НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ
КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ І
СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ

«На правах рукопису»
УДК _____

«До захисту допущено»

Завідувач кафедри СПСКС

_____ В.П.Тарасенко_
(підпис) (ініціали, прізвище)

"___"_____2018р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 123 Комп'ютерна інженерія
Комп'ютерні системи та компоненти

на тему СИСТЕМА ЗАХИСТУ СЕРВІСІВ МЕРЕЖІ SDN З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ NFV

Виконав: студент ІІ курсу, групи КВ-71мп

Кателіков Владислав Ігорович _____

Науковий керівник:

доцент кафедри СПіСКС, к.т.н., доцент Щербина О.А. _____

Рецензент: професор кафедри ОТ, д.т.н., професор Кулаков Ю.О._____

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

Київ – 2018 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра прикладної математики

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність (спеціалізація) – 123 «Комп'ютерна інженерія» («Спеціалізовані комп'ютерні системи»)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ В.П. Тарасенко

«___»_____2018 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Кателікову Владиславу Ігоровичу

1. Тема дисертації «Система захисту сервісів мережі SDN з використанням технології NFV», науковий керівник дисертації Щербина О.А., к.т.н., доцент, затверджені наказом по університету від «___»_____ 2018 р. №_____

2. Термін подання студентом дисертації «14» грудня 2018 р.

3. Об'єкт дослідження: захист сервісів програмно-конфігуровних мереж SDN з використанням NFV технологій.

4. Предмет дослідження: системи захисту сервісів моделей хмарного обчислення програмно-конфігуровної мережі, з віртуалізованими функціями.

5. Перелік завдань, які потрібно розробити:
- провести аналіз існуючих рішень захисту SDN/NFV мереж;
- провести аналіз вразливостей SDN та NFV технологій;
- провести аналіз захисту програмно-конфігуровних мереж;
- провести аналіз захисту технологій віртуалізованих функцій;
- розробити власну систему захисту
- змоделювати роботу тестової мережі з використанням імітаційних атак
- проаналізувати отримані результати

6. Орієнтовний перелік графічного (ілюстративного) матеріалу:
- результати роботи графічної частини;
- код імітаційної атаки на переповнення буферу;
- код імітаційної атаки на відмову у обслуговуванні;

7. Орієнтовний перелік публікацій:
- Тези доповіді " Порівняльний аналіз основних проблем безпеки в комп'ютерних мережах SDN при використані технології NFV"
- Тези доповіді " Система захисту мережі SDN з використанням технології NFV"

8. Консультанти розділів дисертації

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

9. Дата видачі завдання «04» жовтня 2017 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Ґрунтовне ознайомлення з предметною галуззю	17.10.2017	
2.	Визначення структури магістерської дисертації; вивчення літератури, пошук додаткової літератури, патентний пошук	04.12.2017	
3.	Робота над першим розділом магістерської дисертації; проведення наукового дослідження	15.02.2018	
4.	Проведення наукового дослідження; робота над другим розділом магістерської дисертації; розроблення програмного забезпечення	05.04.2018	
5.	Проведення наукового дослідження; робота над статтею за результатами наукового дослідження	15.05.2018	
6.	Проведення наукового дослідження; робота над третім розділом магістерської дисертації	15.06.2018	
7.	Завершення роботи над основною частиною магістерської дисертації; підготовка матеріалів доповіді на конференції ПМК-2018	05.11.2018	
8.	Оформлення текстової і графічної частини магістерської дисертації	04.12.2018	

Студент В.І. Кателіков

Науковий керівник дисертації О.А. Щербина

РЕФЕРАТ

Актуальність теми. З розвитком комп'ютерних технологій зростають потреби користувача у масштабованості та гнучкості його комп'ютерної мережі. Технологія віртуалізації мережевих функцій допомагає зробити мережу більш еластичною, простою в керуванні за рахунок перенесення інтерфейсу керування пристроїв на один спільний, централізований інтерфейс, менш затратною, оскільки додавання нового програмного модуля в мережу виконується набагато легше і швидше ніж додавання апаратної складової. В зв'язку з таким розвитком віртуалізації мережевих функцій актуальною темою постає захист віртуалізованої мережі, всіх її елементів та даних.

Об'єкт дослідження – системи захисту мережі з технологіями SDN (Програмно-конфігуровані мережі) та NFV (Віртуалізація мережевих функцій).

Предмет дослідження – дослідження ефективності захисту елементів мережі, периметра мережі загалом, а також даних, що транспортуються в цій мережі. проведення аналізу сигнатур розповсюджених атак, моделювання тестової мережі, створення імітаційних атак на мережу та порівняльний аналіз захищеності системи.

Мета роботи: є детальний огляд програмно-конфігуровних мереж з використанням в них NFV технології, аналіз вразливих місць таких мереж, аналіз ефективності різних елементів захисту, а також розглянуто актуальність використання автоматичного набору імітаційних атак для кожної віртуальної функції задля забезпечення швидкого виявлення та усунення помилок програмного забезпечення.

Для досягнення мети визначено наступні задачі, які вирішуються в роботі:

1. Провести систематизацію та аналіз видів вразливостей віртуалізованої мережі.

2. Розглянути основні методи захисту мережі від розповсюджених атак.

3. На основі проведеного аналізу розробити комплексну систему захисту ресурсів мережі, в якій використовується технології SDN та NFV.

4. Змоделювати роботу тестової мережі та провести імітації атак на її елементи, після чого на основі отриманих результатів сформулювати рекомендації, які допоможуть уникнути типових помилок при створенні нової комплексної системи захисту .

Об'єкт дослідження – системи захисту мережі з технологіями SDN (Програмно-конфігуровані мережі) та NFV (Віртуалізація мережевих функцій).

Предмет дослідження – дослідження ефективності захисту елементів мережі, периметра мережі загалом, а також даних, що транспортуються в цій мережі. проведення аналізу сигнатур розповсюджених атак, моделювання тестової мережі, створення імітаційних атак на мережу та порівняльний аналіз захищеності системи.

Методи досліджень – використовувався метод експериментального моделювання, для моделювання роботи тестової мережі. Проведено аналіз методів захисту від розповсюджених атак в мережах з різними моделями реалізації віртуалізованних функцій, розробка набору автоматичних тестів, які будуть імітувати атаку на змодельовану мережу і аналізувати реакцію системи захисту.

Наукова новизна одержаних результатів в магістерській дисертації полягає в наступному:

1. Проаналізувавши технології програмно-конфігуровних мереж та віртуалізованих мережевих функцій виявлено, що на сьогоднішній день залишаються не вирішені питання надійної та коректної роботи мереж з даними технологіями, оскільки разом з впровадженням цих технологій з'являються нові вразливості мережі.

2. Розроблено комплексний спосіб та систему захисту ресурсів комп'ютерної мережі, яка відрізняється від наявних тим, що в мережі встановлено централізований контролер, який проводить моніторинг мережі, а також елементом захисту розробленої системи є набір імітаційних атак в поєднані з загальним інтерфейсом контролювання мережевих функцій, за допомогою тестових атак, система та мережевий адміністратор навчаться розрізняти сигнатури схожих атак.

3. Змодельовано роботу тестової мережі з використанням віртуальних функцій, проведено на неї імітації атак та показано як система реагує на дані атаки і як проводиться логування під час атаки. В подальшому дасть змогу швидко і гнучко реагувати на вразливості кожної функції, а також підвищить загальний рівень захисту системи.

Практична цінність одержаних результатів можуть бути використані для вибору найкращого набору методів захисту комп'ютерної мережі.

Апробація роботи. Основні положення і результати роботи представлені та обговорені на:

V Міжнародна науково-технічна Internet-конференція «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами» НУХТ (Київ 22-23 листопада).

XI науковій конференції магістрантів та аспірантів «Прикладна математика та комп'ютинг – ПМК'2018» (Київ, 14 – 16 листопада 2018 року).

Публікації. За результатами магістерської дисертації опубліковано 2 наукові праці, з них 2 тези доповідей.

Структура та обсяг роботи. Магістерська дисертація складається з вступу, трьох розділів, висновків та 4 додатків.

У *вступі* подано загальну характеристику роботи, зроблено оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку досліджень,

сформульовано мету і задачі досліджень, показано наукову новизну отриманих результатів і практичну цінність роботи.

У *першому розділі* наведено аналітичний огляд літературних джерел технології віртуалізованих обчислень, їх загальні характеристики, особливості, структуру та архітектуру.

У *другому розділі* проаналізовані три моделі реалізації віртального хмарного обчислення за допомогою технології NFV, розглянуті їх основні принципи роботи і вразливості при роботі з ними. Розглянуто основні способи захисту мережі при використані цих моделей.

У *третьому розділі* розроблено власну комплексну систему захисту ресурсів комп'ютерної мережі встановлено централізований контролер, який проводить моніторинг мережі, також елементом захисту розробленої системи є набір імітаційних атак в поєднані з загальним інтерфейсом контролювання мережевих функцій, за допомогою тестових атак, система та мережевий адміністратор навчаться розрізняти сигнатури схожих атак.

У *висновках* представлені результати проведеної роботи.

Робота представлена на 78 аркушах, містить посилання на список використаних літературних джерел.

КЛЮЧОВІ СЛОВА: програмно-конфігуровна мережа, гіпервізор, віртуалізована функція, керована атака, хмарне обчислення, віртуалізована інфраструктура, менеджер віртуальних функцій, Nmap, DoS, перевонення буферу, віртуальна машина.

ABSTRACT

Actuality of theme. With the development of computer technology, the user's needs are increasing in the scalability and flexibility of their computer network. Network virtualization technology features helps us to make the network more flexible, easier to manage it via transferring control interface of devices on a common, centralized interface, less costly in adding a new software module implemented in the network much easier and faster than adding hardware component. Whereas with this development of network function virtualization, the actual theme is the protection of the virtualized network, all its elements and data.

Purpose : there is a detailed overview of software-defined networks using NFV technology in them, analysis of vulnerable places such networks, analysis of the effectiveness of various security features, and discussed the relevance of using automatic set of simulation attacks for each virtual function to ensure quick detection and correction of software errors.

To achieve the goal, have been created the following tasks , which are solved in the work:

1. Conduct systematization and analysis of the types of vulnerabilities of the virtualized network.
2. Consider the basic methods of protecting the network against widespread attacks .
3. Develop a comprehensive system for protecting network resources using SDN and NFV technologies based on the analysis .
4. Create a test network model and make attacks on its imitation elements, then based on the results, formulate recommendations that will help avoid common mistakes when creating a new comprehensive security system .

Object of research – network protection systems with software-defined nework technologie and NFV (Virtualization of Network Functions) .

The subject of the study is effectiveness of the protection elements in the network, defense of network perimeter, layer of transporting data on this network . Conducting analysis of signatures of common attacks, simulation of the test network , creating simulation attacks on the network and comparative analysis of the security of the system.

Research methods - the method of experimental simulation used for simulate work of test network. Made analysis of protection methods against common attacks on networks with different models of implementing virtualization's function and, developed a set of automated tests that will simulate an attack on simulated network and analyze the reaction of the system of protection.

Scientific novelty the results obtained in master s dissertation is as follows:

1. After analyzing the technologies of software-defined networks and virtualized network functions, it has been found that today we have unresolved question of the reliable and correct operation of networks with these technologies, via introduction of these technologies there are new vulnerabilities of the network.
2. The complex method and system of protection of computer network resources have created, which differs from the simple ones that created system exists in the network centralized controller that monitors the network, also set of simulation attacks is element of protection developed system. With the help of this simulated atacks, the system and the network administrator will learn to distinguish signatures of similar attacks.
3. Work of the test network with the use of virtual functions has been simulated, immitations of attacks have been carried out on it, and it shows how the system responds to attack data and how logging is done during an attack. In the future, it will allow you to quickly and flexibly respond to the vulnerabilities of each function, as well as increase the overall level of protection of the system.

The practical value of the results obtained can be used to select the best set of methods for protecting the computer network.

Approbation. The main provisions and results of work are presented and discussed at:

V International scientific and technical Internet-conference " Modern methods, information, software and technical support of control systems organizational and technical and technological complexes "NUFT (Kyiv 22 -23 November).

XI and scientific conferences undergraduates and graduate students "Applied mathematics and computing - PMK'201 8" (Kyiv, 14 - 16 November 2018).

Publications According to the results of the master's dissertation, 2 scientific papers, 2 of them abstracts, are published.

Structure and scope of work. The master's dissertation consists of an introduction, three sections, conclusions and 4 appendices .

The *introduction* gives a general description of the work, assesses the current state of the problem, substantiates the relevance of the research direction, formulates the purpose and objectives of the research, shows the scientific novelty of the results obtained and the practical value of the work.

The *first section* provides an analytical review of literary sources of technology of virtualized computing, their general characteristics, features, structure and architecture.

The *second section* analyzes three models of real-time virtual cloud computing using NFV technology, examines their basic principles of work and vulnerability when works with them. The main ways of protecting thr network when using these models are considered.

The *third section* developed own combined system of protection of computer network resources with centralized controller that monitors the network. Also system has set of simulated attacks which is also as part of defense in conjunction with the general interface for controlling network functions, with the help of simulated attacks, the system and the network administrator will learn to distinguish between signatures of similar attacks.

The *conclusions* are the results of the work.

The work is presented on 78 pages, contains a link to the list of used literary sources.

KEYWORDS: software-defined network, hypervisor, virtualized function, managed attack, cloud computing, virtualized infrastructure, virtual functions manager, Nmap, DoS, buffer overflow, virtual machine.

Список умовних скорочень

ПЗ	– Центр Обробки Даних
ЦОД	– Центр Обробки Даних
AES	– Advanced Encryption Standard
API	– Application Programming Interface
CPE	– Customer Premises Equipment
DCI	– Data Cener Interconnect
EMS	– Element Management System
ETSI	– European Telecommunications Standards Institute
IP	– Internet Protocol
IPsec	– IP security protocol
MAC	– Media Access Control
MANO	– Management and Orchestration
NAT	– Network Address Translation
NFV	– Network Function Virtualization
NFVI	– Network Function Virtualization Infrastructure
NFVIaaS	– Network Function Virtualization Infrastructure as a service
NFVO	– Network Function Virtualization Orchestrator
OSS/BSS	– Operation/Business Support System
PNF	– Phisical Network Function
RSA	– Rivest Shamir Adleman (encryption alghoritm)
SDN	– Software Defined Network
TCP	– Transport Control Protocol
TLS	– Transport Layer Security
VIM	– Virtualization Infrastructure Manager
VLAN	– Virtual Local Area Network
VM	– Virtual Machine
VNF	– Virtual Network Function
VNFaaS	– Virtual Network Function as a service
VNFC	– Virtual Network Function Component
VNFD	– Virtual Network Function Descriptor
VNFM	– Virtual Network Function Manager
VNPaaS	– Virtual Network Platform as a service

vPE	– virtual Provider Edge
VPN	– Virtual Private Network
WPA	– Wi-fi Protected Access
XML	– Extensible Markup Language
UDP	– User datagram Protocol
QoS	– Quality of Service

ВСТУП

Комп’ютерні технології розвиваються стрімкими темпами. Більшість програмного забезпечення (ПЗ) мережевих пристроїв прив’язане до апаратних рішень конкретних виробників. Це призводить до того, що компанії, які користуються мережевими пристроями одного виробника, вимушені користуватись і програмним забезпеченням цього виробника, яке підходить до конкретних моделей апаратних рішень. Технологія програмно-конфігуровних мереж SDN (software defined network) у тандемі з технологією віртуалізації мережевих функцій NFV (network function virtualization) забезпечує набагато більшу гнучкість топології мережі, і це може значно послабити або повністю ліквідувати залежність програмного забезпечення від фізичних пристроїв конкретних постачальників. Головна ідея програмно-конфігуровних мереж полягає в відділенні функцій передачі трафіку від функцій управління пристроєм, у окремий пристрій — контролер SDN, а NFV у свою чергу — це технологія, в якій мережеві функції виконуються програмними модулями, які працюють на віртуальних машинах.

Розділ 1. SDN мережі з використанням NFV технології

1.1. SDN технологія. Структура SDN мережі

Програмно-конфігуровна мережа (SDN від англ. Software-defined Networking) — мережа передачі даних, в якій рівень управління мережею відділений від пристроїв передачі даних і реалізується програмно [1]. Ця технологія змінює загальний підхід до побудови та керування комп'ютерною мережею, шляхом абстрагування функцій мережевих пристроїв у окремий централізований пристрій – контролер SDN. Якщо умовно представити мережевий пристрій, то його можна розділити на три логічні рівні, це проілюстровано на рисунку 1.1.

Рисунок 1.1 - Умовне розділення мережевого пристрою на три рівні

Рівень адміністрування пристрою відповідає за управління пристроєм загалом, цей рівень представлений вбудованим веб сервером, або спеціально розробленим додатком до цього пристрою, який дозволяє зручно керувати пристроєм.

Рівень керування трафіком – рівень, який включає в себе різні

алгоритми автоматичної реакції пристрою на зміну напрямку або кількості трафіку, що проходить через цей пристрій. На цьому рівні також відбувається передача пакетів службового трафіку, наприклад налаштування мережі або окремої програми

Рівень передачі трафіку, цей рівень відповідає за передачу даних з пристрою, а також за прийом даних на пристрій, а також відповідає за передачу пакетів даних.

Основна ідея програмно-конфігуровних мереж полягає у тому, що рівні адміністрування пристрою та керування трафіком відділяються від рівня передачі трафіку і розміщуються у контролері SDN мережі, а в самого мережевого пристрою залишається лише функція передачі даних, тобто в нього залишається лише таблиця переадресації даних згідно якої він працює.

В стандарті RFC 7426 була описана основна архітектура програмно-конфігуровних мереж. Згідно цього стандарту SDN мережу можна розділити на декілька абстрактних рівнів [3], як це зображено на рисунку 1.2.

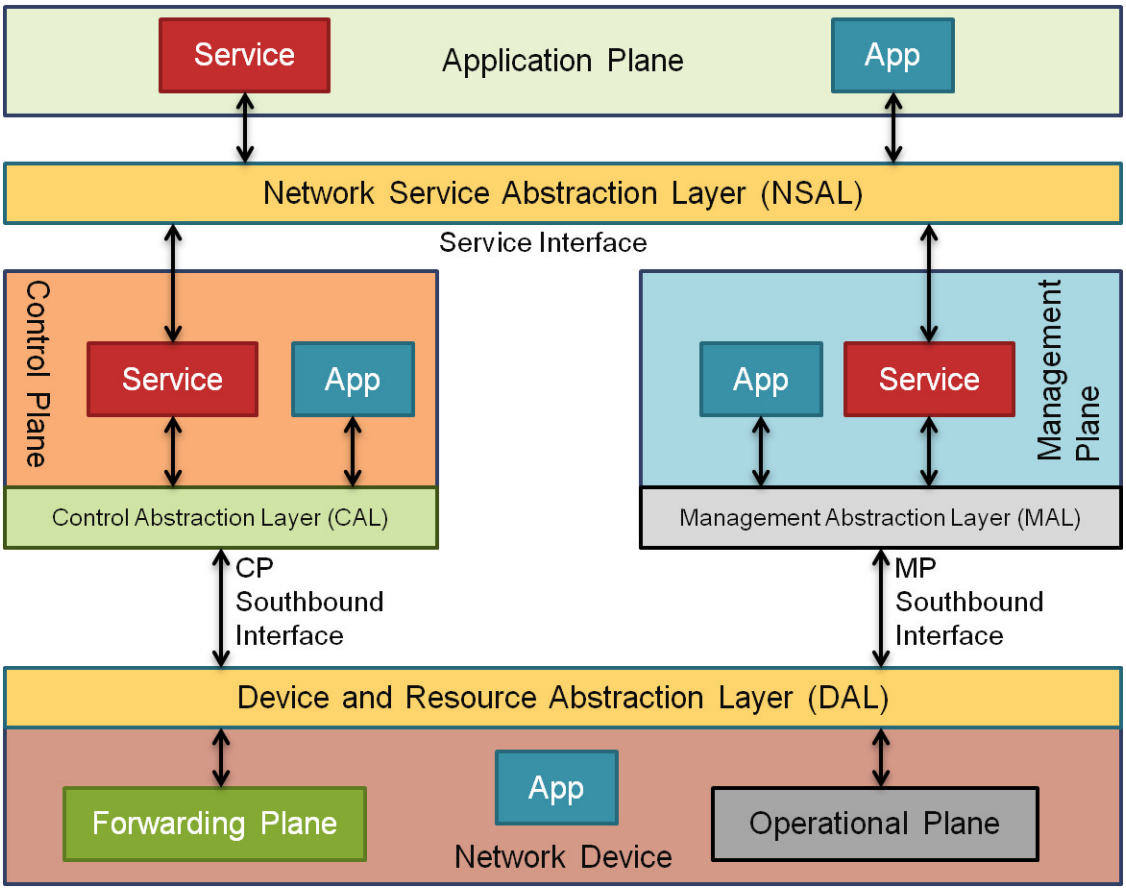

Рисунок 1.2 – Архітектура SDN мережі, запропонована в стандарті RFC 7426

Визначена архітектура забезпечує абстрактний перегляд різних рівнів, що дозволяє розібрати досконало кожний рівень не вдаючись у деталі роботи інших рівнів. Наприклад, для багатьох реалізацій SDN мереж було прийнято, що рівень контролювання (control plane) виступає у ролі служби для рівня управління [3]. На один рівень виноситься набір функції і ресурсів, які відносяться до одного функціоналу, наприклад, функції та ресурси рівня управління трафіком.

За стандартом RFC 7426 виділяється п’ять рівнів SDN мережі

- Транспортний рівень (Forwarding plane, data plane) – відповідає за пересилання, скидання та заміну пакетів на основі правил та інструкцій, які приходять з рівня контролювання. Прикладом програми, яка працює на транспортному рівні, може слугувати класифікатор або лічильник пакетів.
- Операційний рівень (Operational plane) – відповідає за коректну роботу мережевого пристрою, перевіряє стан цього пристрою, вмикає або вимикає його, перевіряє кількість відкритих портів, статус кожного відкритого порту.
- Рівень контролю (Control plane) – цей рівень відповідає за створення та доставлення інструкцій для транспортного рівня. Інструкції передачі пакету з одного пристрою на інший, тобто основною роботою цього рівню є коректне заповнення таблиці перенаправлення пакетів для транспортного рівня.
- Рівень управління (Management plane) – відповідає за моніторинг, налаштування та обслуговування мережевих пристроїв. Також рівень управління може бути використаний, як інструмент для налаштування транспортного рівня.
- Прикладний рівень (Application plane) – рівень на якому розміщуються програми, додатки та сервіси, які визначають суть мережі, тобто для чого вона потрібна, програми і додатки, які безпосередньо підтримують роботу транспортного рівня (наприклад, процеси маршрутизації на

транспортному рівні [3]) не відносяться до прикладного рівню.

Всі описані вище рівні пов’язані між собою через інтерфейси. Інтерфейс може мати кілька форм в залежності від того, чи знаходяться пов’язані інтерфейси на одному мережевому пристрої, чи на різних. Якщо пов’язані ріні на різних пристроях, то інтерфейс виступає в ролі протоколу, який зв’язує ці рівні. Якщо ж пов’язані рівні знаходяться на одному пристрої, то інтерфейс може бути реалізований за допомогою: відкритого власного протоколу, відкритого фірмового інтерфейсу програмного API або за допомогою системних викликів ядра операційної системи.

Важливо розуміти, що рівень контролю і рівень управління відрізняються один від одного, їх особливі ознаки наведені в таблиці 1.1.

Таблиця 1.1 – Особливі ознаки рівня управління та рівню контролю

Ознака	Рівень контролю (control plane)	Рівень управління (Management plane)
Часовий графік	реагує швидше за рівень управління, миттєво відповідаючи на запит	реагує повільніше, оскільки на цьому рівні час не є критичним показником
Стабільність	швидка і часта зміна станів пристрою на рівні контролю	стан пристроїв може залишатися статичним протягом тривалого періоду часу
Розташування	рівень розподілений по всьому пристрою	Рівень зосереджується в одному місці і виноситься за рамки пристрою

Контролювання і налаштування всіх елементів мережі відбувається з SDN контролера мережі, який взаємодіє з усіма елементами мережі за допомогою стандартного для програмно-конфігуровних мереж протоколу OpenFlow. Згідно протоколу OpenFlow контролер використовується для управління таблицями потоків пристроїв, на підставі цих таблиць приймається рішення про передачу прийнятого пакета на конкретний порт

цього пристрою. Таким чином, в мережі формуються прямі з'єднання з мінімальними затримками передачі даних з необхідними параметрами. Ключовим елементом комутатора, що підтримує цей протокол, є таблиця потоків (Flow Table), приклад якої зображено на рисунку 1.3.

MAC src	MAC dst	IP Src	IP Dst	TCP dport	...	Action	Count
•	10:20:..	•	•	•	•	port 1	250
•	•	•	5.6.7.8	•	•	port 2	300
•	•	•	•	25	•	drop	892
•	•	•	192.	•	•	local	120
•	•	•	•	•	•	controller	11

Рисунок 1.3 – Приклад таблиці потоків комутатора

Група стовпчиків в лівій частині таблиці формує поля відповідності, де вказані характеристики потоку: це можуть бути різні параметри, включаючи MAC- і IP-адреси відправника та адресата, ідентифікатор VLAN, номери порту TCP і UDP, а також інша інформація. Ці дані за допомогою протоколу OpenFlow записуються у таблицю комутатора контролера, він же визначає пріоритет різних потоків: чим вище пріоритет, тим вище відповідний запис у потоці таблиць.

Вхідні пакети перевіряються на сумісність з вказаними в таблиці параметрами. Якщо відповідність виявлено, до пакетів застосовується дія, вказана в наступному стовпчику таблиці. Типовою дією є пересилання пакета на один або декілька вихідних портів. Крім того, комутатор може змінити вміст службових полів пакета, скинути його, направити для аналізу на контролер. У випадку, якщо збіг не знайдено, пакет знімається або направляється до контролера, який визначає, як слід обробляти дані потоку, після чого додасть відповідний запис у таблицю [2].

1.2 NFV технологія, використання VNF у SDN мережах

Віртуалізація мережевих функцій (Network Function Virtualization) –

відносно нова технологія, яка використовується у комп'ютерних мережах. Суть цієї технології полягає в тому, що замість того щоб залишати функції мережевих пристроїв таких як роутери, світчі, маршрутизатори, точки доступу та інші, на самих пристроях, ці функції реалізуються як окремі програмні модулі. Зазвичай ці програмні модулі встановлюються на віртуальні машини, які в свою чергу розміщуються на спеціальних серверах. Якщо розглянути цю технологію у зв'язці с технологією програмно-конфігуровних мереж, то ці спеціальні сервери і є контролерами SDN, приклад такої топології зображено на рисунку 1.4. Технологія віртуалізації мережевих функції може працювати і без зв'язки з SDN, але в такому випадку не досягаються найкращі показники роботи в мережі, адже технології NFV та SDN взаємодоповнюють один одного.

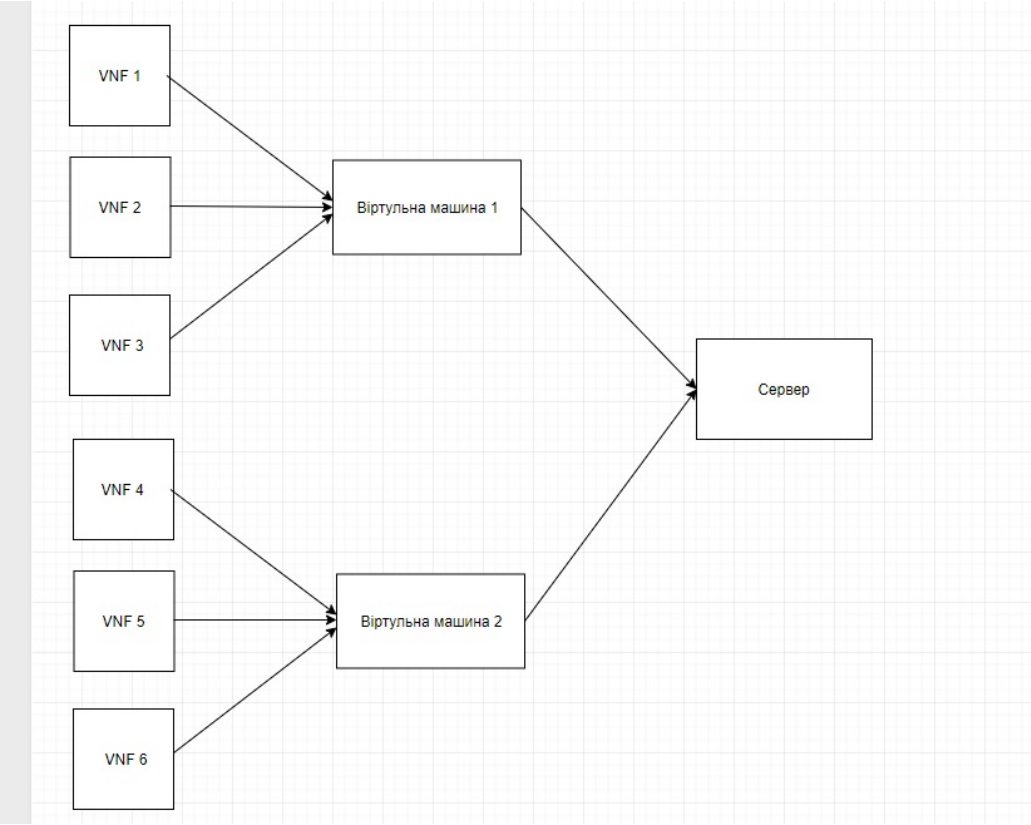

Рисунок 1.4 – Приклад розміщення VNF на віртуальних машинах

На рисунку 1.4 зображено узагальнений приклад як віртуалізовані функції можуть бути розміщенні у мережі. VNF (Virtual network function) під номерами «1», «2» та «3» запущені на першій віртуальній машині, VNF під

номерами «4», «5» та «6» запущені на другій віртуальній машині, а обидві віртуальні машини розміщені на сервері.

Архітектуру взаємодії технологій SDN та NFV зображено на рисунку 1.5. Така архітектура майже повністю реалізується на програмному рівні, за винятком двох елементів OSS/BSS (Operations Support System and Business Support System) та PNF (Physical Network Functions) ці елементи успадковані від традиційної інфраструктури оператора зв'язку. SDN контролер у такому представленні архітектури розділяється на чотири менші контролери, кожний з яких відповідає за різні функції. SDN-контролер дата центру – цей контролер відповідає за створення логічної мережі на центрі обробки даних, де знаходиться інфраструктура NFV (NFVI). SDN контролер WAN створює логічні підключення (з'єднання) між кількома центрами обробки даних, в яких знаходиться фрагментована інфраструктура NFV. Такі підключення між дата центрами називають DCI – Data Center Interconnect [5].

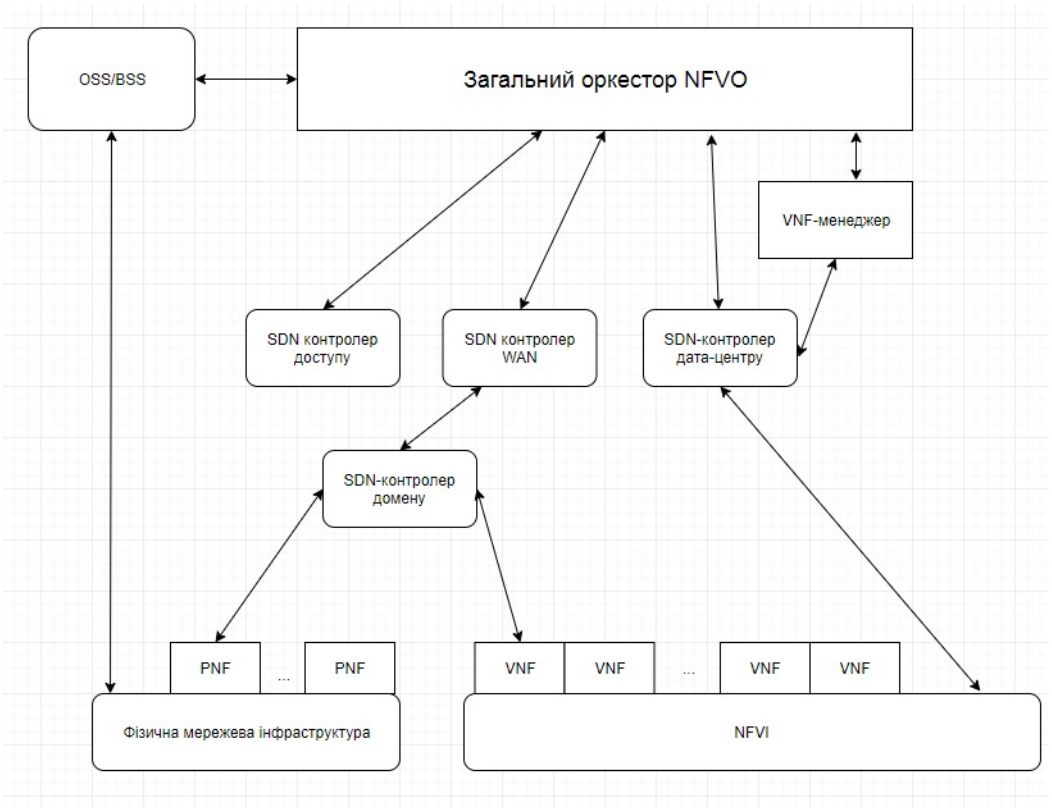

Рисунок 1.5 – Структура взаємодії технологій SDN та NFV

Контролер доступу SDN відповідає за керування доменами доступу мережі оператора. SDN-контролер домену потрібен в тому випадку, коли

виробник фізичного обладнання, та програмного забезпечення на це обладнання, має свої специфічні налаштування для додатків в домені, тоді контролер домену буде «знати» як обробляти додатки в цьому домені. Інші елементи схеми відносяться до NFV технології і будуть детально розглянуті у наступному підрозділі.

Особливість даної архітектури полягає в тому, що вона визнає існування різних доменів мережі, в кожній з яких є свій контролер [5]. Зв’язок і управління між доменами проходить наступним чином:

Якщо домени відносно незалежні, то зв’язок відбувається через загальний оркестратор EEO (End-to-End Orchestration). Наприклад, віртуальному пристрою користувача необхідно скористатися послугою IP-VPN в мережі WAN. Послуга IP-VPN підключається до вже існуючого ланцюга віртуальних функцій (послуг) в центрі обробки даних, далі відбувається налаштування зв’язків з існуючими функціями, трафік підключеної IP-VPN пропускають через мережевий екран, а також через NAT (які в свою чергу також реалізовані у виді віртуальних функцій). Або послуги в SDN-WAN контролері і в дата-центрі можуть бути незалежними один від одного, в такому випадку оркестратор мережевих ресурсів, який пов'язує ці послуги, повинен ідентифікувати граничну точку передачі послуги (наприклад граничний маршрутизатор дата-центру) і встановити зв'язок між їх ідентифікаторами.

1.3 NFV фреймворк запропонований в ETSI

Європейський інститут стандартизації телекомунікацій розробив еталонний високофункціональний фреймворк для роботи з NFV мережою [7], який зображено на рисунку 1.6. Згідно цього фреймворку NFV мережа поділяється на наступні елементи:

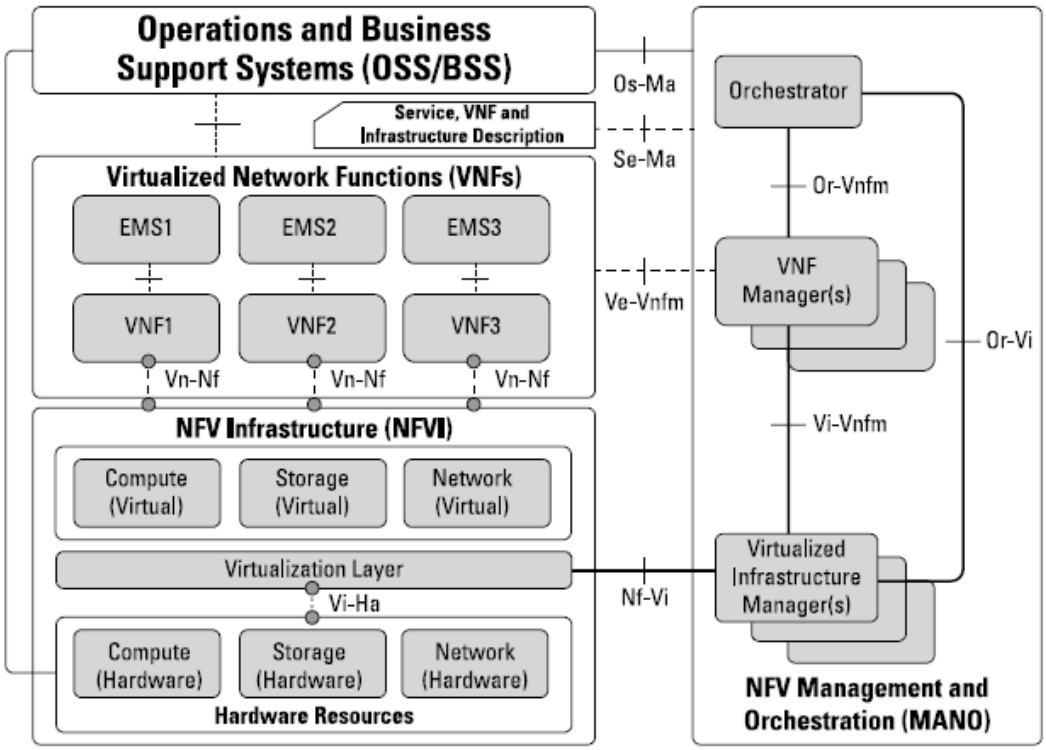

Рисунок 1.6 – Архітектура фреймворку NFV.

1.3.1 VNF

Віртуальні мережеві функції VNF (Virtual Network Function) – основа архітектури NFV. Це і є віртуалізовані функції мережевих елементів. Наприклад, це може бути маршрутизатор (Router VNF), або базова станція (BS VNF) та ін. VNF може представляти також одну з підфункцій мережевого елемента, наприклад, функцію пересилання пакетів (forwarding) в маршрутизаторі. Тоді кілька VNF будуть відповідати одному фізичному мережевому елементу.

Створення і операційна поведінка кожної VNF реєструєтся в дескрипторі VNFD (Virtualized Network Function Descriptor). Кожний VNFD має однозначну відповідність пакету VNF і повністю описує атрибути та вимоги, необхідні для розгортання VNF. Ресурси NFVI призначаються для VNF на основі вимог отриманих з VNFD.

1.3.2 Система адміністрування елементів EMS

Система адміністрування елементів EMS (Element Management System) керує роботою VNF. Вона відповідає за завдання параметрів, тобто адміністрування (Management) операцій VNF. Наприклад, це може бути управління при відмовах (fault management), управління продуктивністю (performance management), і інші функції системи управління мережею. Управління VNF від EMS здійснюється через закриті (proprietary) інтерфейси, тому «референсна точка» між ними ніяк не позначена. Одній VNF може відповідати одна EMS, або одна EMS може керувати кількома VNF. Крім того, сама EMS також може являти собою VNF, керовану від іншої EMS.

1.3.3 Менеджер VNFM

Менеджер VNFM (VNF Manager) управляє роботою однієї або декількох VNF. Він управляє життєвим циклом функцій VNF, тобто, запускає, обслуговує, і зупиняє роботу VNF; оновлює або модифікує ПЗ на окремих екземплярах VNF. Збирає та обробляє інформацію про параметри NFVI, що відносяться до конкретної VNF, а також збирає інформацію про події та відмови; виконує з’єднання декількох VNF для створення більшої функції або послуги; загальна координація та про події між VIM та традиційною EMS оператора. VNFM може виконувати ті ж функції, що і EMS, але через референсну точку, запропоновану ETSI, в архітектурі NFV. Ця референсна точка називається VeNf-Vnfm.

1.3.4 Інфраструктура NFVI

Інфраструктура віртуалізації мережевих функцій NFVI (Network Function Virtualization Infrastructure) – це середовище (інфраструктура), в якій працюють VNF. Ця інфраструктура включає ресурси фізичного обладнання, віртуальні ресурси і рівень віртуалізації.

Віртуальні ресурси: обчислювальні, мережеві і ресурси зберігання. Це віртуальна частина інфраструктури NFVI. Фізичні ресурси абстрагуються в віртуальні ресурси, які використовуються для роботи функцій VNF.

Площина віртуалізації (Virtualization Layer). Площина віртуалізації відповідає за абстрагування фізичних ресурсів у вигляді віртуальних ресурсів. На цьому рівні знаходиться спеціальна програмна платформа «гипервізор» (елемент керування віртуальними функціями), яка виконує відділення програмних засобів від апаратних, тобто дає можливість запускати програми незалежно від обраного обладнання. Наприклад, операційна система може працювати на будь-якому фізичному сервері, який призначається для цієї мети в даний момент.

Апаратні ресурси: обчислювальні, мережеві і ресурси зберігання. Це фізична частина інфраструктури NFVI. Віртуальні ресурси NFV запускаються на фізичних ресурсах NFVI. Це може бути будь-який стандартний комутатор, або фізичний сервер, або пристрій зберігання та інші IT-ресурси.

Припустимо, що площина віртуалізації відсутня. В принципі, VNF можуть працювати безпосередньо на виділених матеріальних ресурсах і бути жорстко до них прив'язані. Однак, при цьому ми не можемо називати їх віртуальними мережевими функціями NVF, в такому випадку, їх слід назвати PNF (Physical Network Functions). Це теж можливо. Однак, в цьому випадку перенесення віртуальної машини з одного фізичного сервера на інший (що часто буває потрібно, наприклад, при зміні місця розташування користувача віртуальної машини VM) потрібно буде виконувати вручну, а це складно, довго і дорого [7].

1.3.5 Менеджер виртуализированной інфраструктури VIM

Менеджер віртуалізованної інфраструктури VIM (Virtualized Infrastructure Manager) – це система адміністрування (management) інфраструктури NFVI. VIM адмініструє використання ресурсів в «одному домені» NFVI. Це можуть бути або фізичні ресурси (сервери, сховища,

мережеві пристрої), або віртуальні ресурси (віртуальні машини), а також програмні ресурси (Гіпервізор). «Один домен» означає, що в архітектурі MANO можуть бути кілька VIM, кожен з яких адмініструє свій домен інфраструктури NFVI. Основні завдання VIM наступні:

- Управління життєвим циклом (lifecycle) віртуальних ресурсів в домені NFVI, тобто, створенням, підтримкою і припиненням роботи віртуальних машин (VM) на фізичних ресурсах в домені NFVI.
- Облік (inventory) віртуальних машин і призначених фізичних ресурсів для їх роботи.
- Управління FCAP (Fault, Configuration, Accounting, Performance) для програм і віртуальних ресурсів.
- Підтримка програмованих інтерфейсів додатків (API) для верхнього рівня, через які, фізичні та віртуальні ресурси надаються іншим системам управління.

1.3.6. Оркестратор NFV

Оркестратор NFV (NFV Orchestrator) генерує, обслуговує і припиняє роботу мережевих сервісів (функцій) VNF, а також ініціює створення закінченої послуги з багатьох VNF.

Оркестратор NFV також відповідає за адміністрування глобальних ресурсів NFVI. Наприклад, він адмініструє ресурси обчислень, зберігання і мережі на кількох менеджерах VIM, в мережі. Оркестратор не взаємодіє безпосередньо з VNF, а тільки через VNFM і VIM.

Є кілька функцій VNF, з яких створена комплексна послуга. Наприклад, це може бути віртуальна базова станція або віртуальний домен опорної мережі EPC. На мережі ці елементи можуть бути представлені обладнанням або від одного, або від різних вендорів. Тоді потрібно створити комплексну (end to end) послугу з використанням декількох VNF. При цьому потрібний оркестратор послуг для комунікації з усіма VNF і створення

комплексної послуги.

1.3.7. Система підтримки операцій і бізнесу OSS/BSS

Система підтримки операцій і бізнесу OSS/BSS (Operation Support System / Business Support System) займається управлінням мережею (network management), управлінням при відмовах (fault management), управлінням конфігурацією (configuration management) і управлінням послугами (service management). BSS відповідає за управління клієнтами (customer management), управління продуктами (product management), управління замовленнями (order management) [7].

В архітектурі NFV наявна BSS/OSS оператора може бути інтегрована з системою NFV MANO за допомогою стандартних інтерфейсів.

1.3.8 Репозиторії

Репозиторії (файли і списки), які зберігають різну інформацію в NFV MANO. Є чотири типи репозиторіїв:

Каталог мережевих послуг (network services, NS Catalog) – це каталог (список) використовуваних мережевих послуг. У ньому містяться шаблони розгортання для мережевих послуг.

Каталог VNF – це репозиторій для всіх використовуваних дескрипторів VNF (VNFD, VNF Descriptor). VNFD - це шаблон розгортання, який описує сценарій розгортання і роботи VNF. Його використовує менеджер VNFM в процесі запуску та управління життєвим циклом функції VNF. Інформація, що міститься в VNFD, також використовується в оркестрі NFVO для управління і оркестрації мережевих послуг і віртуалізованих ресурсів в NFVI.

Список примірників NFV містить всі деталі про приклади мережевих послуг і відповідних примірниках VNF.

Ресурси інфраструктури NFV (NFVI) – це список ресурсів NFVI, використовуваних для формування послуг.

1.3.9 Референсні точки

MANO має безліч референсних точок, наприклад Or-Vi, NF-Vi, Or-Vnfm і ін. Термін «інтерфейс» має на увазі двосторонній обмін сигнальними повідомленнями між об'єктами, які являють собою закінчені або фізичні, або програмні об'єкти. Референсна точка, на відміну від інтерфейсу, визначає взаємодію між функціональними архітектурними блоками, які самі складаються із закінчених програмно-апаратних об'єктів. А оскільки в концепції MANO оперують саме функціональними блоками, а не об'єктами, тому замість слова «інтерфейс» використовується термін «референсна точка» ("reference point") [7].

Vi – Ha (Virtualization Layer – Hardware Resources) – забезпечує зв’язок між рівнем віртуалізації та апаратними ресурсами і також відповідає за створення середовища для роботи віртуальних функцій, збір та виділення необхідних апаратних ресурсів для кожної незалежної функції. Зв’язок Vn – Nf (VNF – NFV Infrastructure) у свою чергу надає середу для виконання функціям VNF. Or – Vnfm (Orchestrator – VNF Manager) – відповідає за обробку ресурснозалежних запитів, таких як: авторизація, валідація, виділення пам’яті VNF менеджером. Відправлення конфігураційної інформації з VNF функції на менеджер функцій. Vi – Vnfm (Virtualised Infrastructure Manager – VNF Manager) – запити виділення ресурсів від VNF менеджера. Or – Vi – (Orchestrator - Virtualised Infrastructure Manager) запити резервації та виділення ресурсів від Оркестра. Nf – Vi (NFVI - Virtualised Infrastructure Manager) – видача віртуальних ресурсів у відповідь на запит про видачу ресурсів. Os – Ma (OSS/BSS – NFV Management and Orchestration) – запити мережевих сервісів, запити на зміну конфігурацій NFVI, перенаправлення користувацьких запитів від NFV.

1.3.10. Дескриптори VNF

У мережевих фізичних функціях взаємозв'язок між внутрішнім програмним компонентами і самим PNF як правило, прихований від оператора, цей взаємозв’язок встановлюється продавцем обладнання. У технології NFV міжкомпонентний зв'язок встановлюється та підтримується в NFVI; тому віртуальні посилання внутрішньої VNF повинні бути визначені як частина VNFD, щоб забезпечити правильне функціонування VNF.

Дескриптор VNF (VNFD) – це шаблон для розгортання віртуальної функції, який описує як повинна працювати VNF з точки зору вимог до розгортання та операційної поведінки. VNFD також містить вимоги до підключення, інтерфейсу до KPI (Key Performance Indicator), які можуть бути використані функціональними блоками NFV-MANO для встановлення відповідних віртуальних посилань у межах NFVI між екземплярами VNFC, або між екземпляром VNF та інтерфейсом кінцевої точки з іншої мережевої функції. VNFD використовуються VNFM-ом для виконання операцій управління життєвим циклом VIM і VNF на інтерфейсах Vi-Vnfm та Ve-Vnfm відповідно.

На додаток до моделей даних файли дескрипторів мають важливе значення для архітектури "End-to-End". Моделі даних описують, як керувати функцією або послугою з точки зору забезпечення ресурсами та моніторингу. Файли дескрипторів описують, як будувати, масштабувати, ремонтувати та оновлювати VNF та/або службу мережі. Файли дескрипторів створюються архітектором по роботі з мережею або дизайнером VNF. Файли дескрипторів лише фіксують інформацію, необхідну на кожному рівні процесу оркестрування. Наприклад, NSD, пов'язаний з SGi-LAN, визначає, що брандмауер повинен бути екземпляром, але не містить деталей про внутрішні частини цього брандмауера. VNFD, який пов'язаний з брандмауером, фіксує внутрішню архітектуру VNFC у VNF.

У VNFM використовується VNFD для виконання функцій керування

життєвим циклом (наприклад, ініціалізація, оновлення, масштабування, ремонтування, завершення) за допомогою шаблону VNFD.

Інтерфейс Vi-Vnfm використовується для визначення необхідної платформи та характеристик мережі, необхідних для кожного VNFC (VNF component). Функції керування життєвим циклом VNFM генерують певні виклики API для VIM за допомогою інформаційних елементів, присутніх у VNFD, так що VIM може призначати оптимізовані та сумісні ресурси для VNFC, а також створювати віртуальні зв'язки між VNFC.

Інтерфейс Ve-Vnfm використовується для виконання додаткових конфігурації та постпроцесних завдань у VNFCs. VNFM має виконувати додаткові скрипти постпроцесного налаштування VNFC у VNFD. Цей інтерфейс використовує вихідні дані від раніше виконуваних функцій управління життєвим циклом, таких як динамічне присвоєння IP-адрес мережевими інтерфейсами VNFC VIM-ом.

Коли VNFC буде активовано та налаштовано, для виконання стандартних етапів життєвого циклу VNF потрібні додаткові процедури. Такі процедури є складними, специфічні для виконання та вимагають конфігурації рівня додатків (application-level) на VNFC компоненті. Ці процедури можуть поширюватися на додаткові елементи за межами VNFC та NFVI. VNFM можуть виконувати ці процедури різними способами, залежно від функцій NFVI та VIM. Прикладом таких процедур може бути:

- Загальний розподіл навантаження на рівні додатків та налаштування балансування навантаження.
- Конфігурація режиму резервування.
- Зміни в мережі (активація фізичної лінії або конфігурація маршруту) поза межами VNFC.

Ці додаткові операції часто виконуються за допомогою скриптів або плагінів з змінними аргументами та вхідними даними на інтерфейсі Ve-Vnfm для застосування необхідних змін у конфігурації.

1.3.11. Дескриптори VNF платформи

Мінімальна репрезентативна підмножина елементів VNFD, які вважаються необхідними для того щоб VNF, були ідентифіковані в NFV. Наступний набір інформаційних елементів є мінімальним набором загальних параметрів платформи, які визначаються на основі кожного VNFC. Ці елементи дозволяють EEO, VNFM та VIM ідентифікувати та призначити необхідні ресурси кожній VNFC, щоб VNFC могла працювати в оптимальному середовищі. Після того, як буде створено екземпляр VNFC, до нього застосовуються додаткові скрипти пост процесорної обробки. Формат дескриптора VNF (VNFD) є специфічним для кожної реалізації та змінюється залежно від вибраних VIM, VNFM та NFVO. Промисловий життєвий цикл функцій керування зазвичай використовує VNFD на базі XML.

1.3.12. L1-L3 дескриптор

Дескриптори VNF (VNFDs) містять опис віртуальних посилань та утворють базову транспортну інфраструктуру, забезпечуючи зв'язок для додатків та інших основних мережевих функцій. Внутрішні віртуальні посилання з'єднують VNFC у VNF. QoS та пропускна здатність визначаються для кожного віртуального каналу зв'язку. Віртуальні посилання включають три типи з'єднання:

E-Line - для простого з’єднання типу точка-точка між VNFC та існуючою мережею.

E-LAN - коли екземпляр VNFC повинен обмінюватися інформацією з усіма іншими екземплярами VNFC в мережі, щоб забезпечити синхронізацію.

E-Tree - коли трафік з одного інтерфейсу повинен бути спрямований у певну гілку іншого, наприклад, у додаток для балансування навантаження. Компоненти віртуальних мереж NFVI повинні забезпечувати зв'язок у внутрішніх VNF між VNFC у межах NFVI для трьох перелічених вище типів з'єднань, а також забезпечити необхідну пропускну спроможність та QoS для

віртуальної лінії зв'язку. Функції з першого і другого рівня неможливо віртуалізувати. Типове зображення L1-L3 дискриптора на рисунку 1.7.

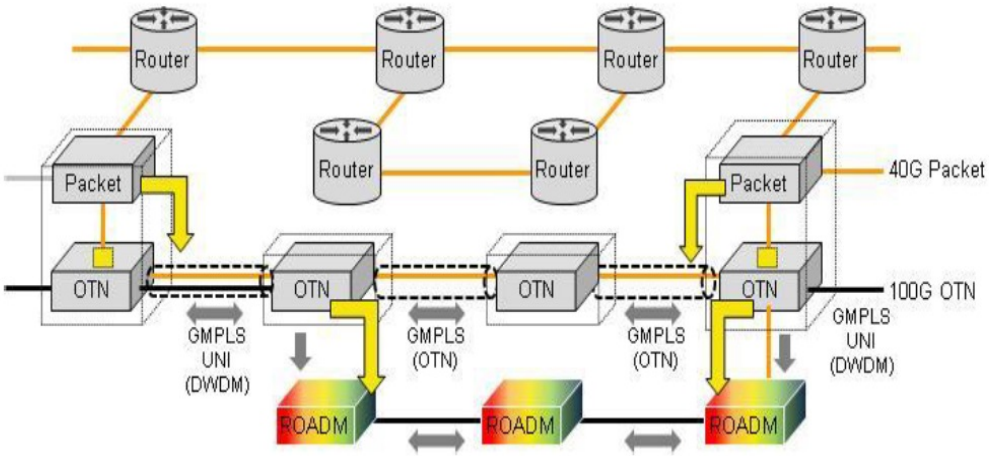

Рисунок 1.7 – Типове зображення L1-L3 дискриптора

Мультиплексор (DWDM) підтримує одночасну передачу кількох оптичних сигналів по одному волокну, забезпечуючи певну довжину хвилі спектру світла для кожного сигналу. Це дозволяє значно збільшити ємність волокна. DWDM система зазвичай налаштовуються як мультиплексор (ROADM), які можна пере налаштувати. Оптична транспортна мережа (OTN) - це стандарт, який розширює поняття SONET/SDH до більш високих швидкостей передачі даних (100 Гбіт/с).

1.3.13 SGi-LAN дескриптор

Нижче наведено два можливі шляхи розгортання архітектури SGi-LAN: Оператор може вирішити будувати свою SGi-LAN мережу з дискретних компонентів, використовуючи Orchestrator для автоматизації координації різних функцій сервісу, кожен з яких постачається як окремий VNF.

Оператор може вибрати розгортання платформи SGi-LAN, яка постачається як єдина композитна VNF (у сенсі VNF, що складається з декількох компонентів VNF або VNFC). Для категорії SGI-LAN VNF додатково до елементів платформи потрібні наступні інформаційні елементи:

- пов'язані PGW (VNFs або PNFs) з певним набором інтерфейсів SGi-LAN для визначення першого кроку служби SGi-LAN;
- асоційовані маршрутизатори, щоб визначити останній хід служби SGi-LAN;
- конфігурація розподілу трафіку;
- правило OpenFlow та пропускна спроможність мережевої структури та кінцевих точок окремих мережевих структур (таких як vswitch та ToR).

Багато таких інформаційних елементів SGi-LAN застосовуються лише тоді, коли послуги SGi-LAN є частиною єдиного VNF, де кожен VNFC це мікро-сервіс порівняно з випадком, коли SGi-LAN реалізується як мережева послуга, використовуючи графіки пересилання між декількома мікросервісами VNF.

Висновки до розділу 1

Розглянуто загальні принципи та архітектура технології програмно-конфігуровних мереж, принципи проектування та побудови мереж з технологією SDN (програмно-конфігуровні мережі). Також проаналізована технологія NFV (віртуалізовані мережеві функції), розглянуто її повний архітектурний шаблон, запропонований інститутом ETSI. Детально проаналізована робота кожного елементу та його зв'язків в мережі.

Проаналізувавши технології програмно-конфігуровних мереж та віртуалізованих мережевих функцій виявлено, що на сьогоднішній день залишаються не вирішені питання надійної та коректної роботи мереж з даними технологіями, оскільки разом з впровадженням цих технологій з'являються нові вразливості в мережі, а також стають більш актуальними деякі з вже існуючих вразливостей, таких як DoS, DDoS, атаки на переповнення буферу.

Розділ 2. Захист SDN мереж та NFV технологій

Загалом SDN та NFV полегшують управління безпекою, але разом з цим також з'являються нові загрози для цих технологій, показані на рисунку 2.1. Гнучкість, яку вони надають мережевій інфраструктурі, дозволяє здійснювати поглиблений моніторинг центрального пристрою, що збирає всю необхідну інформацію (контролер SDN і оркестратор NFV). Однак ці технології мають багато нового програмного забезпечення, нових елементів та протоколів, які всі разом накопичують вразливості для всієї інфраструктури. Крім того, з'являються деякі вразливості, які притаманні виключно для SDN і NFV. Отже SDN та NFV одночасно полегшують адміністрування безпеки за рахунок централізованих головних пристроїв, а з іншого боку з'являються нові вразливості притаманні лише для цих технологій. Хоча SDN і NFV це різні технології, загрози для них можна оцінити за однією системою, а саме, загрози для рівня керування (SDN контролер, NFV оркестратор) та загрози для рівня даних (SDN data plane, NFV virtual functions) [11].

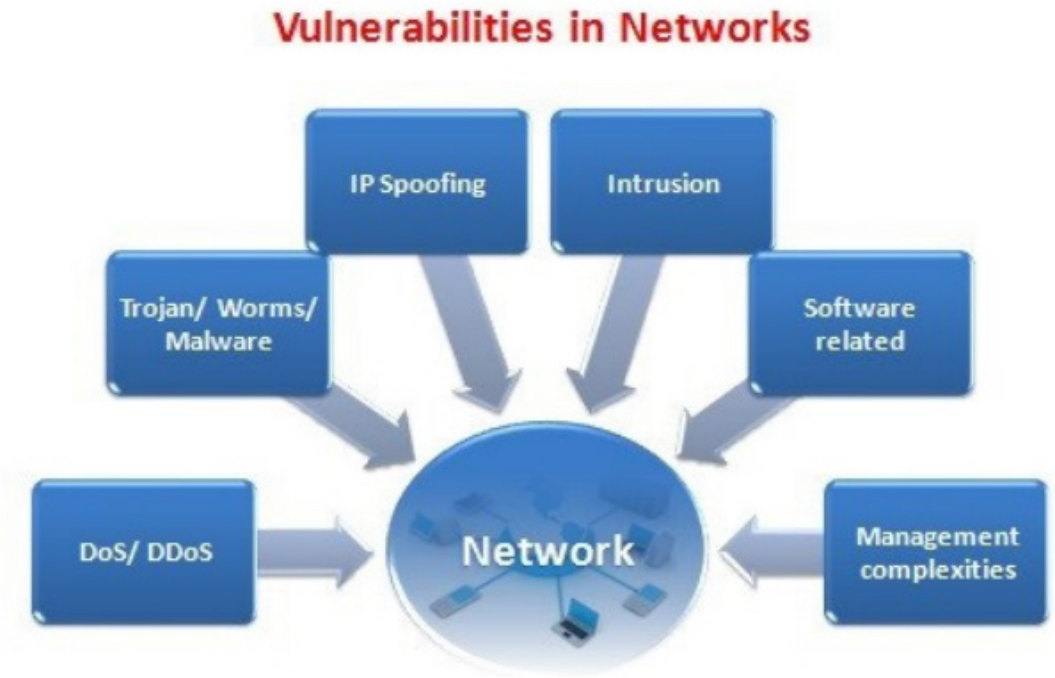

Рисунок 2.1 - Можливі атаки на SDN/NFV мережу.

2.1 Аналіз загальної структури захисту та вразливостей SDN мережі

Згідно технології програмно-конфігуровних мереж, комп'ютерна мережа розділяється на три рівні: транспортний рівень, рівень контролю та рівень додатків. Захист цих рівнів необхідно проводити, як на кожному рівні окремо, так і на всіх точках зв'язку між цими рівнями. В таблиці 2.1 наведено основні типи атак на програмно-конфігуровну мережу, вказані рівні SDN мережі, на які ця атака впливає, а також аспект, який порушує атака.

Таблиця 2.1 - Порівняння атак на SDN мережу

Тип атаки	Рівень, на який націлена атака	Вражений аспект		
		Доступність	Конфіденційність	Цілісність
DDoS	контролю транспортування	x		
DoS	контролю транспортування	x		
Підміна центрального контролера	контролю транспортування додатків	x	x	x
Злоякісні додатки	додатків		x	x
Man-in-the-middle	контролю транспортування та зв'язок між ними		x	x
Blackhole	контролю транспортування та зв'язок між ними	x	x	

2.1.1. Вразливості рівня транспортування

Одною з найпоширеніших атак на програмно-конфігуровні мережі є

атака типу DoS. На транспортному рівні атаки на припинення обслуговування спрямовуються на потоки, перший вид атак використовує таблицю активних потоків даних. Використовуючи дані з цих таблиць, зловмисник починає «флуд» пакетам як на транспортний рівень, так і на рівень контролю. Помилково створені або з зловмисним наміром потоки можуть переповнити таблиці потоків всіх пристроїв в мережі. Згодом це призводить до того, що пам’ять в мережевих пристроях більше не в змозі обробляти потоки та записувати їх в таблиці, що негативно впливає на роботу всієї мережі. Виходячи з цього постає одна з ключових вразливостей на транспортному рівні, нажаль, пристрої не можуть розрізняти, які потоки даних є легітимними, а які з генеровані зловмисником, або ж просто помилкові. Це дозволяє зловмисникам переповнювати буфери мережевих пристроїв неправильними потоками. Другий тип атак на відмову у обслуговуванні пов’язаний з контролером мережі, його буфер також заповнюється запитами на нові потоки, що призводить до припинення його функціонування. Після того як контролер, перестає працювати, мережа ще деякий час може працювати по старим таблицям, але як тільки виникає конфлікт в таблицях і пристрій звернеться до контролера, мережа перестає працювати.

2.1.2 Вразливості рівня контролю

Головна вразливість рівню контролю випливає саме з переваг програмно-конфігуровних мереж, а саме, весь інтелект пристроїв знаходиться в одному централізованому контролері, тому атака на підміну або вивід з ладу цього контролера є дуже небезпечною. Якщо зловмисник отримає контроль над центральним контролером мережі він зможе перенаправляти потоки як йому буде потрібно, зупиняти потрібні пакети, відправляти некоректні пакети на інші пристрої. Також може з компрометувати інший пристрій в мережі так, що через нього буде проходити атака типи Man-in-the-Middle. Іншим сценарієм схожої атаки на

контролер, є внесення в мережу іншого «фальшивого» центрального контролера. Через цей контролер зловмисник також може отримати доступ до інших контролерів мережі, переналаштувати правила безпеки тощо. Тому будь-яка атака націлена на контролер мережі може спричинити руйнівні наслідки. Для того щоб такого сценарію не трапилось, мережевому адміністратору завжди слід перевіряти аутентифікацію центрального контролера перед тим як надавати йому право контролювати мережу.

Не менш важливою проблемою є захищеність зв'язку рівня контролю з рівнем транспортування (зв'язок з південним інтерфейсом). Ця проблема виникає у зв'язку з тим, що згідно протоколу OpenFlow, який функціонує між рівнем контролю та транспортним рівнем, використання протоколу TLS при передачі даних між цими рівнями є не обов'язковим. Тому виникають ризики атак man-in-the-middle та атак типу blackhole.

Атака man-in-the-middle, представляє собою встановлення пристрою, який знаходиться під контролем зловмисника, між рівнем контролю та рівнем передачі. Після чого дані, які ідуть з рівня транспортування на рівень контролю, починають проходити через цей пристрій.

Атака типу blackhole, пристрій зловмисника також встановлюється між пристроєм-жертвою та рівнем контролю, а далі скидує без перенаправлення всі пакети, які пристрій-жертва відправляє на рівень контролю. Це спричиняє знищення коректних зв'язків в мережі і робить сервіс пристроя-жертви недоступним.

2.1.3 Вразливості рівня додатків

В зв'язку з тим, що згідно технології SDN в мережі немає обмежень на підключення різноманітних додатків, можуть виникнути вразливості в роботі мережі через конкретні не правильно налаштовані додатки, які працюють в цій мережі. Досить важко і не практично реалізовувати систему авторизації та аутентифікації для кожного додатку в мережі, тому деякі з додатків можуть бути з компрометовані. Якщо зловмиснику вдасться отримати

контроль над додатком, який в своїй роботі використовує технологію глибокого інспектування пакету, це може спричинити витік даних з мережі.

2.1.4 Захист рівня контролю

Головними і найважливішими компонентами рівню контролю є контролери SDN, тому ці контролери необхідно захистити як від фізичних ушкоджень, так і на програмному рівні. Контролери SDN можуть бути реалізовані в різних конфігураціях, в якості автономних апаратних засобів або віртуалізованих пристроїв, в якості високоефективного кластера з високою доступністю. Для всіх видів реалізації необхідно забезпечити високий рівень надійності контролерів. Перший рівень захисту полягає в тому, що необхідно забезпечити якісний захист сервера, на якому знаходиться реалізований SDN контролер. Далі операційна система контролера повинна бути оновлена і надійно захищена. На рівень контролю заборонено встановлювати сторонні додатки або запускати ці додатки в тому ж середовищі, в якому працюють контролери. Інтерфейси адміністрування контролера повинні бути доступними лише в ізольованій приватній мережі, наприклад, в демілітаризованих зонах. Також необхідно реалізувати надійну систему авторизації та ідентифікації і систему логування всіх подій. Контролери, які реалізовані у виді високо доступних кластерів, потребують додаткового захисту, тому на них розробляють додаткові механізми для ідентифікації дозволених «пірів», бібліотек та сертифікатів. Шаблон повідомлень для кластеризації повинен відповідати всім заходам безпеки, механізму обміну повідомленнями, прикладом може слугувати протокол OpenDaylight, який використовує власну систему авторизації та шифрування повідомлень Infinispan’s Jgroups AUTH and ENCRYPT [9]. Необхідно забезпечити надійний зв'язок між контролерами SDN та іншими двома рівнями SDN структури. Зв’язок з південним інтерфейсом (транспортним рівнем) використовує низку різних протоколів, які мають власні заходи безпеки, OpenFlow, OVSDB. Передача даних має проходити через захищений

канал, наприклад, протоколами TLS, DTLS або IPsec. Якщо розглянути зв'язок з північним інтерфейсом (рівень додатків), то цей зв'язок більш за все піддається атакам типу DoS та підміни пакетів, тому передачу даних виконують за протоколом TLS. У додатків на цьому рівні не повинно бути прав запуску від імені облікового запису адміністратора.

2.1.5 Захист рівня транспортування

Оскільки протокол OpenFlow, що функціонує між рівнем контролю і рівнем транспортування не вимагає обов'язкового використання протоколу TLS, першим кроком для захисту цих рівнів, необхідно все ж таки використовувати протокол шифрування TLS.

Наступною задачею надійної роботи на цьому рівні є захист елементів від скомпрометованих запитів, які надходять з рівня контролю. У тих випадках, коли рівень передачі має можливість конфігурування за допомогою застарілих протоколів маршрутизації, тобто у гібридному середовищі, повинні бути впроваджені заходи безпеки, а також можливість зареєструвати джерело змін, та внести подію в журнал логування.

2.1.6 Захист рівня додатків

Як було зазначено в попередніх розділах, злоякісний додаток може завдати невиправної шкоди для всієї мережі, а також організація захисту на цьому рівні є найскладнішою, оскільки важко контролювати додатки, які запускаються в мережі. Тому необхідно якісно захищати цей рівень, доступними інструментами.

Наприклад, встановлення механізму перевірки нових правил від додатків, які контролюють безпеку (мережеві екрани, проксі сервери, системи виявлення вторгнень та інші.). Спочатку нове правило повинно пройти через всі вже існуючи правила на конкретному пристрої, якщо між новим і існуючим правилом виникає конфлікт – нове правило відміняється. Таким чином мережевий адміністратор повинен додати декілька ключових

правил на пристрій, для того щоб, спроба додати нове компрометуюче правило зі сторони зловмисника, буда не вдалою. Але цей підхід має один суттєвий мінус, мережевий адміністратор може внести такі правила, які потім не дадуть додаткам з захисною функцією коректно додати нові легітимні правила.

Іншим способом захисту рівня додатків може слугувати наступний механізм: створюється невелика віртуальна платформа, яке розміщується між рівнем контролю і рівнем додатків, ця платформа надає середовище для виконання додатків з рівня додатків. Це в свою чергу дає можливість контролювання додатків, що запускаються, за допомогою моніторингу та різних правил, що діють на віртуальній платформі.

2.1.7 SDN контролер безпеки

Вразливі місця та послуги з доданою вартістю, які раніше були притаманні для мобільних мереж тепер виступають як сервіс в хмарному середовищі. У результаті цього безпека SDN мережі повинна розглядатися як безпека у гібридному середовищі. Контролер безпеки дозволяє виконати наступні пункти:

- керування великою кількістю окремих сервісів безпеки;
- оркестрування служб безпеки через віртуалізовані ЦОД;
- не вимагає додаткового навантаження на робочі віртуальні машини;
- надійна інтеграція з віртуалізованими платформами;
- автоматична синхронізація.

Централізована структура контролерів безпеки забезпечує можливість абстрактної інфраструктури безпеки, ін'єкції послуг в робочому процесі на основі політики безпеки.

2.2 Аналіз вразливостей моделей хмарного обчислення з NFV технологією

На жаль використання NFV технологій не позбавлено своїх недоліків зі сторони безпеки, оскільки технологія віртуалізації мережевих функцій знаходиться на стадії розробки і використання одночасно. Тому обов’язково перед тим як реалізовувати цю технологію в мережі, необхідно детально проаналізувати всі аспекти. Тому що загроза коректній роботі кожної VNF функції – це комбінація загроз, які надходять як з фізичної мережі, так і безпека технологій віртуалізації в цілому.

2.2.1 NFV інфраструктура як сервіс (NFVIaaS)

Інфраструктура як сервіс (служба) – це одна з трьох фундаментальних сервісних моделей хмарного обчислення. IaaS (infrastructure as-a-service) сервіс, який забезпечує доступ до обробки, обчислення, та зберігання даних у віртуалізованому середовищі. Користувачі даного сервісу можуть використовувати комп’ютерні ресурси інфраструктури для запуску власних додатків при чому вони не мають доступу до контролювання самої інфраструктури. Приклад NFVIaaS зображено на рисунку 2.2. NFVIaaS надає обчислювальні можливості та можливості підключення до мережі, які можна порівняти з службою IaaS та NaaS (мережа як служба), які в хмарному обчисленні виступають в ролі середовища для виконання, та підтримують динамічні послуги з підключення нових елементів до мережі, відповідно. Постачальники послуг також можуть використовувати власну NFVI хмарну обчислювальну інфраструктуру або інфраструктуру іншого постачальника послуг для виконання власних мережевих функцій (VNF). В інфраструктурі NFVIaaS, обчислювальні вузли розташовуються в центральних офісах або у вбудованому в інше мережевому обладнанні, наприклад, в мобільних пристроях.

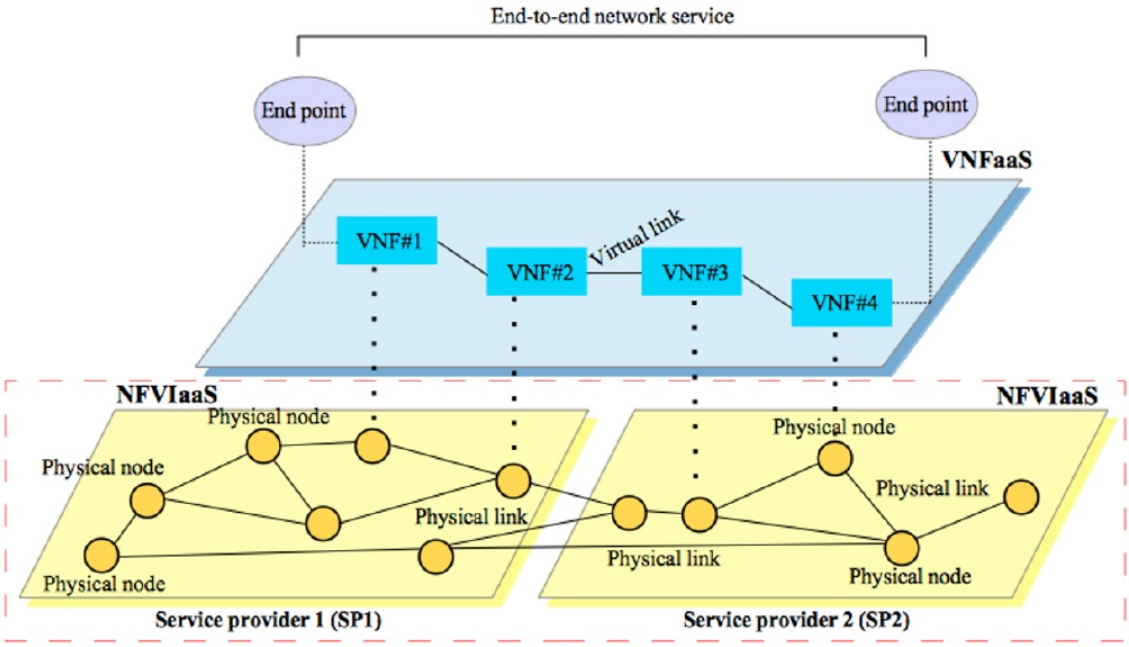

Рисунок 2.2 – Приклад архітектури NFVIaaS

На рисунку 2.2 зображено хмару в якій розташовано NFVIaaS, а також декілька VNF функцій різних провайдерів. Як показано на малюнку, постачальник послуг «2» може запускати власні VNF на хмарній інфраструктурі NFVI іншого постачальника послуг «1», такий підхід дозволяє підвищити гнучкість сервісу та враховуючи досвід користувача зменшити час затримки сигналу. В свою чергу постачальник послуг «1» може вимагати того, щоб лише авторизовані суб'єкти мали доступ до використання їхньої інфраструктури. Також постачальник послуг «2» може запустити свої екземпляри VNF на інфраструктурі NFV постачальника послуг «1» у вигляді послуги «end-to-end», що означає, що всі ресурси як апаратні так і програмні, які необхідні для виконання віртуальних функцій будуть представлені постачальником «2». Одночасно з цим постачальник «2» може запустити тіж самі віртуальні функції на своїй інфраструктурі. Очевидно, що NFVIaaS двох провайдерів є незалежними, і порушення в роботі одної інфраструктурі ніяк не вплине на роботу іншої.

Зазвичай NFVIaaS чутлива до всіх загроз, які притаманні стандартному IT середовищу. Одною з головних проблем для провайдера є те, що додатки,

які виконуються на їх інфраструктурі є «чорними коробками» і постачальник не може ніяк їх перевірити, таким чином, деякі ненадійні додатки користувачів можуть спричинити втрату даних, порушення конфігурації інфраструктури або пропустити зловмисника у периметр мережі. Нижче зазначені найпоширеніші вразливості NFVIaaS:

- втрата даних на компонентах сервісу;
- зловмисник всередині;
- непрозорий моніторинг;
- проблеми віртуалізації платформи;
- вразливості локального хоста;
- вразливості з’єднання з мережею інтернет;
- проблема безпеки розподілених ресурсів;
- припинення або компрометація роботи гіпервізора.

При проектуванні, розробці та підтримці NFVIaaS звертають увагу на низку технологій захисту.

Захист даних при передачі та захист від витоку даних. Дані, які зберігаються в дата центрах, є конфіденціальними, тому вони потребують детального моніторингу. Таким чином необхідно перевіряти чи були доступні захищені дані авторизованим користувачам, спостерігати хто отримує доступ до інформації, з якого місця, які маніпуляції проводилися з даними. Під час передачі дані захищаються за допомогою протоколів TLS та IPSec. Також всі дані, які знаходяться на зберіганнях на серверах повинні бути зашифрованими.

Моніторинг безпеки та виявлення вторгнень. Оскільки площа віртуалізованної інфраструктури досягає дуже великих розмірів, місць для вторгнень зловмисників або проведень атак стає незліченно багато, тому доцільним є використання систем виявлення вторгнень та моніторингу стану мережі.

2.2.2 Віртуальна мережева платформа як сервіс (VNPaaS)

В хмарних обчисленнях модель «платформа як сервіс» (PaaS) визначається як здатність сервіс провайдера запропонувати обчислювальну платформу для користувачів. Таким чином користувачі можуть запускати свої додатки або сервіси, використовуючи запропонований провайдером шаблон сервісу або використовувати певну керуючу функцію. Таким чином провайдер надає набір мережевих інструментів для зручного розвитку, використання, та підтримки додатків користувачів. Зокрема віртуалізовані функції можуть бути частиною платформи для створення віртуальної мережі, тому користувачі можуть використати ці функції для створення своєї власної віртуальної мережі, основаної на їх особистих потребах.

VNPaaS модель схожа з моделлю VNFaaS, відрізняється лише в масштабованості надання сервісу, та обсязі контролю, що надається користувачу. VNPaaS забезпечує більшу масштабованість сервісу віртуальних мереж, ніж окрема віртуалізована функція. VNFaaS забезпечує підприємство можливістю розробити їх власний екземпляр віртуалізованої функції, однак компанія буде обмежена в налаштуваннях набору функцій, можливість налаштувань залишиться у провайдера. Наприклад VNFaaS може виступати в ролі поштового серверу і бути екземпляром окремої VNF з набору, в той час як VNPaaS представляється як хостінг провайдер, який дозволяє користувачу інсталювати поштовий сервер без налаштувань. Після інсталяції поштового серверу підприємство має повні адміністративні права для будь-яких налаштувань на цьому сервері. Крім того, підприємство може використовувати інші екземпляри VNF підключаючи їх до сервера електронної пошти, щоб дозволити розширену конфігурацію (наприклад, захист від спаму, додавання служби DHCP, DNS, проксі-сервери або кешування). Тип служб, що підтримуються у VNPaaS, може коливатися від простого мережевого екрану для одного підприємства до мультимедійної підсистеми IP (IMS) всього бізнесу.

Підводячи підсумок, послуги, надані VNPaaS, дозволяють підприємству розміщувати такі послуги на платформі віртуалізації як мережевий екран, DHCP, DNS, проксі, електронну пошту або послуги зв'язку. Завдяки сценарію VNPaaS підприємства можуть отримати попередньо налаштований сервіс із базовим набором конфігурації для подальших модифікацій на основі власних потреб. Приклад VNPaaS моделі зображено на рисунку 2.3. На рисунку 2.3 зображено приклад поділу мережевих ресурсів відповідно до випадку використання VNPaaS, де хостинг постачальник послуг володіє інфраструктурою та повторно продає розділені ресурси інфраструктури третім особам. Підприємства можуть використовувати автономний екземпляр VNF, який не має жодного зв'язку з іншим екземплярами VNF в мережі постачальника послуг, але це все одно надає зв'язок з корпоративною мережею підприємства. Також можливо використовувати екземпляри VNF, які підключені до інших VNF екземплярів, розміщених у мережі постачальника послуг. Крім того, у мережі постачальника послуг є інтерфейс керування для кожного об'єкта, який використовується для визначення правил політики та подальшого зв'язку між VNF та сервісом провайдерів.

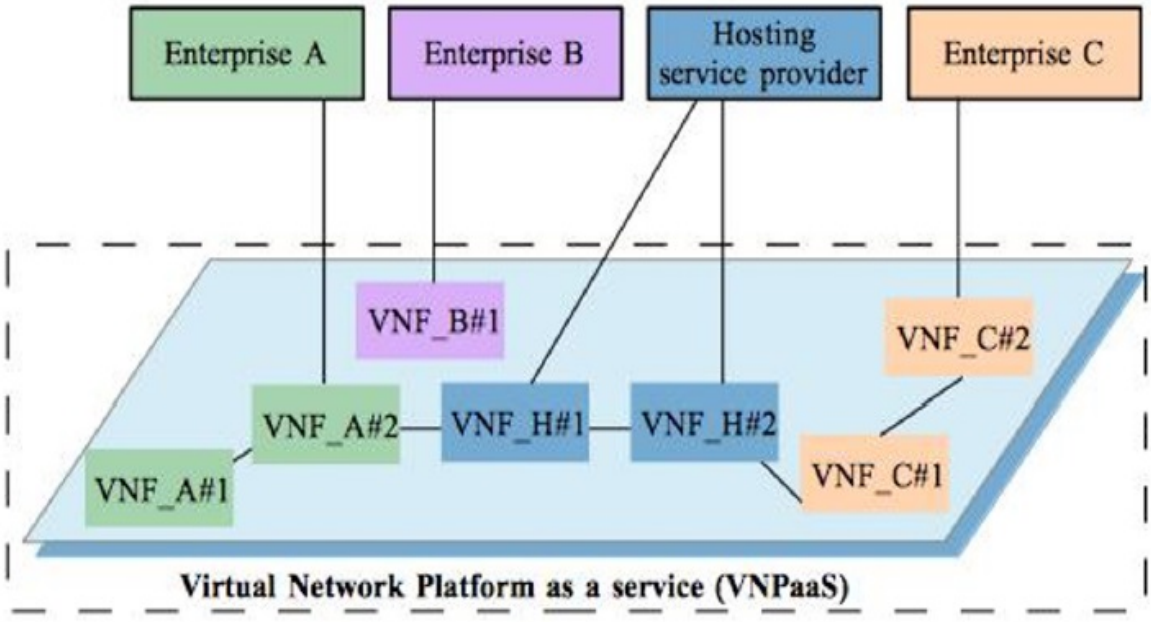

Рисунок 2.3 – Приклад моделі VNPaaS

Віртуалізована платформа як сервіс піддається таким атакам, як витік даних, не прозорий моніторинг, атаки на рівень керування та інші. Але є декілька атак, які притаманні лише для моделі VNPaaS. VNPaaS модель більшу частину ресурсів, а також керування безпекою передає користувачу в той час як сервіс провайдер зберігає лише базові елементи безпеки мережі (мережевий екран системи IPS/IDS), в такому випадку виникає дуже багато векторів загроз. Включаючи такі загрози як дефолтні налаштування додатків, вразливості в протоколі SSL, некоректні права доступу.

Використання стандартних конфігурацій додатків: у VNPaaS моделі, користувачі зазвичай запускають на платформі свої додатки або служби на з дефолтним набором конфігурацій, що потенційно може призвести до атак, які можливі з налаштуванням за замовченням. Наприклад, мережевий екран, який працює з вхідними обліковими даними адміністратора, які встановлені за замовчуванням, такий фаєрвол дуже швидко виводиться з дії. Що більш серйозно, то зловмисник може змінити правила конфігурації і дозволити зловмисному трафіку пройти через фаєрвол. З точки зору безпеки більшість конфігурацій за замовчуванням зазвичай не є надійними і безпечними, тому що налаштування за замовчуванням, як правило, мають базову конфігурацію для аутентифікації, шифрування, авторизації або будь-якого іншого типу контролю безпеки.

Недоліки SSL/TLS: у сценарії VNPaaS SSL/TLS використовується для забезпечення безпечного зв'язку між об'єктами. Незважаючи на використання SSL/TLS у хмарному середовищі, недоліки та вразливості, які виникають в результаті неправильних налаштувань стають привабливою ціллю для зловмисника. Насправді, багато атак, проводяться на теоретичні та практичні недоліки, які вже давно були ідентифіковані. Наприклад, атаки на PKI ключ і протокол зворотного зв'язку «хендшейк». Зокрема зловмисник може змінити повідомлення, надіслане ініціатором підключення, маніпулюючи або повністю замінюючи набір шифру. Крім того алгоритм, що використовується

в SSL/TLS протоколах, може бути передбачений і скомпрометований, оскільки на сьогоднішній день алгоритм спирається на передбачувані значення, наприклад, поточний час, поточний ідентифікатор процесу та ідентифікатор батьківського процесу, які є уразливими для нападів типу брутфорс.

Атака на «боковий» канал. Така атака зазвичай виникає, коли дані виводяться за межі одного апаратного забезпечення, наприклад, на інший сервер, цим може скористатися потенційний зловмисник. Ця атака використовує слабкі місця реалізації криптографічних алгоритмів. Фактично криптографічних алгоритмів вистачає для того щоб не допустити втрату даних, але, дуже часто через вразливості додатків, які працюють з цими алгоритмами, стає можлива атака на алгоритми шифрування. Також в цій атаці зловмисники можуть використовувати сигнали, які виходять з бокових каналів під час нормальної роботи системи, щоб виявити секретну інформацію. Цей тип атаки важко виявити. Наприклад, атака на спільний процесор, який використовується в хмарі PaaS. На першому кроці зловмисники створюють заражений екземпляр VNF функції, для того, щоб запустити цю функцію на іншій платформі, але на тій самій ОС що і цільова жертва. Другим кроком зловмисники витягують конфіденційну інформацію, використовуючи цю точку перерви, ця атака дуже схожа по сценарію на MiTM атаку.

При проектуванні, розробці та підтримці VNPaaS, зважаючи на особливості її роботи, крім звичайних способів захисту, також приділяють увагу на додатковий захист протоколів SSL/TLS на етапі передачі даних.

Усунення конфігурацій за замовчуванням. Опираючись на проведений аналіз роботи моделі VNPaaS було виявлено, що налаштування за замовченням в додатках залишає для зловмисників багато вразливих місць у безпеці всієї системи. Для уникнення проблем пов’язаних з налаштуваннями додатків, персонал провайдера, що надає платформу для виконання VNPaaS, повинен бути висококваліфікованим і володіти знанням і хорошим

розумінням алгоритму роботи моделі VNPaaS. Також провайдеру необхідно пильно провести налаштування всіх правил безпеки на всіх мережевих елементах, а також не залишати на своїх шаблонних додатках налаштувань за замовчуванням.

Безпечне налаштування і використання протоколів SSL/TLS. Так як SSL/TLS відіграє ключову роль у роботі моделі VNPaaS, особливу увагу необхідно приділити їх налаштуванню. Необхідно уникати вразливостей слабкого шифрування та генераторів випадкових чисел. Додатки, які використовують у своїй роботі протоколи SSL/TLS повинні бути якісно налаштовані для уникнення помилок в конфігурації. І на кінець необхідно забезпечити таємну передачу даних, повне дотримання транспортної безпеки та прив'язку до ключів шифрування.

Під таємною передачею даних слід розуміти те, що навіть якщо один з довгострокових ключів шифрування буде перехоплений, то це не приведе до того, що зловмисник зможе отримати весь набір ключів.

Повне дотримання транспортної безпеки, означає те, що кожне з'єднання користувачів повинно проходити по протоколам SSL/TLS, створюючи при цьому нову сесію, і закриваючи її лише тоді, коли сервер сертифікатів обриває сесію.

Прив'язка відкритого ключа. SSL/TLS сервер повинен змусити браузер працювати лише з сертифікатами, які були видані перевіреним центром сертифікації.

2.2.3 Віртуальна мережева функція як сервіс (VNFaaS)

Переміщення мережевих служб з спеціально спрямованих платформ на спеціальне апаратне середовище в хмарному обчисленні забезпечується елементом структури NFVI. Замість того, щоб використовувати інфраструктуру провайдера підприємство може знайти службу NFVI, яка забезпечить надання розширених функцій мережі, включаючи службову та ресурсну еластичність, високу доступність ресурсів, а також високу

мобільність віртуалізованих функцій, що дає змогу міняти їх фізичне місцезнаходження. У цьому випадку підприємство виступає в ролі кінцевого користувача служби, який може управляти та контролювати додатки лише, конфігуруючи їх налаштування, без можливості контролювати та управляти основною інфраструктурою. Замість підприємства, провайдер послуг може розподіляти і керувати ресурсами інфраструктури по запиту підприємства. Створення віртуалізованих мережевих функцій VNF, які доступні для підприємства як сервіс можна порівняти з хмарними обчисленнями (Software as a Service). На сьогоднішній день найбільш поширені моделі, які реалізовані за допомогою (VNFaaS), - це графіки переадресації VNF, віртуальний пакет ядра (vEPC), віртуальне обладнання приміщень клієнта (vCPE), мережа радіодоступу (vRAN), мережа віртуальної доставки контенту (vCDN), віртуальний пристрій (vSTB). Загальна структура поєднання моделей NFVIaaS і VNFaaS зображена на рисунку 2.4.

Зазвичай провайдер надає доступ до маршрутизатора великій кількості користувачів, що користуються у службами постачальника у хмарі, в той час як клієнт-орієнтовне обладнання (CPE) використовується виключно одним користувачем. З використанням моделі VNFaaS, можлива віртуалізація функції CPE (vCPE) у хмарі постачальника послуг та віртуалізацію функції кінцевого роутера провайдера (vPE). Віртуалізація vCPE та vPE – це дві незалежні події і вони можуть виконуватися окремо.

Віртуалізація обладнання користувача (vCPE) більш вигідна ніж віртуалізація кінцевого роутера, як для самого підприємства, так і для провайдера послуг. Як правило, саме постачальник послуг несе відповідальність за розгортання, конфігурування, оновлення та керування операціями vCPE (віртуалізоване клієнт-орієнтоване обладнання) та vPE (віртуалізований кінцевий розтер постачальника послуг) за додатковою угодою про рівень обслуговування (SLA), для надання доступності сервісу через рівень VNFaaS.

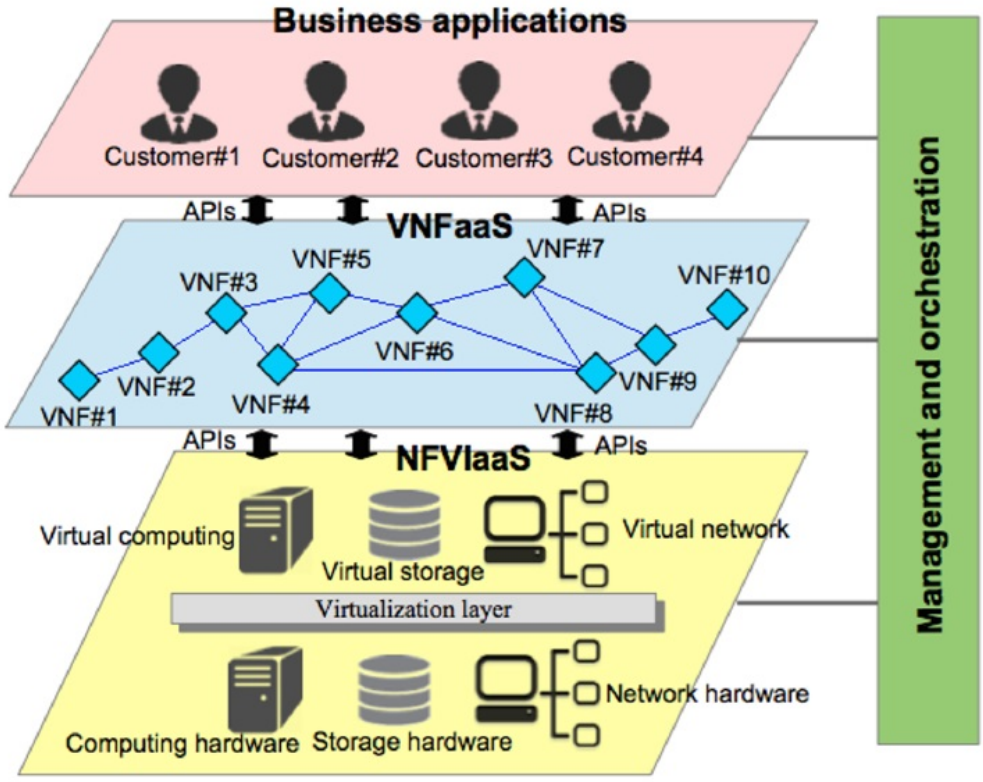

Рисунок 2.4 – Приклад поєднання моделей NFVIaaS і VNFaaS

Приклад віртуалізації (vCPE) зображено на рисунку 2.5. На рисунку зображено приклад, як відбувається взаємодія у випадку, коли в мережі є частина віртуалізованого обладнання, та частина не віртуалізованого. Функції, що надаються віртуалізованим обладнанням користувача, включають: VPN сервіс, мережевий екран, системи виявлення вторгнення, систему перевірки якості та інші. В той час як кінцевий роутер провайдера (vPE) досить складно віртуалізувати на короткий час, тому що він потребує високої пропускної спроможності.

Тим не менше, vPE може покращити масштабованість послуг віртуальної мережі, шляхом динамічної зміни розаоділу обчислювальних віртуальних ресурсів. Сервіс, наданий vPE, складається з IP VPN, Віртуальна приватна мережа (VPLS), Віртуальна приватна мережа Ethernet (EVPN), тощо.

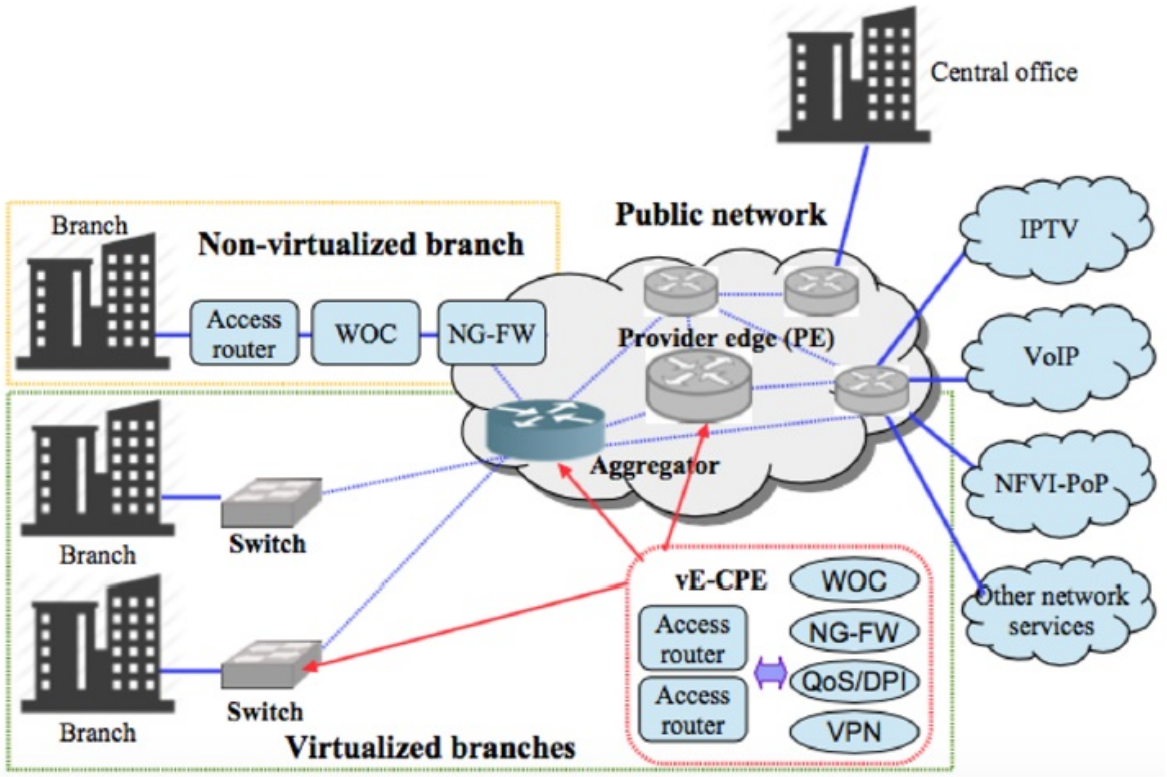

Рисунок 2.5 – Приклад використання CPE та vCPE

2.2.4 VNF Граф переадресації

Граф переадресації це один з прикладів реалізації «end-to-end» VNFaaS, який визначається як послідовність мережевих функцій, які передають мережеві пакети, приклад реалізації графу зображено на рисунку 2.6. VNF граф є аналогом з'єднання фізичних пристроїв через кабель та реалізує логічне з'єднання між віртуальними додатками. Переваги графу полягають в тому, що він пропонує розподіл ресурсів між віртуальними функціями, а також зменшує затримки при використанні та оновленні цих функцій. Іншою ключовою перевагою є додавання нових функцій, які достатньо легкі у використанні та налаштуванні.

На прикладі розглянуто більш складну структуру використання графіку переадресації, завданням якого є реалізація існуючих фізичних функцій у їх віртуалізовані еквіваленти, а також можливість додавання нових функцій.

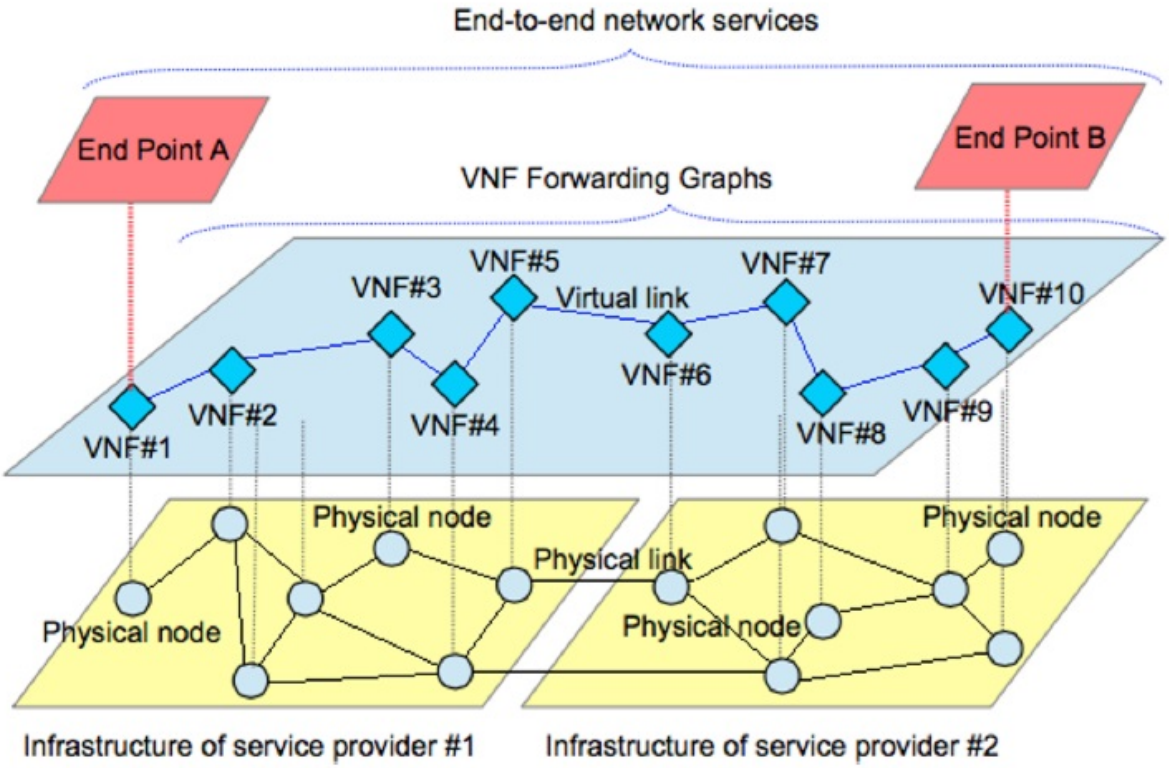

Рисунок 2.6 – Реалізація графу переадресації.

Тому провайдеру необхідно створити і вдосконалювати мережеві сервіси на абстрактному рівні, після чого реалізувати ці сервіси у відповідні ресурси NFV інфраструктури. Сервіси повинні обов'язково містити визначення типу віртуальної функції, та визначення всіх зв'язків цієї функції з оточуючою мережею.

VNF граф переадресації має стандартизований і публічний інтерфейс, який може складатися з перших чотирьох і сьомого рівнів моделі OSI. Граф переадресації можна розділити на фізичну та логічну складову мережі та всі зв'язки між цими складовими. Логічне представлення графу умовно можна розділити на чотири частини: фізичні мережеві функції, логічний інтерфейс, передача пакетів, інфраструктура мережі NFV.

Представлення фізичних мережевих функцій – це загальне представлення сервісу фізичних мережевих служб, таких як фізичний доступ, остав мережі та віртуальні машини (VM), які не можуть бути віртуалізовані провайдером послуг.

Фізичний мережевий інтерфейс – представляє собою зв’язку, для провайдера послуг, між VNF графом та фізичними мережевими функціями. Фізичний інтерфейс базується на полях з заголовку пакетів (джерело відправки, приймач).

Передача пакетів – транспортування пакетів, по всій мережі, пакети однієї групи можуть бути передані по різним шляхам залежно від зроблених налаштувань адресації.

NFV мережева інфраструктура забезпечує зв’язок служб між віртуальними функціями, що представляють вузли графу переадресації, та віртуальними функціями, які знаходяться під керуванням NFVM. Також інфраструктура забезпечує класифікацію та управління трафіком і деякі форми балансування навантаження на мережу.

Фізичне представлення графу переадресації складається з наступних елементів: фізичні зв’язки в мережі, фізичні мережеві порти, транспортні шляхи та середовище віртуальних машин.

Фізичне мережеве з’єднання – це зв'язок між інфраструктурою мережі NFV та портами фізичної мережі, які знаходяться в фізичних функціях мережі. Це зв’язки менеджменту та оркестрації між VNF та PNF.

Фізичний порт – фізичний мережевий порт на фізичній мережевій функції або на фізичному мережевому комутаторі або маршрутизаторі.

Шляхи переадресації мережі – послідовність комутаційних портів на апаратному та програмному забезпеченні. Набір операцій в NFV інфраструктурі, які налаштовані елементами управління та керування (оркестрування) послуг, ці операції представляють логічні вузли графу переадресації з'єднаних з логічним інтерфейсом вузла VNF.

Середовище віртуальних машин описує характеристики обчислювального та мережевого середовища. Граф переадресації VNF забезпечує ряд переваг, у порівнянні з існуючими графами:

- ефективність – різні мережеві служби можуть розподіляти ресурси для всіх функцій;

- еластичність – у деяких випадках функції резервного копіювання можуть бути використані на іншому фізичному обладнані;
- гнучкість – граф переадресації VNF забезпечує швидке створення, видалення та оновлення мережевих послуг;
- простота у використанні – набагато легше використовувати і керувати віртуальними функціями ніж фізичними.

З одного боку модель VNFaaS покращує транспортування ресурсів в мережі, вдосконалює розподіл ресурсів між мережевими елементами все це відбувається без контролювання загальної інфраструктури. З іншого ж боку провайдер не може контролювати кінцеві додатки користувачів, а це в свою чергу призводить до появи вразливостей.

Вразливості рівня контролю і керування. Враховуючи те, що завдяки NFV технології відбувається відокремлення програмного забезпечення від апаратної складової, велика кількість додатків і мережевих пристроїв може бути реалізована у вигляді програмного модуля на сервері провайдера. Зазвичай рівень контролю і керування стає одною з перших цілей зловмисника, так як, на цьому рівні зосереджені всі елементи керування мережею. Наприклад, атака відмови обслуговування, спрямована на рівень контролю, може вивести з ладу всю систему. Більш серйозною є проблема, якщо зловмисник отримує контроль над керуючими елементами, тоді він отримує вільний доступ до всієї мережі та її функціоналу.

Проблема небезпечних додатків і сервісів. Згідно моделі VNFaaS додатки користувачів взаємодіють один з одним через інтерфейси, і при цьому ніяк не контролюються зі сторони безпеки. Необхідно ретельно враховувати проблеми безпеки та довіри протягом всього життєвого циклу віртуальної функції, оскільки будь-яка неточність в налаштуванні або у зв’язку з іншим інтерфейсом в разі атак зловмисника може призвести до втрати контролю над всією мережею. Наприклад, зловмисник може передати по некоректно налаштованому інтерфейсу злоякісний код, який автоматично попаде на всі елементи мережі та інфікує їх.

Вразливості, які виникають внаслідок того, що багато користувачів користуються однією інфраструктурою. Між користувачами розділяються такі обчислювальні ресурси як розмітка дисків, центральний процесор, відео процесор. При цьому виникають нові вразливості системи, оскільки більшість апаратного забезпечення (наприклад, процесорів) не пристосоване для мультивикористання і не мають відповідного захисту.

2.3 Способи захисту мереж з NFV технологією

NFV технологія має дуже великий потенціал для покращення мережевих сервісів безпеки. Наприклад, такі елементи захисту, як мережевий екран, системи виявлення та попередження вторгнень, системи глибокої перевірки пакетів, можуть бути віртуалізовані та перенесені на гіпервізор і мати один зручний інтерфейс керування.

2.3.1. Покращення стандартних сервісів безпеки

З використанням NFV технології досягається наступні покращення в роботі стандартних засобів безпеки таких як фаєрвол, IPS/IDS системи та інші.

Зменшення складності використання і управління, завдяки централізованому інтерфейсу керування, та розділення рівнів керування та передачі даних. В SDN/NFV мережах на відміну від стандартних мереж не потрібно вручну вибудовувати складні послідовності елементів захисту і через них пропускати всі пакети.

Балансування навантаження. Віртуалізовані мережі дозволяють впроваджувати алгоритми балансування навантаження у програмне забезпечення, уникаючи перевантаження, в той час як в стандартних мережах, оператор повинен вручну прописувати правила балансування.

Вирішення проблеми модифікації мережевих пакетів. Оскільки деякі з інструментів захисту мають можливість змінювати заголовки пакетів, оператори мережі повинні були розташовувати всі елементи захисту у

строгому порядку. Наприклад, IDS система повинна стояти виключно після проксі сервера, для того щоб переконатися, що весь трафік змінює свою адресу через проксі. Вручну налаштовувати всі мережеві пристрої. З використанням SDN/NFV всі ці питання вирішуються на логічному рівні, на інтерфейсі контролера.

Покращення ефективності елементів захисту. В стандартній структурі мережі при розробці фаєрволів, серверів проксі, систем виявлення вторгнень зазвичай не береться до уваги те, як ці елементи будуть взаємодіяти між собою, що часто призводить до конфліктних ситуацій в роботі двох або більше інструментів захисту. В технологіях SDN/NFV цей підхід вдосконалений і поділяється на два рівні.

1. Роз'єднання програмного і апаратного забезпечення кожного окремого інструмента, і запуск нового програмного забезпечення на об'єднаній апаратній платформі.
2. Об'єднання функцій керування всіх інструментів на централізованому інтерфейсі, який логічно об'єднує ці елементи в спільний інтерфейс і налаштовує їх згідно політики безпеки компанії.

Оптимальне розгортання (розташування) інструментів захисту. В стандартній мережі для досягнення найкращого захисту, елементи захисту повинні розташовуватися у найбільш сприятливих для цього місцях. В технологіях SDN/NFV використано алгоритми, які дозволяють перенаправити злоякісний трафік відразу на потрібний елемент захисту. Наприклад, DDoS атака відразу перенаправляється на систему запобігання вторгнень і там обробляється згідно відповідних алгоритмів.

2.3.2 Ідентифікація та доступ у мережу NFV

Система ідентифікації та доступу до мережі вважається першим рівнем захисту всієї інфраструктури та сервісів. Основною функцією системи є аутентифікація, авторизація, та контроль доступу до мережі, іншими словами

– надання доступу відповідному користувачу, до відповідних ресурсів, у відповідний час, з відповідних причин. Система ідентифікації та доступу до мережі використовується для створення сесії користувача, запису його дій, та контролювання його дій згідно правил доступу для цього користувача.

Віртуалізована функція контролю доступа (AC-VNF) виконує керування великою кількістю VNF функцій та автентифікацією кінцевих користувачів. Метою AC-VNF є забезпечення сервісів безпеки та забезпечення контролю доступу до послуг згідно політики безпеки підприємства. Зокрема, автентифікація та авторизація – це необхідні механізми для перевірки прав VNF функцій, чи має право ця функція на доступ до запитаних ресурсів чи ні. Крім того, провайдер послуг, який володіє NFV інфраструктурою та розподіляє ресурси для додатків, що виконуються на його інфраструктурі, може визначати довільну політику безпеки, засновану на своїх потребах. Таким чином, кожна VNF функція може бути проконтрольована провайдером згідно встановлених правил безпеки.

AC-VNF реалізується на основі стандарту IEEE 802.1X (Стандарт IEEE для керування доступом до мережі на основі порту (PNAC)) та модифікованої версії стандарту для реалізації контролю доступу «як сервіс». Приклад реалізації функції AC-VNF показано на рисунку 2.7. Архітектура AC-VNF функції взаємодіє з чотирма різними доменами.

1. Кінцевий користувач, який подає запит на надання послуги, запит може бути ініційований як з VM з іншого віртуального сервера, так і з фізичного ПК. Комп’ютер, з якого користувач зробив запит, повинен бути напряму з’єднаний з одним з фізичних портів OpenFlow комутатора.
2. SDN контролер, який є програмним забезпеченням і контролює трафік користувача. Контроль трафіку забезпечує перевірку всього потоку запитів на користування ресурсами. Також функцією цього домену є перенаправлення запитів користувача на віртуальну функцію контролю доступу, для подальшого проходження авторизації та направлення

користувача до ресурсів, які він запитував.

3. Віртуальна функція контролю доступу, яка запущена на операційній системі Linux, і використовує протокол HostAP разом з WPA. Ця функція діє на основі протоколу IEEE 802.1X. Як тільки процедура аутентифікації та авторизації буде успішно виконана, функція контролю доступу передає позитивний результат авторизації на контролер SDN/NFV контролер.
4. Сервіс, який є певним типом ресурсів, наданих провайдером, власник ресурсів може керувати цими ресурсами незалежно, використовуючи різні процедури захисту.

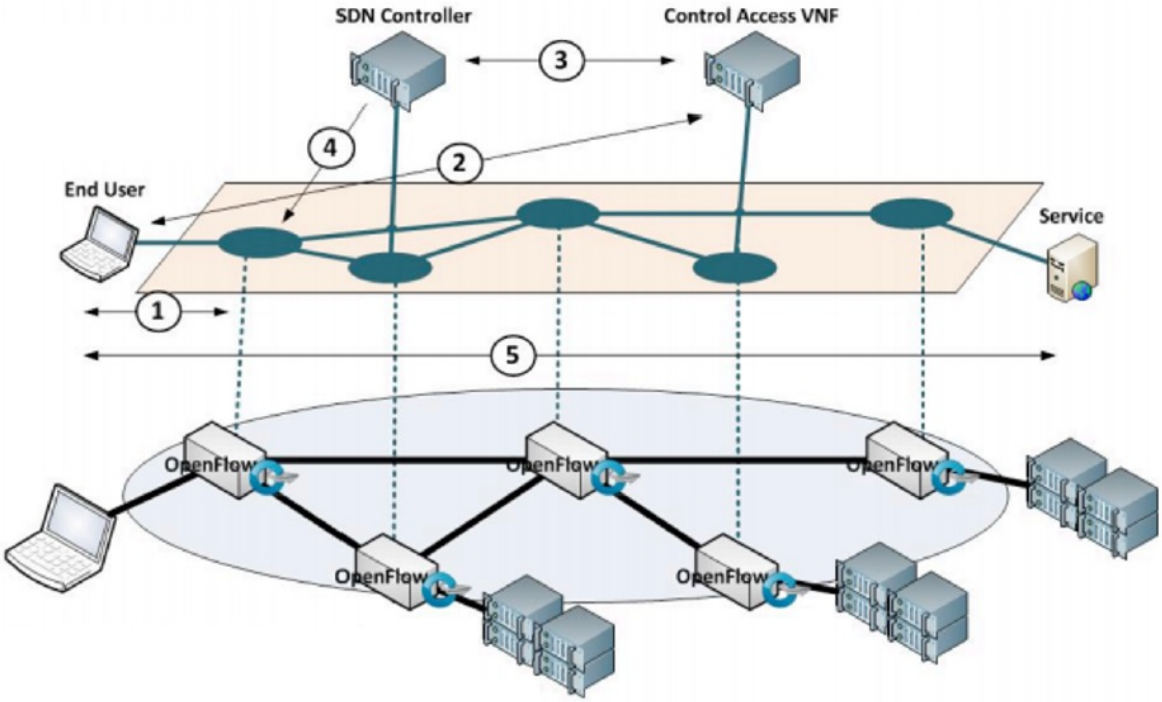

Рисунок 2.7 – Приклад реалізації функції AC-VNF

Опис алгоритму авторизації та отримання доступу до мережі кінцевого користувача, згідно поданого рисунку 2.7 наступний.

1. Доступ користувача до ресурсів мережі обмежений.
2. Запит користувача перенаправляється на функцію контролю доступу.
3. Результат аутентифікації та авторизації передається на SDN контролер.
4. У випадку коректної авторизації SDN контролер буде пропускати через

себе всі запити даних від користувача.

5. Кінцевому користувачу надається доступ до ресурсів мережі.

Враховуючи те, що керування віртуальною функцією контролю доступу просте, в ній можна активувати або деактивувати доступ до мережі, не порушуючи при цьому коректну роботу інших послуг, оператори мережі можуть легко переналаштовувати контроль доступу до VNF інфраструктури за допомогою наданих інструментів.

2.3.3 Ізоляція мережі

Як правило, існує два основних типи мережевої ізоляції: ізоляція мережевого трафіку та ізоляція функцій безпеки мережі. Перше рішення полягає в тому, щоб фізично або логічно сегментувати мережу для забезпечення безпечного з'єднання та надання вищої пропускної спроможності для деяких користувачів. У випадку фізичної ізоляції мережі, ізолюються окремі мережеві інтерфейси контролерів, які як правило, призначені до конкретних послуг мережі. Логічна ізоляція мережі використовує програмне забезпечення для ізоляції фізичних мережевих ресурсів, наприклад послуги VLAN. Таким чином, мережевий трафік від кількох користувачів ділить однакові фізичні інтерфейси, тоді як кожен додаток може використовувати лише свій попередньо виділений ресурс. Друге рішення ізоляції трафіку – це управління ресурсами або керування якості обслуговування QoS. Це рішення покладається на ефективний моніторинг та управління, забезпечуючи те, що користувачі споживають лише свою частку пропускної спроможності в мережі. Крім того, існують додаткові механізми для поліпшення ізоляції мережевого трафіку, такі як:

1) встановлення угоди про надання послуг, що відповідає QoS в мережі;
2) ідентифікація вузьких місць для запобігання перевантажень в мережі;
3) збір інформації моніторингу безпеки для запобігання атак, для цього є кілька технічних способів:
   1) використання мережевих брандмауерів для фільтрування можливих

загроз безпеки, які надходять з недовірених джерел, слідуючи чому, пакети вхідних даних блокуються якщо вони не відповідають правилам політики безпеки;

2) застосування маркування VLAN для забезпечення додаткових рівнів безпеки, контролювання пропускної спроможності і гарантування, що пакети, надіслані одним користувачем, не можуть бути прочитані іншим;
3) прийняття політики безпеки основаної на ролях користувачів, або керування доступом на основі ролі користувачів для забезпечення авторизації до ресурсних об'єктів.

В віртуалізованих технологіях існує велика кількість способів ізоляції. В той час, коли розміри обчислювальної мережі стрімко зростають, ізоляція мережі стає необхідною мірою.

Згідно технології NFV кожна мережа провайдера повинна бути ізольованою, що потребу виконання двох основних вимог. Перша – це необхідність гарантування, того, що сервіси однієї мережі не будуть впливати на іншу мережу, навіть випадково, або під керуванням зловмисника. Друга вимога – створена модель повинна бути економічно доцільною. Приклад логічної ізоляції, реалізованої серед багатьох провайдерів зображено на рисунку 2.8, які поділяють одну інфраструктуру, не вимагаючи при цьому додаткового обладнання. Кожен провайдер бачить тільки свої VM, які підключені до одного віртуального комутатора.

На рисунку 2.8 показана концепція ізоляції мережі TVDc. Дві фізичні машини (PM) зв’язані через мережевий комутатор. На кожній машині працює дві віртуальні машини, один з них позначається як Market Analysis (MA), а інша – Security Underwritting (SU). Вона містить два підключення VLAN, одне підключення здійснено логічно через програмне забезпечення, підключає VMM з кожним PM, а інше підключення реалізоване в апаратній мережі - зовнішній комутатор до PM. VMM на кожному PM обмежує кожне програмне забезпечення VLAN для надсилання та отримання пакетів тільки з

відповідної апаратної VLAN.

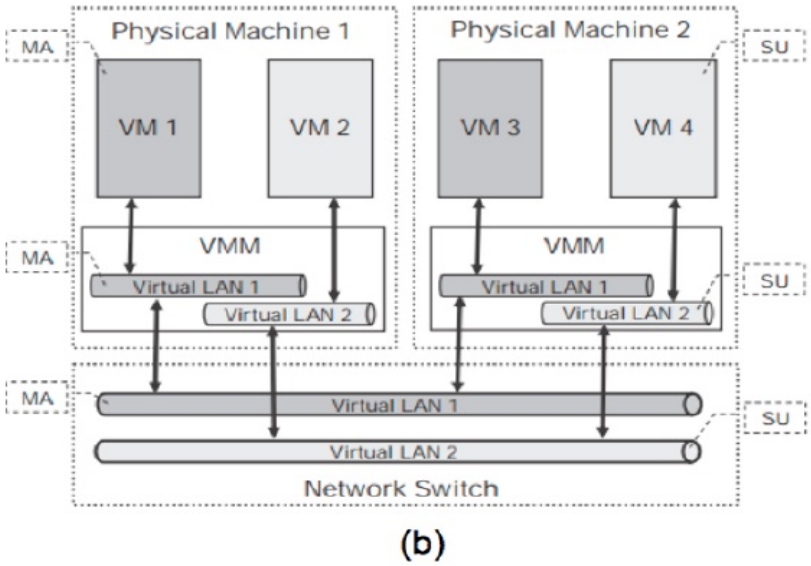

Рисунок 2.8 – Реалізація ізоляції мережі

2.3.4 Захист даних

Захист даних є важливим рішенням для загального захисту в віртуалізованих мережах. Спосіб захисту даних широко використовується для забезпечення конфіденційності та приватності, які дозволяють користувачам контролювати безпеку своєї інформації. Реалізацію захисту даних умовно можна розділити на такі частини:

- шифрування даних;
- управління ключами шифрування;
- ізоляція даних;
- попередження витоку даних;

Шифрування даних. Існує дуже велика кількість різноманітних алгоритмів шифрування даних таких як RSA та AES. Також більшість протоколів криптографічного захисту використовують функції шифрування. Проте досягти повного захисту даних лише шифруванням не вдасться, тим паче враховуючи тенденцію до підключення великої кількості нових додатків з різними особливостями. Наприклад, для того щоб підвищити

рівень безпеки системи шифрування, на стороні дешифрації вводяться додаткові перевірки, розшифрувати повідомлення може лише той користувач, якому це дозволено згідно з політикою безпеки.

Управління ключами. Шифрування використовується для того, щоб забезпечити захист інформації від неавторизованого розкриття. Основна техніка для шифрування потребує наявності ключа для описання того, як інформація буде шифруватися та дешифруватися. Ці ключі створюються і контролюються на протязі усього життєвого циклу менеджером криптографічних ключів.

Ізоляція даних. Основна вимога до ізоляції даних в мережі – це зменшити взаємний вплив одних даних на інші.

Розглянемо приклад реалізації захисту даних у NFV мережі. Менеджер ключів у хмарному середовищі повинен коректно розподіляти велику кількість ключів шифрування. Різні додатки потребують різних ключів шифрування та дешифрування. Кількість ключів, якими потрібно керувати, весь час збільшується, тому для того щоб підвищити швидкодію шифрування, шифрування і дешифрування ключів переноситься на сторону користувача. Це дозволяє користувачу впевнитися в правильності шифрування його даних. Приклад такої реалізації зображено на рисунку 2.9. Елемент з позначкою «1» - менеджер домену, елемент з позначкою «2» клієнт домену.

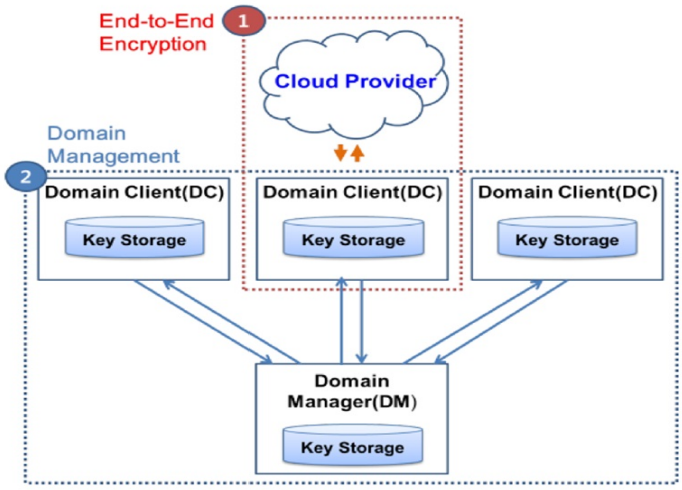

Рисунок 2.9 – Приклад реалізації захисту даних

Користувач «А» на один пристрій встановлює додаток менеджеру домену, а на інший пристрій встановлює клієнт домену. З цією концепцією, дані користувача захищені в межах його особистого домену шляхом повного шифрування та керування доменом.

Менеджер доменів (DM): Відповідальний за управління учасниками свого домену, реєструючи та видаляючи їх. DM додаток створює ключ домену «k» і пересилає цей ключ усім членам домену для виконання подальшого процесу шифрування та дешифрування. Завдання DM є реєстрування та видалення пристроїв та оновлення доменного ключа.

Клієнт домену (DC): Відповідальний за шифрування користувацьких даних перед тим як експортувати їх або зберігати в хмарі, а також розшифровувати зашифровані дані після завантаження з хмари в особистий домен.

Висновки до розділу 2

В другому розділі детально проаналізовані такі моделі реалізації SDN/NFV мережі як віртуалізована структура як сервіс (NFVIaaS), віртуалізована платформа як сервіс (VNPaaS) та віртуалізована функція як сервіс VNFaaS. Кожна з моделей має свої переваги і недоліки. Показано, що модель NFVIaaS підійде для великих підприємств, оскільки згідно цієї моделі провайдер надає кінцевому користувачу інфраструктуру для виконання всіх його додатків на ній, і не накладає обмежень. В той час як VNPaaS та VNFaaS можуть підійти для малих підприємств, оскільки в цих моделях провайдер вже пропонує певний шаблон мережі і ключових сервісів, на основі якого будується мережа.

Проведено аналіз вразливостей та методів захисту моделей побудови віртуальної мережі, та показано що кожна з моделей потребує унікального захисту, на основі проведеного аналізу сформовані основні рекомендації, таким чином у моделі VNFaaS доцільно встановлювати на вході функцію авторизації та керування доступу, а вже в мережі використовувати механізми шифрування даних. Для NFVIaaS важливим елементом захисту являється політика безпеки для користувача інфраструктурою, а також слід уникати залежності елементів однієї інфраструктури від іншої.

Розділ 3. Розроблена система захисту мереж SDN з використанням NFV технології

3.1 Використаний інструментарій мови програмування Python

Для моделювання та розробки тестової програмно-конфігуровної мережі з використанням віртуалізованих функцій, були використані наступні бібліотеки:

- socket;
- Tkinter;
- threading;
- time;
- nmap;

Модуль socket використаний для створення віртуальних функцій сервера та клієнта.

Модуль tkinter використаний для оформлення користувацького інтерфейсу контролера.

Модуль threading для організації мультипоточного прийому даних на віртуалізованих функціях.

Модуль time для генерації системних зупинок в виконанні скрипта.

Модуль nmap для генерації тестових випадків атак на віртуалізовані функції, використовує багато можливостей окремої утіліти nmap.

3.2 Опис компонентів системи захисту та топології мережі

Розроблена система моделювання мережі розроблена за допомогою бібліотек Python. На рисунку 3.1 показана загальна структура проекту python в середовищі розробки PyCharm 2017.2.3.

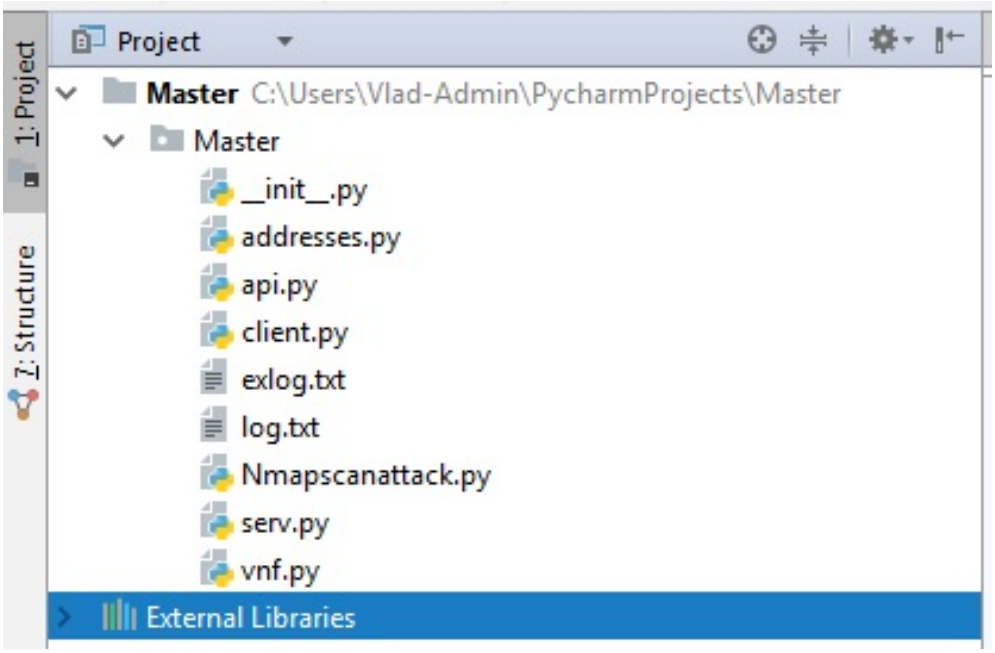

Рисунок 3.1 – Загальна структура розробленої системи захисту

Файл *__init__.py* додається до проекту для того, щоб коректно виконувалось імпортування програмних модулів в середині розробленого проекту.

Файл *addresses.py* в цей файл винесені всі мережеві адреси, які використовуються в проекті.

Файл *api.py* в цьому файлі, за допомогою бібліотеки tkinter, реалізований користувацький інтерфейс програми-контролера віртуалізованої мережі.

Файл *client.py* містить в собі функції, які запускають клієнтську частину віртуалізованих функцій.

Файл *serv.py* містить в собі функції, які запускають серверну частину віртуалізованих мережевих функцій.

Файл *vnf.py* містить в собі шаблони для функцій сервера, та функцій клієнта.

Файл *Nmapscanattack.py* містить в собі скрипти імітаційних керованих атак на віртуалізовані функції мережі, з використанням утіліти nmap та бібліотек мови python пов’язаних з утилітою nmap.

3.2.1 Детальний опис розроблених модулів

Файл *vnf.py*:

```
 def Simple_VNF_client(client_host, client_port, name_c):
    f = open('log.txt', 'a')
    s_client = socket.socket()
    s_client.connect((client_host, client_port))

    s_client.sendall(('Im Client '+ name_c).encode())
    #f.write('Im Client ' + name_c + '\n')
    time.sleep(2)
    try:
        dat = s_client.recv(1024)
        print(dat)
        #f.write(dat)
    except:
        print('No data from server')

    finally:
        f.close()
```

Шаблон функції для встановлення підключення клієнтської частини віртуалізованої функції до серверної частини, використовуючи сокети, підключення за адресою яка буде взята з окремого файлу *addresses.py*. Також ведеться легування подій на серверній та клієнтській частині.

Значення «s_client.recv(1024)» встановлено 1024 байти, що означає, що клієнтська частина в змозі прийняти відповідь від сервера тільки в розмірі одного мегабайта, якщо повідомлення буде більшим за розміром, частина, яка виходить за межі мегабайту буде відкинута.

```
def Simple_VNF_server(serv_host, serv_port,  name_s):
 f = open('log.txt', 'a')
 s_server = socket.socket()
 s_server.bind((serv_host, serv_port))
 s_server.listen(5)

 print ('Server ', name_s, ' has activated')
 f.write('Server '+ name_s + ' has activated' + '\n')

 f.close()
 return s_server
```

Шаблон серверної частини віртуалізованої функції. Відбувається ініціалізація сокету, на якому будуть прийматися з’єднання від клієнта. Рядок

s_server.listen (5) – означає, що сокет серверу має чергу з'єднань у розмірі п'яти з'єднань, якщо підключень більше – лишні з'єднання відкидаються. Функція повертає як результат сокет, до якого виконалась прив'язка прослуховуючого сервера.

```
def accepting_con(Socket):
    f1 = open ('exlog.txt', 'a')
    f = open('log.txt', 'a')
    while True:

        conn, addr = Socket.accept()


        print('Got connection from  ', addr)
        f.write('Got connection from'  + str(addr) + '\n')

        try:
            data = conn.recv(1064).decode()
            print(data, '\n')
            if data:
                f1.write(data + '\n')
            conn.send('You have connected'.encode())

        except:
            print('No data from client ')
        finally:

            time.sleep(3)
            print('Get active connection with:
',conn.getpeername())
            f.write('Get active connection with:
'+str(conn.getpeername())+ '\n')
            time.sleep(1)
            print('Connection with: ', conn.getpeername(), ' has
closed')
            f.write('Connection with: '+ str(conn.getpeername())
+ ' has closed'+ '\n')
            conn.close()
            f.close()
            f1.close()
```

Функція прийняття кожного окремого з'єднання на порт сервера, як параметр приймає сокет, на який і будуть виконуватися з'єднання. Ця функція буде викликатися в декількох потоках, для того, щоб сервер одночасно міг обробляти декілька з'єднань, які знаходяться в його черзі. conn.recv(1064).decode() – рядок означає, що буфер серверу розрахований на

повідомлення не більше ніж 1064 байти, частина повідомлення, яка перевищує задане обмеження, буде відкинута. Оскільки між сокетами дані передаються у байтах, команда decode() декодує повідомлення і перетворює його в символьний рядок.

Файл serv.py містить в собі функцію активації серверної частини віртуалізованої функції.

```
class Test():

    def t1_start(event):
        global Mysocket
        global Server_T1
        global Server_T2
        global Server_T3

        Mysocket =
vnf.Simple_VNF_server(addresses.server_host_first_vnf,
addresses.server_port_first_vnf, addresses.name_first)
        Server_T1 = threading.Thread(target = vnf.accepting_con,
args= (Mysocket,))
        Server_T1.start()
        Server_T2 = threading.Thread(target = vnf.accepting_con,
args= (Mysocket,))
        Server_T2.start()
        Server_T3 = threading.Thread(target = vnf.accepting_con,
args= (Mysocket,))
        Server_T3.start()
```

За допомогою асинхроних потоків запускається серверна частина віртуалізованої функції. Згідно написаним вище рядкам кожний сервер в змодельованій мережі буде приймати одночасно по три з'єднання від клієнтів.

Файл client.py містить в собі реалізацію клієнтських підключень через потоки

```
class Client_Thread(threading.Thread):
    def __init__(self, client_host, client_port, name_c):
        threading.Thread.__init__(self)
        self.client_host = client_host
        self.client_port = client_port
        self.name_c = name_c

    def run(self):

        vnf.Simple_VNF_client(self.client_host,
```

```
self.client_port, self.name_c)

class Test():

    def t1_start(event):
        t1_serv = Client_Thread(addresses.server_host_first_vnf,
addresses.server_port_first_vnf, addresses.name_first)
        t1_serv.start()

    def t2_start(event):
        t2_serv =
Client_Thread(addresses.server_host_second_vnf,
addresses.server_port_second_vnf, addresses.name_second)
        t2_serv.start()

    def t3_start(event):
        t3_serv = Client_Thread(addresses.server_host_third_vnf,
addresses.server_port_third_vnf, addresses.name_third)
        t3_serv.start()

    def t4_start(event):
        t4_serv =
Client_Thread(addresses.server_host_fourth_vnf,
addresses.server_port_fourth_vnf, addresses.name_fourth)
        t4_serv.start()

    def t5_start(event):
        t5_serv = Client_Thread(addresses.server_host_fifth_vnf,
addresses.server_port_fifth_vnf, addresses.name_fifth)
        t5_serv.start()
```

Тут реалізовано клас Client_Thread, в якому цільовою функцією для запуску потоку є функція run, в якій відбувається виклик шаблонного з'єднання з заданими параметрами. Функції класу Test виконують ініціалізацію об'єктів класу Client_Thread та запускають відповідні з'єднання в окремих потоках.

Файл api.py - в цьому файлі реалізовано користувацький інтерфейс контролеру мережі. На рисунку 3.2 зображено стартовий інтерфейс користувацького додатку.

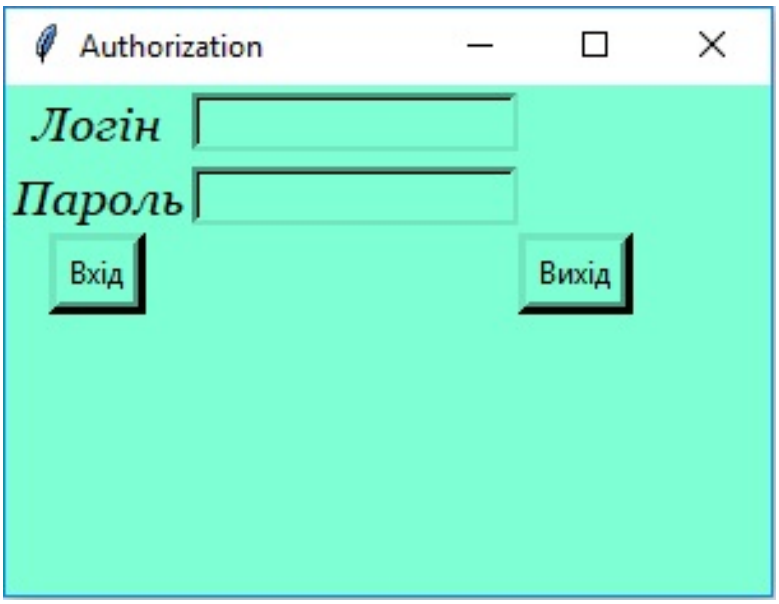

Рисунок 3.2 – Форма авторизації на мережевому контролері

Після вводу коректного логіну та паролю користувача на формі авторизації, що зображена на рисунку 3.2. користувача направляє на головну сторінку мережевого контролера, яка зображена на рисунку 3.3.

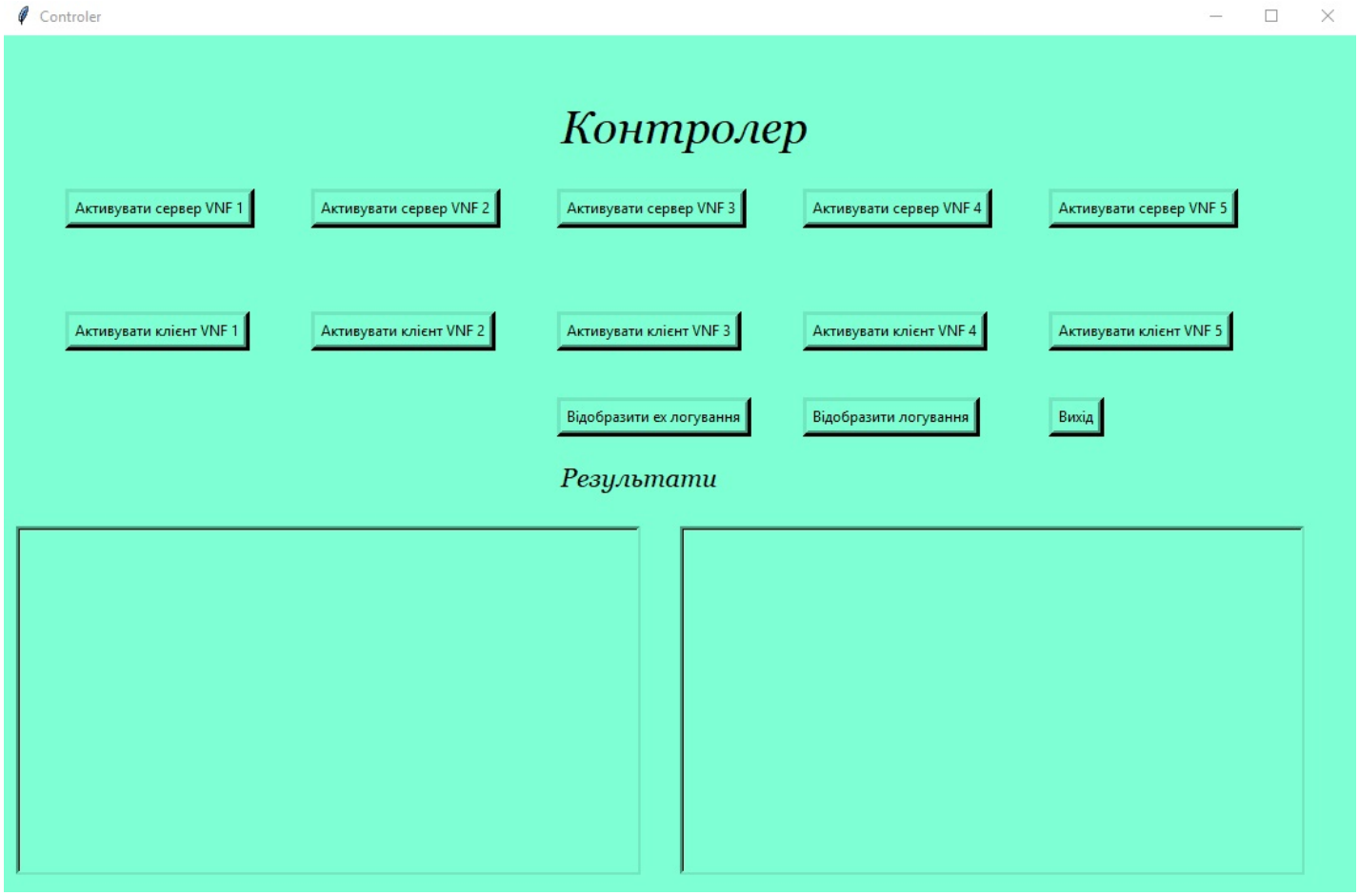

Рисунок 3.3 – Основний інтерфейс мережевого контролера

На мережевому інтерфейсі мережевого контролеру розміщено наступні елементи:

- кнопки активації серверної частини віртуалізованих функцій;
- кнопки активації клієнтських з’єднань з серверною частиною;
- кнопка виводу звичайного логування;
- кнопка виводу розширеного логування;
- кнопка виходу;
- два поля для виводу логування.

Файл *addresses.py*

```
name_first = 'VNF 1'
server_host_first_vnf = '192.168.1.247'
server_port_first_vnf = 9999

name_second = 'VNF 2'
server_host_second_vnf = '192.168.1.247'
server_port_second_vnf = 9998

name_third = 'VNF 3'
server_host_third_vnf = '192.168.1.247'
server_port_third_vnf = 9997

name_fourth = 'VNF 4'
server_host_fourth_vnf = '192.168.1.247'
server_port_fourth_vnf = 9996

name_fifth = 'VNF 5'
server_host_fifth_vnf = '192.168.1.247'
server_port_fifth_vnf = 9995
```

«192.168.1.247» - локальна IP адреса комп’ютера, на якому запущено розроблену систему. Значення змінних server_port відповідають номеру порта, на якому буде розвернуто серверну частину відповідної віртуальної функції.

Файл *Nmapscanattack.py* містить в собі скрипти імітаційних керованих атак на віртуалізовані функції мережі, з використанням утіліти nmap та

бібліотек мови python, пов’язаних з утилітою nmap. Запускаються ці скрипти з іншого комп’ютера, який знаходиться в цій же фізичній мережі за адресою «192.168.1.201». На рисунку 3.4 зображено фрагмент коду імітованої атаки, яка виконується за допомогою утіліти nmap з іншого комп’ютера.

```
1  import nmap
2
3  nmScan = nmap.PortScanner()
4
5  print (nmScan.scan ('192.168.1.247', '9999'))
6
```

Рисунок 3.4 - Фрагмент керованої атаки сканування портів nmap

На рисунку 3.5 наведено фрагмент коду імітованої атаки на переповнення буферу сервера.

```
12 time.sleep(10)
13 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
14
15 s.connect(("192.168.1.247", 9999))
16
17 buff = '\x41' * 2006
18
19 shellcode = ...
20
21 nop = '\x90' * 16
22
23 overflow = 'TRUN .' + buff + '\x05\x12\x50\x62' + nop
24 s.send(overflow.encode())
```

Рисунок 3.5 Фрагмент коду атаки на переповнення буферу

Після з’єднання з серверною частиною здійснюється спроба відправити велику кількість даних для того, щоб сокет серверу не зміг коректно опрацювати це з’єднання.

На рисунку 3.6 зображено приклад DoS атаки. В коді описаний наступний спосіб атаки, після ініціалізації сокету в нескінченному циклі відбувається створення нових потоків з підключенням до серверу, і кожний з них в нескінченному потоці відправляє дані на сервер.

```
28  import threading
29  import socket
30
31  s_client = socket.socket()
32
33  def dos():
34      while True:
35          s_client.connect(("192.168.1.247", 9999))
36          s_client.send('1')
37
38  while True:
39      threading.Thread(target=dos).start()
40
```

Рисунок 3.6. – Фрагмент керованої DoS атаки.

3.3. Аналіз захищеності мережі

Для запуску і тестування мережі використовується два фізичних комп'ютера, які знаходяться у одній мережі «198.162.1.247» - основна машина, «198.162.1.201» - допоміжна машина.

1-й крок: на обох машинах необхідна утиліта nmap на допомыжный маштні встановлено Kali linux там утіліта nmap є за замовчуванням. Для основної машини під Windows 10 необхідно завантажити утиліту, а також встановити відповідні пакети для інтерпретатору python. На рисунку 3.7 ілюстрація встановлення пакетів python.

```
C:\Users\Vlad-Admin>pip install python-nmap
Requirement already satisfied: python-nmap in c:\users\vlad-admin\appdata\local\programs\python\python35\lib\site-packag
es (0.6.1)
```

Рисунок 3.7 – Встановлення nmap пакетів для python

2-й крок. Запускаємо скрипт api.py у вікні авторизації вводимо відповідні дані облікового запису адміністратора та заходимо в користувацький інтерфейс мережевого конртолера, прохід авторизації показано на рисунку 3.8.

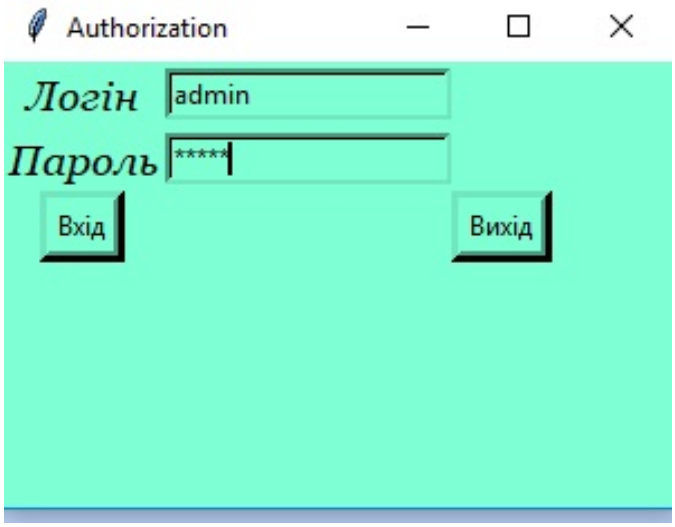

Рисунок 3.8. – проведення авторизації

3-й крок. По черзі активуємо всі п’ять серверних частин виводимо логування або дивимось в консоль. На рисунку 3.9 зображена активація серверних частин.

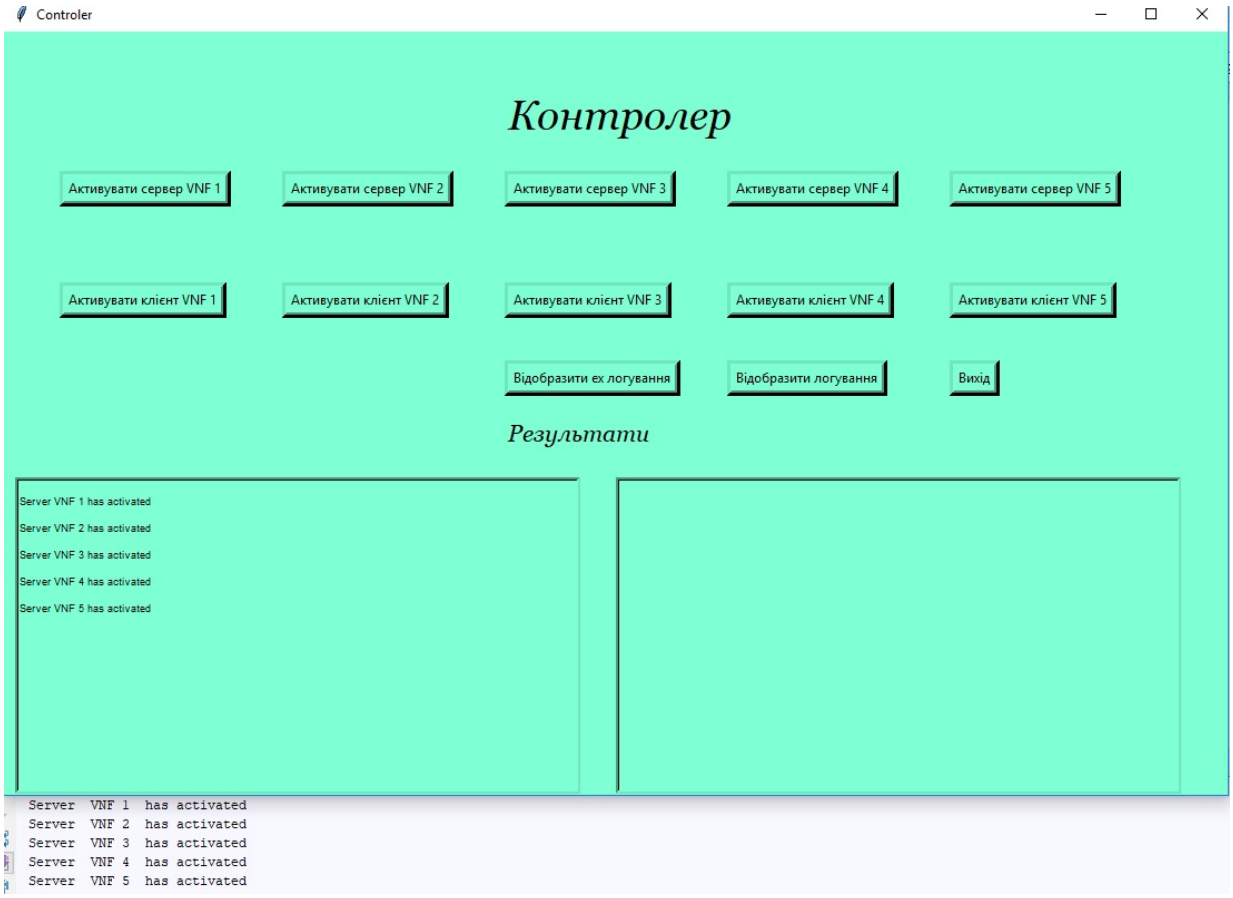

Рисунок 3.9 - Активація серверних частин

4-й крок. Додатковий крок для перевірки коректності відкритих портів,

виконуємо сканування відкритих портів на «основній» машині. На рисунку 3.10 зображено результат тестової перевірки відкритих портів «основної» машини.

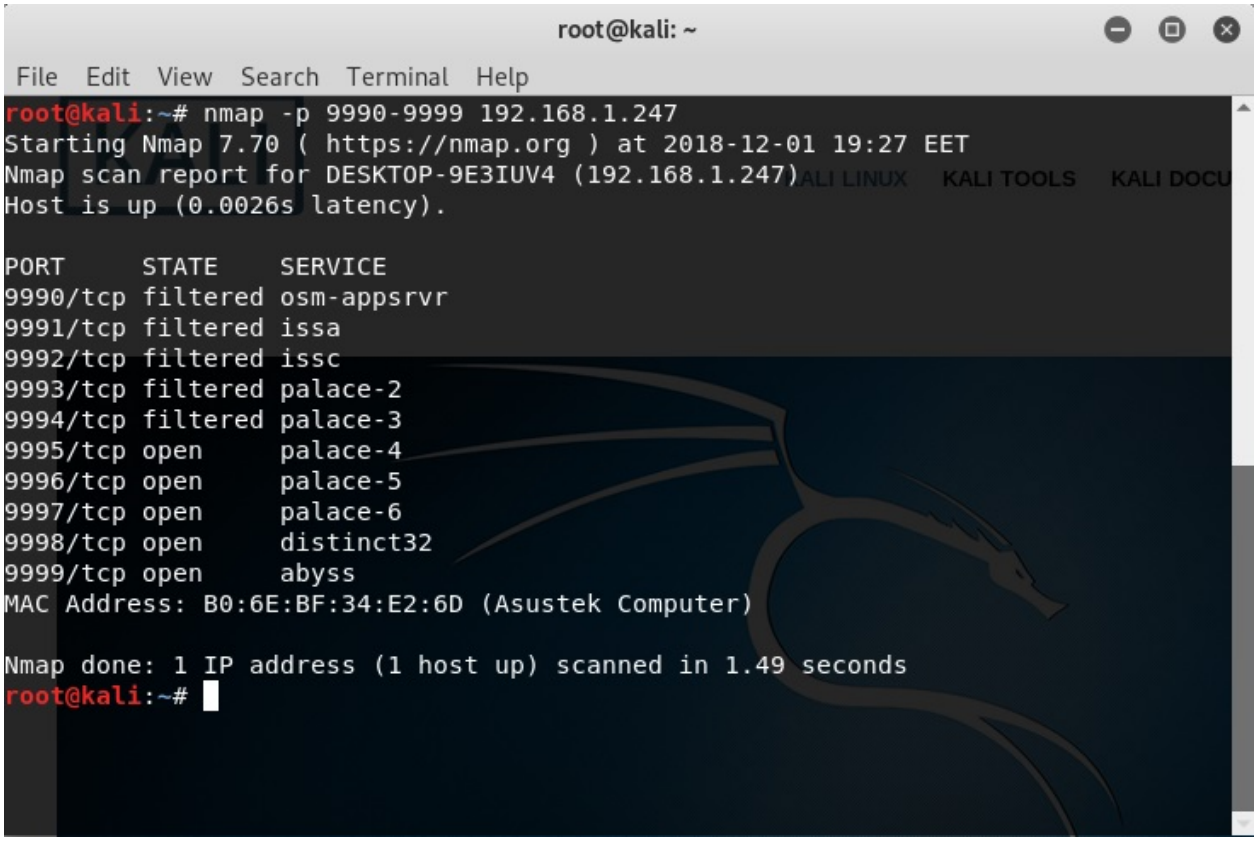

Рисунок 3.10 – Тестова перевірка відкритих портів

Виходячи з результатів виконаної перевірки бачимо, що порти 9995-9999 були коректно відкриті через додаток, та очікують з’єднання. Інші порти мають статус filtered, це означає, що сервери захищені від прямого з’єдняння і знаходяться під захистом мережевого екрану.

5-й крок. Запускаємо першу клієнтську частину один раз. На рисунку 3.11 зображено виконання п’ятого кроку. Після натискання кнопки активації маємо сповіщення серверу, що з ним встановлено з’єднання з відповідної адреси і порту. Далі виведено сповіщення клієнта та відповідь серверу. І останнім рядком показане активне з’єднання.

```
Got connection from   ('192.168.1.247', 24922)
Im Client VNF 1

b'You have connected'
Get active connection with:  ('192.168.1.247', 24922)
```

Рисунок 3.11 – Перше з’єднання з сервером

6-й крок. За налаштуваннями сервер тримає з’єднання з клієнтом 20 секунд, після чого з’єднання розривається і сервер приймає нове з’єднання з черги, якщо воно там є, або чекає нових з’єднань. Також одночасно на сервері працює три потоки, які приймають з’єднання. Кожен з них має свою чергу, яка задається параметром socket.listen(). Ілюстрація встановлення відразу кількох з’єднань зображена на рисунку 3.12.

```
Server  VNF 1  has activated
Server  VNF 2  has activated
Server  VNF 3  has activated
Server  VNF 4  has activated
Server  VNF 5  has activated
Got connection from   ('192.168.1.247', 24927)
Im Client VNF 1

Got connection from   ('192.168.1.247', 24928)
Im Client VNF 2

Got connection from   ('192.168.1.247', 24929)
Im Client VNF 3

Got connection from   ('192.168.1.247', 24930)
Im Client VNF 1

b'You have connected'
Got connection from   ('192.168.1.247', 24931)
Im Client VNF 2

b'You have connected'
Got connection from   ('192.168.1.247', 24932)
Im Client VNF 4

Get active connection with:  ('192.168.1.247', 24927)
b'You have connected'
Get active connection with:  ('192.168.1.247', 24928)
b'You have connected'
b'You have connected'
Get active connection with:  ('192.168.1.247', 24929)
Get active connection with:  ('192.168.1.247', 24930)
b'You have connected'
Get active connection with:  ('192.168.1.247', 24931)
Get active connection with:  ('192.168.1.247', 24932)
```

Рисунок 3.12 – Ілюстрація встановлення кількох з’єднань

7-й крок. Для того, щоб не відбулось DoS атаки, реалізується відкидання підключень, які не вміщаються у чергу. Встановлюємо server.listen(3), маємо чергу з трьох підключень для кожного з трьох потоків, і пам’ятаємо, що сервер тримає з’єднання 20 секунд. Отже сервер може прийняти по одному з’єднанню на сокет і плюс по три в чергу для кожного потоку, тому маємо 12 з’єднань. Тепер робимо тестовий випадок «міні DoS» атаки на 20 з’єднань. На допоміжній машині запускаємо наступний скрипт (рисунок 3.14). Результат на основній машині рисунок 3.13 всі з’єднання відбулися, але на допоміжній машині маємо помилку тайм ауту.

```
Got connection from   ('192.168.1.201', 36610)
Got connection from   ('192.168.1.201', 36612)
Got connection from   ('192.168.1.201', 36614)
3

0

1

Get active connection with:  ('192.168.1.201', 36612)
Get active connection with:  ('192.168.1.201', 36614)
Get active connection with:  ('192.168.1.201', 36610)
Connection with:  ('192.168.1.201', 36614)  has closed
Connection with:  ('192.168.1.201', 36612)  has closed
Got connection from   ('192.168.1.201', 36618)
Got connection from   ('192.168.1.201', 36622)
2

6

Connection with:  ('192.168.1.201', 36610)  has closed
Got connection from   ('192.168.1.201', 36616)
4

Get active connection with:  ('192.168.1.201', 36622)
Get active connection with:  ('192.168.1.201', 36618)
Get active connection with:  ('192.168.1.201', 36616)
Connection with:  ('192.168.1.201', 36622)  has closed
Connection with:  ('192.168.1.201', 36618)  has closed
Got connection from   ('192.168.1.201', 36626)
Got connection from   ('192.168.1.201', 36620)
7

5

Connection with:  ('192.168.1.201', 36616)  has closed
Got connection from   ('192.168.1.201', 36624)
8

Get active connection with:  ('192.168.1.201', 36620)
Get active connection with:  ('192.168.1.201', 36626)
Get active connection with:  ('192.168.1.201', 36624)
Connection with:  ('192.168.1.201', 36620)  has closed
Got connection from   ('192.168.1.201', 36630)
10

Connection with:  ('192.168.1.201', 36624)  has closed
Connection with:  ('192.168.1.201', 36626)  has closed
Got connection from   ('192.168.1.201', 36628)
9
```

Рисунок 3.13 – Результат на основній машині

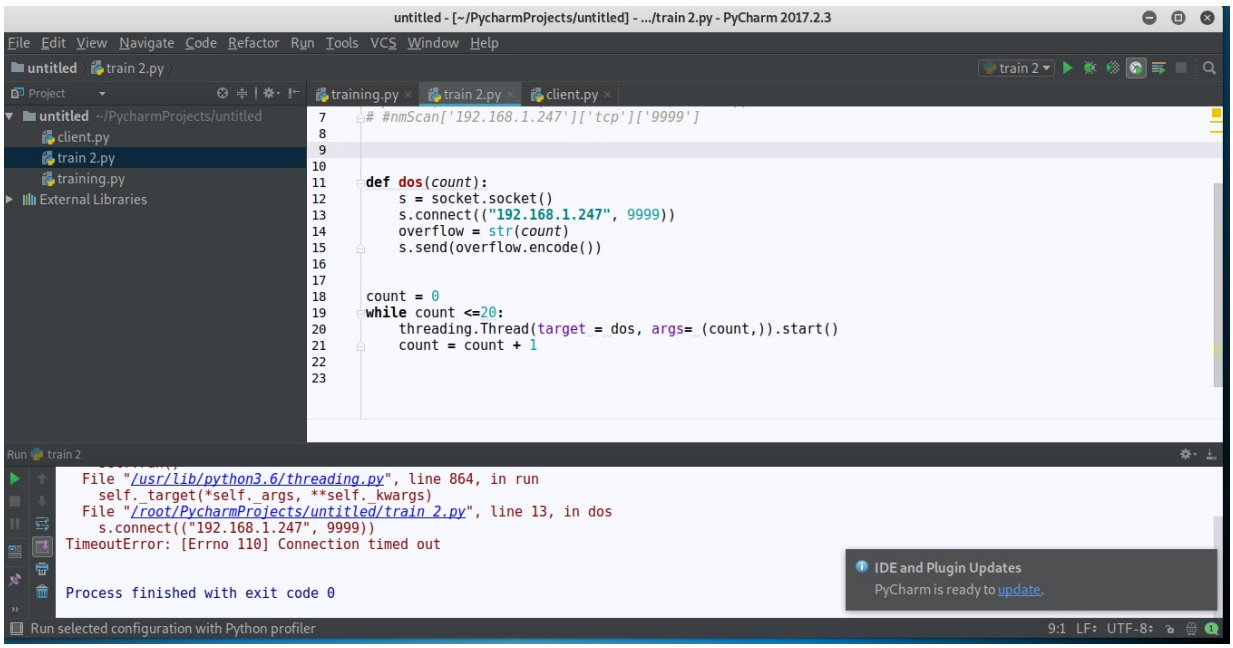

Рисунок 3.14 – Результат виконання 20 підключень з допоміжної машини

В допоміжному скрипті будо запущено 20 з'єднань у двадцяти асинхронних потоках. Таким чином, ми бачимо, що сервер в деякий момент часу є недоступним (поки не вивільняється черга). Для вирішення цієї проблеми можна збільшити чергу з'єднань на сервері або збільшити кількість потоків, що обробляють з'єднання.

8-крок. Тепер виконаємо керовану атаку на переповнення буферу сервера. Для цього на «допоміжній машині» виконуємо наступний скрипт на рисунку 3.15.

```
import socket
import time

s = socket.socket()

s.connect(("192.168.1.247", 9999))

buff = 'A' * 10000

overflow = '.' + buff
s.send(overflow.encode())
```

Рисунок 3.15 - Спроба переповнення буферу

А на основній машині для простоти перевірки встановити буфер повідомлень на сервері data = conn.recv(20).decode().

На рисунку 3.16 показано результат прийому на сервері

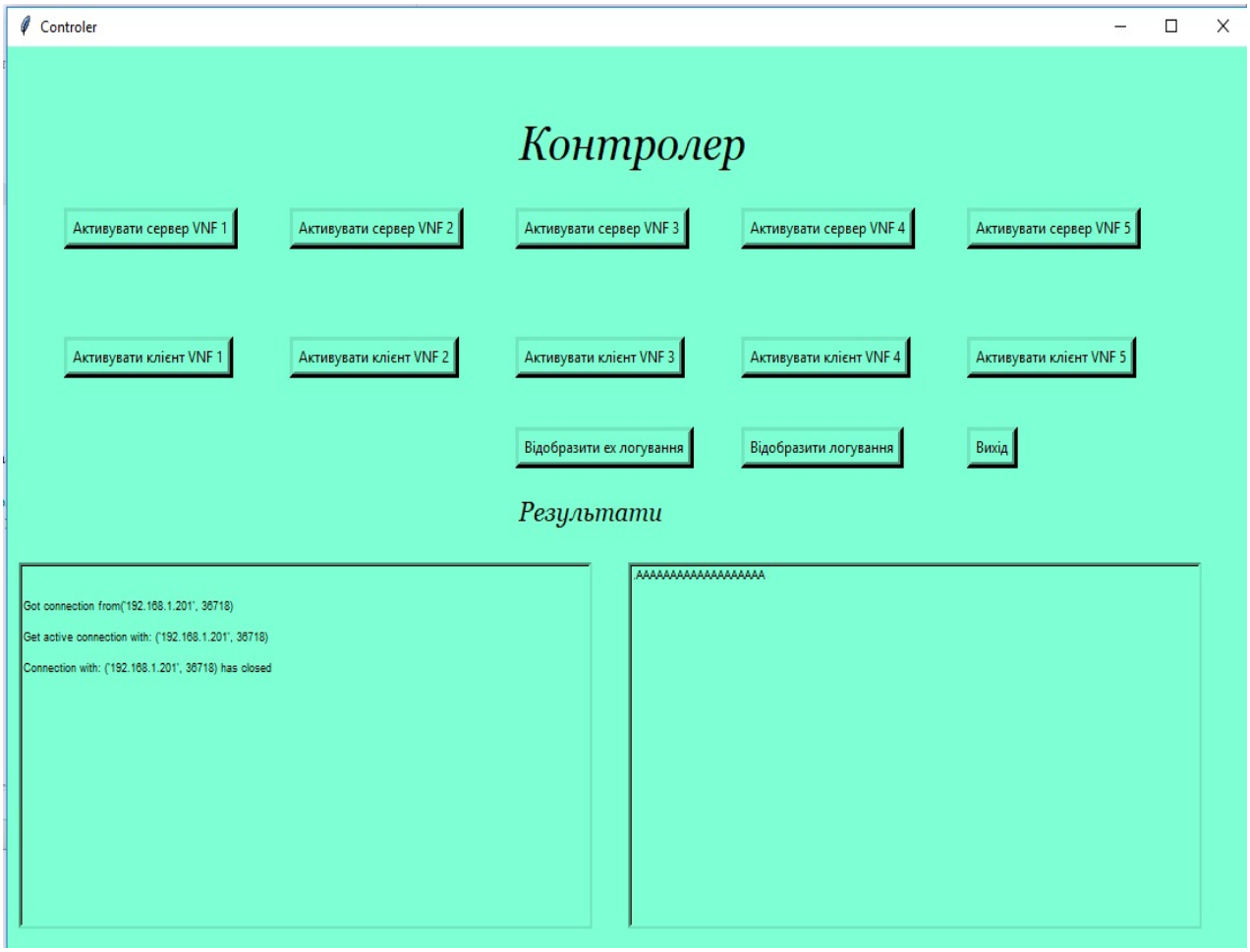

Рисунок 3.16 – Результат спроби атаки на переповнення буферу

Виходячи з результатів бачимо, що все повідомлення, що не вмістилося в буфер сервера було відкинуто.

9-крок. Перевірка на сканування портів, для цього прикладу, зробимо кількість потоків, що приймають з’єднання = 22. Оскільки утиліта nmap у розширеному скані портів, буде відправляти у декількох підключеннях з різних портів деякі пакети, і аналізувати відповіді сервера на ці пакети.

На «допоміжній машині» запускаємо скрипт, представлений на рисунку 3.17.

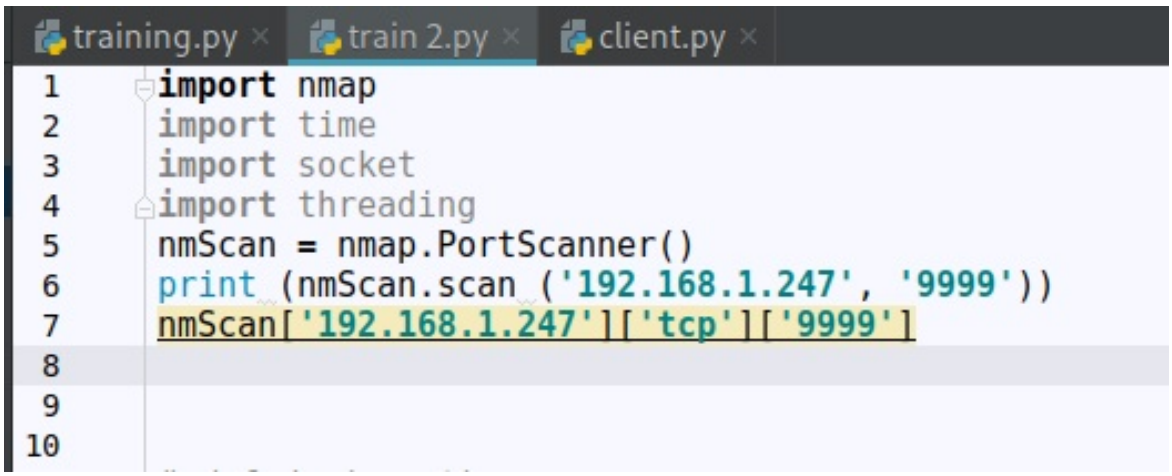

Рисунок 3.17 – Скрипт на розширене сканування порту.

Результат на допоміжній машині у форматі JSON:

{'nmap': {'command_line': 'nmap -oX - -p 9999 -sV 192.168.1.247', 'scaninfo': {'tcp': {'method': 'syn', 'services': '9999'}}, 'scanstats': {'timestr': 'Sat Dec 1 21:27:40 2018', 'elapsed': '122.48', 'uphosts': '1', 'downhosts': '0', 'totalhosts': '1'}}, 'scan': {'192.168.1.247': {'hostnames': [{'name': 'DESKTOP-9UV4', 'type': 'PTR'}], 'addresses': {'ipv4': '192.168.1.247', 'mac': 'B0:8E:8F:34:E3:64'}, 'vendor': {'B0:8E:8F:34:E3:64': 'Asustek Computer'}, 'status': {'state': 'up', 'reason': 'arp-response'}, 'tcp': {9999: {'state': 'open', 'reason': 'syn-ack', 'name': 'abyss', 'product': '', 'version': '', 'extrainfo': '', 'conf': '3', 'cpe': ''}}}}}

Тобто потенціальний зловмисник зміг би отримати досить багато інформації про сервер.

На основній машині маємо наступний результат логування (рисунок 3.18).

Виходячи з цих даних, наприклад, рядок 10, можна зрозуміти, що зловмисник намагається просканувати порти сервера, що може означати атаку в найближчий час.

Тому прочитавши вчасно логи, адміністратор зможе прийняти захисні міри.

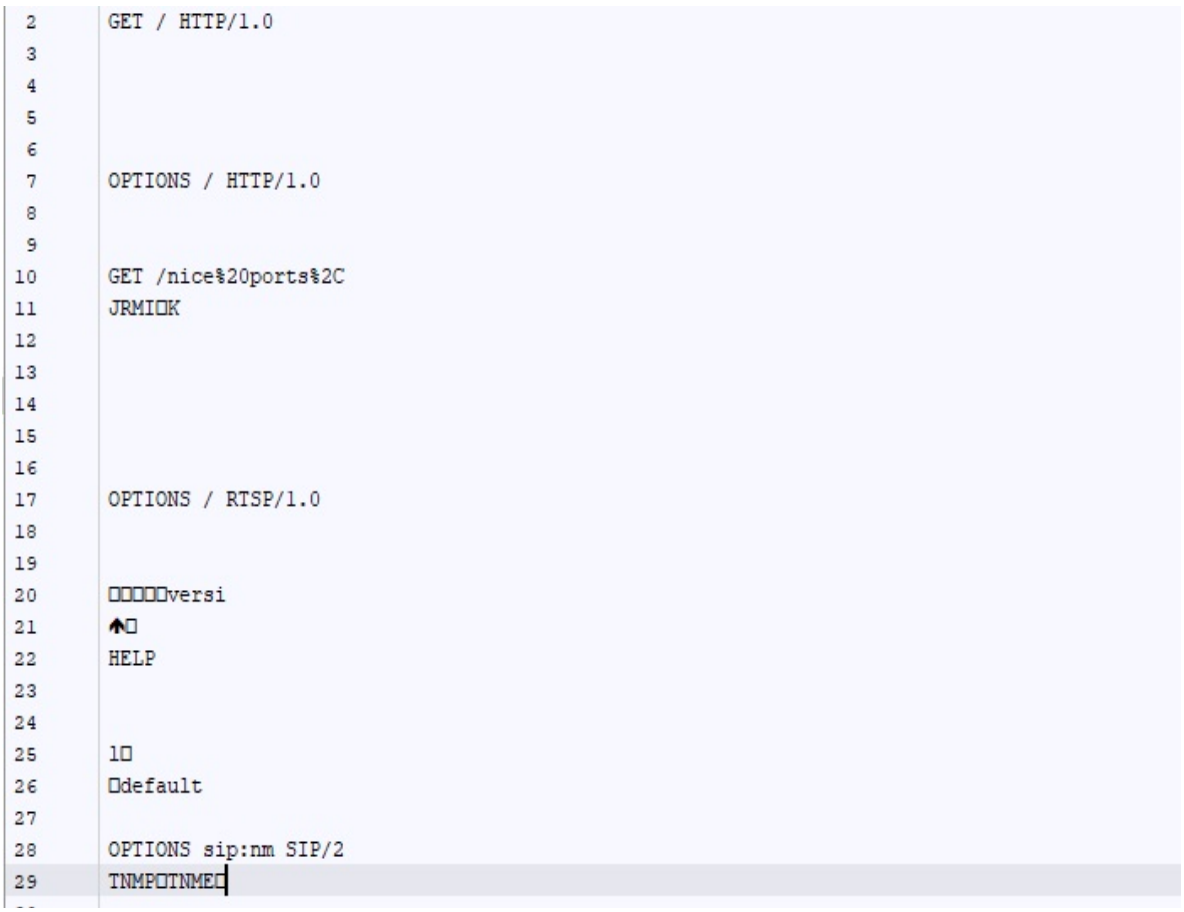

Рисунок 3.18 - Фрагмент даних отриманий з клієнта

Висновки до розділу 3

Розроблено комплексний спосіб та систему захисту ресурсів комп'ютерної мережі, яка відрізняється від наявних тим, що в мережі встановлено централізований контролер, який проводить моніторинг мережі, а також елементом захисту розробленої системи є набір імітаційних атак в поєднані з загальним інтерфейсом контролювання мережевих функцій, за допомогою тестових атак, система та мережевий адміністратор навчаться розрізняти сигнатури схожих атак.

Змодельовано роботу тестової мережі з використанням віртуальних функцій, проведено на неї імітації атак та показано як система реагує на дані атаки і як проводиться логування під час атаки. Спираючись на проведені дослідження в перших розділах в тестуванні були використані наступні найпоширеніші атаки на віртуальні функції:

- сканування портів;
- атака з використанням переповнення буферу;
- DoS атака.

Для уникнення атаки сканування портів, мережевому адміністратору необхідно ретельно і періодично перевіряти логування з активних та завершених підключень, в такому випадку можна встигнути виконати захисні дії від подальших загроз.

Для уникнення атак на переповнення буферу найефективнішим варіантом є написання чистого коду з дотриманням правил (наприклад, якщо з'єднання відкрили, до кінця виконання програми його необхідно обов'язково закрити). Також використання обмеження на кількість даних, що може бути прийнята буфером серверу, інші дані повинні бути відкинуті.

Для уникнення DoS необхідно чітко та ретельно контролювати на сервері чергу очікуючих підключень – server.listen(), а також можливо збільшити кількість потоків, що обробляють з'єднання з черги, також хорошим способом захисту буде створення надійної політики безпеки, згідно якої підозрілі відправники будуть блокуватися.

Отже підбиваючи підсумки, можна зазначити, що система справляється з найпоширенішими сигнатурами атак. Тому використання тестових атак на проникнення в розроблених системах захисту є актуальним способом перевірки, чи зможе система протистояти справжній схожій атаці. Після виконання тестових ми можемо побачити як приблизно система буде реагувати на схожі відхилення в роботі. Тому в разі справжньої атаки мережевий адміністратор може бути готовий до атаки, і взагалі система може сама її знешкодити, якщо на основі тестових атак були зроблені висновки і розроблені захисні інструменти.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

В магістерської дисертації були вирішені наступні задачі:

Розглянуто загальні принципи та архітектура технології програмно-конфігуровних мереж, принципи проектування та побудови мереж з технологією SDN (програмно-конфігуровні мережі). Також проаналізована технологія NFV (віртуалізовані мережеві функції), розглянуто її повний архітектурний шаблон, запропонований інститутом ETSI. Детально проаналізована робота кожного елементу та його зв'язків в мережі.

Проведено аналіз вразливостей та методів захисту моделей побудови віртуальної мережі (віртуалізована мережева інфраструктура як сервіс, віртуалізована мережева платформа як сервіс та віртуалізована мережева функція як сервіс). Показано що кожна з моделей потребує свого унікального захисту, на основі проведеного аналізу сформовані основні рекомендації щодо захисту ресурсів при використані цих моделей.

Розроблено комплексний спосіб та систему захисту ресурсів комп'ютерної мережі, з встановленим централізованим контролером, який проводить моніторинг мережі, також елементом захисту розробленої системи є набір імітаційних атак, що допоможе навчитися розрізняти сигнатури схожих атак та швидко реагувати на них.

Змодельовано роботу тестової мережі з використанням віртуальних функцій, проведено на неї імітації атак та показано як система реагує на дані атаки і як проводиться логування під час атаки.

Список літературних джерел

1. SDN // [Електронний ресурс]. — Режим доступу: http://it-ua.info/news/2015/03/12/merezhev-tehnolog-sdn-software-defined-networking.html

2. Програмно керовані мережі SDN // [Електронний ресурс]. — Режим доступу: http://www.sibis.com.ua/ua/services/corporatenetworks/intelligent-network-iwan-sd-wan/

3. Overview of RFC 7426: SDN Layers and Architecture Terminology // [Електронний ресурс]. — Режим доступу: https://sdn.ieee.org/newsletter/september-2017/overview-of-rfc7426-sdn-layers-and-architecture-terminology

4. 7 Advantages of Software Defined Networking [Електронний ресурс]. — Режим доступу: http://www.ingrammicroadvisor.com/data-center/7-advantages-of-software-defined-networking

5. Security position paper Network Function Virtualization [Електронний ресурс]. – Режим доступу: https://cloudsecurityalliance.org/download/security-position-paper-network-function-virtualization/

6. Руководство по SDN и NFV [Електронний ресурс]. — Режим доступу: https://shalaginov.com/2018/01/16/%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE-%D0%BF%D0%BE-sdn-%D0%B8-nfv-1/

7. Shankar L. NFV: Security Threats and Best Practices / L. Shankar, T. Tarik, D. Ashutosh., 2017.

8. Protecting SDN and NFV networks from Cyber Security Vulnerabilities [Електронний ресурс] // Telco System Public Information. – 2015.

9. Balamurali T. Network function virtualization for dummies / Thekkedath Balamurali., 2016. – 37 с.

10. SDN-NFV Reference Architecture [Електронний ресурс]. — Режим доступу: http://innovation.verizon.com/content/dam/vic/PDF/Verizon_SDN-NFV_Reference_Architecture.pdf

11. Zonghua Z. Security in Network Functions Virtualization / Zonghua Zhang, Ahmed Meddahi / Elsevier. London 2017.

12. A SDN and NFV use-case: NDN implementation and security monitoring [Електронний ресурс]. — Режим доступу: https://hal.inria.fr/hal-01652639/document.

13. Порівняльний аналіз основних проблем безпеки в комп'ютерних мережах SDN при використані технології NFV: матеріали 11-ої міжнар. наук.-техн. конф. ПМК'2018, 14-16 листопад 2018, Київ, Україна / НТУУ «КПІ», Ф-т прикладної матем. – К., 2018. –Парал. тит. арк. англ.

14. Система захисту мережі SDN з використанням технології NFV: V Міжнародна науково-технічна Internet-конференція «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами» НУХТ, 22-23 листопада, Київ. – К., 2018.