МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

шифр «**Master 1**»

Наукова робота на тему:

МЕТОД ТА АЛГОРИТМИ ВИЯВЛЕННЯ ПОРУШЕННЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ ПРИ DDOS-АТАКАХ

2019

ЗМІСТ

ВСТУП

Актуальність роботи. DDoS-атака - розподілена атака, спрямована на відмову в обслуговуванні. В результаті атаки такого типу мережевий ресурс, що атакується, отримує лавиноподібну кількість запитів, які не встигає обробити сервер. Джерелом шкідливих запитів є так звані зомбі-мережі, що складаються переважно з комп'ютерів звичайних користувачів, в силу якихось причин заражених шкідливим ПЗ.

Щорічно різні компанії, що надають послуги в галузі забезпечення інформаційної безпеки і протидії кібер-атакам, фіксують збільшення кількості DDoS-атак і їх потужність. Періодичні повідомлення в засобах масової інформації про недоступність тих чи інших ресурсів в результаті розподілених атак, спрямованих на відмову в обслуговуванні, говорять про неефективність засобів протидії такого роду атак. На фоні зазначених вище атак на провідні IT-корпорації також збільшується кількість атак і до невеликих, «середніх» сайтів, які до недавнього часу не становили інтересу для зловмисників. Однак, в даний час, у зв'язку зі збільшенням їх важливості і затребуваності, перебої в їх роботі можуть бути критичними. Разом з цим змінюються і мотиви, які рухають зловмисниками, якщо раніше серед причин виникнення DDoS-атак можна було виділити протест, хуліганство і т.д., то сьогодні все частіше DDoS-атаки є наслідком шантажу і способом вимагання грошей. Це переводить DDoS-атаки з площини одиничних протестних акцій в область кримінального бізнесу, які не обмежується вимаганням, але і є інструментом екстремістських і терористичних організацій [8].

Для паралізації невеликого регіонального ресурсу досить невеликої за потужністю атаки і як наслідок невеликої бот-мережі. Обслуговування та підтримка таких бот-мереж є менш витратним, і потенційно створити такі мережі може більшість зловмисників. Цей факт на фоні відсутності адекватних засобів протидії робить загрози безпеки регіональних ресурсів в результаті DDoS-атак

особливо значущими. Засоби протидії, спеціалізовані саме на забезпечення безпеки невеликих і середніх ресурсів, отримали менший розвиток через переважання в минулому саме великих атак. І в даний час відстають від еволюції самих DDoS-атак [10].

Метою дослідження є створення актуальною методу та інструментарію для раннього виявлення розподілених атак, спрямованих на відмову в обслуговуванні, і подальшого виявлення шкідливого трафіку на стороні ресурсу, що атакується і його блокування власними силами.

Для досягнення зазначеної мети в роботі поставлено і вирішено такі задачі:

1. Проведено моніторинг сучасних DDoS-атак. Виявлено тенденцію до розвитку атак середньої і малої потужності, спрямованих на регіональні ресурси.

2. Досліджено особливості DDoS-атак регіонального рівня. Вироблені вимоги до методу та засобів з виявлення атак і подальшої їм протидії.

3. Вирішено завдання по створенню методу і програмного комплексу по виявленню DDoS-атак і шкідливих запитів.

Об'єктом дослідження є комп'ютерні мережі і розподілені атаки, спрямовані на відмову в обслуговуванні, що здійснюються в цих мережах.

Предметом дослідження виступають моделі та методи виявлення розподілених атак, спрямованих на відмову в обслуговуванні, і виділення шкідливого трафіку цих атак.

В якості основних методів дослідження використаних в роботі, застосовувалися методи теорії ймовірності та математичної статистики, кластерного і системного аналізу, методи машинного навчання.

Наукова новизна досліджень: розроблено метод раннього виявлення та протидії розподіленим атакам, спрямованим на відмову в обслуговуванні. Особливостями методу є: врахування сезонних періодів, орієнтація використання на кінцевому ресурсі, універсальність.

Основні результати:

1. Метод виявлення та блокування шкідливого трафіку DDoS-атак грунтується на аналізі даних мережевого трафіку і формальному описі сезонності. Метод дозволяє визначати шкідливий трафік на ранньому етапі початку атаки і з високою точністю.

2. Алгоритм раннього виявлення початку DDoS-атаки, в середньому виявлення відбувається в чотири рази швидше.

Програмне забезпечення для виявлення початку атаки і блокування шкідливого трафіку, розроблене на основі зазначених алгоритмів, дозволяє організувати ефективний захист від DDoS-атак середньої потужності силами сервера, що атакується.

Обгрунтованість і достовірність поданих у роботі положень і результатів забезпечується за рахунок коректної постановки задачі, ретельного аналізу поточного стану досліджень в даній області, строгістю застосування математичних моделей і несуперечністю отриманих результатів, а також теоретичної апробацією в результаті наукових публікацій, виступів і практичним застосуванням отриманих результатів.

Практична значимість дослідження полягає в створенні методу і алгоритмів забезпечення безпеки мережевих ресурсів від DDoS-атак, що дозволяють проводити активну протидію безпосередньо на стороні ресурсу, що атакується, і в можливості практичного використання розроблених методів і алгоритмів для підтримки безпеки роботи мережевих ресурсів. Це підтверджено розробкою і наступним впровадженням розробленого програмного комплексу по виявленню розподілених атак, спрямованих на відмову в обслуговуванні, і подальшого блокування шкідливих запитів на різних рівнях. Отримані результати можуть бути використані при дослідженнях в суміжних областях, а також при розробці та створенні нових програмних і програмно-апаратних комплексів щодо забезпечення безпеки від DDoS-атак.

Апробація результатів роботи. За темою дослідження опубліковано 1 наукова стаття і 2 тези доповідей, 2 акти впровадження.

Структура і обсяг роботи. Робота складається зі вступу, основної частини, що містить 6 розділів, висновків і списку використаних джерел. Загальний обсяг роботи - 47 сторінок. Робота містить 7 ілюстрацій. Список використаної літератури включає 32 бібліографічних джерела.

1 АНАЛІЗ МЕТОДІВ І ЗАСОБІВ ПРОТИДІЇ DDoS - АТАКАМ

DDoS-атака - абревіатура від англійського Distributed Denial of Service, розподілена атака, спрямована на відмову в обслуговуванні. Атаки такого типу можуть швидко виснажити мережеві ресурси або потужності сервера, що призведе до неможливості отримати доступ до ресурсу і викличе серію негативних наслідків: втрачений прибуток, неможливість скористатися послугами і зробити різні транзакції і т.д. [1]. У DDoS-атаці в ролі атакуючого виступає так звана бот-мережа, або зомбі-мережа. Зомбі-мережа може налічувати від декількох десятків до тисяч хостів. Зазвичай це нейтральні комп'ютери, які в силу якихось причин (відсутність файрволу, застарілі бази антивіруса і т.д.) були заражені, шкідливими програмами. Програми, працюючи у фоновому режимі, безперервно посилають запити на сервер що атакується, виводячи його таким чином з ладу [2]. На даний момент не існує універсального засобу для протидії DDoS-атакам. Навіть такі великі компанії, як Microsoft, eBay, Amazon, Yahoo, страждають від DDoS-атак і не завжди можуть з ними впоратися [2].

Методи захисту від DDoS-атак можна розділити на дві групи: це методи, що передують початку атаки, які спрямовані на запобігання самого факту атаки і методи, які застосовуються вже після початку атаки, це методи активної протидії і пом'якшення результатів атаки.

До методів щодо запобігання атаки можна віднести організаційно правові заходи. Наприклад, неприпустимість залучення в конфліктні ситуації, або заходи, спрямовані на ліквідацію результатів, яких хоче досягти зловмисник, наприклад, розмежування та маскування критичних ресурсів.

Також на цьому етапі реалізується усунення вразливостей і підтримка задіяного апаратно-програмного комплексу в актуальному стані. Деякі види мережевих атак спрямовані саме на експлуатацію різного роду вразливостей. Це можуть бути вразливості в програмному забезпеченні сервера, або вразливості,

пов'язані з використанням не оптимізовані програмних скриптів, які можуть надмірно витрачати ресурси сервера. В даному випадку для досягнення переслідуваного ефекту зловмисникові потрібно організувати менш потужну атаку.

Після початку атаки використовуються активні заходи, спрямовані на протидії атаки [2]. Основними з цих заходів є нарощування ресурсів і фільтрація трафіку. Нарощування ресурсів передує детальний аналіз завантаженості сервера і мережевого сегмента з метою забезпечення вузьких місць. Так, наприклад, якщо в нормальному робочому режимі сервер витрачає значну частину каналу зв'язку, можна припустити, що в разі початку атаки, зловмисник може добитися повного заповнення каналу шкідливими запитами. У цьому випадку доцільно завчасно збільшити пропускну здатність каналу зв'язку.

Виділені в результаті аналізу «вузькі місця» наділяються додатковими ресурсами. Якщо основним споживачем ресурсів на фізичному сервері є сервер баз даних, можливо, доцільно розмістити його на окремо виділеному сервері або ж навіть створити розподілений кластер серверів баз даних. Аналогічно можна вчинити і з іншими сервісами. Якщо продовжити розгляд в якості суб'єкта, що атакується web-сервер, то, крім баз даних, підвищене навантаження може генерувати сам web-сервер або ж розміщенні на ньому скрипти. В цьому випадку необхідно збільшити ресурси самого сервера: додаткова пам'ять, більш потужний процесор і т.д. Або ж за допомогою спеціальних інструментів виділити веб-сервер в окремий кластер. Наприклад, це можна зробити за допомогою використання зв'язки web-серверів Apache та Nginx.

Такий підхід до збільшення ресурсів не є панацеєю від мережевих атак, крім того, має ряд мінусів: нарощування ресурсів пов'язано зі зміною апаратного комплексу і не може бути оперативно проведено в момент початку атаки; підтримка надлишкових ресурсів економічно недоцільно в період очікування атаки.

Для подолання цих мінусів оптимальним є використання хмарних технологій, які дозволяють нарощувати ресурси в міру потреби. Так, наприклад, на сьогоднішній момент на ринку представлені пропозиції від хостинг-провайдерів з надання послуг хмарного хостингу [23]. В результаті надання послуги клієнтам надається необхідне в даний момент кількість ресурсів. У разі збільшення навантаження, це може бути, як збільшення числа легітимних користувачів, так і збільшення шкідливих запитів, відбувається надання додаткових потужностей, що дозволяють обробити кожен запит. В результаті не відбувається відмови сервера. Єдиним мінусом даного підходу є його економічна складова. Так як клієнтові хмарного хостингу потрібно оплачувати додаткові потужності, тобто по суті, платити за обробку шкідливих запитів.

Інший підхід збільшення ресурсів дозволяє нарощувати ресурси в рамках мережі доставки контенту (CDN - абревіатура від англійського, Content Delivery Network) [24]. В результаті реалізації такого підходу відбувається кешування вмісту web-сервера і його доставка через розподілену мережу вузлів. При виборі оптимального маршруту враховується, як правило, географічне розташування клієнта мережі по відношенню до найближчого вузла. В результаті мережевих атак шкідливий трафік ділиться між різними вузлами мережі і втрачає свою потужність, також можливий варіант, коли шкідливий трафік просто не потрапить на вузол, в більшості своїй використовуваний легітимними користувачами. Наприклад, в разі атаки з зомбі-мережі, що знаходиться за кордоном, шкідливі запити будуть концентруватися на найближчому до них вузлі CDN-мережі.

У будь-якому випадку використання цих підходів безпосередньо недоцільно, так як має на увазі обробку всіх запитів без винятку, в тому числі і зловмисних.

Тому наступною групою методів, спрямованих на запобігання атаки, є методи, пов'язані з фільтрацією трафіку. Блокування свідомо шкідливих запитів і обробка надійних і підозрілих дозволяють істотно заощадити засоби, що виділяються на збільшення ресурсів.

Для фільтрації шкідливого трафіку застосовуються різні програмні і апаратні засоби, які базуються на кількісному і якісному аналізі трафіку. В основі методів аналізу, даних засобів, лежать методи кластерного аналізу і математичної статистики, теорії ймовірності, поведінкові методи і т.д.

Для ефективних заходів протидії і фільтрації трафіку необхідно рішення двох тісно пов'язаних задач. Перше завдання пов'язана з виявленням факту початку атаки, друга - з визначенням джерела атаки, тобто джерела шкідливого трафіку. Чим точніше будуть вирішені ці завдання, тим ефективнішими будуть заходи протидії. На сьогоднішній день існують два підходи, пов'язані з визначенням початку атаки. Це підхід, заснований на аналізі зловживань, і підхід, заснований на аналізі аномалій. У першому підході виявлення атаки відбувається шляхом порівняння даних, що характеризують поточний стан системи, з даними, характерними для типових атак. Другий підхід оцінює поточний стан системи з її нормальним станом. Обидва ці підходи мають свої мінуси, так, наприклад, перший підхід може бути неефективним при виявленні принципово нових типів атак. Це особливо актуально в контексті саме DDoS-атак, так як зловмисники намагаються перейти від аномальної поведінки і симулювати дії реальних користувачів. Для успішної експлуатації другого підходу необхідно накопичення статистичних даних, що свідчать про нормальне функціонування системи. Таким чином, для побудови ефективної системи виявлення доцільно використовувати обидва ці підходи.

В результаті роботи такої системи відбувається постійний збір даних, що характеризують стан системи, потім їх обробка та аналіз на предмет відмінності від модельних даних. У разі початку атаки задіюються механізми виявлення джерела трафіку. Проводити порівняння з модельними даними можна різними методами.

До найбільш простих методів, відносяться методи, реалізовані на основі правил. Суть цих методів полягає в установці певних правил, що характеризують

нормальну і аномальну поведінку системи. Правила можуть описувати як поведінку системи в цілому, так і поведінку її окремих частин, наприклад, частоту запитів, певний набір полів запиту і т.д. При достатній простоті і легкості використання, дані методи, проте, досить ефективні.

До найбільш популярних методів, можна віднести групу методів, заснованих на кількісному аналізі. Методи даної групи намагаються виявити атаку по зростаючій навантаженості. Серед цієї групи методів можна виділить, такі: метод MULTOPS аналізує співвідношення прийнятих і відправлених пакетів; метод MIB variables враховує кількість пакетів, їх тип і кількість запитів; методи ACC враховують кількість пакетів з різних підмереж; в Network-Aware Clustering відбувається угруповання запитів, що входять по підмережах і їх порівняння; в Hop-Count Filtering ведеться облік відстаней в хопах (скачках) до підмереж для фільтрації пакетів з помилковою адресою відправника; метод Gateway based, розділяє проходящий трафік на потоки на основі величини «вражаючого впливу»; D-Ward перевіряє легітимність трафіку за такими протоколами: по протоколу TCP - кількість пакетів TCP-ACK, по протоколу ICMP - кількість пакетів ICMP, по протоколу UDP - кількість з'єднань і пакетів в з'єднанні.

Також до перспективних методів можна віднести методи, засновані на виявленні відхилень щодо змін імовірнісних параметрів даних. Їх суть полягає в наступному. Береться часовий ряд певного параметра станів системи, що захищається (набір величин, порахованих за певну кількість інтервалів часу поспіль). Значення цього параметра розглядаються як випадкові величини, отримані по якомусь закону розподілу. Робиться припущення, що в момент атаки вид цього розподілу змінюється, тобто змінюються імовірнісні параметри набору даних. Існує ряд методів для виявлення таких «точок переходу» (change-point detection).

Даний метод використовується в Active Distributed Defense System, Improved D-Ward, Source IP address monitoring і SYN flooding CUSUM detection. У першому

методі аналізується зміна кількості нових IP-адрес від вхідних з'єднань. Останні три методи засновані на методі CUSUM, який дозволяє ітеративно відслідковувати зміну заданого параметра, виявляючи «точки переходу». Суть методу полягає в наступному. Якщо певний параметр протягом декількох періодів часу був вище норми, то задіюються захисні заходи. У Improved D-Ward цим параметром є співвідношення потоку TCP-пакетів від джерела до потоку підтверджень від одержувача, в Source IP address monitoring - кількість нових IP-адрес, а в SYN flooding CUSUM detection - співвідношення пакетів TCP SYN-FIN (RST).

Менше поширення набули методи виявлення атак за допомогою витягнення даних (Data Mining). До таких методів можна віднести, наприклад, методи, які використовують ієрархічної системи різних навчаючих класифікаторів. Серед досліджень можна виділити метод, запропонований в статті «Розробка системи виявлення розподілених мережевих атак типу «Відмова в обслуговуванні»». Суть методу полягає в ймовірнісній оцінці втрат заявок в мережі. У статті «Виявлення розподілених атак на інформаційну систему підприємства» пропонується підхід, заснований на багатоагентному моделюванні. У статті «Активні методи виявлення SYN-flood атак» запропонована модифікація методу активного зондування DARB, яка оцінює напіввідкриті з'єднання на підставі загального мережевого завантаження.

Більшість описаних систем, що використовують модельні дані, мають два режими функціонування: режим навчання (побудова моделі або налаштувнні порогових параметрів) і режим виявлення.

Зі спеціалізованого обладнання для запобігання атакам і фільтрації трафіку, можна виділити систему запобігання атак Proventia Network IPS від компанії IBM. Даний апаратно - програмний комплекс має наступні можливості: обробка понад 200 протоколів різних типів і рівнів; контроль витоків інформації; різні способи реагування; близько 3000 різних алгоритмів для аналізу; забезпечує захист взаємодіючих в корпоративній мережі комп'ютерних систем за допомогою агента

IBM Proventia Desktop; підтримка користувальницьких сигнатур; всілякі варіанти реагування на вторгнення; розміщення підозрілого трафіку в карантин; продуктивність IPS для швидкостей до 40 Гбіт/сек і підтримка IPS для Crossbeam; підтримка 10 Гбіт інтерфейсів за допомогою Network Security Controller і Crossbeam.

Компанія Cisco пропонує своє рішення для запобігання мережевих атак на базі Cisco Guard. Цей апаратно-програмний комплекс, взаємодіючи з детекторами аномалій трафіку в режимі реального часу, перенаправляє трафік, призначений для цільового пристрою, здійснює його аналіз та фільтрацію.

Як правило, такі засоби припускають захист всієї мережі в цілому і зазвичай недоцільні для побудови захисту одиничного кінцевого ресурсу. У зв'язку з цим в даний момент великого поширення набули засоби захисту і протидії, що встановлюються в проміжних мережах. Ефективність цих засобів безпосередньо пов'язана з кількістю вузлів, з яких вони складаються. Таким чином, ефективні розподілені системи мають високу економічну вартість. Власниками таких системам є або великі компанії, або спеціалізовані сервіси з протидії мережевим атакам. Наприклад, подібну послугу по фільтрації трафіку від шкідливих запитів пропонує компанія «Лабораторія Касперського» в рамках надання послуги Kaspersky DDoS Prevention. В рамках розподіленої мережі доставки контенту Cloud Flare також реалізована подібна можливість. Використання таких сервісів з протидії атакам досить ефективно. Економічно вигідніше, в порівнянні з особистим використанням засобів протидії. Однак, передбачає включення проміжної ланки, власне, самого сервісу протидії між клієнтом і сервісом.

Серед цих систем можна виділити дві групи програмних засобів: засоби, що діють на рівні операційної системи, сервера що атакується; засоби, що діють на рівні додатку.

До першої групи засобів можна віднести програмні фаєрволи і спеціалізовані засоби.

Conlimit. Модуль для iptables, в * nix системах, дозволяє налаштувати різні ліміти для підключень. Наприклад, не більше одного одномоментного підключення з однієї адреси.

Ddos deflate. Аналог Colimit. У разі перевищення обраних лімітів блокує доступ на заданий час.

До другої групи можна віднести засоби, які працюють лише на рівні атакованих додатків. Це можуть бути різні плагіни або додаткові модулі, наприклад, для web-серверів, баз даних, інших мережевих сервісів.

Mod_evasive для web-сервера Apache дозволяє проводити блокування на підставі різних правил: обмеження за адресою, кількістю запитів, можливість вибрати тривалість блокування і т.д.

Ngx_http_limit_zone_module аналогічний модуль, але вже для web-сервера Nginx. Володіє подібним функціоналом блокувань.

2 ПОСТАНОВКА ЗАДАЧІ

В результаті проведеного дослідження відмічено, що в даний час значно збільшилася кількість DDoS-атак середньої і малої інтенсивності, спрямованих, як правило, на регіональні ресурси. Це збільшення цілком прогнозовано - з розвитком мережі збільшується потенційна кількість можливих жертв. Крім того, вдосконалюється сам механізм проведення атак. Для зловмисника проведення атаки вже не є настільки складним. А зомбі-комп'ютери намагаються емулювати дії самих користувачів. Все це веде до загального збільшення числа атак.

Аналіз засобів протидії показав, що в даний час більший розвиток отримала група засобів протидії, призначена для відбиття потужних атак. У цю групу входять, як правило, дорогі засоби, призначені для великих провайдерів або компаній. Засоби протидії невеликим і середнім атакам, що базуються на атакуємий сервер, представлені в незначній кількості. Це пов'язано з незначною кількістю таких атак в минулому.

При цьому аналіз вхідного трафіку на рівні додатків може бути більш ефективним. З одного боку, проведення такого аналізу економічно затратно, з іншого - може бути цілком достатнім для відображення малих і середніх атак, тенденція домінування яких вже намітилася.

3 МОДЕЛЬ ВИЯВЛЕННЯ ПОЧАТКУ АТАКИ І ШКІДЛИВОГО ТРАФІКУ

Оптимальним рішенням для виявлення початку атаки і подальшого виявлення шкідливого трафіку буде рішення, засноване на аналізу аномалій, в результаті якого відбувається порівняння поточного стану системи з її нормальним станом. Порівняння станів системи в контексті DDoS-атак можна проводити шляхом порівняння різних властивостей мережевої активності. До цих властивостей можуть бути віднесені: кількість запитів, тип запитів, кількість запитів певного типу або протоколу, IP адреса джерела, швидкість надходження запитів, їх час і т.д.

Нехай множина $A(a_1, a_2, a_3, ..., a_n)$ - набір всіх можливих властивостей для всіх мережевих клієнтів. Множина $B(b_1, b_2, b_3, ..., b_m)$ - множина легітивних клієнтів конкретного мережевого ресурсу. Кожен мережевий клієнт має набір індивідуальних властивостей. Наприклад, клієнт b_1 має властивості $A1(a_4, a_8, a_{10}, a_{14})$, клієнт b_2 має властивості $A2(a_3, a_8, a_{11}, a_{14})$ і т.д. Дані властивості представляють набір підмножин множини *A*. Перетин всіх цих підмножин характеризує клієнтів мережевого ресурсу, за якими вони можуть бути класифіковані. Точно так нелегітивні клієнти матимуть свій набір властивостей, за яким вони також можуть бути класифіковані.

На сьогоднішній день DDoS-атаки ускладнюються, і зловмисники намагаються повністю імітувати поведінку легітивних клієнтів. У цій ситуації перевагу при аналізі властивостей мережевої активності необхідно віддати тим властивостям, які не можуть бути підроблені зловмисниками.

Таким чином, завдання по визначенню і виявлення шкідливих запитів в контексті даної роботи зводиться до їх класифікації на підставі властивостей мережевої активності. Оптимальним рішенням для виявлення шкідливого трафіку є використання різних класифікаторів і нейронних мереж. Складністю в реалізації

даного рішення є той факт, що для нормального функціонування класифікатора потрібно мати дві актуальні навчальні вибірки, відповідно шкідливому і легітивному трафіку. Однак до моменту початку атаки отримати ці вибірки не представляється можливим.

Для подолання цієї проблеми необхідно точно визначити точку початку атаки. Це дасть можливість весь попередній трафік віднести до легітивного і відкриє додаткові можливості по розділенню змішаного трафіку, який приходить після початку атаки, на легітивний і шкідливий. В цьому випадку методика виявлення шкідливого трафіку, в першому наближенні, буде зводитися до наступних кроків: визначаємо актуальні сезонні періоди; з урахуванням сезонності визначаємо точку початку атаки; відносимо весь попередній перед початком атаки трафік до легітивного; класифікуємо змішаний трафік на легітивний і шкідливий; порівнюємо легітивний трафік виділений зі змішаного з трафіком що надійшов до початку атаки; на підставі результатів, отриманих в попередньому кроці і вироблених критеріїв успішності, коригуємо вибірки; весь вступник трафік аналізуємо з урахуванням отриманих даних.

Початок DDoS-атаки пов'язаний зі збільшенням числа запитів до атакуємого сервера. Таким чином, для фіксації факту атаки необхідно встановити границю по кількості запитів до сервера, при порушенні якої однозначно буде фіксуватися нештатна ситуація. Такою границею може виступати максимальна кількість запитів до сервера, плюс деякий запас можливих запитів. Можливість установки граничної межі, після якої буде відбуватися оповіщення адміністраторів, активація необхідних модулів і т.д., реалізована в різних мережевих засобах як програмного, так і апаратного рівня. При цьому такий підхід має ряд мінусів:

1. Для запобігання випадкових спрацьовувань, межа що встановлюється повинна бути істотно вище максимального рівня кількості запитів. Що, в свою чергу, призводити до виникнення похибки при виявленні атаки.

2. Мережевий ресурс може відчувати різне навантаження в залежності від часу доби і днів тижня. В цьому випадку атака, що почалася в період затишшя, наприклад, у вихідний день або вночі, буде зафіксована із запізненням. Якщо в системі запобігання вторгнень передбачено використання класифікаторів та навчання фільтрів на підставі вхідного трафіку, є ймовірність в їх негативному навчанні.

Для вирішення зазначених проблем необхідно використовувати ковзаючу оцінку, що характеризує поточну мережеву активність. На підставі цієї оцінки встановлювати динамічну границю, актуальний для періоду можливого початку атаки. В якості ковзаючої оцінки можливо використовувати середньоквадратичне відхилення:

$$\sigma = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} \left(x_i - \bar{x} \right)^2}, \qquad (3.1)$$

де σ - середньоквадратичне відхилення; n - кількість розглянутих часових періодів; x_i - кількість запитів за і-період; $\bar{x}$ - середнє арифметичне запитів по всіх періодах. В результаті експериментів, було встановлено, що для різних сайтів оптимальне значення верхньої межі може відрізнятися і перебувати, як правило, в діапазоні від 2.2σ до 2.9σ. З цієї причини, для більш гнучкого налаштування програмного забезпечення по виявленню початку DDoS-атаки, цей параметр задається не жорстко. У оператора програмного комплексу є можливість варіювати значення даного параметра. Однак такий підхід також має потенційну вразливість, пов'язану з тим, що зловмисник може поступово нарощувати потужність атаки, зрушуючи при цьому границю середньоквадратичного відхилення. усунути дану вразливість може облік сезонних коливань.

Нехай мережевий ресурс зазнає сезонне добове навантаження x_i - кількість запитів до мережного ресурсу за одну годину. Кількість добових періодів n. Тоді запити до мережного ресурсу можна виписати у вигляді такої матриці:

$x_{11},\ x_{12},\ x_{13}, \dots, x_{124}$

$x_{21},\ x_{22},\ x_{23}, \dots, x_{224}$

.....

$x_{n1},\ x_{n2},\ x_{n3}, \dots, x_{n24}$

Кожен рядок матриці включає добові дані про кількість запитів. Перший рядок відображає дані поточної доби, в зв'язку з цим вона може бути заповнена не до кінця. Розрахунок середньоквадратичного відхилення в цьому випадку може проводитися двома способами.

1. Звичайним способом - з урахуванням певного числа останніх значень, наприклад, так:

$x_{21},\ x_{22},\ x_{23}, \dots, x_{224}, x_{11},\ x_{12},\ x_{13}, x_{14}$

Значення беруться з рядків матриці.

2. З урахуванням сезонності - розрахунок проводиться за стовпцями.

$x_{n1}, \dots, x_{21},\ x_{11}$

Якщо спостерігач знаходиться в *i*-тому періоді, можна розрахувати границю для $i + 1$ періоду, використовуючи значення $i + 1$ стовпчика. Якщо мережевий ресурс зазнає навантаження, пов'язане з тижневими або добовими циклами, то необхідно виключити рядки, які відповідають святковим і вихідним дням. Або навіть використовувати тільки кожен сьомий рядок, тобто порівнювати, наприклад, тільки період з 13:00 до 14:00 для кожного вівторка. Такий підхід дозволяє формувати досить точну верхню межу, порушення якої може бути витлумачено як виникнення мережевої аномалії. Збільшення точності дозволяє зменшити час, необхідний для виявлення атаки, і досить точно зафіксувати її початок (рис. 3.1).

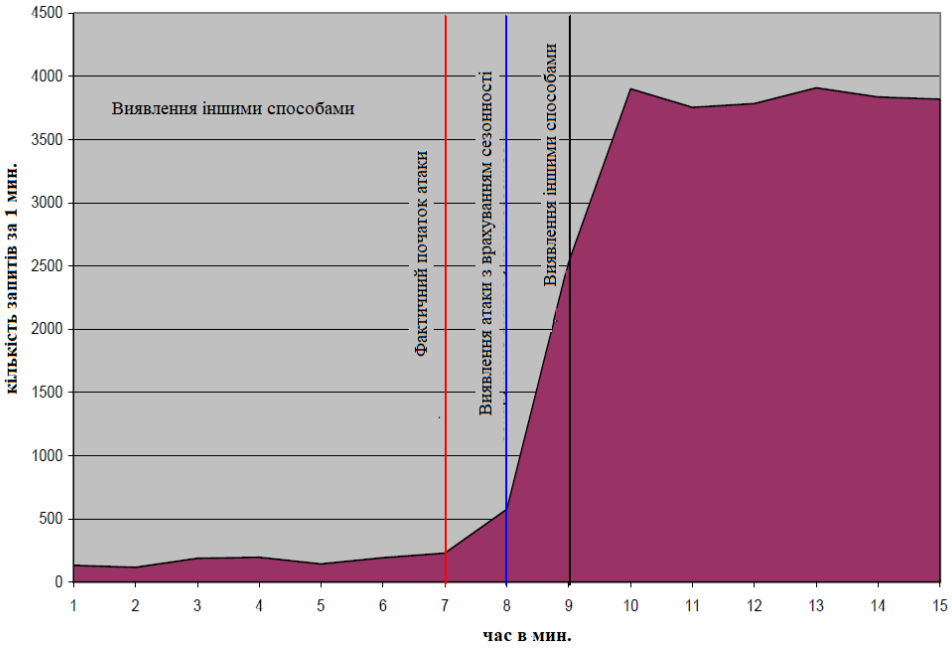

Рисунок 3.1 - Стандартний початок атаки

Крім того, в рамках такого підходу виключаються можливості негативного навчання фільтрів і спрацьовування системи виявлення з запізненням шляхом поступового нарощування потужності атаки (рис. 3.2). Так як межа в цьому випадку буде будуватися за схожими сезонним періодами. Наприклад, поступове нарощування потужності атаки протягомдня буде зафіксовано при порівнянні кількості запитів з кількістю запитів актуальних сезонних періодів за минулу добу.

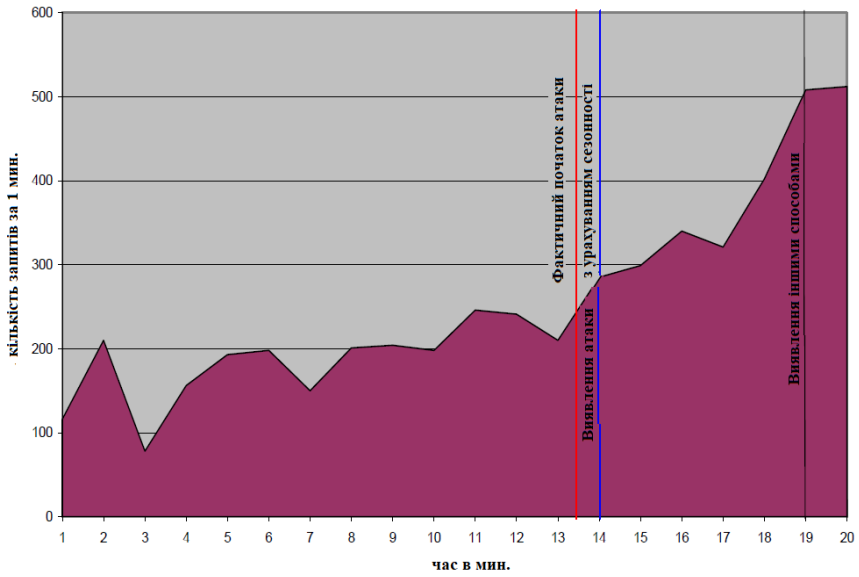

Рисунок 3.2 – Поступове нарощування потужності атаки

Виявлення і дослідження сезонності. В рамках раннього виявлення початку атаки, і з огляду на перспективність підходу необідно враховувати сезонні коливання, провести додаткове дослідження, що вивчає сезонні коливання кількості запитів до мережевих Internet ресурсів. Основним завданням дослідження було доказ існування сезонних періодів в роботі web-сайтів. А також вирішення питання, чи може випадковий сплеск в відвідуваності Internet ресурсу, викликаний, наприклад, публікацією на нього посилання з високо відвідуваного ресурсу, викликати порушення сезонності, і, як наслідок, помилкове спрацьовування.

Всього в рамках дослідження було проаналізовано статистика 60 різних сайтів. В результаті аналізу було об'єктивно встановлено наявність сезонних періодів в роботі web-сайтів. Наприклад, якщо розглядати місячну відвідуваність одного з найбільших новинних порталів - Інформаційне агентство «Амітел» (сайт http://amic.com), то в його роботі чітко видно тижневі сезонні періоди (рис. 3.3). Спостерігається сплеск перегляду сторінок на початку робочого тижня, спад переглядів протягом усього тижня і низька активність користувачів на вихідних.

Якщо розглянути протягом доби кількість переглядів, то так само видно наявність сезонних періодів (рис. 3.4).

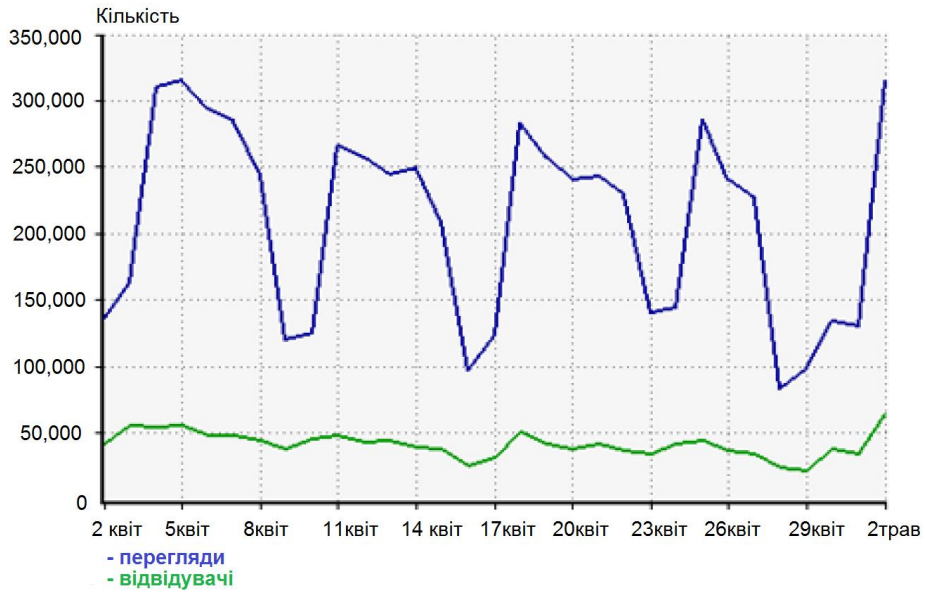

Рисунок 3.3 - Графік відвідуваності і перегляду сторінок, сайту ІА «Амітел»

Кількість переглядів мінімально вночі, зростає з початком робочого дня і зменшується до вечора.

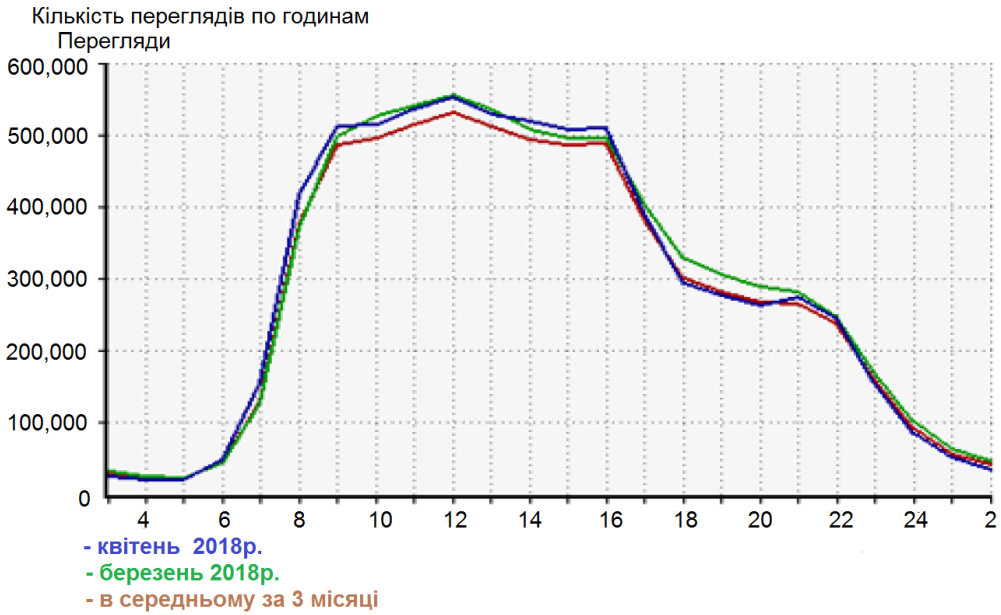

Рисунок 3.4 - Кількість переглядів сторінок сайту ІА «Амітел» по годинах

Аналогічну картина видно і на графіку, що характеризує добову активність користувачів за останні два місяці і в середньому за три місяці.

4 МЕТОД КЛАСТЕРИЗАЦІЯ НА ОСНОВІ АЛГОРИТМУ K-MEANS

Вибір і обгрунтування методу кластеризації. Нехай безліч *T* - множина клієнтських запитів, надійшли до початку атаки. Множина клієнтських запитів, що надходять після початку атаки - це об'єднання множин *H* - шкідливих клієнтських запитів і множина T* - надійних клієнтських запитів. Таким чином, для отримання вибірки, що характеризує шкідливий трафік необхідно буде розділити трафік, одержуваний після початку атаки, на дві групи.

Так як число кластерів заздалегідь відомо, в якості гіпотези можна припустити, що оптимальним буде використання алгоритму k-means. Однак, використання даного алгоритму буде ресурсно затратним при обробці великих об”ємів даних і потребують додаткових ресурсів. Якщо враховувати, що даний метод буде використовуватися в рамках сервера, що атакується, який вже відчуває нестачу в обчислювальних ресурсах, то процес кластеризації даним методом може бути нездійсненний. Також особливістю даного алгоритму є чутливість до викидів, які можуть спотворювати середнє. І хоча при дослідженні сезонності було встановлено вкрай рідкісна поява великих викидів, їх потенційна наявність може негативно позначитися на результатах як кластеризації, так і роботи алгоритму в цілому.

Для вирішення проблеми, пов'язаної з викидами, кращі результати може дати модифікація алгоритму k-середніх, алгоритмом k-медіани (kmedoids). Алгоритм менш чутливий до шумів і викидів даних, ніж алгоритм k-means, оскільки медіана менше піддається впливам викидів. Але при цьому алгоритм не ефективний при обробці великих об”ємів даних.

Вирішити всі зазначені вище проблеми можливо використанням алгоритму CLARA (Clustering LARge Applications), який був розроблений Kaufmann і Rousseeuw для кластеризації даних у великих базах даних.

Алгоритм CLARA витягує множину зразків з бази даних. Кластеризація застосовується до кожного із зразків, на виході алгоритму пропонується найкраща кластеризація. Для великих баз даних даний алгоритм ефективніший, ніж алгоритм PAM. Однак, ефективність алгоритму залежить від обраного в якості зразка набору даних, оптимальний вибір якого в ситуації постійно змінюваній мережевій картини може бути утруднений.

З метою вибору оптимального алгоритму кластеризації, також були розглянуті нові, перспективні методи кластеризації. це методи Clarans, CURE, DBScan і метод WaveCluster.

Метод WaveClaster показує хорошу ефективність при аналізі невеликих об”ємів даних, алгоритм не чутливий до шумів і може будувати кластери довільної форми. Але при збільшенні розмірності даних його ефективність різко знижується.

Методи Clarans, CURE, DBScan хоча і призначені для обробки надвеликих об”ємів даних, для вирішення поточної задачі виявилися неефективні. Так як суть цих методів - початкова установка певної щільності точок, для оптимізації роботи з великими об”ємами даних не може бути ефективно виконана в контексті поточної задачі. За підсумками розгляду поданих вище методів кластеризації, був обраний метод k-means. Даний методи був досліджений на предмет ефективності для вирішення поточної задачі кластеризації.

Попередньо, для підтвердження гіпотези про існування двох кластерів була створена візуалізація наявних даних. Візуалізація є двомірна діаграма розсіювання. Зниження розмірності було проведено за допомогою методу головних компонент. Діаграма підтверджує гіпотезу про існування двох кластерів.

Наступним кроком дослідження став вибір оптимальних ознак для проведення кластеризації. В результаті експериментів було встановлено, що найбільшу ефективність при кластеризації забезпечує враування таких ознак: джерело запитів; швидкість надходження запитів; сторінка призначення запиту;

офлайн дані; браузер клієнта; операційна система клієнта; обсяг переданих даних; обсяг отриманих даних.

В цілому, вибрані ознаки, а також їх кількість відповідають результатам, отриманими в аналогічних дослідження. Скорочення розмірності вихідного масиву даних, дозволило збільшити швидкість кластеризації. Наступним питанням став вибір оптимальної кількості даних для аналізу. З огляду на те, що дані для аналізу беруться з лог-файлу, і одному клієнтського запиту може відповідати кілька записів з цього файлу, (при зверненні до однієї сторінки можуть завантажуватися різні зображення, файли каскадних таблиць стилів скриптів і т.д.), вказати точний розмір даних (в кількості рядків), оптимальних для аналізу, не представляється можливим. У зв'язку з цим оптимальний розмір даних вказується в кількості періодів.

Оптимальна кількість періодів для розрахунку середньоквадратичного відхилення рівне 22. При зменшенні цього значення можливі хибні спрацьовування, при збільшенні може відбуватися незначне зростання точності, який при подальшому збільшенні кількості періодів, як правило, змінюється спадом, внаслідок зменшення чутливості (рис. 4.1).

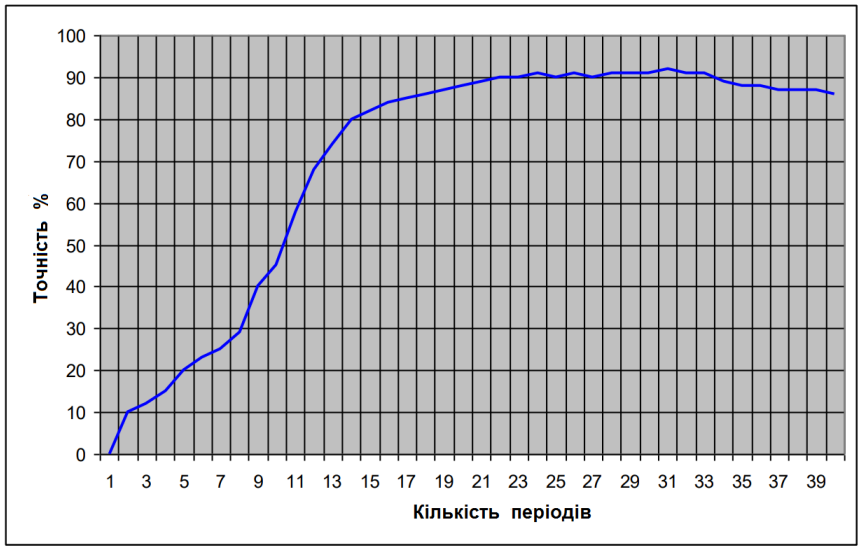

Рисунок 4.1 - Графік залежності точності визначення початку атаки від кількості розглянутих періодів

В результаті дослідження обраний метод показав свою високу ефективність. Метод забезпечує прийнятну точність, мінімальну навантаження і оптимальну швидкість роботи.

Первинна кластеризація на основі алгоритму k-means. Для первинного поділу змішаного трафіку на надійний і шкідливий оптимально скористатися алгоритмом кластеризації k-means. Даний алгоритм дозволяє провести кластеризацію при заздалегідь відомому числі кластерів. Алгоритм має прийнятну точність, необхідну для первинного поділу, і більш високу швидкість роботи, по порівнянню з іншими алгоритмами.

Суть алгоритму полягає в виділення двох кластерів і обчисленні їх центрів мас, на наступних ітераціях відбувається корекція кластерів (перенесення елементів в відповідно до розрахованих центрів мас) і перерахування центрів мас.

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} \left(x_j - \mu_i \right)^2 \qquad (4.1)$$

де *k*- число кластерів, S_i - отримані кластери, i – 1,2, … , k_i , μ_i - центри мас векторів $x_j \in S_i$.

В результаті роботи алгоритму змішаний трафік буде розділений на два кластера, відповідних надійному і шкідливому трафіку. Таким чином, на даному етапі доступні для аналізу і обробки три групи трафіку:

1. Відповідна, надійному трафіку, що передує початку атаки - *T*.

2. Відповідна, надійному трафіку, виділена із змішаної - *T**.

3. Відповідна шкідливому трафіку, виділена із змішаної - *H*.

Критерії успішності, корекція отриманих кластерів. Для оцінки ефективності кластеризації розглянемо рівняння стаціонарних ймовірностей:

$$
\begin{aligned}
&p_0\lambda = p_1\mu \\
&(\lambda + i\mu)\, p_i = \lambda p_{i-1} + (i+1)\cdot \mu p_{i+1},\ i = 1,\ldots,K-2, \\
&(\lambda + (K-1)\mu)\, p_{K-1} = \lambda p_{K-2} + KN\cdot \mu p_K, \\
&(\lambda^* + KN\mu)\, p_K = \lambda p_{K-1} + KN\cdot \mu p_{K+1}, \\
&(\lambda^* + i\mu^*)\, p_i = \lambda^* p_{i-1} + (i+1)\cdot \mu^* p_{i+1},\ i = K+1,\ldots,N-1
\end{aligned}
\qquad (4.2)
$$

де, λ - інтенсивність навантаження, λ_L - інтенсивність навантаження, створювана легальними користувачами, S - інтенсивність шкідливого трафіку, μ - інтенсивність звільнення черги запитів, μ^* - інтенсивність звільнення черги запитів при активованому фільтрі, E_1, E_2 - помилки першого і другого роду, K - межа активації фільтра, N - обсяг черги запитів, $\lambda = \lambda_L + S$ - навантаження в момент атаки, $\lambda^* = \lambda_L + S(1-E_2)$ - навантаження при активованому фільтрі,

Так як перевірка запиту знижує продуктивність сервера, в деяких випадках інтенсивність звільнення черги запитів, можна оцінити як $Z\mu$, де, Z - коефіцієнт уповільнення.

При нормуванні $\sum_{i=1}^{N} p_i = 1$, отримаємо ймовірність блокування запиту.

$$
p_{BLK} = \frac{\frac{1}{N!}\left(\frac{\lambda}{\mu}\right)^{K-1} \frac{\lambda}{\mu^*}\left(\frac{\lambda^*}{\mu^*}\right)^{N-K}}{\sum_{i=0}^{K-1} \frac{\left(\frac{\lambda}{\mu}\right)^{i}}{i!} + \sum_{i=K}^{N} \frac{1}{i!}\left(\frac{\lambda}{\mu}\right)^{K-1} \frac{\lambda}{\mu^*}\left(\frac{\lambda^*}{\mu^*}\right)^{i-K}} \qquad (4.3)
$$

Таким чином, ефективність поділу шкідливого і надійного трафіку можна оцінити, як $R = (1-E_1)\cdot(1-p_B)$.

На підставі отриманої оцінки, були вироблені критерії успішності.

На наступному кроці алгоритму проводиться корекція отриманих вибірок з урахуванням наступних критеріїв:

1. Критерій розмірності отриманих кластерів. Якщо в поточному періоді, що відноситься до атаки, кількість запитів - *n*, а в аналогічних сезонних періодах, що відносяться до надійного трафіку - *m*, то кількість шкідливих запитів буде наближено рівним *n*-*m*. Це ж справедливо і для різних властивостей мережевої активності (кількість запитів до цільової сторінці, цільового порту, за певним протоколом і т.д.)

2. Критерій схожості надійних вибірок. Максимальна схожість надійної вибірки, що передує початку атаки, з надійною вибіркою, виділеної із змішаного трафіку.

3. Критерій відповідності центрів мас. Центр мас надійної вибірки, виділеної із змішаного трафіку, повинен відповідати аналогічного сезонному періоду надійного трафіку, що передує початку атаки. Іншими словами, відстань між цими центрами мас повинен наближатися до нуля.

Для подальшого уточнення можна розрахувати ймовірність приналежності кожного елемента своєму класу. Елементи з найменшою ймовірністю переносяться в протилежні групи з урахуванням критерію розмірності груп.

Для розрахунку схожості надійних кластерів і надалі для класифікації запитів, що надходять можна скористатися «Байєсовким класифікатором». В якості ймовірнісної моделі для класифікатора використовуємо умовну ймовірність $p(C \mid F_1,...,F_n)$ над залежною змінною класу *C* з малою кількістю результатів або *класів*, що залежить від декількох змінних $F_1,...,F_n$.

Використовуючи теорему Байєса, запишемо:

$$p(C \mid F_1,...,F_n) = \frac{p(C) \cdot p(F_1,...,F_n \mid C)}{p(F_1,...,F_n)}$$

Умовний розподіл по класовій змінній *C* може бути виражено так:

$$p(C \mid F_1,...,F_n) = \frac{1}{Z} p(C) \prod_{i=1}^{n} p(F_i \mid C)$$

Таким чином, для класифікації трафіку за двома класами маємо:

$P(T \mid D) = \frac{P(T)}{P(D)} \prod_{i=1}^{n} P(w_i \mid T)$ - для класу надійних користувачів

$P(H \mid D) = \frac{P(H)}{P(D)} \prod_{i=1}^{n} P(w_i \mid H)$ - для класу ненадійних користувачів

В якості навчальних вибірок використовуються множина *T* і множина *H*. Після закінчення цього кроку елементи з множини *T**, віднесені до групи шкідливого трафіку, міняються місцями з елементами множини *H* з урахуванням зазначених вище критеріїв. Даний крок повторюється до тих пір, поки всі елементи множини *T* не будуть позначені як надійні, або поки алгоритм не досягне порогового значення ітерацій.

Отримані вибірки, відповідні надійному і шкідливому трафіку, а також механізм їх підтримки в актуальному стані дозволяють використовувати їх з різними класифікаторами.

На рис. 4.2 показані принципові схеми алгоритмів по визначенню початку атаки і виділенню шкідливого трафіку. Перша схема (рис. 4.2а), пояснює алгоритм виділення шкідливого трафіку, друга (рис. 4.2б) і третя (рис. 4.2в) алгоритми визначення початку атаки.

На першому кроці відбувається виклик підпрограм по виявленню сезонних періодів, розрахунку для них допустимої межі кількості запитів, і визначення початку атаки. У разі початку атаки, алгоритм повинен розподілити змішаний трафік на два кластери, один містить шкідливі запити, інший надійні запити. дані кластери уточнюються. Нові запити аналізуються на приналежність того або іншому кластеру і по результату додаються до відповідного кластеру.

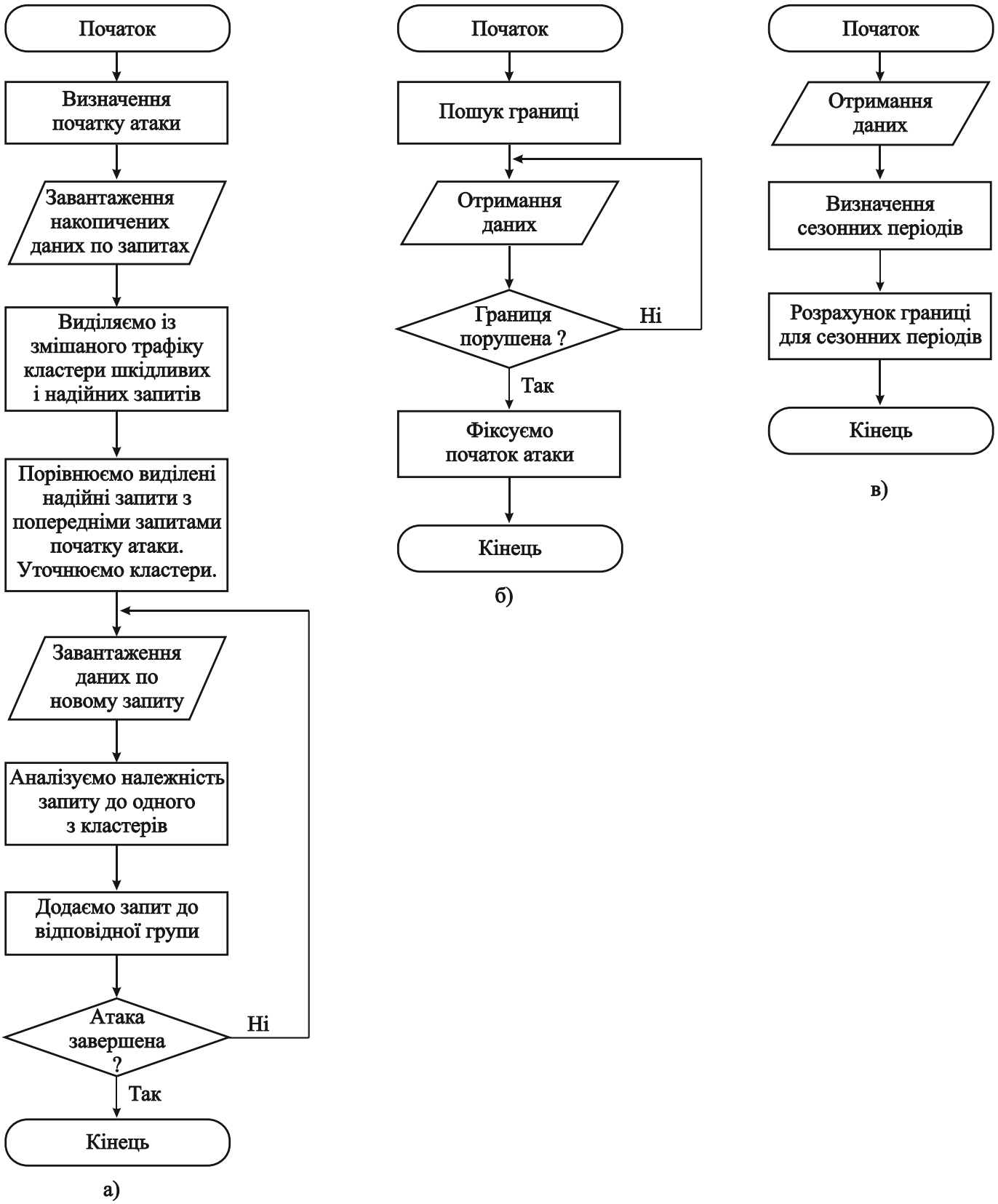

Рисунок 4.2 - Алгоритму по визначенню початку атаки і виділенню шкідливого трафіку

В рамках запропонованого підходу виявлення DDoSатак і шкідливого трафіку розроблений оригінальний алгоритм виявлення на ранніх стадіях точки початку розподіленої атаки, спрямованої на відмову в обслуговуванні. Алгоритм враховує сезонні відхилення в навантаженні, що дає можливість виявляти точку початку атаки на ранніх стадіях і з більшою точністю. Додатково проведено дослідження, спрямоване на підтвердження існування сезонності і виявлення типових сезонних періодів. В результаті дослідження виявлені тижнева, добова і невизначена сезонність і причини її виникнення.

Для поділу змішаного трафіку використовується алгоритм кластеризації k-means. Вибір даного алгоритму обгрунтований, проведено доказ його ефективності. Для алгоритму підібрані оптимальні характеристики і розмірність даних, вироблені критерії успішності. Розроблені алгоритми складають основу узагальненого методу виявлення DDoS-атак і шкідливого трафіку, яка в загальному вигляді може бути записана так:

1. За допомогою статистичних даних, визначаємо існуючі сезонні періоди.

2. Для кожного сезонного періоду визначаємо допустиму верхню межу кількості запитів.

3. У разі порушення границі, фіксуємо точку початку атаки.

4. Відносимо весь, що передує початку атаки, трафік до кластеру, відповідному легітивному трафіку.

5. За допомогою алгоритму k-means класифікуємо змішаний трафік на легітивний і шкідливий.

6. Порівнюємо трафік, що передує початку атаки, з кластером, легітивного трафіку, виділеного зі змішаного трафіку.

7. На підставі результатів, отриманих на попередньому кроці, і з урахуванням вироблених критеріїв успішності, коригуємо кластери.

8. Весь трафік, що надходить, аналізуємо з урахуванням отриманих в попередньому пункті результатів.

5 КОМПЛЕКСНИЙ МЕТОД ФІЛЬТРАЦІЇ ТРАФІКА

У зв'язку зі значним переважанням серед DDoS-атак атак типу HTTP-flood, спеціалізація розробляємого засобу повинна бути орієнтована на протидії атакам цього типу.

На підставі даних моніторингу та ріст DDoS-атак середньої і малої інтенсивності, спрямованих на регіональні ресурси, і відсутності засобів протидії, ефективних і актуальних для даних атак, доцільно створюваний засіб адаптувати для вирішення зазначених завдань в контексті безпеки регіональних ресурсів.

На сьогоднішній день регіональний web-ресурс - це:

1. Ресурс, розташований на власному виділеному сервері середньої конфігурації або у одного з хостинг-провайдерів на тарифному плані середнього рівня.

2. Ресурс, який використовує в якості системи управління вмістом (CMS, content managment system) власну CMS або вільно розповсюдженої CMS з відкритим вихідним кодом.

3. Ресурс, обмежений в матеріальних засобах.

На підставі цих властивостей можна виділити наступні проблеми, що виникають при протидії DDoS-атакам на рівні регіонального web-ресурсу:

1. У зв'язку з відсутністю власної мережі, немає технічної можливості використовувати матеріальні ресурси для протидії DDoS-атакам. Крім того, використання таких засобів для забезпечення безпеки регіонального web-ресурсу економічно невигідно.

2. У зв'язку з відокремленістю, немає можливості використовувати для аналізу дані, отримані на вищих вузлах.

3. Також немає можливості заблокувати трафік на вищих вузлах.

4. Існує обмеження в матеріальних ресурсах сервера. Залучення нових ресурсів і збільшення потужності, представляється складним.

5. Обмеження матеріальних засобів не дозволяє використовувати сторонні засоби для фільтрації трафіку і мережі доставки контенту.

6. Використання власної CMS або адаптованої вільно розповсюджуваної CMS небезпечно наявністю потенційних вразливостей і підвищеною витратою ресурсів, що дає можливість зловмиснику досягти мети при використанні атаки меншої потужності.

На підставі зазначених проблем і потреб можна виділити наступні вимоги, що пред'являються до розробленої системі фільтрації трафіку:

1. Розроблюваний засіб має бути орієнтований на протидію атакам типу HTTP-flood.

2. У разі необхідності засіб може бути використано для протидії DDoS-атак різних типів.

3. Для блокування шкідливого трафіку використовуються засоби та інструменти, доступні в рамках фізичного або віртуального хостингу.

4. Засіб має відповідати вимогам платформ. Використовуватися на різних операційних системах, з різними web-серверами і мережевими сервісами.

5. У процесі роботи засіб повинен генерувати мінімальне навантаження на ресурси сервера, так як в результаті DDoS-атаки мережевий ресурс може відчувати брак в вільних ресурсах.

Як середовище програмування обраний PHP. Вибір PHP дозволяє виконати вимогу платформ, що застосовується до розробляємого засобу. Дане середовище програмування являє собою скриптову мова програмування, інтерпретатор компілюючого типу. PHP є одним з найпопулярніших мов для розробки web-додатків. Даний інтерпретатор підтримується різними web-серверами, розташованими на різних платформах. Інтерпретатор доступний у більшості хостинг провайдерів на самих різних тарифних планах. Таким чином, засіб, розроблений за допомогою PHP, буде відповідати вимогам платформ і функціонувати незалежно від операційної системи і web-сервера.

В якості системи управління базами даних був обраний mySQL. Дане програмне забезпечення також відповідає вимогам платформ. Є вільно поширюваним і доступним у більшості хостинг-провайдерів. Використання в якості СУБД mySQL в рамках даної розробки не є критичним, при необхідності і мінімальних доробках розробляється засіб може використовувати різні реляційні СКБД.

Вибір даних для аналізу. Розроблюваний програмний засіб призначений для виконання на ресурсі що, захищається, таким чином, дані для аналізу обмежені рамками кінцевого сервера. В якості таких даних можуть виступати різні log-файли, наприклад, log-файли web-сервера, сервера баз даних і т.д., або системні log-файли. А також дані, отримані безпосередньо з мережевого інтерфейсу. Якщо мережевий ресурс функціонує у вузьких рамках віртуального хостингу, то єдиними даними, доступними для аналізу, будуть виступати дані, що зберігаються в log-файлах сервера.

На сьогоднішній день одним з найпопулярніших web-серверів є web-сервер Apache. Реалізація програмного засобу буде відбувається на прикладі аналізу даних, отриманих з log-файлів цього сервера. Дані звернень до web-сервера Apache зберігаються в log-файлі access. log, що має наступну структуру –

%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" ,

де:

- % h - хост / IP-адреса, з якого зроблений запит до сервера;

- % t - час запиту до сервера і часовий пояс сервера;

- % r - тип запиту, його вміст і версія;

- % s - код стану HTTP;

- % b - кількість відданих сервером байт;

- % {Referer} - URL-джерело запиту;

- % {User-Agent} - HTTP-заголовок, що містить інформацію про запит (клієнтську програму, мову і т.д.);

- % {Host} - ім'я Virtual Host, до якого йде звернення.

Приклад запису з файлу access. Log представлено в лістингу 5.1.

Лістинг 5.1- Приклад запису з файлу access. log

127.0.0.1 - frank [10 / Oct / 2017: 13: 55: 36 -0700] "GET /apache_pb.gif HTTP / 1.0" 200 2326 127.0.0.1 - frank [10 / Oct / 2017: 13: 55: 36 - 0700] "GET /style.css HTTP / 1.0" 200 3214 127.0.0.1 - frank [10 / Oct / 2017: 13: 55: 36 -0700] "GET /index.php HTTP / 1.0" 200 1406

Подібну структуру файлу access.log використовують і інші web-сервера: Nginx, Lighthttpd і т.д. Крім того, принципова схема файлу access. log, аналогічна log-файлів інших мережевих сервісів.

Комплексний метод фільтрації трафіку включає наступні кроки:

1. З заданим інтервалом часу актуальні дані витягуються з файлу access.log, проходять обробку і завантажуються в базу даних.

2. Дані, завантажені в базу даних, постійно аналізується на предмет початку атаки.

3. У разі виявлення початку атаки виконуються наступні дії:

- проходить оповіщення зацікавлених осіб за допомогою розсилки електронної пошти;

- в автоматичному режимі виконуються скрипти, підготовлені системним адміністратором;

- запускається механізм класифікації трафіку.

4. В результаті роботи механізму класифікації трафіку:

- в базі даних створюються дві таблиці, відповідні надійному і шкідливому

трафіку;

- проходить первинне заповнення таблиць на підставі проведеної кластеризації;

- запити, що приходять знову класифікуються;

- на підставі цієї класифікації уточнюються отримані кластери;

5. Трафік, позначений як шкідливий, блокується.

Для створення гнучкості і платформ розробляється програмний засіб розділений на три модуля:

- модуль обробки даних і їх завантаження в базу даних
- модуль виявлення початку атаки
- модуль фільтрації трафіка.

Засіб обробки і завантаження даних. Виділення засобу обробки і завантаження даних в окремий модуль дозволяє організувати роботу програмного комплексу з log-файлами різних типів і різного змісту. Обробка і приведення даних до одного виду на першому етапі роботи програмного засобу в подальшому дозволяє спростити цей процес і аналізувати їх без внесення будь-яких змін в інші модулі. Таким чином програмний засіб, що розробляється може бути використано не тільки для захисту web-ресурсів від атак типу http-flood, але і для захисту різних мережевих ресурсів і сервісів від DDoS-атак різних типів.

При розробці програмного засобу експорту даних з log-файлів в базу даних особлива увага була приділена навантаженню, яку цей засіб може генерувати. Питання навантаження є досить актуальним, так як засіб призначений для обробки великих об’ємів інформації. Одному зверненню до сервера в log-файлі відповідає, як правило, кілька записів. Якщо web-сайт, розташований на сервері, є популярним, його відвідує велика кількість користувачів, і, якщо ці користувачі переглядають велику кількість сторінок, інформація в log-файлах сервера накопичується з більшою швидкістю. Крім того, можливий варіант, коли в рамках одного фізичного web-сервера підтримується кілька віртуальних. У цьому випадку кількість інформації для експорту в базу даних також збільшується.

Засіб обробки і завантаження даних із заданим інтервалом часу виробляє читання з log-файлу, актуальних даних, призводить їх до необхідного виду і робить завантаження в базу даних.

Кожен рядок в log-файлі відповідає окремим запитам до мережного ресурсу. При отриманні даних з log-файлу відбувається порядкове зчитування даних. Надалі з кожного рядка дані витягуються в масив в або в окремі змінні. Розбір даних з рядка відбувається з використанням функцій для роботи з регулярними виразами, такими як preg_split (), split (). В результаті тестування було помічено, що в деяких випадках ці функції генерують значне навантаження. У зв'язку з цим розроблений додатковий спосіб отримання даних з рядків log-файлу (лістинг 5.2).

Лістинг 5.2 - Фрагмент коду розбору рядка за допомогою функції substr ()

```
$ip=substr($line,0,strpos($line," "));
$line=substr($line,strpos($line," ")+1,strlen($line));
$t1=substr($line,0,strpos($line," "));
$line=substr($line,strpos($line," ")+1,strlen($line));
$t2=substr($line,0,strpos($line," "));
$line=substr($line,strpos($line," ")+1,strlen($line));
$date=substr($line,1,strpos($line,"] ")-1);
$line=substr($line,strpos($line,"] ")+1,strlen($line));
$get=substr($line,2,strpos($line,'" ')-2);
$line=substr($line,strpos($line,'" ')+2,strlen($line));
$t3=substr($line,0,strpos($line," "));
$line=substr($line,strpos($line," ")+1,strlen($line));
$t4=substr($line,0,strpos($line," "));
$line=substr($line,strpos($line," ")+1,strlen($line));
$url=substr($line,0,strpos($line,'" ')+1);
$line=substr($line,strpos($line,'" ')+1,strlen($line));
```

6 АРХІТЕКТУРА ПРОГРАМНОГО КОМПЛЕКСУ ФІЛЬТРАЦІЇ ТРАФІКА

Засіб виявлення початку атаки в режимі реального часу розраховує середньоквадратичне відхилення з урахуванням актуальних сезонних періодів за кількістю запитів до мережного ресурсу в кожному періоді. Програмний засіб дає можливість задати розмірність розглянутих періодів: 1 хвилина, 15 хвилин, 1 годину і т.д. А також вести моніторинг відразу по декількох періодів. На підставі розрахованого середньоквадратичного відхилення задається верхня межа кількості запитів до мережного ресурсу відповідного періоду.

Засіб виявлення початку атаки має гнучкі налаштування, що дозволяють задати чутливість до можливої атаки (лістинг 6.1). Конфігураційні дані виділені в окремий php-файл, що дає додаткові можливості як з точки зору зручності, так і з точки зору безпеки. Чутливість варіюється за допомогою корекції границі, а також порушенням границі відразу в декількох періодах різного розміру. Наприклад, при порушенні границі на хвилинних інтервалах засіб може тільки сповістити зацікавлених осіб про збільшення активності. У разі порушення границі також на п'ятихвилинному інтервалі відбувається повне включення механізму захисту.

Лістинг 6.1 - Фрагмент конфігураційного файлу

```
//час періоду мережевої активності в секундах, 86400 добу, 604800 тиждень
    $Loop = 40400;
//період для дослідження в секундах
    $User_per = 300;
//кількість періодів для аналізу
    $Count_user_per = 100;
//період, який необхідно відступити від початку атаки для позначки
//благополучного трафіку
    $Safe_per = 600;
```

//число в процентах, на яке благополучний трафік повинен відрізнятися від //шкідливого

$Pogresnost = 10.

У разі виявлення початку атаки виконуються наступні дії:

1. Розсилка повідомлень. В автоматичному режимі відбувається розсилка повідомлень електронною поштою.

2. Виконання скриптів. Запускаються скрипти або сторонні програми, підготовлені для виконання системним адміністратором. Це можуть бути як скрипти, що включають додаткові рівні кешування або ж відключають модулі web-ресурсу, що генерують підвищене навантаження, так і системні скрипти та програми.

3. Активація засобів фільтрації трафіка.

Засіб фільтрації трафіка. На підставі розробленого алгоритму засіб фільтрації трафіку проводить первинну кластеризацію. В результаті первинної кластеризації в базі даних створюються дві таблиці, що характеризують шкідливий і надійний трафік. Отримані таблиці використовуються в якості навчальних вибірок при класифікації запитів, що надходять. В процесі роботи таблиці уточнюються і доповнюються.

Дані, що містяться в таблиці, що характеризує шкідливий трафік, використовуються для блокування трафіку. У розробляємому програмному засобі передбачена можливість вилучення з таблиці, відповідної шкідливому трафіку, клієнтських IP адрес і створення на їх основі заборонних правил. Крім цього, на підставі даних про шкідливий трафік можливо реалізувати додаткові механізми захисту. Наприклад, при надходженні шкідливих запитів до конкретної сторінки можна в автоматичному режимі тимчасово заблокувати цю сторінку або ж підмінити її статичної або кеш-версією. В цьому випадку шкідливий трафік, який був некоректно класифікований і не був заблокований на попередньому рівні, завдасть меншої шкоди.

Блокування шкідливих запитів. Для блокування шкідливих запитів передбачено два варіанти. В першому варіанті блокування здійснюється за допомогою створення відповідних забороняючих правил для iptables. Другий варіант буде актуальний якщо засіб функціонує у вузьких рамках віртуального хостингу, в цьому випадку блокування шкідливого трафіку здійснюється за допомогою заборонних правил, зазначених у файлі .htaccess (Лістинг 6.2).

В обох випадках блокування трафіку здійснюється повністю в автоматичному режимі. Також передбачений механізм експорту даних про шкідливий трафік для блокування його в різних програмних файерволах або ж на вищих мережевих вузлах.

Лістинг 6.2 - Приклад блокування IP-адрес у файлі .htaccess

```
order allow, deny
deny from 192.168.0.1
deny from 192.168.0.2
allow from all
```

Архітектура програмного комплексу представлена на рис. 6.1. Взаємодія модулів програмного комплексу, один з одним і з WEB-сервером, відбувається за наступною схемою:

1. В результаті обробки запитів, що приходять до WEB-сервера з мережі інтернет, в журнал WEB-сервера додаються відповідні події.

2. Модуль завантаження даних з заданим інтервалом часу зчитує нові дані з журналу і завантажує їх в базу даних.

3. Модуль виявлення початку атаки, аналізує дані про запити, що містяться в базі даних. У разі виявлення початку атаки, цей модуль створює в базі даних дві порожні таблиці. Одну для легітимних запитів, другу для шкідливих.

4. Модуль виявлення шкідливого трафіку відстежує появу і стан зазначених вище таблиць БД. Якщо таблиці незаповнені, модуль проводить кластеризацію і первинне заповнення таблиць. Якщо в таблицях вже є дані, модуль аналізує

запити, шо надійшли на предмет приналежності до груп надійних або шкідливих запитів, і додає дані про запит в відповідну таблицю.

5. Модуль блокування запиту отримує список IP-адрес з таблиці, що містить шкідливі запити і вносить їх в «чорні списки» брандмауера або передає для блокування на вищестоящий мережевий сегмент.

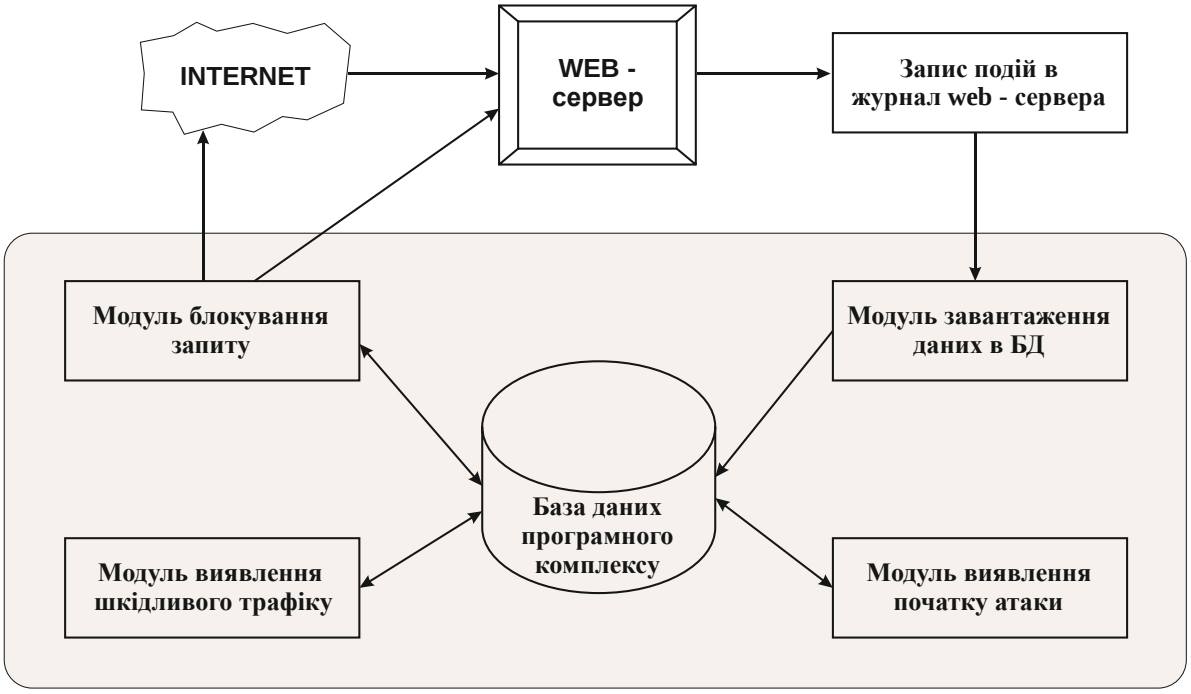

Рисунок 6.1 - Архітектура програмного комплексу

Розроблений програмний засіб повністю відповідає поставленим завданням. Основні риси створеного програмного комплексу для виявлення і протидії DDoS-атакам це кроссплатформенність, універсальність і масштабованість.

Програмний комплекс може застосовуватися в якості засобу забезпечення безпеки так званої «останньої милі». основною спеціалізацією комплексу є забезпечення безпеки web-серверів від DDoS-атак типу http-flood. Програмний комплекс підтримує різні операційні системи, він може бути використаний з більшістю сучасних web-серверів. При цьому інсталяція комплексу може здійснюватися як в рамках фізичного сервера, так і в рамках віртуального хостингу.

Універсальність програмного комплексу виявлення і протидії DDoS-атакам полягає в можливості його використання не тільки для виявлення http-flood'a, а також інших DDoS - атак різних типів. При незначних змінах, що не зачіпають основний модуль, програмний засіб може аналізувати різні дані, що містяться в log-файлах різних мережевих сервісах, або ж використовувати дані, отримані від мережевих локаторів.

У даній реалізації весь програмний комплекс складається з трьох модулів, розміщується на кінцевому мережевому ресурсі. В разі необхідності, модулі програми можуть бути розміщені в різних місцях мережі. Так, наприклад, на кінцевому сервері може знаходитися тільки засіб завантаження даних. Засоби виявлення атаки і фільтрації трафіку можуть бути встановлені на окремому сервері, недоступному для атаки з зовнішньої мережі. При такій установці програмний засіб зможе нормально функціонувати і проводити класифікацію трафіку навіть в випадку відмови атакуємого сервера.

Можливий варіант інсталяції, коли на вузлі, безпеку якого потрібно підтримувати, взагалі не встановлено ніяких модулів програмного засобу. У цьому випадку дані для аналізу можуть бути отримані від мережевих локаторів або вищестоящих маршрутизаторів. Блокування трафіку може бути здійснена на вищому вузлі.

Також програмний комплекс підтримує мультіінсталляцію при одночасному запуску декількох однойменних модулів. Так наприклад, дані для аналізу можуть надходити в базу даних з декількох джерел. Дані про шкідливий трафік можуть бути передані для блокування на різні рівні.

ВИСНОВКИ

У роботі запропонований метод раннього виявлення початку DDoS-атаки і подальшого визначення шкідливих запитів. В основі розробленого методу лежать методи теорії ймовірності, кластерного і статистичного аналізу, принципи машинного навчання.

В якості основних результатів дослідження можна виділити наступні:

1. Проведено моніторинг сучасних розподілених атак, спрямованих на відмову в обслуговуванні. Виділена нова група атак середньої і малої інтенсивності, спрямованих в основному на регіональні ресурси. Проведено моніторинг різних програмних і апаратних засобів протидії та засобів виявлення атак такого типу. Виявлено відсутність засобів, що дозволяють адекватно вирішувати поставлені завдання з виявлення і протидії, для даної групи атак.

2. Запропоновано і обгрунтована гіпотеза про існування сезонності в роботі різних мережевих ресурсів. З'ясовано причини, що впливають на формування та особливості сезонних періодів.

3. Дослідження моделі атаки дозволило створити метод раннього виявлення та протидії DDoS-атакам середньої і малої інтенсивності. Метод є універсальний, враховує, як регіональні особливості, так і інші фактори, і може бути застосований для виявлення і протидії DDoS-атакам різних типів і різної потужності. А також для виявлення аномальних даних в різних сферах діяльності.

4. У процесі розробки запропоновано два алгоритму: алгоритм визначення точки початку атаки і алгоритм поділу змішаного трафіку на легітивний і шкідливий. Відмінною рисою алгоритмів є врахування сезонних коливань мережевого навантаження.

5. На основі запропонованого методу розроблено програмний засіб по виявленню початку атаки і подальшого виявлення і блокування шкідливих запитів. Розроблений засіб відповідає вимогам платформ, універсальності,

відкритості. Відмінною рисою розробленого засобу є модульність і універсальність. При незначній зміні окремих модулів засіб може бути застосовано для забезпечення безпеки різних мережевих ресурсів і їх захист від атак різних типів.

ЛІТЕРАТУРА

1. Алферов, А.П. Основы криптографии / А.П. Алферов, А.Ю. Зубов, А.В. Черемушкин. – М.: Гелиос АРБ, 2002. – 480 с.

2. Анин, Б.А. Защита компьютерной информации / Б.А. Анин – Спб.: БХВ-Петербург. 2000. – 384с.

3. Бабаш, А.В. Информационная безопасность. Лабораторный практикум : учеб. пособие / А. В. Бабаш, Е. К. Баранова, Ю. Н. Мельников. - 2-е изд., стер. - М. : КНОРУС, 2016. - 132 с.

4. Бабаш, А.В. Криптографические методы защиты информации : учебник для студ. вузов / А. В. Бабаш, Е. К. Баранова. - М. : КНОРУС, 2016. - 190 с.

5. Бабенко, Л.К. Современные алгоритмы блочного шифрования и методы их анализа / Л.К. Бабенко, Е.А. Ищукова - М.: Гелиос АРВ, 2006.-376 с., ил.

6. Батурин, Ю.М. Компьютерная преступность и компьютерная безопасность / Ю.М. Батурин, А.М. Жодзинский. – М.: Юридическая литература, 2006. – 160 с.

7. Борисов, М. А. Основы программно-аппаратной защиты информации : учеб. пособие для вузов / М. А. Борисов, И. В. Заводцев, И. В. Чижов. - 4-е изд., перераб. и доп. - М. : ЛЕНАНД, 2016. - 416 с.

8. Брассар, Ж.Б. Современная криптология / Ж.Б. Брассар; Ред. А.Н. Лебедева. – М.: Издательско-полиграфическая фирма ПОЛИМЕД, 1999. – 176 с.

9. Васильева, И.Н. Криптографические методы защиты информации : учебник и практикум для академ. бакалавриата / И. Н. Васильева. - Санкт-Петерб. гос. эконом. ун-т . - М. : Юрайт, 2017. - 349 с.

10. Галатенко, В. А. Информационная безопасность / В.А. Галатенко – М.: Финансы и статистика, 1997. –158 с.

11. Герасименко, В.А. Основы защиты информации / В.А.Герасименко, А.А. Малюк. – М.: Инфо, 2004. – 540с.

12. Глинская, Е. В. Информационная безопасность конструкций ЭВМ и систем : учеб. пособие для студ. вузов / Е. В. Глинская, Н. В. Чичварин. - М. : ИНФРА-М, 2016. - 118 с.

13. Гмурман, А.И. Информационная безопасность / А.И. Гмурман - М.: «БИТ-М», 2004.-387с.

14. Даниленко, А.Ю. Безопасность систем электронного документооборота: Технология защиты электронных документов / А. Ю. Даниленко. - М. : ЛЕНАНД, 2015. - 232 с.

15. Джулій В.М. Метод виявлення та протидії розподіленим атакам, спрямованим на відмову в обслуговуванні / В.М. Джулій, В.І. Чорненький, О.О. Савіцька, - Хмельницький: Вісник Хмельницького національногоуніверситету, 2019. - Вип. №1. – С.127-134

16. Ищейнов, В.Я. Организационное и техническое обеспечение информационной безопасности. Защита конфиденциальной информации : учеб. пособие для студ. вузов / В. Я. Ищейнов, М. В. Мецатунян. - 2-е изд., перераб. и доп. - М. : Форум ; М. : ИНФРА-М, 2014. - 256 с.

17. Кабакова, Н. В. Система защиты информации ViPNet: курс лекций : учеб. пособие / Н. В. Кабакова [и др.] ; под ред. А. О. Чефрановой. - М. : ДМК Пресс, 2014. - 392 с.

18. Конеев, И. Р. Информационная безопасность предприятия /И.Р. Конеев, А.В. Беляев - СПб.: БХВ-Петербург, 2003.- 752с.:ил.

19. Медведовский, И.П. Атака через Internet / И.П. Медведовский, П.В. Семьянов, В.А. Платонов; Ред. П.Д. Зегжды. – СПб.: Мир и Семья – 95, 1997 – 296 с.

20. Мелюк, А. А. Введение в защиту информации в автоматизированных системах / А.А. Мелюк, С.В. Пазизин - М.: Горячая линия - Телеком, 2001.- 48с.:ил.

21. Мельников, Ю.Н. Информатика: учеб. пособие / Ю.Н. Мельников; ред. П.Б. Хореев. – М.: Папирус Про, 2003. – 662 с.

22. Мельников, Ю.Н. Обеспечение целостности информации в вычислительных системах / Ю.Н. Мельников, В.А. Мясников, Ю.П. Лутковский // Защита информации – 2007. – № 1. – С. 72 – 79.

23. Немет, Эви. Unix, руководство системного администратора. / Эви Немет, Гарт Снайдер – пер с англ. – СПб.: Питер: К.Издательская группа BHV,2002. – 928 с.

24. Нестеров, С.А. Основы информационной безопасности : учебник / С. А. Нестеров. - СПб. : Лань, 2017. – 423 с.

25. Петраков, А.В. Утечка и защита информации в телефонных каналах / А.В. Петраков. - 7-е изд., доп. - М. : РадиоСофт, 2017. - 448 с.

26. Оглтри, Т. Практическое применение межсетевых экранов: Пер. с англ. / Т. Оглтри, - М.: ДМК Пресс, 2001.- 400с.:ил.

27. Олифер, В.Г. Безопасность компьютерных сетей / В. Г. Олифер, Н. А. Олифер. - М. : Горячая линия-Телеком, 2017. - 644 с.

28. Савіцька О.О. Архітектура програмного комплексу забезпечення безпеки виявлення і протидії DDoS-атакам / О.О. Савіцька, В.М. Джулій. - «Інтелектуальний потенціал – 2018» - збірник наукових праць молодих науковців і студентів з нагоди 30-річчя підготовки ІТ- фахівців в ХНУ/ Колектив авторів – Хмельницький: ПВНЗ УЕП, 2018. – Ч.3: Кібербезпека та актуальні проблеми комп'ютерних систем і мереж — С. 44 - 47.

29. Соколов, А. В. Защита от компьютерного терроризма. Справоч-ное пособие. / А.В. Соколов, О.М. Степанюк - СПб.: БХВ - Петербург, Арлит, 2002.- 496с.:ил.

30. Таранцев А. А. Инженерные методы теории массового обслуживания / А. А. Таранцев. – Санкт-Петербург: Наука, 2007. – 164 с.

31. Чмора, А. Л. Современная прикладная криптография. 2-е изд. / А.Л. Чмора. - М.:Гелиос АРВ, 2002. - 256с.:ил.

32. Шаньгин, В. Ф. Информационная безопасность и защита информации / В.Ф. Шаньгин. - М. : ДМК Пресс, 2017. - 702 с.

ДОДАТОК А

Код (лістинг) програмного забезпечення аналізу трафіка

```
#!/usr/bin/python
from __future__ import division
import pandas as pd
from pandas.io import sql
import psycopg2
import base64
import numpy as np
import struct
from socket import inet_ntoa
import time
import datetime
from sklearn import preprocessing
from sklearn import neighbors
from sklearn import svm
from sklearn import tree
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
start_time = time.time()
SIZE_OF_HEADER = 24
SIZE_OF_RECORD = 48
ACCUR = 4
PROTOCOL = { 1: "ICMP", 6: "TCP", 14: "Telnet", 17: "UDP" }
# формування словника з інформацією про потоки
def nfdata_new():
 nfdata = {}
 nfdata['flow_count'] = 0
 nfdata['pcount'] = 0
 nfdata['bcount'] = 0
 nfdata['protocol'] = { 'TCP': 0, 'ICMP': 0, 'Telnet': 0, 'UDP': 0 }

 return nfdata

# додавання даних про потік в словник
def nfdata_add(nfdata, flow):
 nfdata['pcount'] += flow['pcount']
 nfdata['bcount'] += flow['bcount']
 nfdata['flow_count'] += 1
 for p in flow['protocol']:
 nfdata['protocol'][p] += flow['protocol'][p]

return nfdata
# групування даних про трафік
def get_group(flow, t=None):
 if t == None or t == "all":
 return "all"
 s = ""
```

```
 for x in t.split('_'):
 if 'd' in x:
 s += "_"
 s += flow[x[0] + "addr"]
 if 'p' in x:
 s += ":" + str(flow[x[0] + "port"])
 return s
# вибірка трафіку з БД
def select(conn, t1=None, t2=None):
 limit = ""
 if t1 != None and t2 != None:
 limit = "WHERE time > %s AND time < %s" % (t1, t2)
 elif t1 != None:
 limit = "WHERE time == %s" % t1
 # формування запиту до БД
 sqlquery = "SELECT * FROM flows" + limit
 data = sql.read_sql(sqlquery, conn, index_col='id')
 return data
# формування DataFrame для обробки
def get_data(conn, t1=None, t2=None, nf_group_type=None):
 data = select(conn, t1, t2)
 result = { }
 print('* get_data() starts ', datetime.datetime.now())
 df = pd.DataFrame(
 columns=(
 'saddr',
 'sport',
 'daddr',
 'dport',
 'pcount',
 'bcount',
 'first',
 'last',
 'duration',
 'bpp',
 'bps',
 'pps',
 'protocol',
 'label',
 )
)
 col_names = list(df)
 for i in range(len(data)):
 nfc_group = { }
 #X-threads:
 for j in range(len(data.iloc[i]["data"])):
 #1-thread:
 s_buf = data.iloc[i]["data"][j]
 if s_buf == "":
```

```
 continue
 buf = s_buf.decode('base64')
 flow_count = struct.unpack('B', buf[3:4])[0]
 for index in xrange(flow_count):
 offset = SIZE_OF_HEADER + (index * SIZE_OF_RECORD)
 flow = { }
 # розбір запису netflow
 if len(buf) - offset > 47:
 # розпакування структури
 d = struct.unpack('!IIIIHH',buf[offset + 16:offset + 36])
flow['saddr'] = inet_ntoa(buf[offset + 0:offset + 4])
flow['sport'] = d[4]
 flow['daddr'] = inet_ntoa(buf[offset + 4:offset + 8])
flow['dport'] = d[5]
 flow['pcount'] = d[0]
flow['bcount'] = d[1]
 flow['protocol'] = { }
flow['protocol'][PROTOCOL[ord(buf[offset + 38])]] = 1
flow['upairs'] = set()
flow['upairs'].add((buf[offset + 0:offset + 4], d[4],
buf[offset + 4:offset + 8], d[5]))
flow['first'] = d[2]
flow['last'] = d[3]
flow['duration'] = d[3] - d[2]
 flow['proto'] = ord(buf[offset + 38])
 flow['bpp'] = round(flow['bcount'] / flow['pcount'], ACCUR)
 flow['bps'] = 0 if not flow['duration'] else
round((flow['bcount'] * 8) / flow['duration'], ACCUR)
flow['pps'] = 0 if not flow['duration'] else
round(flow['pcount'] / flow['duration'], ACCUR)
 flow['label'] = 'benign' if d[3] - d[2] == 0 else
'irc_botnet'
 temp_row = pd.Series(
 [
 flow['saddr'],
flow['sport'],
flow['daddr'],
flow['dport'],
flow['pcount'],
flow['bcount'],
flow['first'],
flow['last'],
flow['duration'],
flow['bpp'],
flow['bps'],
flow['pps'],
flow['proto'],
flow['label'],
 ], index=col_names
```

```
 )
 df = df.append(temp_row, ignore_index=True)
s = get_upair_group(flow, nf_group_type)
if s not in nfc_group:
 nfc_group[s] = nfdata_new()
 nfc_group[s] = nfdata_add(nfc_group[s], flow)
 for k in nfc_group:
 nfc_group[k]['ucount'] = len(nfc_group[k]['upairs'])
 del nfc_group[k]['upairs']
 result[data.iloc[i]["time"]] = nfc_group
 # return result
 df.to_csv('~/traf.csv')
 return df
def analyze(df):
 print('analyze', datetime.datetime.now())
 new_df = pd.concat(
 [
 df['pcount'],
 df['bcount'],
 df['duration'],
 df['bpp'],
 df['bps'],
 df['pps'],
df['protocol'],
 df['label']
 ],
 axis=1,
 )
 X = np.array(new_df.drop(['label'], 1))
 y = np.array(new_df['label'])
 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
 gnb = GaussianNB()
 y_pred = gnb.fit(X_test, y_test)
 gnb_accuracy = clf.score(X_test, y_test)
 print('NB Accuracy: {}'.format(gnb_accuracy))
 tree_clf = tree.DecisionTreeClassifier()
 tree_clf.fit(X_train, y_train)
 tree_accuracy = tree_clf.score(X_test, y_test)
 print('Tree Accuracy: {}'.format(tree_accuracy))
 knn_clf = neighbors.KNeighborsClassifier()
 knn_clf.fit(X_train, y_train)
 knn_accuracy = clf.score(X_test, y_test)
 print('KNN Accuracy: {}'.format(knn_accuracy))
 clf = svm.SVC()
 clf.fit(X_train, y_train)
 svm_accuracy = clf.score(X_test, y_test)
 print('SVM Accuracy: {}'.format(svm_accuracy))
```

ДОДАТОК Б
Перелік наукових праць

УДК 004.891

В.М. ДЖУЛІЙ
Хмельницький національний університет
В. І. ЧОРНЕНЬКИЙ
Хмельницький національний університет
О.О. САВІЦЬКА
Хмельницький національний університет

МЕТОД ВИЯВЛЕННЯ ТА ПРОТИДІЇ РОЗПОДІЛЕНИМ АТАКАМ, СПРЯМОВАНИМ НА ВІДМОВУ В ОБСЛУГОВУВАННІ

В роботі запропоновано актуальний метод та інструментарій для раннього виявлення розподілених атак, спрямованих на відмову в обслуговуванні, і подальшого виявлення шкідливого трафіку на стороні ресурсу, що атакується і його блокування власними силами. Для поділу змішаного трафіку використовується алгоритм кластеризації k-means. Вибір даного алгоритму обгрунтований, проведено доказ його ефективності, підібрані оптимальні характеристики і розмірність даних, вироблені критерії успішності. Розроблені алгоритми складають основу узагальненої методики виявлення DDoS-атак і шкідливого трафіку,

Ключові слова: моделі, алгоритми, ефективність виявлення атак, метод, мережевий трафік, інформаційна безпека.

V.M. Dzhuliy
Khmelnytsky National University, Khmelnytsky, Ukraine
V. I. CHORNENKY
Khmelnytsky National University, Khmelnytsky, Ukraine
O.O. SAVITSKAYA
Khmelnytsky National University, Khmelnytsky, Ukraine

METHODS OF DETECTION AND COUNTERACTION TO THE DISTRIBUTED ATTACKS AIMED AT FAILURE IN SERVICE

Abstract. The purpose of the work is to create an actual method and tool for early detection of the distributed attacks aimed at denial of service, and further detection of any harmful traffic directed on the attacked resource and its own blocking.

Countermeasures specialized to ensure the security of small and medium-sized resources, have received less development due to the predominance of large severe attacks in the past. And now they are lagging behind the evolution of DDoS attacks themselves. As a part of the method development for detecting DDoSatak and malicious traffic, an original algorithm has been created for detecting a distributed attack at the early stages of a denial of service attack. The algorithm takes into account seasonal deviations in the load, which makes it possible to detect the point of attack in the early stages and with greater accuracy. Besides, research was conducted to confirm the existence of seasonality and to identify any typical seasonal periods.

As a result of the research the week, daily and uncertain seasonality and the reasons of its emergence are revealed. It was revealed the tendency of medium and low power attacks on regional resources. The peculiarities of regional level DDoS attacks were investigated and the task of creating a method and software complex for the detection of DDoS attacks and malicious inquiries was solved. The received technique was tested on the data - for legitimate requests, the completeness of the detection was 0.9991 with an accuracy of 0.99811, for malicious inquiries, completeness of 0.9975, accuracy of 0.9924.

Keywords: models, algorithms, attack detection efficiency, method, network traffic, information security.

Вступ

DDoS-атака - розподілена атака, спрямована на відмову в обслуговуванні. В результаті атаки такого типу мережевий ресурс, що атакується, отримує лавиноподібну кількість запитів, які не встигає обробити сервер. Джерелом шкідливих запитів є так звані зомбі-мережі, що складаються переважно з комп'ютерів звичайних користувачів, в силу якихось причин заражених шкідливим програмним забезпеченням. Періодичні повідомлення в

засобах масової інформації про недоступність тих чи інших ресурсів в результаті розподілених атак, спрямованих на відмову в обслуговуванні, говорять про неефективність засобів протидії такого роду атак. Також збільшується кількість атак і до невеликих, «середніх» сайтів, які до недавнього часу не становили інтересу для зловмисників. Однак, в даний час, у зв'язку зі збільшенням їх важливості і затребуваності, перебої в їх роботі можуть бути критичними. Разом з цим змінюються і мотиви, які рухають зловмисниками, якщо раніше серед причин виникнення DDoS-атак можна було виділити протест, хуліганство і т.д., то сьогодні все частіше DDoS-атаки є наслідком шантажу і способом вимагання грошей. Це переводить DDoS-атаки з площини одиничних протестних акцій в область кримінального бізнесу, які не обмежується вимаганням, але і є інструментом екстремістських і терористичних організацій [2].

Сьогодні у всьому світі стали звичайною ситуацією атаки на сайти державної влади напередодні виборів або важливих політичних подій [1,5]. Для паралізації невеликого регіонального ресурсу досить невеликої за потужністю атаки і як наслідок невеликої бот-мережі. Обслуговування та підтримка таких бот-мереж є менш витратним, і потенційно створити такі мережі може більшість зловмисників. Цей факт на фоні відсутності адекватних засобів протидії робить загрози безпеки регіональних ресурсів в результаті DDoS-атак особливо значущими. З одного боку, для протидії таким атакам можуть бути ефективно застосовані засоби, призначені для відображення великих атак. З іншого - впровадження і підтримка таких засобів є економічно затратною і не по кишені регіональним ресурсам. Засоби протидії, спеціалізовані саме на забезпечення безпеки невеликих і середніх ресурсів, отримали менший розвиток через переважання в минулому саме великих атак. І в даний час відстають від еволюції самих DDoS-атак[6].

Відповідно до звіту, опублікованого компанією «Лабораторія Касперського», число DDoS-атак постійно збільшується [3,4]. Так, наприклад, за друге півріччя 2017 р. значно збільшилася кількість атак. При цьому збільшилася і потужність проведених атак, в порівнянні з першим півріччям вона виросла на 57%:

Разом з кількістю і потужністю постійно зростає і складність самих атак. Зловмисники шукають принципово нові методи проведення атак, і дуже часто існуючі засоби захисту виявляються безсилими перед ними. Так, наприклад, порівняно новий вид DDoS-атак - THC-SSL-DOS експлуатує особливості SSL протоколу і дає можливість одному комп'ютеру зробити недоступним сервер середньої конфігурації [3]. В 2017р. був атакований сайт американської біржі Nasdaq. В результаті атаки сайт повністю перестав реагувати на запити. При цьому біржа Nasdaq є найбільшою електронної фондовою біржею США і другою в світі за величиною ринкової капіталізації [5]. Україна також не відстає від світової тенденції зростання DDoS-атак, а по деяких позиціях займає навіть перші місця. За повідомленнями засобів масової інформації, з лютого по березень 2017р. з України було проведено понад 2,4 мільйона кібер-атак.

Найбільш характерним проявом DDoS-атак є «затоплення» або flooding каналу зв’язку або конкретного мережного пристрою величезною кількістю мережевих пакетів. В залежності від типу пакетів, це може призвести до перевантаження каналу і, як наслідок, неможливості проходження по ньому легітимного трафіку, або до підвищеної завантаженості пристрою (заповнення доступного обсягу оперативної пам’яті і завантаження ресурсів процесора).

При достатніх обчислювальних і серверних потужностях, можливо зробити перенаправлення трафіку назад до атакуючого. Цей метод досить складний в реалізації і вимагає не тільки хорошої матеріальної бази, а й високої кваліфікації адміністратора серверного ресурсу.

Постановка задачі

В результаті проведеного дослідження відмічено, що в даний час значно збільшилася кількість DDoS-атак середньої і малої інтенсивності, спрямованих, як правило, на регіональні ресурси. Це збільшення цілком прогнозовано - з розвитком мережі збільшується потенційна кількість можливих жертв. Крім того, вдосконалюється сам механізм проведення атак. Для зловмисника проведення атаки вже не є настільки складним. А зомбі-комп'ютери намагаються емулювати дії самих користувачів. Все це веде до загального збільшення числа атак.

Аналіз засобів протидії показав, що в даний час більший розвиток отримала група засобів протидії, призначена для відбиття потужних атак. У цю групу входять, як правило, дорогі засоби, призначені для великих провайдерів або компаній. Засоби протидії невеликим і середнім атакам, що розміщенні на сервері, представлені в незначній кількості. При цьому аналіз вхідного трафіку на рівні додатків може бути більш ефективним. З одного боку, проведення такого аналізу економічно менш затратно, з іншого - може бути цілком достатнім для відбиття малих і середніх атак, тенденція домінування яких вже намітилася.

Основна частина

Оптимальним рішенням для виявлення початку атаки і подальшого виявлення шкідливого трафіку буде рішення, засноване на аналізі аномалій, в результаті якого відбувається порівняння поточного стану системи з її нормальним станом. Порівняння станів системи в контексті DDoS-атак можна проводити шляхом порівняння різних властивостей мережевої активності. До цих властивостей можуть бути віднесені: кількість запитів, тип запитів, кількість запитів певного типу або протоколу, IP адреса джерела, швидкість надходження запитів, їх час і т.д.

Нехай множина $A\left(a_1, a_2, a_3, ..., a_n\right)$ - набір всіх можливих властивостей для всіх мережевих клієнтів. Множина $B\left(b_1, b_2, b_3, ..., b_m\right)$ - множина легітивних клієнтів конкретного мережевого ресурсу. Кожен мережевий клієнт має набір індивідуальних властивостей. Наприклад, клієнт b_1 має властивості $A1\left(a_4, a_8, a_{10}, a_{14}\right)$, клієнт b_2 має властивості $A2\left(a_3, a_8, a_{11}, a_{14}\right)$ і т.д. Дані властивості представляють набір підмножин множини *A*. Перетин всіх цих підмножин характеризує клієнтів мережевого ресурсу, за якими вони можуть бути класифіковані. Точно так нелегітивні клієнти матимуть свій набір властивостей, за яким вони також можуть бути класифіковані.

На сьогоднішній день DDoS-атаки ускладнюються, і зловмисники намагаються повністю імітувати поведінку легітивних клієнтів. У цій ситуації перевагу при аналізі властивостей мережевої активності необхідно віддати тим властивостям, які не можуть бути підроблені зловмисниками. При нестачі таких властивостей необхідно вводити штучні властивості, наприклад, проходження модифікації тест Тьюринга - введення даних з картинки.

Таким чином, завдання по визначенню і виявлення шкідливих запитів в контексті даної роботи зводиться до їх класифікації на підставі властивостей мережевої активності. Оптимальним рішенням для виявлення шкідливого трафіку є використання різних класифікаторів і нейронних мереж. Складністю в реалізації даного рішення є той факт, що для нормального функціонування класифікатора потрібно мати дві актуальні навчальні вибірки, відповідно шкідливому і легітивному трафіку. Однак до моменту початку атаки отримати ці вибірки не представляється можливим. Це цілком очевидно для вибірки, що відповідає шкідливому трафіку, так як до початку атаки шкідливі запити відсутні. Але це також справедливо і для вибірки, що характеризує легітивний трафік. Так як мережева картина постійно змінюється, буде змінюватися і вміст вибірки відповідного легітивного трафіка. Таким чином, вибірка по легітивності трафіка, наприклад, місячної давності, може бути не актуальна для поточної мережевої ситуації. Крім того, є ризик, що в цій вибірці можуть виявитися дані, відповідні шкідливим запитам, що в подальшому викличе помилки в роботі класифікатора.

Дана проблема актуальна, тому що зловмисник може спеціально почати підмішувати до легітивного трафіку незначне число шкідливих запитів, які не зможуть бути ідентифіковані як почаок атаки, але зможуть негативно «навчити» вибірку, що характеризує легітивний трафік. Для подолання цієї проблеми необхідно точно визначити точку початку атаки. Це дасть можливість весь попередній трафік віднести до легітивного і відкриє додаткові можливості по розділенню змішаного трафіку, який приходить після початку атаки, на легітивний і шкідливий.

В цьому випадку методика виявлення шкідливого трафіку, в першому наближенні, буде зводитися до наступних кроків: визначаємо актуальні сезонні періоди; з урахуванням сезонності визначаємо точку початку атаки; відносимо весь попередній перед початком атаки трафік до легітивного; класифікуємо змішаний трафік на

легітивний і шкідливий; порівнюємо легітивний трафік виділений зі змішаного з трафіком що надійшов до початку атаки; на підставі результатів, отриманих на попередньому кроці і вироблених критеріїв успішності, коригуємо вибірки; весь вступний трафік аналізуємо з урахуванням отриманих даних.

Початок DDoS-атаки пов'язаний зі збільшенням числа запитів до атакуємого сервера. Таким чином, для фіксації факту атаки необхідно встановити границю по кількості запитів до сервера, при порушенні якої однозначно буде фіксуватися нештатна ситуація. Такою границею може виступати максимальна кількість запитів до сервера, плюс деякий запас можливих запитів. Можливість установки граничної межі, після якої буде відбуватися оповіщення адміністраторів, активація необхідних модулів і т.д., реалізована в різних мережевих засобах як програмного, так і апаратного рівня. При цьому такий підхід має ряд мінусів:

1. Для запобігання випадкових спрацьовувань, межа що встановлюється повинна бути істотно вище максимального рівня кількості запитів. Що, в свою чергу, призводити до виникнення похибки при виявленні атаки.

2. Мережевий ресурс може відчувати різне навантаження в залежності від часу доби і днів тижня. В цьому випадку атака, що почалася в період затишшя, наприклад, у вихідний день або вночі, буде зафіксована із запізненням. Якщо в системі запобігання вторгнень передбачено використання класифікаторів та навчання фільтрів на підставі вхідного трафіку, є ймовірність в їх негативному навчанні.

Для вирішення зазначених проблем необхідно використовувати ковзаючу оцінку, що характеризує поточну мережеву активність. На підставі цієї оцінки встановлювати динамічну границю, актуальну для періоду можливого початку атаки. В якості ковзаючої оцінки можливо використовувати середньоквадратичне відхилення:

$$\sigma = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} \left(x_i - \bar{x} \right)^2} , \qquad (1)$$

де σ - середньоквадратичне відхилення; n - кількість розглянутих часових періодів; x_i - кількість запитів за i-період;

$\bar{x}$ - середнє арифметичне запитів по всіх періодах.

В результаті експериментів, було встановлено, що для різних сайтів оптимальне значення верхньої межі може відрізнятися і перебувати, як правило, в діапазоні від 2.2σ до 2.9σ. З цієї причини, для більш гнучкого налаштування програмного забезпечення по виявленню початку DDoS-атаки, цей параметр задається не жорстко. У оператора програмного комплексу є можливість варіювати значення даного параметра. Однак такий підхід також має потенційну вразливість, пов'язану з тим, що зловмисник може поступово нарощувати потужність атаки, зрушуючи при цьому границю середньоквадратичного відхилення. усунути дану вразливість може облік сезонних коливань.

Такий підхід дозволяє сформувати досить точну верхню межу, порушення якої може бути витлумачено як виникнення мережевої аномалії. Збільшення точності дозволяє зменшити час, необхідний для виявлення атаки, і досить точно зафіксувати її початок.

Крім того, в рамках такого підходу виключаються можливості негативного навчання фільтрів і спрацьовування системи виявлення з запізненням шляхом поступового нарощування потужності атаки. Так як межа в цьому випадку буде будуватися за схожими сезонним періодами. Наприклад, поступове нарощування потужності атаки протягом дня буде зафіксовано при порівнянні кількості запитів з кількістю запитів актуальних сезонних періодів за минулу добу.

Виявлення і дослідження сезонності. В рамках раннього виявлення початку атаки, і з огляду на перспективність підходу необідно враховувати сезонні коливання, провести додаткове дослідження, що вивчає сезонні коливання кількості запитів до мережевих Internet ресурсів. Основним завданням дослідження був доказ існування сезонних періодів в роботі web-сайтів. А також вирішення питання, чи може випадковий сплеск в відвідуваності Internet ресурсу, викликаний, наприклад, публікацією на нього посилання з високо відвідуваного ресурсу, викликати порушення сезонності, і, як наслідок, помилкове спрацьовування.

В результаті дослідження для поділу змішаного трафіку на легітивний і шкідливий обраний метод кластеризації на основі алгоритму k-means. Метод забезпечує прийнятну точність, мінімальне навантаження і оптимальну швидкість роботи. Даний алгоритм дозволяє провести кластеризацію при заздалегідь відомому числі кластерів. Алгоритм має прийнятну точність, необхідну для первинного поділу, і більш високу швидкість роботи, порівнянню з іншими алгоритмами.

Суть алгоритму полягає в виділення двох кластерів і обчисленні їх центрів мас, на наступних ітераціях відбувається корекція кластерів (перенесення елементів в відповідно до розрахованих центрів мас) і перерахування центрів мас.

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} \left(x_j - \mu_i \right)^2 \tag{2}$$

де k- число кластерів;

S_i - отримані кластери, i – 1,2, … , k_i ;

μ_i - центри мас векторів $x_j \in S_i$.

В результаті роботи алгоритму змішаний трафік буде розділений на два кластера, відповідних легітивному і шкідливому трафіку.

Таким чином, на даному етапі доступні для аналізу і обробки три групи трафіку:

1. Відповідно, легітивному трафіку, що передує початку атаки - *T*.
2. Відповідно, легітивному трафіку, виділений із змішаного трафіку - *T**.
3. Відповідно шкідливому трафіку, виділений із змішаного трафіку - *H*.

Критерії успішності, корекція отриманих кластерів. Для оцінки ефективності кластеризації розглянемо рівняння стаціонарних ймовірностей:

$$\begin{aligned}
&p_0 \lambda = p_1 \mu \\
&\left(\lambda + i\mu\right) p_i = \lambda p_{i-1} + \left(i+1\right) \cdot \mu p_{i+1}, \ i = 1,..., K-2, \\
&\left(\lambda + \left(K-1\right)\mu\right) p_{K-1} = \lambda p_{K-2} + KN \cdot \mu p_K, \\
&\left(\lambda^* + KN\mu\right) p_K = \lambda p_{K-1} + KN \cdot \mu p_{K+1}, \\
&\left(\lambda^* + i\mu^*\right) p_i = \lambda^* p_{i-1} + \left(i+1\right) \cdot \mu^* p_{i+1}, \ i = K+1,..., N-1
\end{aligned} \tag{3}$$

де λ - інтенсивність навантаження;

λ_L - інтенсивність навантаження, створювана легальними користувачами;

S - інтенсивність шкідливого трафіку, μ - інтенсивність звільнення черги запитів;

μ^* - інтенсивність звільнення черги запитів при активованому фільтрі,

K - межа активації фільтра;

N - обсяг черги запитів;

$\lambda = \lambda_L + S$ - навантаження в момент атаки;

$\lambda^* = \lambda_L + S\left(1 - E_2\right)$ - навантаження при активованому фільтрі;

E_1, E_2 - помилки першого і другого роду,

При нормуванні $\sum_{i=1}^{N} p_i = 1,$ отримаємо ймовірність блокування запиту.

$$p_{BLK} = \frac{\frac{1}{N!}\left(\frac{\lambda}{\mu}\right)^{K-1}\frac{\lambda}{\mu^*}\left(\frac{\lambda^*}{\mu^*}\right)^{N-K}}{\sum_{i=0}^{K-1}\frac{\left(\frac{\lambda}{\mu}\right)^i}{i!} + \sum_{i=K}^{N}\frac{1}{i!}\left(\frac{\lambda}{\mu}\right)^{K-1}\frac{\lambda}{\mu^*}\left(\frac{\lambda^*}{\mu^*}\right)^{i-K}} \quad (4)$$

Таким чином,ефективність поділу шкідливого і легітивного трафіку можна оцінити, як $R = (1-E_1)\cdot(1-p_B)$.

На підставі отриманої оцінки, були вироблені критерії успішності.

На наступному кроці алгоритму проводиться корекція отриманих вибірок з урахуванням наступних критерій:

1. Критерій розмірності отриманих кластерів. Якщо в поточному періоді, що відноситься до атаки, кількість запитів - *n*, а в аналогічних сезонних періодах, що відносяться до легітивного трафіку - *m*, то кількість шкідливих запитів буде наближено рівним *n*-*m*. Це ж справедливо і для різних властивостей мережевої активності (кількість запитів до цільової сторінки, цільового порту, за певним протоколом і т.д.)

2. Критерій схожості легітивних вибірок. Максимальна схожість легітивної вибірки, що передує початку атаки, з легітивною вибіркою, виділеної із змішаного трафіку.

3. Критерій відповідності центрів мас. Центр мас надійної вибірки, виділеної із змішаного трафіку, повинен відповідати аналогічному сезонному періоду надійного трафіку, що передує початку атаки. Іншими словами, відстань між цими центрами мас повинена наближатися до нуля. Для подальшого уточнення можна розрахувати ймовірність приналежності кожного елемента своєму класу. Елементи з найменшою ймовірністю переносяться в протилежні групи з урахуванням критерію розмірності груп.

Для розрахунку схожості надійних кластерів і надалі для класифікації запитів, що надходять можна скористатися «Байєсовким класифікатором». В якості ймовірнісної моделі для класифікатора використовуємо умовну ймовірність $p(C \mid F_1,...,F_n)$ над залежною змінною класу *C* з малою кількістю результатів або *класів*, що залежить від декількох змінних $F_1,...,F_n$. Використовуючи теорему Байєса, запишемо:

$$p(C \mid F_1,...,F_n) = \frac{p(C)\cdot p(F_1,...,F_n \mid C)}{p(F_1,...,F_n)}$$

Умовний розподіл по класовій змінній *C* може бути виражено так:

$$p(C \mid F_1,...,F_n) = \frac{1}{Z}\,p(C)\prod_{i=1}^{n} p(F_i \mid C)$$

Таким чином, для класифікації трафіку за двома класами отримаємо:

$P(T \mid D) = \frac{P(T)}{P(D)}\prod_{i=1}^{n} P(w_i \mid T)$ - для класу надійних користувачів;

$P(H \mid D) = \frac{P(H)}{P(D)}\prod_{i=1}^{n} P(w_i \mid H)$ - для класу ненадійних користувачів.

В якості навчальних вибірок використовуються множина *T* і множина *H*. Після закінчення цього кроку елементи з множини *T**, віднесені до групи шкідливого трафіку, міняються місцями з елементами множини *H* з урахуванням зазначених вище критеріїв. Даний крок повторюється до тих пір, поки всі елементи множини *T* не будуть позначені як легітивні, або поки алгоритм не досягне порогового значення ітерацій.

Отримані вибірки, відповідні легітивному і шкідливому трафіку, а також механізм їх підтримки в актуальному стані дозволяють використовувати їх з різними класифікаторами. На рис. 1 показані принципові схеми

алгоритмів по визначенню початку атаки і виділенню шкідливого трафіку. Перша схема (рис. 1 а), пояснює алгоритм виділення шкідливого трафіку, друга (рис. 1б) і третя (рис. 1в) алгоритми визначення початку атаки.

На першому кроці відбувається виклик підпрограм по виявленню сезонних періодів, розрахунку для них допустимої межі кількості запитів, і визначення початку атаки. У разі початку атаки, алгоритм повинен розподілити змішаний трафік на два кластери, один містить шкідливі запити, інший надійні запити. дані кластери уточнюються. Нові запити аналізуються на приналежність того або іншому кластеру і по результату додаються до відповідного кластеру.

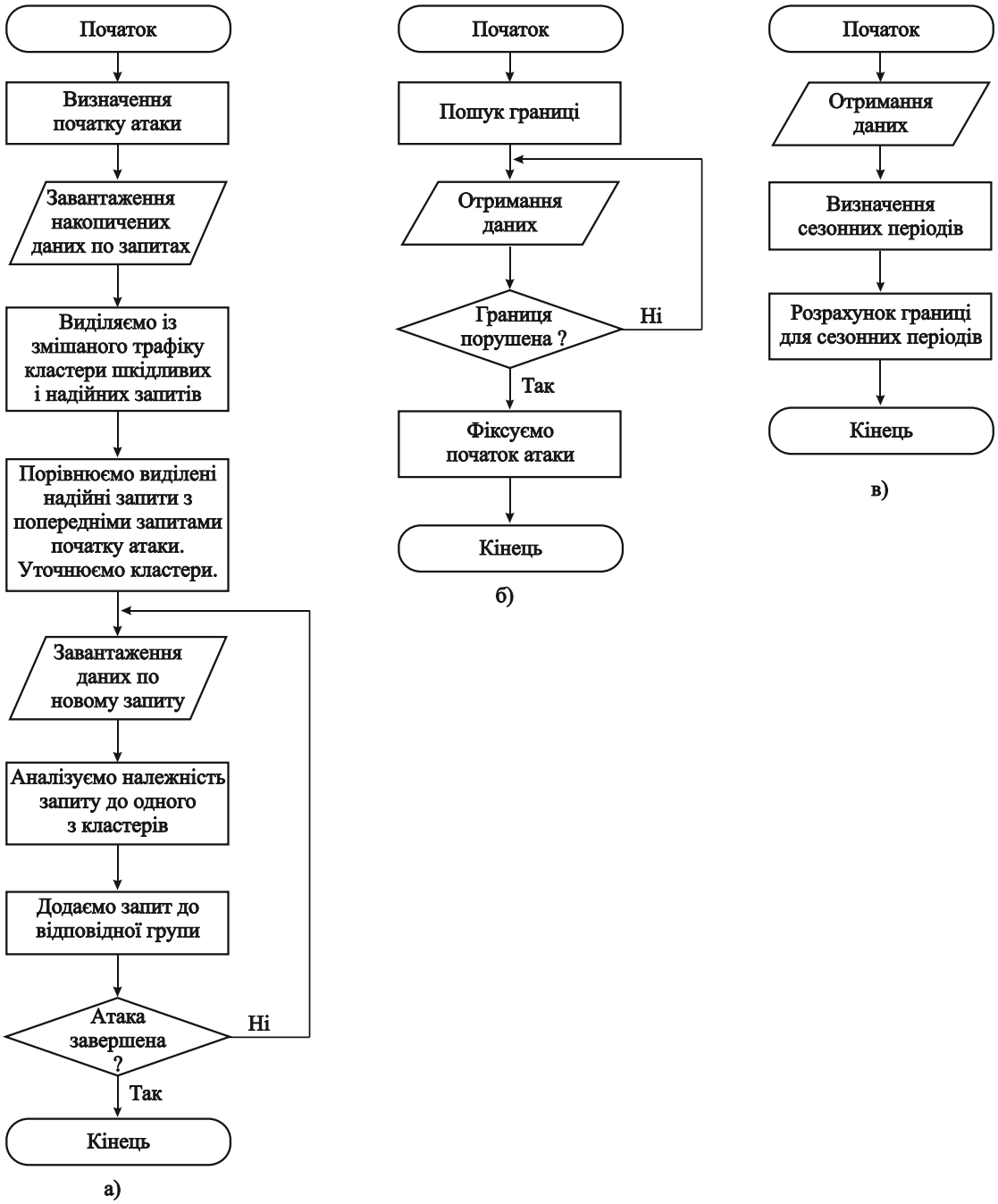

Рис. 1. Алгоритми по визначенню початку атаки і виділенню шкідливого трафіку

Висновки

В рамках розробки методу виявлення DDoSатак і шкідливого трафіку розроблений оригінальний алгоритм виявлення на ранніх стадіях точки початку розподіленої атаки, спрямованої на відмову в обслуговуванні. Алгоритм враховує сезонні відхилення в навантаженні, що дає можливість виявляти точку початку атаки на ранніх стадіях і з

більшою точністю. додатково проведено дослідження, спрямоване на підтвердження існування сезонності і виявлення типових сезонних періодів. В результаті дослідження виявлені тижнева, добова і невизначена сезонність і причини її виникнення.

Розроблено метод отримання навчальних вибірок та класифікації трафіку, що надходить, на групи шкідливих і легітивних запитів. Для поділу змішаного трафіку використовується алгоритм кластеризації k-means. Вибір даного алгоритму обгрунтований, проведено доказ його ефективності. Для алгоритму підібрані оптимальні характеристики і розмірність даних, вироблені критерії успішності. Розроблені алгоритми складають основу узагальненої методики виявлення DDoS-атак і шкідливого трафіку, яка в загальному вигляді може бути описана так: за допомогою статистичних даних, визначаємо існуючі сезонні періоди; для кожного сезонного періоду визначаємо допустиму верхню межу кількості запитів; у разі порушення границі, фіксуємо точку початку атаки; відносимо весь, що передує початку атаки, трафік до кластеру, відповідного легітивного трафіку; за допомогою алгоритму k-means класифікуємо змішаний трафік на легітивний і шкідливий; порівнюємо трафік, що передує початку атаки, з кластером, легітивного трафіку, виділеного зі змішаного трафіку; на підставі результатів, отриманих на попередньому кроці, і з урахуванням вироблених критеріїв успішності, коригуємо кластери; весь трафік, що надходить, аналізуємо з урахуванням отриманих в попередньому пункті результатів.

Літертура

1. Бабаш, А.В. Криптографические методы защиты информации : учебник для студ. вузов / А. В. Бабаш, Е. К. Баранова. - М. : КНОРУС, 2016. - 190 с.

2. Батурин, Ю.М. Компьютерная преступность и компьютерная безопасность / Ю.М. Батурин, А.М. Жодзинский. – М.: Юридическая литература, 2006. – 160 с.

3. Борисов, М.А. Основы программно-аппаратной защиты информации : учеб. пособие для вузов / М. А. Борисов, И. В. Заводцев, И. В. Чижов. - 4-е изд., перераб. и доп. - М. : ЛЕНАНД, 2016. - 416 с.

4. Васильева, И.Н. Криптографические методы защиты информации : учебник и практикум для академ. бакалавриата / И. Н. Васильева. - Санкт-Петерб. гос. эконом. ун-т . - М. : Юрайт, 2017. - 349 с.

5. Нестеров, С.А. Основы информационной безопасности : учебник / С. А. Нестеров. - СПб. : Лань, 2017. – 423 с.

6. Олифер, В.Г. Безопасность компьютерных сетей / В. Г. Олифер, Н. А. Олифер. - М. : Горячая линия-Телеком, 2017. - 644 с.

7. Тихоненко, О. М. Модели массового обслуживания в информационных системах: учебное пособие для ВУЗов / О. М. Тихоненко. – Минск: Технопринт, 2003. – 327 с.

8. Шаньгин, В. Ф. Информационная безопасность и защита информации / В.Ф. Шаньгин. - М. : ДМК Пресс, 2017. - 702 с.

Рецензія/Peer review :

Надрукована/Printed :

Рецензент: д.т.н., проф. , проф. , кібербезпеки та комп’ютерни систем і мереж, Хмельницький національний університет, Мясіщев О.А.

Анотація

на наукову роботу під шифром «**Master 1**»

Об’єм роботи складає 47 сторінок. Робота містить 7 рисунків. 32 бібліографічних джерела.

Ключові слова: моделі, алгоритми, ефективність виявлення атак, метод, мережевий трафік, інформаційна безпека.

Актуальність роботи. Щорічно різні компанії, що надають послуги в галузі забезпечення інформаційної безпеки і протидії кібер-атакам, фіксують збільшення кількості DDoS-атак та їх потужність. Також збільшується кількість атак і до невеликих, «середніх» сайтів, які до недавнього часу не становили інтересу для зловмисників. Сьогодні у всьому світі стали звичайною ситуацією атаки на сайти державної влади напередодні виборів або важливих політичних подій. Засоби протидії, спеціалізовані саме на забезпечення безпеки невеликих і середніх ресурсів, отримали менший розвиток через переважання в минулому саме великих атак. І в даний час відстають від еволюції самих DDoS-атак.

Метою дослідження є створення **актуального** методу та інструментарію для раннього виявлення розподілених атак, спрямованих на відмову в обслуговуванні, і подальшого виявлення шкідливого трафіку на стороні ресурсу, що атакується і його блокування власними силами.

Для досягнення зазначеної мети поставлено і вирішено такі задачі: проведено моніторинг сучасних DDoS-атак; виявлено тенденцію до розвитку атак середньої і малої потужності, спрямованих на регіональні ресурси; досліджено особливості DDoS-атак регіонального рівня; вирішено завдання по створенню методу і програмного комплексу по виявленню DDoS-атак і шкідливих запитів.

Наукова новизна досліджень полягає в розробці методу раннього виявлення та протидії розподіленим атакам, спрямованих на відмову в обслуговуванні. Особливостями методу є: врахування сезонних періодів, орієнтація використання на кінцевому ресурсі, універсальність.

Об'єктом дослідження є комп'ютерні мережі і розподілені атаки, спрямовані на відмову в обслуговуванні, що здійснюються в цих мережах.

Предметом дослідження виступають моделі та методи виявлення розподілених атак, спрямованих на відмову в обслуговуванні, і виділення шкідливого трафіку цих атак.

Методи дослідження. В якості основних методів дослідження застосовувалися методи теорії ймовірності та математичної статистики, кластерного і системного аналізу, методи машинного навчання.