**ОСВІТИ І НАУКИ УКРАЇНИ**

**Національний університет кораблебудування імені адмірала Макарова**

Навчально-науковий інститут автоматики і електротехніки

Кафедра комп'ютерних технологій та інформаційної безпеки

«ДОПУЩЕНИЙ ДО ЗАХИСТУ»
Завідувач кафедри комп'ютерних технологій та інформаційної безпеки, к.т.н., доцент
_____ С.М. Нужний
«17» ____06____ 2026р.

**КВАЛІФІКАЦІЙНА РОБОТА**

на здобуття ступеня вищої освіти «бакалавр»

# РОЗРОБКА КОМПЛЕКСНОЇ СИСТЕМИ ЗАХИСТУ КОРПОРАТИВНОЇ МЕРЕЖІ НА ОСНОВІ ТЕХНОЛОГІЙ СЕГМЕНТАЦІЇ ТА АКТИВНОГО МОНІТОРИНГУ ТРАФІКУ

Галузь знань: 12 «Інформаційні технології»
Спеціальність: 125 «Кібербезпека»
Освітньо-професійна програма: «Системи технічного захисту інформації, автоматизація її обробки»

Виконав: студент 4381 групи
Висоцький Марк Костянтинович
(прізвище та ініціали) (підпис)

Керівник роботи: ст. викладач кафедри комп'ютерних технологій та інформаційної безпеки
Трибулькевич С.Л.
(прізвище та ініціали) (підпис)

Миколаїв – 2026

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Національний університет кораблебудування імені адмірала Макарова**

Навчально-науковий інститут автоматики та електротехніки

Кафедра комп'ютерних технологій та інформаційної безпеки

Спеціальність **125«Кібербезпека»**
Освітня програма **«Системи технічного захисту інформації, автоматизація її обробки»**

ЗАТВЕРДЖУЮ
Гарант освітньої програми, к.т.н.,
доцент, доцент кафедри КТІБ
_____ С.М. Нужний
«10» 02 2026р.

***ЗАВДАННЯ***
***НА КВАЛІФІКАЦІЙНУ РОБОТУ***
**на здобуття ступеня вищої освіти «бакалавр»**

Студенту Висоцькому Марку Костянтиновичу
(Прізвище, ім'я, по батькові)

1. Тема роботи: Розробка комплексної системи захисту корпоративної мережі на основі технологій сегментації та активного моніторингу трафіку.

Керівник роботи ст. викладач кафедри комп'ютерних технологій та інформаційної безпеки, Трибулькевич Сергій Леонідович

Затверджені наказом ректора №286-г від 22.04.2026

2. Термін подання роботи: 5.06.2026
3. Вихідні дані по роботі: Нормативно-правова база і теоретичні основи побудови захищених комп'ютерних мереж.

4. Перелік питань, що належать до розробки (найменування розділів):

1. Провести аналіз актуальних кіберзагроз інформаційній безпеці корпоративних мереж та дослідити еволюцію засобів мережевого захисту.

2. Дослідити теоретичні основи та принципи функціонування технологій логічної сегментації (VLAN), апаратної фільтрації (Extended ACL) та систем глибокого аналізу пакетів (IDS Suricata).

3. Спроєктувати та практично розгорнути ієрархічну архітектуру захищеної мережі у середовищі апаратної емуляції PNETLab.

4. Інтегрувати систему моніторингу трафіку та провести експериментальне тестування ефективності розробленого комплексу шляхом симуляції цілеспрямованих кібератак.

5. Перелік презентаційних матеріалів:. слайди в якості результатів дослідження у відповідності до переліку питань, що належать до розробки.

6. Консультант - внутрішній рецензент роботи

| Прізвище, ініціали та посада | Дата подання роботи на рецензію | Дата отримання рецензії | Підпис |
|---|---|---|---|
| Передерій В.І., професор, д.т.н. | 2.06.26 | 5.06.26 | |

7. Дата видачі завдання 10.02.2026

**КАЛЕНДАРНИЙ ПЛАН**

| № | Назва етапів роботи | Терміни виконання | Примітка |
|---|---|---|---|
| 1 | Вибір теми, визначення мети, завдань, об'єкта та предмета дослідження | 10.02.2026 | |
| 2 | Попередній варіант оглядових розділів, підбір і опрацювання списку використаних джерел | 20.03.2026 | |
| 3 | Попередній варіант повної КР | 15.04.2026 | |
| 4 | Складання ЄДКІ зі спеціальності на ступінь «бакалавра» | 16.04.2026<br>14.05.2026 | |
| 5 | Попередній варіант презентації | 29.05.2026 | |
| 6 | Подання на кафедру КТІБ тексту остаточного варіанту роботи, підписаного її керівником в електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020)<br>Отримання внутрішньої рецензії | 05.06.<br>2026 | не пізніше, ніж за 14 діб до захисту |
| 7 | Отримання висновку керівника;<br>Отримання висновку кафедри щодо наявності / відсутності плагіату в роботі<br>Попередній захист роботи на засіданні кафедри КТІБ<br>Висновок кафедри про КР, допуск до захисту | 10.06.2026 | |
| 8 | Отримання зовнішньої рецензії | 15.06.2026 | |
| 9 | Підготовка презентації та доповіді | 17.06.2026 | |
| 10 | Подання на кафедру КТІБ:<br>– друкованого і зшитого варіанту пояснювальної записки до кваліфікаційної роботи та/або (за рішенням кафедри) електронного варіанту в форматі *.pdf<br>– друкованого варіанту презентації (в 5-ти екземплярах в разі захисту оф-лайн, в одному екземплярі в разі захисту он-лайн (за рішенням кафедри)) та електронного варіанту презентації;<br>– зовнішньої рецензії;<br>– документів щодо відсутності плагіату і передачі авторських прав;<br>– файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.). | 22.06.2026 | |
| 11 | Захист кваліфікаційної роботи | 24 - 25.<br>06.2026 | |

**Студент** _____ М.К. Висоцький
(підпис) (ПІБ)

**Керівник роботи** _____ С.Л. Трибулькевич
(підпис) (ПІБ)

# АНОТАЦІЯ

*Висоцький М.К.* Розробка комплексної системи захисту корпоративної мережі на основі технологій сегментації та активного моніторингу трафіку. – Кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота на здобуття ступеня вищої освіти «бакалавр» за спеціальністю F5 «Кібербезпека та захист інформації» галузі знань *F* «Інформаційні технології», освітньо-професійна програма «Системи технічного захисту інформації, автоматизація її обробки». – Національний університет кораблебудування імені адмірала Макарова, Миколаїв, 2026.

Пояснювальна записка: 73 с., 10 рис., 1 табл., 21 посилання. 1 додаток

Кваліфікаційна робота присвячена актуальній науково-практичній задачі – побудові надійної багаторівневої системи захисту корпоративних комп'ютерних мереж від сучасних кіберзагроз. Об'єктом дослідження є процес забезпечення безпеки інформаційної інфраструктури корпоративної мережі. Предметом дослідження є технології логічної сегментації (*VLAN*), апаратної фільтрації (*ACL*) та системи глибокого аналізу і виявлення вторгнень (*IDS*). Методами дослідження є системний аналіз, ієрархічне мережеве проєктування, апаратна емуляція та практичне тестування (симуляція кібератак). Метою роботи є проєктування, практична реалізація та експериментальна верифікація комплексної системи захисту корпоративної мережі, що поєднує превентивні бар'єри та засоби активного моніторингу трафіку в середовищі емуляції *PNETLab*.

В роботі проведено аналіз архітектурних вразливостей пласких мережевих топологій та класифікацію сучасних мережевих загроз (*ARP-spoofing*, *TCP SYN Scan*, *ICMP*/*TCP Flood*). Спроєктовано ієрархічну модель мережі на базі згорнутого ядра (*Collapsed Core*). Практично реалізовано логічну сегментацію інфраструктури за допомогою *VLAN* та налаштовано міжсегментну фільтрацію

на основі розширених списків доступу (*Extended ACL*). Інтегровано систему виявлення вторгнень (*IDS Suricata*) з використанням технології дзеркалювання портів (*SPAN*) у нерозбірливому режимі (*Promiscuous mode*) для глибокого аналізу пакетів (*DPI*). Успішно проведено експериментальне тестування розробленої системи шляхом симуляції *DoS*-атак та мережевої розвідки, що підтвердило високу ефективність кастомних сигнатур та апаратних політик доступу..

Практична значимість одержаних результатів полягає в тому, що запропонована архітектура та конфігурації обладнання можуть бути безпосередньо використані для захисту реальних корпоративних мереж малого та середнього бізнесу, а також впроваджені в навчальний процес зі спеціальності F5 «Кібербезпека та захист інформації».

**Ключові слова**: кібербезпека, корпоративна мережа, сегментація мережі, *VLAN*, *ACL*, моніторинг трафіку, система виявлення вторгнень, *IDS*, *Suricata*, *PNETLab*.

## *ANNOTATION*

*Vysotskyi M.K.* Development of a comprehensive corporate network protection system based on segmentation technologies and active traffic monitoring. – Qualification work as a manuscript.

Qualification work for a higher education degree "Bachelor" in specialty F5 "Cybersecurity and Information Protection", field of knowledge F "Information Technologies", educational and professional program "Systems of technical protection of information, automation of its processing". – Admiral Makarov National University of Shipbuilding, Mykolaiv, 2026.

Explanatory note: 73 pp., 10 figures, 1 table, 21 references, 1 appendix.

The qualification work is devoted to a highly relevant scientific and practical task – building a reliable multi-level protection system for corporate computer networks against modern cyber threats. The object of the research is the process of ensuring the security of the corporate network's information infrastructure. The subject of the research is logical segmentation technologies (*VLAN*), hardware filtering (*ACL*), and deep packet inspection intrusion detection systems (*IDS*). The research methods are system analysis, hierarchical network design, hardware emulation, and practical testing (cyberattack simulation). The purpose of the work is the design, practical implementation, and experimental verification of a comprehensive corporate network protection system that combines preventive barriers and active traffic monitoring tools in the *PNETLab* emulation environment.

The thesis analyzes the architectural vulnerabilities of flat network topologies and classifies modern network threats (A*RP-spoofing, TCP SYN Scan, ICMP/TCP Flood*). A hierarchical network model based on a Collapsed Core architecture was designed. Logical segmentation of the infrastructure was practically implemented using *VLANs*, and inter-segment filtering was configured using Extended Access Control Lists (*ACL*). An intrusion detection system (*IDS Suricata*) was integrated using port mirroring technology (SPAN) in promiscuous mode for deep packet inspection (*DPI*). Experimental testing of the developed system was successfully conducted by

simulating *DoS* attacks and network reconnaissance, confirming the high efficiency of custom signatures and hardware access policies.

The practical significance of the obtained results lies in the fact that the proposed architecture and equipment configurations can be directly used to protect real small and medium-sized business corporate networks, as well as implemented in the educational process for specialty F5 "Cybersecurity and Information Protection".

Keywords: cybersecurity, corporate network, network segmentation, *VLAN*, *ACL*, traffic monitoring, intrusion detection system, *IDS*, *Suricata*, *PNETLab*.

# ЗМІСТ

# ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення

ЦОД – центр обробки даних

*ACL* (*Access Control List*) – список контролю доступу

*ARP* (*Address Resolution Protocol*) – протокол розв'язання адрес

*ASIC* (*Application-Specific Integrated Circuit*) – спеціалізована інтегральна схема

*CPU* (*Central Processing Unit*) – центральний процесор

*DDoS* (*Distributed Denial of Service*) – розподілена атака на відмову в обслуговуванні

*DHCP* (*Dynamic Host Configuration Protocol*) – протокол динамічної конфігурації мережевого вузла

*DNS* (*Domain Name System*) – система доменних імен

*DoS* (*Denial of Service*) – атака на відмову в обслуговуванні

*DPI* (*Deep Packet Inspection*) – технологія глибокого аналізу пакетів

*EIGRP* (*Enhanced Interior Gateway Routing Protocol*) – розширений протокол внутрішньої маршрутизації

*ELK* (*Elasticsearch*, *Logstash*, *Kibana*) – стек систем управління та аналітики даних

*ERSPAN* (*Encapsulated Remote Switched Port Analyzer*) – інкапсульований віддалений аналізатор комутованих портів

*HTTP* / *HTTPS* (*HyperText Transfer Protocol* / *Secure*) – протокол передачі гіпертексту / захищений

*ICMP* (*Internet Control Message Protocol*) – протокол керуючих сповіщень Інтернету

*IDS* (*Intrusion Detection System*) – система виявлення вторгнень

*IEEE* (*Institute of Electrical and Electronics Engineers*) – Інститут інженерів електротехніки та електроніки

*IETF* (*Internet Engineering Task Force*) – Інженерна рада Інтернету

*IOL* (*IOS on Linux*) – мережева операційна система маршрутизації для середовищ емуляції

*IP* (*Internet Protocol*) – міжмережевий протокол

*IPS* (*Intrusion Prevention System*) – система запобігання вторгненням

*JSON* (*JavaScript Object Notation*) – текстовий формат обміну структурованими даними

*L2, L3, L4, L7* (*Layer* 2, 3, 4, 7) – рівні еталонної моделі *OSI* (канальний, мережевий, транспортний, прикладний відповідно)

*LACP* (*Link Aggregation Control Protocol*) – протокол керування агрегацією каналів

*MAC* (*Media Access Control*) – управління доступом до середовища (апаратна адреса пристрою)

*NAT* (*Network Address Translation*) – трансляція мережевих адрес

*NIC* (*Network Interface Card*) – мережевий інтерфейс (контролер)

*NIDS* (*Network Intrusion Detection System*) – мережева система виявлення вторгнень

*OSI* (*Open Systems Interconnection*) – еталонна модель взаємодії відкритих систем

*OSPF* (*Open Shortest Path First*) – протокол динамічної маршрутизації

*PNETLab* (*Packet Network Emulator Tool Lab*) – віртуальне середовище апаратно-програмної емуляції мереж

*QoS* (*Quality of Service*) – якість обслуговування

*RAM* (*Random Access Memory*) – оперативна пам'ять

*RFC* (*Request for Comments*) – серія документів, що описують технічні стандарти Інтернету

*SIEM* (*Security Information and Event Management*) – система централізованого управління інформацією та подіями безпеки

*SMB* (*Small and Medium Business*) – малий та середній бізнес

*SOC* (*Security Operations Center*) – центр управління кібербезпекою

*SPAN* (*Switched Port Analyzer*) – аналізатор комутованих портів (технологія дзеркалювання трафіку)

*SSH* (*Secure Shell*) – протокол безпечного віддаленого керування

*STP* (*Spanning Tree Protocol*) – протокол остовного дерева

*SVI* (*Switch Virtual Interface*) – віртуальний інтерфейс комутатора

*TCAM* (*Ternary Content-Addressable Memory*) – троїчна асоціативна пам'ять

*TCP* (*Transmission Control Protocol*) – протокол керування передачею

*UDP* (*User Datagram Protocol*) – протокол користувацьких датаграм

*VID* (*VLAN Identifier*) – ідентифікатор віртуальної локальної мережі

*VLAN* (*Virtual Local Area Network*) – віртуальна локальна мережа

# ВСТУП

Зосередження великих обсягів критичної інформації, комерційної таємниці, баз даних та персональних відомостей у корпоративних мережах зумовлює надзвичайну актуальність захисту цих ресурсів від цілеспрямованих і випадкових загроз. Сучасна мережева інфраструктура постійно піддається деструктивним впливам: від масових атак на відмову в обслуговуванні (*DoS*/*DDoS*) та мережевої розвідки до витончених методів перехоплення даних (*ARP-spoofing*) і латерального переміщення зловмисників усередині периметра. Практика показує, що використання традиційних "пласких" мережевих топологій та застосування виключно базових міжмережевих екранів вже не забезпечує достатнього рівня безпеки. Тому розробка комплексних, багаторівневих систем захисту, які поєднують превентивні архітектурні бар'єри та засоби проактивного виявлення вторгнень, є одним із найпріоритетніших завдань для сучасних підприємств.

Розробці методів та засобів захисту корпоративних мереж присвячено багато наукових робіт вітчизняних і зарубіжних науковців. На даний час на IT-ринку наявний широкий спектр мережевого обладнання для логічної сегментації та фільтрації трафіку (*VLAN*, *ACL*), а також потужних систем виявлення вторгнень (*IDS*/*IPS*). Проте, системною проблемою багатьох інфраструктур залишається відсутність комплексного підходу: засоби сегментації та моніторингу часто працюють ізольовано, а налаштування систем глибокого аналізу пакетів (*DPI*) ускладнюється специфічними апаратно-програмними артефактами (наприклад, технологією *TCP Checksum Offloading*). На сьогодні існує потреба в адаптованій та перевіреній методиці розгортання таких інтегрованих комплексів на базі еталонних ієрархічних моделей та відкритого аналітичного програмного забезпечення.

Метою даної кваліфікаційної роботи є проєктування, практична реалізація та експериментальна верифікація комплексної системи захисту корпоративної

мережі на основі інтеграції технологій логічної сегментації, апаратної фільтрації та активного моніторингу трафіку.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- провести аналіз актуальних загроз інформаційній безпеці корпоративних мереж та дослідити еволюцію базових засобів мережевого захисту;

- дослідити теоретичні основи та розробити масштабовану ієрархічну архітектуру мережі з використанням технологій логічної сегментації (*VLAN*) і списків контролю доступу (*Extended ACL*);

- інтегрувати в мережеву інфраструктуру систему виявлення вторгнень (*IDS Suricata*) на базі технології дзеркалювання портів (*SPAN*) для здійснення глибокого аналізу пакетів (*DPI*);

- здійснити практичне розгортання розробленої системи у середовищі апаратної емуляції *PNETLab* та провести її експериментальне тестування шляхом симуляції типових кібератак.

Об’єктом дослідження в даній роботі є процес забезпечення безпеки інформаційної інфраструктури корпоративної мережі.

Предметом дослідження в даній роботі є технології логічної сегментації мережі (*VLAN*), апаратної фільтрації (*ACL*) та системи глибокого аналізу і виявлення вторгнень (*IDS*).

Методами дослідження є системний аналіз (для класифікації загроз та архітектурних принципів), ієрархічне мережеве проєктування, метод апаратно-програмної емуляції мережевих інфраструктур, а також емпіричні методи практичного тестування та експериментальної верифікації (симуляція кібератак).

Наукова новизна даної кваліфікаційної роботи полягає в удосконаленні підходу до комплексної протидії сучасним мережевим кіберзагрозам шляхом глибинної інтеграції превентивних механізмів канального (*L2*) і мережевого (*L3*) рівнів із системами глибокого аналізу пакетів (*DPI*) транспортного та прикладного рівнів (*L4-L7*), з урахуванням особливостей обробки протокольних аномалій у віртуалізованих комутаційних середовищах.

Практична значимість кваліфікаційної роботи полягає у тому, що розроблена архітектура «згорнутого ядра», конфігурації комутаційного обладнання (*Cisco IOL*) та налаштовані кастомні сигнатури системи моніторингу *Suricata* можуть бути безпосередньо впроваджені в *IT*-інфраструктуру реальних підприємств малого та середнього бізнесу для підвищення загального рівня їхньої кіберстійкості.

Результати роботи можуть бути застосовані у галузі проєктування захищених корпоративних мереж та розгортання центрів управління безпекою (*SOC*), а також у навчальному процесі зі спеціальності F5 «Кібербезпека та захист інформації».

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА АКТУАЛЬНИХ ЗАГРОЗ КОРПОРАТИВНИХ МЕРЕЖ

## 1.1 Архітектура сучасних корпоративних мереж

Сучасна корпоративна комп'ютерна мережа є складним, високотехнологічним інфраструктурним комплексом, який забезпечує єдине комунікаційне середовище для функціонування бізнес-процесів організації [1]. Вона об’єднує різнорідні технічні засоби – робочі станції користувачів, серверні потужності, периферійне обладнання та системи автоматизації – задля забезпечення спільного доступу до інформаційних ресурсів, корпоративних баз даних та зовнішньої мережі Інтернет. Ефективне проєктування такої інфраструктури вимагає суворого дотримання архітектурних принципів, оскільки структурні помилки, допущені на етапі планування топології, практично неможливо компенсувати програмними засобами захисту на вищих рівнях еталонної моделі взаємодії відкритих систем (*OSI*) [1, 18]. Побудова надійної та безпечної мережі повинна підпорядковуватися чітким закономірностям, які визначають процеси масштабованості, керованості та прогнозованості руху інформаційних потоків.

На початкових етапах автоматизації підприємств або в умовах обмеженого бюджету поширеним інженерним рішенням є розгортання однорангової, або так званої «пласкої» (*flat*) мережевої топології. У такій конфігурації всі кінцеві пристрої, незалежно від їхнього функціонального призначення, функціонують у межах одного адресного простору та єдиного широкомовного домену (*Broadcast Domain*) другого рівня моделі *OSI* [18]. З інженерного погляду даний підхід є неефективним через суттєві обмеження у масштабованості. Головна проблема полягає в механізмі обробки службового трафіку: будь-який широкомовний кадр, пов'язаний із розв'язанням *IP*-адрес через протокол *ARP* або динамічним отриманням конфігурацій через *DHCP*, безумовно копіюється комутаторами на всі активні порти [12, 18]. Коли кількість хостів у такому сегменті перевищує

критичну межу, обсяг службового трафіку починає зростати експоненціально, викликаючи явище широкомовного шторму (*Broadcast Storm*). Це призводить до критичного зниження пропускної здатності каналів зв'язку та перевантаження центральних процесорів кінцевих вузлів, змушених безперервно обробляти нецільові пакети.

З погляду інформаційної безпеки пласка архітектура несе в собі критичні ризики через повну відсутність внутрішнього периметра та механізмів стримування загроз [11]. У такому середовищі неможливо впровадити принцип найменших привілеїв або гранульовані політики розмежування доступу між різними структурними підрозділами компанії. Якщо зловмисник успішно компрометує хоча б одну робочу станцію, він миттєво отримує пряму видимість на канальному рівні (*L2*) до всієї внутрішньої інфраструктури [1]. Латеральное переміщення (*Lateral Movement*) всередині такої мережі здійснюється безперешкодно, оскільки трафік між вузлами не проходить через жодну внутрішню точку контролю чи фільтрації [15]. Це дозволяє зловмиснику, діючи зсередини периметра, вільно атакувати критично важливі сервери баз даних, бухгалтерські системи та сховища резервних копій.

Для подолання обмежень пласких тополій у світовій практиці використовується модульний підхід до проєктування, заснований на класичній трирівневій ієрархічній моделі комп'ютерних мереж, описаній у фундаментальних посібниках *Cisco Press* [1, 15]. Дана концепція передбачає розділення мережі на три логічні рівні: рівень доступу (*Access Layer*), рівень розподілу (*Distribution Layer*) та рівень ядра (*Core Layer*). Кожен із цих прошарків виконує строго визначений набір функцій, що дозволяє оптимізувати обробку пакетів, спростити процеси адміністрування та чітко розмежувати зони відповідальності засобів захисту інформації [11].

Нижній ешелон ієрархії представляє рівень доступу, який є безпосередньою межею мережі, до якої підключається все кінцеве клієнтське обладнання. Комутатори рівня доступу забезпечують фізичне підключення пристроїв та реалізують превентивні політики безпеки на межі інфраструктури.

Саме на цьому етапі відбувається первинна логічна сегментація шляхом групування фізичних портів у різні віртуальні локальні мережі (*VLAN*) [5]. Окрім комутації кадрів, на рівні доступу розгортаються механізми захисту від внутрішніх атак, такі як *Port Security*, контроль доступу на основі легітимних *MAC*-адрес, автентифікація пристроїв за стандартом *IEEE* 802.1X, а також первинне маркування трафіку для забезпечення якості обслуговування (*QoS*) [5,6].

Інтелектуальною сполучною ланкою між периферією та магістраллю виступає рівень розподілу, головною функцією якого є агрегація висхідних каналів зв'язку та забезпечення маршрутизації трафіку між різними віртуальними мережами (*Inter-VLAN Routing*) [18]. Рівень розподілу є чіткою межею між широкомовними доменами *L2* та доменами маршрутизації *L3*. Тут реалізується основний комплекс аналітичних завдань безпеки: застосовуються розширені списки контролю доступу (*Extended ACL*) для міжсегментної фільтрації, налаштовуються міжмережеві екрани, визначаються межі довіри та здійснюється початковий моніторинг аномальної активності [10].

Центральним елементом усієї інфраструктури є рівень ядра, який проектується за принципом максимальної продуктивності та абсолютної відмовостійкості. Єдине завдання обладнання цього рівня – максимально швидке пересилання великих обсягів пакетів між різними модулями рівня розподілу без жодних затримок [1]. На магістральних комутаторах ядра категорично заборонено виконувати складні логічні маніпуляції з пакетами, такі як глибока фільтрація, інспекція вмісту на рівні застосунків чи трансляція адрес (*NAT*), оскільки покрокова перевірка заголовків створює обчислювальні затримки. Обладнання ядра базується на спеціалізованих апаратних мікросхемах (*ASIC*) і орієнтоване виключно на швидкісний транспорт даних.

У мережевій інфраструктурі малих та середніх підприємств розгортання повної трирівневої структури часто є економічно недоцільним через високу вартість великої кількості апаратних засобів. Для оптимізації витрат без втрати переваг ієрархічного проєктування застосовується модифікована дворівнева

архітектура «згорнутого ядра» (*Collapsed Core Architecture*) [15]. У цьому дизайні функції магістрального ядра (*Core*) та інтелектуального рівня розподілу (*Distribution*) об'єднуються і виконуються на одному консолідованому пристрої високої продуктивності – багатошаровому комутаторі L3 або потужному центральному маршрутизаторі, до якого безпосередньо підключаються периферійні комутатори рівня доступу.

Саме архітектурна модель згорнутого ядра була обрана як концептуальна основа для побудови захищеної корпоративної мережі в межах даного дослідження у віртуальному середовищі *PNETLab* [7, 17]. Центральний комутатор *Core-Switch* виконує функції згорнутого ядра: він агрегує канали від периферії, забезпечує високошвидкісний обмін даними та виступає точкою термінації віртуальних інтерфейсів (*SVI*), де пролягають логічні межі між сегментами підприємства. Комутатор *Access-Switch* виконує класичну роль рівня доступу, розподіляючи кінцевих користувачів за відповідними доменами безпеки (*VLAN* 10 – *Management*, *VLAN* 20 – *Users*, *VLAN* 30 – *Servers*).

Класичним інженерним рішенням для реалізації міжвланової маршрутизації в умовах використання реального або емульованого обладнання є схема «маршрутизатор на паличці» (*Router-on-a-Stick*). При такій конфігурації один фізичний інтерфейс прикордонного маршрутизатора *Edge-Router* з'єднується з магістральним портом (*Trunk*) комутатора ядра. На маршрутизаторі цей фізичний інтерфейс логічно розділяється на кілька віртуальних підінтерфейсів (*Subinterfaces*), кожен з яких налаштовується на роботу з конкретним ідентифікатором тегу стандарту *IEEE* 802.1*Q* та отримує власну *IP*-адресу [6]. Ця адреса стає шлюзом за замовчуванням для пристроїв відповідного *VLAN*. Така схема забезпечує жорсткий контроль над міжсегментними потоками, оскільки будь-який пакет, що прямує з одного логічного відділу до іншого, змушений пройти через підінтерфейс маршрутизатора, де до нього застосовуються централізовані правила апаратної фільтрації [10, 19].

Надійність функціонування корпоративної мережі безпосередньо корелює з її загальним рівнем захищеності, оскільки відмова будь-якого критичного вузла

або каналу зв'язку призводить до порушення доступності інформаційних сервісів. Для запобігання появі єдиної точки відмови при проєктуванні мереж впроваджуються механізми надлишковості на фізичному та канальному рівнях за допомогою резервних лінків та технологій агрегації каналів (*LACP*), які дозволяють об'єднувати кілька фізичних інтерфейсів в один логічний канал (*EtherChannel*). Для запобігання виникненню комутаційних петель у надлишкових топологіях другого рівня обов'язково розгортаються протоколи сімейства *Spanning Tree* (*STP* / *Rapid STP*).

На рівні маршрутизації у великих мережах зазвичай впроваджуються протоколи динамічного обміну маршрутною інформацією (*OSPF*, *EIGRP*), які автоматично перераховують маршрути слідування пакетів у разі аварій [18]. Проте в межах розробленого лабораторного стенду для забезпечення взаємодії між сегментами та виходу в зовнішню мережу застосовано статичну маршрутизацію (маршрути за замовчуванням). Такий підхід є цілком виправданим та безпечним для інфраструктур малого масштабу. На відміну від динамічних протоколів, які потребують додаткових обчислювальних ресурсів та самі по собі можуть виступати вектором кібератак шляхом ін'єкції підроблених маршрутів, статична конфігурація повністю виключає подібні вразливості, забезпечуючи максимальну прогнозованість і стабільність руху інформаційних потоків [1]. Таким чином, аналіз архітектурних особливостей є першим і необхідним кроком перед розробкою комплексної системи захисту.

## 1.2 Основні загрози для корпоративної мережі

Функціонування будь-якої сучасної корпоративної комп'ютерної мережі відбувається в умовах безперервного впливу агресивного зовнішнього та внутрішнього інформаційного середовища [1]. Забезпечення належного рівня безпеки мережевої інфраструктури неможливе без чіткої класифікації та систематизації потенційних загроз, оскільки архітектура засобів захисту повинна безпосередньо відповідати специфіці та природі атакуючих векторів [11].

Корпоративне середовище піддається деструктивним впливам на всіх рівнях еталонної моделі взаємодії відкритих систем (*OSI*), проте найбільш критичними для цілісності, конфіденційності та доступності інформаційних ресурсів є загрози, що експлуатують вразливості протоколів канального (*L2*) та мережевого (L3) рівнів. Саме на цих етапах формуються базові механізми передачі даних, такі як комутація кадрів *Ethernet*, динамічне розв'язання адрес через протокол *ARP*, встановлення сесій транспортного рівня та маршрутизація пакетів між логічними сегментами підприємства [18]. У слабо сегментованих або однорангових мережах реалізація загроз на цих рівнях здатна призвести до повного паралічу бізнес-процесів, несанкціонованого перехоплення конфіденційної інформації або тотального руйнування систем керування [5].

Клас загроз, орієнтованих на канальний рівень (*L2*) моделі *OSI*, є надзвичайно небезпечним через те, що базові протоколи локальних мереж історично проєктувалися без урахування механізмів автентифікації пристроїв [1]. Однією з поширених загроз цього рівня є підміна апаратних адрес, відома як *MAC-spoofing*. Суть даної загрози полягає в умисній фальсифікації зловмисником унікальної *MAC*-адреси свого мережевого інтерфейсу з метою імітації легітимного вузла мережі [14]. Оскільки комутатори рівня доступу будують свої таблиці пересилання кадрів (*MAC Address Table*) на основі аналізу полів адреси відправника у вхідних фреймах, ін'єкція пакета з підробленою адресою змушує комутатор оновити внутрішню таблицю і пов'язати *MAC*-адресу жертви з фізичним портом зловмисника. Це дозволяє обійти статичні фільтри портів, нейтралізувати політики *Port Security* та перенаправити інформаційні потоки, призначені для легітимного користувача, на компрометований вузол [1].

Ще більш деструктивною загрозою канального рівня є атака на отруєння *ARP*-кешу, відома як *ARP-spoofing* або *ARP-poisoning*. Протокол *ARP*, який відповідає за відображення *IP*-адрес третього рівня на *MAC*-адреси другого рівня у межах одного широкомовного домену, функціонує за принципом абсолютної довіри до отриманих повідомлень [18]. Коли кінцевий вузол надсилає широкомовний запит для пошуку апаратної адреси свого шлюзу за

замовчуванням, зловмисник, перебуваючи в тому самому логічному сегменті, має можливість надіслати підроблену *ARP*-відповідь (*ARP Reply*), у якій пов'язує *IP*-адресу маршрутизатора зі своєю власною *MAC*-адресою [1, 18]. Більше того, архітектура багатьох операційних систем дозволяє приймати та обробляти навіть неініційовані *ARP*-відповіді (*Gratuitous ARP*), що дає змогу зловмиснику масово «отруювати» таблиці комутації всіх хостів у підмережі.

Результатом успішної реалізації *ARP-spoofing* є перехоплення інформаційних каналів та реалізація небезпечної парадигми «людина посередині» (*Man-in-the-Middle*) [14]. Увесь трафік, який легітимні користувачі спрямовують до зовнішніх мереж або внутрішніх серверів, починає безперешкодно проходити через компрометовану робочу станцію зловмисника. Це відкриває можливості для пасивного прослуховування сесій, модифікації даних у реальному часі, викрадення нешифрованих облікових записів або здійснення атак на протоколи криптографічного захисту. В умовах пласкої інфраструктури *ARP-spoofing* стає плацдармом для компрометації суміжних систем, оскільки дозволяє перехоплювати навіть адміністративні сесії керування [5, 12].

На мережевому (L3) та транспортному (*L4*) рівнях моделі *OSI* вектори загроз зміщуються у бік несанкціонованого збору інформації та порушення доступності сервісів підприємства [18]. Будь-якому цілеспрямованому зловмисному втручанню обов'язково передує етап мережевої розвідки та сканування, метою якого є визначення топології мережі, виявлення активних хостів, інвентаризація відкритих портів та ідентифікація версій запущених операційних систем і служб [11]. Класичним інструментом для реалізації цієї загрози є напіввідкрите *TCP*-сканування, відоме як *TCP SYN Scan*. Механізм цієї загрози базується на маніпуляції процесом триетапного узгодження з'єднання (*TCP Three-Way Handshake*) [18]. Зловмисник відправляє на цільовий порт пакет із встановленим прапорцем *SYN*. Якщо порт є відкритим, цільова система повертає пакет *SYN-ACK*, підтверджуючи готовність до встановлення сесії.

Проте зловмисник навмисно перериває процес, надсилаючи у відповідь пакет *RST* (*Reset*) замість фінального *ACK*.

Перевага використання *TCP SYN Scan* для зловмисника полягає у високій швидкості сканування та прихованості, оскільки більшість стандартних операційних систем та серверних додатків фіксують з'єднання у своїх журналах подій лише після повного завершення процедури триетапного узгодження. Окрім сканування портів, мережева розвідка включає використання утиліт трасування маршрутів (*Traceroute*) для визначення кількості проміжних маршрутизаторів та меж внутрішніх сегментів, а також експлуатацію вразливостей *DNS*-серверів підприємства з метою отримання повної картини внутрішнього адресного простору. Повністю заблокувати процеси розвідки неможливо, оскільки вони базуються на стандартних діагностичних функціях мережевих протоколів, проте обмеження кількості інформації, яка може потрапити до зовнішнього периметра, є критично важливим інженерним завданням [11, 14].

Найбільш масовою та небезпечною категорією загроз мережевого рівня є атаки на відмову в обслуговуванні (*Denial of Service*, *DoS*) та їх розподілені модифікації (*Distributed Denial of Service*, *DDoS*). Різниця між ними полягає в масштабі реалізації: *DoS*-атака здійснюється з одного контрольованого джерела, тоді як *DDoS* передбачає одночасну атаку з величезної кількості розподілених вузлів, які зазвичай об'єднуються у зловмисні мережі – ботнети. Наслідки реалізації *DDoS*-атак є значно критичнішими, оскільки геометричне розсіювання джерел шкідливого трафіку робить неефективними класичні методи фільтрації на основі поодиноких *IP*-адрес [1, 11].

Одним із фундаментальних варіантів реалізації загрози відмови в обслуговуванні на мережевому рівні є шторм *ICMP*-пакетів, або *ICMP Flood*. Протокол керуючих сповіщень Інтернету (*ICMP*) призначений для діагностики та обміну службовими повідомленнями, зокрема для перевірки доступності вузлів за допомогою утиліти *Ping* [18]. Під час реалізації атаки *ICMP Flood* зловмисник або мережа ботів починає генерувати аномально високу щільність

запитів *Echo Request* у напрямку цільового пристрою. Це змушує центральний процесор жертви витрачати монопольні обчислювальні ресурси на деградацію та обробку кожного вхідного пакета, а також на формування обов'язкових зворотних відповідей *Echo Reply* [1]. Якщо сумарний обсяг сміттєвого трафіку перевищує фізичну пропускну здатність вхідного каналу зв'язку або вичерпує ліміти переривань мережевого адаптера, легітимні користувачі повністю втрачають доступ до сервісів, а комутаційне обладнання ядра мережі переходить у стан стійкого перевантаження. Особлива небезпека *ICMP Flood* полягає в тому, що цей трафік часто не блокується внутрішніми пакетними фільтрами підприємств, оскільки адміністратори використовують його як базовий інструмент мережевого моніторингу [14].

Паралельно з *ICMP*-флудом значну загрозу становить атака на вичерпання ресурсів транспортного рівня – *TCP SYN Flood*. Цей вектор експлуатує обмеженість системних ресурсів, що виділяються операційною системою для утримання черги напіввідкритих з'єднань (*SYN Backlog Queue*). Надсилаючи безперервний потік *TCP*-пакетів із прапорцем *SYN* та підробленими (спуфінговими) адресами відправників, зловмисник змушує сервер виділяти пам'ять для кожного запиту та переходити в стан очікування фінального пакета *ACK*. Оскільки відповіді *SYN-ACK* надсилаються на неіснуючі або недоступні *IP*-адреси, тайм-аут очікування завершення сесії призводить до швидкого заповнення системного буфера. Щойно черга *Backlog* вичерпується, сервер втрачає технічну здатність приймати будь-які нові запити, включаючи легітимні підключення від клієнтів компанії. Аналогічно працює загроза класу *UDP Flood*, де ресурси пристрою вимиваються за рахунок змушення операційної системи безперервно перевірити випадкові *UDP*-порти та формувати службові відповіді про їхню недосяжність [18].

Сучасне шкідливе програмне забезпечення та активність кіберзлочинців рідко обмежуються експлуатацією лише одного вразливого протоколу. Реальні загрози завжди мають комплексний, багатокроковий характер, де кожен успішний етап є каталізатором для наступного. Успішна фішингова атака

призводить до зараження робочої станції користувача троянським модулем, який активує внутрішню мережеву розвідку. Оцінивши топологію пласкої мережі, шкідливе ПЗ використовує *ARP-spoofing* для перехоплення трафіку, здійснює латеральне переміщення до серверного сегмента, впроваджує приховані канали передачі даних (наприклад, тунелювання через легітимний *DNS*-трафік) і завершує атаку масованою ін'єкцією вірусу-шифрувальника або DoC-штормом для приховування слідів злочину [12, 13].

Додатковим деструктивним фактором виступає людський фактор та випадкові помилки конфігурування мережевого обладнання. Неправильно налаштовані права доступу, забуті тимчасові дозволи в *ACL*, відсутність парольного захисту на інтерфейсах керування або некоректне підключення периферійних пристроїв створюють технологічні проломи в периметрі безпеки, які миттєво ідентифікуються автоматизованими сканерами зловмисників [5, 10]. Таким чином, захист корпоративного середовища не може бути побудований навколо одного ізольованого інструменту. Для нейтралізації всього спектра загроз канального, мережевого та транспортного рівнів необхідна реалізація концепції ешелонованої оборони, де превентивні інструменти логічної сегментації та апаратної фільтрації щільно інтегровані з детективними системами безперервного активного моніторингу та глибокого аналізу мережевих пакетів [1, 11].

## 1.3 Еволюція засобів захисту: від міжмережевих екранів до систем класу *IDS*/*IPS* та постановка задачі дослідження

Еволюція класичного підходу до забезпечення безпеки корпоративних мереж нерозривно пов'язана з постійним ускладненням методів та інструментарію кібератак [1]. На початкових етапах розвитку мережевих технологій базовим інструментом захисту виступали прості пакетні фільтри, розгорнуті безпосередньо на прикордонних маршрутизаторах [10]. Головним механізмом такого захисту були списки контролю доступу (*Access Control Lists*,

*ACL*), які дозволяли або блокували проходження трафіку на основі статичних критеріїв мережевого та транспортного рівнів: *IP*-адрес відправника та отримувача, типів протоколів і номерів портів. Проте досить швидко стало очевидним, що використання виключно технологій трансляції мережевих адрес (*NAT*) у поєднанні зі статичними правилами *ACL* є вкрай недостатнім для захисту сучасного інформаційного середовища [1, 11]. Трансляція адрес забезпечує лише приховування внутрішньої топології, але не перевіряє вміст транзитних пакетів, тоді як апаратні списки доступу не враховують контекст встановленого з'єднання, поточний стан протоколу чи поведінкові аномалії користувачів [1, 19].

Наступним еволюційним кроком стало впровадження міжмережевих екранів із функцією інспекції стану з'єднань (*Stateful Inspection Firewalls*) [1]. На відміну від простих маршрутизаторів, такі системи почали динамічно відстежувати життєвий цикл кожної *TCP*-сесії або *UDP*-потоку. Зберігаючи інформацію про стан підключень у спеціальній таблиці (*State Table*), міжмережевий екран перевіряв, чи належить вхідний пакет до легітимно ініційованого внутрішнього запиту, і автоматично блокував будь-який несанкціонований трафік, що намагався проникнути ззовні без попереднього встановлення сесії [1, 18]. Це суттєво підвищило рівень безпеки, проте навіть технологія *Stateful Inspection* має фундаментальне обмеження: вона функціонує переважно на рівні транспортних заголовків і не здійснює глибокого аналізу корисного навантаження (*Payload*) на прикладному рівні моделі *OSI* [14].

Сучасні вектори кібератак вийшли далеко за межі простих маніпуляцій із мережевими портами [1]. Зловмисники, розробники вірусів-черв'яків та програм-вимагачів (*Ransomware*) активно використовують легітимні порти та стандартизовані протоколи, такі як *HTTP* (порт 80), *HTTPS* (порт 443) або *DNS* (порт 53), для безперешкодного проходження крізь класичні міжмережеві екрани [1, 11]. Шкідливий код інкапсулюється безпосередньо в тіло дозволеного повідомлення, маскуючись під звичайний веб-запит. Для протидії таким комплексним загрозам, що експлуатують вразливості програмного забезпечення та методи соціальної інженерії, виникла гостра необхідність в інтелектуальних

інструментах, здатних розпізнавати семантику прикладних протоколів та виявляти специфічні поведінкові патерни. Відповіддю на ці виклики стала розробка систем виявлення вторгнень (*Intrusion Detection Systems*, *IDS*) [14, 16].

Архітектура систем класу *IDS* концептуально відрізняється від принципів роботи міжмережевих екранів. Основне завдання *IDS* полягає не в лінійному блокуванні трафіку, а в його безперервному, глибокому моніторингу (*Deep Packet Inspection*), аналізі та генеруванні деталізованих сповіщень про виявлену підозрілу активність. Для забезпечення повної видимості транзитних потоків впроваджуються мережеві системи виявлення вторгнень (*NIDS*), які функціонують на основі трьох ключових підсистем: механізму збору трафіку, аналітичного рушія та модуля реагування [14, 16].

Перший етап – збір трафіку – є критично важливим для забезпечення прозорості моніторингу. Для того, щоб сенсор *IDS* міг аналізувати мережевий обмін без внесення затримок у продуктивне середовище, застосовується технологія дзеркалювання портів (*Switched Port Analyzer*, *SPAN*). Вона дозволяє апаратно копіювати кадри з транкових або користувацьких портів комутатора та спрямовувати їх на виділений аналітичний інтерфейс [1]. Обов'язковою умовою коректної роботи сенсора є переведення його мережевої карти у так званий нерозбірливий режим (*Promiscuous Mode*). У стандартному стані мережевий адаптер ігнорує всі кадри, *MAC*-адреса призначення яких не збігається з його власною апаратною адресою. Нерозбірливий режим деактивує цей апаратний фільтр на рівні драйвера, змушуючи операційну систему приймати та передавати аналітичному рушію абсолютно всі пакети, що фізично проходять через кабель підключення [14, 16].

Другим і найскладнішим етапом є інтелектуальний аналіз зібраного трафіку, який реалізується за допомогою сигнатурного методу та виявлення аномалій [14, 21]. Сигнатурний аналіз передбачає зіставлення структури та вмісту кожного перехопленого пакета з величезною базою даних відомих шаблонів кібератак (сигнатур). Наприклад, в інтелектуальному рушії *Suricata* кожна сигнатура описується суворим синтаксисом, що містить заголовок

правила (протокол, напрямок, *IP*-адреси) та його опції [8, 20]. Завдяки цьому система здатна ідентифікувати специфічну послідовність байтів у корисному навантаженні, перевірити розмір фрагментованих даних, проаналізувати нестандартні комбінації *TCP*-прапорців або виявити використання заборонених команд у протоколах віддаленого керування [20, 21]. Незважаючи на високу точність та мінімальний рівень хибних спрацювань, сигнатурний аналіз має недолік: він є реактивним і не здатен ідентифікувати атаки «нульового дня» (*Zero-Day*), для яких ще не випущено відповідних правил[14, 16].

Для компенсації цього обмеження застосовується метод аналізу аномалій, який функціонує за принципом відхилення від еталона. Аналітичний рушій контролює протокольні стандарти та фіксує невідповідності технічним специфікаціям *RFC*. Яскравим і надзвичайно важливим прикладом такої аномалії є виявлення пакетів із некоректною контрольною сумою транспортного рівня (*TCP*/*UDP Checksum*). Контрольна сума є базовим математичним механізмом, що гарантує цілісність переданих даних [1, 18]. Проте у сучасних апаратно-програмних комплексах та віртуалізованих середовищах повсюдно використовується технологія апаратного розвантаження (*TCP Checksum Offloading*), суть якої полягає в делегуванні процесу обчислення контрольної суми від центрального процесора безпосередньо до контролера мережевої карти. Оскільки сенсор *IDS* перехоплює трафік через дзеркальний порт ще до того моменту, як мережева карта відправника встигає підрахувати і вставити цю суму у заголовок пакета, система виявлення вторгнень фіксує порушення цілісності та генерує попередження про недійсну контрольну суму (*invalid checksum*). Глибоке інженерне розуміння таких протокольних артефактів та особливостей взаємодії віртуального комутаційного обладнання є критично необхідним для уникнення хибних спрацювань і правильної інтерпретації журналів безпеки [14, 21].

Оскільки класичні *IDS* працюють з копією трафіку (*Out-of-Band*), вони виконують функцію пасивного аудиту, передаючи зібрані дані до систем централізованого управління інцидентами (*SIEM*) для подальшої кореляції. Логічним розвитком цього напрямку є переведення систем безпеки у режим

активного запобігання вторгненням (*IPS*), що передбачає встановлення сенсора в розрив каналу зв'язку (*Inline Mode*). Така архітектура дозволяє не лише фіксувати, але й миттєво відкидати шкідливі пакети [14, 16]. У сучасній ешелонованій системі захисту міжмережеві екрани виступають першою лінією оборони, відсікаючи масовий небажаний трафік на мережевому рівні, тоді як системи класу *IDS*/*IPS* здійснюють глибоку інспекцію легітимних потоків на наявність прихованих загроз [11, 16].

З огляду на проведений аналіз архітектурних недоліків пласких топологій, розмаїття векторів кібератак та обмеженості традиційних інструментів фільтрації, формується чітка постановка задачі для даного дипломного дослідження. Метою роботи є проєктування, практична реалізація та експериментальна верифікація комплексної багаторівневої системи захисту корпоративної мережі у середовищі апаратної емуляції *PNETLab* [2, 7, 17]. Для досягнення цієї мети необхідно вирішити низку інженерних завдань. По-перше, розробити масштабовану ієрархічну топологію мережі на базі комутаторів рівня ядра та доступу. По-друге, реалізувати жорстку логічну сегментацію інфраструктури за допомогою технології *VLAN* та впровадити політики найменших привілеїв, налаштувавши апаратну фільтрацію міжсегментного обміну даними через списки контролю доступу (*Extended ACL*) [5, 10]. По-третє, інтегрувати в архітектуру мережі сервер активного моніторингу на базі *IDS Suricata*, забезпечивши перехоплення трафіку через *SPAN*-порти. Фінальним етапом дослідження має стати проведення серії контрольованих кібератак (мережевої розвідки, сканування та симуляції відмови в обслуговуванні) для підтвердження ефективності розроблених кастомних сигнатур безпеки та оцінки здатності побудованої системи своєчасно ідентифікувати й блокувати несанкціоновану активність [14, 21].

# 2 ТЕОРЕТИЧНІ ОСНОВИ ТЕХНОЛОГІЙ СЕГМЕНТАЦІЇ ТА МОНІТОРИНГУ ТРАФІКУ

## 2.1 Технології логічної сегментації та маршрутизації (*VLAN*)

Впровадження концепції ешелонованої оборони всередині сучасного корпоративного периметра вимагає розгортання стійких інструментів ізоляції інформаційних потоків безпосередньо на нижніх рівнях еталонної моделі взаємодії відкритих систем [1, 11]. Фундаментальним механізмом, що забезпечує виконання даного завдання на канальному рівні (*L2*), є технологія віртуальних локальних мереж (*Virtual Local Area Networks*, *VLAN*). Вона дозволяє здійснити логічне розділення єдиної фізичної інфраструктури комутації на кілька ізольованих широкомовних доменів. Завдяки цьому суттєво підвищується ефективність використання смуги перепускної здатності, оптимізується керування адресним простором та формуються первинні межі безпеки, які перешкоджають безконтрольному поширенню шкідливого трафіку та реалізації внутрішніх кібератак [1, 5].

Технологічна основа функціонування віртуальних локальних мереж регламентується міжнародним стандартом *IEEE* 802.1*Q* [6]. Цей стандарт визначає уніфіковані правила маркування (тегування) кадрів *Ethernet*, що дозволяє комутаційному обладнанню ідентифікувати належність кожного пакета до конкретного логічного сегмента під час його транспортування через спільні магістральні канали зв'язку. У класичному кадрі *Ethernet II*, який використовується в несегментованих мережах, заголовки містять лише базову інформацію про апаратні *MAC*-адреси відправника та отримувача, а також поле типу протоколу вищого рівня. Специфікація стандарту *IEEE* 802.1*Q* передбачає модифікацію структури кадру шляхом інтеграції додаткового чотирьохбайтового (32-бітного) поля, відомого як тег *VLAN*, яке впроваджується безпосередньо між полем адреси джерела (*Source MAC*) та полем типу або довжини (*EtherType*/*Length*). Додавання цього заголовка дещо змінює

максимальний розмір кадру *Ethernet*, збільшуючи його з класичних 1518 байтів до 1522 байтів, що вимагає від сучасних мережевих інтерфейсів підтримки обробки таких модифікованих структур без виникнення апаратних помилок типу *Baby Giant* [1].

Внутрішня архітектура чотирьохбайтового тегу стандарту *IEEE* 802.1*Q* складається з кількох чітко розмежованих функціональних полів, кожне з яких відіграє важливу роль у процесах комутації та забезпечення якості обслуговування. Перші два байти (16 бітів) відведені під ідентифікатор протоколу тегування (*Tag Protocol Identifier*, *TPID*). Для стандарту 802.1*Q* це поле завжди містить фіксоване шістнадцяткове значення 0x8100 [6]. Коли комутатор приймає кадр і виявляє у полі типу дане значення, він миттєво розпізнає пакет як тегований і переходить до аналізу наступних двох байтів, які мають назву інформації керування тегом (*Tag Control Information*, *TCI*). Поле *TCI*, своєю чергою, поділяється на три субполя. Перші 3 біти складають поле пріоритету коду точки (*Priority Code Point*, *PCP*), яке сумісне зі специфікацією *IEEE* 802.1p і визначає один із восьми рівнів пріоритету кадру. Це дозволяє впроваджувати механізми якості обслуговування (*QoS*) та забезпечувати першочергове пересилання критичного до затримок трафіку, наприклад, пакетів *IP*-телефонії або відеоконференцій, порівняно зі звичайним веб-трафіком [6, 18]. Наступний 1 біт відведений під індикатор придатності до скидання (*Drop Eligible Indicator*, *DEI*), який раніше мав назву канонічного формату (*CFI*) і використовувався для сумісності з мережами *Token Ring*. У сучасних інфраструктурах поле *DEI* вказує на пакети, які можуть бути знищені комутаційним обладнанням у першу чергу в умовах виникнення критичного перевантаження або заторів у каналах зв'язку. Остання, найбільш вагома частина інформації керування тегом, займає 12 бітів і являє собою безпосередній ідентифікатор віртуальної мережі (*VLAN Identifier*, *VID*). Оскільки для кодування *VID* виділено 12 бітів, математично максимальна кількість унікальних логічних сегментів, які можна створити в межах однієї інфраструктури під керуванням стандарту 802.1*Q*, становить 4096 (від 0 до 4095). При цьому значення 0 та 4095

є строго зарезервованими для системних потреб, а ідентифікатор 1 за замовчуванням призначений для стандартної віртуальної мережі (*Default VLAN*), у якій функціонують усі порти комутатора до моменту їхнього адміністрування. Решта значень у діапазоні від 2 до 4094 складають робоче поле утилітарних ідентифікаторів, які розподіляються між структурними підрозділами організації.

Для забезпечення коректного руху тегованих та нетегованих потоків даних інтерфейси комутаційного обладнання конфігуруються у різних режимах роботи, основними з яких є порти доступу (*Access ports*) та магістральні транкові порти (*Trunk ports*) [5]. Порт доступу призначений для взаємодії з кінцевим обладнанням користувачів, операційні системи якого зазвичай не підтримують ігнорують або не вміють обробляти заголовки стандарту 802.1*Q*. Такий порт логічно прив'язується виключно до одного конкретного ідентифікатора *VID*. Коли нетегований кадр від робочої станції надходить на вхідний інтерфейс порту доступу, комутатор всередині своєї логічної матриці асоціює його з відповідним *VLAN*. Під час пересилання кадру з вихідного порту доступу в напрямку іншого кінцевого вузла, внутрішній тег повністю видаляється, і клієнтський пристрій отримує стандартний кадр *Ethernet II*. Натомість магістральні транкові порти розгортаються на інтерфейсах, що з'єднують комутатори між собою або підключають комутатор до прикордонного маршрутизатора чи сервера моніторингу [1]. Головне завдання транкового каналу – мультиплексування трафіку багатьох віртуальних мереж через один фізичний кабель. Усі кадри, що проходять через транковий порт, обов'язково містять 12-бітне поле *VID*, що дозволяє приймаючому пристрою безпомилково визначити, до якого саме ізольованого сегмента належить пакет, та спрямувати його у відповідний домен комутації.

Важливою архітектурною концепцією функціонування транкових каналів є поняття рідної віртуальної мережі (*Native VLAN*). За замовчуванням, з метою забезпечення зворотної сумісності зі старим мережевим обладнанням, трафік, що належить до *Native VLAN* (традиційно це *VLAN* 1), передається через магістральний транковий канал у сирому, нетегованому вигляді. Якщо

комутатор приймає з транку кадр, який не містить чотирьохбайтового тегу 802.1*Q*, він автоматично направляє його у домен, визначений як *Native*. З погляду інформаційної безпеки, збереження стандартних налаштувань *Native VLAN* несе в собі критичні ризики реалізації атак класу *VLAN Hopping*, зокрема методом подвійного тегування (*Double Tagging*) [1, 5]. Зловмисник, перебуваючи у *VLAN* 1, має можливість сформувати шкідливий пакет із двома вкладеними тегами: зовнішнім (*VLAN* 1) та внутрішнім (наприклад, *VLAN* 10, де розташовані сервери адміністрування). Під час проходження через перший комутатор зовнішній тег *Native VLAN* видаляється, а пакет у нетегованому вигляді з рештою заголовків передається у транк. Наступний комутатор, аналізуючи кадр, виявляє другий тег і помилково спрямовує пакет у захищений сегмент, забезпечуючи зловмиснику несанкціонований односторонній доступ в обхід засобів маршрутизації. Для нівелювання даного вектора загрози в інженерній практиці застосовують комплекс превентивних заходів: *Native VLAN* обов'язково перевизначається зі стандартного першого на спеціально створений ізольований ідентифікатор, який не призначається жодному користувацькому порту, а на магістральних стиках активується режим обов'язкового тегування навіть для рідного трафіку [5].

Оскільки кожна віртуальна мережа є абсолютно ізольованим широкомовним доменом, пряма взаємодія між пристроями з різних *VLAN* на канальному рівні є технічно неможливою. Для організації легітимного обміну даними між відділами виникає необхідність підключення засобів третього (мережевого) рівня моделі *OSI*, що реалізують механізми міжвіланної маршрутизації (*Inter-VLAN Routing*). В інфраструктурах середнього масштабу для цього найчастіше використовуються два основні підходи: класична схема «маршрутизатор на паличці» (*Router-on-a-Stick*) або розгортання багатошарових комутаторів L3 з використанням віртуальних інтерфейсів комутатора (*Switch Virtual Interfaces*, *SVI*) [5]. Схема *Router-on-a-Stick* передбачає підключення одного фізичного інтерфейсу маршрутизатора до транкового порту центрального комутатора. На самому маршрутизаторі цей єдиний інтерфейс логічно конфігурується у вигляді кількох підінтерфейсів (*Subinterfaces*), кожен з

яких жорстко прив'язується до конкретного тегу dot1q і отримує унікальну *IP*-адресу, що виступає шлюзом за замовчуванням (*Default Gateway*) для відповідної підмережі. Будь-який пакет, що прямує, наприклад, з користувацького сегмента до серверного, змушений покинути межі комутатора, піднятися до підінтерфейсу маршрутизатора, пройти через таблицю маршрутизації та повернутися назад у комутаційну матрицю вже з новим ідентифікатором *VID*. Даний підхід забезпечує високу централізацію керування, проте створює обмеження смуги пропускання та формує потенційне вузьке місце системи. Натомість використання багатошарової комутації (*Multilayer Switching*) на базі інтерфейсів *SVI* дозволяє виконувати маршрутизацію пакетів безпосередньо всередині центрального комутатора на швидкості роботи апаратних мікросхем *ASIC*, що мінімізує затримки та оптимізує транспортні потоки [1].

У межах розробленої практичної моделі захищеної мережі технологія логічної сегментації є фундаментальною основою, на якій базуються всі подальші рівні системи безпеки. Створення ізольованих сегментів (*VLAN* 10 для адміністраторів, *VLAN* 20 для загальних користувачів та *VLAN* 30 для серверного обладнання) дозволяє кардинально знизити площу потенційного ураження. Локалізація широкомовного трафіку межами одного ідентифікатора *VID* повністю ліквідує можливість реалізації атак класу *ARP-spoofing* між різними відділами: навіть якщо робоча станція в користувацькому сегменті буде повністю скомпрометована шкідливим програмним забезпеченням, її фальшиві *ARP*-відповіді фізично не зможуть досягти або отруїти кеш пристроїв з адміністративної чи серверної підмереж [1]. Будь-яка міжсегментна взаємодія стає контрольованою та детермінованою, оскільки трафік гарантовано проходить через вузли маршрутизації згорнутого ядра, де з'являється технічна можливість застосування засобів глибокого аналізу, інспекції заголовків та апаратної фільтрації пакетів [11, 14].

## 2.2 Механізми фільтрації на базі списків доступу (*ACL*)

Реалізація логічної сегментації мережі за допомогою технології *VLAN* вирішує завдання ізоляції широкомовних доменів, проте не забезпечує розмежування прав доступу на мережевому рівні, оскільки будь-який міжсегментний трафік, що проходить через вузол маршрутизації, за замовчуванням дозволяється. Для впровадження концепції найменших привілеїв (*Principle of Least Privilege*) та побудови контрольованого внутрішнього периметра застосовуються списки контролю доступу (*Access Control Lists*, *ACL*) [10]. З технічного погляду, *ACL* являє собою впорядкований набір дискретних правил (умов), які послідовно застосовуються до кожного мережевого пакета, що проходить через інтерфейс маршрутизатора або багатошарового комутатора. Головним завданням цього інструменту є класифікація інформаційних потоків та прийняття безальтернативного рішення щодо долі кожного пакета: дозволити його подальше проходження (*permit*) або жорстко відкинути (*deny*) на основі аналізу заголовків третього (мережевого) та четвертого (транспортного) рівнів моделі *OSI* [18, 19].

Історично склалося два основні типи списків контролю доступу, які підтримуються більшістю виробників мережевого обладнання: стандартні (*Standard ACL*) та розширені (*Extended ACL*) [10]. Стандартні списки доступу мають вкрай обмежений функціонал, оскільки їхній аналітичний алгоритм здатний інспектувати лише одне поле в заголовку *IP*-пакета – мережеву адресу відправника (*Source IP*). Маршрутизатор, на якому застосовано стандартний *ACL*, не перевіряє адресу призначення, тип протоколу інкапсуляції чи номери портів. Внаслідок цього стандартні списки не здатні забезпечити гранульовану фільтрацію [1, 10]. Наприклад, з їхньою допомогою неможливо дозволити користувачу доступ до корпоративного веб-сервера (порти 80/443), але при цьому заборонити йому ж підключення до того самого сервера по протоколу віддаленого керування *SSH* (порт 22), оскільки будь-яке правило застосовуватиметься до всього трафіку від даного джерела безумовно [19]. Через

таку специфіку стандартні *ACL* сьогодні рідко використовуються для розмежування доступу користувачів і застосовуються переважно для службових завдань, таких як фільтрація оновлень маршрутної інформації або визначення адрес для трансляції *NAT* [10].

Для розгортання сучасних політик інформаційної безпеки використовуються розширені списки контролю доступу (*Extended ACL*). Цей тип списків забезпечує глибоку інспекцію заголовків пакетів, аналізуючи одночасно цілий комплекс параметрів: *IP*-адресу джерела, *IP*-адресу призначення, тип транспортного протоколу (*TCP*, *UDP*, *ICMP*, *OSPF* тощо), а також номери портів джерела та призначення [18, 19]. Більше того, розширені списки здатні аналізувати додаткові бітові прапорці всередині протоколів. Наприклад, правило *ACL* може бути налаштоване таким чином, щоб дозволяти проходження *TCP*-пакетів лише з встановленими прапорцями *ACK* або *RST* (параметр *established*), що фізично унеможливлює ініціювання нових з'єднань із зовнішньої мережі, але дозволяє внутрішнім користувачам отримувати відповіді на свої запити. Аналогічно для протоколу *ICMP* можна дозволити проходження лише пакетів типу *Echo Reply*, заблокувавши можливість пінгування внутрішніх вузлів ззовні [1, 19].

Архітектурна логіка обробки пакетів механізмом *ACL* є строго послідовною. Коли пакет надходить на мережевий інтерфейс, до якого прив'язаний список доступу, маршрутизатор починає перевіряти його параметри на відповідність правилам списку, рухаючись строго зверху вниз, від першого (найвищого) рядка до останнього. Фундаментальним принципом є правило першого збігу (*First Match Rule*): як тільки параметри пакета задовольняють умови певного рядка, маршрутизатор негайно виконує вказану дію (пропускає або відкидає пакет) і припиняє подальший аналіз списку. Усі наступні правила для цього конкретного пакета ігноруються. Ця особливість вимагає від мережевого інженера бездоганного планування порядку записів: найбільш специфічні та вузькі правила (наприклад, заборона доступу для одного конкретного *IP*-адреси) повинні розміщуватися на самому початку списку, тоді

як найзагальніші (наприклад, дозвіл доступу для цілої підмережі) – у самому кінці. Якщо порушити цю ієрархію і поставити правило дозволу всім перед правилом заборони одиниці, заборонне правило ніколи не спрацює [10, 19].

Ще однією критично важливою особливістю роботи *ACL* є наявність прихованого правила глобальної заборони (*Implicit Deny Any*). Будь-який список доступу, створений на обладнанні корпоративного класу, містить невидимий рядок у самому кінці переліку, який безумовно відкидає будь-який трафік. Якщо параметри транзитного пакета не збіглися з жодним із явно прописаних адміністратором правил, такий пакет автоматично знищується маршрутизатором з генеруванням відповідного *ICMP*-повідомлення про недосяжність вузла (*Destination Unreachable*, *Administratively Prohibited*). Таким чином, кожен *ACL*, що застосовується на інтерфейсі, повинен містити щонайменше одне правило дозволу (*permit*), інакше він призведе до повного блокування мережевого обміну [10, 19].

З метою визначення діапазонів *IP*-адрес у правилах *ACL* використовується специфічний математичний апарат – зворотні маски (*Wildcard Masks*). На відміну від класичних масок підмереж, які безперервно заповнюються бітовими одиницями зліва направо для позначення мережевої частини адреси, зворотна маска функціонує за принципом інверсної бітової перевірки. У двійковому поданні зворотної маски кожен біт, що має значення «0», є жорсткою вимогою – він вказує маршрутизатору на те, що відповідний біт у *IP*-адресі пакета повинен абсолютно точно збігатися з бітом у правилі. Натомість біт зі значенням «1» у зворотній масці означає «ігнорувати» (Don't *Care*) – маршрутизатору байдуже, яке значення має цей біт у пакеті, що перевіряється. Наприклад, якщо необхідно заблокувати цілу користувацьку підмережу 192.168.20.0/24, у правилі *ACL* буде вказано адресу 192.168.20.0 та зворотну маску 0.0.0.255. У двійковому форматі ця маска виглядає як 24 нулі та 8 одиниць, що наказує маршрутизатору строго перевірити перші три октети (вони обов'язково мають бути 192.168.20), а останній октет може набувати будь-якого значення від 0 до 255. Гнучкість зворотних масок дозволяє створювати складні нелінійні правила, наприклад,

фільтрувати виключно парні або непарні *IP*-адреси всередині одного сегмента, що є абсолютно неможливим при використанні традиційних масок маршрутизації [1, 19].

Окрім логіки побудови, ефективність роботи списків доступу критично залежить від напрямку їх застосування на мережевому інтерфейсі. Кожен *ACL* може бути прив'язаний до порту або віртуального інтерфейсу (*SVI*) у двох напрямках: вхідному (*Inbound*) або вихідному (*Outbound*). У разі застосування вхідного *ACL* маршрутизатор перевіряє пакет одразу після його отримання з кабелю мережевою картою, ще до моменту пошуку маршруту в таблиці маршрутизації. Якщо пакет порушує правило безпеки, він відкидається негайно, не витрачаючи ресурси процесора на пошук шляху. Вихідний *ACL*, своєю чергою, перевіряє пакет уже після прийняття рішення про маршрутизацію, перед самою відправкою в цільовий інтерфейс. Згідно з класичними інженерними правилами дизайну, розширені списки доступу слід розміщувати якомога ближче до джерела трафіку (на вхідних інтерфейсах), щоб відсікати небажані пакети ще на стадії їх зародження і не завантажувати транзитні магістралі сміттєвим трафіком [10, 19].

У високопродуктивних корпоративних середовищах процес інспекції трафіку механізмами *ACL* не уповільнює загальну швидкість роботи мережі завдяки використанню спеціалізованої апаратної пам'яті *TCAM* (*Ternary Content-Addressable Memory*). На відміну від звичайної оперативної пам'яті (*RAM*), де пошук відбувається послідовно і вимагає часу процесора, *TCAM* дозволяє оцінювати пакет на відповідність одразу всім правилам списку доступу за один такт синхронізації (апаратно). Це означає, що незалежно від того, складається *ACL* з 10 чи 1000 правил, затримка на обробку пакета буде мінімальною і сталою величиною. Проте пам'ять *TCAM* є надзвичайно дорогою та має суворо обмежений об'єм, тому оптимізація списків доступу та об'єднання кількох правил (сумаризація) за допомогою зворотних масок залишається важливим інженерним завданням [1, 18, 19].

У межах розробленої в даному дослідженні архітектури корпоративної мережі механізм розширених *ACL* виступає головним інструментом забезпечення внутрішнього контролю доступу. Практичне завдання полягало в тому, щоб надати звичайним користувачам (*VLAN* 20) доступ до серверів (*VLAN* 30) та виходу в Інтернет, але жорстко заблокувати будь-які спроби комунікації з пристроями в мережі адміністраторів (*VLAN* 10). Для реалізації цього завдання на підінтерфейсі маршрутизатора (або віртуальному шлюзі комутатора), який обслуговує підмережу користувачів, було застосовано розширений список доступу у вхідному напрямку (*inbound*). Першим (найвищим) правилом у цьому списку стала жорстка заборона міжсегментної маршрутизації: *deny ip* 192.168.20.0 0.0.0.255 192.168.10.0 0.0.0.255. Це правило гарантує, що будь-який пакет, джерелом якого є *IP*-адреса з діапазону користувачів, а метою – *IP*-адреса з діапазону адміністраторів, буде миттєво відкинутий ще до потрапляння в ядро мережі. Другим, замикаючим правилом, було додано безумовний дозвіл *permit ip any any*, який перекриває приховане системне блокування і дозволяє всьому іншому легітимному трафіку безперешкодно проходити процедуру маршрутизації до серверів та зовнішньої мережі. Таким чином, інтеграція технології *VLAN* на рівні доступу та апаратних механізмів *ACL* на рівні розподілу дозволяє побудувати надійний превентивний бар'єр, який ізолює загрози та значно зменшує площу потенційної атаки на критичні вузли інфраструктури [1, 11].

## 2.3 Архітектура та принципи роботи систем виявлення вторгнень (*IDS*)

Забезпечення комплексного інформаційного захисту сучасної корпоративної інфраструктури неможливо реалізувати виключно за допомогою превентивних механізмів, таких як логічна сегментація або пакетна фільтрація на базі списків доступу (*ACL*). Традиційні міжмережеві екрани та маршрутизатори приймають рішення щодо дозволу або блокування трафіку на основі поверхневої інспекції заголовків мережевого та транспортного рівнів,

залишаючись абсолютно сліпими до вмісту корисного навантаження (*Payload*) мережевих пакетів. Усвідомлюючи це обмеження, сучасні кіберзлочинці активно інкапсулюють шкідливий код, команди управління ботнетами та експлойти всередину легітимних протоколів прикладного рівня (наприклад, *HTTP*, *DNS* або *TLS*), які безперешкодно проходять крізь класичні периметрові засоби захисту. Для виявлення таких прихованих загроз, проактивного моніторингу внутрішнього середовища та своєчасного інформування адміністраторів про спроби компрометації систем використовуються системи виявлення вторгнень (*Intrusion Detection Systems*, *IDS*) [1, 14]. На відміну від міжмережевих екранів, мережеві системи виявлення вторгнень (*NIDS*) функціонують за принципом глибокого аналізу пакетів (*Deep Packet Inspection*, *DPI*), здійснюючи безперервний семантичний розбір усього транзитного трафіку [14, 16].

З архітектурної точки зору мережева система виявлення вторгнень є багатокомпонентним програмно-апаратним комплексом, який логічно поділяється на три ключові підсистеми: модуль перехоплення мережевого трафіку, інтелектуальне аналітичне ядро (*Detection Engine*) та підсистему журналювання і генерування сповіщень. Оскільки класична *IDS* працює в режимі пасивного аудиту і не повинна вносити жодних затримок у продуктивний обмін даними, вона встановлюється не в розрив каналу зв'язку (*Inline*), а паралельно до нього (*Out-of-Band*) [16]. Для забезпечення сенсора *IDS* необхідними даними використовується технологія дзеркалювання портів, відома як *SPAN* (*Switched Port Analyzer*) або *Port Mirroring*. При налаштуванні локальної *SPAN*-сесії процесор центрального комутатора програмується таким чином, щоб створювати точну апаратну копію кожного кадру *Ethernet*, який надходить на визначені користувацькі або магістральні порти (*Source Ports*), і миттєво перенаправляти цю копію на спеціально виділений аналітичний порт (*Destination Port*), до якого фізично підключений сервер *IDS*. У великих географічно розподілених мережах замість локального *SPAN* використовується технологія інкапсульованого віддаленого дзеркалювання (*ERSPAN*), яка дозволяє загортати

скопійований трафік у *GRE*-тунелі та передавати його через маршрутизовані магістралі до централізованого дата-центру безпеки [14].

Для того щоб мережевий інтерфейс сервера *IDS* міг коректно приймати та передавати на обробку перехоплений дзеркальований трафік, він в обов'язковому порядку переводиться у так званий нерозбірливий режим функціонування (*Promiscuous Mode*). Відповідно до стандартів сімейства *IEEE* 802.3, контролер мережевої карти (*NIC*) за замовчуванням здійснює апаратну фільтрацію вхідних кадрів: він перериває роботу центрального процесора лише в тому випадку, якщо *MAC*-адреса призначення у вхідному кадрі збігається з його власною апаратною адресою, або якщо кадр є широкомовним (*Broadcast*). Оскільки дзеркальований трафік містить чужі *MAC*-адреси кінцевих користувачів та серверів, стандартний мережевий адаптер ігнорував би ці пакети. Активація нерозбірливого режиму на рівні драйвера операційної системи деактивує даний апаратний фільтр, змушуючи мережеву карту приймати абсолютно всі електричні сигнали з кабелю, формувати з них кадри та передавати сирі сокети (*Raw Sockets*) безпосередньо до аналітичного рушія системи виявлення вторгнень [14, 21].

Після надходження трафіку до оперативної пам'яті сервера в роботу вступає аналітичне ядро *IDS*, яке застосовує два фундаментальні методи виявлення загроз: сигнатурний аналіз (*Signature-based Detection*) та аналіз відхилень від еталона (*Anomaly-based Detection*). Сигнатурний аналіз базується на пошуку в транзитному трафіку специфічних патернів (послідовностей байтів, характерних текстових рядків, комбінацій заголовків), які достовірно відповідають ознакам відомих кібератак. Високопродуктивні рушії, такі як *Suricata*, використовують для цього розгалужену мову правил [8, 20]. Кожна сигнатура складається із заголовка, що визначає базові мережеві параметри (протокол, *IP*-адреси, порти, напрямок потоку), та тіла правила з набором опцій. Опції дозволяють здійснювати глибокий розбір корисного навантаження, наприклад, шукати ключове слово у тілі *HTTP*-запиту, перевіряти розмір пакета за допомогою параметра *dsize* (що є критично важливим для ідентифікації атак класу *Ping of Death* або *ICMP Flood*) або контролювати нестандартні комбінації

*TCP*-прапорців. Оскільки в сучасних умовах система повинна порівнювати кожен транзитний пакет із десятками тисяч сигнатур на гігабітних швидкостях, рушії використовують складні математичні алгоритми мультишаблонного пошуку підрядків, такі як алгоритм Ахо-Корасік (*Aho-Corasick*) або алгоритми сімейства Хіршберга, які дозволяють виконувати пошук за один прохід по тексту пакета незалежно від кількості правил у базі даних [20, 21].

Сучасні зловмисники чудово обізнані з принципами роботи сигнатурних аналізаторів і активно використовують методи обходу (*Evasion Techniques*), найпоширенішими з яких є фрагментація *IP*-пакетів та сегментація *TCP*-потоків. Штучно розбиваючи шкідливий запит на кілька десятків дрібних фрагментів, хакер досягає ситуації, коли жоден окремий пакет не містить повної сигнатури атаки, що робить її невидимою для простих систем моніторингу. Для протидії цьому вектору обходу архітектура сучасної *IDS* обов'язково включає модулі дефрагментації *IP* (*Defragmentation*) та перезбирання потоків (*TCP Stream Reassembly*). Перш ніж передати дані на сигнатурну перевірку, система виявлення вторгнень виділяє значні обсяги оперативної пам'яті для тимчасового зберігання фрагментованих пакетів, відстежує їхні порядкові номери (*Sequence Numbers*) і відновлює первісний вигляд всього сеансу зв'язку так, як би його побачила операційна система цільового сервера. Лише після такого нормалізованого складання (*Reassembly*) застосовуються правила пошуку [21].

Паралельно із сигнатурним пошуком система здійснює безперервний аналіз протокольних аномалій. Цей механізм не шукає конкретний шкідливий код, а перевіряє трафік на сувору відповідність специфікаціям *Request for Comments* (*RFC*) від інженерної ради Інтернету (*IETF*). Якщо пакет сформований з порушенням стандарту (наприклад, містить одночасно встановлені прапорці *SYN* та *FIN*, має некоректну довжину заголовка або хибну контрольну суму), система класифікує його як спробу експлуатації вразливостей мережевого стеку [1]. Одним із найцікавіших та найскладніших для інтерпретації інженерних артефактів, з яким стикаються фахівці під час налаштування *IDS* у віртуалізованих інфраструктурах, є масове хибне виявлення помилок

контрольних сум пакетів транспортного рівня (*TCP*/*UDP Invalid Checksum*). З точки зору теорії мереж, контрольна сума є математичним значенням (зазвичай 16-бітним), яке обчислюється відправником на основі всього вмісту пакета і дозволяє отримувачу переконатися, що дані не були спотворені під час передачі кабелями чи маршрутизаторами [18].

Поява попереджень про недійсні контрольні суми в логах *IDS* пояснюється функціонуванням технології апаратного розвантаження (*Checksum Offloading*), яка повсюдно активується за замовчуванням у сучасних операційних системах та гіпервізорах. Для економії та оптимізації ресурсів центрального процесора (*CPU*) операційна система делегує ресурсомістке завдання підрахунку контрольних сум безпосередньо мікропроцесору фізичної або віртуальної мережевої карти [18]. Відповідно, ядро операційної системи формує *TCP*-пакет, залишає поле контрольної суми порожнім (або заповнює його нулями) і відправляє його на мережевий інтерфейс. Оскільки сенсор *IDS*, підключений до дзеркального *SPAN*-порту всередині віртуального комутатора, перехоплює копію цього пакета ще на проміжному етапі – коли пакет покинув ядро ОС, але ще не пройшов етап обчислення контрольної суми апаратним контролером, – аналітичний рушій *Suricata* отримує сирий пакет. Здійснюючи власний математичний підрахунок та порівнюючи його з порожнім полем заголовка, система фіксує невідповідність і піднімає тривогу пріоритетного рівня. Розуміння цієї низькорівневої апаратно-програмної взаємодії є критично необхідним для мережевого інженера, оскільки воно вимагає точкового калібрування конфігураційних файлів *IDS* (наприклад, відключення інспекції *Checksum* для внутрішніх локальних мереж) з метою усунення хибнопозитивних спрацювань та збереження обчислювальної потужності сервера для аналізу реальних загроз [14, 21].

Інтеграція системи виявлення вторгнень у загальну архітектуру захисту перетворює статичну мережеву інфраструктуру на кероване середовище з високим рівнем видимості (*Visibility*) [11]. У той час як механізми логічної сегментації *VLAN* обмежують шляхи поширення загроз, а списки контролю

доступу *ACL* жорстко блокують нелегітимні комунікації на мережевому рівні, *IDS* забезпечує глибокий аудит тих інформаційних потоків, яким дозволено переміщуватися всередині підприємства. Отримуючи дзеркальні копії трафіку з ядра мережі, система не лише фіксує факти здійснення *DDoS*-атак чи сканування портів, але й зберігає повний пакетний слід інциденту, що є безцінним матеріалом для подальшого криміналістичного розслідування (*Forensics*) та вдосконалення політик кібербезпеки організації [1, 11].

## 2.4 Технології глибокого аналізу пакетів (*DPI*) та централізованого управління інцидентами (*SIEM*)

Фундаментальною відмінністю сучасних систем виявлення вторгнень від класичного комутаційного обладнання та міжмережевих екранів є підтримка технології глибокого аналізу пакетів (*Deep Packet Inspection*, *DPI*). Традиційні засоби захисту периметра та механізми апаратної фільтрації, такі як списки контролю доступу (*ACL*), приймають рішення щодо дозволу або блокування трафіку виключно на основі перевірки заголовків мережевого (L3) та транспортного (*L4*) рівнів моделі *OSI*. Вони інспектують *IP*-адреси, типи протоколів та номери портів, розглядаючи корисне навантаження пакета (*Payload*) як неподільний і невідомий масив даних. Технологія *DPI*, натомість, долає ці обмеження, забезпечуючи повноцінний семантичний розбір інформаційного наповнення на прикладному рівні (*L7*). Це дозволяє системі розрізняти легітимні команди всередині дозволених протоколів від спроб експлуатації вразливостей, незалежно від того, які нестандартні порти використовує зловмисник для маскування своєї активності [1, 5, 14].

Процес глибокого аналізу пакетів в інтелектуальних рушіях на кшталт *Suricata* є складним багатокроковим алгоритмом, який вимагає значних обчислювальних ресурсів. Перш ніж застосувати сигнатурний пошук до перехопленого трафіку, система здійснює його протокольну нормалізацію (*Protocol Decoding*) [20]. Сучасні кіберзлочинці активно використовують техніки

обходу (*Evasion*), навмисно спотворюючи запити на прикладному рівні – наприклад, застосовуючи шістнадцяткове кодування (*URL Encoding*), змішування регістрів або впровадження зайвих символів у *HTTP*-запити [1, 14]. Звичайний текстовий пошук не зможе виявити таку загрозу, оскільки її бітове подання відрізняється від еталонної сигнатури. Модуль нормалізації системи *DPI* попередньо декодує ці запити, приводить їх до стандартизованого, уніфікованого вигляду і лише після цього передає в аналітичне ядро. Здатність системи розпізнавати понад сотню прикладних протоколів (*HTTP*, *DNS*, *TLS*, *SMB*, *SSH*) дозволяє не лише шукати шкідливий код, але й витягувати метадані сесій: наприклад, ідентифікувати версію браузера (*User-Agent*), доменне ім'я, сертифікати шифрування або імена файлів, що передаються мережею [8, 21].

Для забезпечення високої швидкодії під час порівняння транзитних потоків із десятками тисяч актуальних сигнатур використовуються складні мультишаблонні математичні алгоритми пошуку. Одним із найефективніших та найпоширеніших у системах *DPI* є алгоритм Ахо-Корасік (*Aho-Corasick*) [21]. На відміну від класичних методів, які перевіряють текст на збіг із кожним правилом по черзі, цей алгоритм будує в оперативній пам'яті сервера єдиний детермінований скінченний автомат (складне дерево станів) на основі всіх існуючих сигнатур. Це дозволяє аналітичному рушію сканувати корисне навантаження транзитного пакета лише один раз і миттєво знаходити всі можливі збіги з будь-якими правилами у базі даних. Саме використання таких оптимізованих алгоритмів дозволяє системам виявлення вторгнень аналізувати гігабітні інформаційні потоки в реальному часі без втрати пакетів (*Packet Drop*) [14, 21].

Практична цінність технології *DPI* найбільш яскраво розкривається під час ідентифікації нестандартних мережевих аномалій та атак, спрямованих на вичерпання ресурсів. Класичним прикладом є виявлення *DoS*-атаки типу *ICMP Flood* з використанням пакетів аномально великого розміру. Звичайний маршрутизатор не бачить різниці між легітимним діагностичним запитом *Echo Request* (стандартний розмір якого становить 64 байти) та шкідливим запитом,

штучно збільшеним до 1000 байт задля перевантаження буферів цільового сервера [18]. Завдяки *DPI*-можливостям рушій створює та застосовує правила, які інспектують внутрішні метрики корисного навантаження – зокрема параметр довжини даних (*dsize*). Якщо цей параметр перевищує визначений безпечний поріг (наприклад, *dsize*:>500), система безпомилково класифікує трафік як атаку та генерує тривогу пріоритетного рівня. Здатність системи заглядати всередину пакета і вимірювати його корисне навантаження є ключовою перевагою проактивного моніторингу над традиційною фільтрацією [11, 14].

Незважаючи на високу ефективність систем глибокого аналізу, їхнє ізольоване використання має певні експлуатаційні обмеження. У межах лабораторних стендів або малих мереж результати роботи *IDS* зазвичай записуються у локальні текстові журнали (наприклад, файл *fast.log*), що дозволяє адміністратору вручну відстежувати інциденти в режимі реального часу. Проте у масштабах реального підприємства, де генеруються тисячі або мільйони подій на добу, ручне опрацювання розрізнених сирих логів стає фізично неможливим. Настає ефект «втоми від тривог» (*Alert Fatigue*), коли критично важливі попередження губляться у масиві менш значущих інформаційних повідомлень. Для вирішення цієї проблеми перспективним напрямком розвитку будь-якої архітектури безпеки є інтеграція сенсорів *IDS* із системами централізованого управління інформацією та подіями безпеки (*Security Information and Event Management*, *SIEM*) [1, 5].

Система класу *SIEM* є стратегічним ядром сучасного центру управління кібербезпекою (*Security Operations Center*, *SOC*). Її головна мета полягає в агрегації, нормалізації та кореляції даних із різноманітних джерел: маршрутизаторів, комутаторів, міжмережевих екранів, антивірусного програмного забезпечення, контролерів домену та систем виявлення вторгнень [1, 5]. Для того щоб *SIEM* могла автоматизовано обробляти лог-файли *IDS*, остання повинна генерувати події у структурованому, машиночитному форматі. Сучасні рушії, зокрема *Suricata*, використовують для цього формат розширюваної реєстрації подій – *Extensible Event Format* (*eve.json*). На відміну

від плоских текстових файлів, кожна подія у форматі *JSON* структурована у вигляді пар «ключ-значення», що містять вичерпну інформацію про інцидент: точні мітки часу, *IP*-адреси, порти, геолокацію відправника, номер спрацьованої сигнатури, категорію загрози та сам фрагмент перехопленого корисного навантаження [20, 21].

Процес інтеграції здійснюється за допомогою легкоагованих програмних агентів (наприклад, *Filebeat*), які встановлюються безпосередньо на сервер *IDS*. Ці агенти в реальному часі зчитують нові записи з файлу *eve.json* і безпечно передають їх по мережі до центрального аналітичного ядра *SIEM* – найчастіше до стеків *ELK* (*Elasticsearch*, *Logstash*, *Kibana*) або комплексних рішень з відкритим вихідним кодом на кшталт *Wazuh*. Потрапляючи в *SIEM*, сирі дані проходять етап нормалізації – приведення до єдиного стандарту, після чого починає працювати механізм крос-кореляції. Кореляція дозволяє системі автоматично знаходити неочевидні взаємозв'язки між різними подіями. Наприклад, *SIEM* здатна зіставити тривогу *IDS* про сканування мережі (*TCP SYN Scan*) із журналом маршрутизатора про множинні відхилення пакетів (*ACL Deny*) та логом контролера домену про серію невдалих спроб авторизації [10, 14]. Зібравши ці три розрізнені факти, система формує єдиний комплексний інцидент безпеки, що свідчить про цілеспрямовану атаку на інфраструктуру [1, 5].

Використання технологій *DPI* у симбіозі зі стеком *SIEM* трансформує сирі текстові попередження на візуально зрозумілі, інтерактивні дашборди. Аналітики з кібербезпеки отримують можливість відстежувати географічні карти джерел атак, будувати трендові графіки мережевого навантаження, швидко фільтрувати події за ступенем критичності та формувати детальні звіти для керівництва. Таким чином, технології глибокого аналізу пакетів та централізованого управління подіями є завершальними компонентами побудови комплексної системи захисту корпоративної мережі [5]. Перехід від локального перехоплення трафіку до його структурованого експорту та кореляції забезпечує можливість масштабування розроблених рішень від невеликих сегментів підприємства до рівня глобальних розподілених корпоративних інфраструктур,

створюючи надійне підґрунтя для безперервного та проактивного управління ризиками інформаційної безпеки [12, 13, 15].

# 3 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ЗАХИЩЕНОЇ МЕРЕЖІ В *PNETLAB*

## 3.1 Архітектура корпоративної мережі в *PNETLab*

На першому етапі практичної реалізації було розгорнуто топологію корпоративної мережі у віртуальному середовищі емуляції *PNETLab*. Вибір даного середовища зумовлений тим, що воно дозволяє використовувати реальні образи операційних систем мережевого обладнання (*Cisco IOL*), що гарантує максимальну наближеність результатів тестування до умов роботи реального апаратного забезпечення.

Спроєктована архітектура базується на класичній дворівневій ієрархічній моделі *Cisco*, яка оптимізована для підприємств середнього масштабу. Графічне представлення розробленої схеми зв'язків та розташування ключових інфраструктурних вузлів наведено на Рисунку 3.1. Як можна бачити на наведеній схемі, архітектура включає рівень ядра (*Core Layer*) та рівень доступу (*Access Layer*).

Основними вузлами розробленої інфраструктури, що відображені на Рисунку 1, є:

1. Прикордонний маршрутизатор (*Edge-Router*): Виконує роль шлюзу до глобальної мережі Інтернет, забезпечує трансляцію мережевих адрес (*NAT*) та фільтрацію зовнішнього трафіку:
2. Центральний комутатор ядра (*Core-Switch*): Виконує функції маршрутизації між віртуальними локальними мережами (*Inter-VLAN Routing*), виступає шлюзом за замовчуванням для всіх внутрішніх підмереж та забезпечує дзеркалювання трафіку для системи безпеки;
3. Комутатор доступу (*Access-Switch*): Забезпечує фізичне підключення кінцевих пристроїв користувачів та агрегує їхній трафік для передачі на рівень ядра;

4. Сервер безпеки (*Suricata-IDS*): Вузол на базі ОС *Ubuntu Server*, який виконує глибокий аналіз пакетів (*DPI*) та виявлення аномалій у режимі реального часу.

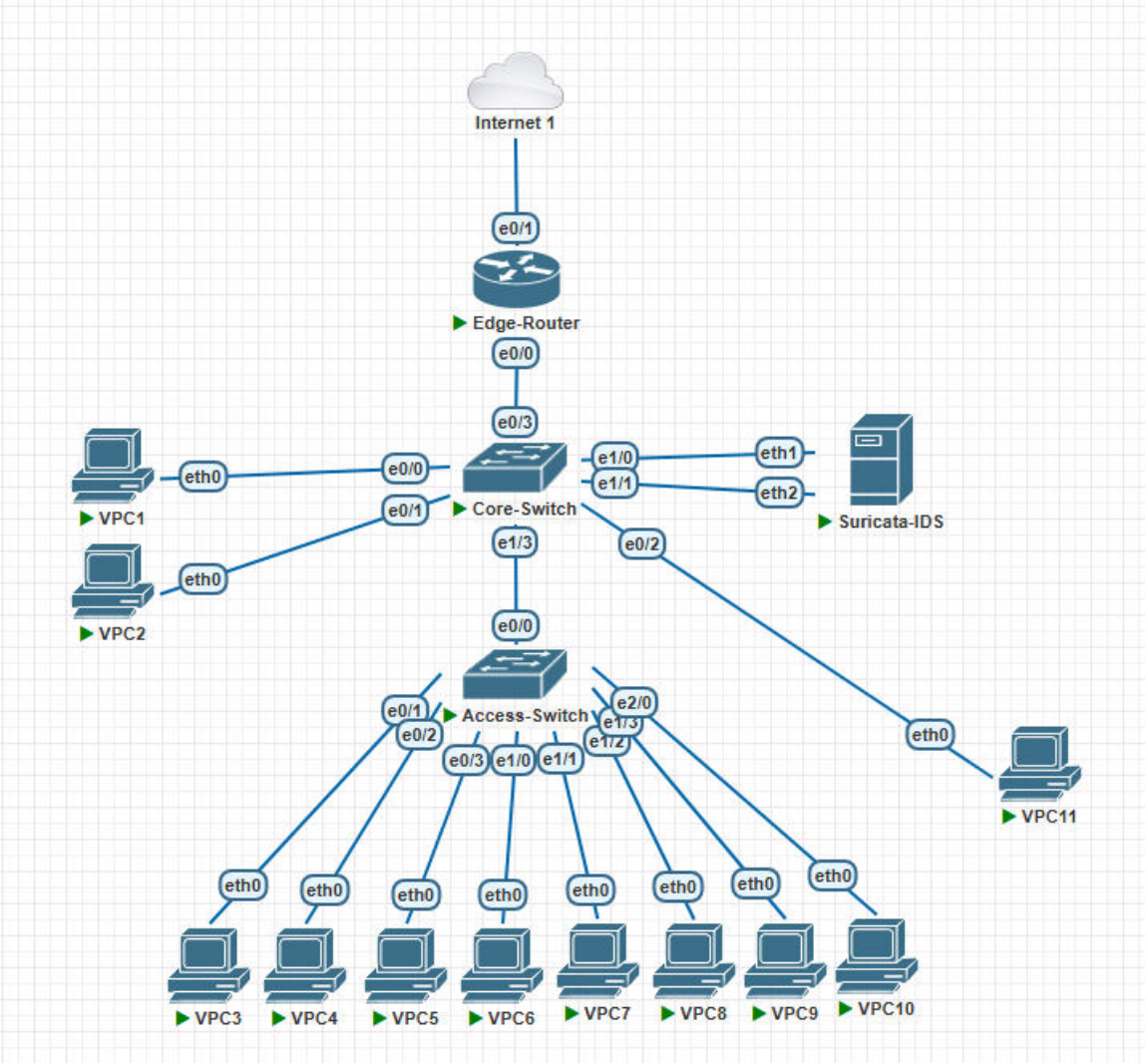


Рисунок 3.1 – Топологія корпоративної мережі у віртуальному середовищі емуляції *PNETLab*

Для забезпечення структурованості та керованості мережі було розроблено єдиний план *IP*-адресації. Логічна сегментація реалізована за допомогою технології віртуальних локальних мереж (*VLAN*). Розподіл адресного простору між усіма елементами, які було продемонстровано раніше на Рисунку 1, детально наведено у Таблиці 3.1.

Таблиця 3.1 – План *IP*-адресації корпоративної мережі

| Пристрій | Інтерфейс/*VLAN* | Логічний сегмент | *IP*-адреса/Маска | Шлюз (*Gateway*) |
|---|---|---|---|---|
| *Edge-Router* | *Ethernet* 0/1 | Зв'язок з Провайдером | *DHCP* | Від провайдера |
| *Edge-Router* | *Ethernet* 0/0 | *Transit* (*VLAN* 99) | 10.0.0.1 / 252 | - |
| *Core-Switch* | *VLAN* 99 | *Transit* (*VLAN* 99) | 10.0.0.2 / 252 | 10.0.0.1 |
| *Core-Switch* | *VLAN* 10 | *Admins* (Адміністратори) | 192.168.10.1 / 24 | - |
| *Core-Switch* | *VLAN* 20 | *Users* (Користувачі) | 192.168.20.1 / 24 | - |
| *Core-Switch* | *VLAN* 30 | *Servers* (Сервери) | 192.168.30.1 / 24 | - |
| *VPC* 1, 2 | eth0 | *Admins* (*VLAN* 10) | 192.168.10.10-11 / 24 | 192.168.10.1 |
| *VPC* 3-10 | eth0 | *Users* (*VLAN* 20) | 192.168.10.10-17 / 24 | 192.168.20.1 |
| *VPC* 11 | eth0 | *Servers* (*VLAN* 30) | 192.168.30.10 / 24 | 192.168.30.1 |
| *Suricata-IDS* | eth1 (*Mgmt*) | *Admins* (*VLAN* 10) | 192.168.10.100 / 24 | 192.168.10.1 |

Такий підхід до адресації мінімізує розмір широкомовних доменів, що знижує навантаження на комутаційне обладнання та є першим етапом побудови захищеної мережі.

## 3.2 Налаштування сегментації та політик доступу

Сегментація мережі є критично важливим механізмом кібербезпеки, оскільки вона обмежує поширення загроз (наприклад, мережевих черв'яків або *ARP-spoofing* атак) межами одного відділу.

На комутаторі *Core-Switch* було створено логічні інтерфейси (*SVI*) для кожного *VLAN*, які виконують роль маршрутизаторів між відділами. Поточний стан розподілу цифрових ідентифікаторів *VLAN* та їхніх назв у базі даних комутатора відображено на Рисунку 3.2. Як підтверджує верифікація конфігурації (Рисунок 3.2), порти пристрою успішно ініціалізовані та розподілені між відповідними технологічними підмережами підприємства.

```
Core-Switch#show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Et1/1, Et1/2, Et2/0, Et2/1
                                                Et2/2, Et2/3
10   Admins                           active    Et0/0, Et0/1, Et1/0
20   Users                            active
30   Servers                          active    Et0/2
99   Transit                          active    Et0/3
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
```

Рисунок 3.2 – Результат перевірки розподілу *VLAN* на комутаторі ядра (*Core-Switch*)

Зв'язок між комутатором доступу (*Access-Switch*) та ядром мережі здійснюється через магістральний порт (*Trunk*). Для його налаштування було використано протокол інкапсуляції *IEEE* 802.1*Q*. Параметри функціонування даного з'єднання наведено на Рисунку 3.3. Згідно з результатами перевірки (Рисунок 3.3), магістральний порт Et0/0 на комутаторі доступу коректно переведений в режим «*trunking*» і дозволяє безперешкодну передачу тегованого трафіку користувацького *VLAN* 20 до рівня ядра.

```
Access-Switch>show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Et0/0       on               802.1q         trunking      1

Port        Vlans allowed on trunk
Et0/0       1-4094

Port        Vlans allowed and active in management domain
Et0/0       1,20

Port        Vlans in spanning tree forwarding state and not pruned
Et0/0       1,20
```

Рисунок 3.3 – Стан магістрального (транкового) з'єднання на комутаторі доступу (*Access-Switch*)

Для забезпечення доступу внутрішніх вузлів інфраструктури до зовнішніх інтернет-ресурсів на прикордонному маршрутизаторі було розгорнуто технологію *NAT* (*Network Address Translation*). Успішне підняття фізичних лінків та динамічне отримання зовнішньої *IP*-адреси від провайдера підтверджується виведенням стану інтерфейсів, яке наведено на Рисунку 3.4.

```
Edge-Router>show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
Ethernet0/0                10.0.0.1        YES manual up                    up
Ethernet0/1                10.0.137.68     YES DHCP   up                    up
Ethernet0/2                unassigned      YES unset  administratively down down
Ethernet0/3                unassigned      YES unset  administratively down down
NVI0                       10.0.137.68     YES unset  up                    up
```

Рисунок 3.4 – Стан мережевих інтерфейсів та отримання IP-адреси на прикордонному маршрутизаторі (*Edge-Router*)

Для захисту найвразливішої частини інфраструктури – мережі адміністраторів – від потенційно скомпрометованих комп'ютерів звичайних користувачів, було впроваджено жорстку політику контролю доступу. На *Core-Switch* було застосовано розширений список контролю доступу (*Extended ACL*), який фільтрує трафік на мережевому рівні моделі *OSI*. Правило *deny ip* 192.168.20.0 0.0.0.255 192.168.10.0 0.0.0.255 було застосовано на вхідний

інтерфейс віртуального шлюзу *VLAN* 20, що фізично відхиляє будь-які спроби ініціювати з'єднання з боку користувачів.

### 3.3 Налаштування моніторингу та журналювання

Для реалізації повноцінного активного моніторингу трафіку необхідно було забезпечити систему виявлення вторгнень (*IDS*) копіями всіх мережевих пакетів, які проходять через ядро мережі. У стандартному режимі роботи комутатор передає пакети лише на той порт, де знаходиться *MAC*-адреса отримувача. Щоб обійти це обмеження, на комутаторі *Core-Switch* було розгорнуто технологію дзеркалювання портів (*Switch Port Analyzer – SPAN*).

Поточний статус активності та параметри поточної сесії перехоплення інформаційних потоків проілюстровано на Рисунку 3.5. Як підтверджує звіт утиліти моніторингу (Рисунок 3.5), локальна сесія налаштована на копіювання як вхідного, так і вихідного трафіку (*Both*) з портів Et0/0-3 та Et1/3 (що повністю покриває внутрішній обмін даними та вихід в Інтернет). Усі скопійовані фрейми безумовно перенаправляються у виділений аналітичний порт Et1/1.

```
Core-Switch#show monitor session 1
Session 1
---------
Type                      : Local Session
Source Ports              :
    Both                  : Et0/0-3,Et1/3
Destination Ports       : Et1/1
    Encapsulation       : Native
```

Рисунок 3.5 – Конфігурація сесії дзеркалювання трафіку (*SPAN*) на комутаторі ядра

Зі сторони сервера (ОС *Ubuntu*) мережевий інтерфейс eth2 був переведений у так званий «нерозбірливий режим» (*Promiscuous mode*) за допомогою команди

*ip link set* eth2 *promisc on*. Це дозволяє мережевій карті сервера приймати та обробляти пакети, *MAC*-адреси призначення яких не належать самому серверу.

Для обходу обмежень підсистеми безпеки ядра *Linux* (*AppArmor*), яка за замовчуванням блокує доступ сторонніх програм до сирих сокетів, утиліту перехоплення трафіку було ізольовано та запущено під власним псевдонімом.

Валідація працездатності розгорнутого *SPAN*-каналу та підсистеми первинного збору даних відображена на Рисунку 3.6. Базове тестування підсистеми моніторингу (Рисунок 3.6) підтверджує абсолютну прозорість інфраструктури для аналітичних засобів. Під час генерації тестового трафіку комп'ютером користувача, сніфер, що прослуховує інтерфейс eth2 на сервері *IDS*, успішно перехопив весь ланцюжок запитів та відповідей між клієнтом та публічним сервером 8.8.8.8. Таким чином, інфраструктурна база для розгортання інтелектуального рушія *Suricata* була успішно підготовлена.

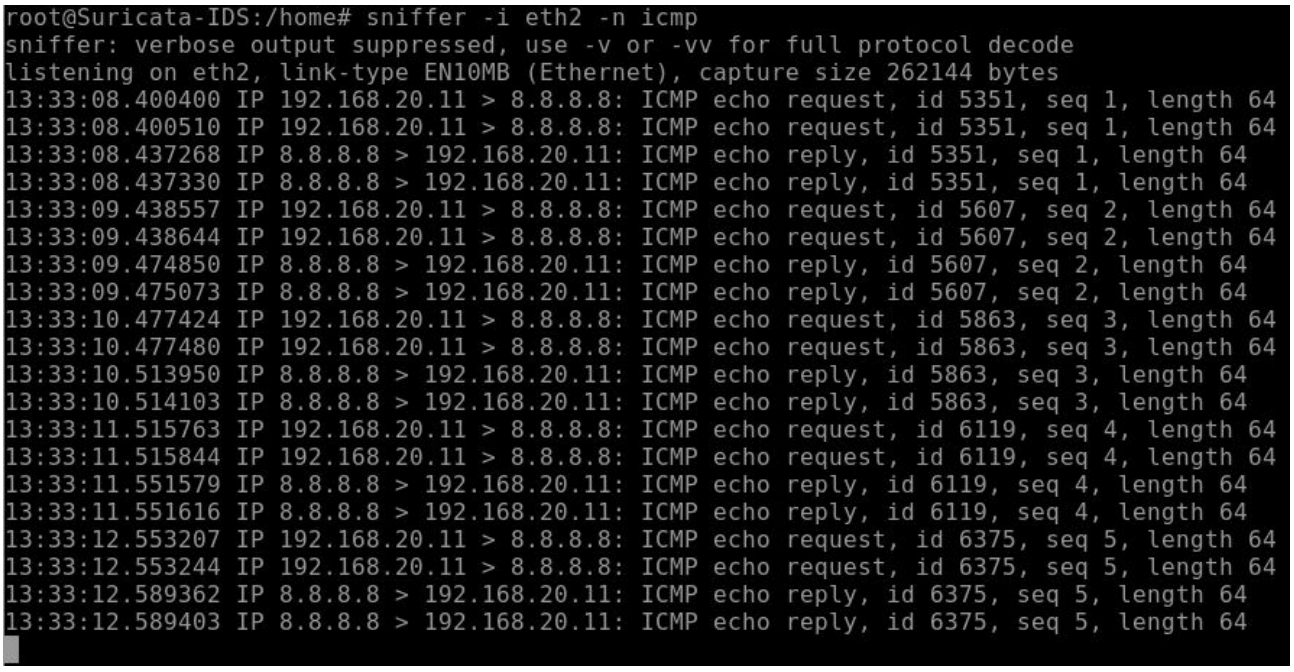


Рисунок 3.6 - Перехоплення дзеркальованого трафіку на мережевому інтерфейсі сервера *Suricata-IDS*

# 4 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ТА РЕКОМЕНДАЦІЇ ЩОДО ВПРОВАДЖЕННЯ

## 4.1 Сценарії перевірки працездатності

Для комплексної оцінки ефективності розробленої системи безпеки, а також для перевірки надійності захисту корпоративної інфраструктури, було сформульовано та реалізовано програмно-технічний план тестування. Головною метою цього етапу є симуляція реальних кіберзагроз та мережевих аномалій з метою верифікації точності спрацювання підсистеми захисту (спусків контролю доступу *ACL*) та підсистеми активного моніторингу (*IDS Suricata*).

Експериментальне дослідження було розділено на три взаємопов'язані сценарії:

Сценарій 1. Контроль легітимної зв'язності та верифікація апаратних політок безпеки. Передбачає перевірку базових маршрутів (доступ до мережі Інтернет) та підтвердження того, що створені списки контролю доступу (*Extended ACL*) успішно блокують спроби несанкціонованого міжвіланного обміну даними між користувачами та адміністраторами.

Сценарій 2. Симуляція *DoS*-атаки типу *ICMP Flood*. Сценарій спрямований на перевірку здатності системи *Suricata* аналізувати корисне навантаження пакетів (*payload*) та виявляти аномалії розміру даних за допомогою кастомних (власноруч створених) сигнатур.

Сценарій 3. Симуляція прихованої розвідки та сканування мережі. Відтворює дії зловмисника, який намагається виконати *TCP*-сканування закритих портів у мережі адміністраторів з боку зовнішнього інтерфейсу (маршрутизатора), і перевіряє здатність системи моніторингу виявляти порушення на рівні заголовків транспортного протоколу.

Для реалізації Сценаріїв 2 та 3 у фоновому режимі сервера *Suricata* було створено та активовано спеціалізовані правила (сигнатури). Перша сигнатура налаштована на триггер за умовою *dsize*:>500, що змушує систему піднімати

тривогу рівня *CRITICAL* у разі виявлення *ICMP*-пакетів, розмір яких перевищує стандартний. Друга сигнатура була орієнтована на перехоплення *TCP*-пакетів із встановленим прапорцем *SYN* (*flags*:*S*), які спрямовані на захищений адресний простір *VLAN* 10.

## 4.2 Аналіз результатів тестування

Практична реалізація розроблених сценаріїв дозволила отримати вичерпні результати, які були зафіксовані в системних журналах обладнання та файлах журналювання *IDS*.

Під час виконання Сценарію 1 було підтверджено повну ізоляцію критично важливих вузлів. При спробі кінцевого пристрою з підмережі користувачів (192.168.20.10) надіслати *ICMP*-запит на адресу адміністратора (192.168.10.10), центральний комутатор миттєво заблокував цей трафік на рівні інтерфейсу *SVI*.

Результати перевірки доступності та спрацювання *ACL* наведено на Рисунку 4.1. Як демонструє ілюстрація, запити до Інтернету (*ping* 8.8.8.8) та серверного сегмента (*ping* 192.168.30.10) проходять успішно, тоді як спроба підключення до адміністратора повертає повідомлення «*Communication administratively prohibited*» (код *ICMP type* 3, *code* 13). Це є прямим доказом того, що розроблена політика розмежування прав доступу функціонує бездоганно.

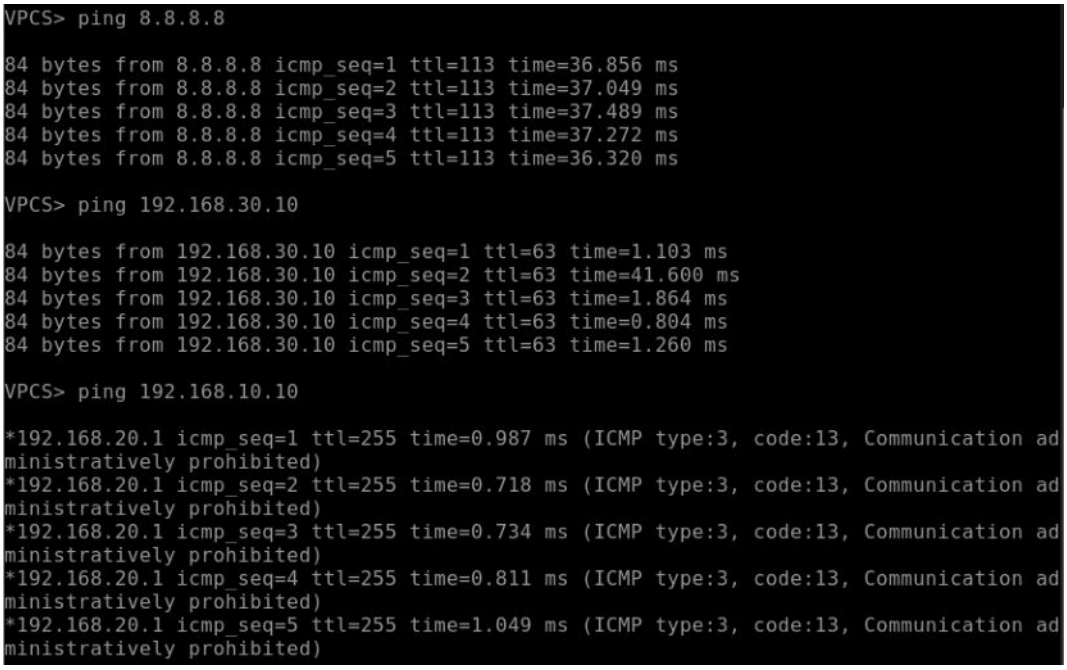


Рисунок 4.1 – Результати верифікації політик безпеки *ACL* (спрацювання блокування доступу)

Під час реалізації Сценарію 2 з комп'ютера користувача було запущено генерацію "важких" пакетів із розміром корисного навантаження в 1000 байт за допомогою команди *ping* 192.168.30.10 *-l* 1000. Процес відправки шкідливого трафіку відображено на Рисунку 4.2. На скриншоті чітко видно, що розмір сформованого пакета (разом із заголовками) становить 1028 байт, що суттєво перевищує стандартні ліміти локальної мережі.

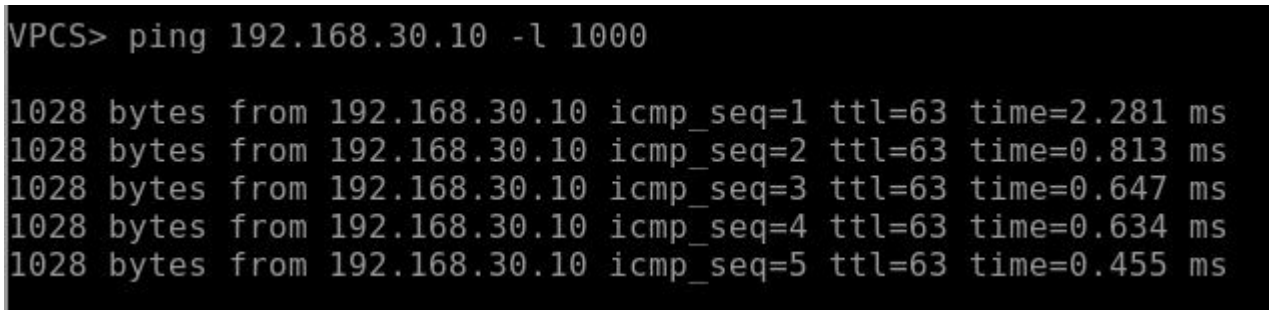


Рисунок 4.2 – Симуляція *DoS*-атаки (*ICMP Flood*) за допомогою пакетів аномального розміру

Цей трафік був успішно скопійований через *SPAN*-порт і переданий на аналіз рушію *Suricata*. Результат аналізу журналу тривог *Suricata* (/*var*/*log*/*suricata*/*fast*.*log*) у реальному часі наведено на Рисунку 4.3.

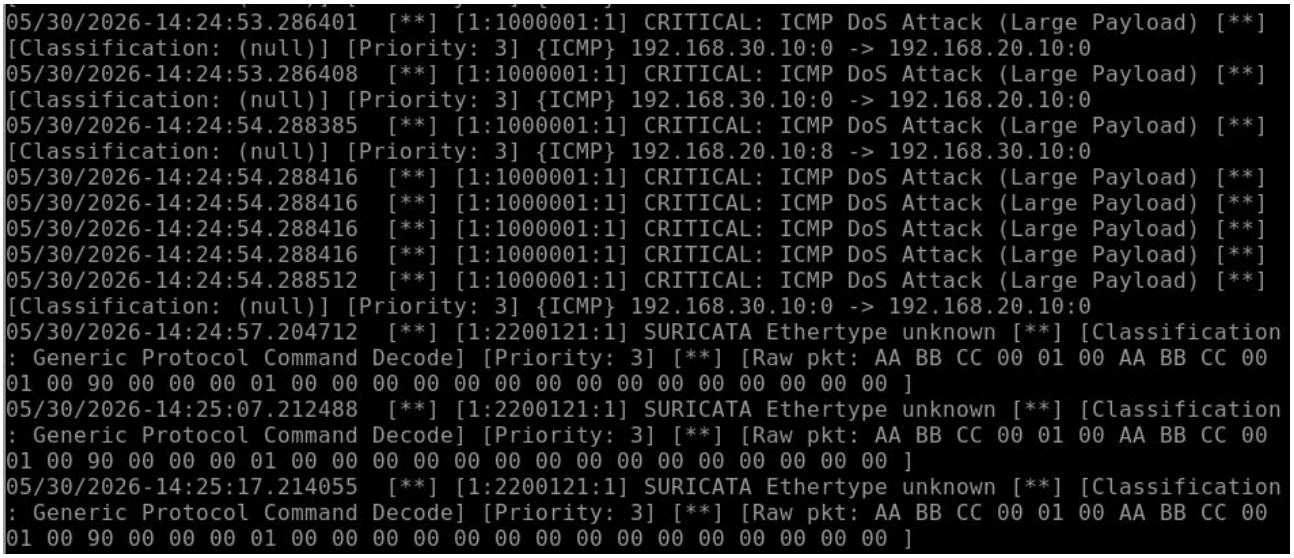


Рисунок 4.3 – Журнал тривог *Suricata* з фіксацією атаки *ICMP DoS Attack*

Як демонструє Рисунок 4.3, розроблена кастомна сигнатура відпрацювала зі 100% точністю. В логах зафіксовано серію повідомлень *CRITICAL*: *ICMP DoS Attack* (*Large Payload*). Важливою інженерною деталлю є те, що система чітко розпізнала як вхідний атакуючий пакет від джерела 192.168.20.10 до сервера 192.168.30.10, так і зворотну ехо-відповідь сервера. Це доводить, що система моніторингу коректно працює в двосторонньому (*bidirectional*) режимі.

Найбільший науковий інтерес викликали результати реалізації Сценарію 3 (спроба несанкціонованого підключення по протоколу *Telnet* з маршрутизатора 10.0.0.1 на порт 23 адміністратора 192.168.10.10). Отриманий лог системи безпеки наведено на Рисунку 4.4.

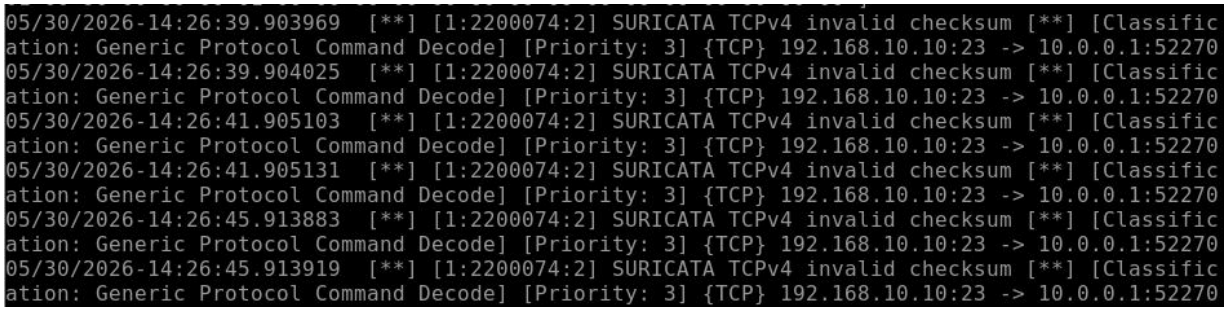


Рисунок 4.4 – Журнал тривог *Suricata* з виявленням протокольної аномалії TCPv4 *invalid checksum*

Замість стандартного кастомного повідомлення, система безпеки згенерувала серію випереджаючих попереджень типу *SURICATA TCPv*4 *invalid checksum* з пріоритетом *Priority*: 3, як це проілюстровано на Рисунку 4.4. Проведення глибокого інженерного аналізу цієї ситуації дозволило встановити причину такої поведінки системи.

У сучасних віртуалізованих інфраструктурах та середовищах емуляції за замовчуванням активна технологія *TCP Checksum Offloading*. Суть цієї технології полягає в тому, що операційна система процесу-відправника не витрачає ресурси центрального процесора на підрахунок контрольних сум заголовків пакетів, а делегує це завдання апаратній мережевій карті безпосередньо в момент виходу пакета у фізичний кабель. Оскільки наш сервер *Suricata* підключений до дзеркального порту (*SPAN*) всередині віртуального комутатора, він перехоплює копію пакета на проміжному етапі — коли реальний підрахунок контрольної суми мережевою картою ще не відбувся.

З точки зору класичної теорії захисту мереж, поява пакетів із недійсною контрольною сумою (*invalid checksum*) є відомою ознакою аномалії або спроби обходу систем захисту (техніка фаєрволу *Evasion*), коли хакери навмисно викривлюють заголовки для дезорієнтації систем *IDS*. Рушій *Suricata* виявив цю протокольну аномалію на нижчому рівні (*Generic Protocol Command Decode*) і миттєво підняв тривогу, заблокувавши подальший сигнатурний аналіз. Для нашого проєкту це є надзвичайно позитивним результатам, оскільки він

доводить, що розгорнута *IDS* здатна ідентифікувати підозрілу активність не лише за текстовими шаблонами, а й на основі глибокого аналізу цілісності мережевих протоколів.

## 4.3 Рекомендації щодо впровадження

На основі досвіду, отриманого під час проєктування, розгортання та тестування захищеної корпоративної мережі в середовищі *PNETLab*, було сформульовано практичні рекомендації для впровадження запропонованого рішення в реальну інфраструктуру підприємства:

1. Оптимізація продуктивності *IDS*: Оскільки дзеркалювання всього трафіку підприємства через *SPAN*-порт створює високе навантаження на процесор сервера *Suricata*, у реальному виробничому середовищі (*Production*) рекомендується вимкнути перевірку контрольних сум для внутрішніх інтерфейсів у файлі конфігурації *suricata.yaml* (параметр *checksum-validation*: *no*). Це дозволить уникнути хибних спрацювань *invalid checksum*, викликаних технологією *Offloading*, та звільнить до 30% обчислювальної потужності сервера для аналізу сигнатур;

2. Перехід на технологію *RSPAN*/*ERSPAN*: У разі масштабування мережі та появи нових корпусів чи філій, класичний локальний *SPAN* не зможе передавати трафік. Рекомендується впроваджувати технологію *ERSPAN* (*Encapsulated Remote SPAN*), яка дозволяє інкапсулювати дзеркальний трафік у *GRE*-тунелі та безпечно передавати його на центральний сервер *Suricata* через маршрутизовані канали L3;

3. Впровадження автоматичного блокування (Перехід до *IPS*): Наступним етапом розвитку побудованої системи є переведення *Suricata* з режиму моніторингу (*IDS*) у режим активного захисту (*IPS*). Для цього сервер безпеки необхідно підключити не через сліпий кабель, а в розрив трафіку (режим *Inline* за допомогою підсистеми *NFQUEUE*), що дозволить системі не просто фіксувати атаки в логах, а й автоматично скидати (*DROP*) шкідливі пакети.

# ВИСНОВКИ

В даній кваліфікаційній роботі:

− проведено аналіз архітектурних вразливостей пласких корпоративних мереж та класифікацію актуальних кіберзагроз (*ARP-spoofing*, *TCP SYN Scan*, *ICMP*/*TCP Flood*), а також обґрунтовано необхідність переходу до ешелонованої моделі захисту з використанням превентивних бар'єрів та проактивних систем виявлення вторгнень;

− розроблено масштабовану ієрархічну архітектуру корпоративної мережі та реалізовано жорстку логічну сегментацію інфраструктури за допомогою віртуальних локальних мереж (*VLAN*) і технологій міжсегментної апаратної фільтрації на базі розширених списків доступу (*Extended ACL*);

− розроблено та налаштовано алгоритми глибокого моніторингу трафіку шляхом інтеграції системи виявлення вторгнень (*IDS Suricata*) з використанням технології дзеркалювання портів (*SPAN*) у нерозбірливому режимі (*Promiscuous mode*);

− побудовано інформаційну модель та структуру захищеної мережі в середовищі апаратної емуляції *PNETLab*, а також проведено успішне експериментальне тестування розробленого комплексу шляхом симуляції цілеспрямованих кібератак.

Розроблений підхід до побудови комплексної системи захисту корпоративної мережі має наукову новизну, оскільки, на відміну від існуючих ізольованих рішень, дозволяє інтегрувати механізми захисту канального (*L2*) та мережевого (L3) рівнів із засобами глибокого аналізу пакетів (*DPI*) в єдину систему з урахуванням специфічних протокольних аномалій віртуалізованих середовищ (зокрема, механізмів *TCP Checksum Offloading*).

Розроблена архітектура, алгоритми налаштування комутаційного обладнання та кастомні сигнатури системи моніторингу мають практичну значимість. Результати роботи можуть бути використані при проєктуванні та модернізації захищених локальних мереж підприємств малого і середнього

бізнесу, при розгортанні центрів управління безпекою (*SOC*), а також в навчальному процесі зі спеціальності F5 «Кібербезпека та захист інформації».

Перспективи подальших досліджень у даному напрямку полягають у розширенні функціоналу розробленої системи шляхом переведення *IDS* у режим активного запобігання вторгненням (*IPS*) та інтеграції її журналів подій із системами централізованого управління інцидентами безпеки (*SIEM*).

# ПЕРЕЛІК ПОСИЛАНЬ

1. Безпека інформаційних систем. ДТЕУ. *URL*: *https*://*ur.knute.edu.ua* /*items*/8fb49cdc-6a75-4685-ac9e-d7082aa8e6e3/*full*

2. Комплексний аналіз програмного забезпечення для моделювання та емуляції комп'ютерних мереж: інструменти, застосування та майбутні напрями | Технічна інженерія. Технічна інженерія. *URL*: *https*://*ten.ztu.edu.ua* /*article*/*view*/307481

3. Про захист інформації в інформаційно-комунікаційних системах. Офіційний вебпортал парламенту України. *URL*: *https*://*zakon.rada.gov.ua*/*laws* /*show*/80/94-вр#*Text*

4. Про основні засади забезпечення кібербезпеки України. Офіційний вебпортал парламенту України. *URL*: *https*://*zakon.rada.gov.ua*/*laws*/*show*/2163-19#*Text*

5. Розмежування електронної комунікаційної системи мережі та електронної інформаційної системи в умовах забезпечення кібербезпеки організацій. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка». *URL*: *https*://*csecurity.kubg.edu.ua*/*index.php*/*journal*/*article*/*view*/1030

6. Учасники проєктів Вікімедіа. *IEEE* 802.1*Q* – вікіпедія. Вікіпедія. *URL*: *https*://*uk.wikipedia.org*/*wiki*/*IEEE*_802.1*Q*

7. Шаров А. *PNETLab* – простая и бесплатная среда для экспериментов с сетями и инфраструктурой. часть 1. Хабр.*URL*:*https*://*habr.com*/*ru*/*articles*/987332/

8. *Quickstart guide – Suricata* 9.0.0-*dev documentation*. *Suricata User Guide – Suricata* 8.0.5 *documentation*. *URL*: *https*://*docs.suricata.io*/*en*/*latest*/*quickstart.html*

9. *Installation – Suricata* 9.0.0-*dev documentation*. *Suricata User Guide – Suricata* 8.0.5 *documentation*. *URL*: *https*://*docs.suricata.io*/*en*/*latest*/*install.html*

10. *Configure IP access lists*. *Cisco*. *URL*: *https*://*www.cisco.com*/ *c*/*en*/*us*/*support*/*docs*/*security*/*ios-firewall*/23602-*confaccesslists.html*

11. *Cybersecurity framework*. *NIST*. *URL*: *https*://*www.nist.gov*/*cyberframework*

12. *DSpace ELAKPI* Репозитарій КПІ ім. Ігоря Сікорського. *URL*: *https*://*ela.kpi.ua*/*server*/*api*/*core*/*bitstreams*/2fc5d863-fb1a-4334-a33d-ceee718222ff/*content*

13. *DSpace*. *eNTUKhPIIR*. *URL*: *https*://*repository.kpi.kharkov.ua*/*bitstreams*/14e6f470-80c3-4005-9181-5ff8f40780d2/*download*

14. *IDS*/*IPS* системи для мережевої безпеки та як вони працюють. *IT Education Center Blog*. *URL*: *https*://*itedu.center*/*ua*/*blog*/*articles*/*ids-ips-systema-vyiavlennia-vtorhnen*/

15. *KNUCA Repository*. *URL*:*https*://*repositary.knuba.edu.ua*/*server*/*api*/*core*/*bitstreams*/003ab636-d14b-48a9-9654-342eff04dd7a/*content*

16. *Olkov E*. 5. *check point* на максимум. *IPS*. частина 1. Хабр. *URL*: *https*://*habr.com*/*ru*/*companies*/*tssolution*/*articles*/415083/

17. *PNETLab*: *lab is simple*. *PNETLab*: *Lab is Simple*. *URL*: *https*://*pnetlab.com*/*pages*/*documentation*

18. *RFC* 1180: *TCP*/*IP tutorial*. *IETF Datatracker*. *URL*: *https*://*datatracker.ietf.org*/*doc*/*html*/rfc1180

19. *Services S*. *Cisco IOS ACLs*. Хабр. *URL*: *https*://*habr.com*/*ru*/*articles*/154879/

20. *Suricata*/*doc*/*userguide at main OISF*/*suricata*. *GitHub*. *URL*: *https*://*github.com*/*OISF*/*suricata*/*tree*/*main*/*doc*/*userguide*

21. *Suricata User Guide – Suricata* 8.0.5 *documentation*. *Suricata User Guide – Suricata* 8.0.5 *documentation*. *URL*: *https*://*docs.suricata.io*/*en*/*suricata*-8.0.5/

# ДОДАТОК А

Налаштування Маршрутизатора (*Edge-Router*)

*enable*

*configure terminal*

*hostname Edge-Router*

*no service config*

! Підключення до Інтернету (отримання *IP* по *DHCP*)

*interface Ethernet* 0/1

*ip address dhcp*

*ip nat outside*

*no shutdown*

*exit*

! Підключення до локальної мережі

*interface Ethernet* 0/0

*ip address* 10.0.0.1 255.255.255.252

*ip nat inside*

*no shutdown*

*exit*

! Налаштування *NAT* для виходу в Інтернет

*access-list* 1 *permit* 192.168.0.0 0.0.255.255

*ip nat inside source list* 1 *interface Ethernet* 0/1 *overload*

! Зворотний маршрут до внутрішніх підмереж

*ip route* 192.168.0.0 255.255.0.0 10.0.0.2

*end*

*write memory*

Налаштування Ядра Мережі (*Core-Switch*)

```
enable
configure terminal
hostname Core-Switch
no service config

! Увімкнення маршрутизації
ip routing

! Створення VLAN
vlan 10
 name Admins
vlan 20
 name Users
vlan 30
 name Servers
vlan 99
 name Transit
exit

! Налаштування шлюзів для підмереж
interface vlan 10
 ip address 192.168.10.1 255.255.255.0
 no shutdown
interface vlan 20
 ip address 192.168.20.1 255.255.255.0
 no shutdown
interface vlan 30
 ip address 192.168.30.1 255.255.255.0
 no shutdown
```

```
interface vlan 99
 ip address 10.0.0.2 255.255.255.252
 no shutdown
exit

! Лінк до Роутера
interface Ethernet 0/3
 switchport mode access
 switchport access vlan 99
 no shutdown
 exit

! Транковий лінк (Trunk) до Access-Switch
interface Ethernet 1/3
 switchport trunk encapsulation dot1q
 switchport mode trunk
 no shutdown
 exit

! Порти Адміністраторів (VPC1, VPC2)
interface range Ethernet 0/0 - 1
 switchport mode access
 switchport access vlan 10
 spanning-tree portfast
 no shutdown
 exit

! Порт Сервера (VPC11)
interface Ethernet 0/2
 switchport mode access
```

*switchport access vlan* 30
*spanning-tree portfast*
*no shutdown*
*exit*

! Порт управління *Suricata-IDS*
*interface Ethernet* 1/0
*switchport mode access*
*switchport access vlan* 10
*spanning-tree portfast*
*no shutdown*
*exit*

! Маршрут за замовчуванням в Інтернет
*ip route* 0.0.0.0 0.0.0.0 10.0.0.1
*end*
*write memory*

Налаштування Комутатора Доступу (*Access-Switch*)
*enable*
*configure terminal*
*hostname Access-Switch*
*no service config*

! Створення *VLAN* для користувачів
*vlan* 20
*name Users*
*exit*

! Транковий лінк (*Trunk*) до *Core-Switch*

```
interface Ethernet 0/0
 switchport trunk encapsulation dot1q
 switchport mode trunk
 no shutdown
 exit

! Налаштування портів для Користувачів (VPC3 - VPC5)
interface range Ethernet 0/1 - 3
 switchport mode access
 switchport access vlan 20
 spanning-tree portfast
 no shutdown
 exit

! Налаштування портів для Користувачів (VPC6 - VPC9)
interface range Ethernet 1/0 - 3
 switchport mode access
 switchport access vlan 20
 spanning-tree portfast
 no shutdown
 exit

! Налаштування порту для Користувача (VPC10)
interface Ethernet 2/0
 switchport mode access
 switchport access vlan 20
 spanning-tree portfast
 no shutdown
 exit
```

*end*

*write memory*

Налаштування Політики Безпеки (*ACL*) на *Core-Switch*

*configure terminal*

! Створення списку доступу

*ip access-list extended BLOCK_USERS*

*deny ip* 192.168.20.0 0.0.0.255 192.168.10.0 0.0.0.255

*permit ip any any*

*exit*

! Застосування правила на віртуальний шлюз користувачів

*interface vlan* 20

*ip access-group BLOCK_USERS in*

*exit*

*end*

*write memory*

Налаштування *IP*-адрес на комп'ютерах (*VPCS*)

На VPC1: *ip* 192.168.10.10 24 192.168.10.1 (*Enter*, потім *save*)

На VPC2: *ip* 192.168.10.11 24 192.168.10.1 (*Enter*, потім *save*)

Для Сервера (*VLAN* 30):

На VPC11: *ip* 192.168.30.10 24 192.168.30.1 (*Enter*, потім *save*)

Для Користувачів (*VLAN* 20):

На VPC3: *ip* 192.168.20.10 24 192.168.20.1 (*Enter*, потім *save*)

На VPC4: *ip* 192.168.20.11 24 192.168.20.1 (*Enter*, потім *save*) (аналогічно налаштувати VPC5-10, просто змінюючи останню цифру: 12, 13, 14...)

Налаштування дзеркалювання (*SPAN*) на *Core-Switch*

*enable*
*configure terminal*

! Копіюємо трафік від Адмінів, Сервера та Роутера
*monitor session* 1 *source interface Ethernet* 0/0 - 3 *both*

! Копіюємо трафік від Користувачів (з транкового порту)
*monitor session* 1 *source interface Ethernet* 1/3 *both*

! Відправляємо копії в "сліпий" кабель до *Suricata*
*monitor session* 1 *destination interface Ethernet* 1/1

*end*
*write memory*

Розгортання *Suricata-IDS*

# 1. Піднімаємо мережу управління та даємо доступ до Інтернету
*ip addr add* 192.168.10.100/24 *dev* eth1
*ip link set* eth1 *up*
*ip route add default via* 192.168.10.1
*echo* "*nameserver* 8.8.8.8" > /*etc*/*resolv.conf*

# 2. Встановлюємо залежності, сніфер та *Suricata*
*apt update*
*apt install software-properties-common curl tcpdump -y*
*add-apt-repository ppa*:*oisf*/*suricata-stable -y*
*apt update*
*apt install suricata -y*

# 3. Налаштовуємо "сліпий" порт для прослуховування (*Promiscuous mode*)

*ip link set* eth2 *up*

*ip link set* eth2 *promisc on*

*sed -i* 's/*interface*: eth0/*interface*: eth2/g' /*etc*/*suricata*/*suricata.yaml*

# 4. Обходимо захист ядра *Linux* (*AppArmor*) для нашого сніфера

*cp* /*usr*/*sbin*/*tcpdump* /*usr*/*bin*/*sniffer*

# 5. Оновлюємо бази хакерських сигнатур та запускаємо двигун *IDS*

*suricata-update*

*suricata -c* /*etc*/*suricata*/*suricata.yaml -i* eth2 *-D*

#6 Тест роботи

*sniffer -i* eth2 *-n icmp*

Створення власних правил на *Suricata-IDS*

# 1. Зупиняємо фоновий процес *Suricata* для оновлення правил

*pkill suricata*

# 2. Додаємо Правило №1: Виявлення *ICMP*-пакетів аномального розміру (більше 500 байт)

*echo* 'alert *icmp any any -> any any* (*msg*:"*CRITICAL*: *ICMP DoS Attack* (*Large Payload*)"; *dsize*:>500; *sid*:1000001; *rev*:1;)' >> /*var*/*lib*/*suricata*/*rules*/*suricata.rules*

# 3. Додаємо Правило №2: Виявлення спроби *TCP*-підключення до мережі Адміністраторів

*echo* 'alert *tcp any any* -> 192.168.10.0/24 (*msg*:"*WARNING*: *External TCP Scan to Admin VLAN*"; *flags*:*S*; *sid*:1000002; *rev*:1;)' >> /*var*/*lib*/*suricata*/*rules*/*suricata.rules*

# 4. Запускаємо *Suricata* знову

*suricata -c /etc/suricata/suricata.yaml -i* eth2 *-D*

Команда, яка буде в реальному часі показувати журнал тривог:
*tail -f /var/log/suricata/fast.log*

Для тесту вводимо у VPC3:
*ping* 192.168.30.10 *-l* 1000

*Edge-Router* підключення по протоколу *Telnet* (порт 23):
*telnet* 192.168.10.10

Якщо виникли проблеми з *Suricata-IDS* вводимо три команди по черзі:
# 1. Видаляємо "завислий" файл блокування
*rm /var/run/suricata.pid*

# 2. Запускаємо *Suricata* знову
*suricata -c /etc/suricata/suricata.yaml -i* eth2 *-D*

# 3. Відкриваємо журнал тривог і чекаємо на атаки
*tail -f /var/log/suricata/fast.log*