Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут телекомунікаційних систем
(повна назва інституту/факультету)

Кафедра телекомунікацій
(повна назва кафедри)

До захисту допущено
В.о. завідувача кафедри
Явіся В.С.

“04” червня 2020 р.

Дипломна робота

на здобуття освітнього ступеня «бакалавр»

Спеціальність 172 Телекомунікації та радіотехніка,

на тему: Аналіз методів виявлення та захисту від DDoS атак в мережах SDN

Виконав студент 4 курсу, групи ТМ-61
(шифр групи)

Єфименко Олексій Сергійович _____ _____
(прізвище, ім’я, по батькові) (підпис)

Керівник доцент, к.т.н. Валуйський Станіслав Вікторович _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент старший викладач Новіков В.І _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.
Студент _____
(підпис)

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут телекомунікаційних систем
(повна назва)

Кафедра телекомунікацій
(повна назва)

Освітній ступінь – бакалавр

Спеціальність 172 Телекомунікації та радіотехніка
(код і назва)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
Явіся В.С. _____
(ініціали, прізвище) (підпис)
«22» січня 2020 р.

ЗАВДАННЯ
на дипломну роботу студенту
Єфименко Олексію Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Аналіз методів виявлення та захисту від DDoS атак в мережах SDN,
керівник роботи Валуйський Станіслав Вікторович, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від «30» березня 2020 р. №924-с
2. Термін подання студентом роботи 04.06.2018
3. Вихідні дані до роботи: SDN мережі, технології захисту від DDoS атак
4. Зміст роботи:
   1) Аналіз побудови SDN мереж;
   2) Аналіз впливу і методів захисту від DoS атак в SDN мережах;
   3) Аналіз прозорої системи виявлення вторгнень (TIDS);
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) _____
6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 15 січня 2020 року.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Пошук та аналіз літератури	01.02.2020	
2.	Аналіз побудови SDN мереж	13.03.2020	
3.	Аналіз впливу і методів захисту від DoS атак в SDN мережах	26.04.2020	
4.	Аналіз прозорої системи виявлення вторгнень (TIDS)	03.05.2020	
5.	Оформлення дипломної роботи	03.06.2020	

Студент _____ Єфименко О.С.
(підпис) (ініціали, прізвище)

Керівник роботи _____ Валуйський С.В
(підпис) (ініціали, прізвище)

Реферат

Обсяг пояснювальної записки становить 50 сторінок, 3 ілюстрації, 2 таблиці та 37 джерел за переліком посилань.

Метою цієї роботи є аналіз методів виявлення та захисту від DDoS атак в мережах SDN

Завдання:

– Аналіз побудови SDN мереж;
– Аналіз впливу і методів захисту від DoS атак в SDN мережах;
– Аналіз прозорої системи виявлення вторгнень (TIDS);

Було опубліковано тези «Detecting DoS attacks in SDN networks using Shannon entropy» на конференції Проблеми телекомунікацій 2020.

Новизна: в ході виконання роботи були проаналізові існуючі механізми та технології захисту мереж передачі даних SDN від DDoS атак та визначений найбільш ефективний та масштабований метод – прозора система виявлення вторгнень (TIDS).

Структура роботи: Робота складається з реферату, змісту, списку умовних скорочень, вступу, трьох розділів, висновків до кожного розділу, загального висновку та списку використаних джерел.

Ключові слова: SDN, DDoS, захист від атак.

Abstract

The amount of the explanatory note is 50 pages, 3 illustrations, 2 tables and 37 sources for references.

The purpose of this work is to analyze DDoS attacks detection and protection methods in SDN networks

Task:

- Analysis of the construction of SDN networks;

- Analysis of the impact and methods of protection against DoS attacks in SDN networks;

- Analysis of a transparent intrusion detection system (TIDS);

It was published thesis "Detecting DoS attacks in SDN networks using Shannon entropy" conference issues of telecommunications in 2020.

Novelty: in the course of the work the existing mechanisms and technologies of protection of SDN data transmission networks from DDoS attacks were analyzed and the most effective and scalable method was identified - a transparent intrusion detection system (TIDS).

Structure of work: The work consists of an abstract, table of contents, list of abbreviations, introduction, three sections, conclusions to each section, general conclusion and a list of sources used.

Keywords: SDN, DDoS , protection against attacks.

Зміст

Перелік умовних скорочень

Англомовні скорочення:

SDN – Software Defined Network

DoS – Denial of Service

DDoS – Distributed Denial Of Service

API – Application Programing Interface

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

DNS – Domain Name System

HTTP – HyperText Transfer Protocol

ASs – Autonomous Systems

AITF – Active Internet Traffic Filtering

DPI – Deep Packet Inspection

RADIUS – Remote Authentication Dial-In User Service

IPS – Intrusion Prevention System

IDS – Intrusion Detection System

TIDS – Transparent Intrusion Detection System

PDU – Protocol Data Unit

NIC – Network Interface Card

EAL – Environment Detection System

CER – Configured Entropy Ratio

RIT – Rebalancing Timer

ВСТУП

За останні роки набирає обертів використання SDN (software-defined networks). На базі SDN мереж будуються великі масштабовані системи передачі даних, мережеві сервіси та хмарні обчислювальні середовища. За рахунок того, що SDN мережі мають суттєві відмінності від звичайних мереж - вони мають як переваги, так і недоліки в системах захисту інформації.

Останні розробки в SDN надають переваги та відкривають нові можливості для захисту мереж від DoS-атак. Однак, оскільки центральне керування є головною перевагою SDN, це також є єдиною точкою відмови в тому випадку, якщо зловмисникам вдається досягти своєї мети - створити надмірне навантаження на мережу та порушити нормальне функціонування мережевих пристроїв.

DoS атаки є одними з найпоширеніших типів атак, спрямованих на порушення функцій обслуговування та надійності мереж передачі даних. DoS-атаки генеруються за рахунок надмірно великої кількості небажаного мережевого трафіку або шляхом примушування обчислювальних ресурсів до опрацювання фейкових процесів та зберігання даних. Виявлення DoS-атак вимагає обробляти великі обсяги даних та здійснювати їх детальну перевірку. Це може створити слабке місце і порушити роботу інших служб, якщо це відбувається на шляху проходження трафіку та якщо обчислювальна потужність мережевих пристроїв недостатня. Тому для ефективного захисту необхідно використовувати масштабоване рішення для виявлення атак, яке може бути розгорнуте в найскладніших середовищах.

У цьому документі проаналізовані існуючі рішення, які використовують переваги багатоядерної обробки даних, механізми розподілення навантаження, що дозволяє їм подолати обмеження стандартного підходу до запобігання загрозам мережі та підтримувати безпеку, не порушуючи нормальної роботи мережі.

РОЗДІЛ 1 АНАЛІЗ ПОБУДОВИ SDN МЕРЕЖ

1.1 Архітектура SDN мережі

Нещодавно з'явилася нова парадигма управління мережами - Software Defined Networking (SDN). Ця технологія несе в собі гнучкі та адаптивні рішення по передачі і переадресації пакетів та програмування елементів мережі від центрального контролера. Ці властивості дозволяють розробити системи виявлення атак, які можуть легко розділити навантаження на декілька процесорів та здійснити дії зворотного зв'язку, які можуть зупинити атаку.

Новий підхід, на якому базується побудова SDN мереж базується на тому, що цей підхід роз'єднує площину управління та площину даних мережевих комутаторів [1]. Управління мережею в SDN залежить від площини управління, яка відіграє значну роль у продуктивності всієї мережі через незалежний від постачальника інтерфейс, такий як OpenFlow. Архітектура SDN складається з трьох рівнів, таких як рівень додатків, контрольний рівень та рівень інфраструктури [2]. На рівні додатків містяться програми, які керують мережею через northbound інтерфейс відповідно до політики, представленої адміністратором мережі. Контрольний рівень містить інтерфейс додатків мережі та контролера, який взаємодіє з мережевими елементами на інфраструктурному рівні через southbound інтерфейс. Рівень інфраструктури містить елементи мережі, такі як точки доступу та комутатори. Ці елементи відіграють важливу роль в переадресації пакета.

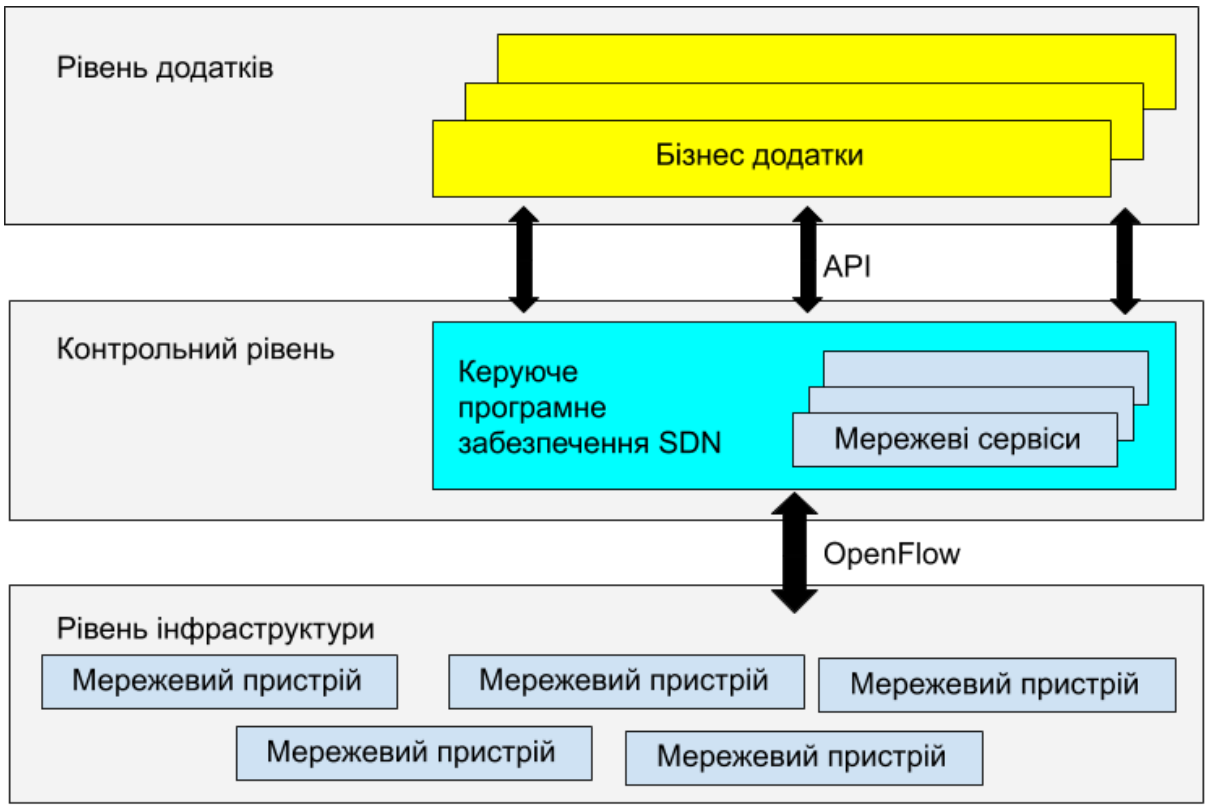

Рисунок 1 Архітектура SDN мережі

У SDN протокол OpenFlow відповідає за зв'язок між контролером та комутаторами через захищений канал [3]. OpenFlow визначає поведінку переадресації пакетів низького рівня в площині даних. Це допомагає розробникам програмувати мережу з більш високого рівня, не заглиблюючись у деталі обробки пакетів та пересилання в фізичні пристрої на нижчому рівні. Кожен мережевий пристрій має таблицю потоків, яка включає записи потоку в протоколі OpenFlow. Коли новий пакет досягне комутатора, він відповідатиме своїм таблицям. Якщо буде знайдено збіг, пакет буде передано за інструкцією в інший мережевий вузол. В іншому випадку пакет буде переданий з метою обробки в контролер. Контролер має два варіанти або додасть новий відтік (outflow) до таблиці, або додасть базу для скидання (overthrow) подібних пакетів. У випадку DoS атак, підроблені адреси в пакетах не матимуть відповідності в таблиці, тому вони будуть відправлені до контролера для обробки. Усі ресурси в контролері будуть схильні обробляти злоякісні пакети, якщо швидкість доступу пакетів до контролера висока. Контролер може бути повністю завалений атакою з

високою швидкістю, яка зробить його недоступним для законних користувачів. Це може призвести до руйнування архітектури SDN мережі.

1.2 Хмарні обчислювальні середовища

Хмарні обчислення - одна з передових технологій, яка дозволяє отримати величезний обсяг зберігання даних та надання послуг, в той же час вони пропонують високу продуктивність і низьку вартість. Хмарні обчислення, відомі як обчислювальна модель, яка керує діапазоном обчислювальних ресурсів з можливістю гнучкого налаштування. На основі моделей розгортання хмарні обчислювальні середовища можна класифікувати як середовища з публічним доступом, середовища з обмеженим доступом та гібридні середовища.

Хмарні обчислювальні середовища стали легкими мішенями для зловмисників для заволодіння інформацією користувачів. DoS-атаки є найбільшою загрозою для цієї галузі обчислень і найбільшою перешкодою для широкого впровадження хмарних обчислювальних середовищ. DOS-атаки вважаються одними з найбільш значущих проблем, що стоять перед зростанням популярності хмарних середовищ. Кілька причин роблять DoS-атаки серйозною загрозою: наприклад, така атака є руйнівною в цьому середовищі, і її легко здійснити. Крім того, зловмисники використовують підроблені IP-адреси, таким чином, відстежувати джерело нападу стає важко.

1.3 Висновки до розділу 1

В розділі 1 були розглянуті загальні принципи роботи SDN мереж, архітектура побудови SDN та протокол OpenFlow, за допомогою якого

мережеві пристрої передають системну інформацію один одному. Що допомогло в подальшому аналізі слабких місць SDN та вразливостей OpenFlow протоколу. Також була з’ясована класифікація хмарних обчислювальних середовищ та слабкі місця перед DoS атаками.

РОЗДІЛ 2 АНАЛІЗ ВПЛИВУ І МЕТОДІВ ЗАХИСТУ ВІД DOS АТАК В SDN МЕРЕЖАХ

2.1 Вплив DoS атак на хмарні середовища та мережі передачі даних

Зловмисники можуть запускати різні DoS атаки, які можуть впливати на приватні мережі, на загальнодоступні мережі, або на всі типи мереж. Ці атаки орієнтовані на такі ресурси, як пропускна здатність мережі, пам'ять та процесор, а також такі, що орієнтовані на додатки, такі як служба баз даних та веб-додатки. DoS-атаки генеруються за рахунок надмірно великої кількості небажаного мережевого трафіку або шляхом примушування обчислювальних ресурсів до опрацювання фейкових процесів та зберігання даних. Можуть трапитися три види впливу:

- Direct denial of service - операційна система хмарного обчислювального середовища розглядає додаткове навантаження на конкретну службу як спосіб працювати проти зловмисника.
- Indirect denial of service - атака направлена на обчислювальну потужність. Негативний вплив прямої атаки набувається шляхом перенасичення нерелеватною інформацією однієї служби, що впливає на інші служби, що працюють на тих же апаратних засобах, вони можуть страждати від навантаження, спричиненого атакою на іншу службу. Тому у випадку, коли

служба працює на одному апаратному засобі з іншими, це вплине на їх загальну доступність.

Основним впливом атаки за допомогою перенасичення нерелеватною інформацією (flooding) на хмарні середовища є стягнення плати з клієнтів за використання ресурсів. Це означає, що немає обмежень у використанні обчислювальної потужності.

Хмарні обчислення мають деякі властивості, які впливають на захист від DoS атак. Методи захисту узагальнені у трьох поняттях: виявлення, ідентифікація та фільтрація. По-перше, хмарні обчислювальні середовища забезпечені фізичними серверами, які керують мережевими та обчислювальними ресурсами замість користувачів. По-друге, з точки зору захисника, виділення ресурсів та міграція віртуальної машини є новими джерелами змін в топології мережі, і процеси швидко розвиваються. Отже, захист від DoS атак повинен мати можливість налаштовуватися на динамічну мережу, що має часті зміни в топології, зберігаючи високий показник швидкості виявлення та здатність швидко реагувати. По-третє, хмарні обчислення дозволяють усім користувачам ділитися однаковою мережевою інфраструктурою, що викликає потребу надійного розділення мережі. При традиційному захисті від DoS атак ця вимога не враховувалася. Операції виявлення та захисту від DoS атак не повинні впливати на роботу мережі і користувачів, але і користувачі не повинні впливати на ці операції.

Операційна модель хмарних середовищ представила інші завдання для DDoS атак, такі як динамічна топологія мережі та розширений периметр оборони. Для ефективного вирішення цих проблем адміністратор хмарного середовища повинен мати можливість легко доручити управління мережі користувачам хмарного середовища та швидко налаштувати управління на основі змін топології мережі. SDN забезпечує розширену логіку виявлення та легко реалізує результативні операції, а також забезпечує ефективність обробки пакетів. Тим часом, затримки в мережі та потоці трафіку, що виникають у зв'язку з керуванням мережею та змінами в топології,

наприклад, зв'язок між схемами захисту від DDoS та комутаторами, може генерувати нову хвилю атаки та призвести до відмови в одній точці мережі [29]. Щоб уникнути нової вразливості в безпеці SDN, слід врахувати втрати на обчислення та комунікаційні витрати при розробці рішення захисту від DDoS-атак. Підсумувавши можна сказати, що SDN може мати перевагу в захисті від DDoS-атаки на хмарні середовища та мережі передачі даних, коли всі комунікаційні та обчислювальні процеси будуть вивірені та оптимізовані.

2.2 Структура та класифікація DDoS атак

DDoS атака здатна знищити великі веб-сервіси, для яких зазвичай потрібні тисячі скомпрометованих машин. Структура такої розподіленої атаки є складною. Зловмисники завжди намагаються зробити одну чи обидві наступні речі [4]:

- Перша мета атаки - відключити з'єднання для законного користувача через споживання пропускної здатності мережевих ресурсів або ємності маршрутизатора (атаки мережевого і транспортного рівнів методом перенасичення)
- Друга мета атаки - відключити послуги для законного користувача через вичерпання ресурсів сервера (атаки перенасичення на рівні додатків).

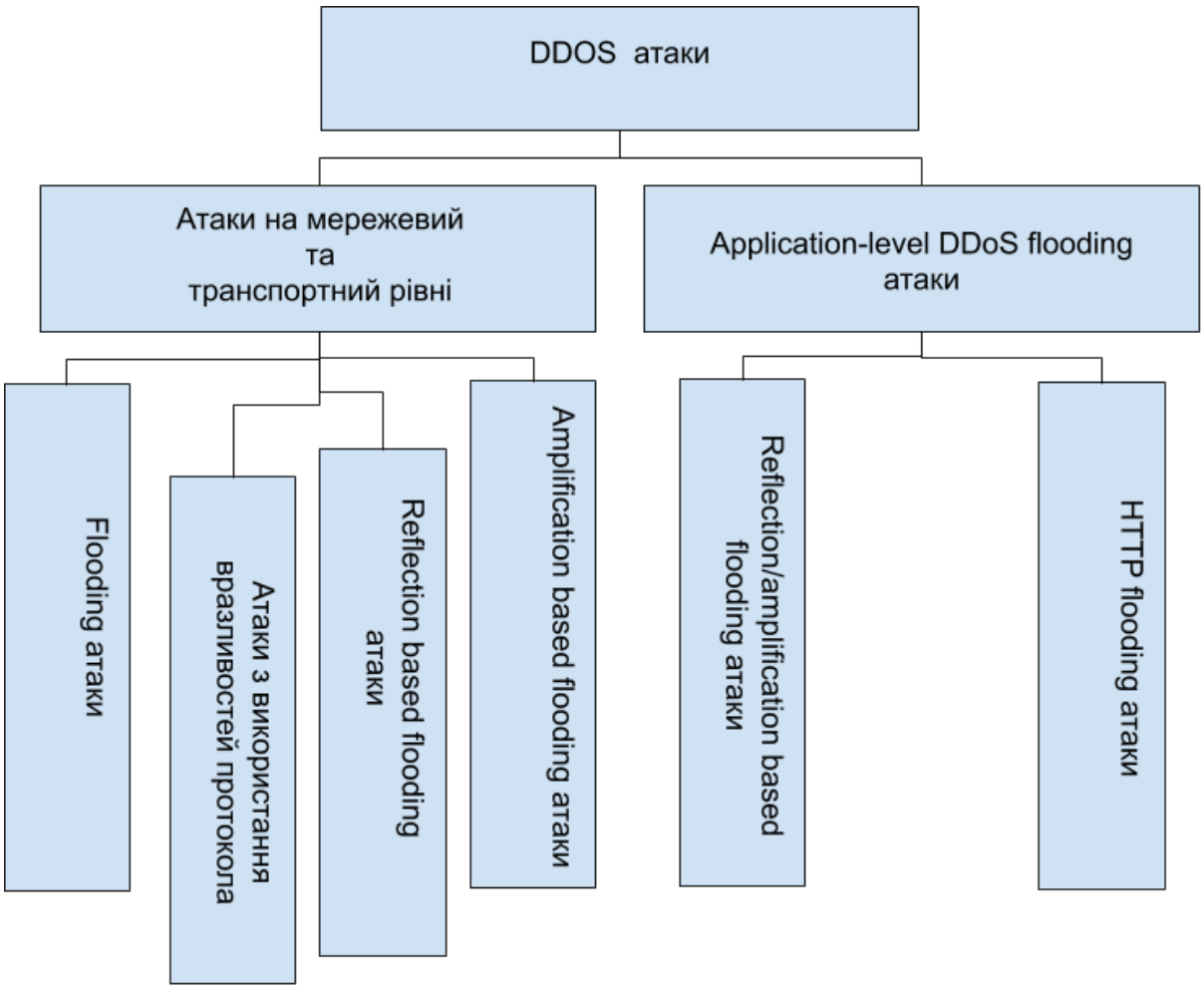

Рисунок 2 Класифікація DDoS атак

2.2.1 DDoS атаки на мережевий і транспортний рівні

DDoS-атаки на мережевий та транспортний рівні запускаються через UDP, TCP та DNS [5]. У цій категорії атаки класифікуються на чотири підкатегорії:

- Reflection-based flooding атаки: щоб вичерпати ресурси жертви, зловмисники надсилають сфальсифіковані запити, а не прямі запити.
- Атаки з використанням вразливостей протокола: з метою надмірного споживання ресурсів атакованої мережі,

зловмисники впроваджують помилки в мережеві протоколи атакованої мережі.

- Amplification-based flooding атаки: з метою посилення трафіку до атакованої мережі, зловмисники генерують кожне повідомлення, яке вони отримують.
- Flooding атаки: для розриву зв'язку з законним користувачем зловмисники вичерпують пропускну здатність мережі для атакованої мережі.

2.2.2 DDoS атаки на рівень додатків

Оскільки основною функцією цих атак є зрив роботи служб, вони споживають меншу пропускну здатність. Як правило, DDoS-атаки на рівні додатків мають однаковий вплив на сервіси, оскільки вони орієнтовані на конкретні характеристики в таких додатках, як DNS та HTTP.

- HTTP flooding атаки: використовуються щоб відключити веб-сервер, що є ціллю атаки, зловмисники надсилають величезну кількість HTTP-запитів.
- Reflection/amplification based flooding атаки: Зловмисники використовують ті самі методи на мережевому і транспортному рівнях з метою збоїв у системі трафіку.

2.3 Аналіз методів захисту від DDoS атак

В цьому розділі даної роботи висвітлюються різні методи пом'якшення DDoS атак та актуальні технічні проблеми. "Denial-of-Service атаки, можуть завдати серйозної шкоди інфраструктурі жертви нападу" [5].

Оскільки жоден законний користувач не може користуватися послугою, це призводить до величезних фінансових втрат та втрат продуктивності. Таким чином, необхідно створити механізми для пом’якшення впливу цих атак на сервіси мережі. У літературі представлено кілька механізмів пом'якшення DDoS-атак [6] [7].

Можна виділити чотири категорії механізмів захисту від DDoS-атак на мережевому та транспортному рівні на основі класифікацій [18]:

- Source-based механізми - розташовуються близько до джерел атаки, щоб зменшити шкоду для користувачів мережі від DDoS атак (фільтрація вхідного та вихідного трафіку).
- Network-based механізми - механізми розподіляються всередині мереж на маршрутизаторах автономних систем (Autonomous Systems - ASs) (наприклад, фільтрування пакетів на основі маршруту).
- Destination-based механізми - механізми виявлення та реагування на напад застосовуються в пункті призначення (тобто в атакованому вузлі).
- Hybrid (Distributed) механізми - розгортаються на кількох сайтах, включаючи джерело та місце призначення, наприклад, активна фільтрація інтернет-трафіку AITF.

На рівні додатків можна виділити дві категорії механізмів захисту від DDoS атаки на основі класифікацій [6]:

- Destination-based (server-side) механізми: використовують статистичні методи в DDoS-Shield як основний механізм захисту для виявлення характеристик HTTP сессій.
- Гібридні (розподілені) механізми: Public Turing test to tell Computers and Humans Apart (CAPTCHA) [9].

В [21] механізми захисту класифікуються на три основні типи:

- Proactive defense механізми - спочатку запропонована методика не починає боротьбу з DDoS атакою, а підготовлює базу правил

потоку, які застосовуються безпосередньо під час атаки на мережу. Це означає, що атакований потерпілий має доступ до ресурсів, які можуть протистояти такій атаці та функціонувати. У наші дні ця інфраструктура може бути хмарною, коли ресурси виділяються лише коли це потрібно, і ці ресурси можуть бути основою стратегії виживання, коли немає механізму захисту мережі.

- Reactive defense механізми - полегшують наслідки або перешкоджають DDoS атаці. Вони розпізнають наступну атаку за певними зразками поведінки мережі. Наприклад, система виявлення вторгнень (IDS) діє як рухомий екран (movement screen) та аналізатор [21]. Це означає, що даний механізм захисту від DDoS-атаки хороший лише як розподілений IDS.
- Post attack аналіз: основна мета аналізу після атаки - дослідити атаку та використовувати отримані дані для виявлення нападника [21].

2.4 Переваги SDN мереж в захисті від DDoS атак

SDN пропонує нову архітектуру мережі для хмарних обчислювальних середовищ та мереж передачі даних, яка має багато переваг для виявлення DDoS-атак [9][10]:

- Площини управління та даних відокремлені одна від одної. У SDN мережеві пристрої не мають функціоналу управління, і вони стали простими вузлами переадресації.
- Рішення щодо переадресації пакетів базуються на основі потоку трафіку. Набір значень пакетів та набір дій визначаються як потік. Потік у SDN - це послідовність пакетів між джерелом та пунктом призначення. Програмування потоку забезпечує безпрецедентну гнучкість, яка обмежується лише

можливостями потоку в таблицях реалізації. Швидке реагування на DDoS-атаки відбувається з динамічним оновленням правил переадресації. Нова або оновлена політика безпеки, заснована на аналізі трафіку, може бути розгорнута по всій мережі у вигляді правил потоку, щоб блокувати трафік атаки без зволікань.

- Передача логіки контролера зовнішньому об’єкту. Призначення контролера схоже на традиційну операційну систему, яка забезпечує основні ресурси та абстракції відповідно до логіки контролера для допомоги програмуванню пристроїв переадресації на основі абстрактного виду мережі та логічної централізації [22].
- Мережа програмована. Основна особливість SDN мережі - мережа налаштовується програмами, що працюють на контролері та взаємодіють з іншими мережевими пристроями.
- Логічно централізований контролер. У SDN контролер будує послідовну політику безпеки завдяки даним системи моніторингу та аналізу моделей трафіку на предмет можливих загроз безпеці мережі. Інформація, отримана під час аутентифікації (наприклад, через RADIUS сервер) та реєстрації користувачів, дає можливість організувати централізоване керування SDN для забезпечення динамічного блокування скомпрометованих хостів та аутентифікації легітимних хостів.
- Аналіз трафіку на основі програмного забезпечення дозволяє знаходити нові методи боротьби, оскільки для покращення можливостей комутатора можливо використовувати всі типи інтелектуальних алгоритмів, баз даних та будь-яких інших програмних засобів. Підозрілий трафік може бути спрямований на IPS для проведення Deep Packet Inspection [10].

Таблиця 1 Механізми захисту

Механізми	**Опис механізмів**	**Особливість**	**Застосування**
Source-based механізми	Дозволяє SDN контролерам ідентифікувати трафік атаки, фільтрувати скомпрометовані пакети або виявляти IP-адресу джерела атаки	Аналіз трафіку	Базова станція під контролем Openflow використовує аналіз в режимі реального часу для виявлення зловмисних програм [11]
		Програмованість	Програмованість домашнього роутера з використання OpenFlow-сумісних пристроїв для виявлення проблем в безпеці мережі [12]
Network-based механізми	Виявляють DoS атаки в потоці трафіку [7]	Аналіз трафіку	Ідентифікують зловмисний трафік за допомогою статистичних даних у таблиці потоків [8]
		Програмованість	Фреймворк безпеки FRESCO на базі OpenFlow [13]
Destination-based механізми	Використовує зворотнє IP трасування (IP trace back) для дослідження джерела та шляху розповсюдження	Динамічне оновлення правил безпеки	Метод покрокового розподілу [14] згідно вибіркових потоків для зворотнього IP трасування

	атак, хоча більшість технологій для розгортання зворотнього IP трасування важко реалізувати в мережі Інтернет	централізоване керування з Global views	NetSight - це платформа, яка використовує додатки для відновлення історії пакетів [15]

2.5 Аналіз існуючих рішень захисту SDN мереж від DDoS атак

Останнім часом з’явилося кілька запропонованих рішень, які були створені для зменшення наслідків від DDoS атак в SDN. Однак, невелика кількість з них достатньо зосереджена на механізмах захисту від DDoS-атаки. У таблиці 2 узагальнено можливі DDoS-атаки та існуючі рішення в SDN.

Таблиця 2 Існуючі рішення захисту

Тип DDoS атаки	Методи атаки	Наявні рішення
DDoS атака на рівень додатків	Через додаток	FortNOX [33]
	Через атаку на northbound API	
DDoS атака на контрольний рівень	Через атаку на контролер	FortNOX [33] Transport Security Layer (TLS) [34] AVANT-GUARD [35]
	Через атаку на northbound, southbound, westbound or eastbound API	
DDoS атака на рівень інфраструктури	Через атаку на комутатор	Transport Security Layer (TLS) [34] AVANT-GUARD [35] FLOODGUARD [36]
	Через атаку на southbound API	

У роботі [16] досліджено атаки за допомогою перенасичення нерелеватною інформацією (flooding) мереж та мережевих пристроїв SDN.

Початковим кроком є визначення мережі. У SDN контролеру потрібно більше часу для видачі та обробки нового запису в таблиці для нового пакету, ніж для вже відомих пакетів в порівнянні зі звичайними мережами, які зазвичай мають попередньо заповнену таблицю переадресації. Відповідно, не потрібно додаткового часу для обробки та введення потоку для нових пакетів. З цієї сторони SDN дає більше можливостей для зловмисників визначати, чи мережа побудована на архітектурі SDN чи ні, перевіряючи час відповіді. Останнім кроком є введення незвичайних пакетів у SDN, таким чином, час обробки контролером збільшується, а якість обслуговування зводиться до мінімуму через все більшу обробку записів потоку.

Оскільки можливі DDoS-атаки на рівень додатків можуть бути реалізовані через атакуючу програму або northbound API головним завданням є виявлення та узгодження конфліктних правил потоку, які накладаються через динамічні програми OpenFlow.

FortNox пропонує аутентифікацію на основі ролі та застосовує обмеження безпеки для захищеної авторизації, забезпечує цілісність процесу для NOX OpenFlow контролера для кожної Open-Flow програми [17].

OpenFlow пропонує підтримку шифрованого захисту безпеки транспортного рівня (TLS) та сертифікат обміну сертифікатами між контролером та комутаторами, а також можливість використовувати декількох моделей з декількома органами сертифікації [18]. Крім того, забезпечення взаємодії з шифруванням за допомогою реплік контролера може бути корисним для отримання корректного повідомлення від контролера [18]. Крім того, для забезпечення безпеки між площиною даних та пристроями площини управління можуть використовуватись динамічні та автоматизовані механізми пристроїв.

У SDN фреймворк OpenFlow ставить перед собою проблеми безпеки, наприклад, DDos може експлуатувати атакований контролер, щоб порушити роботу мережі.

AVANT-GUARD - це нова структура, яка пропонує вдосконалений захист та гнучкість в мережах OpenFlow, як розширення до площини даних OpenFlow, яка вирішує дві проблеми безпеки [19]. По-перше, використовує техніку міграції з'єднання на площині даних, щоб захистити інтерфейс між площиною управління та площиною даних та забезпечити взаємодію, що виникає під час DDoS-атак. По-друге, створення спрацьовуючих тригерів на службах збору статистики в площині даних, які дозволяють площині управління прискорити реагування та виявлення для зміни правил потоку, таким чином, додатки можуть реагувати на загрози через мережеву статистику.

У [20] наведені рішення для пом’якшення атак, які перевантажують інфраструктуру SDN (FLOODGUARD). Це рішення надає незалежний від протоколу захисний фреймворк, який використовує два методи запобігання атак: міграцію пакетів та активний аналіз потоку. Модуль міграції пакетів тимчасово зберігає підозрілі пакети як кеш та надсилає їх до контролера OpenFlow з обмеженою швидкістю, щоб захистити контролер від перевантаження.

В [21] представлена методологія на основі процедури фільтрації source-based IP адрес для боротьби з DDoS-атаками. Цей підхід працює з протоколом OpenFlow та вивчає клієнтський трафік для виявлення та запобігання таких атак. Методологія полягає у використанні (Т-таблиці) в контролері SDN для підтримки джерела IP-адрес для пакету, який передається комутатором. Незважаючи на те, що ця методика може зменшити наслідки DDoS-атак, вона недостатньо ефективна, коли трафік атаки надто потужний.

2.6 Висновки до розділу 2

В другому розділі були розглянуті основні принципи, за якими зловмисники за допомогою DDoS атак можуть впливати на побудовані на базі SDN мережі передачі даних та хмарні обчислювальні середовища. Були розглянуті DDoS атаки на мережевий та транспортний рівні, а також на рівень додатків, що допомогло знайти вразливі місця SDN мереж. На основі цього були знайдені переваги SDN мереж в захисті від DDoS атак а також розглянуті існуючі методи захисту. Були розглянуті базові принципи роботи існуючих рішеннь захисту (такі як TLS, AVANT-GUARD та інші), що допомогло у подальшому аналізі вразливостей SDN мереж та аналізі принципів роботи та переваг перед DDoS атаками Прозорої Системи Виявлення Вторгнень (TIDS).

РОЗДІЛ 3 АНАЛІЗ ПРОЗОРОЇ СИСТЕМИ ВИЯВЛЕННЯ ВТОРГНЕНЬ (TIDS)

3.1 Принципи роботи механізмів захисту від DoS атак в SDN мережах

Системи виявлення вторгнень (IDS) функціонують як пристрої пасивного контролю, які попереджають адміністратора мережі або іншого мережевого пристрою у разі підозрілої мережевої активності. З іншого боку, системи запобігання вторгнень (IPS) активно беруть участь в обробці трафіку, оскільки процес запобігання вторгнень вимагає від них контролю за передачею даних.

Обидва типи пристроїв безпеки можна класифікувати за їх підходом до аналізу трафіку або методологією виявлення загроз. Одним з часто використовуваних методів аналізу трафіку є повна перевірка пакетів (DPI), яка забезпечує найширший спектр можливостей для виявлення, оскільки процес виявлення може базуватися на будь-якій інформації, що міститься в заголовку або в полі даних пакета. Останнім часом спостерігається значний розвиток та ріст популярності в іншому підході до аналізу трафіку, який називається «flow-based detection», незважаючи на певні властиві проблеми ефективності, що стосуються механізмів експорту та відбору зразків потоку у високошвидкісних мережах. Однак було запропоновано кілька розширень, які повинні підвищити продуктивність виявлення атак на основі потоку трафіку, особливо в області DDoS-атак.

Залежно від методології виявлення загрози виявлення вторгнень може бути на основі підпису або аномалії. В алгоритмі виявлення на основі підписів припускають, що кожна атака, з різною точністю, може бути описана набором правил та моделей пакетів, які порівнюються сканером з кожним вхідним пакетом в режимі реального часу. Отже, підпис атаки повинен бути відомий до того, як трапиться атака, щоб правильно конфігурувати відповідний шаблон. Насправді це часто не так, і таку

стратегію зловмисники можуть легко використовувати на свою користь, встановивши, які шаблони розпізнаються пристроєм виявлення, і відповідно змінюючи налаштування атаки.

Інша група алгоритмів виявлення, заснована на аномалії, покладається на те, що під час атаки одне або більше значень мережевих параметрів будуть значно відрізнятися від вимірюваних значень параметрів мережі. Ці алгоритми вимагають від сканера пройти навчальний період для встановлення базових значень. Виявлення на основі аномалії потребує меншого втручання та налаштувань з боку адміністратора і, ймовірно, більш надійне з точки зору виявлення атак, які описуються у раніше невідомих моделях атак.

Незалежно від підходу до виявлення атак, IPS повинен виконувати свою функцію, не вводячи значних затримок, оскільки перевірка безпосередньо збільшує навантаження на мережеві пристрої та їх пропускну здатність. Пристрої IDS зазвичай незначно збільшують навантаження на мережеві пристрої та вводять не таку велику затримку, оскільки вони зазвичай отримують копію трафіку, який передається через комутатор.

Найбільш поширена точка розгортання IPS або IDS у захищеній мережі (тобто мережі, яка є ціллю атаки) максимально наближена до ймовірного джерела атаки. Таке розгортання збільшує шанс на позитивне виявлення, оскільки пристрій IPS або IDS має доступ до агрегованого трафіку, що дозволяє йому отримати цілісний вигляд атаки в разі нападу. З іншого боку, розміщення пристрою призводить до створення вразливого місця і являється єдиною точкою відмови для високошвидкісних мереж. Тому логічним рішенням цього питання є впровадження масштабованого пристрою, який розподілятиме навантаження процесора, необхідне для обробки пакетів, на декілька процесорів, зберігаючи при цьому високу ефективність. Сучасні пристрої безпеки (побудовані для виявлення або запобігання атакам) повинні працювати майже в реальному часі, щоб мати

можливість ефективно реагувати на загрози, пов’язані з великими обсягами трафіку, які часто зустрічаються в сучасних мережах.

Якщо процес виявлення ґрунтується на перевірці пакетів, то робота IPS та IDS споживає значну продуктивності мережевих пристроїв та додає затримку якщо IPS/IDS вставлена на шляху трафіка. Тому забезпечення ефективної обробки пакетів є важливою умовою таких систем IDS та IPS. Цього можна запобігти, розподіливши навантаження на декілька мережевих процесорів, а також за допомогою ефективної обробки пакетів у процесорах. Більшість алгоритмів розподілення навантаження розроблені для роботи з мережевими середовищами, які мають певні конфігурації хостів (наприклад, центри обробки даних з розподіленими веб-серверами тощо). Наприклад, деякі алгоритми базуються на розподілі запитів клієнтів порівну на кількість серверів дата-центра. Такі алгоритми за допомогою розподільного пристрою змінюють адреси призначення пакетів, тим самим перенаправляючи трафік до відповідного пункту призначення.

В [35] описане рішення, в якому використовується апаратний IDS, що містить спеціальний компонент розподілення трафіку, який розподіляє трафік на певну кількість датчиків, які відповідають за перевірку та перенаправлення пакетів. Для підвищення ефективності системи, яка використовує балансування навантаження, зв’язок між приймаючими вузлами повинен бути зведений до мінімуму, що є метою, яка часто підкреслюється в дослідженнях ([31], [31]).

Рішення, описане в цій роботі, в значній мірі покладається на переваги, які надають програмно-конфігуровані мережі SDN, що забезпечує високу ефективність та зручність використання в різних програмах [25], [26], [28]. Висока гнучкість налаштувань роблять його гарною платформою для впровадження адаптивних мережевих вузлів, які легко реагують на зміни в мережевому середовищі та топології мережі. Дослідження, опубліковані в [29] та [35], показують користь використання механізмів балансування навантаження в мережах SDN. Однак

продуктивність SDN є одним з найбільших недоліків, що часто згадується в [31] і [32], особливо для високошвидкісних мереж, які використовують реактивну обробку потоку трафіку або гранулярність великих потоків (large flow granularity). TIDS вирішує цю проблему, делегуючи більшість складних обчислень низці процесорів, тим самим знімаючи навантаження з контролера SDN.

3.2 Структура та принципи роботи TIDS

Основна мета, з якою створювалась TIDS (Transparent Intrusion Detection System) полягає у створенні активної, масштабованої та безпечної системи, побудованої на існуючому обладнанні. Система в першу чергу працює як IDS, розроблена з використанням методології виявлення атак на основі аномалії, але також забезпечує функцію запобігання DoS атакам. Безпека системи досягається через її прозорість (невидимість) для інших вузлів всередині мережі, підтримуючи повний контроль над трафіком, який проходить через систему, тим самим роблячи її стійкою до атак, спрямованих на саму систему.

Масштабованість таких систем досягається шляхом розподілу логіки обробки на довільну кількість процесорів за допомогою алгоритму балансування навантаження. Необхідно, щоб рішення працювало без порушень нормальної роботи мережі, незалежно від пропускної здатності або будь-яких інших характеристик.

3.3 Архітектура TIDS

TIDS складається з двох мережевих комутаторів (switches), основного SDN контролера та деякої кількості процесорів для обробки пакетів (сканерів), як показано на рисунку 3.

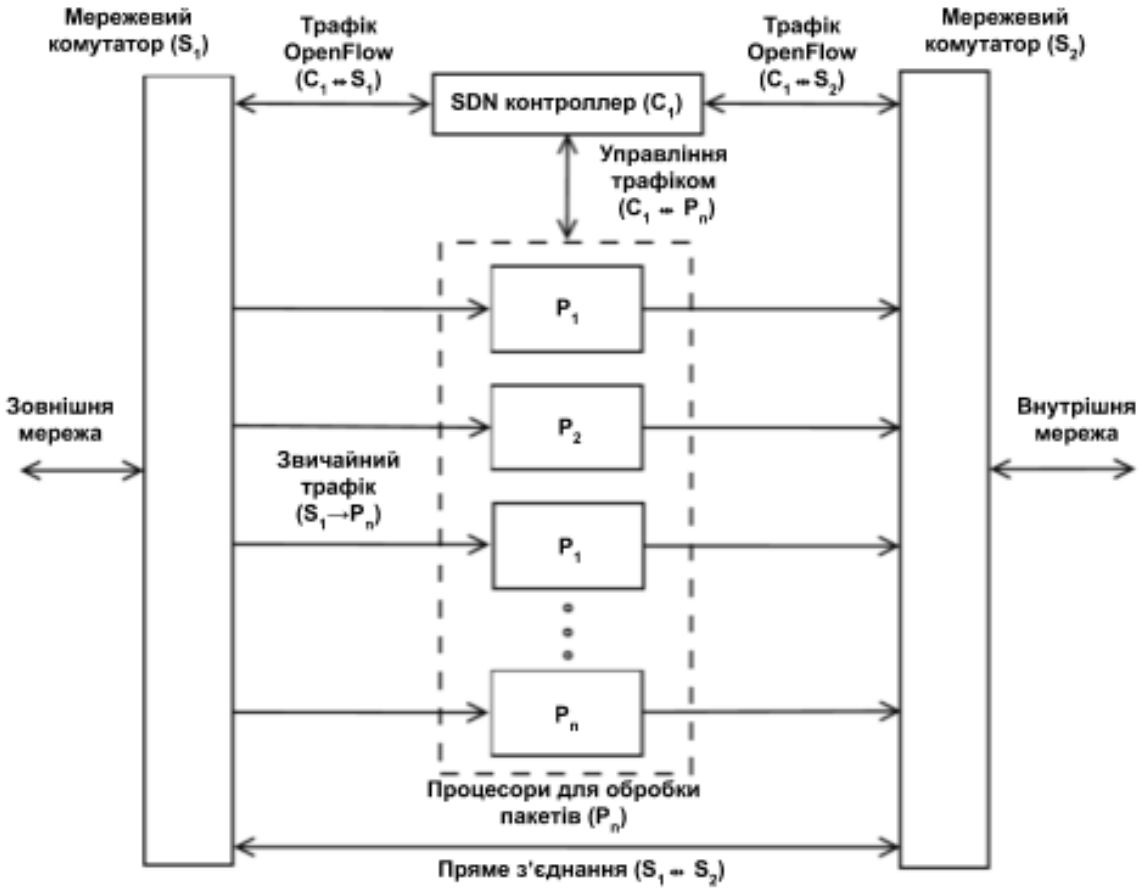

Рисунок 3 Архітектура системи виявлення вторгнень на базі SDN пристроїв (TIDS)

Мережеві комутатори (S_1 та S_2 на рисунку 1) представляють собою вхідну та вихідну точки системи виявлення вторгнень. Мережевий комутатор S_1 відповідає за розподілення трафіку, який надходить до системи виявлення вторгнень із зовнішньої мережі, а S_2 збирає трафік від сканерів та направляє трафік до внутрішньої мережі. Коли трафік надходить до системи виявлення вторгнень, він пропускається через процесори для обробки пакетів (P_1, P_2...P_n).

TIDS розроблений як детектор виявлення вторгнень, що надходять із зовнішньої мережі. Коли трафік потрапляє в TIDS, він розподіляється між активними вузлами обробки трафіку (процесори або сканери, показані на малюнку 1 як P_1, P_2 і т.д.).

Даний алгоритм використовує тільки інформацію, що зберігається в адресному полі мережевого рівня PDU (модель OSI), тож система не повинна бути пакетно-орієнтованою, також не потрібно застосовувати глибоку перевірку пакетів DPI. Кожний мережевий пристрій (сканер або процесор) обробляє вхідний трафік та визначає атаку використовуючи алгоритм, який базується на теоремі Шеннона. Якщо один з процесорів виявляє вище, ніж нормальне навантаження по трафіку, правила розподілення трафіку на комутаторі S_1 налаштовуються для балансування кількості трафіку, спрямованого на кожен зі сканерів. Це полегшує процес виявлення, оскільки TIDS використовує SDN лише для здійснення контролю та розподілу інформації, але обчислювально операції (такі як перевірка пакетів та обчислення) виконуються процесорами.

Трафік, який проходять через TIDS , можна розділити на три групи: трафік управління, трафік контролю та звичайний трафік. Система управління передає повідомлення управління між вузлами обробки та контролером C_1. Звичайний трафік - це той трафік, який підлягає аналізу. Всі з'єднання між S_1 і S_2, за винятком прямого зв'язку (показаного внизу малюнка 1), передають звичайний трафік від S_1 до S_2 пропускаючи через один з вузлів обробки вздовж шляху трафіку. З іншого боку, прямий зв'язок використовується для передачі трафіку, який не аналізується, але повинен передаватися між двома пристроями (між ними передається контрольний трафік). Крім контрольного трафіку, який включає різні види локально згенерованого широкомовного (broadcast) і адресного (unicast) трафіку, це посилання використовується для пересилання вихідного трафіку з внутрішньої мережі (нещодавно отримані запити або відповіді на попередні запити). Передбачається, що трафік, що надходить із захищеної мережі, є безпечним і що для цього трафіку не потрібно виконувати перевірку. Таким чином комутатори реалізують асиметричну маршрутизацію - зворотний трафік із внутрішньої мережі може бути спрямований на вихідну точку мережі без затримок. Це спрощує процедуру переадресації, оскільки логіка

врівноваження навантаження потрібна лише для S_1, тоді як S_2 лише агрегує та спрямовує трафік. У випадках, коли захищена мережа може містити джерело атаки, інший екземпляр TIDS може бути розміщений на тому самому каналі, через який трафік надходить від внутрішньої мережі). Імовірно, менша кількість хостів у захищеній мережі (порівняно з кількістю хостів в Інтернеті) спростила б виявлення зловмисної активності, оскільки зміна ентропії буде більш помітною. Прямий зв'язок між комутаторами S_1 та S_2 існує як резервне з'єднання незалежно від кількості активних вузлів обробки. Це гарантує, що у разі несправності вузла обробки трафік не втрачається через відсутність шляху переадресації.

Контролер C_1 - єдиний вузол, який безпосередньо спілкується з процесорами. Він контролює активні вузли та динамічно перерозподіляє навантаження, коли кількість активних процесорів змінюється під час роботи. Не існує зв'язку між вузлами обробки, що означає, що трафік повинний бути розподілений таким чином, щоб кожен вузол мав достатньо інформації для позитивної ідентифікації DDoS атаки. Процесори не знають про кількість вхідного трафіку, який він отримає, і не беруть активної участі у перерозподілі трафіку.

Процесори розміщуються як пропускні пристрої, які перевіряють та передають трафік далі, по своєму початковому шляху. Інформація, отримана при огляді пакетів, використовується для обчислення значень ентропії при розподіленні префіксів IP адрес джерел та призначення. Після того, як процесор виявляє атаку, набір адрес, з яких вона надходить, передається контролеру, який робить одну або декілька змін в логіці роботи комутатора S_1 в спробі пом'якшити атаку.

3.4 Компоненти програмного забезпечення TIDS

Є два основні програмні компоненти, що використовуються в TIDS. Перший компонент - це програмне забезпечення контроллера SDN, яке використовується для управління мережевим пристроєм. TIDS використовує контролер Ryu SDN [33]. Процесори є основними компонентами для прийому та обробки пакетів. Захоплення пакетів, як правило, базується на перериваннях, генерованих мережевою картою інтерфейсів (NIC), які обслуговуються процедурою, яка виконує захоплення. Для мереж з невеликою швидкістю цей режим роботи є оптимальним для забезпечення функціонування фреймворку захоплення, який має незначну вірогідність втрати пакетів, і що ніякого значного збільшення продуктивності не досягається за допомогою використання іншого рівня програмного захоплення (capturing software layer). Однак у високошвидкісних середовищах кількість часу, витраченого на обслуговування запиту на переривання, є надто значимою для будь-якого сервісу.

Device polling (опитування пристроїв) - це технологія, яка дозволяє частково або повністю зменшити повільну реакцію на обробку пакетів на основі переривання в середовищах, де потрібна низька затримка (або висока пропускна здатність). Замість того, щоб витрачати багато часу на обслуговування запитів переривань, які надходять від NIC, процессор маскує всі подальші переривання, згенеровані карткою, і запуская задачу, яка регулярно проводить опитування NIC. Це призводить до значного поліпшення ефективності та швидкості обробки пакетів. Прикладом бібліотеки опитування пристроїв є бібліотека DPDK Intel, яка використовується для впровадження процесорів. DPDK - це набір користувацьких бібліотек, які функціонують над рівнем абстракції (EAL), щоб забезпечити платформу для мережевих додатків працюючих у режимі опитування. Звіти та прес-релізи компанії Intel показують, що DPDK може

витримати великі навантаження в мережі, понад 40 Гбіт/с [34]. DPDK не пред'являє жодних спеціальних вимог до впровадження, крім сумісних NIC та операційної системи на базі Linux. Це дає змогу процесорам реалізовуватися як віртуальні пристрої. У віртуальних середовищах ємність системи (з точки зору наявної пропускної здатності) обмежена можливостями середовища віртуалізації, а не TIDS.

Як наслідок програмованості системи та контролю, який вона підтримує над вмістом різних полів заголовків пакетів, вся система може бути повністю прозорою для інших мережевих пристроїв. Пакети передаються без зміни будь-якого з полів всередині мережевого рівня та рівня зв’язку інформації , приховуючи тим самим факт, що пакет пройшов один або більше вузлів на своєму шляху. Сюди також входить трафік контролю між двома сусідніми вузлами мережі, які фізично були розділені TIDS. Тому вузли в системі TIDS не вимагають інформації про адресацію (окрім можливого інтерфейсу управління, який не вважається частиною системи). Це дозволяє уникнути цілої низки мережевих загроз, які націлені на систему IPS, намагаючись вивести з ладу всю мережу позаду IPS від решти світу.

3.5 Протокол обміну інформації

Протокол зв'язку, що використовується між процесорами та контролером, є простим протоколом, який передає дані управління. Повідомлення управління використовуються контролером SDN для запиту додаткової інформації від процесорів, або ж процесори виконують запит на контролер для модифікації таблиці маршрутизації на S_i, надсилаючи повідомлення про модифікацію потоку. Повідомлення інкапсульовані в заголовку Ethernet з ether-type полем, встановленим на поточне невикористане значення 0x9009[35], що дозволяють чітко відрізняти протокол від будь-якого існуючого протоколу третього рівня. Кожне

повідомлення містить поле типу, що вказує інкапсульовану команду. Визначено такі типи повідомлень:

- Повідомлення про збереження працездатності (keepalive) (тип 1) - використовується для повідомлення про статус роботи процесора для контролера. Процесори надсилають keepalive повідомлення через регулярні короткі інтервали конфігурованої довжини. Після отримання першого keepalive повідомлення від процесора він додає нововиявлений процесор до своєї локальної бази даних та додає відповідний порт до списку активних портів. Це дає можливість адміністратору розширити можливості системи, просто додавши необхідну кількість процесорів та підключивши їх до S_1. Якщо будь-якому з процесорів не вдалося надіслати 3 послідовних повідомлення keepalive, він визнається недоступним для контролера, і потік трафіку направляється на ішний вузол мережі. Повідомлення keepalive також містять кількість пакетів попереднього інтервалу, який використовується для збалансування навантаження.
- Повідомлення про вимкнення (тип 2) - використовується для попередження контролера про те, що процесор вимикається і більше не буде активним на цьому конкретному порту. Порт видаляється зі списку активних портів, і потоки перерозподіляються відповідно.
- Запит на модифікацію потоку (тип 3) - використовується для запуску змін в базі даних. Ці повідомлення надсилаються процесором, який виявляє атаку, і містить одну або більше мережеву адресу, яка беруть участь у атаці. . Контролер, у свою чергу, створює необхідні правила блокування на основі вмісту повідомлення.

- Повідомлення запиту / відповіді переліку префікс (prefix list) (тип 4) - використовується як контролером, так і процесорами для передачі детальної інформації про попередні налаштування мережі та кількість пакетів. Відправляючи це повідомлення, контролер запитує детальний перелік мереж призначення від процесора, до якого був відправлений щонайменше один пакет протягом попереднього інтервалу вимірювання. У відповідь процесор створює список, що містить кількість пакетів для кожного з попередніх пунктів разом з мережевою адресою, і передає його контролеру. Список використовується контролером для перерахунку навантаження на мережеві пристрої. Хоча контролер SDN може отримувати статистику для кожного з процесорів з комутатора, такий підхід знижує навантаження на комутатор і дає можливість процесорам опрацьовувати різну кількість пакетів при різних налаштуваннях. Ця інформація недоступна на комутаторі на ранніх стадіях балансування навантаження, через більшу ступінь деталізації, ніж вимагає процесор. Крім того, дослідження показують, що компоненти SDN можуть відображати певні неточності в розрахунках, які можуть мати небажаний вплив на обґрунтованість результатів.

3.6 Механізм виявлення

Алгоритм виявлення заснований на величині ентропії, обчисленій для адрес призначення трафіку, який кожен процесор отримує від комутатора. Ентропія розраховується за методом Шеннона, де p позначає ймовірність появи IP-адреси, як місця призначення пакету.

$$H = -\sum p * \log_2 p \qquad (1)$$

Основна ідея алгоритму полягає в тому, що зменшення значення ентропії для адрес призначення показує, що кількість трафіку, спрямованого на невелику кількість хостів, збільшилася і що зараз вона займає значну частину загального трафіку. Якщо зниження ентропії різке (воно відбувається за короткий проміжок часу), це може сигналізувати про те, що DoS атака розпочалась. Різниця в ентропії більш очевидна, коли трафік атаки займає більшу частку від загальної кількості трафіку (або коли інтенсивність фонового трафіку менша порівняно з трафіком атаки). Розподіляючи загальну кількість трафіку на декілька вузлів обробки, TIDS зменшує кількість фонового трафіку, який обробляється кожним процесором, тим самим підкреслюючи аномальний трафік та спрощує його виявлення. Крім того, процедура врівноваження тимчасово припиняється під час підозрілих мережевих дій (тобто коли один з процесорів сигналізує про те, що в даний час можлива атака), щоб запобігти перерозподілу трафіку атаки між процесорами. Перерозподіл зменшив би інтенсивність трафіку атаки на процесор, який виявив підозрілу активність, тим самим збільшивши значення ентропії, що зменшило б шанси на успішне виявлення атаки. TIDS розділяє інтервал моніторингу на часові вікна (також відомі як фрагменти), які містять статистику пакетів за цей часовий інтервал. Тривалість окремих фрагментів впливає на надійність та чутливість системи. Більш тривалі відрізки часу забезпечують стійкість до помилкових позитивних виявлень, які можуть з’являтися через короткі сплески мережевого трафіку, тоді як більш короткі часові відрізки збільшують чутливість системи.

Часові вікна використовуються процесорами для зберігання значень ентропії та підрахунку пакетів для кожного пункту призначення. Часові вікна розміщуються в циклічний список, так що додаткова пам'ять не повинна бути зарезервована, оскільки часові вікна заповнюються даними. Після закінчення останнього інтервалу найдавніший часовий відрізок очищається та використовується для нових даних. Загальна кількість

фрагментів перевищує максимальну кількість фрагментів, необхідних для позитивного виявлення , щоб необхідні дані не видалялися, поки вони ще потрібні процесору. Кожен процесор може працювати з трьома різними режимами роботи:

- Режим простою - це стан, який вводиться при запуску системи або коли процесор не отримує даних протягом короткого періоду часу. Тривалість цього режиму роботи впливає на процедуру виявлення і повинна бути невеликою часткою тривалості часового відрізка. Перебуваючи в цьому стані, процесор не виконує обчислення ентропії, оскільки базове значення було б неправильно інтерпретоване, і будь-який трафік, що надходить після цього стану, можна вважати трафіком атаки, якщо порівнювати зі значеннями ентропії, коли система не працює.
- Режим навчання - це стан, під час якого процесор збирає дані та налаштовує систему. Система завжди переходить у цей стан з режиму очікування після отримання першого пакету. Більш тривалий режим навчання враховував би більшу кількість попередніх фрагментів під час обчислення базової ентропії. Короткі зміни значення ентропії протягом періоду навчання не мали б значного впливу на систему (через більшу кількість фрагментів у навчальний період). Це призвело б до зменшення чутливості системи до коротших скачків кількості трафіку, що може призвести до помилкових позитивних виявлень. З іншого боку, маючи коротший термін навчання для встановлення базового режиму (baseline), система виявиться не сприйнятливою до коротких сплесків в обсязі трафіку, які не являються атаками. Вибір значення повинен здійснюватися з урахуванням шаблонів трафіку захищеної мережі в нормальному станы, і він повинен бути достатньо довгим, щоб

уникнути можливості випадкового вибору короткого сплеску трафіку як базового значення, за яким алгоритм виявлення буде встановлений.

- Посилений режим - це стан, під час якого процесор реагує на будь-яку підозрілу мережеву діяльність та виявляє вхідні атаки, порівнюючи поточні значення ентропії з встановленою baseline.

Після того, як процесор переходить у посилений режим, передбачається, що вже пройшла достатня кількість інтервалів навчання і що процесор має відповідні базові дані. Кількість пакетів для різних налаштувань кожного часового вікна зберігається в відсортованій хеш-мапі для кращої ефективності пошуку. Процес розміщення префіксів (тобто ключа в хеш-карті) та зберігання кількості пакетів є найбільш трудомістким розрахунком, який може ввести значну затримку і не повинен використовувати велику кількість обчислювальних ресурсів, навіть при розгортанні, що складається з невеликої кількості попередніх префіксів. Після закінчення терміну дії часового вікна, кількість пакетів з мапи використовується для обчислення поточного значення ентропії. Потім це значення порівнюється із середнім значенням трьох попередніх відрізків часу без нападу (також відомих як "безпечне" значення), обчислюючи їх відношення. Основна мета початкового режиму навчання - забезпечити, щоб система пройшла щонайменше три відрізки часу без нападу. Оскільки в цей час ще є недостатні дані, адміністратор повинен переконатися, що атака зараз не виконується протягом навчального періоду. Якщо розрахований коефіцієнт ентропії нижче, ніж значення корегованого порогового значення (його також називають конфігурованим коефіцієнтом ентропії (CER), процесор надсилає перший сигнал тривоги, що вказує на можливу атаку. Важливо зазначити, що значення CER виражається у відсотках від безпечного значення (середнє значення від трьох попередніх часових вікон без нападів) тим самим забезпечуючи надійність системи, дозволяючи їй реагувати на атаки адаптивним чином. У наступному

часовому відрізку повторюється той же розрахунок (знову ж таки, порівнюючи поточну ентропію адреси призначення з безпечним середнім значенням). Це робиться для того, щоб уникнути класифікації коротких сплесків трафіку з однієї IP-адреси як атаки. Якщо значення ентропії зберігається під час другого відрізка часу, воно законно розглядається як атака і процесор надсилає другий сигнал тривоги. Допускається невелика різниця між значеннями ентропії на етапах тривоги 1 і 2, щоб запобігти впливанню незначних змін кількості трафіку і не порушити процес виявлення (тобто коротке зменшення об'єму трафіку атаки може впливати на значення ентропії). Якщо нове значення ентропії повернеться вище заданого порогу, процесор припустить, що атака вже пройшла, і поверне систему в безпечний стан. Кількість часових вікон для підтвердження атаки (2 відрізки від першого відхилення від ентропії) вибирається для того, щоб пристосуватися до різких змін в трафіку (які трапляються протягом хвилини). Також впроваджено додатковий механізм, який не дозволяє зловмиснику обходити механізм виявлення, поступово збільшуючи інтенсивність атаки протягом більш тривалого періоду часу (10 і більше часових відрізків). Якщо зміни в ентропії та кількості пакетів протягом цього періоду є монотонними, а фінальна ентропія знаходиться нижче порогу CER, спрацьовує сигнал тривоги та система перейде до стадії зменшення атаки.

3.7 Пом’якшення наслідків атаки

Після того, як атака була підтверджена, процесор витрачає ще один відрізок часу, щоб визначити тип атаки. Визначення типу атаки передбачає підрахунок пакетів, спрямованих на атакованих хостів, та групування підрахунків за IP-адресою джерела. TIDS починає моніторинг адрес джерела лише після підтвердження атаки та виявлення атакованого пункту

призначення. Це знижує потреби в пам'яті для TIDS, особливо під час нормальної роботи (без активної атаки). Хост джерела вважається підозрілим (можливим зловмисником), порівнюючи його кількість пакетів із середнім числом пакетів для всіх відстежених джерел.

Якщо напад приходить від невеликої кількості хостів (тобто атака не поширюється), TIDS може легко пом'якшити цю атаку, заблокувавши підозрілі джерела, не порушуючи роботу сервісів для постійних користувачів. Якщо налаштоване значення кількості хостів зловмисників менше фактичної кількості джерел атаки, кількість пакетів яких перевищує обчислену середню кількість, системі знадобиться більше часу для повного блокування атаки (після того, як перший набір джерел атаки буде заблокований, TIDS виявить ту саму атаку яка цього разу походить від різного набору нападників). Отже, для цього параметра немає верхньої межі, за винятком тієї, яка накладається на об'єм пам'яті, доступної для TIDS. TIDS здатний одночасно відстежувати велику кількість атакованих хостів та пом’якшити всі атаки, які зараз тривають. Після виділення джерела та місця атаки процесор надсилає запит на зміну потоку трафіку, що містить список адрес, які забороняються. Заборони встановлюються протягом обмеженого періоду часу, після чого вони автоматично вимикаються комутатором. У системі застосовується схема пеналізації, яка подвоює попередню тривалість заборони для повторних атак, що походять від одного і того ж хоста щоб мінімізувати наслідки повторних атак. Початкова тривалість заборони також є впевненою (встановлюється на 300 секунд для тестування) і повинна бути принаймні довшою за час, необхідний для виявлення нападу. У разі розподілених DoS-атак, атака зазвичай надходить з великої кількості адрес. Повністю викорінити цей тип атаки практично неможливо, не дозволяючи атакуючим хостам досягти мети, оскільки ізолювати зловмисників від законних користувачів важко, не аналізуючи дані на рівні додатків. Однак, як було зазначено раніше, це може суттєво збільшити складність системи та погіршити її ефективність. Неможливість

відрізнити зловмисні хости від звичайних користувачів викликана тим, що один атакуючий хост виконує лише декілька запитів, як правило, не більше ніж звичайний користувач, який не бере участь в атаці.

Іншим важливим моментом є процес відновлення після виявлення нападу та вжиття відповідних заходів у відповідь. Якщо алгоритм балансування навантаження тимчасово призупинено після сигналу нападу, може пройти достатньо часу, поки він не буде відновлений знову, через час налаштування нових правил розподілу. Щоб пришвидшити процес відновлення, контролер видаляє всі додаткові правила та розпочинає процедуру розповсюдження спочатку (тобто поточні правила розповсюдження виконуються, але попередньо встановлені заборони залишаються в силі). Це не впливає на процес виявлення, але пришвидшує процес повернення в збалансований стан.

3.8 Механізм балансування навантаження

Балансування навантаження - це техніка масштабування ємності, яка часто використовується в комп'ютерних мережах, протоколи маршрутизації обчислюють одну і ту ж метрику на більш ніж одному шляху одночасно. Оскільки пристрої балансування навантаження розміщуються на шляху трафіку, алгоритми балансування навантаження повинні бути ефективними та додавати незначну затримку. Отже, найчастіше використовуються методи, засновані на хеші, вхідний трафік приймається, аналізуються деякі ключові частини заголовка пакета, і пакети розподіляються на кілька виходів. Ефективність алгоритму залежить від статистичних властивостей трафіку та відповідного алгоритму. Більшість потоків, з яких складається мережевий трафік, є короткочасними, вони не вносять суттєвого вкладу в об'єм трафіку(більше 80% потоків тривають менше 10с)[36]. Однак є невеликий відсоток довготривалих потоків, які складають значний відсоток від загального об’єму. Існування таких довготривалих потоків представляє серйозний виклик для розподілення трафіку рівномірно по декількох

шляхах, оскільки будь-який алгоритм узгодження, заснований на хеші, навряд чи може розділити такий потік на паралельні шляхи або співставити його пропускну здатність з іншими переданими потоками на альтернативних шляхах. RFC 7424[37] досліджує деякі вказівки для досягнення максимально наближеного до оптимального балансування навантаження.

Алгоритм, представлений у цій роботі, був розроблений для вирівнювання навантаження на кожен з активних процесорів та для того, щоб адміністратор міг збільшити потужність системи, включивши додаткові процесори. У той же час він розроблений так, щоб не перешкоджати процедурам виявлення вторгнень або не піддавати непотрібне навантаження самому контролеру SDN. Система передбачає, що будь-яка спроба (D)DoS-атаки спрямована на одного або декілька конкретних хостів всередині захищеної мережі. Таким чином, будь-який трафік, призначений одному і тому ж хосту, буде переданий через те саме з'єднання компонентів і оброблятися тим самим процесором в будь-який момент часу. Його можна перебалансувати на інший процесор, лише якщо система зробить висновок, що немає ризику нападу. Виходячи з своєї роботи, цей тип балансування навантаження класифікується як серверний, на відміну від типів по запиту,по потоку або по адресі призначення, знайдених у літературі.

У високошвидкісних середовищах обробка кожного нового потоку індивідуальна, з реактивним налаштуванням власних правил, викликаних вхідним трафіком, може виявитись обчислювально дорогою. Отже, при запуску контролера, TIDS проактивно генерує правила резервної переадресації, які розділяють трафік на регіони на основі бітів низького порядку цільової IP адреси, при цьому кількість бітів, що використовуються в масці, залежить від кількості активних процесорів (найменше число бітів n, для яких 2_n більша або дорівнює кількості процесорів). Хоча результуючі навантаження на деякі процесори можуть бути неоднаковими, кількість

правил, необхідних для адаптації схеми розподілу, суттєво менша, ніж це було б, якби кожен з потоків оброблявся індивідуально. Кожен новий процесор (ідентифікований повідомленням keepalive, отриманим через раніше неактивний порт) призведе до того, що контролер повторно врівноважить трафік. Ці правила мають нижчий пріоритет, вони функціонують як резервні маршрути у випадках, коли не існує конкретного правила.

Після початку роботи TIDS та встановлення початкових правил, контролер несе відповідальність лише за відновлення балансу одного або декількох префіксів щоб врахувати будь-які розбіжності між навантаженнями різних процесорів. Трафік врівноважується за допомогою перенаправлення префіксів певної довжини від одного компоненту мережі до іншого, базуючись на поточному навантаженні кожного. Кожен з процесорів відслідковує кількість пакетів для кожного префіксу мережі призначення протягом інтервалу спостереження між двома послідовними keepalive повідомленнями. Кожне повідомлення keepalive, яке надсилає процесор, містить підсумковий підрахунок всіх пакетів, оброблених протягом попереднього інтервалу. Після отримання повідомлення keepalive від кожного з процесорів контролер порівнює зібрані кількості пакетів. Якщо зустрічається розбіжність, що перевищує поріг дисбалансу, контролер надсилає всім процесорам запит списка префіксів для детального аналізу та підрахунку кількості пакетів. Виходячи з цих відповідей, контролер здійснює перебалансування, встановлюючи ряд правил більш високого пріоритету, які перенаправляють конкретні префікси з перенавантаженого вузла на вузол з найменшим навантаженням. Алгоритм вибирає префікси шляхом оцінки кількості пакетів після завершення процедури перенаправлення (виходячи з кількості пакетів для кожного префіксу за попередній відрізок часу). Префікси розміщуються у списку у порядку зменшення кількості пакетів, щоб мінімізувати кількість ітерацій, необхідних для відновлення навантаження. Після встановлення нових

правил контролер переходить в інтервал відновлення, протягом якого не допускається перерахунок навантаження, незалежно від підсумкових підрахунків. Цей інтервал використовується для підвищення стабільності системи за рахунок запобігання частих перерахунків. Цей інтервал можна відрегулювати в TIDS через значення параметра, що називається таймером зрівноваження (RIT).

3.9 Висновки до 3 розділу

В третьому розділі була проаналізована прозора система виявлення вторгнень TIDS. Були досліджені принципи роботи, архітектура системи, робота мережевих елементів (комутатор, контролер та інші), взаємодія з протоколом OpenFlow. Був розглянутий механізм розподілення навантаження на мережеві пристрої, виявлено слабкі місця алгоритма перерозподілу навантаження. Підсумувавши, можна сказати, що на даний момет TIDS являється найбільш досканалою та ефективною системою виявлення вторгнень та пом’якшення наслідків DDoS атак в SDN мережах, виходячи із знайдених недоліків та сильних сторін.

Висновки

По темі «Аналіз методів виявлення та захисту від DDoS атак в мережах SDN» були розглянуті принципи роботи та архітектура SDN мереж, був проведений аналіз існуючих типів DDoS атак та методів їх впливу на мережі передачі даних, аналіз методів захисту від DDoS атак, аналіз існуючих рішень для захисту від DDoS атак. В результаті чого був встановлений найбільш оптимальний механізм захисту від DDoS атак – TIDS (Прозора система виявлення вторгнень), яка досягає максимальної ефективності за рахунок багатьох факторів (повної невидимості для інших пережевих пристроїв, масштабованості, гнучким налаштуванням, ефективному використанню протокола OpenFlow та обміну внутрішніми сигнальними повідомленнями з іншими елементами системи виявлення вторгнень).

Поставлена мета роботи досягнута, отримані результати відповідають сформульованим завданням та задовольняють їх.

Для досягнення поставленої мети дослідження вирішено наступні задачі:

1) У першому розділі успішно проаналізована архітектура та принципи роботи SDN мереж.
2) У другому розділі увага надається аналізу впливу і методів захисту від DoS атак в SDN мережах
3) У третьому розділі проаналізована прозора система виявлення вторгнень (TIDS), знайдені переваги в захисті від DDoS атак та показані вразливі місця.

Список використаних джерел

[1] Tri, N., Hiep, T., & Kim, K. (2015, January). Assessing the impact of resource attack in Software Defined Network. In Information Networking (ICOIN), 2015 International Conference on (pp. 420-425). IEEE.

[2] “Sdn architecture,” june 2014, accessed: 2014-09-12. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdnresources/technicalreports/TR SDN ARCH 1.0 06062014.pdf

[3] “SDN Central”. 2014, [Online]. Available at: http://www.sdncentral.com/announced-sdn-products/

[4] Saman Taghavi Zargar, James Joshi, and David Tipper, “A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks,” IEEE Comm. Survey & Tutorials, vol. 15, no. 4, pp. 2046-2069, Fourth Quater 2013.

[5] M. Sachdeva, G. Singh, K. Kumar, and K. Singh, “Measuring impact of ddos attacks on web services,”2010.

[6] S. T. Zargar, J. Joshi, and D. Tipper, “A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks,” IEEE Commun. Surveys & Tutorials, vol. 15, no. 4, pp. 2046–2069, Fourth Quarter 2013.

[7] S. Farahmandian, M. Zamani, A. Akbarabadi, J. M. Zadeh, S. M. Mirhosseini, and S. Farahmandian, “A survey on methods to defend against DDoS attack in cloud computing,” in Proc. Recent Advances in Knowledge Engineering and System Science, Feb. 2013.

[8] R. Braga, E. Mota, and A. Passito, “Lightweight DDoS flooding attack detection using nox/openflow,” in Proc. 35th IEEE Conf. Local Computer Networks (LCN), 2010, pp. 408–415. [21] G. Zhang and M. Parashar, “Cooperative defence against ddos attacks,” Journal of Research and Practice in Information Technology, vol. 38, no. 1, pp. 69–84, 2006.

[9] Yan, Q., Yu, R., Gong, Q., & Li, J. (2015). Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges.

[10] Q. Yan and F. R. Yu, "Distributed denial of service attacks in software defined networking with cloud computing," IEEE Commun. Mag., vol. 53, no. 4, pp. 52–59, Apr. 2015.

[11] R. Jin and B. Wang, "Malware detection for mobile devices using software-defined networking," in IEEE Second GENI on Research and Educational Experiment Workshop (GREE), 2013, pp. 81–88.

[12] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting traffic anomaly detection using software defined networking," in Proceedings of Recent Advances in Intrusion Detection, 2011, pp. 161–180.

[13] S. Shin, P. A. Porras, V. Yegneswaran, M. W. Fong, G. Gu, and M. Tyson, "Fresco: Modular composable security services for softwaredefined networks." in Proc. ISOC Network and Distributed System Security Symposium, 2013.

[14] H. Tian and J. Bi, "An incrementally deployable flow-based scheme for IP traceback," IEEE Commun. Letters, vol. 16, no. 7, pp. 1140–1143, July 2012.

[15] N. Handigol, B. Heller, V. Jeyakumar, D. Mazieres, and N. McKeown, "I know what your packet did last hop: using packet histories to troubleshoot networks," in Proc. Symp. Networked Systems Design and Implementation (NSDI), 2014.

[16] S. Shin and G. Gu, "Attacking Software-defined networks: a first feasibility study," in Proc. the second ACM SIGCOMM workshop on Hot topics in software defined networking, 2013.

[17] P. Porras et al., "A Security Enforcement Kernel for OpenFlow Networks," Proc. 1st Wksp. Hot Topics in Software Defined Networks, 2012, pp. 121–126.

[18] B. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, “A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks,” IEEE Commun. Surveys & Tutorials, vol. 16, no. 3, Third Quarter 2014, pp. 1617–34.

[19] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, “Avant- Guard: Scalable and Vigilant Switch Flow Management in Software-Defined Networks,” Proc. ACM SIGSAC Conf. Computer & Commun. Security, 2013, pp. 413–24.

[20] Wang, H., Xu, L., & Gu, G. (2015, June). FloodGuard: A DoS Attack Prevention Extension in Software-Defined Networks. In Dependable Systems and Networks (DSN), 2015 45th Annual IEEE/IFIP International Conference on(pp. 239-250). IEEE.

[21]Dao, N. N., Park, J., Park, M., & Cho, S. (2015, January). A feasible method to combat against DDoS attack in SDN network. In Information Networking (ICOIN), 2015 International Conference on (pp. 309-311). IEEE.

[22] H. Lai, S. Cai, H. Huang, J. Xie, H. Li, A parallel intrusion detection system for high-speed networks

[23] K. Xinidis, I. Charitakis, S. Antonatos, K. G. Anagnostakis,

E. P. Markatos, An active splitter architecture for intrusion detection

and prevention, IEEE Transactions on Dependable and

Secure Computing 3 (2006)

[24] J. Coppens, S. V. den Berghe, H. Bos, E. P. Markatos, F. D.

Turck, A. Oslebo, S. Ubik, Scampi: A scalable and programmable

architecture for monitoring gigabit networks, management of multimedia networks and services

[25] R. Kanagavelu, B. S. Lee, R. F. Miguel, L. N. T. Dat, L. N.

Mingjie, Software defined network based adaptive routing for

data replication in data centers

[26] S. Fang, Y. Yu, C. H. Foh, K. M. M. Aung, A loss-free multipathing solution for data center network using software defined networking approach

[27] A. Tavakoli, M. Casado, T. Koponen, S. Shenker, Applying nox to the datacenter, in: Proceedings of workshop on Hot Topics in Networks (HotNets-VIII), 2010.

[28] T. D. Nadeau, K. Gray, SDN: Software Defined Networks, OReilly Media, Inc., 2013.

[29] M. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, D. R. Cheriton, Policycop: An autonomic qos policy enforcement framework for software defined networks

[30] S. Namal, I. Ahmad, A. Gurtov, M. Ylianttila, Sdn based inter technology load balancing leveraged by flow admission control

[31] B. Nunes, M. Mendonca, N. Xuan-Nam, K. Obraczka, T. Turletti, A survey of software-defined networking: Past, present, and future of programmable networks

[32] A. Lara, A. Kolasani, B. Ramamurthy, Network innovation using openflow

[33] R. S. F. Community, Ryu sdn controller, http://osrg.github.io/ryu/ (2014).

[34] Intel, Dpdk performance report, http://www.intel.com/content/www/us/en/intelligentsystems/intel-technology/intel-dpdk-programmers-guide.html (2013).

[35] I. S. Association, Ethertype,

http://standards.ieee.org/develop/regauth/ethertype/eth.txt (2014).

[36] S. Sengupta, A. Greenberg, P. Patel, R. Chaiken,

The nature of data center traffic: Measurements & analysis

[37] R. Krishnan, L. Yong, A. Ghanwani, N. So, B. Khasnabish,

Mechanisms for optimizing link aggregation group (lag) and

equal-cost multipath (ecmp) component link utilization in network