на центри обробки даних, і програми-сервіси, що розміщуються на серверах, і послуги в інтернеті тощо. Для роботи користувачів із системами, що дозволяють виконувати вищеперераховані активності, потрібно мати доступ до цих систем.

Уряд та підприємства також перенесли свої всі або майже всі обчислювальні потужності у хмарне середовище, через велику кількість переваг у порівнянні з локальною інфраструктурою. Ці переваги включають швидку доступність ресурсів за запитом користувача, зручну оплату рахунків за оренду обладнання, ефективніше його використання, відсутність внутрішніх витрат на амортизацію та обслуговування.[1]

Велику загрозу для хмарних сервісів становлять DDoS атаки. Вони знаходяться на 4 місті по популярності серед кібер атак і з кожним роком їх кількість тільки продовжує зростати.[11]

1 ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО DDoS АТАКИ

1.1 Загрози хмарних обчислень

Національний інститут стандартів і технологій (NIST) визначає, що хмарні обчислення – це модель для забезпечення зручного мережевого доступу на вимогу користувача до загальнодоступних обчислювальних ресурсів, таких як мережі, сервери, сховища, програми та сервери. Їх можна швидко орендувати та почати застосовувати з мінімальними зусиллями на керування та постачання необхідного обладнання.[2]

Модель надання хмарних послуг розділена на три типи: інфраструктура як послуга (IaaS), програмне забезпечення як послуга (SaaS) та платформа як послуга (PaaS).

Модель розгортання заснована на на типі доступу до хмари: публічна, приватна та гібридна.[3]

Хмарні обчислення мають багато потенційних загроз, які класифікуються як:

- Зловживання використанням хмарних обчислювальних ресурсів
- Атаки на дані
    - Шкідливий інсайдер
    - Кібер крадіжка в Інтернеті
- Атаки на безпеку хмар
    - Атаки з ін’єкцію шкідливого програмного забезпечення
    - Атаки в обгортці

Оскільки для доступ до критичних послуг здійснюється із сервера через мережу, необхідно забезпечити доступність цих послуг для легітимного користувача. Для обмеження доступу, зловмисники будуть чинити перешкоди. Прикладом таких перешкод є зловживання використанням хмарних обчислювальних ресурсів атакуючими. Зловмисники переривають або відмовляють у наданні послуг легітимному користувачу за допомогою атаки "Відмова в обслуговування" (DoS).[4]

1.2 Атака на відмову в обслуговуванні

Атака «відмова в обслуговуванні» (DoS) — це тип кібератаки, під час якої зловмисник намагається зробити сервер чи інший пристрій недоступним для користувачів, перериваючи його нормальне функціонування.[5] Атаки DoS, як правило, діють шляхом перевантаження або заповнення цільової машини запитами, поки сервер не втратить змогу обробляти звичайний, що призводить до відмови в обслуговуванні. DoS-атака характеризується використанням одного комп'ютера для запуску атаки. Атака, яка надходить з багатьох джерел називається "розподілена атака на відмову в обслуговуванні" (DDoS)

Під час проведення DoS атаки зловмисник намагається досягти одну або обидві із наступних цілей:

- Порушити підключення законного користувача через виснаження пропускної здатності, потужностей обробки маршрутизатора або мережевих ресурсів
- Порушити надання послуг законному користувачу, виснаживши ресурси сервера (наприклад: сокети, ЦП, пам'ять, пропускну здатність диска/бази даних та пропускну здатність вводу/виводу). Вони включають атаки флудінгу на прикладному рівні (application-level flooding attacks)

Для досягнення мети зловмисник зазвичай користується одним із двох типів атак: атака переповнення буфера або атаки флуду.

1.2.1 Атаки переповнення буфера

Тип атаки, при якому переповнення буфера пам’яті, зображене на рис 1.1, може призвести до того, що машина починає споживати весь доступний простір на жорсткому диску, пам’ять або час ЦП. Ця форма експлойту часто призводить до млявої поведінки, системних збоїв або інших шкідливих дій сервера, що призводить до відмови в обслуговуванні.

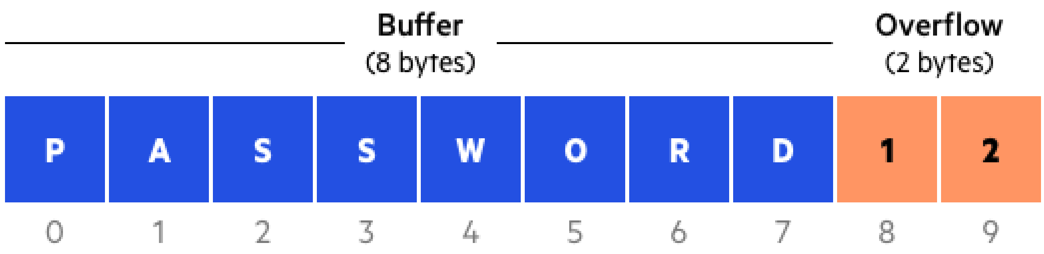


Рисунок 1.1 -- Переповнення буфера [20]

1.2.2 Атаки флуду

Відправляючи на цільовий сервер величезну кількість пакетів, зловмисник може перенасичувати потужність сервера, що призводить до відмови в обслуговуванні. Для того, щоб більшість атак DoS-Flood були успішними, зловмисник повинен мати більшу доступну пропускну здатність, ніж ціль. Схематично атака флуду зображена на рис. 1.2

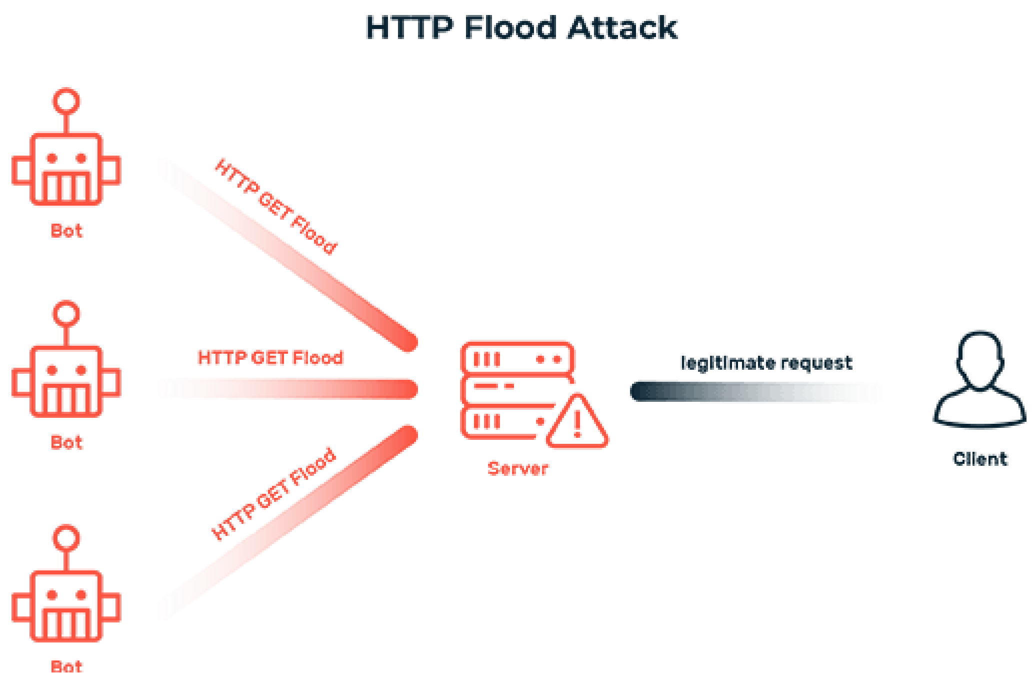


Рисунок 1.2 – HTTP Flood Attack [21]

1.3 Розподілена атака відмови в обслуговуванні (DDoS).

1.3.1 Різниця між DDoS-атакою та DOS-атакою

Відмінною відмінністю між DDoS і DoS є кількість з'єднань, використаних під час атаки. Деякі атаки DoS, такі як повільні атаки на відмову в обслуговуванні, як-от Slowloris, мають свою силу в простоті та мінімальних вимогах, необхідних для їх ефективності.

DoS використовує одне з'єднання, тоді як DDoS атака використовує багато джерел трафіку атаки.

1.3.2 Розподілена атака на відмову в обслуговуванні

Серед кіберзлочинців та організованих злочинних груп розподілені атаки відмови в обслуговуванні набувають все більшої популярності. Ці організації об’єдналися в складні ієрархії та структури для координації та посилення наслідків атаки. Крім того, ці групи іноді використовують свою організацію для здійснення злочинів з вимаганням або іншими схемами нелегального заробітку.

DDoS-атаку характеризує величезна кількість скомпрометованих джерел з який одночасно запускається атака на веб-сервер, щоб відмовити дійсним користувачам у наданні послуг.[6] DDoS-атака є надзвичайно простим, але потужним типом атаки, яка перевантажує пропускну здатність мережі та кількість можливих підключень на постійній або тимчасовій основі. Потік трафіку під час атаки DDoS зовні виглядає так легітимний, тому стає важко відрізнити запити законного користувача від запитів зловмисника. Останнім часом DDOS-атаки стали дуже серйозною небезпекою для хмарних, мобільних та веб-додатків. Як правило, DDoS-атака запускається зловмисником з набору скомпрометованих систем, відомих як ботнет. Схематично ботнет зображено на рис. 1.3

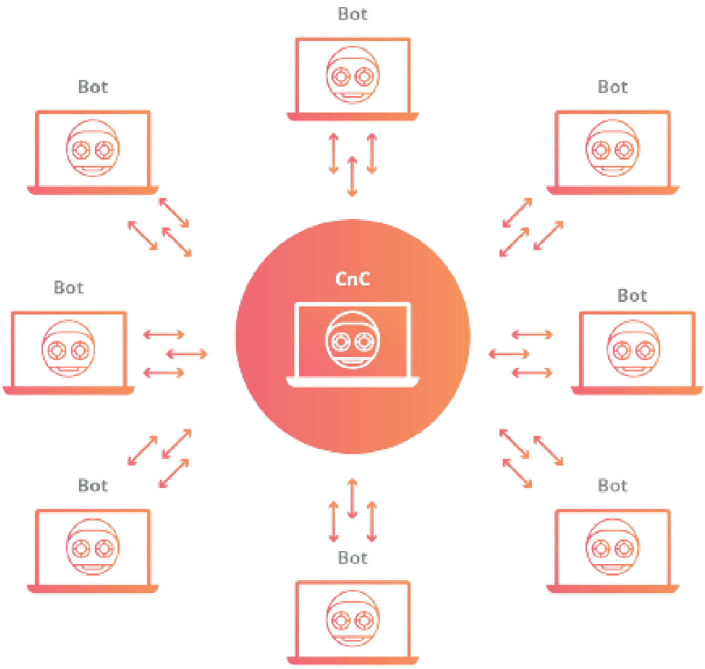


Рисунок 1.3 – Ботнет [22]

Для атаки зловмисник використовує технологію клієнт-сервер. Загалом DDoS-атака складається з майстер-програми, обробника, агентів і жертви. Зомбі (агенти або боти) — це машини, які використовує майстер-програма для формування ботнету. Чим більше зомбі, тим більш руйнівною буде атака. Крім того, зв'язок між зловмистиком і агентами здійснюється за допомогою обробників. За допомогою них зловмисник надсилає команди для керування ботнетом. Фактично атаку здійснюють скомпроментовані комп’ютери, що об’єднані у ботнет.

Зловмисник використовує багато методів сканування для пошуку вразливих до атак машин. Найбільш уразливим рівнем стеку OSI та TCP/IP для DDoS-атаки є транспортний рівень (рівень 3 OSI) і мережевий рівень (рівень 4 стеку OSI та TCP/IP) системи зв'язку. Атаки, спрямовані на ці рівні, призначені для заповнення мережевого інтерфейсу трафіком атаки, щоб перевантажити його ресурси та позбавити здатності реагувати на законний трафік. Крім того, атаки прикладного або сьомого рівня тепер стають більш популярними і складними для протидії видами атак DDoS, тому що трафік під час них майже подібний до законного. Але їх виконання ускладнюється і для зловмисника – для початку необхідно встановити справжнє з'єднання із жертвою.

1.4 DDoS-атаки прикладного рівня.

Атаки прикладного рівня спрямовані на рівень 7 стеку OSI, який, звертається безпосередньо до кінцевого користувача, надаючи доступ до сервісів, до яких звертається користувач. Більше того, цей шар вважається найбільш досяжним і найбільш помітним для зовнішнього світу в мережі. DDoS-атака прикладного рівня створює менше мережевого трафіку, ніж атаки інших рівнів, отже виявити її стає важче. Крім того, вона спричиняє більше накладних витрат на систему з еквівалентним обсягом трафіку шкідливих запитів на стороні сервера та показує більшу можливість обійти системи вторгнення та виявлення, ніж традиційна DDoS-атака. Найбільш цільовими сайтами є фінансові, провайдери, сфера розваг, уряд, охорона здоров'я та освіта. Як ми згадували раніше, DDoS-атаки на прикладному рівні поділяються на два типи: споживання пропускної спроможності (HTTP flooding) і виснаження ресурсів. У першому типі та через потік легітимних запитів зловмисники атакують сервер-жертву. Іншими словами, при атаці на вичерпання на сервер надсилається величезна кількість фальшивих запитів HTTP, щоб завантажити великий файл, оскільки в результаті цей трафік споживає повну пропускну здатність, а сервер-жертви не надає йому послуг. У другому типі підроблені HTTP-запити надсилаються на сервер, щоб споживати ресурси сервера, такі як сокети, центральний процесор, пам'ять, пропускна здатність диска/бази даних і пропускна здатність вводу-виводу. Зі збільшенням обчислювальної складності в Інтернет-додатках і більшою пропускною здатністю мережі ресурси сервера можуть стати вузьким місцем цих програм. Цей тип атаки може використовувати менше комп'ютерів-ботів, але атака завдає високої шкоди веб-сайту.

1.4.1 Послідовність створення TCP з'єднання

Перед початком обміном інформацією за допомогою TCP, між двома комп'ютерами необхідно встановити стабільний зв'язок. Це відбувається за допомогою "трьохетапного рукостискання". Перший комп'ютер відправляє другому пакет з SYN бітом, що означає пропозицію створити з'єднання. Другий комп'ютер відправляє зворотній пакет з бітами ACK та SYN, на знак підтвердження. Перший комп'ютер відповідає пакетом з ACK бітом, чим ініціює з'єднання. Тепер комп'ютери готові обмінюватися інформацією.

Поки інформація відправляється за допомогою TCP, приймаюча сторона повинна відповідати пакетом з бітом ACK на кожен прийнятий пакет з даними. Числа номеру в послідовності та ACK є частиною заголовку TCP та допомагають комп'ютерам відстежувати, які дані були втрачені, а які надіслані двічі чи прийняті не в свою чергу.

Для ініціації завершення з'єднання, перший комп'ютер відправляє пакет із встановленим бітом FIN. Другий відповідає йому пакетом із бітами FIN та ACK, на що перший комп'ютер відповідає пакетом із бітом ACK і завершує з'єднання.

TCP зв'язок може визначати втрачені пакети, використовуючи таймер. Після відправки пакету з даними. Після відправлення пакета відправник запускає таймер і поміщає пакет в чергу повторної передачі. Якщо таймер закінчився, а відправник ще не отримав підтвердження від одержувача, він знову надсилає пакет. Процес з'єднання зображено на рис. 1.4

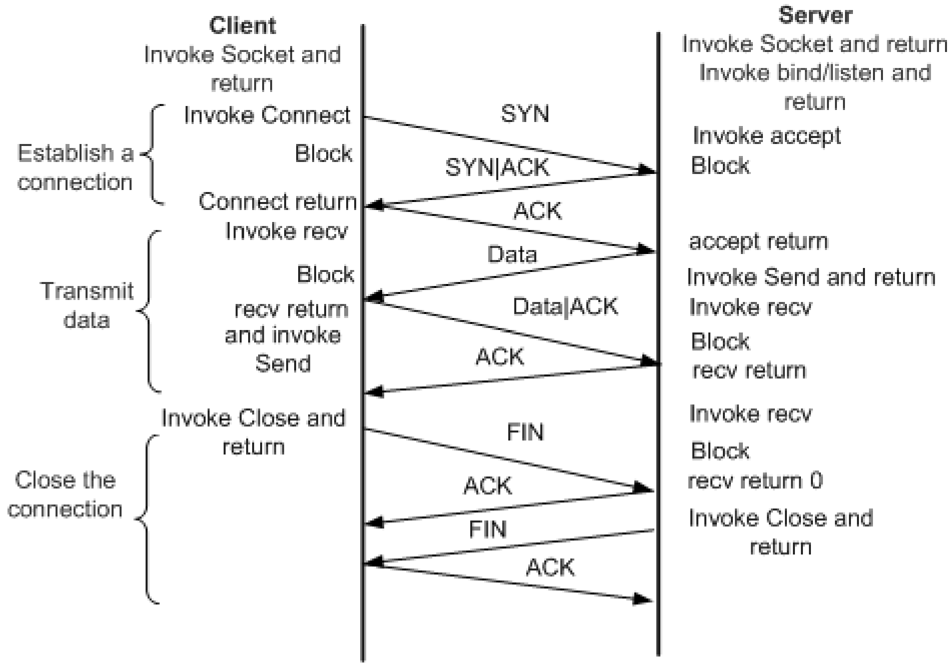


Рисунок 1.4 – TCP з'єднання/роз'єднання [23]

TCP є надійним протоколом, який перевіряє помилки та оцінює, чи були отримані пакети під час спілкування між двома машинами. Отримуючи відповідь сервера поступово та використовуючи маленький розмір вікна TCP, повільні DoS-атаки рівня додатків підтримують відкриту лінію з'єднання. Відправнику не потрібно надавати будь-які додаткові дані для підтримки зв'язку, щоб тримати тунель відкритим. Активувавши вікно з'єднання з нульовим байтом, ціль створює вразливість до повільної DoS-атаки на прикладному рівні. Масштабні атаки DDoS, швидше за все, будуть виявлені негайно, проте атаки «повільного» рівня можуть залишатися непоміченими протягом тривалого періоду. Ці атаки призведуть до відмови або погіршення якості обслуговування законних клієнтів. Клієнти, які купують продукти та потребують доступу до Інтернет-рахунків у будь-який час доби, очікують, що мережі будуть мати швидкий та ефективний доступ для своїх щоденних операцій. Через високу залежність бізнесу від постійного доступу до сервера, зловмисники звернули свою увагу на такі типи серверів. DoS-атаки покликані перешкодити клієнтам і співробітникам отримувати послуги. Нижче наведено методи, які використовуються для DDoS-атак на рівні програми.

1.4.2 Статистика DDoS атак

За даними CloudFlare [7], у четвертому кварталі 2021 року кількість атак атак на відмову в обслуговуванні на прикладному рівні на підприємства зросло на 641%, новий за допомогою нового ботнету Meris, здійснили одну із найбільших HTTP-атак за всю історію. Порівняльний графік надано на рис. 1.5

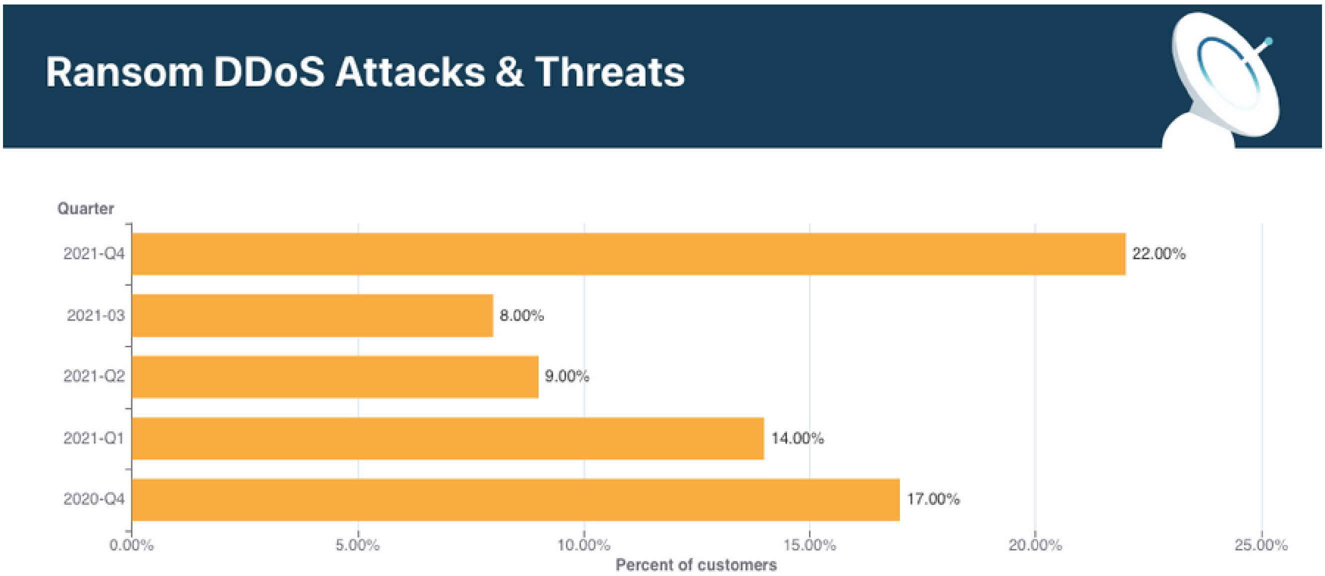


Рисунок 1.5 -- Статистика DDoS Cloud Flare [24]

Фахівці з Link11 стверджують[9], що окрім постійного зростання розподілених атак на відмову в обслуговуванні за останні роки, додається геополітична напруженість. На тлі війни в Україні слід очікувати, що кібератаки також продовжуватимуть зростати як засіб асиметричної війни. Основна увага тут зосереджена на DDoS-атаках, які призводять до збою складної інформаційної інфраструктури, наприклад в державних органах чи фінансових установах, з метою їх саботування та виводу з ладу.

За даними NexusGuard[8] в першому кварталі 2021 кількість дрібних DDoS-атак зросла на 233% . Порівняльний рис. 1.6

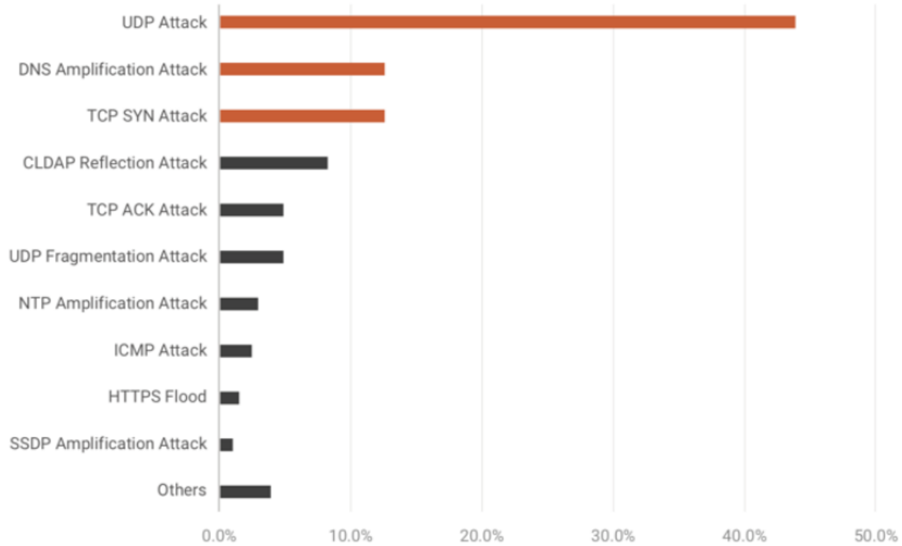


Рисунок 1.6 -- Статистика DDoS NexusGuard [25]

1.5 HTTP DDoS-атаки та їх типи

HTTP DDoS-атаки — це атаки рівня програм у системі. Ці атаки мають на меті зробити онлайн-сервіси недоступними для законних кінцевих користувачів. [10]

Зазвичай атаки прикладного рівня зосереджуються на протоколах передачі даних, оскільки зловмисники можуть використовувати наявні в них слабкості. Націлювання на ресурси програми може приймати різні форми, наприклад атаки HTTP GET, HTTP POST, Slow Read і Apache Header [12]. Протокол HTTP містить слабкі сторони, які зловмисники використовують, застосовуючи низькорівневі та повільні тактики. Повільна атака спрямована на ресурси програми або сервера з невеликим потоком постійного трафіку. Повільні атаки, на відміну від стандартних атак грубої сили, потребують надзвичайно малої пропускної здатності. Їм може бути важко протистояти, оскільки їх трафік атаки добре поєднується зі звичайним трафіком, що робить майже неможливим розрізнити між ними.

1.5.1 DDoS-атаки, що заповнюють протокол HTTP

Атаки з переповненням HTTP – це атаки рівня додатків і зосереджені на заповненні надмірним запитом до веб-сервера, щоб перевантажити веб-сервер і зробити його не в змозі обробити вхідні запити. Згодом роботу сервісу буде припинено.

1.5.2 Атака регулярних виразів (ReDoS).

Ці DoS або ReDoS використовують спеціально створене повідомлення з регулярним виразом, щоб використати недолік у бібліотеці програмного забезпечення на стороні сервера. Недолік дозволяє серверу витрачати свої ресурси на обчислення регулярного виразу на основі введеної користувачем інформації.

1.5.3 Атака колізії хешування

Атаки з колізією хешування використовують загальних недоліки безпеки в фреймворках веб-додатків. Хеш-таблиці створюються на серверах додатків для індексації параметрів сеансу POST. Під час повернення порівнянних значень хешування сервери додатків повинні керувати колізією хешування. Операції вирішення колізій споживають додатковий процесорний час, наприклад, коли зловмисник надсилає повідомлення POST з великою кількістю аргументів у сценаріях DoS-атаки з колізією хешування. Колізії хеш DoS-атак надзвичайно успішні. Вони можуть виконуватися навіть з однієї машини, поступово виснажуючи ресурси сервера.

1.5.4 Атаки HTTP флудінгу

Атаки HTTP флудінгу є найпоширенішими DDoS-атаками, спрямованими на ресурси сервера. Ці атаки виглядають як звичайні HTTP-запити GET або POST до веб-сервера жертви, що ускладнює їх ідентифікацію. Атаки HTTP флудінгу часто включають велику кількість комп’ютерів-ботів. Ці боти роблять численні запити на цільовий сайт, чим призводять до DoS.. Під час атак HTTP GET флудингу, зловмисники можуть надсилати різні HTTP-запити на веб-сервер. Веб-сервер може мати кілька підключень від одного клієнта до одного сервера. Кожному клієнтському процесу буде призначено новий номер порту. Навіть якщо вони спробують отримати доступ до одного і того ж серверного процесу, усі вони матимуть різні клієнтські простори та будуть представляти унікальні з’єднання. Цей процес дозволяє виконувати кілька одночасних запитів до одного веб-сайту з одного комп’ютера. Зловмисники можуть націлювати свої запити на головну веб-сторінку, випадкову веб-сторінку, інші ресурси, наприклад файли зображень, або навіть на їх комбінацію. На відміну від масових атак із високою пропускною здатністю, атаки з низькою пропускною здатністю, які здійснюють зловмисники на прикладному рівні, покладаються на атаки Slow Read, щоб уникнути виявлення. Немає потреби в армії ботів, оскільки цей тип атаки можна здійснити лише з однією машиною та використовувати меншу пропускну здатність у порівнянні з традиційними атаками флудингу. Під час повільних атак такого типу, трафік виглядає як законний. Клієнт HTTP – це веб-браузер, який встановлює з’єднання із сервером для надсилання одного чи кількох повідомлень із запитом HTTP.

Диференціація трафіку такої атаки та звичайного трафіку є складною задачею і вимагає досвіду в цій галузі.

1.5.5 Атака за допомогою HTTP GET запитів

Шкідливий GET HTTP-запит монополізує можливості сервера, використовуючи велику кількість відкритих з’єднань. Зловмисник створює і передає часткові GET HTTP-запити на сервер, через що кожен запит на підключення відкривається в окремому потоці. Зловмисник часто відправляє дані заголовка HTTP, щоб гарантувати, що з’єднання залишаються відкритими та не перериваються. Ця передача відбувається повільно, і сервер чекає необмежений час. Цей процес призводить до спорожнення таблиці підключень, що призводить до DoS. Атаки на основі HTTP GET легше генерувати і можуть більш ефективно масштабуватися в сценарії для ботнету.

Іншим підходом до атак у цьому контексті є атака Slowloris. Вона базується на вразливості у запиті HTTP GET. Під час цієї атаки зловмисник передає неповний заголовок, чим змушує сервер чекати недостаючу частину.

1.5.6 Атака на заголовки Range у Apache

Атака Apache Range Header використовує особливості бінарного фільтра сервера Apache HTTP. Ця вада дозволяє зловмисній атаці запускати DoS-атаку через заголовок range. Віддалений зловмисник може уповільнити або виснажити ресурси служби або сервера, що робить його нездатним вчасно реагувати на законних клієнтів. Результатом цієї вразливості є DoS. Сервер не може обслуговувати будь-які запити та відмовляється від будь-яких додаткових підключень. Проста команда генерує запит HEAD з діапазоном заголовків 0-, x-1, x-2, x-3, x-y, де x встановлюється аргументом -a, y встановлюється - b аргумент і збільшується на 1 байт. Тест працює з різними швидкостями з'єднання та номерами через SSL.

1.5.7 Приклади із реального життя

Кілька поширених прикладів того, що спостерігається під час реальних атак:

• DoS-атака з компроментацією веб-сервера і подальше проникнення на веб-сервер піддає жертву, що знаходиться у публічному доступі. Звичайний користувач, швидше за все, стане свідком зниження доступності веб-сторінки або веб-ресурсу компанії.

• Back-End Resources – це інфраструктурні компоненти, які підтримують ресурс, який є видимим для громадськості, наприклад, веб-додаток. Коли DDoS-атака вимикає серверний ресурс, такий як база даних клієнтів або кластер серверів, вона фактично вимикає всі зовнішні ресурси.

• DDoS-атаки, специфічні для мережі та комп'ютера, також можуть бути запущені з локальної мережі, щоб скомпрометувати всю мережу або певний вузол, наприклад сервер або клієнтську систему.

1.6 Повільні HTTP DDoS-атаки.

Повільні HTTP DDoS-атаки є ще однією формою HTTP DDoS-атак, які використовують поведінку протоколу HTTP законним чином. Ця атака повільно споживає всі доступні ресурси, залишаючись малопомітною для детекторів.

1.6.1 Повільні атаки на прикладному рівні

Атаки на прикладному рівні з низькою пропускною здатністю зосереджуються на тому, щоб залишатися непоміченими для програм-детекторів, щоб уникнути виявлення. Атаки на прикладному рівні не вимагають великої кількості ботів і можуть виконуватися на одному комп’ютері. У порівнянні з традиційними атаками типу flooding, ці атаки використовують меншу пропускну здатність сервера. Мережевий трафік йде, коли веб-браузер використовується для створення з’єднання із серверами за допомогою повідомлень запитів HTTP. Потім сервер прийме з’єднання та відповість на запити HTTP спеціальним повідомленням-відповіддю. Особливість цієї атаки у тому, що її важко відрізнити від легітимного мережевого трафіку. Запити HTTP POST або HTTP GET є популярними методами, які використовуються при атаках на прикладний рівень, виснажуючи ресурси веб-серверів. Зловмисники все частіше націлені на послуги HTTP, DNS і SMTP. Деякі з цих атак можуть бути більш успішними, порівняно з іншими, якщо вони потребують меншої кількості мережевих з’єднань.

DoS-атаки важко відстежити, тому що існує велика кількість методів для ухилення від виявлення. Зловмисники змінюють IP-адреси, щоб замаскувати походження трафіку, що ускладнює відстеження DoS. Повільні DoS-атаки прикоадного рівня зосереджені на службах і вразливостях у протоколах, що дозволяє атаці викликати відмову в обслуговуванні без виявлення. Атака проявляє себе у короткому сплеску трафіку, спрямованому на ресурси програми або сервера. Атаки прикладного рівня, що здійснюються за допомогою з’єднання TCP (Transmission Control Protocol), дозволяють атакувати через звичайний мережевий трафік як дійсне з’єднання.

Під час використання TCP, частина IP-пакету форматується як його сегмент. Кожен сегмент містить в собі заголовок та дані. Заголовок варіюється від 20 до 60 байт, в залежності розміру пакету та додаткових полів.

Такі атаки на веб-сервер можуть відбуватися в трьох різних форматах. Детальне пояснення кожного типу наведено нижче. Наступні характеристики створюють особливі потреби та складнощі, пов'язані з виявленням HTTP перевантажень. Порівняльний графік зі звичаний трфіком (жовта лінія), DDoS флуд(зелена лінія) і повільна DDoS атака(фіолентова лінія) зображено на граіфк 1.7

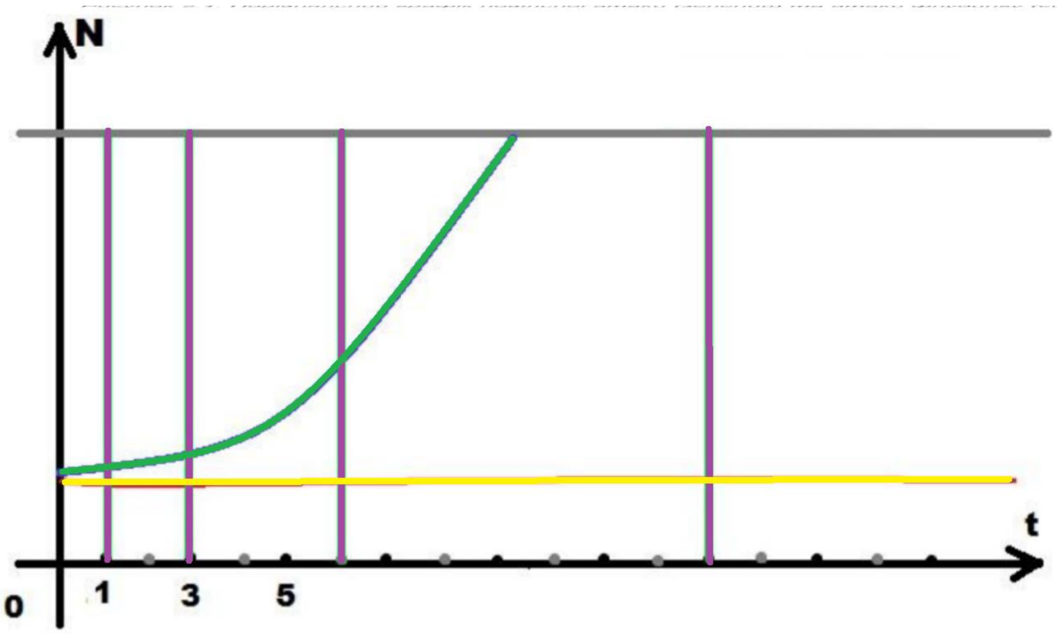


Рисунок 1.7 -- Порівняльний графік Slow DDoS та Flood

1.7 Види повільних атак на відмову в обслуговуванні

1.7.1 Атака Slowloris

Атаки Slowloris (або) Повільні атаки заголовка HTTP: запити HTTP Get зазвичай надсилаються з дійсним заголовком HTTP. Веб-сервер обробляє запит HTTP Get після отримання повного заголовка. Зловмисники використовують цю поведінку протоколу HTTP для здійснення DDoS-атаки на веб-сервер. У цьому методі зловмисник надсилає неповний HTTP-заголовок на веб-сервер і змушує

сервер чекати повного повідомлення. Зловмисник створює багато таких запитів до сервера, поки сервер не зможе обробити жодні запити. Дійсний HTTP-заголовок завжди закінчується послідовним CR LF CR LF (Значення ASCII 0d 0a 0d 0a), як показано на рис. 2.2. Під час атаки заголовок HTTP має лише один CR LF (значення ASCII 0d 0a), щоб вказати неповний заголовок на веб-сервер
Згідно з дослідженням, у 2020 році 19% веб-серверів були вразливими до цієї атаки, тоді як у 2019 році позначка була на рівні 7,5% [13].

Атака проходить в 4 етапи:

Перший: встановлення зловмисником декількох TCP-з'єднань з сервером за допомогою тристороннього рукостискання (SYN, SYN/ACK, ACK), а потім посилання сигналу «keep alive»

Другий: жертва відкриває потік виконання для кожного вхідного запиту з наміром закрити потік після завершення з'єднання. Зазвичай якщо з'єднання займає занадто багато часу, сервер закриває його, звільняючи потік для наступного запиту.

Третій: підтримання встановленого з'єднання за допомогою часткових HTTP-запитів. Пакети «Keep alive» містять прапори PSH ACK і мають неповний HTTP-заголовок. На рисунку 1.3 можемо побачити, що заголовок закінчується на 0d0a, хоча за специфікацією протоколу HTTP, повний заголовок має закінчуватись на 0d0a0d0a.

Четвертий: цільовий сервер ніяк не може звільнити жодне з відкритих часткових з'єднань, очікуючи завершення запиту. Як усі доступні потоки будуть використані, сервер не зможе відповідати на додаткові запити від легітимного трафіку, що призведе до відмови в обслуговуванні.

Оскільки надсилаються не несправні пакети, а просто часткові, Slowloris може легко прошмигнути через традиційні системи виявлення атак (IDS). Якщо атака Slowloris не виявлена, або не припинена командою безпеки жертви, вона може тривати дуже довго. Коли спливає час очікування підключення, Slowloris просто відновлює з'єднання, продовжуючи використовувати усі ресурси серверу.

1.7.2 Повільна HTTP Post атака

Slow HTTP Body Attacks (або slow HTTP Post) здійснюється за допомогою запитів HTTP Post [14]. Атаку Slow HTTP Post зловмисники починають з надсилання законного запиту на сервер. На відміну від атак slowloris, ця атака має повний заголовок. Зловмисник надсилає запит HTTP Post з дуже малим вмістом, розмір якого варіюється від одного до кількох байтів із повільними інтервалами часу. Особливістю є те, що інформація про відправлений пакет містить в собі надзвичайно велике значення поля в заголовку, що відповідає за розмір даних у пакеті. Однак, оскільки запит є законним, спілкування між сервером і зловмисником не блокується системами спостереження і контролю. Сервер чекає на отримання всього тіла повідомлення, визначеного значенням заголовка. Під час цього зв’язку зловмисник надсилатиме дані зі швидкістю всього один байт за передачу. Зловмисник відкриває багато з'єднань із сервером, щоб використати всі його ресурси, щоб сервер був недоступний для законних користувачів. Оскільки дані передаються через регулярні проміжки часу, тайм-аут клієнта сервера не запускається, таким чином зв’язуючи сервер і його ресурси до завершення запиту.

Цей тип атаки може бути реалізований розподіленим способом, використовуючи кілька хостів, або може бути налаштований для здійснення кількох екземплярів атаки з однієї машини залежно від використовуваного інструменту. Це дозволяє зловмиснику здійснити потенційно руйнівну атаку, використовуючи дуже мало ресурсів. Різні інструменти та сценарії DoS Slow HTTP POST є загальнодоступними та вільно доступними для зловмисників, наприклад Low Orbit Ion Cannon (LOIC), R U Dead Yet (R.U.D.Y.) і Open Web Application Security Project (OWASP) Switchblade.

1.7.3 Повільні атаки зчитування

HTTP (slow HTTP Read): це ще один метод DDoS-атаки на веб-сервер. У цій моделі зловмисник надсилає дійсний заголовок і запит на отримання веб-серверу, але він затримує відповідь на читання від сервера. Веб-клієнт/браузер на стороні зловмисника часто надсилає веб-серверу повідомлення про те, що йому недостатньо місця для отримання повідомлення, щоб затримати відповідь сервера. Поле розміру вікна TCP використовується під час методу 3-стороннього рукостискання для обміну розміром вікна між веб-клієнтом і сервером для узгодження кількості повідомлень, які вони можуть обробляти. Після встановлення з’єднання веб-браузер запитує більше даних, але каже, що недостатньо місця для їх отримання.

Це змушує сервер резервувати виділені ресурси для цієї транзакції. Багато транзакцій цього типу відкриваються з боку зловмисника до веб-сервера-жертви, щоб веб-сервер не міг обробляти будь-які вхідні запити. Під час сценарію атаки веб-браузер надсилає кілька повідомлень TCP ZeroWindow на веб-сервер.

1.8 Методи визначення та пом’якшення повільних DDoS атак

Досліджувані методи виявлення Slow DoS були засновані на аналізі мережевого трафіку. Вхідний пакет був розібраний на окремі заголовки протоколу рівня, потім його аналізували та порівнювали з відомими характеристиками атаки, описаними вище. Якщо збіг було знайдено, адреса Інтернет-протоколу (IP) віддаленого хоста та деталі TCP-з’єднання були збережені, а трафік хоста відстежувалося для подальшого аналізу. У зв’язку з легкою плутаниною у відокремленні трафіку зловмисників і користувачів, довелося встановити граничні параметри для точного визначення атак. Ці параметри встановлювалися для кожної атаки окремо через різний тип трафіку. Параметрами моніторингу були: кількість відкритих TCP-з'єднань з одного хоста; максимальний час отримання одного HTTP-запиту; мінімальна швидкість передачі даних; максимальна кількість частин HTTP-запиту.

1.8.1 Метод визначення Slowloris

Якщо було виявлено вхідний запит HTTP GET без дійсного завершення, може бути задіяна атака Slowloris. Контролювалися наступні параметри зв’язку конкретного хоста:

- кількість незавершених частин HTTP-запиту GET, отриманих на TCP-з'єднанні,
- час, що минув з моменту отримання першої частини незавершеного запиту HTTP GET у з'єднанні,
- кількість відкритих TCP-з'єднань з однієї IP-адреси, на яку надійшли незавершені HTTP-запити GET.

Якщо будь-який з параметрів було перевищено, атака була виявлена, а вихідна IP-адреса заблокована.[15]

1.8.2 Метод виявлення Slow HTTP POST

Для вхідних запитів HTTP POST значення довжини вмісту порівнювалося з фактичним розміром отриманих даних. Якщо значення не збігаються, було виявлено початок атаки Slow POST і відстежено віддаленого користувача. Відстежувалися ті самі параметри трафіку, що й під час атаки Slowloris. Також відстежувалося зниження швидкості передачі даних. При перевищенні граничних значень була виявлена атака.[15]

1.8.3 Метод виявлення Slow HTTP Read

Сегмент TCP SYN фіксувався під час кожного рукостискання TCP між веб-сервером і хостом. Значення Window Scale Factor у заголовку сегмента TCP SYN було використано для обчислення розміру вікна TCP. Якщо розмір вікна був підозріло малим, з’єднання відстежувалося. Контролювалися наступні параметри зв’язку хоста:

• тривалість відкритого TCP-з'єднання,

• зниження швидкості передачі даних,

• кількість відкритих з'єднань з конкретної IP-адреси, з якої було отримано сегмент TCP з дуже малим розміром вікна.

Якщо будь-який з параметрів було перевищено, атака була виявлена, а вихідна IP-адреса заблокована.[15]

1.8.4 Методи пом’якшення атак

Для того, щоб захистити свій веб-сервер від повільних HTTP DDoS-атак, необхідно зробити декілька налаштувань:

- встановити максимальну тривалість підключення, базуючись на середній тривалості підключення для вашої моделі Інтернет-трафіку;
- обмежити заголовок та тіло повідомлення до мінімальної адекватної довжини;

- встановити мінімальну швидкість вхідних даних, інакше закривати з’єднання. [15]

Окрім цього, варто змінити конфігурацію апаратного забезпечення так, щоб брандмауер відкидав вхідні ICMP-пакети або блокував DNS-відповіді з-за меж вашої внутрішньої мережі, заблокувавши порт UDP 53. Це допоможе захистити від певних атак DNS та DDoS-атак на основі ping.

Таким чином, ви не допустите умов, за яких повільна атака буде вдалою, однак потрібно дуже обережно обирати мінімальні та максимальні значення, описані вище, аби не зробити ваш сервіс недоступним для легітимних користувачів.

Для захисту від атак мережевого рівня, можна розмістити свої ресурси за мережами розповсюдження вмісту (CDN). Підвищення продуктивності пов’язане з тим, що вміст кешується і таким чином зменшується навантаження на ваш веб-сервер. Якщо веб-сайт є об’єктом DDoS-атаки, CDN допоможе упевнитися, що він не досягне вихідного сервера і не зробить ваш сайт повністю недоступним. Коли на сервер потрапляє більше трафіку, ніж він може обробити, трафік просто надсилається на інші сервери.

Висновки до розділу 1

Прикладний рівень поступово стає все більш привабливим для здійснення атак у комп’ютерних мережах. Атаки варіюються від складних руткітів до DoS, головною метою яких є скомпрометувати комп’ютерні мережі. Повільні HTTP DoS-атаки також здійснюються на прикладному рівні. Під час цього типу атак використовується менший обсяг атакуючих ресурсів, ніж під час традиційних DoS-атак Також повільні DoS-атаки часто можуть залишатися непоміченими пристроями та програмами моніторингу мережі стільки часу, скільки необхідно для успіху атаки. Для виявлення таких атак дослідникам доводиться розробляти нові методи спостереження за мережею та користуватися новими підходами до аналізу мережевого трафіку.

Через збільшення розмірів мереж, збільшується і кількість з’єднань до них. Відвопідно до зростоання клількості з’єднань, зростає кількість необхідного простору для збереження дапмів трафіку, тому що збереження кожного пакеа окремо вимагає великих сховищ. Тому постає питання, як можна визначати повільні DDoS атаки, маючи обмежений обсяг збережених даних. Хоч способи розпізнавання повільних атак та пом’якшення їх впливу вже давно існують, залишаються питання, як розпізнавати такі види атак, не занурюючись у вміст пакета.

З’являється потреба в розробці методу детектування повільний DDoS атак, за допомогою аналізу набору мережевого трафіку, що займає менеше місця на сховищах – набору TCP потоків.

2 РОЗПІЗНАВАННЯ ПОВІЛЬНИХ DDoS АТАК У TCP ПОТОКАХ

2.1 Чому нам потрібно аналізувати трафік

Як ми побачили у першому розділі – атака на відмову в обслуговуванні завдають багато шкоди для держави, бізнесу та звичайних користувачів на всіх рівнях. Тому нам необхідно вміти точно і вчасно розпізнавати ці атаки та виживати відповідні заходи для захисту від них. У першу чергу – це блокування адрес, з яких надходять атаки. Це означає, що потрібно не просто зафіксувати акт атаки, а також визначити конкретні IP адреси, що задіяні під час її проведення. Для отримання переліку адрес, нам необхідно проаналізувати кожне з’єднання на відповідність критеріям легітимного з’єднання та зробити висновок щодо шкідливості цього з’єднання.

У першому розділі ми розглянули особливості поведінки мережі, яка знаходиться під впливом повільної атаки, а також чим саме відрізняються шкідливі пакети, що відправляє атакуючий відрізняються від законних в реальному житті у нас не завжди є можливість подивитися всередину пакета, щоб зрозуміти за допомогою наведених методів, чи є він шкідливим. Так само стає неможливим відстежити атаки у ретроспективі, тобто аналізуючи вже зібраний і збережений інтернет трафік. Це може знадобитися для задач із форензики або для дослідження поведінки різних систем захисту в певний момент часу в минулому під час повільної атаки.

Проблемою є відсутність інструмента для аналізу наборів даних про мережеву активність у вигляді двонаправлених TCP потоків.

2.2 Дані про мережеву активність. TCP потоки

Набір даних мережі – це набір даних, який призначений для здійснення мережевого аналізу. Зазвичай він складається з рядків, що представляють поля з інформацією про з’єднання. Набори мережевих даних добре підходять для моделювання з’єднань сервера із клієнтом . У цих наборах зберігається інформація про дані спілкування в дві сторони – запити і відповіді. Цей набір даних представляє собою перелік TCP потоків.

Потік TCP — це з'єднання від точки до точки та інформація про те, як дані протікають через мережу. TCP потік починається після закінчення 3-стороннього рукостискання і завершується після будь-якого переривання з’єднання. Від рукостискання і до завершення існує активний TCP сеанс, й інформація про всі пакети, що були надіслані під час цього сеансу, логується.

Технологія контролю за потоками була створена, щоб пакети між двома хостами не втрачалися. Якщо одержувач приймає приймає і обробляє повідомлення швидше, ніж відправник їх формує і відправляє – все буде добре. Проте якщо ситуація склалась навпаки і одержувач виявився повільніше, повідомлення все одно будуть надходити і додаватися в чергу одержувача. Саме для подолання проблеми швидкого відправника і повільного одержувача в комп’ютерних мережах існує концепція керування потоком. У вигляді TCP потоків дані про мережеву активність зазвичай і зберігаються. Такі файли схожі на таблиці або бази даних, де є хедер із назвами стовпців, розділених комами, та значення, що розміщуються нижче і також розділені комами відповідно до стовпців. Особливістю є те, що ми не маємо даних про кожен пакет окремо, а зберігаємо вибіркову статистику про весь TCP сеанс, наприклад кількість переданих байт або протяжність з’єднання. Схематичне зображення TCP потоку показано на рис. 2.1

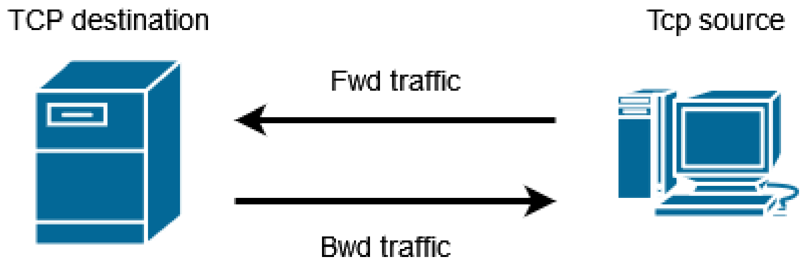


Рисунок 2.1 TCP Flow

2.3 Відмінності збереження даних про TCP потік у файлах .pcap і .csv.

При описі класичних методів детектування повільних атак на відмову мають на увазі, що ми розглядаємо кожен пакет окремо і можемо проаналізувати його побайтово із середини.

Такий аналіз можливий, якщо ми розглядаємо дамп трафіку у вигляді .pcap файлу.

Файли .pcap зберігають дамп даних, захоплених в режимі реального часу. Вони містять в собі всю інформацію про конкретні пакети: заголовки, розмір, спеціальні біти, прапори, інкапсульовані протоколи, тощо. Вони використовуються для аналізу мереж, моніторингу використання смуги пропускання, виявлення неавторизованих DHCP-серверів, виявлення шкідливого програмного забезпечення, кібератак, дозволу DNS, реагування на інциденти та усунення загальних проблем продуктивності.

Їх особливість у тому, що вони зберігають інформацію про кожен пакет. Недоліком такого зберігання даних мережі є те, що велика кількість з'єднань буде займати багато місця на диску. Для миттєвого розпізнавання атаки це можливо, але для аналізу збереженого трафіку вже викликає труднощі, тому що зазвичай у нас немає ресурсів для збереження всіх пакетів окремо.

Рішенням цієї проблеми є збереження даних про активність у мережі у вигляді файлів .csv з інформацією про TCP потоки. Через те, що зберігається не кожен пакет окремо, а тільки загальна інформація про конкретний TCP потік, ці файли займають набагато менше місця і дозволяють аналізувати набагато більший об'єм трафіку.

Також таку інформацію простіше обробляти, аналізувати та будувати маршрути і карти мережі. Для цих дій ми маємо всю необхідну інформацію у простому вигляді.

Набір даних про мережу у вигляді TCP потоків у .csv файлі містить в собі поля, надані у таблиці 2.1

Таблиця 2.1 – Поля у файлі з TCP потоками

Feature name	**Description**
Flow Duration	Тривалість потоку в мікросекундах
total Fwd Packet	Усього пакетів у прямому напрямку
total Bwd packets	Загальна кількість пакетів у зворотному напрямку
total Length of Fwd Packet	Загальний розмір пакету в прямому напрямку
total Length of Bwd Packet	Загальний розмір пакета в зворотному напрямку
Fwd Packet Length Min	Мінімальний розмір пакета в прямому напрямку
Fwd Packet Length Max	Максимальний розмір пакета в прямому напрямку
Fwd Packet Length Mean	Середній середній розмір пакету в прямому напрямку
Fwd Packet Length Std	Розмір стандартного відхилення пакета в прямому напрямку
Bwd Packet Length Min	Мінімальний розмір пакета в зворотному напрямку
Bwd Packet Length Max	Максимальний розмір пакета в зворотному напрямку
Bwd Packet Length Mean	Середній середній розмір пакета в зворотному напрямку
Bwd Packet Length Std	Стандартне відхилення розміру пакета в зворотному напрямку
Flow Bytes/s	Кількість байтів потоку в секунду
Flow Packets/s	Кількість пакетів потоку в секунду
Flow IAT Mean	Середній час між двома пакетами, надісланими в потоці
Flow IAT Std	Стандартне відхилення часу між двома пакетами, надісланими в потоці

Продовження таблиці 2.1

Feature name	**Description**
Flow IAT Max	Максимальний час між двома пакетами, надісланими в потоці
Flow IAT Min	Мінімальний час між двома пакетами, надісланими в потоці
Fwd IAT Min	Мінімальний час між двома пакетами, надісланими в прямому напрямку
Fwd IAT Max	Максимальний час між двома пакетами, надісланими в прямому напрямку
Fwd IAT Mean	Середній час між двома пакетами, надісланими в прямому напрямку
Fwd IAT Std	Стандартний час відхилення між двома пакетами, надісланими в прямому напрямку
Fwd IAT	Загальний загальний час між двома пакетами, надісланими в прямому напрямку
Bwd IAT Min	Мінімальний час між двома пакетами, надісланими в зворотному напрямку
Bwd IAT Max	Максимальний час між двома пакетами, надісланими в зворотному напрямку
Bwd IAT Mean	Середній час між двома пакетами, надісланими в зворотному напрямку
Bwd IAT Std	Час стандартного відхилення між двома пакетами, надісланими в зворотному напрямку
Bwd IAT	Загальний Загальний час між двома пакетами, надісланими в зворотному напрямку
Fwd PSH flags	Кількість разів, коли прапор PSH був встановлений у пакетах, що рухаються в прямому напрямку (0 для UDP)
Bwd PSH Flags	Кількість разів, коли прапор PSH був встановлений у пакетах, що рухаються в зворотному напрямку (0 для UDP)
Fwd URG Flags	Кількість разів, коли прапор URG встановлювався в пакетах, що рухаються в прямому напрямку (0 для UDP)
Bwd URG Flags	Кількість разів, коли прапор URG був встановлений у пакетах, що рухаються в зворотному напрямку (0 для UDP)
Fwd Header Length	Загальна кількість байтів, що використовуються для заголовків у прямому напрямку
Bwd Header Length	Загальна кількість байтів, що використовуються для заголовків у зворотному напрямку
FWD Packets/s	Кількість пакетів пересилання в секунду
Bwd Packets/s	Кількість зворотних пакетів в секунду
Packet Length Min	Мінімальна довжина пакета
Packet Length Max	Максимальна довжина пакета
Packet Length Mean	Середня довжина пакета
Packet Length Std	Довжина стандартного відхилення пакета
Packet Length Variance	Дисперсія довжини пакета

Продовження таблиці 2.1

Feature name	**Description**
FIN Flag Count	Кількість пакетів з FIN
SYN Flag Count	Кількість пакетів із SYN
RST Flag Count	Кількість пакетів із RST
PSH Flag Count	Кількість пакетів із PUSH
ACK Flag Count	Кількість пакетів з ACK
URG Flag Count	Кількість пакетів з URG
CWR Flag Count	Кількість пакетів із CWR
ECE Flag Count	Кількість пакетів з ECE
down/Up Ratio	Коефіцієнт завантаження та завантаження
Average Packet Size	Середній розмір пакета
Fwd Segment Size Avg	Середній розмір, що спостерігається у прямому напрямку
Bwd Segment Size Avg	Середній розмір, що спостерігається у зворотному напрямку
Fwd Bytes/Bulk Avg	Середня кількість байтів масової швидкості в прямому напрямку
Fwd Packet/Bulk Avg	Середня кількість пакетів масової швидкості в прямому напрямку
Fwd Bulk Rate Avg	Середня кількість масових тарифів у прямому напрямку
Bwd Bytes/Bulk Avg	Середня кількість байтів масової швидкості в зворотному напрямку
Bwd Packet/Bulk Avg	Середня кількість пакетів у зворотному напрямку
Bwd Bulk Rate Avg	Середня кількість масової ставки в зворотному напрямку
Subflow Fwd Packets	Середня кількість пакетів у підпотоці в прямому напрямку
Subflow Fwd Bytes	Середня кількість байтів у підпотоці в прямому напрямку
Subflow Bwd Packets	Середня кількість пакетів у підпотоці в зворотному напрямку
Subflow Bwd Bytes	Середня кількість байтів у підпотоці в зворотному напрямку
Fwd Init Win bytes	Загальна кількість байтів, надісланих у початковому вікні в прямому напрямку
Bwd Init Win bytes	Загальна кількість байтів, надісланих у початковому вікні в зворотному напрямку
Fwd Act Data Pkts	Кількість пакетів із принаймні 1 байтом корисного навантаження даних TCP у прямому напрямку
Fwd Seg Size Min	Мінімальний розмір сегмента, що спостерігається у прямому напрямку

Продовження таблиці 2.1

Feature name	**Description**
Active Min	Мінімальний час, протягом якого потік був активним перед тим, як перейти в режим простою
Active Mean	Середній час, коли потік був активним до того, як перейти в режим простою
Active Max	Максимальний час, протягом якого потік був активним перед тим, як перейти в режим простою
Active Std	Час стандартного відхилення, коли потік був активним, перш ніж перейти в режим простою
Idle Min	Мінімальний час простою потоку перед тим, як він стане активним
Idle Mean	Середній час простою потоку до того, як він став активним
Idle Max	Максимальний час простою потоку перед тим, як він стане активним
Idle Std	Час стандартного відхилення, коли потік не став активним

2.4 Складність аналізу повільних DDoS атак у .csv файлах

Через те, що ми не зберігаємо кожен пакет окремо, у нас немає можливості подивитися чи наявні всі байти у заголовку, чи співпадає заявлений розмір пакету з реальним, чи розмір вікна надто малий та чи не відправляється надто багато пакетів keep alive. Тобто ті ознаки, що дозволяють ідентифікувати повільну атаку на відмову в обслуговуванні, розглядаючи кожний пакет окремо. В цьому полягає складність виявлення даних атак у наборах даних мережі з потоками TCP. Проте ми маємо поля, що характеризують з’єднання. Для точного визначення факту атаки і адреси, з яких вона здійснюється, нам необхідно мати уявлення про звичайну роботу мережі, яку ми аналізуємо та визначити граничні значення, після яких трафік вважається підозрілим.

Для того щоб визнати підключення шкідливим, при аналізі такого набору даних, нам не достатньо мати один критерій. Ми повинні обрати поля із заданого набору даних, що можуть характеризувати повільну атаку на відмову в обслуговуванні. Тобто такі поля, значення яких змінюють свою звичайну поведінку під час атаки. Також необхідно обрати такі поля, що або не залежать один від одного, або залежать не повністю. Це потрібно для того, щоб при кожному перевищенні TCP потоком граничних значень, ймовірність того, що підключення є шкідливим – збільшувалась.

2.5 Аналіз трафіку методами математичної статистики

Для знаходження граничних значень, порівняння їх один з одним, вирахування ймовірності того, що відбувається атака та конкретне з’єднання є шкідливим, в цій роботі було використано апарат математичної статистики для аналізу зібраних даних.

2.5.1 Математична статистика

Математична статистика – це сучасна галузь математичної науки, яка займається статистичним описом спостережень, а також побудовою математичних моделей, що містять поняття ймовірності. Теоретичною базою математичної статистики служить теорія ймовірностей. [16]
Але на відміну від теорії ймовірностей, де модель явища вважалася заданою та проводилися розрахунки ймовірностей можливих змін, у математичній статистиці ми виходимо з відомих реалізацій випадкових подій (статистичних даних) і розробляємо методи, які дозволяють за цими даними підібрати відповідну теоретико-ймовірнісну модель.

Основна математична модель випадкового явища базується на понятті ймовірнісного простору (Ω, U, P) .

При вивченні конкретного експерименту ймовірність $P(\cdot)$ рідко буває відомою повністю. Часто апріорі можна стверджувати лише те, що $P(\cdot)$ є елементом деякої сім'ї ймовірностей $\mathbb{P}$.

Клас $\mathbb{P}$ може містити всі ймовірності, які можна задати на U – ситуація повної невизначеності. В інших випадках є деякою вужчою сім'єю ймовірностей, заданою в якій-небудь формі. Якщо фіксовано клас $\mathbb{P}$, то кажуть, що задано статистичну (імовірнісно-статистичну) модель, і розуміють під цим набір $(\Omega, U, \mathbb{P})$ [17]

Де Ω – простір елементарних результатів. Тобто множина всіх полів, які можуть набувати значення.

U - Сукупність усіх підмножин

$\mathbb{P}$ - Класи ймовірностей

Набір мережевого трафіку у виді TCP потоків, який ми аналізуємо для отримання граничних значень, є скінченною величиною. Значення, що набувають поля є скінченною сукупністю випадкових величин. Кількість потоків у дампі трафіку позначає кількість випробувань. Вибіркою називатиметься сукупність випадкових величин, що спостерігаються в експерименті.

Для отримання граничних значень при аналізі трафіку нам необхідно буде знаходити середні значення та медіани вибірки.

Значення деяких полів, що під час нормальної роботи мережі суттєво різняться між собою, починають бути дуже схожими під час атаки. Тому для аналізу на однорідність нам необхідно буде розбити вибірку на класи по IP адресам, розрахувати дисперсію, та коефіцієнт варіації для певних полів, щоб отримати об’єктивну числову міру варіації і порівняти її зі шкідливим трафіком.

2.5.2 Дисперсія

Середнє відхилення – це сума всіх абсолютних відхилень від середнього, поділена на об’єм вибірки. Однак цей вимір варіації є недостатньо інформативним, тому використовують інший підхід до визначення виміру варіації – дисперсію. Вона ґрунтується на використанні квадратів відхилень, яку розділяють на число елементів, для того, щоб значення не було чутливе до числа елементів ряду. Отриманий показник і є дисперсією.[18]

Таким чином дисперсію можна знайти за формулою:

$$var[y(k)] \ = \ \sigma_y^2 = \frac{1}{N-1}\sum_{k=1}^{N}[\,y(k) - \mu_y]^2 \qquad (2.1)$$

де var – позначення дисперсії, μ_y – середнє значення ряду розподілу {y(k)}.

Ділення суми квадратів відхилень від середнього на N-1 забезпечує незміщенність оцінки дисперсії.

Розрахункову формулу для дисперсії можна представити також в іншому вигляді:

$$\sigma_y^2 = \frac{\sum_{k=1}^{N}[y(k)-\mu_y]^2}{N-1} = \frac{\sum_{k=1}^{N} y^2(k) - 2\mu_y \sum_{k=1}^{N} y(k) + N\mu_y^2}{N-1} = \frac{\sum_{k=1}^{N} y^2(k)}{N-1} - \frac{N\mu_y^2}{N-1} + \left(\frac{\sum_{k=1}^{N} y(k)}{N-1}\right)^2 =$$

$$= \frac{\sum_{k=1}^{N} y^2(k)}{N-1} - \frac{N\mu_y^2}{N-1} \qquad (2.2)$$

або, підставляючи формулу середнього значення:

$$\sigma_y^2 = \frac{\sum_{k=1}^{N} y^2(k)}{N-1} - \frac{(\sum_{k=1}^{N} y(k))^2}{N(N-1)} \qquad (2.3)$$

де N - кількість елементів у вибірці,

$y(k)$ – наша вибірка

Тобто дисперсія – це середній квадрат відхилення. Спочатку ми розраховуємо середнє значення, а потім обираємо різницю між кожним даним середнім значенням, підносимо до квадрату, складаємо і ділимо на кількість значень даної вибірки. Різниця між окремими значеннями і середнім показує міру відхилення.

2.5.3 Коефіцієнт варіації

Значення стандартного відхилення залежить від масштабу самих даних, що не дозволяє порівнювати варіативність різних вибірок. Щоб усунути вплив масштабу і отримати значення, за яким можна порівнювати різні за масштабом вибірки, необхідно розрахувати коефіцієнт варіації по формулі:

$$V = \frac{\sigma_y^2}{\mu_y} \qquad (2.4)$$

де σ_y^2 – дисперсія,

μ_y – середнє значення розподілу

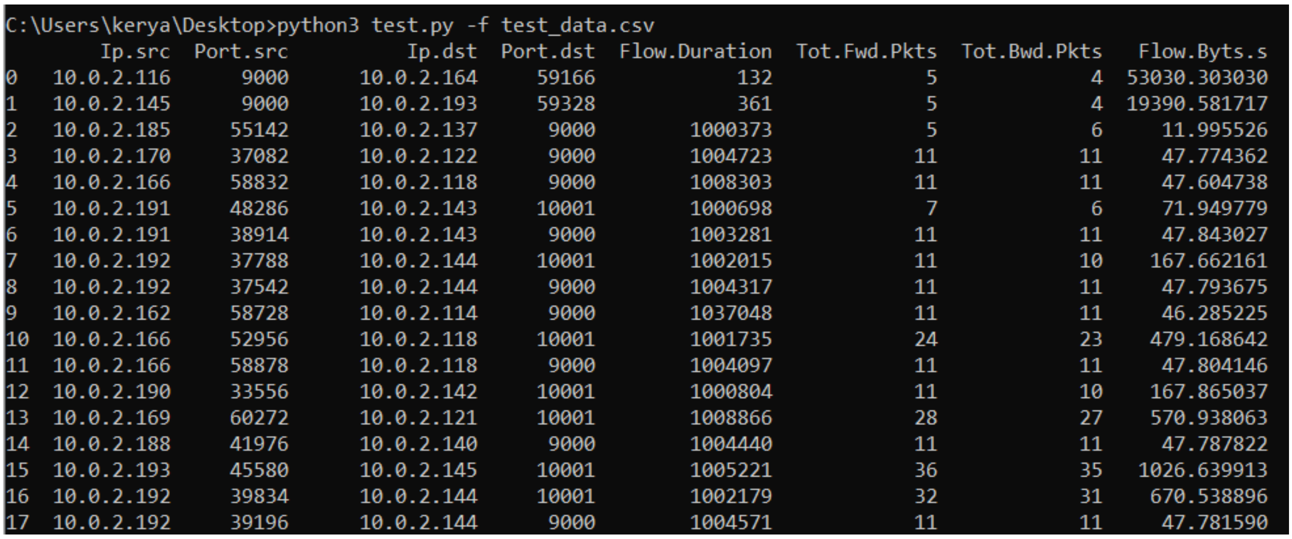
Мінливість вважається слабкою, якщо ν<10%; якщо ν від 11-33%, то середньою і значною за ν>33% [19]

2.6 Аналіз нормального трафіку

Отримання картини нормальної роботи мережі необхідно, щоб розуміти як поводить себе трафік, як довго зазвичай відбуваються з’єднання і яка кількість даних під час них передається. На основі цієї інформації відбувається виявлення повільних атак на відмову в обслуговуванні.

Такі атаки відрізняються від звичайного трафіку насамперед довгим з’єднанням і малою (або відсутньою) кількістю переданих даних під час сеансу

TCP. Шкідливий трафік також має деяку однорідність, тобто значення деяких полів не сильно відрізняються від потоку до потоку під час атаки. В цій роботі було проаналізовано ці та інші відхилення від нормальної роботи мережі. Також були створені методи пошуку критеріїв, що дозволяють зробити висновок про шкідливість з'єднання. Зауважимо, що для коректного формування граничних значень, необхідно мати велику вибірку даних з нормальним мережевим трафіком. Це дозволить пом'якшити аномальні відхилення, які присутні навіть у легітимних з'єднаннях. Приклад набору трафіку приведено на рис. 2.2. Приклад файлу з декількома потоками наведено у Додатку Б.



```
C:\Users\kerya\Desktop>python3 test.py -f test_data.csv
        Ip.src  Port.src        Ip.dst  Port.dst  Flow.Duration  Tot.Fwd.Pkts  Tot.Bwd.Pkts   Flow.Byts.s
0   10.0.2.116      9000    10.0.2.164     59166            132             5             4  53030.303030
1   10.0.2.145      9000    10.0.2.193     59328            361             5             4  19390.581717
2   10.0.2.185     55142    10.0.2.137      9000        1000373             5             6     11.995526
3   10.0.2.170     37082    10.0.2.122      9000        1004723            11            11     47.774362
4   10.0.2.166     58832    10.0.2.118      9000        1008303            11            11     47.604738
5   10.0.2.191     48286    10.0.2.143     10001        1000698             7             6     71.949779
6   10.0.2.191     38914    10.0.2.143      9000        1003281            11            11     47.843027
7   10.0.2.192     37788    10.0.2.144     10001        1002015            11            10    167.662161
8   10.0.2.192     37542    10.0.2.144      9000        1004317            11            11     47.793675
9   10.0.2.162     58728    10.0.2.114      9000        1037048            11            11     46.285225
10  10.0.2.166     52956    10.0.2.118     10001        1001735            24            23    479.168642
11  10.0.2.166     58878    10.0.2.118      9000        1004097            11            11     47.804146
12  10.0.2.190     33556    10.0.2.142     10001        1000804            11            10    167.865037
13  10.0.2.169     60272    10.0.2.121     10001        1008866            28            27    570.938063
14  10.0.2.188     41976    10.0.2.140      9000        1004440            11            11     47.787822
15  10.0.2.193     45580    10.0.2.145     10001        1005221            36            35   1026.639913
16  10.0.2.192     39834    10.0.2.144     10001        1002179            32            31    670.538896
17  10.0.2.192     39196    10.0.2.144      9000        1004571            11            11     47.781590
```

Рисунок 2.2 -- Приклад набору мережевого трафіка

2.7 Пошук граничних значень

2.7.1 Тривалість з'єднання

За це значення відповідає поле 'Flow Duration'. Під час нормальної роботи мережі це поле може набувати різних значень – від дуже малих до великих.

Для формування критерію нам необхідно знайти середнє арифметичне значення цієї множини.

2.7.2 Кількість байтів і потоків в секунду

За ці значення відповідають поля ‘Flow Bytes/s’ та ‘Flow Packets/s’. Вони показують скільки байтів проходило у сесії за секунду. Під час повільної DDoS атаки цей показник повинен бути дуже малий, тому що даних між атакуючим та сервером майже не відправляється, а сервер знаходиться в режимі очікування, не розриваючи з’єднання. Нам необхідно знайти медіану кожної з двох вибірок.

2.7.3 Кількість пакетів, відправлених із джерела

За це значення відповідає поле ‘Forward Packets/s. Воно означає кількість переданих даних відправником за одну секунду. Ми знаходимо медіану нашої вибірки всіх сеансів. Таким чином ми визначаємо яку саме кількість даних клієнт може відправити в середньому

2.7.4 Кількість пакетів, відправлених на джерело

За це значення відповідає поле ‘Backward Packets/s. Воно означає кількість переданих даних назад до відправника відправника за одну секунду. Ми також знаходимо медіану нашої вибірки, враховуючи дані всіх сеансів.

2.7.5 Мінімальне значення підключень з однієї адреси, створених за одну хвилину

Для цього нам потрібно поле ‘DateTime’ та ‘IP Source’. Нам треба розбити вибірку на класи, групуючи значення по IP адресам джерела і визначати кількість підключень протягом кожної хвилини. Із даної вибірки обираємо середнє. Це значення допоможе нам виявити як багато підключень в середньому здійснюється з однієї адреси за одну хвилину. Мінімальним значення з якого почнемо спостереження будемо вважати одне підключення на секунду, тобто шістдесят підключень на хвилину. Менша кількість підключень у вибірку входити не буде.

2.7.6 Мінімальний коефіцієнт варіації для вибірки із кількості переданих байтів в потоці з кожної адреси

Для пошуку цього значення потрібно використати поля ‘Flow Bytes/s’ та ‘IP Source’. Нам потрібно розбити всю вибірку відправлених байтів на класи по адресам джерела. Записується значення кількості байт для кожного підключення з конкретної IP адреси, для кожного класу розраховується дисперсія та середнє значення. Потім знаходяться коефіцієнти варіації для кожної IP адреси, серед яких обираємо два мінімальних значення і знаходимо серед них середнє. Це значення і буде складовою критерія.

2.8 Формування критеріїв для визначення шкідливих з’єднань

Критерій 1: Залежність тривалості сеансу від кількості переданої інформації

Зазвичай, чим більша тривалість сеансу, тим більше пакетів передається між хостами.Під час повільної атаки на відмову в обслуговуванні відбувається протилежне – при великій тривалості з’єднання ми маємо дуже малу кількість переданих пакетів. З’єднання вважається підозрілим по першому критерію, якщо відповідає наступним умовам:

1. Тривалість TCP сеансу більша за граничну
2. Кількість переданої інформації менша за граничну

Критерій 2: Однорідність множин переданої інформації та тривалості

При нормальній роботі мережі множини значення кількості байт в секунду можуть суттєво відрізнятися від сеанса к сеансу, створених з однієї адреси, тому вважаються неоднорідною множиною. Так само як і значення тривалості сеансу. Однак під час атаки ці поля будуть приймати схожі значення в кожному потоці. Тривалість з’єднання буде однорідною, тому зо під час атаки зловмисник намагається якомога довше зайняти сервер. Тобто єдине, що скине з’єднання, буде часовий ліміт сервера, тому тривалість буде варіюватися навколо конкретного значення.

Підключення вважається підозрілим, якщо з’єднання з однієї адреси будуть відповідати таким вимогам:

1. Коефіцієнт варіації кількості переданих байті даних з даної адреси менший, ніж значення, що визнано граничним у даній мережі
2. Значення тривалості підключень з даної адреси є однорідною множиною, тобто її коефіцієнт варіації приймає значення менше на 33%

Критерій 3: Велика кількість підключень за хвилину

При атаці на відмову в обслуговуванні ми спостерігаємо велику кількість з'єднань, що відкриваються з адреси атакуючого. Після того, як кількість підключень за хвилину прийме значення більше, ніж зазначено у файлі з границями, підключення буде вважатися підозрілим по другому критерію

Приведені критерії були сформовані виходячи із обмежень у вигляді конкретного набору полів, якими ми можемо оперувати у наборі мережевого трафіку поданого у вигляді двонапрвлених TCP потоків.

На основі кожного критерію окремо не можна робити висновки щодо шкідливості програм, тому що їм можуть відповідати і легітимні з'єднання, що викличе хибні спрацювання. Однак співпадіння по кожному наступному критерію підвищує ймовірність того, що сеанс виявиться шкідливим і з даної адреси відбувається атака. Якщо TCP потік відповідає трьом критеріям одночасно, ми можемо зробити висновок, що з адреси джерела відбувається атака і з'єднання є шкідливим.

Висновки до розділу 2

Для економного збереження дампів мережевого трафіку доречно використовувати файли з TCP потоками, що займають значно менше місця, ніж всі збережені пакети окремо та все ще містять достатньо даних, щоб проводити аналіз поведінки мережі. Вони становлять модель мережі, яку потім можна досліджувати та аналізувати.

За допомогою статистичного аналізу набора даних мереж, розроблені критерії, що дозволяють виявляти факт атаки, а також джерела, з яких вона була здійснена. Апарат математичної статистики дозволяє проаналізувати нормальний трафік та визначити граничні значення. Завдяки цим значенням ми можемо сформувати критерії, за якими будемо оцінювати підключення і робити висновок щодо їх шкідливості.

Існує проблема відсутності інструмента для виявлення повільних атак на відмову в обслуговуванні у таких наборах трафіку. Вона вирішується за допомогою створення програми, яка реалізовує наведений метод виявлення.

3 РОЗРОБКА АНАЛІЗАТОРА ПОВІЛЬНИХ DDoS АТАК В TCP ПОТОКАХ

3.1 Програмне забезпечення та білбіотеки

Програма була розроблена на мові програмування Python 3.9.

Для роботи використовувались наступні бібліотеки:

- Pandas – бібліотека для аналізу .csv файлів
- Statistics – бібліотека для аналізу вибірок та розрахування значень математичної статистики
- Argparse – бібліотека для опрацювання переданих аргументів

Для тестування програми було завантажено тестовий набір мережевого трафіку, що містить в собі 692704 TCP потоки з нормальними та зловмисними підключеннями, які імітували повільні DDoS атаки.

Операційна система: Windows 10

3.2 Аналіз нормального трафіку

Для аналізу нормального трафіку, нам потрібні наступні поля із набору даних: *'Ip.src', 'Date.time', ‘Flow.Duaration', ' Flow.Pkts.s’, 'Flow.Byts.s', 'Fwd.Pkts.s', 'Bwd.Pkts.s'*

Нам неообхідно проаналізувати кожен TCP сеанс та зібрати значення полів в деякі вибірки або розбити їх на класи по адресам джерела. Далі “і” означає індекс конкретного сеансу у даній вибірці.

3.2.1 Пошук значень для першого критерія

Спершу було зібрано кожне значення полів *'Flow.Duaration', ' Flow.Pkts.s', 'Flow.Byts.s', 'Fwd.Pkts.s', 'Bwd.Pkts.s'* у відповідні масиви, для знаходження їх середніх значень. Ці значення будуть використані для аналізу підозрілості трафіку за першим критерієм. На рис. 3.1 наведено програмну реалізацію збору даних для цього критерія.

```
while(i < count_rows):

    duration = to_float(traffic.loc[i, 'Flow.Duration'])
    flow_duration.append(duration)

    flow_p_s = to_float(traffic.loc[i, 'Flow.Pkts.s'])
    flow_pkts_s.append(flow_p_s)

    flow_b_s = to_float(traffic.loc[i, 'Flow.Byts.s'])
    flow_byts_s.append(flow_b_s)

    fwd_p_s = to_float(traffic.loc[i, 'Fwd.Pkts.s'])
    fwd_pkts_s.append(fwd_p_s)

    bwd_p_s = to_float(traffic.loc[i, 'Bwd.Pkts.s'])
    bwd_pkts_s.append(bwd_p_s)
```

Рисунок 3.1 -- Програмна реалізація отримання даних для критерія 1

3.2.2 Пошук значень для другого критерія

Далі було знайдено значення *'Flow.Byts.s'*, що показує кількість переданих байт за секунду в потоці і характерезує кількість переданої інформації. Множину цих значень було розбито на класи по адресам джерела (рис. 3.2).

```
ip = traffic.loc[i, 'Ip.src']

if ip in found_ip:

    ip_flow_b_s[ip].append(to_float(traffic.loc[i, 'Flow.Byts.s']))

else:
    found_ip.append(ip)

    ip_flow_b_s[ip] = []
    ip_flow_b_s[ip].append(to_float(traffic.loc[i, 'Flow.Byts.s']))
```

Рисунок 3.2 -- Програмна реалізація розбиття вибірки на класи по адресам джерел

Також тут відбувається запис всіх адрес, що мають TCP потік у даному наборі трафіку. Це дозволить нам отримувати множини значень конкретного класу у майбутньому. Простими словами ми бачимо скільки було підключень з однієї адреси та яка кількість даних була передана при кожному підключенні. Це дозволяє там розрахувати наскільки однорідними є ці множини. Тобто аналізуючи поведінку реальних користувачів, ми робимо висновки, наскільки хаотичними є їхні дії. Це дозволить там відрізнити реальних користувачів від ботнету, який має схожу значення впродовж кожного підключення. Для порівняння однорідності множин, нам необхідно найти коефіцієнт варіації для кожної підключеної адреси. (Рис.3.3)

```python
for ip in found_ip:
    if(len(ip_flow_b_s[ip]) > connection_per_minute):
        dev = statistics.stdev(ip_flow_b_s[ip])
        mean = statistics.mean(ip_flow_b_s[ip])
        if(mean > 0):
            coef_variation.append(dev/mean)
```

Рисунок 3.3 -- Розрахунок коефіцієнта варіації

Нас цікавлять тільки класи довжиною більше, ніж значення допустимої кількості підключень за хвилину. Воно задається вручну і дорівнює для нашого досліду 60 підключень за хвилин. Це означає, що адреси, які здійснюють за хвилину меньше 61 підключення нас не цікавлять і впливати на значення не будуть. Таке значення було обране, тому що заданий сервер точно здатний витримати одне підключення за секунду і збросить їх його по таймауту, не встигнувши перенавантажитися навіть під впливом атаки. Наступним кроком ми обираємо два мінімальних коефіцієнта варіації і знаходим серед них середнє. Так ми застережемо себе від аномально низьких значень, але не зіпсуємо критерій надто великим показником. Зауважимо, що чим більше вибірка реального трафіку, тим точніше будуть результати підбору граничних значень.

Для розрахунку коефіцієнту варіації необхідно знайти дисперсію вибірки та поділити її на середнє значення цієї ж вибірки. Якщо середє значення дорівнює нулю, ми переходимо до наступного клас множин. (Рис. 3.4)

```python
def limit_coef_of_variation(coef_variation):
    coefs = []
    for i in range(2):
        min_coef = min(coef_variation)
        coefs.append(min_coef)
        coef_variation.remove(min_coef)

    return mean(coefs)
```

Рисунок 3.4 -- Розрахунок граничного значення коефіцієнту варіації

3.2.3 Пошук значень для третього критерія

Останнє, що нам доведеться знайти, це значення кількості підключень за одну хвилину з конкретної IP адреси. Воно розраховується завдяки наявному полу *'Date.time'*.

Формат поля *'Date.time'* наступний: *5/7/2017 8:42.* Тобто час розраховується з точністю до хвилини. Якщо підключення нове, ми створюємо про нього запис у масиві та призначаємо кількості підключень за хвилину значення один та заносимо інформацію про час останнього підключення з цієх адреси. Наступного разу, коли нам знову поадеться ця адреса, ми звіримо час підключення і якщо він співпадає з теперишнім (тобто дане підклчення і попереднє з цієї ж адреси відбулись в одну і ту саму хвилину), то значення кількості підключень за хвилину з цієї IP адреси інкрементується

Нас знову цікавлять тільки адреси, які мали 61 і більше підключень за хвилину, щоб надто малі значення не псували вибірку, тому що є апріорі не підозрілими.

Після цих дій ми знаходимо середню кількість підключень. Це значення і буде вважатися граничним для нашої мережі.

3.2.4 Записування граничних значень у файл

Ми формуємо рядок у форматі запису, аналогічному .csv і записуємо результати аналізу у файл limits.csv, (рис. 3.5) з якого аналізатор трафіку буде брати інформацію про граничні критерії і робити висновок. Приклад файлу limits.csv наведено в Додатку А

```
limits_str = "Flow.Duration.std,Flow.Pkts.s.std,Flow.Byts
limits_str += str(statistics.mean(flow_duration)) + "," +
limits_str += str(statistics.median(flow_byts_s)) + "," +
limits_str += str(statistics.median(bwd_pkts_s)) + "," +

with open('limits.csv', 'w') as file:
    file.write(limits_str)
```

Рисунок 3.5 -- Створення файлу з границями

Запуск програми в режимі аналізу трафіку (рис. 3.6) відбувається командою:

python3 SDoSAD.py -b <file.csv>

```
C:\Users\kirDy\Documents\Diploma>python3 SDoSAD.py -b normal_traffic.txt
limits.csv:

Flow.Duration.std,Flow.Pkts.s.std,Flow.Byts.s.std,Fwd.Pkts.s.std,Bwd.Pkts.s.std,Connection.threshold,Deviation.Flow.Byts.s.min
31324453.92805027,100.288329,5378.6582395000005,60.0258111,23.56767458,102,0.6856274616769584
```

Рисунок 3.6 -- Вивід роботи програми в режимі аналізу нормального трафіка

3.3 Визначення зловмисних адрес

3.3.1 Перевірка за першим критерієм

Перший критерій використовує найменше обчислювальних ресурсів та дозволяє на початку відкинути велику частину не підозрілих адрес.

Для перевірки ми порівнюємо значення полів *‘Flow.Duaration',* *'Flow.Pkts.s’, 'Flow.Byts.s', 'Fwd.Pkts.s', 'Bwd.Pkts.s'* з еталонними і якщо тривалість сеансу потоку більша за граничне значення, а кількість переданої інформації менша, ми сприймаємо цей сеанс як підозрілий, згоідно першому критерію і проловжуємо його аналіз. Програмна реалізація показана на рис. 3.7

```python
i = 0
count_rows = len(traffic) - 1
while(i < count_rows):

    ip = traffic.loc[i, 'Ip.src']
    is_attaker = 0
    is_suspicious_duration = 0

    if (to_float(traffic.loc[i, 'Flow.Duration']) > std_flow_duration):
        is_suspicious_duration = 1

    if(to_float(traffic.loc[i, 'Flow.Pkts.s']) < std_flow_pkts_s):
        is_attaker += 1

    if(to_float(traffic.loc[i, 'Flow.Byts.s']) < std_flow_byts_s):
        is_attaker += 1

    if(to_float(traffic.loc[i, 'Fwd.Pkts.s']) < std_fwd_pkts_s):
        is_attaker += 1

    if(to_float(traffic.loc[i, 'Bwd.Pkts.s']) < std_bwd_pkts_s):
        is_attaker += 1

    if(to_float(traffic.loc[i, 'Bwd.IAT.Min']) == 0.0):
        is_attaker += 1

    if((is_suspicious_duration == 1) & (is_attaker > 3)):
```

Рисунок 3.7 -- Аналіз за першим критерієм

3.3.2 Перевірка за третім критерієм

Підключення, що пройшли перевірку за першим критерієм продовжують досліджуватися далі. Для оптимізації процесу порівняння за критеріями відбувається не послідовно. Під час перевірки на адреси за третім критерієм ми збираємо дані для перевірки за другим, щоб не витрачати на ще один перебір набору даних зайвий час.

Якщо адреса з'явилась вперше, вона заноситься в словник (структура даних на python) підключених адрес та в словник, який рахує кількість підключень за одну хвилину, а також збираються значення з полів *'Flow.Byts.s'* та *'Flow.Duration'*, для майбутнього аналізу за другим критерієм. (рис. 3.8)

Коли адреса з'явиться наступний раз, вона звірить час з попереднім підключенням з цього джерела і якщо він співпаде, це буде означати, що підключення відбулось в одну і ту саму хвилину і значення інкрементується.

Нас цікавлять значення, що у файлі *limits.csv* містить моле *'Connection.threshold'*, тобто кількість підключень, після якої з’єднання буде вважатися підозрілим. Якщо кількість підключень з даної IP адреси перивищило мінімальну допустиму кількість підключень, то адреса заносить у перелік підозрілих адрес, які потім будуть перевірятися за другим критерієм.

```
if((is_suspicious_duration == 1) & (is_attaker > 3)):

    if ip in last_connect:

        ip_flow_b_s[ip].append(to_float(traffic.loc[i, 'Flow.Byts.s']))

        ip_duration[ip].append(to_float(traffic.loc[i, 'Flow.Duration']))

        if(last_connect[ip] == str(traffic.loc[i, 'Date.time'])):
            one_minute_connection[ip][1] += 1
            count_of_connections = one_minute_connection[ip][1]

            if(count_of_connections >= connection_threshold):

                if((ip in count_connect_ip) == False):
                    count_connect_ip[ip] = []

                if(count_of_connections == connection_threshold):
                    count_connect_ip[ip].append(connection_threshold)
                    one_minute_connection[ip][0] += 1

                else:
                    count_connect_ip[ip][one_minute_connection[ip][0]] += 1
        else:
            last_connect[ip] = str(traffic.loc[i, 'Date.time'])
            one_minute_connection[ip][1] = 0
    else:
        one_minute_connection[ip] = [-1,0] #[minute, count_of_conections]
        last_connect[ip] = str(traffic.loc[i, 'Date.time'])
```

Рисунок 3.8 -- Аналіз за третім критерієм

3.3.3 Перевірка за другим критерієм

Після перевірки за першим і третім критеріями ми отримали вибірку адрес, серед яких треба визначити шкідливі. Для цього розраховуються коефіцієнти варіації *'Flow.Byts.s'* та *'Flow.Duration'*, для множин підключень, що містить в собі кожна адреса. Значення коефіцієнта варіації *'Flow.Byts.s'* порівнюється зі значенням у *'Deviation.Flow.Byts.s.min'* із файлу *limits.csv*, а *'Flow.Duration'* зі значенням 0.34 (як було сказано раніше, якщо коефіцієнт варіації приймає значення менше 33%, то множина вважається однорідною). Якщо обидва цих значення у підозрілої адреси менші за граничні, то адреса визнається шкідливою (рис. 3.9)

```
for ip in count_connect_ip:
        ip_deviation_flow_b_s[ip] = statistics.stdev(ip_flow_b_s[ip])/statistics.mean(ip_flow_b_s[ip])
        ip_deviation_duration[ip] = statistics.stdev(ip_duration[ip])/statistics.mean(ip_duration[ip])

if(info == True):
    malicious_ip_info_str = ""
    for ip in ip_deviation_flow_b_s:
        if((ip_deviation_flow_b_s[ip] < deviation_flow_b_s) & (ip_deviation_duration[ip] < deviation_duration)):
            malicious_ip_info_str += "\nIp addresse: " + str(ip) + "\nMax connection per minute: " + str(max(count_connect_ip[ip])) \
            + "\nCoef of variation for duration: " + str(ip_deviation_duration[ip]) + "\nCoef of variation for flow bytes/s: "+ \
            str(ip_deviation_flow_b_s[ip]) + "\n"

    return malicious_ip_info_str

else:
    malicious_ip = []
    for ip in ip_deviation_flow_b_s:
        if((ip_deviation_flow_b_s[ip] < deviation_flow_b_s) & (ip_deviation_duration[ip] < deviation_duration)):
            malicious_ip.append(ip)
    return malicious_ip
```

Рисунок 3.9 -- аналіз за другим критерієм

3.3.4 Вивід програми

При аналізі трафіку на шкідливі підключення програма може виводити інформацію в двох виглядах. Перший вигляд (рис. 3. 10) – вивід масиву шкідливих адрес. Передбачається, що цей вивід може використовуватися для спрямування виводу цієї програми в іншу, для обробки шкідливих адрес та прийняття дій.

Запускається простою командою:

python3 SDoSAD.py -f <file.csv> -l <limits.csv>

```
C:\Users\kirDy\Documents\Diploma>python3 SDoSAD.py -f dataset_1.txt -l limits.csv
Malicious ip:

['172.16.0.1', '192.168.10.50']
```

Рисунок 3.10 -- Вивід роботи програми в режимі розпізнавання атаки

Другий вигляд(рис. 3.11) – З’являється додаткова інформація про шкідливе підключення, що допоможе підтвердити правильність роботи програми.

Запускається командою із додатковим правором –i або --info:

python3 SDoSAD.py -f <file.csv> -l <limits.csv> -info

```
C:\Users\kirDy\Documents\Diploma>python3 SDoSAD.py -f dataset_1.txt -l limits.csv --info

Ip addresse: 172.16.0.1
Max connection per minute: 9898
Coef of variation for duration: 0.12168044575439342
Coef of variation for flow bytes/s: 0.29274256054339165

Ip addresse: 192.168.10.50
Max connection per minute: 221
Coef of variation for duration: 0.13318517148217268
Coef of variation for flow bytes/s: 0.25873070995444264
```

Рисунок 3.11 -- Вивід роботи програми з додатковою інформацією в режимі розпізнавання атаки

Висновок до розділу 3

Під час роботи над третім розділом була розроблена програма, що грунтуєтсья на теорії з першого та другого розділів. Були реалізовані методи аналізу нормального трафіку та виявлення окремих граничних значень. Також було реалізовано аналіз трафіку за критеріями, розробленими у другому розділі, для вявлення повільних атак статистичними методами.

Програма була використана на тестовому наборі даних, що складався більше ніж з 600 тис. TCP потоків. Було досліджено, як працює апарат математичної статистики на вибірці із набору трафіку у вигляді TCP потоків. Програма спрацювала коректно – було знайдено 2 шкідливі адреси, з яких відбувались повільні DDoS атаки. Результат підтвердив, що глобальна аналітика в першому розділі, щодо поведінки трафіку під час повільних атак, а також більш конкретна аналітика в другому розділі виявились вірними.

ВИСНОВОК

Оскільки уряд і бізнес масово преходиять в хмари – кількість одночасно активних користувачів в мережі збільшуєтсья. Через це зростає необхідність мати більші сховища для збереження лоігв мережевого трафіку. Також підвищується кількість кібератак на дані мережі.

Атаки на відмову в осблуговуванні стають знаходяться в переліку найпопулярніших атак і завдають все більше збитків. Дослідники продовжують створюватинові ефективні методи їх розпізнавання. Для дослідження повільних атак також докладаються зусилля і ми вже маємо методи їх виявлення.

У дипломній роботі були надані короткі відомості про основні типи DDoS атак, а також детальні спопоби їх реалізації, особливості їх поведінки та способи розпізнавання, які необхідні для розробки методів статистичного аналізу трафіку для виявлення цих атак. Також були наведені особливості збереження трафіку у вигляді TCP потоків і обгрунтування такого підходу.

Для розпізнавання повільних атак на відмову в ослуговуванні у TCP потоках було використано апарат математичної статистики, що дозволяє виявляти атаки не розглядаючи кожен пакет окремо, а викоритовуючи конкретні статистичі дані про TCP сеанси в мережі.

Під час виконання роботи мені вдалось досягти результатів в дослідженні методів виявлення повільних атак на відмову в обслуговуванні. Були розроблені методики аналізу набору мережевого трафіку у вигляді двонаправлених TCP потоків, за якими можна виявити не тільки факт атаки, а також визначити IP адреси з яких вона відбувалась.

Під час роботи була розроблена програма, в якій втілюються досліджені методи. Вона автоматично аналізує нормальну роботу мережі та розраховує значення, за якими формулюються критерії зловмисної активності. Використовуючи ці значення ми отримуємо результат у вигляді виявлення зловмисних адрес, з яких відбувається повільна DDoS атака.

Програма була протестована на вибірці з більше ніж 600 тисяч з'єднань і успішно виявила серед них 2 шкідливі адреси.

Під час цієї роботи я виконав усі поставлені задачі: від створення методик до реалізації їх у вигляді програми.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ ПОСИЛАНЬ

1. “DDoS Attacks in Cloud Computing: Issues, Taxonomy, and Future Directions” G. Somani, M. Singh Gaur, D. Sanghic, M. Conti, R. Buyya. Central University of Rajasthan, Ajmer, India. Malaviya National Institute of Technology, Jaipur, India. Indian Institute of Technology, Kanpur, India. University of Padua, Padua, Italy. The University of Melbourne, Melbourne, Australia. – Режим доступу до ресурсу: https://www.semanticscholar.org/paper/DDoS-attacks-in-cloud-computing%3A-Issues%2C-taxonomy%2C-Somani-Gaur/594f8ad906911c5f438d68fed3d7e563cb35e35f
2. National Institute of Standards and Technology (NIST) 09-Jan-2019 -- Режим доступу до ресурсу: https://www.nist.gov/programs-projects/nist-cloud-computing-program-nccp
3. International Journal of Computer Science and Information Technology - June 2013 “SECURITY THREATS ON CLOUD COMPUTING VULNERABILITIES” -- Режим доступу до ресурсу: https://www.researchgate.net/publication/289756317_Security_Threats_on_Cloud_Computing_Vulnerabilities
4. A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks (2013) by Saman Taghavi Zargar , James Joshi , David Tipper IEEE COMMUNICATIONS SURVEYS & TUTORIALS -- Режим доступу до ресурсу: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.692.9498&q=A%20collaborative%20approach%20to%20facilitate%20intrusion%20detection%20and%20response%20against%20DDoS%20attacks.
5. “What is a denial-of-service (DoS) attack?” – Team of the Cloudflare -- Режим доступу до ресурсу: https://www.cloudflare.com/learning/ddos/glossary/denial-of-service/

6. “Application Layer Distributed Denial of Service Attacks Defense Techniques” Academic Journal of Nawroz University 2018 Subhi R. M. Zeebaree, Karzan Husseinб, Roshna Muhamad -- Режим доступу до ресурсу: https://www.researchgate.net/publication/330191079_Application_Layer_Distributed_Denial_of_Service_Attacks_Defense_Techniques_A_review

7. DDoS attacks trends for 2021 Q4 – Cloudflare -- Режим доступу до ресурсу: https://blog.cloudflare.com/ddos-attack-trends-for-2021-q4/
8. DDoS thread report for 2021 – Nexusguard -- Режим доступу до ресурсу: https://blog.nexusguard.com/threat-report/ddos-threat-report-fhy-2021
9. DDoS report for 2021 – Link11 -- Режим доступу до ресурсу: https://www.link11.com/en/blog/press/new-link11-ddos-report-for-2021/
10. “The Slow HTTP Distributed Denial of Service Attack Detection in Cloud” Dhanapal A. Ph.D -- Режим доступу до ресурсу: https://www.researchgate.net/publication/332829479_The_Slow_HTTP_Distributed_Denial_of_Service_Attack_Detection_in_Cloud
11. The 14 most commo cyber attacks -- Режим доступу до ресурсу: https://www.crowdstrike.com/cybersecurity-101/cyberattacks/most-common-cyberattacks/
12. Collection and Analysis of Slow Denial of Service Attacks Using Machine Learning Algorithms – ProQuest -- Режим доступу до ресурсу: https://www.proquest.com/openview/c5edb8073f8b9e10eb12c9c6e588d92e/1?pq-origsite=gscholar&cbl=18750&diss=y
13. The Invicti AppSec Indicator Spring 2021 Edition: Acunetix Web Vulnerability Report -- Режим доступу до ресурсу: https://www.acunetix.com/white-papers/acunetix-web-application-vulnerability-report-2021/
14. “Collection and Analysis of Slow Denial of Service Attacks Using Machine Learning Algorithms” by Kemp Flifford -- Режим доступу до ресурсу: https://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS19/paper/view/18318/17435

15. “SLOW DOS ATTACKS DETECTION AND MITIGATION” by Marek Sikora -- Режим доступу до ресурсу: https://dspace.vutbr.cz/bitstream/handle/11012/186725/510_eeict2019.pdf?sequence=1

16. Навчальний посібник “Математична статистика” В. М. Руденко -- Режим доступу до ресурсу: https://shron1.chtyvo.org.ua/Rudenko_Volodymyr/Matematychna_statystyka.pdf

17. Навчальний посібник “Математична статистика” Є. О. Лебєдєв, Г. В. Лівінська, І. В Розора, М. М. Шарапов -- Режим доступу до ресурсу: http://applstat.univ.kiev.ua/ukr/docs/materials/matstat.pdf

18. Навчальний посібник “Математична статистика” П. І. Бідюк, Б. П. Ткач, Т. Харрінгтон -- Режим доступу до ресурсу: https://maup.com.ua/assets/files/lib/book/prikladna_statist_2018.pdf

19. “Біометрія” Лакин Г. Ф.— С. 51

20. Зображення «Buffer overflow example” -- Режим доступу до ресурсу: https://www.imperva.com/learn/application-security/buffer-overflow/

21. Зображення “HTTP Flood Attack” -- Режим доступу до ресурсу: https://www.myrasecurity.com/en/http-flood-attack/

22. Зображення “BotNet” -- Режим доступу до ресурсу: https://www.cloudflare.com/learning/ddos/what-is-a-ddos-botnet/

23. Зображення “TCP connection” -- Режим доступу до ресурсу: https://support.huawei.com/enterprise/en/doc/EDOC1100058931/a1faac62/tcp

24. Зображення “Ransom DDoS Attacks & Treats” -- Режим доступу до ресурсу: https://blog.cloudflare.com/ddos-attack-trends-for-2021-q4/

25. Зображення “Chart of DDoS attacks” -- Режим доступу до ресурсу: https://blog.nexusguard.com/threat-report/ddos-threat-report-fhy-2021

ДОДАТОК А
ПРИКЛАД ФАЙЛУ limits.csv

Flow.Duration.std,Flow.Pkts.s.std,Flow.Byts.s.std,Fwd.Pkts.s.std,Bwd.Pkts.s.std,Connection.threshold,Deviation.Flow.Byts.s.min
31324453.92805027,100.288329,5378.6582395000005,60.0258111,23.56767458,102,0.6856274616769584

ДОДАТОК Б
ПРИКЛАД ТЕКСТОВОГО НАБОРУ ТРАФІКА

Flow.Id,Ip.src,Port.src,Ip.dst,Port.dst,Protocol,Date.time,Flow.Duration,Tot.Fwd.Pkts,Tot.Bwd.Pkts,TotLen.Fwd.Pkts,TotLen.Bwd.Pkts,Fwd.Pkt.Len.Max,Fwd.Pkt.Len.Min, Fwd Packet Length Mean,Fwd.Pkt.Len.Std,Bwd.Pkt.Len.Max,Bwd.Pkt.Len.Min, Bwd Packet Length Mean,Bwd.Pkt.Len.Std,Flow.Byts.s,Flow.Pkts.s,Flow.IAT.Mean,Flow.IAT.Std,Flow.IAT.Max,Flow.IAT.Min,Fwd IAT Total,Fwd.IAT.Mean,Fwd.IAT.Std, Fwd IAT Max,Fwd.IAT.Min,Bwd.IAT.Tot,Bwd.IAT.Mean,Bwd.IAT.Std,Bwd.IAT.Max,Bwd.IAT.Min,Fwd.PSH.Flags, Bwd PSH Flags, Fwd URG Flags, Bwd URG Flags, Fwd Header Length, Bwd Header Length,Fwd.Pkts.s,Bwd.Pkts.s,Pkt.Len.Min,Pkt.Len.Max,Pkt.Len.Mean,Pkt.Len.Std,Pkt.Len.Var,FIN.Flag.Cnt,SYN.Flag.Cnt,RST.Flag.Cnt,PSH.Flag.Cnt, ACK Flag Count, URG Flag Count, CWE Flag Count, ECE Flag Count, Down/Up Ratio, Average Packet Size, Avg Fwd Segment Size, Avg Bwd Segment Size, Fwd Header Length,Fwd Avg Bytes/Bulk, Fwd Avg Packets/Bulk, Fwd Avg Bulk Rate, Bwd Avg Bytes/Bulk, Bwd Avg Packets/Bulk,Bwd Avg Bulk Rate,Subflow Fwd Packets, Subflow Fwd Bytes, Subflow Bwd Packets, Subflow Bwd Bytes,Init.Fwd.Win.Byts,Init.Bwd.Win.Byts, act_data_pkt_fwd,Fwd.Seg.Size.Min,Active Mean, Active Std, Active Max, Active Min,Idle.Mean, Idle.Std,Idle Max,Idle.Min,label

192.168.10.3-192.168.10.25-389-49152-6,192.168.10.25,49152,192.168.10.3,389,6,5/7/2017 8:45,226530,5,2,0,0,0,0,0,0,0,0,0,0,30.90098442,37755,92401.12482,226368,3,226530,56632.5,113157.0085,226368,3,4,4,0,4,4,0,0,0,0,184,80,22.07213173,8.828852691,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,184,0,0,0,0,0,0,5,0,2,0,65535,8192,0,32,0,0,0,0,0,0,0,0,BENIGN

192.168.10.3-192.168.10.25-389-49153-6,192.168.10.25,49153,192.168.10.3,389,6,5/7/2017 8:45,1265,11,5,92,5820,39,0,8.363636364,15.39657577,2910,0,1164,1217.350401,4673517.787,12648.22134,84.33333333,161.8167688,633,1,1265,126.5,203.67635,633,1,377,94.25,181.1691199,366,3,0,0,0,0,376,176,8695.652174,3952.56917,0,2910,347.7647059,815.8355785,665587.6912,0,0,0,1,0,0,0,0,0,369.5,8.363636364,1164,376,0,0,0,0,0,0,11,92,5,5820,65535,2081,4,32,0,0,0,0,0,0,0,0,BENIGN

192.168.10.255-192.168.10.19-138-138-17,192.168.10.19,138,192.168.10.255,138,17,5/7/2017 8:43,20,11,0,2585,0,235,235,235,0,0,0,0,0,1.29E+08,550000,2,1.825741858,7,1,20,2,1.825741858,7,1,0,0,0,0,0,0,0,0,0,220,0,550000,0,235,235,235,0,0,0,0,0,0,0,0,0,0,0,0,256.3636364,235,0,220,0,0,0,0,0,0,11,2585,0,0,-1,-1,10,20,0,0,0,0,0,0,0,0,BENIGN