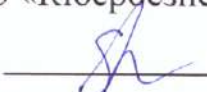


Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Проект підсистеми виявлення та протидії
ексфільтрації даних через ISMP-протокол»

Виконав: студент 4-го курсу групи КБ-21-2
спеціальності 125 «Кібербезпека»

 Роман ШАРКОВСЬКИЙ

Керівник: старший викладач кафедри КІ та КБ

 Олександра ПОКОТИЛО

Рецензент: Доцент кафедри КІ та КБ, к.т.н., доцент

 Юрій РОССІНСЬКИЙ

Житомир – 2025

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 125 «Кібербезпека»
Освітня програма «Кібербезпека»

ЗАТВЕРДЖУЮ
Завідувач
кафедри комп'ютерної
інженерії та кібербезпеки
 Андрій Єфіменко
3 березня 2025 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу бакалавра

Студент: Шарковський Роман Олександрович

Тема роботи: «Проект підсистеми виявлення та протидії ексфільтрації даних через ICMP-протокол»

Наказом університету від 3 березня 2025 р. №80/с

Термін здачі студентом закінченої роботи 27 червня 2025 р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

Об'єкт дослідження: процес забезпечення мережевої безпеки, виявлення мережевих вторгнень в каналах прихованої передачі даних.

Предмет дослідження: методи виявлення та блокування ексфільтрації даних, які використовують ICMP-протокол як канал витоку.

Обладнання та програмне забезпечення:

1. GNS3
2. Oracle VM VirtualBox
3. WireShark
4. Suricata
5. Prometheus
7. NetData
8. OS KaliLinux

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Покотило О. А.	30.05.2025	30.05.2025
2	Покотило О. А.	16.06.2025	16.06.2025
3	Покотило О. А.	17.06.2025	17.06.2025

Студент _____ Роман ШАРКОВСЬКИЙ

Керівник _____ Олександра ПОКОТИЛО

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання	03.03.2025	
2	Аналіз літературних джерел	10.03.2025	
3	Аналіз роботи ISMP-протоколу та методів ексфільтрації	17.06.2025	
4	Аналіз систем виявлення та протидії ексфільтрації з відкритим кодом	17.06.2025	
5	Проектування мережевої інфраструктури	18.06.2025	
7	Проектування роботи системи виявлення та протидії ексфільтрації	18.06.2025	
8	Налаштування мережевих пристроїв та системи виявлення та протидії ексфільтрації	18.06.2025	
9	Проведення тестування роботи системи виявлення та протидії ексфільтрації	19.06.2025	
10	Написання звіту	19.06.2025	
11	Отримання відгуку та рецензії		
12	Захист	30.06.2025	

Студент _____ Роман ШАРКОВСЬКИЙ

Керівник _____ Олександра ПОКОТИЛО

АНОТАЦІЯ

Пояснювальна записка до курсової роботи на тему «Проект підсистеми виявлення та протидії ексфільтрації даних через ICMP-протокол» складається з вступу, трьох розділів, висновків та списку використаної літератури.

Основна текстова частина викладена на 81 сторінці друкованого тексту.

Пояснювальна записка має 5 сторінок додатків. Список використаних джерел містить 24 найменування і займає 2 сторінки. В роботі наведено 79 рисунків. Загальний обсяг роботи – 86 сторінок.

У першому розділі було розглянуто теоретичні відомості про функціонування протоколу ICMP та ICMP-трафіку, теоретичні відомості про методи прихованої передачі даних через ping/echo-запити, методи аналізу та фільтрації трафіку з прихованою передачею даних та проведено аналіз систем виявлення ICMP-ексфільтрації з відкритим кодом.

У другому розділі розглянуто основи та нюанси побудування мереж з використанням даного протоколу: розроблено топологію мережі, схему IP-адресації, проведено проектування роботи Suricata, налаштування правил на кінцевих користувачах та проектування симуляції атак.

У третьому розділі було проведене налаштування мережевих пристроїв, встановлено та налаштовано Suricata, Prometheus з BlackBox та NetData, проведено тестування роботи мережі з імітацією атак ексфільтрації ICMP-протоколу.

Висновок містить в собі результати виконаної роботи.

У додатках представлено конфігураційні файли мережевих пристроїв.

Ключові слова: ХОСТ, ДАТАГРАМА, МЕРЕЖА, МАРШРУТИЗАТОР, КОМУТАТОР, ЗАГОЛОВОК, ПОВІДОМЛЕННЯ, ICMP, МОНІТОРИНГ, ПАКЕТ, ПОВІДОМЛЕННЯ, ПРАВИЛО.

					ККІКБ.КР.Б - 125 - 21 - ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Шарковський Р.О.		30.06.25	Звіт з лабораторних робіт	Літ.	Арк.	Аркушів
Керівник		Покотило О.А.		30.06.25			4	86
Реценз.		Росинський Ю.М.		30.06.25		ФІКТ, Гр. КБ-21-2		
Н. контр.		Єфименко Н.А.		30.06.25				
Затверд.		Єфіменко А.А.		30.06.25				

ANNOTATION

This explanatory note consists of an introduction, three chapters, conclusions, and a list of references.

The main body of the text spans 81 pages of printed material.

The explanatory note also contains 5 pages of appendices. The list of references includes 24 sources and takes up 2 pages. The paper includes 79 figures. The total volume of the work is 86 pages.

The first chapter discusses the theoretical background of the ICMP protocol and ICMP traffic, theoretical methods of covert data transmission via ping/echo requests, techniques for analyzing and filtering traffic with hidden data transfer, and an overview of open-source ICMP exfiltration detection systems.

The second chapter addresses the principles and specific aspects of building networks using this protocol: the development of a network topology, an IP addressing scheme, the design of Suricata deployment, rule configuration on endpoint hosts, and simulation of attacks.

In the third chapter, network devices were configured, Suricata, Prometheus with BlackBox, and NetData were installed and set up, and testing of the network operation under simulated ICMP exfiltration attacks was conducted.

The conclusion summarizes the results of the work.

The appendices contain configuration files for the network devices.

Keywords: HOST, DATAGRAM, NETWORK, ROUTER, SWITCH, HEADER, MESSAGE, ICMP, MONITORING, PACKET, MESSAGE, RULE.

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 АНАЛІЗ ПРОЦЕСІВ ЕКСФІЛЬТРАЦІЇ ICMP ПРОТОКОЛУ	10
1.1 Теоретичні відомості про функціонування протоколу ICMP та ICMP-трафіку	10
1.2 Теоретичні відомості про методи прихованої передачі даних через ping/echo-запити	18
1.3 Методи аналізу та фільтрації трафіку з прихованою передачею даних.....	21
1.4 Аналіз систем виявлення та протидії експільтрації ICMP-протоколу з відкритим кодом	22
Висновок до розділу	30
РОЗДІЛ 2 ПРОЄКТУВАННЯ СИСТЕМИ ВИЯВЛЕННЯ ТА НАПИСАННЯ ПРАВИЛ СИСТЕМИ ВИЯВЛЕННЯ ТА ПРОТИДІЇ ЕКСФІЛЬТРАЦІЇ ICMP-ПРОТОКОЛУ	31
2.1 Розробка топології мережі.....	31
2.2 Розробка системи IP-адресації	36
2.3 Проєктування роботи Suricata.....	41
2.4 Проєктування сервісів на кінцевих користувачах.	43
2.5 Проєктування симуляції атак з використанням експільтрації ICMP-протоколу	44
Висновки до розділу.....	45
РОЗДІЛ 3 НАЛАГОДЖЕННЯ ТА ТЕСТУВАННЯ СИСТЕМИ	46
3.1 Вибір обладнання для організації роботи мережі.....	46
3.2 Налаштування мережі	52
3.3 Встановлення Suricata	59
3.4 Встановлення Netdata та Prometheus	61
3.5 Налаштування системи виявлення та протидії експільтрації ICMP-протоколу	64
3.6 Тестування роботи системи.....	71
Висновки до розділу.....	77

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВИСНОВКИ.....	78
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	79
ДОДАТКИ.....	81

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ОС – операційна система

ПЗ – програмне забезпечення

ICMP – Internet Control Message Protocol

TCP – Transmission Control Protocol (Протокол управління передачею)

UDP – User Datagram Protocol (Протокол користувацьких датаграм)

VLAN – Virtual Local Area Network (Віртуальна локальна мережа)

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. У сучасних інформаційно-комунікаційних системах ICMP-протокол зазвичай використовується для діагностики мережі, але в руках зломисника він може бути застосований для прихованої ексфільтрації конфіденційної інформації за межі організації, оминаючи традиційні засоби захисту. Через зростання кількості кіберзагроз, пов'язаних з використанням легітимних мережевих протоколів у шкідливих цілях, важливим є створення ефективних рішень для виявлення та протидії.

Метою кваліфікаційної роботи є розробка підсистеми виявлення та протидії ексфільтрації даних через ICMP-протокол.

Визначена мета обумовлює наступні завдання:

- дослідити роботу ICMP-протоколу;
- дослідити механізми, які використовуються для ексфільтрації ICMP-протоколу;
- розробити проєкт підсистеми виявлення та протидії ексфільтрації ICMP-протоколу;
- реалізувати та налаштувати систему виявлення та протидії в тестовому середовищі;

Об'єктом дослідження є процес забезпечення мережевої безпеки, виявлення мережевих вторгнень в каналах прихованої передачі даних.

Предметом дослідження є методи виявлення та блокування ексфільтрації даних, які використовують ICMP-протокол як канал витоку.

У процесі виконання кваліфікаційної роботи було розроблено проєкт підсистеми виявлення та протидії ексфільтрації ICMP-протоколу. Складено схему інформаційно-комунікаційної мережі. Проведено моделювання атаки ICMP-ексфільтрації.

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ПРОЦЕСІВ ЕКСФІЛЬТРАЦІЇ ICMP ПРОТОКОЛУ

1.1 Теоретичні відомості про функціонування протоколу ICMP та ICMP-трафіку

ICMP, рідше ICMPv4 (Internet Control Message Protocol) – це допоміжний протокол у стеку протоколів TCP/IP, що використовується мережевими пристроями для надсилання операційної інформації та повідомлень про помилки, що сповіщають про успішність чи невдачу комунікації з іншою IP-адресою (успішність опрацювання датаграм) [1].

ICMP є невід’ємною частиною IP та описаний в RCF 792. На відміну від TCP та UDP, ICMP не є транспортним протоколом, а виконує службову функцію і є частиною мережевого рівня моделі OSI поруч з IPv4, IPv6, IGMP, IPSec та ін.

Під час використання ICMP немає гарантій, що датаграма буде доставлена або керуюче повідомлення повернеться до відправника, тому деякі датаграми можуть бути недоставлені без будь-якого повідомлення про їх втрату. У разі необхідності забезпечення надійного і безпечного зв’язку, протоколи вищого рівня, що використовують IP і його частину – ICMP, повинні реалізовувати власні процедури забезпечення надійності.

У змісті пакетів заголовка ICMP починається після заголовка IP та ідентифікується за номером протоколу, 1. ICMP-повідомлення має 8-байтовий заголовок та розділ з даними змінного розміру. Хоча загальний формат заголовка і відрізняється залежно від типу повідомлення, перші 4 байти є однаковими для усіх типів.

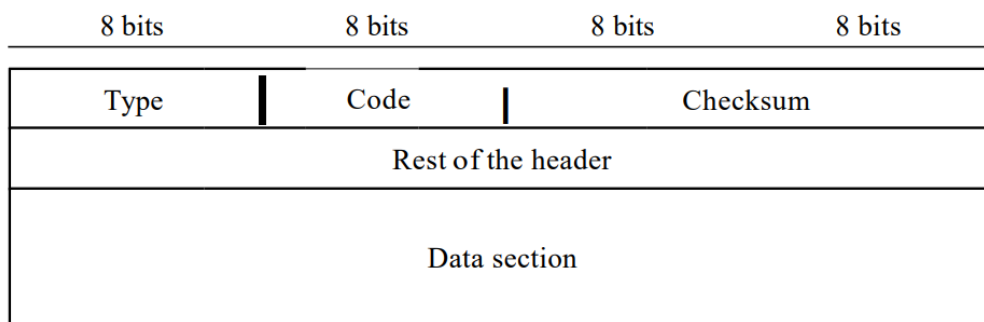


Рис. 1.1 - Загальна структура ICMP-повідомлень

Як показано на Рис. 1.2, перше поле «Тип» (Type) визначає тип ICMP-повідомлення, поле «Код» (Code) вказує причину конкретного типу, а останнім спільним полем є поле «Контрольна сума» (Checksum), що використовується для перевірки цілісності даних під час передавання по мережі. Решта заголовка є специфічною для кожного типу повідомлення. Розділ «Дані» (Data section) у повідомленнях про помилки містить інформацію для пошуку оригінального пакета, що містив помилку, а в повідомленнях запиту – додаткову інформацію залежно від типу запиту [6].

ICMP-повідомлення надсилаються в різних ситуаціях, яким відповідають типи повідомлення із цифрою у полі «Type» і діляться на дві групи: Destination Unreachable, Source Quench, Time Exceeded, Parameter Problems та Redirection, які повідомляють про помилки (див. Рис. 1.2.), та Echo, Echo Reply, Timestamp, Timestamp Reply, Information Request та Information Reply, які є діагностичними (див. Рис. 1.3.). Окремо можна виділити ще два види ICMP-повідомлень, що працюють з IPv6 та протоколом виявлення маршрутизатора IRDP (RFC 1256) – Router Solicitation та Router Advertisement, яким відповідають значення 10 та 9 у полі «Type», відповідно [4].

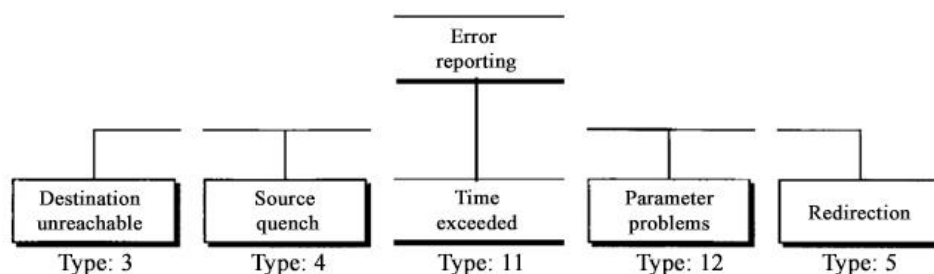


Рис. 1.2 - ICMP-повідомлення про помилки

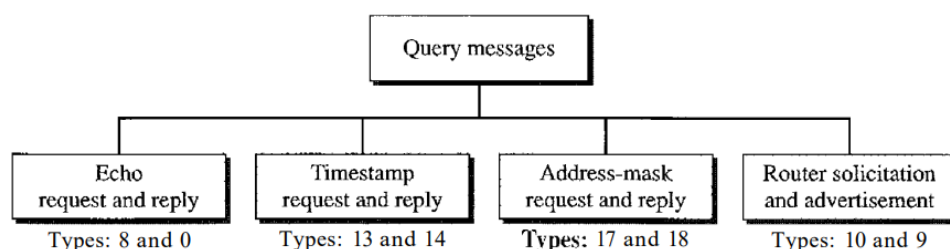


Рис. 1.3 - Діагностичні ICMP-повідомлення

Тепер окремо розглянемо кожен із типів ICMP-повідомлень з їх особливими параметрами та для чого вони можуть використовуватися:

– Пункт призначення недоступний (Destination Unreachable). У випадках, коли маршрутизатор не може прокласти маршрут датаграмі або не може доставити датаграму, вона відкидається і маршрутизатор або хост надсилає повідомлення про недоступність пункту призначення назад хосту-джерелу, який ініціював датаграму. Варто звернути увагу, що повідомлення про недоступність пункту призначення можуть бути створені як маршрутизатором, так і хостом-одержувачем.

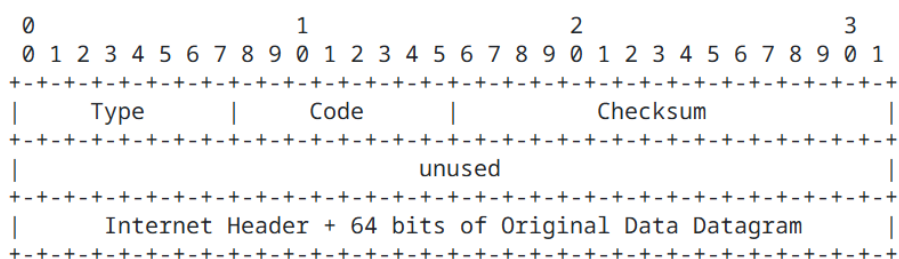


Рис. 1.4 - Загальна структура повідомлення «Destination Unreachable»

Зі структури повідомлення на Рис. 1.4 розберемо деякі поля. Цей тип повідомлення визначається параметром 3 в полі «Type». Поле «Code» може містити такі параметри: 0 – мережа недоступна, 1 – хост недоступний, 2 – протокол недоступний, 3 – порт недоступний, 4 – потрібна фрагментація, та встановлено прапорець DF та 5 – вихідний маршрут помилковий. Коди 0, 1, 4 та 5 зазвичай можуть бути отримані від шлюзу, а 2 та 3 – від хоста.

– Гасіння джерела (Source Quench). Через те, що IP-протокол не має вбудованого механізму контролю потоку в протоколі, це може створити серйозну проблему перенавантаження в роботі IP. Маршрутизатор та хост мають буфери обмеженого розміру для вхідних датаграм, які очікують на пересилання. У разі переповнення цього буфера маршрутизатор або хост не має іншого вибору, окрім як відкинути деякі датаграми, після чого він надсилає повідомлення Source Quench відправнику датаграми для інформування того, що датаграму відкинута, та попере-

дження про те, що десь на шляху існує перенавантаження і потрібно надсилати повідомлення повільніше. Від 2012 року Source Quench є застарілим методом сповіщення про перенавантаження через низку причин [7].

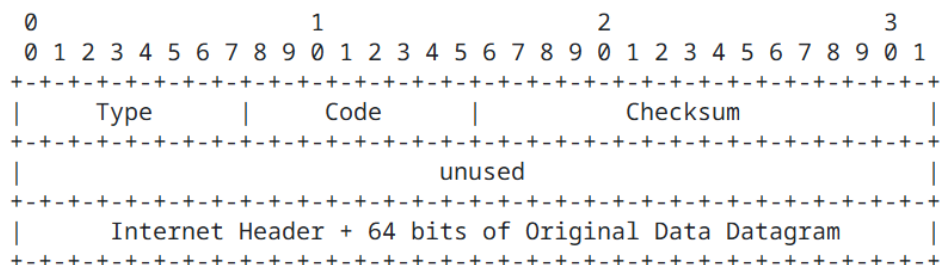


Рис. 1.5 - Загальна структура повідомлення «Source Quench»

Зі структури повідомлення на Рис. 1.5 розберемо деякі поля. Цей тип повідомлення визначається параметром 4 в полі «Type». Поле «Code» може містити лише параметр 0. Код 0 може бути отриманий від шлюзу або хоста.

– Перевищення часу очікування (Time exceeded). Через помилки в таблицях між двома маршрутизаторами, датаграми можуть нескінченно пересилатися між ними, або ходити по петлі. Для запобігання цьому кожна датаграма має поле «Lifetime», і з кожним прибуттям до маршрутизатора параметр цього поля зменшується на 1, а коли він досягає 0, маршрутизатор відкидає датаграму та відправляє повідомлення про те, що час життя датаграми закінчився. Цей тип повідомлення також використовується для сповіщення того, що не всі фрагменти, з яких складається повідомлення, прийшли у межах певного ліміту часу.

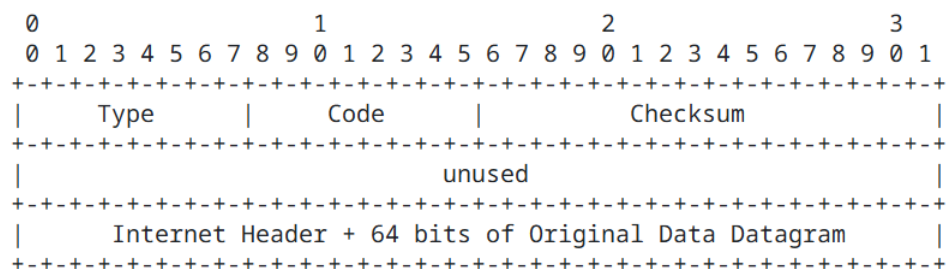


Рис. 1.6 - Загальна структура повідомлення «Time exceeded»

Зі структури повідомлення на Рис. 1.6 розберемо деякі поля. Цей тип повідомлення визначається параметром 11 в полі «Type». Поле «Code» може містити такі параметри: 0 – перевищено час життя під час передачі та 1 – перевищено час повторного складання фрагмента. Код 0 зазвичай може бути отриманий від шлюзу, а 1 – від хоста.

– Проблема з параметрами (Parameter Problems). Будь-яка невизначеність чи неточність у заголовку датаграми може створити серйозні проблеми під час її передачі через Інтернет. Якщо маршрутизатор або хост призначення виявляє помилкове або відсутнє значення в будь-якому полі датаграми, він відкидає її та надсилає повідомлення про цю проблему назад до джерела.

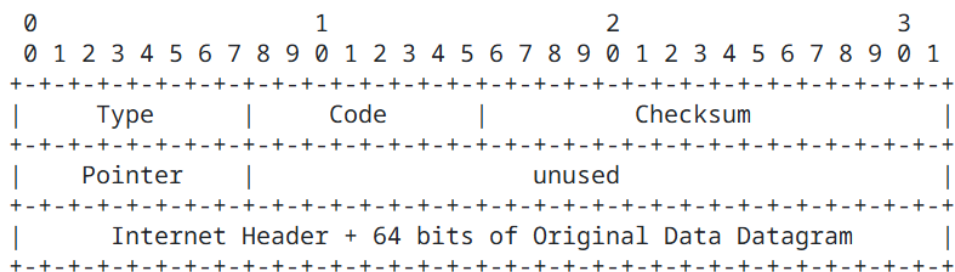


Рис. 1.7 - Загальна структура повідомлення «Parameter Problems»

Зі структури повідомлення на Рис. 1.7 розберемо деякі поля. Цей тип повідомлення визначається параметром 12 в полі «Type». Поле «Code» може містити параметр 0 у випадках, коли поле «Pointer» вказує на помилку. Поле «Pointer» описує октет, в якому було виявлено помилку. Код 0 може бути отриманий від шлюзу або хоста. Для IPv6 поле Поле «Code» може містити також параметри 1 – відсутній обов’язковий параметр та 2 – виявлено помилкову довжину параметра.

– Перенаправлення (Redirection). Це повідомлення надсилається маршрутизатором до вихідного хоста для стимуляції ефективнішої маршрутизації, але маршрутизатор все одно пересилає оригінальний пакет до пункту призначення. Це дозволяє таблицям маршрутизації хоста залишатися малими, оскільки необхідно знати адресу лише одного маршрутизатора, навіть якщо цей маршрутизатор не надає найкращого шляху.

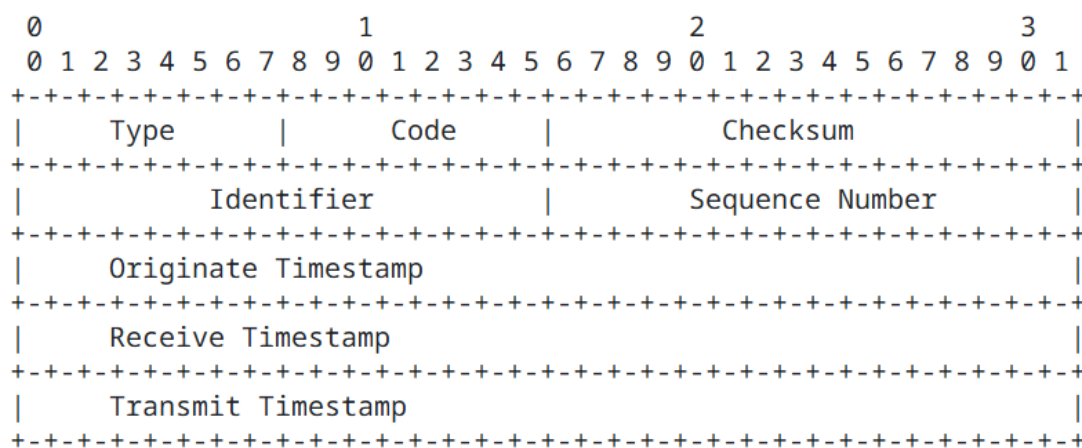


Рис. 1.9 - Загальна структура повідомлень «Timestamp Request and Reply»

Зі структури повідомлення на Рис. 1.9 розберемо деякі поля. Ці типи повідомлення визначається параметрами 13 та 14 в полі «Type» відповідно. Поле «Code» може містити лише параметр 0. У повідомленнях відсутнє поле «Unused». «Identifier» використовується для співставлення позначки часу запиту і відповіді, а «Sequence Number» – номер послідовності. «Originate Timestamp», «Receive Timestamp» та «Transmit Timestamp» визначають час надсилання, час прибуття запиту та час відправлення відповіді, відповідно, у мілісекундах з початку доби за UTC. Код 0 може бути отриманий від шлюзу або хоста.

– Запит та рекламування маршрутизатора (Router Solicitation and Advertisement). Ці типи повідомлень можуть допомогти в ситуаціях, коли хосту потрібно знати як відправити повідомлення до пункту призначення в іншій мережі. Хост може надсилати повідомлення запиту маршрутизатора, а маршрутизатор або маршрутизатори, які отримують повідомлення запиту, транслюватимуть свою маршрутну інформацію (про власну присутність та присутність сусідніх) за допомогою повідомлення рекламування маршрутизатора. Маршрутизатор також може періодично надсилати повідомлення реклами маршрутизатора, навіть якщо жоден хост не запитував. Для IPv4 у використанні IRDP та в IPv6 у використанні NDP ці повідомлення мають дещо різні поля, та в обох випадках у повідомленнях відсутні поля «Unused».

– Ехо-запити та ехо-відповіді (Echo Request and Reply). Ця пара найвідоміших ICMP-повідомлень може бути використана для виявлення проблем у мережі. Їх комбінація визначає, чи можуть два пристрої взаємодіяти один з одним на рівні IP-протоколу, адже ICMP-повідомлення інкапсулюються в IP-датаграми. Сьогодні більшість систем надають версію команди «ping», яка може створювати серію таких повідомлень, надаючи статистичну інформацію про стан зв'язку.

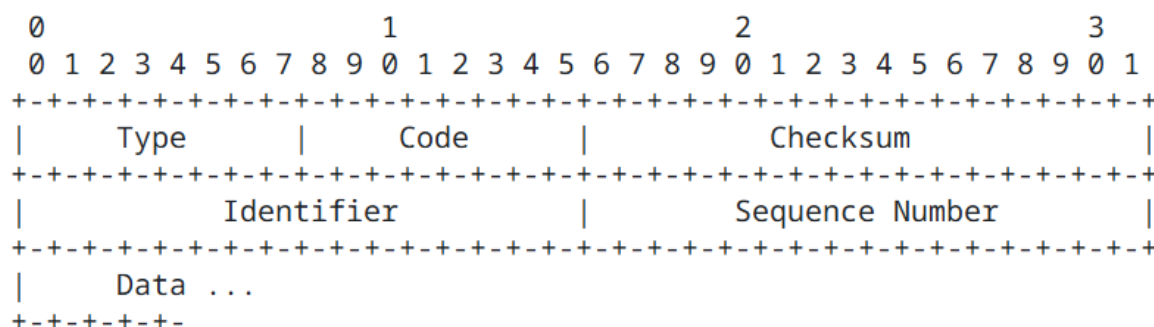


Рис. 1.10 - Загальна структура повідомлень «Echo Request and Reply»

Зі структури повідомлення на Рис. 1.10 розберемо деякі поля. Ці типи повідомлення визначається параметрами 8 та 0 в полі «Type» відповідно. Поле «Code» може містити лише параметр 0. Поле «Identifier» – це ідентифікатор, який дозволяє відслідковувати відповідь на запит, «Sequence Number» – це послідовний номер пакету, використовується для перевірки втрат, а поле «Data» – поле, що має змінну довжину та зазвичай не перевіряється міжмережевими екранами чи IDS/IPS системами, оскільки ICMP не використовує порти, і часто дозволений за замовчуванням для діагностики. Саме в це поле можна приховано вкладати текст, бінарні дані або навіть зашифровану інформацію для передачі назовні.

Також важливо пам'ятати декілька істинних тверджень про ICMP-протокол:

- 1) У відповідь на датаграму, що містить повідомлення про помилку ICMP, повідомлення про помилку ICMP не генерується задля запобігання рекурсії.
- 2) Для фрагментованої датаграми, яка не є першим (нульовим) фрагментом, повідомлення про помилку ICMP не генерується.

3) Для датаграми з груповою адресою повідомлення про помилку ICMP не генерується.

4) Для датаграми зі спеціальною адресою, такою як 127.0.0.0 або 0.0.0.0, повідомлення про помилку ICMP не генерується.

5) Позаяк ICMP-протокол знаходиться на мережевому рівні моделі OSI, з ним не асоціюють жоден конкретний порт.

Отже ICMP – це допоміжний протокол у стеку протоколів TCP/IP, що використовується мережевими пристроями для надсилання операційної інформації та повідомлень про помилки, що сповіщають про успішність чи невдачу комунікації з іншою IP-адресою. Можна виділити дві окремі групи ексфільтрації ICMP-повідомлень залежно від поля, яке використовується. Першою групою буде «Unused» (типи Destination Unreachable, Source Quench, Redirect та Time Exceeded). Другою групою буде «Data» (типи Echo Request and Reply).

1.2 Теоретичні відомості про методи прихованої передачі даних через ping/echo-запити

Дослідивши будову різних типів ICMP-повідомлень ми можемо дізнатися про методи прихованої передачі даних через цей протокол.

За визначенням атак від MITRA, описаному в TA0010, ексфільтрація – це збірка методів, які зловмисники можуть використовувати для викрадення даних з вашої мережі. Після збору даних зловмисники часто упаковують їх, щоб уникнути виявлення під час видалення. Це може включати стиснення та шифрування. Методи вилучення даних з цільової мережі зазвичай включають передачу їх через канал командування та управління або альтернативний канал, а також можуть включати обмеження розміру передачі [8].

Ексфільтрація через ICMP-протокол підпадає під окрему категорію T1048 «Ексфільтрація за альтернативним протоколом», коли зловмисники можуть викрасти дані, витягнувши їх через протокол, відмінний від протоколу існуючого каналу командування та управління. Дані також можуть бути надіслані до альтернативного мережевого розташування з головного сервера командування та управління.

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		18

Альтернативні протоколи включають FTP, SMTP, HTTP/S, DNS, SMB або будь-який інший мережевий протокол, який не використовується як основний канал командування та управління. Зловмисники також можуть вирішити зашифрувати та/або приховати ці альтернативні канали [9].

Програма ping, яка використовує ICMP може використовуватися для перевірки чи хост активний та відповідає. Хост-джерело надсилає повідомлення ICMP Echo-Request (тип: 8, код: 0), хост-одержувач, якщо активний, відповідає повідомленнями ICMP Echo-Reply. Ping встановлює поле ідентифікатора в повідомленнях Echo-Request та Echo-Reply та замінює порядковий номер на 0. Це число збільшується на 1 щоразу, коли надсилається нове повідомлення. Варто звернути увагу, що ping може обчислити час обміну повідомленнями, завдяки тому, що вставляє час відправлення в розділ даних повідомлення, і коли пакет прибуває, він віднімає час прибуття від часу відправлення, щоб отримати час обміну повідомленнями (RTT). Стандартним розміром ping пакету для ОС Windows є 32 байти, а ОС Linux – 64.

```

Командний рядок
Microsoft Windows [Version 10.0.18363.592]
(c) Корпорація Майкрософт (Microsoft Corporation), 2019. Усі права захищено.

C:\Users\Student>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=10ms TTL=255
Reply from 192.168.1.1: bytes=32 time=5ms TTL=255
Reply from 192.168.1.1: bytes=32 time=6ms TTL=255
Reply from 192.168.1.1: bytes=32 time=10ms TTL=255

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 5ms, Maximum = 10ms, Average = 7ms

C:\Users\Student>_

```

Рис. 1.11 - Команда ping у командному рядку ОС Windows

Програма ping надсилає повідомлення з порядковими номерами, що починаються з 0. Поле TTL (час життя) в IP-датаграмі, яка інкапсулює ICMP-повідомлення, встановлено на параметр, що означає, що пакет не може пройти більше заданої кількості переходів. Після завершення (переривання) ping друкує статистику

зондів. Вона повідомляє нам кількість надісланих пакетів, кількість отриманих пакетів, загальний час та мінімальне, максимальне та середнє значення RTT (час затримки).

Саме через це розповсюджене використання ping, найбільш поширеним типом ексфільтрації ICMP-протоколу є втручання в поля повідомлень Echo-Request та Echo-Reply. Розберемо декілька методів ексфільтрації ICMP-протоколу.

1) ICMP Payload (Data) Embedding (Вставлення корисних даних в ICMP) – метод, коли поле даних ICMP-датаграми (data або payload) містить довільну інформацію: команди (атаки IcmpDoor) чи частини конфіденційної інформації. Часто цей метод поєднується з інструментами тунелювання.

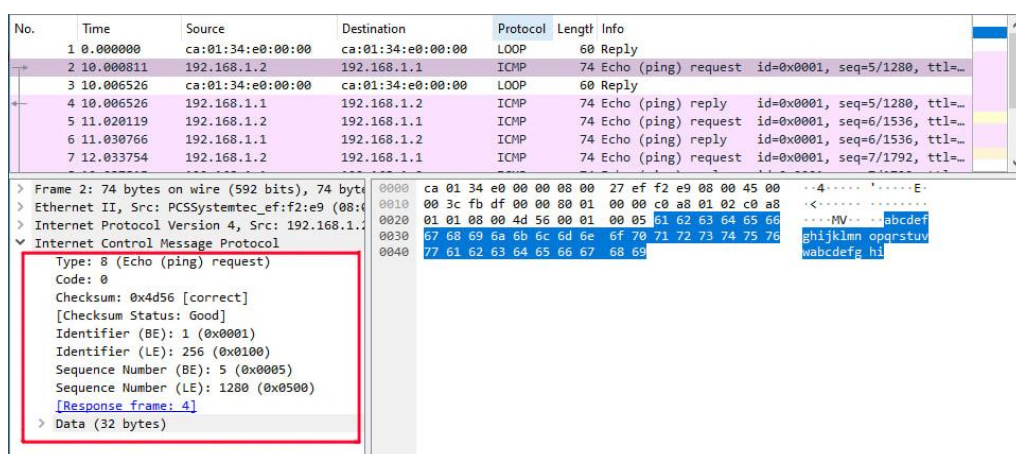


Рис. 1.12 - Поле «data» у ICMP-пакеті типу «Echo-Request» програми ping, перехоплений у WireShark

2) ICMP Header Manipulation (Маніпуляція заголовками ICMP) – це метод, що полягає в модифікації заголовків ICMP-пакетів, наприклад відправка одного байта інформації через кожен ICMP echo request, де значення Identifier поля відповідає ASCII-коду символу.

3) Timing Channel (Таймінг-канал) – це повільний метод, що полягає у кодуванні інформації та її передачі не через вставлення у поле пакета, а через задані затримки між пакетами, які кодують бінарні значення або символи.

4) Checksum/TTL Abuse (Аб'юз Контрольної суми або TTL) – це метод, схожий на попередній, але використовує не затримки між пакетами, а поле Checksum чи значення TTL.

5) Fragmentation-based Covert Channel (Прихований канал на основі фрагментації) – це метод у ході якого дані приховуються у фрагментах IP/ICMP-пакетів, які збираються лише на кінцевій системі, наприклад, зловмисник створює серію фрагментів ICMP-пакету з частинами прихованого повідомлення.

Реалізація цих методів може бути різноманітною: від звичайного відправлення ping із спеціальними параметрами без повного доступу до пристрою цілі до надскладних комбінованих атак для ексфільтрації на різні протоколи, від використання специфічних авторських бібліотек, до використання таких інструментів як Metasploit чи IcmpDoor, які потребують повного контролю над пристроєм жертви.

1.3 Методи аналізу та фільтрації трафіку з прихованою передачею даних

Перед тим, як перейти до аналізу систем виявлення та протидії з відкритим кодом, потрібно ознайомитися з методами аналізу та фільтрації трафіку з прихованою передачею даних, які вони використовують.

Можна виділити такі методи аналізу трафіку, які допоможуть у виявленні та протидії ексфільтрації ICMP-протоколу:

1) Статистичний аналіз трафіку. Перевірка розміру пакетів, частоти інтервалів та аномальної поведінки трафіку (частотності, циклічності), наприклад, виявлення постійного збільшеного розміру ICMP-пакетів з кожним виконанням ping чи виявлення стабільної затримки між echo-запитами.

2) Машинний аналіз трафіку. Навчання моделі на типовому трафіку, в результаті чого, будь-яке незначне відхилення трафіку буде помічено системою.

3) Аналіз таймінгів між пакетами. Полягає у журналюванні позначок часу усіх ICMP-пакетів з подальшою побудовою гістаграм та пошуком специфічних затримок.

4) Інспекція заголовків (Deep Packet Inspection). Полягає у виявленні нестандартних значення у полях «Identifier» та «Sequence Number» та аналіз вмісту поля data, що мають нестандартні типи даних.

5) Кореляційний аналіз логів. Поєднання логів з різних систем: SIEM (наприклад, Splunk, ELK). Виявлення підозрілої поведінки, що відбувається одразу в багатьох місцях, наприклад в трафіку та в процесах з підозрілими параметрами.

Можна виділити такі методи фільтрування трафіку, які допоможуть у виявленні та протидії ексфільтрації ICMP-протоколу:

1) Блокування або часткове фільтрування ICMP. Дозвіл тільки на echo-request до шлюзу з фіксованим розміром (наприклад, 64 байти). Блокування ICMP echo-reply ззовні до внутрішньої мережі.

2) Фільтрація за розміром та контентом. ICMP із великим полем даних можуть бути заблоковані. Блокування ping із текстовим вмістом або з нетиповими заголовками.

3) Обмеження пропускної здатності для ICMP. Throttle/Rate limiting для ICMP: наприклад, не більше 1 пакету/сек.

Найбільш ефективним рішенням буде імплементація не одного методу, а їх комбінації, що захистить не тільки від ексфільтрації ICMP-протоколу, а і ексфільтрації інших протоколів, адже зломисники зазвичай не використовують ексфільтрацію ICMP-протоколу як єдине джерело витоку інформації.

1.4 Аналіз систем виявлення та протидії ексфільтрації ICMP-протоколу з відкритим кодом

Snort — це відкрите мережеве IDS/IPS-рішення (система виявлення та запобігання вторгненням), яке аналізує мережевий трафік у реальному часі та порівнює його з набором правил або сигнатур, що описують підозрілу чи шкідливу активність. Його головна особливість — це гнучка система правил, яка дозволяє детально налаштовувати, що саме в трафіку вважати загрозою [10]. Snort був створений Марті Роешем у 1998 році та зараз підтримується компанією Cisco через підрозділ

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

Cisco Talos Intelligence Group. Він доступний під ліцензією GNU GPL і використовується в тисячах організацій, від академічних мереж до критично важливої інфраструктури. Однією з головних переваг Snort є його мова правил, яка дозволяє користувачам точно задавати умови, за яких трафік вважається шкідливим. Наприклад, можна створити правило, яке спрацює, коли в ICMP пакеті присутній текст команди на кшталт `ls -la`, що є типовим для ексфільтрації даних.

З точки зору функціональності, Snort може працювати в трьох основних режимах. У режимі "sniffer" він просто виводить мережеві пакети в консоль у текстовому вигляді. У режимі "packet logger" він зберігає весь мережевий трафік у файли журналів для подальшого аналізу. Найпотужніший — це режим IDS або IPS, де Snort застосовує набір правил до кожного пакету і може або просто логувати підозрілу активність, або активно блокувати її (у режимі IPS) [11].

Snort чудово підходить для виявлення ексфільтрації даних через ICMP. Наприклад, якщо зловмисник використовує ping або ICMP-пакети для передачі команд, Snort може знайти ці команди у вмісті пакетів. Для цього створюються сигнатури, які шукають конкретні шаблони у payload, наприклад, текст команди `"ls -la"`. Крім того, Snort здатен виявити великі за розміром ICMP-пакети, що теж може свідчити про тунелювання або витік даних. Він не здатен прямо виявляти таймінгові канали, але може бути використаний для логування трафіку з підозрілою частотою чи затримками, які потім аналізуються окремо.

Інтеграція Snort у мережу відносно проста: він підтримує більшість поширених платформ, а його правила легко модифікувати. Для зручності адміністрування можна підключити інтерфейси візуалізації типу Snorby чи Sguil або ж інтегрувати його з системами SIEM для побудови комплексної картини подій у мережі. Крім того, існує утиліта PulledPork, яка дозволяє автоматично оновлювати базу правил з інтернету.

Серед обмежень Snort варто зазначити, що він не бачить вмісту зашифрованих протоколів, таких як HTTPS, тому не здатен виявити загрози всередині них без додаткових рішень типу SSL-розшифрування. Також Snort, як сигнатурне IDS, не завжди здатен виявляти нові або нестандартні техніки, зокрема ті, що використовують

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		23

приховані канали, такі як таймінгові атаки. Втім, при правильному налаштуванні та регулярному оновленні правил Snort залишається одним з найефективніших інструментів контролю мережевої безпеки.

Suricata – це сучасна система виявлення та запобігання вторгненням (IDS/IPS), а також повноцінний мережевий моніторинговий рушій з відкритим кодом, розроблений Open Information Security Foundation (OISF) [12]. Вона виконує аналіз мережевого трафіку у режимі реального часу та вирізняється підтримкою багатопотокової обробки, що дозволяє ефективно працювати в мережах з великим навантаженням.

На відміну від Snort, Suricata має вбудований механізм розбору протоколів прикладного рівня і вміє працювати з HTTP, TLS, DNS, FTP, SMB, SSH, ICMP та іншими протоколами, тобто не просто перевіряє байти у потоці, а інтерпретує значення полів і дійсно бачить, що відбувається у трафіку [13]. Це особливо важливо для виявлення складних форм ексфільтрації через ICMP, де Suricata здатна не лише розпізнавати підозрілий payload, а й ідентифікувати незвичні шаблони, аномалії в структурі пакетів або навіть поведінкові особливості.

Suricata використовує формат журналювання EVE JSON, який дозволяє легко інтегрувати її з такими аналітичними платформами, як ELK stack (Elasticsearch, Logstash, Kibana), Splunk або Graylog [14]. Завдяки цьому можлива побудова складних систем поведінкового аналізу, виявлення аномалій, візуалізація подій і гнучка кореляція з іншими джерелами подій. Наприклад, якщо злоумисник використовує ICMP-трафік з однаковими інтервалами між пакетами або передає певні бінарні структури у payload – ці закономірності можна виявити навіть без сигнатур.

Suricata дозволяє налаштовувати детальні правила, які включають не лише вміст пакетів, але й такі параметри, як розмір payload, шаблони поведінки, частота повторення, час життя пакетів (TTL), час надходження тощо [15]. Для ICMP це означає можливість виявляти підозріло велику кількість echo-запитів, передачу даних у незвично великих ICMP-пакетах, або ж обхід фаєрволів за допомогою спеціально модифікованих ICMP-запитів – типова ознака тунелів на кшталт icmpsh, ptunnel чи Icmpth.

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

Ще однією сильною стороною Suricata є її підтримка багатоядерності. Завдяки цьому вона здатна ефективно масштабуватися й використовувати ресурси системи по максимуму, що особливо важливо для аналізу трафіку в реальному часі на гігабітних або десятигігабітних інтерфейсах. Це дозволяє розгортати Suricata не лише на окремих вузлах, а й інтегрувати її у маршрутизатори або апаратні мережеві сенсори.

З погляду користувача, конфігурація Suricata може здатися дещо складнішою, ніж у Snort, однак її потужність і гнучкість компенсують цей поріг входу. Офіційна документація, а також активна спільнота користувачів на GitHub і форумах OISF, надають широкий спектр прикладів, шаблонів конфігурацій і налаштувань для найрізноманітніших сценаріїв.

Zeek (раніше відомий як Bro) – це потужна платформа для мережевого моніторингу, орієнтована не стільки на сигнатурне виявлення атак, як у Snort чи Suricata, а на аналіз поведінки, логування та інспекцію протоколів у гнучкий і розширюваний спосіб [16]. Розроблений в UC Berkeley, Zeek активно використовується в науковій спільноті, CERT-структурах, на підприємствах та в критичній інфраструктурі.

Zeek відрізняється тим, що не просто шукає відомі шаблони атак, а створює структуровані журнали (логи) для кожного задіяного протоколу: HTTP, DNS, SMTP, SSL/TLS, FTP, SSH, ICMP та багатьох інших. Це дає змогу фахівцям з безпеки або автоматизованим аналітичним системам помічати незвичну або аномальну активність у мережі наприклад, надмірно великі ICMP-пакети, повторювані echo-запити, затримки між ними, або наявність нетипових бінарних шаблонів у тілі ICMP.

Оскільки Zeek веде розширене логування ICMP-запитів і відповідей, його часто використовують для виявлення ексфільтрації даних через протоколи, які зазвичай вважаються "безпечними" або "шумовими", на зразок ICMP. На практиці це дозволяє виявити:

- спроби передачі файлів у кількох ICMP echo-запитах;

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		25

– таймінгові канали (наприклад, коли байти передаються через часові інтервали між echo-запитами);

– аномально велику частоту ICMP-повідомлень між двома вузлами.

Zeek використовує власну мову сценаріїв (Zeek scripting language), яка дає змогу створювати адаптивні логічні правила. Наприклад, можна створити скрипт, що повідомляє, якщо певна IP-адреса надсилає понад N ICMP-повідомлень за короткий час або якщо поле payload містить підозрілу бінарну послідовність. Усе це – без модифікації ядра системи чи перезавантаження [17].

Важливо, що Zeek не працює як IPS – він не блокує трафік, а лише спостерігає та логгує. Тому його часто комбінують з іншими системами, як-от Snort, Suricata або фаєрволами. Але саме завдяки своєму потужному логуванню Zeek є одним з найефективніших інструментів для виявлення прихованих каналів зв'язку, які можуть не містити явного шкідливого коду, але виявляються через поведінку.

Zeek також інтегрується з такими інструментами, як ELK Stack, Splunk, Malcolm, Grafana, забезпечуючи візуалізацію логів і поведінковий аналіз. Існують також готові Zeek-скрипти, спрямовані на виявлення ICMP тунелів, наприклад, у випадках використання icmpsh, IcmpDoor, ptunnel, Icmptx, pingchat тощо.

На додаток, Zeek має активну спільноту, постійно оновлюється і підтримується як Zeek Project, у партнерстві з NSF, NCSA, Corelight та іншими.

IcmpTunnel Detector – це спеціалізований інструмент з відкритим кодом, розроблений для виявлення прихованих каналів ексфільтрації даних, що використовують ICMP-протокол як транспорт. На відміну від універсальних IDS/IPS-систем, як Snort чи Suricata, цей інструмент створено вузько спрямовано, з фокусом саме на виявлення тунелювання даних через ICMP echo-запити (тип 8) і відповіді (тип 0).

Основна ідея icmpTunnel Detector полягає в аналізі аномалій у структурі ICMP-трафіку та статистичних відхилень у нормальній поведінці хостів. Програма не покладається на сигнатури, а натомість аналізує такі характеристики, як:

- довжина payload;
- частота echo-запитів;
- наявність повторюваних шаблонів у вмісті;

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		26

- ентропія (міра випадковості даних);
- сталість інтервалів між пакетами (ознака таймінгового каналу) [18].

Цей інструмент був створений як дослідницький проєкт і публічно доступний на GitHub. Він дозволяє аналізувати PCAP-файли або працювати в реальному часі, підключаючись до мережевого інтерфейсу. У режимі реального часу він зчитує ICMP-пакети, обчислює метрики (ентропію, розмір, інтервали) та формує список підозрілих з'єднань із ймовірністю тунелювання.

Один з важливих аспектів icmpTunnel Detector – використання алгоритмічного підходу до класифікації, зокрема оцінки відхилень від нормальної поведінки мережі (наприклад, хост, який ніколи не використовував ICMP, раптом надсилає сотні echo-пакетів із змінним вмістом). Такі ознаки часто є характерними для тунелів на кшталт ptunnel, icmpsh, Icmpth, pingtunnel тощо.

Інструмент має простий CLI-інтерфейс і не вимагає глибокої конфігурації. Його зручно використовувати як додатковий аналізатор у зв'язці з Wireshark, Zeek або Suricata. Він не намагається замінити IDS, але добре доповнює їх, фокусуючись на вузькому класі атак, які ті можуть не виявити без спеціалізованих правил.

За результатами аналізу, icmpTunnel Detector може формувати звіт із позначенням IP-адрес, які:

- відправили підозрілу кількість ICMP echo-запитів;
- мали аномально високу ентропію у вмісті;
- передавали пакети з підозрілими інтервалами;
- не отримували echo-відповідей (ознака односторонньої передачі, характерна для бекдорів).

Проєкт підтримується ентузіастами безпеки, є повністю відкритим (GNU GPL), і його можна інтегрувати у сценарії SIEM-аналізу або як частину системи інструментів "blue team".

Netdata та Prometheus – це високопродуктивні платформи для збору, моніторингу та візуалізації метрик з систем, мереж і додатків. Хоча обидві не є системами

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

виявлення вторгнень (IDS) у класичному розумінні, вони відіграють критично важливу роль у виявленні непрямих ознак аномальної активності, включно з ексфільтрацією даних через ICMP.

Netdata – це рішення "з коробки" для реального часу, яке надає інтерактивні графіки про стан системи: мережевий трафік, кількість ICMP-запитів, інтерфейсну активність, середній розмір пакетів, RTT (затримки) та інші метрики, які можуть опосередковано свідчити про аномальне тунелювання [18]. Якщо зломисник використовує ICMP для передавання даних (наприклад, ptunnel або icmpsh), це може проявитися у вигляді постійного трафіку ICMP echo request/response, який буде легко візуалізувати в Netdata через окремі графіки ICMP-пакетів і трафіку за протоколами.

Netdata має мінімальне навантаження на систему й може бути розгорнута без складної конфігурації. Вона підтримує автоматичне виявлення мережевих інтерфейсів і генерує історію змін в реальному часі. Дані можна експортувати до сторонніх систем, зокрема, в Prometheus, де аналітика досягає ще вищого рівня.

Prometheus – це система збору метрик із потужною мовою запитів PromQL, яка дозволяє формувати складні аналітичні правила для виявлення аномалій. Наприклад, можна створити правило, яке сповіщає, якщо за останні 5 хвилин ICMP echo-запитів було в 5 разів більше, ніж середнє значення за попередню годину – що вже є потенційною ознакою тунелю[19].

Prometheus збирає дані за допомогою експортерів, зокрема:

- node_exporter – для системних метрик (мережевий трафік, load, файлові дескриптори);
- blackbox_exporter – для моніторингу доступності через ICMP-ping, з можливістю логування затримок, втрат пакетів, нестандартних відповідей.

Ще одна перевага – можливість інтеграції з Grafana, що дозволяє створювати гнучкі дашборди для моніторингу мережевої поведінки хостів, зокрема:

- графіки ICMP echo request/response за часом;
- виявлення хостів, які раптово генерують ICMP-потік;

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		28

- порівняння часу між echo-запитами (для виявлення таймінгових каналів);
- створення тригерів для оповіщення через Alertmanager [20].

Таким чином, хоча Netdata і Prometheus не призначені безпосередньо для виявлення ексфільтрації через ICMP, вони забезпечують контекст і спостережуваність, необхідні для оперативного реагування. Вони особливо ефективні у складі Defense-in-Depth архітектури, де доповнюють IDS-системи, такі як Zeek або Suricata, на рівні поведінкового контролю та довготривалого тренду спостережень.

Порівняння функцій усіх згаданих інструментів можна переглянути в Табл. 1.1. Які із параметрів, важливих при виявленні ексфільтрації ICMP-протоколу перевіряються цими системами, наявність логування та роботи у реальному часі та легкість інтеграції можна переглянути в Табл. 1.2.

Таблиця 1.1

Порівняння основну функціоналу систем для виявлення та протидії ексфільтрації ICMP-протоколу

Назва	Тип	Основні функції	Особливості
Snort	IDS/IPS	Сигнатурне виявлення, контроль ICMP-пакетів, кастомні правила	Велика база правил
Suricata	IDS/IPS/NSM	Аналіз потоків, JSON-логування, EVE формат, розширені сигнатури	Підтримка NetFlow
Zeek (Bro)	NSM/IDS	Поведінковий аналіз, логування ICMP, кастомні скрипти	Гнучкий scripting
Icmp-tunnel-detect	Спеціалізований	Аналіз ICMP-трафіку на тунелях	Заточений під тунелі
Netdata / Prometheus	Моніторинг	Логування метрик ICMP/мережі, виявлення аномалій по частоті	Візуалізація графіків

Таблиця 1.2

Додаткова інформація про системи для виявлення та протидії ексфільтрації ICMP-протоколу

Назва	Виявлення таймінгових каналів	Аналіз Payload	Логування	Реальний час	Складність інтеграції
Snort	Ні	Так	Так	Так	Низька
Suricata	Часткове	Так	Так	Так	Низька
Zeek (Bro)	Так	Так	Детальне	Ні	Середня
icmp-tunnel-detect	Так	Частково	Часткове	Так	Низька
Netdata / Prometheus	Ні	Ні	Так	Так	Низька

Висновок до розділу

У межах цього розділу було досліджено особливості ICMP-протоколу, його трафіку та видів повідомлень, що ним використовуються.

Також було досліджено методи прихованої передачі даних через цей протокол, зокрема через типи повідомлень Echo-Request та Echo-Reply та проаналізовано методи аналізу та фільтрації трафіку з прихованою передачею даних.

Окрім того, було розглянуто деякі системи виявлення та протидії ексфільтрації ICMP-протоколу з відкритим кодом. Кожен інструмент має свої переваги та недоліки, проте комбінація Suricata з Netdata та Prometheus є найбільш надійним рішенням, оскільки покриває як глибокий мережевий аналіз (через Suricata), так і безперервний моніторинг системних і мережевих метрик (через Netdata/Prometheus). Таке поєднання дозволяє виявити як явні, так і приховані ICMP-канали завдяки точному логуванню, поведінковому аналізу, часовим графікам та аномаліям у метриках. Suricata фіксує, що саме відбулося, а Netdata з Prometheus — коли, як часто і наскільки це відрізняється від звичної поведінки, що критично для виявлення тунелювання та таймінгових каналів.

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЄКТУВАННЯ СИСТЕМИ ВИЯВЛЕННЯ ТА НАПИСАННЯ ПРАВИЛ СИСТЕМИ ВИЯВЛЕННЯ ТА ПРОТИДІЇ ЕКСФІЛЬТРАЦІЇ ІСМР-ПРОТОКОЛУ

2.1 Розробка топології мережі

Одним із найважливіших етапів є розробка та топології мережі. Для проєкту підсистеми виявлення та протидії ексфільтрації даних через ІСМР-протокол в інформаційно-комунікаційній мережі головним рішенням буде сегментація, централізація процесів моніторингу та можливість впровадження правил обмеження трафіку на кінцевих пристроях. Для моніторингу та керування мережею можна створити окрему підмережу, а сегментація гілок із кінцевими пристроями дозволить зменшити зону атак, покращити моніторинг, а в майбутньому – локалізувати ІСМР-трафік чи керувати ним через Firewall.

Основні принципи побудови мережі включають кілька ключових аспектів, які забезпечують її ефективність, надійність та масштабованість. Ось основні принципи:

- 1) Мережа повинна мати ієрархічну архітектуру:
 - Core Layer: Найвищий рівень мережі, який забезпечує високу швидкість та надійність передачі даних між різними сегментами мережі.
 - Distribution Layer: Проміжний рівень, який виконує функції агрегації, контролю доступу та маршрутизації між кінцевими вузлами та ядром.
 - Access Layer: Нижчий рівень, який забезпечує доступ користувачів до мережі через комутатори та точки доступу.

Наша мережа складатиметься із центрального маршрутизатора, що забезпечить високошвидкісний зв'язок та надійність передачі, багаторівневого комутатора, що буде виконувати роль маршрутизації та сегментувати мережу за допомогою технології VLAN, та декількома комутаторами в різних сегментах, що забезпечать доступ кінцевих пристроїв до мережі.

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

2) Мережа повинна бути побудована з окремих модулів, які можна легко замінювати або оновлювати без впливу на всю систему. Це підвищує гнучкість та спрощує управління.

Наша мережа скрадатиметься із сегментів, які легко замінити чи додати, вони зможуть комунікувати між собою за допомогою InterVLAN Routing.

3) Мережа повинна мати можливість розширюватися за рахунок додавання нових пристроїв і сегментів без значної перебудови існуючої інфраструктури.

Використовуючи інструменти проєктування мережі в GNS3, поруч з двома неадміністративними сегментами нашої мережі ми зможемо додати велику кількість нових. Це означає, що в будь-який час до мережі можна буде під'єднати новий мережевий пристрій. Також варто зазначити, що лінійна топологія цих частин мережі дозволяє швидко наростити масштаби.

4) Керованість. Забезпечення можливостей для моніторингу, управління і діагностики мережі для своєчасного виявлення і усунення проблем.

Позаяк наша система буде побудова з використанням серверу моніторингу для виявлення ексфільтрації ICMP-протоколу, вона буде швидко виявляти та усувати проблеми.

5) Безпека. Впровадження заходів безпеки на всіх рівнях мережі, таких як брандмауери, системи виявлення вторгнень, автентифікація користувачів і шифрування даних.

Наявність списку доступу на багаторівневому комутаторі та користувачів на усіх мережевих пристроях адміністративної мережі дозволить керувати підключеннями.

Ці принципи є фундаментом для побудови сучасних мереж, які можуть ефективно підтримувати різні додатки і сервіси, забезпечуючи при цьому надійність і безпеку.

Як зазначалося вище, для розробки мережі буде використана сегментна трирівнева структура. Майбутня мережа буде мати такий вигляд:

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				32
Змн.	Арк.	№ докум.	Підпис	Дата		

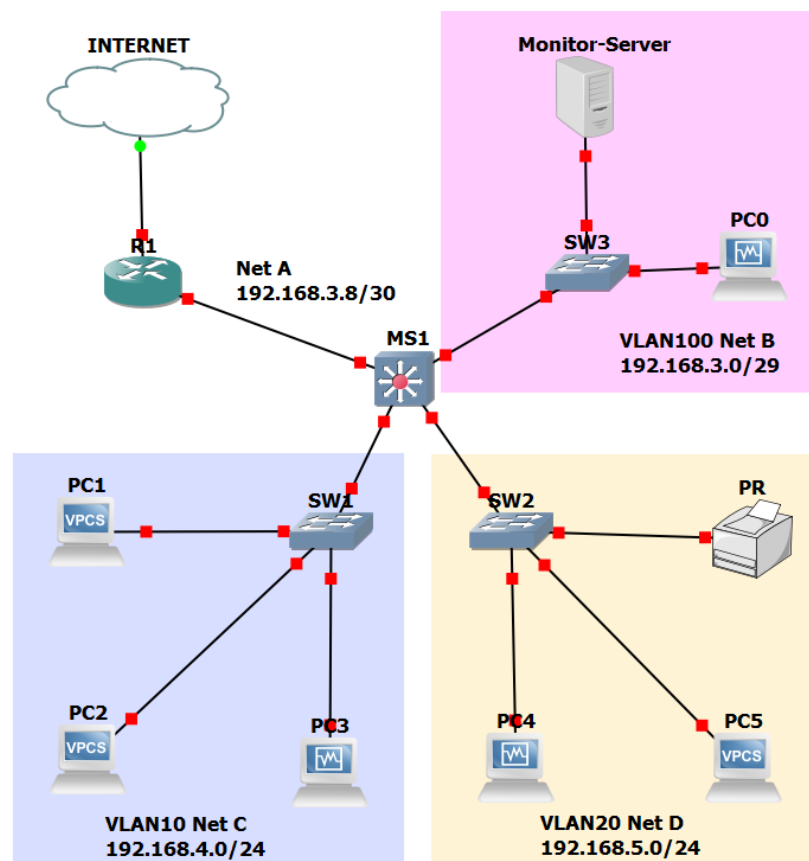


Рис. 2.1 - Загальна топологія мережі

Для налагодження підсистеми автентифікації з використанням протоколу TACACS+ нам потрібні такі ключові елементи:

- багаторівневий комутатор для впровадження сегментації;
- сервер моніторингу з можливістю розгортання Suricata та Prometheus з NetData;
- Клієнти з можливістю впровадження правил обмеження ICMP-трафіку;

Рівень міститимеме у собі центральний маршрутизатор.

Рівень розподілу буде містити у собі багаторівневий комутатор.

Рівень доступу скрадатиметься з трьох однорівневих комутаторів на трьох різних сегментах мережі – двох підмереж користувачів, та однієї підмережі моніторингу та адміністрування.

Загалом мережа скрадатиметься з одного маршрутизатора, одного багаторівневого комутатора, трьох однорівневих комутаторів, сервера моніторингу та 7 кінцевих пристроїв.

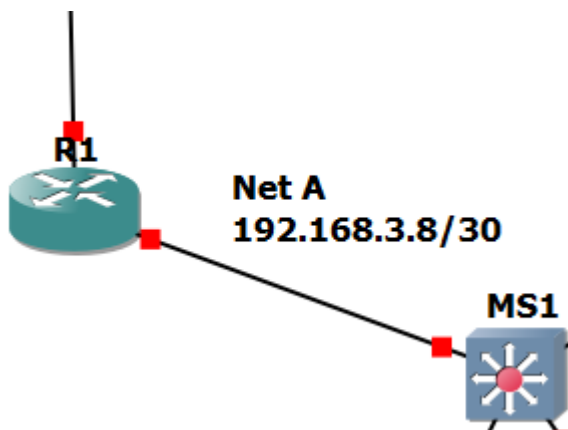


Рис. 2.2 - Підмережа А

Підмережа А складається з маршрутизатора R1, підключеного до багаторівневого комутатора MS1. Для цієї мережі потрібно забезпечити адресацію для 2 пристроїв, тобто підмережа є каналом зв'язку.

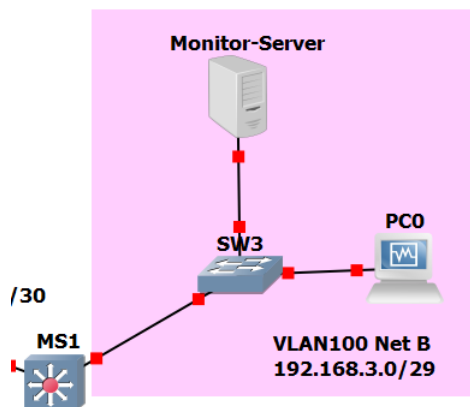


Рис. 2.3 - Підмережа В

Підмережа В є підмережою, яка забезпечує зв'язок для чотирьох пристроїв: MS1, SW3, Monitor-Server та PC0.

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

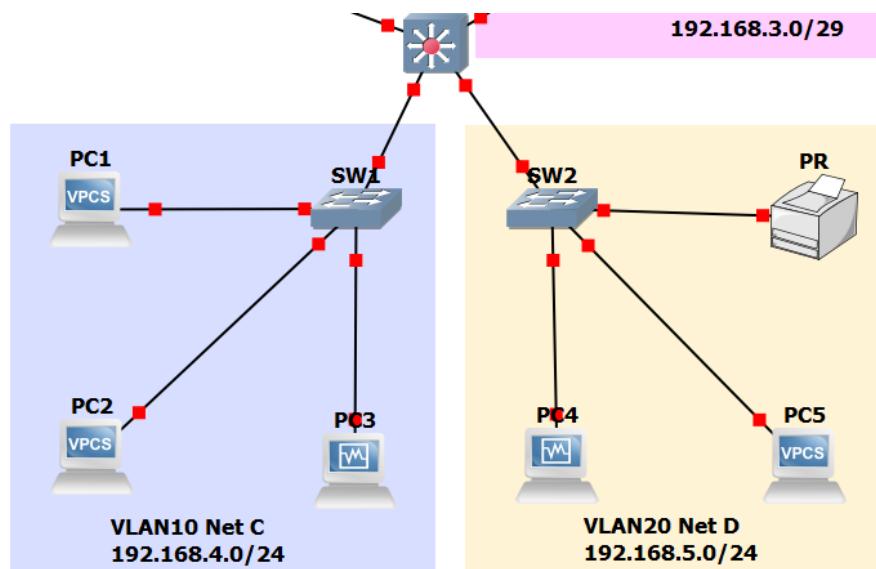


Рис. 2.4 - Підмережі C та D

Підмережа C містить багаторівневий комутатор MS, SW1 та кінцевих користувачів. Варто примітити, що саме SW1 має забезпечувати зв'язок великій кількості користувачів. Підмережа D побудована аналогічно.

Як можна помітити з попередніх розрахунків, багаторівневий комутатор MS1 знаходиться в декількох підмережах, тому для забезпечення зв'язку між різними сегментами мережі на ньому буде налаштовано маршрутизацію.

Усі підключення та параметри інтерфейсів пристроїв можна переглянути в описовій таблиці 2.1:

Таблиця 2.1

Підключення та параметри інтерфейсів пристроїв

Пристрій	Інтерфейс	Підключення до пристрою	Підключення до
R1	Fa0/0	MS1	Eth0/0
MS1	Eth0/0	R1	Fa0/0
	Eth0/1	SW1	
	Eth0/2	SW2	
	Eth0/3	SW3	
SW1	Eth0/0	MS1	Eth0/1
	Eth1/1	PC1	Eth0
	Eth1/2	PC2	
	Eth1/3	PC3	

SW2	Eth0/0	MS1	Eth0/2
	Eth1/1	PC4	Eth0
	Eth1/2	PC5	
	Eth1/3	PR	
SW3	Eth0/0	MS1	Eth0/3
	Eth0/1	Monitor-Server	Eth0
	Eth0/2	PC0	
Monitor-Server	Eth0	SW3	Eth0/1
PC0		SW1	Eth0/2
PC1			Eth1/1
PC2			Eth1/2
PC3			Eth1/3
PC4		SW2	Eth1/1
PC5			Eth1/2
PR			Eth1/3

2.2 Розробка системи IP-адресації

Наступним не менш важливим кроком в розробці мережі є розробка схеми адресації для правильного та доцільного використання IP-адрес.

IP-адреса (Internet Protocol address) – унікальний числовий ідентифікатор мережевого рівня, що використовується для адресації комп'ютерів чи пристроїв у мережах, які побудовані з використанням стека протоколів TCP/IP (наприклад, Інтернет). У мережі Інтернет потрібна глобальна унікальність адрес, а в локальній мережі – у межах мережі. У версії протоколу IPv4 IP-адреса має довжину 4 байти, а у версії IPv6 – 16 байтів.

Маска підмережі – число, яке містить одиниці в тих розрядах, які відносяться до розширеного мережевого префікса. Маска підмережі дозволяє поділити IP-адресу на дві частини: номер підмережі та номер пристрою у цій підмережі.

Статична (постійна, незмінна) IP-адреса – це адреса, яка вручну налаштовується користувачем у налаштуваннях пристрою, або як адресу, що надається автоматично при підключенні пристрою до мережі та не може бути одночасно присвоєна двом або більше пристроям. Саме таку адресацію ми будемо використовувати у нашому проєкті.

Основним об'єктом нашої мережі є Підмережа В, яка виконує роль моніторингу та адміністрування, відповідно інших кінцевих пристроїв для цієї мережі, окрім перелічених раніше, не має бути передбачено розрахунками.

Для Підмережі А було вибрано адресу 192.168.3.0. Для цієї підмережі нам потрібно забезпечити адресацію для 4 пристроїв мережі: MS, SW3, PC0 та Monitor-Server.

Як розрахувати яку маску потрібно використати? Для розв'язання такої задачі слід скористатися такими залежностями, що описують довжини IP-адреси та префіксу в загальному вигляді:

$$N + H = 32 \text{ біти}, P = N$$

Де N – кількість бітів, які виділені для адресації мережі (номер мережі), H – кількість бітів, які виділені для адресації вузлів мережі, P – кількість бітів, які виділені для формування префіксу мережі. Кожному вузлу мережі відповідає одна IP-адреса.

Слід пам'ятати, що, окрім IP-адрес вузлів, у мережі наявні і розраховуються IP-адреса мережі та широкомовна IP-адреса. Тому до заданої кількості IP-адрес вузлів необхідно додати ще дві адреси. Оскільки адресація починається з нуля, то необхідно одну IP-адресу відняти.

Для визначення значення H формується число X вигляду:

$$X = K_{\text{вузлів}} + 2 - 1, X = 4 + 2 - 1 = 5$$

Отримане число X переводиться із десяткової у двійкову систему числення:

$$X_{10} \rightarrow X_2, 5_{10} = 101_2$$

Кількість бітів у даному числі $H = 3$, і саме вони використовуються для нумерації вузлів.

Префікс мережі визначається як:

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		37

$P = 32 - H, P = 32 - 3 = 29$ бітів.

Префікс відповідно має вигляд – /29, а у десятковому вигляді – 255.255.255.248

Кількість вузлів (IP-адрес вузлів), які можуть входити в мережу, становить:

$$K_{\text{вузлів}} = 2^{(32-29)} - 2 = 2^3 - 2 = 6 \text{ вузлів.}$$

З них 4 (66,66%) IP-адреси будуть використовуватися, а 2 (33,33%) – не використовуватимуться. Отже,

Мінімальна адреса мережі: 190.168.3.1/29

Максимальна адреса мережі 190.168.3.6/29

Широкомовною адреса мережі: 190.168.3.7/29

Усі детальні параметри Підмережі В наведені в таблиці 2.2:

Таблиця 2.2

Адресація Підмережі А

Призначення	Значення
Адреса підмережі	190.168.3.0
Маска підмережі	255.255.255.248/29
Інверсна маска	0.0.0.7
Мінімальна адреса	190.168.3.1
Максимальна адреса	190.168.3.6
Широкомовна адреса	190.168.3.7
Шлюз за замовчуванням	190.168.3.1
Адреса маршрутизатора	190.168.3.1
Адреса комутатора	190.168.3.2
Діапазон вільних адрес	190.168.3.3 – 190.168.3.6

Іншим елементом побудованої мережі є канальне з'єднання між маршрутизатором R1 та багаторівневим комутатором MS1 (див. Рис. 2.2.) Позаяк необхідність є тільки у забезпеченні зв'язку між двома пристроями, за методом, описаним вище, знаходимо оптимальну маску для Підмережі А – 192.168.3.8

$$X = K_{\text{вузлів}} + 2 - 1$$

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

$$X = 2 + 2 - 1 = 3$$

Отримане число X переводиться із десятичної у двійкову систему числення:

$$X_{10} \rightarrow X_2$$

$$1410 = 11_2$$

Кількість бітів у даному числі $N = 2$, і саме вони використовуються для нумерації вузлів.

Префікс мережі визначається як:

$$P = 32 - N$$

$$P = 32 - 2 = 30 \text{ бітів.}$$

Префікс відповідно має вигляд $-/30$, а у десятичному вигляді – 255.255.255.252

Кількість вузлів (IP-адрес вузлів), які можуть входити в мережу, становить:

$$K_{\text{вузлів}} = 2^{(32-30)} - 2 = 2^2 - 2 = 2 \text{ вузла.}$$

З них 2 (100%) IP-адреси будуть використовуватися.

Усі детальні параметри Підмережі А наведені в таблиці 2.3:

Таблиця 2.3

Адресація Підмережі В

Призначення	Значення
Адреса підмережі	192.168.3.8
Маска підмережі	255.255.255.252/30
Інверсна маска	0.0.0.3
Мінімальна адреса	192.168.3.9
Максимальна адреса	192.168.3.10
Широкомовна адреса	192.168.3.11
Шлюз за замовчуванням	192.168.3.9
Адреса маршрутизатора	192.168.3.9
Адреса комутатора	192.168.3.10

Наступним кроком буде розрахування мереж, де розташовані кінцеві пристрої – С та D.

Для цих підмереж було обрано наступні доступні адреси – 192.168.4.0 та 192.168.5.0. Для забезпечення великої кількості підключень скористаймося маскою 255.255.255.0/24, що дозволить надати адреси 254 пристроям. Детальна інформація про зазначені підмережі наведена в Табл. 2.4:

Таблиця 2.4

Адресація підмереж С та D

Назва	Підмережа	Шлюз	Комутатор	Вільні адреси
Підмережа С	192.168.4.0/24	192.168.4.1	192.168.4.2	192.168.4.3 – 192.168.4.254
Підмережа D	192.168.5.0/24	192.168.5.1	192.168.5.2	192.168.5.3 – 192.168.5.254

Адресацію усіх пристроїв мережі наведено у таблиці 2.5:

Таблиця 2.5

Параметри адресації

Пристрій/Підмережа	Інтерфейс	ІР-адреса	Маска	Префікс
Підмережа А	-	192.168.3.8	255.255.255.252	/30
Підмережа В	-		255.255.255.248	/29
Підмережа С	-		255.255.255.0	/24
Підмережа D	-		255.255.255.0	/24
R1	Fa0/0	192.168.3.9	255.255.255.252	/30
MS1	Eth0/0	192.168.3.10	255.255.255.252	/30
	VLAN 10	192.168.4.1	255.255.255.0	/24
	VLAN 20	192.168.5.1	255.255.255.0	/24
	VLAN 100	192.168.3.1	255.255.255.248	/29
SW1	VLAN 10	192.168.4.2	255.255.255.0	/24
	Шлюз	192.168.4.1	-	-
SW2	VLAN 20	192.168.5.2	255.255.255.0	/24
	Шлюз	192.168.5.1	-	-
SW3	VLAN 100	192.168.3.2	255.255.255.248	/29
	Шлюз	192.168.3.1	-	-

Monitor-Server	Eth0	192.168.3.3	255.255.255.248	/29
	Шлюз	192.168.3.1	-	-
PC0	Eth0	192.168.3.4	255.255.255.248	/29
	Шлюз	192.168.3.1	-	-
PC1	Eth0	192.168.4.3	255.255.255.0	/24
	Шлюз	192.168.4.1	-	-
PC2	Eth0	192.168.4.4	255.255.255.0	/24
	Шлюз	192.168.4.1	-	-
PC3	Eth0	192.168.4.5	255.255.255.0	/24
	Шлюз	192.168.4.1	-	-
PC4	Eth0	192.168.5.3	255.255.255.0	/24
	Шлюз	192.168.5.1	-	-
PC5	Eth0	192.168.5.4	255.255.255.0	/24
	Шлюз	192.168.5.1	-	-
PR	Eth0	192.168.5.5	255.255.255.0	/24
	Шлюз	192.168.5.1	-	-

2.3 Проектування роботи Suricata

Правила Suricata — це текстові інструкції, які описують, які мережеві події слід виявляти, логувати або блокувати. Вони нагадують правила IDS/IPS системи Snort і зазвичай зберігаються у файлах з розширенням .rules.

За замовчуванням, правила зберігаються в каталозі /etc/suricata/rules/. Основний файл для власних правил — це local.rules. У конфігурації suricata.yaml вказується список файлів правил, які завантажуються при старті. Кожне правило має дві частини — заголовок та опції.

Приклад заголовку правила для перехоплення ICMP-пакетів:

```
alert icmp any any -> any any
```

Пояснення: alert — дія (може бути alert, drop, reject, pass тощо), icmp — протокол, any any -> any any — напрямок трафіку: джерело IP/порт до призначення IP/порт

Приклад опцій правил для перехоплення ICMP-пакетів(в дужках):

```
(msg:"ICMP detected"; sid:1000001; rev:1;)
```

Пояснення: msg — текст повідомлення, що з'явиться в логах, sid — унікальний ідентифікатор правила, rev — ревізія (номер версії правила)

Також можуть використовуватись інші поля: content, http_uri, flow, threshold, classtype тощо — залежно від складності перевірки.

Для застосування правил потрібно їх записати (або відредагувати існуючі) Ви у відповідному .rules файлі. У suricata.yaml перед початком роботи треба переконатися, що файл підключено, та перезапустити Suricata. Трафік, що відповідає умовам правила, буде зафіксований у логах (fast.log, eve.json тощо), або буде заблокований — залежно від дії (alert / drop).

Якщо з'єднати дві частини, описані вище, то ми отримаємо правило, яке буде сповіщати про те, що пристрій отримав ICMP-пакет. Проте, нам не потрібно отримувати усі пакети, що буде отримувати та відправляти Monitor-Server, а лише ті, які можуть потенційно нести шкоду, тож створимо декілька власних правил.

```
drop icmp any any -> any any (msg: "LARGE ICMP PACKET"; dsize:>50; sid:1000061; rev:1;)
```

Функція drop відкидає пакет та створює сповіщення. dsize перевіряє розмір пакету, в даному випадку правило буде спрацьовувати, якщо пакет буде розміром більше 50 мегабайтів.

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		42

```
drop icmp any any any any (msg: "ICMP FLOOD"; threshold:type both, track by  
src, count 3, seconds 1, sid:1000002; rev:1;
```

Однією з багатьох ознак простих атак із використанням ICMP-ексфільтрації є велика кількість ICMP-повідомлень у короткий час. Це правило буде перевіряти, щоб кількість (count) повідомлень за секунду (seconds) не перевищувала 3.

```
drop icmp any any any any (Rsg: "POSSIBLE PASSWORD EXFIL: content:  
"password"; sid: 1000003; rev:1;)
```

Content буде перевіряти поле DATA у ICMP-пакеті на вміст ключових слів. У даному випадку це password. Це слово було вибрано, адже злочинці часто можуть за допомогою ексфільтрації викрадати дані для автентифікації.

2.4 Проектування сервісів на кінцевих користувачах.

Windows Firewall може допомогти контролювати ICMP-трафік шляхом створення гнучких правил фільтрації, які визначають, які ICMP-пакети дозволені чи заборонені. Це дозволяє зменшити ризик ICMP-ексфільтрації або сканування мережі, залишаючи при цьому можливість для легітимного використання ICMP (наприклад, ping або tracert). Створимо для Windows Firewall правила, щоб обмежити ICMP трафік.

```
New-NetFirewallRule -DisplayName "Block Fragmented ICMP" -Protocol  
ICMPv4 -IcmpType 3,4 -Direction Outbound -Action Block
```

Це правило буде блокувати трафік ICMP-повідомлень 3 та 4 типу.

```
New-NetFirewallRule -DisplayName "Allow Internal ICMP Only" -Protocol  
ICMPv4 -Direction Outbound -RemoteAddress 192.168.4.0/24 -Action Allow
```

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		43

Це правило буде блокувати трафік для зовнішніх мереж.

2.5 Проєктування симуляції атак з використанням ексфільтрації ICMP-протоколу

Для виконання симуляції атаки ми використаємо віртуальну машину ОС Kali-Linux та інструмент Scapy з власноруч написаними скриптами.

Scapy — це потужна Python-бібліотека для створення, аналізу, надсилання та прийому мережевих пакетів. Вона широко використовується для мережевих досліджень, тестування безпеки, написання сканерів, виявлення вразливостей, і навіть для атаки чи захисту (у навчальних цілях) [25].

Для атаки з перевищенням розміру поля DATA створимо скрипт мовою Python.

```
from scapy.all import *
target_ip = "192.168.3.3"
file_path = "secret.txt"
with open(file_path, "rb") as f:
    data = f.read()
#Розбиваємо на фрагменти по 100 байт
chunks = [data[i:i+100] for i in range(0, len(data), 100)]
for i, chunk in enumerate(chunks):
    pkt = IP(dst=target_ip)/ICMP(type=8, id=0x1234, seq=i)/Raw(load=chunk)
    send(pkt)
    print(f"Sent chunk {i+1}/{len(chunks)}")
```

Цей код буде використовувати зміст файлу secret.txt, ділити його на шматки по 100мб, та відправляти на задану адресу.

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		44

Наступний скрипт створимо для відправлення автентифікаційних даних. Припустимо, що файл буде містити значення поля паролю, тоді скрипт буде виглядати так:

```
from scapy.all import *

# Цільова IP-адреса
target_ip = "192.168.3.3"

# Дані для ексфільтрації
payload = "password:12345"

# Формування ICMP-пакета з даними
packet = IP(dst=target_ip)/ICMP()/payload

# Відправка пакета
send(packet)
```

Завдяки виконанню таких кодів ми зможемо перевірити можливості нашої системи виявлення та протидії.

Висновки до розділу

У межах цього розділу було створено проєкт інформаційно-комунікаційної мережі для реалізації проєкту моніторингу. Було розроблену схему адресації мережі, досліджено процедури написання правил Suricata, створені правила для перехоплення і блокування ICMP пакетів як у Suricata, так і на кінцевих пристроях.

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		45

РОЗДІЛ 3 НАЛАГОДЖЕННЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Вибір обладнання для організації роботи мережі

Для забезпечення надійної роботи мережі необхідно підібрати відповідне обладнання.

Маршрутизатор Cisco 7200 є популярною серією маршрутизаторів, яка використовується в корпоративних мережах і провайдерами послуг завдяки своїй надійності та високій продуктивності.



Рис. 3.1 - Маршрутизатор Cisco 7200

Основні характеристики маршрутизатора Cisco 7200 включають:

1. Процесор та продуктивність:
 - Можливість використання процесорів різної продуктивності для задоволення потреб різних конфігурацій.
 - Підтримка високої швидкості передачі даних і здатність обробляти великий обсяг трафіку.
2. Інтерфейси та модулі:
 - Підтримка різноманітних інтерфейсів, таких як Ethernet, Fast Ethernet, Gigabit Ethernet, а також серійні інтерфейси для WAN-з'єднань.

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		46

- Можливість розширення функціональності за допомогою додаткових модулів (Network Processing Engine - NPE, Input/Output Controller - I/O Controller).

3. Пам'ять:

- Конфігурації з різними обсягами оперативної пам'яті (DRAM) і флеш-пам'яті, що дозволяє зберігати великі таблиці маршрутизації та запускати складні мережеві додатки.

4. Маршрутизаційні можливості:

- Підтримка різноманітних протоколів маршрутизації, включаючи OSPF, EIGRP, BGP, RIP, і статичну маршрутизацію.
- Підтримка багатопроTOCOLЬНОЇ маршрутизації для IPv4 і IPv6.

5. QoS (Якість обслуговування):

- Можливість налаштування QoS для пріоритезації різних типів трафіку (наприклад, для VoIP, відео-конференцій).

6. Безпека:

- Вбудовані функції безпеки, такі як ACL (Access Control Lists), підтримка VPN (Virtual Private Network), IPSec, і інших механізмів безпеки.
- Можливість інтеграції з системами IDS/IPS для виявлення та запобігання вторгнень.

7. Можливості управління:

- Підтримка різних протоколів управління, таких як SNMP, Telnet, SSH, і Cisco IOS для зручного керування та моніторингу мережі.
- Підтримка веб-інтерфейсу для зручного налаштування та управління.

8. Надійність і відмовостійкість:

- Підтримка резервування компонентів, таких як джерела живлення, для підвищення надійності роботи.
- Підтримка функцій для автоматичного відновлення після збоїв.

Ці характеристики роблять маршрутизатори Cisco 7200 універсальними пристроями для побудови масштабованих та надійних мереж, здатних забезпечити високий рівень продуктивності та безпеки.

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		47



Рис. 3.2 - Комутатор Cisco WS-C2960-24-S

Комутатори серії Cisco Catalyst 2960 є популярними мережевими пристроями, призначеними для корпоративних мереж. Вони забезпечують надійне та масштабоване комутаційне рішення з високою продуктивністю та безпекою. Основні характеристики комутатора Cisco Catalyst 2960 включають:

1. Інтерфейси:
 - Порти Fast Ethernet (10/100 Мбіт/с) або Gigabit Ethernet (10/100/1000 Мбіт/с) для підключення кінцевих пристроїв.
 - Декілька моделей також підтримують порти SFP для підключення оптичних ліній зв'язку.
2. Продуктивність:
 - Висока комутаційна швидкість та пропускна здатність, що дозволяє обробляти великий обсяг трафіку.
 - Підтримка швидкості комутації на рівні лінії (line rate) для всіх портів.
3. Power over Ethernet (PoE):
 - Підтримка PoE і PoE+ для живлення IP-телефонів, бездротових точок доступу та інших пристроїв через Ethernet-кабелі.
 - Моделі з різною кількістю PoE-портів та максимальною потужністю на порт.
4. Безпека:

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				48
Змн.	Арк.	№ докум.	Підпис	Дата		

- Функції безпеки, такі як ACL (Access Control Lists), 802.1X аутентифікація, захист від атак на мережевий рівень (DHCP Snooping, Dynamic ARP Inspection).

- Підтримка безпечного управління через SSH, SSL та SNMPv3.

5. QoS (Якість обслуговування):

- Можливість налаштування QoS для пріоритезації трафіку, що дозволяє забезпечити якість обслуговування для критичних додатків, таких як VoIP та відео.

6. Управління:

- Підтримка різних протоколів управління, таких як SNMP, Telnet, SSH, RMON, та веб-інтерфейс для зручного управління та моніторингу.

- Інтеграція з Cisco Network Assistant для централізованого управління та налаштування мережі.

7. Програмне забезпечення:

- Працює на операційній системі Cisco IOS, яка забезпечує широкий спектр функцій для управління мережею та безпеки.

- Підтримка функцій Layer 2, таких як VLAN, STP (Spanning Tree Protocol), EtherChannel.

8. Модульність та масштабованість:

- Можливість об'єднання кількох комутаторів у стек для спрощення управління та підвищення масштабованості мережі.

- Підтримка функцій управління стеками, що забезпечує єдину точку управління для кількох пристроїв.

9. Надійність:

- Висока надійність та тривалий термін експлуатації.

- Підтримка функцій резервування та швидкого відновлення після збоїв, таких як RSTP (Rapid Spanning Tree Protocol).

Ці характеристики роблять комутатори Cisco Catalyst 2960 ідеальними для побудови надійних, масштабованих та ефективних мереж у різних організаціях.

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		49



Рис. 3.3 - Багаторівневий комутатор Cisco WS-C3560-24PS

Комутатор Cisco Catalyst 3650-24PS є потужним багаторівневим мережевим пристроєм корпоративного класу, який підтримує як рівень 2, так і рівень 3 маршрутизації. Цей комутатор забезпечує високу продуктивність, розширені функції безпеки та інтеграцію з бездротовими мережами, що робить його ідеальним вибором для сучасних підприємств.

Основні характеристики Cisco Catalyst 3650-24PS включають:

1. Інтерфейси:
 - 24 порти Gigabit Ethernet (10/100/1000 Мбіт/с) для підключення кінцевих пристроїв.
 - 4 SFP-порти для підключення оптичних ліній зв'язку (uplink).
 - Консольний порт RJ-45 і USB для управління пристроєм.
2. Продуктивність:
 - Комутаційна здатність до 88 Gbps.
 - Пропускна здатність до 41,66 Mpps.
 - Підтримка комутації та маршрутизації на рівні лінії (line-rate).
3. Power over Ethernet (PoE):
 - Підтримка PoE і PoE+ для живлення IP-телефонів, точок доступу, відеокамер тощо.
 - Загальна потужність PoE: до 390 Вт (в залежності від блоку живлення).
 - Автоматичне визначення пристроїв, що підтримують PoE.

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				50
Змн.	Арк.	№ докум.	Підпис	Дата		

4. Багаторівнева маршрутизація (Layer 3):
 - Підтримка статичної маршрутизації, RIP, OSPF, EIGRP та PIM.
 - Підтримка інкапсуляції VRF-Lite, Policy-Based Routing (PBR) та інших функцій.
5. Інтеграція з бездротовими мережами:
 - Вбудований контролер бездротових точок доступу (до 25 точок доступу).
 - Централізоване управління провідною та бездротовою інфраструктурою.
6. Безпека:
 - Розширена підтримка ACL, 802.1X, DHCP Snooping, IP Source Guard.
 - Безпечне управління через SSH, SSL, SNMPv3.
 - Підтримка TrustSec та MACsec (на вибраних моделях).
7. QoS (Якість обслуговування):
 - Гнучке керування пріоритетами трафіку.
 - Підтримка політик на основі портів, VLAN або IP-протоколів.
 - Оптимізація продуктивності для VoIP, відео та критичних сервісів.
8. Управління:
 - Повноцінне управління через Cisco IOS XE, Cisco Prime, CLI, SNMP, RMON.
 - Веб-інтерфейс і підтримка Cisco DNA Center.
 - Smart Licensing.
9. Масштабованість та стекування:
 - Підтримка StackWise-160: об'єднання до 9 пристроїв в єдиний логічний блок.
 - Пропускна здатність стека – до 160 Gbps.
 - Централізоване управління стеком.
10. Надійність:
 - Гаряча заміна блоків живлення (на вибраних моделях).

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				51
Змн.	Арк.	№ докум.	Підпис	Дата		

- Підтримка резервування з використанням протоколів HSRP, VRRP, RSTP.
- Високий рівень готовності та стабільності роботи в корпоративному середовищі.

Ці моделі пристроїв дозволять нам побудувати надійну інформаційно-комунікаційну мережу та легко її використовувати.

3.2 Налаштування мережі

Розпочнемо налаштування із загальних налаштувань (налаштування портів, VLAN та адресації).

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int fa0/0
R1(config-if)#ip address 192.168.3.9 255.255.255.252
R1(config-if)#end
R1#
R1#wr
```

```
MS1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
MS1(config)#int eth0/0
MS1(config-if)#ip address 192.168.3.10 255.255.255.252
MS1(config-if)#ip default-gateway 192.168.3.9
MS1(config)#
```

Рис. 3.4 - Налаштування підмережі А на MS1 та R1

```
MS1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
MS1(config)#int vlan 10
MS1(config-if)#ip address
*Jun 27 16:16:23.637: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state to down
MS1(config-if)#ip address 192.168.4.1 255.255.255.0
MS1(config-if)#no shut
MS1(config-if)#int vlan 10
*Jun 27 16:16:42.816: %LINK-3-UPDOWN: Interface Vlan10, changed state to down
MS1(config-if)#int vlan 20
MS1(config-if)#ip address 192.168.4.1 255.255.255.0
*Jun 27 16:16:48.441: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed state to down
MS1(config-if)#ip address 192.168.5.1 255.255.255.0
MS1(config-if)#no shut
MS1(config-if)#ip address 192.168.5.1 255.255.255.0
*Jun 27 16:16:59.363: %LINK-3-UPDOWN: Interface Vlan20, changed state to down
MS1(config-if)#vlan 100
MS1(config-vlan)#no shut
%VLAN 100 is not shutdown.
```

Рис. 3.5 - Налаштування на MS1 (1)

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

```

MS1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
MS1(config)#int eth0/2

MS1(config)#int eth0/1
MS1(config-if)#switchport trunk encapsulation dot1q
MS1(config-if)#end

MS1(config-if)#switchport trunk encapsulation dot1q
MS1(config-if)#ex
MS1(config)#int eth0/3
MS1(config-if)#switchport trunk encapsulation dot1q
MS1(config-if)#end

```

Рис. 3.6 - Налаштування на MS1 (2)

```

SW1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)#int range eth0/1-3
SW1(config-if-range)#switchport
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#
*Jun 27 16:20:41.203: %LINK-3-UPDOWN: Interface Ethernet0/1, changed state to up
*Jun 27 16:20:41.203: %LINK-3-UPDOWN: Interface Ethernet0/2, changed state to up
*Jun 27 16:20:41.203: %LINK-3-UPDOWN: Interface Ethernet0/3, changed state to up
*Jun 27 16:20:42.207: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/
1, changed state to down
*Jun 27 16:20:42.207: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/
2, changed state to down
*Jun 27 16:20:42.207: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/
3, changed state to down
SW1(config-if-range)#
*Jun 27 16:20:43.215: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/
1, changed state to up
*Jun 27 16:20:43.215: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/
2, changed state to up
*Jun 27 16:20:43.215: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/
3, changed state to up
SW1(config-if-range)#switchport access vlan 10
SW1(config-if-range)#ex
SW1(config)#int vlan 10
SW1(config-if)#ip addr
*Jun 27 16:21:04.471: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state to down
SW1(config-if)#ip address 192.168.4.2 255.255.255.0
SW1(config-if)#end

SW1(config)#int vlan 10
SW1(config-if)#no shut
SW1(config-if)#
*Jun 27 16:27:27.051: %LINK-3-UPDOWN: Interface Vlan10, changed state to up
SW1(config-if)#
*Jun 27 16:27:27.055: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state to up
SW1(config-if)#

SW1(config-if)#exit
SW1(config)#int e0/0
SW1(config-if)#switchport trunk encapsulation dot1q
SW1(config-if)#

```

Рис. 3.7 - Налаштування на SW1

Налаштування на інших комутаторах є аналогічним.

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				53
Змн.	Арк.	№ докум.	Підпис	Дата		

```

PC5> ip 192.168.5.4/24 192.168.5.1
Checking for duplicate address...
PC1 : 192.168.5.4 255.255.255.0 gateway 192.168.5.1

PC5> save
Saving startup configuration to startup.vpc
. done

```

Рис. 3.8 - Налаштування адресації на PC5

Протокол Інтернету версії 4 (TCP/IPv4) – властивості

Загальні

Параметри протоколу IP можуть призначатися автоматично, якщо ваша мережа підтримує таку можливість. В іншому разі слід отримати параметри IP в адміністратора мережі.

☐ Отримати IP-адресу автоматично

☒ Використовувати таку IP-адресу:

IP-адреса: 192 . 168 . 4 . 5

Маска підмережі: 255 . 255 . 255 . 0

Основний шлюз: 192 . 168 . 4 . 1

☐ Отримати адресу DNS-сервера автоматично

☒ Використовувати такі адреси DNS-серверів:

Основний DNS-сервер: . . .

Альтернативний DNS-сервер: . . .

☐ Під час виходу підтверджувати настройки

Додатково...

OK Скасувати

Рис. 3.9 - Налаштування адресації на PC3

Скасувати **Дротовий** Застосувати

Подробиці Засвідчення **IPv4** IPv6 Безпека

Метод IPv4

☐ Автоматично (DHCP) ☐ Лише місцевий

☒ Вручну ☐ Вимкнути

☐ Спільне разом з іншими комп'ютерами

Адреси

Адреса	Маска	Шлюз	
192.168.5.3	255.255.255.0	192.168.5.1	🗑️
			🗑️

Рис. 3.10 - Налаштування адресації на PC4

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

На інших кінцевих пристроях налаштування адресації є ідентичним. Перед налаштуванням маршрутизації перевіримо наявність зв'язку в межах підмереж.

```
PC1> ping 192.168.4.2
84 bytes from 192.168.4.2 icmp_seq=1 ttl=255 time=0.461 ms
84 bytes from 192.168.4.2 icmp_seq=2 ttl=255 time=0.632 ms
84 bytes from 192.168.4.2 icmp_seq=3 ttl=255 time=0.938 ms
84 bytes from 192.168.4.2 icmp_seq=4 ttl=255 time=0.802 ms
84 bytes from 192.168.4.2 icmp_seq=5 ttl=255 time=0.550 ms

PC1> ping 192.168.4.1
host (192.168.4.1) not reachable

PC1> ping 192.168.4.1
84 bytes from 192.168.4.1 icmp_seq=1 ttl=255 time=0.592 ms
84 bytes from 192.168.4.1 icmp_seq=2 ttl=255 time=1.391 ms
84 bytes from 192.168.4.1 icmp_seq=3 ttl=255 time=1.122 ms
84 bytes from 192.168.4.1 icmp_seq=4 ttl=255 time=1.276 ms
84 bytes from 192.168.4.1 icmp_seq=5 ttl=255 time=1.362 ms
```

Рис. 3.11 - Перевірка зв'язку PC1 з MS1 та SW1

```
Командний рядок
Microsoft Windows [Version 10.0.18363.592]
(c) Корпорація Майкрософт (Microsoft Corporation), 2019. Усі права захищено.

C:\Users\Student>ping 192.168.4.1

Pinging 192.168.4.1 with 32 bytes of data:
Reply from 192.168.4.1: bytes=32 time=1ms TTL=255
Reply from 192.168.4.1: bytes=32 time<1ms TTL=255
Reply from 192.168.4.1: bytes=32 time<1ms TTL=255
Reply from 192.168.4.1: bytes=32 time=1ms TTL=255

Ping statistics for 192.168.4.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\Student>ping 192.168.4.2

Pinging 192.168.4.2 with 32 bytes of data:
Reply from 192.168.4.2: bytes=32 time=1ms TTL=255
Reply from 192.168.4.2: bytes=32 time<1ms TTL=255
Reply from 192.168.4.2: bytes=32 time<1ms TTL=255
Reply from 192.168.4.2: bytes=32 time=1ms TTL=255

Ping statistics for 192.168.4.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\Student>
```

Рис. 3.12 - Перевірка зв'язку PC3 з MS1 та SW1

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		55

```

student@student-VirtualBox:~$ ping 192.168.5.1
ЛУНА-ІМПУЛЬС 192.168.5.1 (192.168.5.1) 56(84) байтів даних.
64 байтів з 192.168.5.1: icmp_seq=1 ttl=255 час=2.00 мс
64 байтів з 192.168.5.1: icmp_seq=2 ttl=255 час=1.19 мс
64 байтів з 192.168.5.1: icmp_seq=3 ttl=255 час=1.56 мс
64 байтів з 192.168.5.1: icmp_seq=4 ttl=255 час=1.22 мс
64 байтів з 192.168.5.1: icmp_seq=5 ttl=255 час=1.26 мс
64 байтів з 192.168.5.1: icmp_seq=6 ttl=255 час=1.10 мс
^C
--- Статистика луна-імпульсів 192.168.5.1 ---
Передано 6 пакетів, отримано 6, втрачено 0% пакетів, час 5007 мс
rtt мін/сер/макс/mdev = 1.095/1.386/2.000/0.309 мс
student@student-VirtualBox:~$ ping 192.168.5.2
ЛУНА-ІМПУЛЬС 192.168.5.2 (192.168.5.2) 56(84) байтів даних.
64 байтів з 192.168.5.2: icmp_seq=1 ttl=255 час=1.67 мс
64 байтів з 192.168.5.2: icmp_seq=2 ttl=255 час=1.18 мс
64 байтів з 192.168.5.2: icmp_seq=3 ttl=255 час=1.21 мс
64 байтів з 192.168.5.2: icmp_seq=4 ttl=255 час=0.779 мс
64 байтів з 192.168.5.2: icmp_seq=5 ttl=255 час=0.906 мс
64 байтів з 192.168.5.2: icmp_seq=6 ttl=255 час=0.798 мс
^C
--- Статистика луна-імпульсів 192.168.5.2 ---
Передано 6 пакетів, отримано 6, втрачено 0% пакетів, час 5020 мс
rtt мін/сер/макс/mdev = 0.779/1.091/1.674/0.310 мс
student@student-VirtualBox:~$

```

Рис. 3.13 - Перевірка зв'язку PC4 з SW2 та MS1

Для налаштування маршрутизації на MS1 виконуємо команду ip routing:

```

MS1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
MS1(config)#ip routing
MS1(config)#exit
MS1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

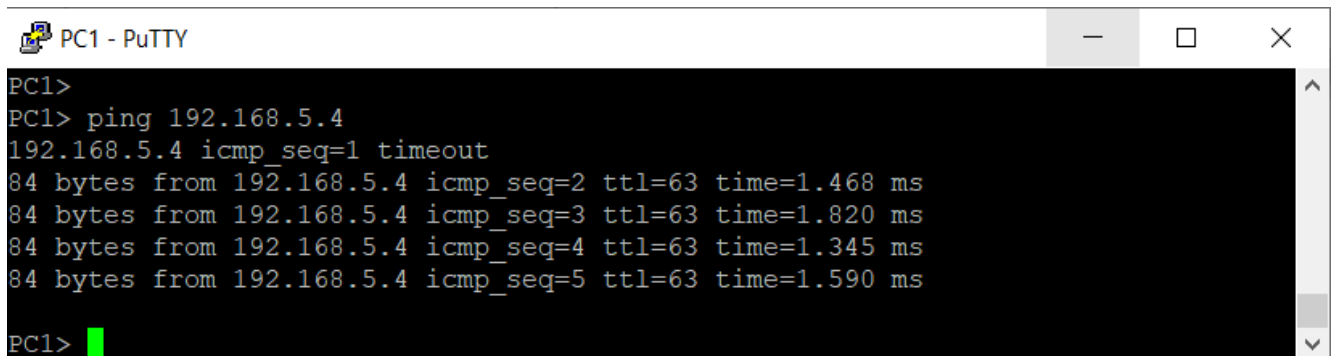
    192.168.3.0/24 is variably subnetted, 4 subnets, 3 masks
C       192.168.3.0/29 is directly connected, Vlan100
L       192.168.3.1/32 is directly connected, Vlan100
C       192.168.3.8/30 is directly connected, Ethernet0/0
L       192.168.3.10/32 is directly connected, Ethernet0/0
    192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.4.0/24 is directly connected, Vlan10
L       192.168.4.1/32 is directly connected, Vlan10
    192.168.5.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.5.0/24 is directly connected, Vlan20
L       192.168.5.1/32 is directly connected, Vlan20
MS1#
*Jun 27 18:53:56.074: %SYS-5-CONFIG_I: Configured from console by console
MS1#

```

Рис. 3.14 - Налаштування та перевірка маршрутизації

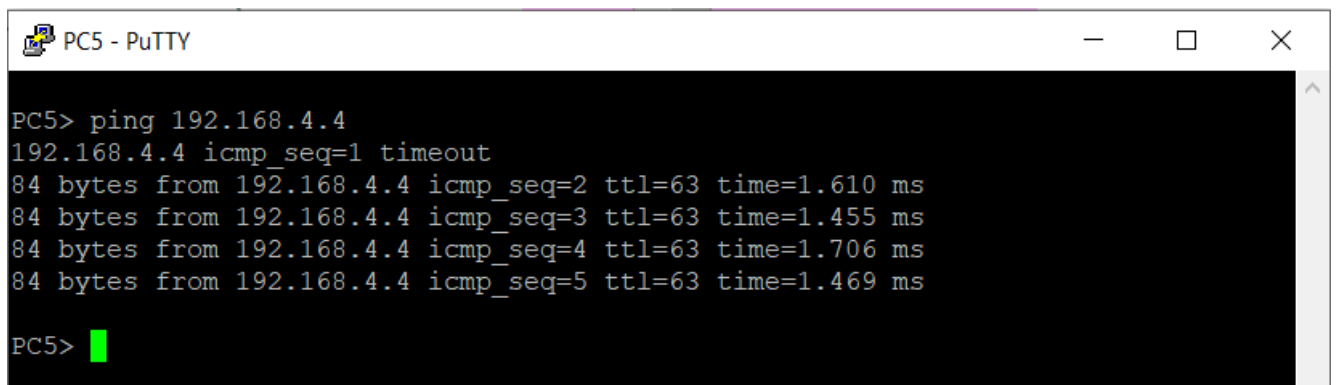
		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		56

Перевірка зв'язку між пристроями віддалених підмереж:



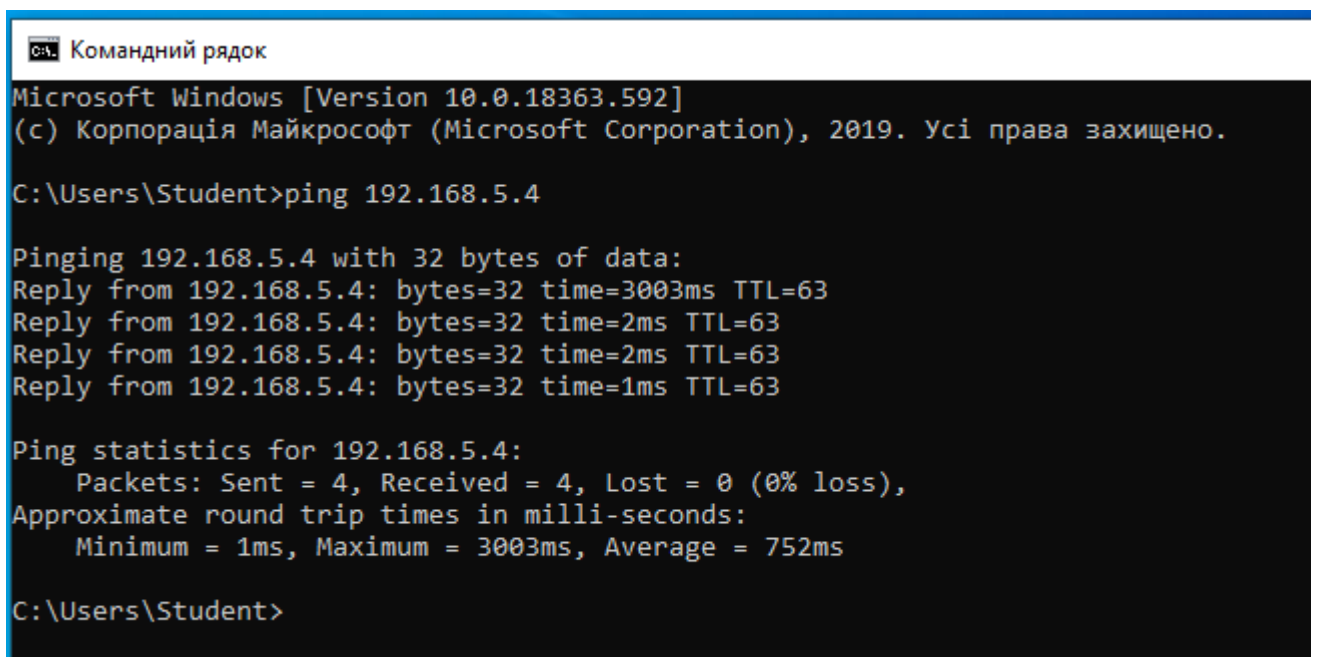
```
PC1>
PC1> ping 192.168.5.4
192.168.5.4 icmp_seq=1 timeout
84 bytes from 192.168.5.4 icmp_seq=2 ttl=63 time=1.468 ms
84 bytes from 192.168.5.4 icmp_seq=3 ttl=63 time=1.820 ms
84 bytes from 192.168.5.4 icmp_seq=4 ttl=63 time=1.345 ms
84 bytes from 192.168.5.4 icmp_seq=5 ttl=63 time=1.590 ms
PC1>
```

Рис. 3.15 - Зв'язок PC1 з PC5



```
PC5> ping 192.168.4.4
192.168.4.4 icmp_seq=1 timeout
84 bytes from 192.168.4.4 icmp_seq=2 ttl=63 time=1.610 ms
84 bytes from 192.168.4.4 icmp_seq=3 ttl=63 time=1.455 ms
84 bytes from 192.168.4.4 icmp_seq=4 ttl=63 time=1.706 ms
84 bytes from 192.168.4.4 icmp_seq=5 ttl=63 time=1.469 ms
PC5>
```

Рис. 3.16 - Зв'язок PC5 з PC2



```
Командний рядок
Microsoft Windows [Version 10.0.18363.592]
(c) Корпорація Майкрософт (Microsoft Corporation), 2019. Усі права захищено.
C:\Users\Student>ping 192.168.5.4

Pinging 192.168.5.4 with 32 bytes of data:
Reply from 192.168.5.4: bytes=32 time=3003ms TTL=63
Reply from 192.168.5.4: bytes=32 time=2ms TTL=63
Reply from 192.168.5.4: bytes=32 time=2ms TTL=63
Reply from 192.168.5.4: bytes=32 time=1ms TTL=63

Ping statistics for 192.168.5.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 3003ms, Average = 752ms
C:\Users\Student>
```

Рис. 3.17 - Зв'язок PC3 з PC5

```

Командний рядок
C:\Users\Student>ping 192.168.4.1

Pinging 192.168.4.1 with 32 bytes of data:
Reply from 192.168.4.1: bytes=32 time=1ms TTL=255
Reply from 192.168.4.1: bytes=32 time=1ms TTL=255
Reply from 192.168.4.1: bytes=32 time=1ms TTL=255
Reply from 192.168.4.1: bytes=32 time=1ms TTL=255

Ping statistics for 192.168.4.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms

C:\Users\Student>ping 192.168.4.3

Pinging 192.168.4.3 with 32 bytes of data:
Reply from 192.168.4.3: bytes=32 time=3003ms TTL=63
Reply from 192.168.4.3: bytes=32 time=1ms TTL=63
Reply from 192.168.4.3: bytes=32 time=2ms TTL=63
Reply from 192.168.4.3: bytes=32 time=1ms TTL=63

Ping statistics for 192.168.4.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 3003ms, Average = 751ms

C:\Users\Student>

```

Рис. 3.18 - Зв'язок PC0 з MS1 (мережа C) та PC1

Пристрої різних підмереж мають зв'язок між собою.

Тепер перейдемо до налаштувань безпеки. Вимикаємо можливість віддаленого доступу та створюємо користувачів.

```

SW3(config)#enable secret n-<tA25f8
SW3(config)#service password-encryption
SW3(config)#line con 0
SW3(config-line)#login local
SW3(config-line)#line vty 0 4
SW3(config-line)#login local
SW3(config-line)#username campusadmin1 privilege 15 secret y3KJ%f@iAJ801!
SW3(config)#username campusadmin1 privilege 15 secret y3KJ%f@iAJ801!
SW3(config)#username campustech1 privilege 5 secret I$~qfi;)4o1Ylt

SW3(config)#line vty 0 4
SW3(config-line)#transport output none
SW3(config-line)#transport input none
SW3(config-line)#end

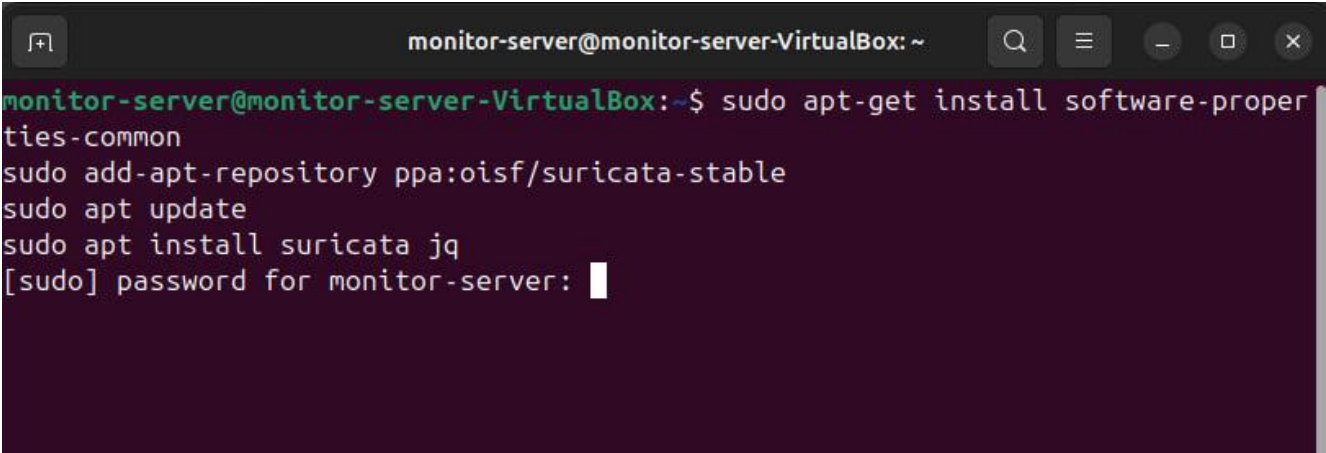
```

Рис. 3.19 - Налаштування безпеки на SW3

Налаштування на інших мережевих пристроях є аналогічним.

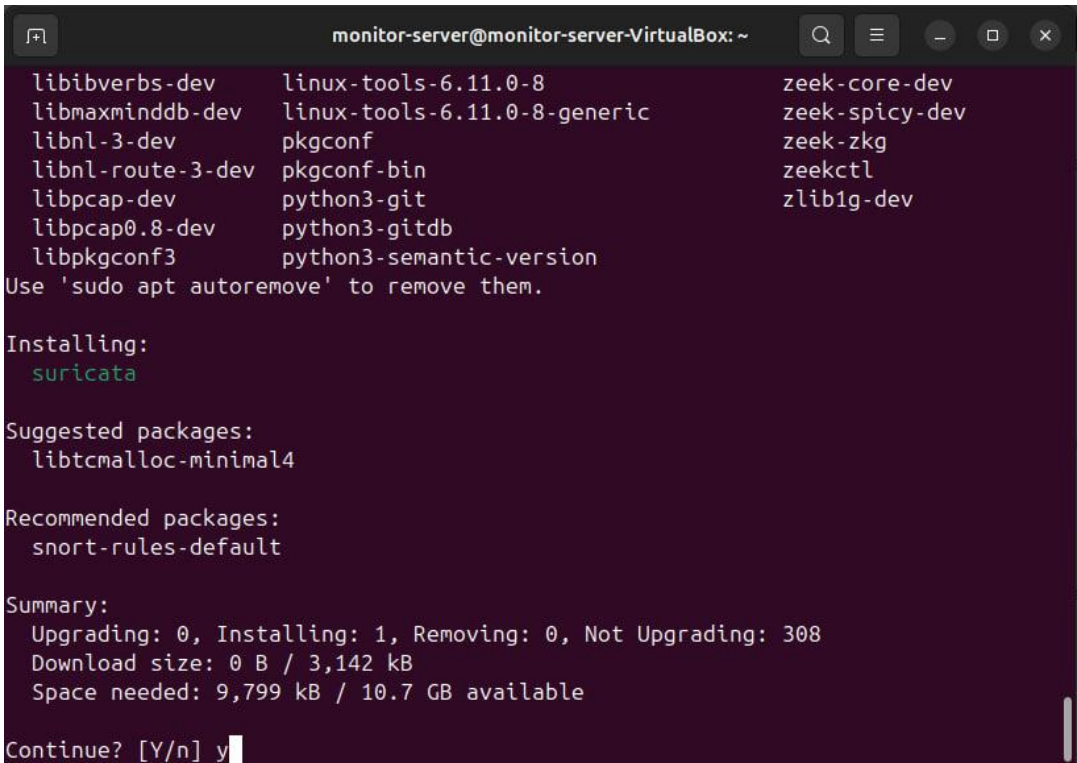
3.3 Встановлення Suricata

Першим кроком буде завантаження стабільної версії, керуючись офіційним процесом встановлення [21].



```
monitor-server@monitor-server-VirtualBox: ~  
monitor-server@monitor-server-VirtualBox:~$ sudo apt-get install software-properties-common  
sudo add-apt-repository ppa:oisf/suricata-stable  
sudo apt update  
sudo apt install suricata jq  
[sudo] password for monitor-server: 
```

Рис. 3.20 - Завантаження стабільної версії Suricata



```
monitor-server@monitor-server-VirtualBox: ~  
libibverbs-dev      linux-tools-6.11.0-8      zeek-core-dev  
libmaxminddb-dev    linux-tools-6.11.0-8-generic zeek-spicy-dev  
libnl-3-dev         pkgconf                   zeek-zkg  
libnl-route-3-dev   pkgconf-bin              zeekctl  
libpcap-dev         python3-git               zlib1g-dev  
libpcap0.8-dev      python3-gitdb  
libpkgconf3         python3-semantic-version  
Use 'sudo apt autoremove' to remove them.  
  
Installing:  
  suricata  
  
Suggested packages:  
  libtcmalloc-minimal4  
  
Recommended packages:  
  snort-rules-default  
  
Summary:  
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 308  
  Download size: 0 B / 3,142 kB  
  Space needed: 9,799 kB / 10.7 GB available  
  
Continue? [Y/n] y
```

Рис. 3.21 - Процес встановлення

Після встановлення Suricata ви можете перевірити, яка версія Suricata у вас запущена та з якими опціями, а також стан служби: `sudo suricata --build-info` чи `sudo systemctl status suricata`.

```
monitor-server@monitor-server-VirtualBox: ~
processing triggers for man-db (2.12.1-3) ...
monitor-server@monitor-server-VirtualBox:~$ sudo suricata --build-info
This is Suricata version 7.0.6 RELEASE
Features: NFQ PCAP_SET_BUFF AF_PACKET HAVE_PACKET_FANOUT LIBCAP_NG LIBNET1.1 HAVE_HTTP_URI_NORMALIZE_HOOK PCRE_JIT HAVE_NSS HTTP2_DECOMPRESSION HAVE_LUA HAVE_JA3 HAVE_JA4 HAVE_LUAJIT HAVE_LIBJANSSON TLS TLS_C11 MAGIC RUST P
OPCNT64
SIMD support: SSE_4_2 SSE_4_1 SSE_3 SSE_2
Atomic intrinsics: 1 2 4 8 16 byte(s)
64-bits, Little-endian architecture
GCC version 13.3.0, C version 201112
compiled with _FORTIFY_SOURCE=2
L1 cache line size (CLS)=64
thread local storage method: _Thread_local
compiled with LibHTP v0.5.48, linked against LibHTP v0.5.48

Suricata Configuration:
AF_PACKET support: yes
AF_XDP support: yes
DPDK support: yes
eBPF support: yes
XDP support: yes
PF_RING support: no
NFQueue support: yes
NFLOG support: yes
IPFW support: no
Netmap support: no
DAG enabled: no
Napatech enabled: no
WinDivert enabled: no
Unix socket enabled: yes
```

Рис. 3.22 - Перевірка встановлення

Далі виконуємо запуск Suricata за допомогою `systemctl` та перевіряємо роботу за допомогою `systemctl status`.

```
monitor-server@monitor-server-VirtualBox: ~
suricata.service - Suricata IDS/IDP daemon
Loaded: loaded (/usr/lib/systemd/system/suricata.service; enabled; preset: enabled)
Active: failed (Result: exit-code) since Sat 2025-06-28 18:56:21 UTC; 1min 31s ago
Duration: 101ms
Invocation: f25206eff1194a8e8625dfaae1fe33a8
Docs: man:suricata(8)
man:suricatasc(8)
https://suricata.io/documentation/
Process: 1640 ExecStart=/usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/s
Main PID: 1670 (code=exited, status=1/FAILURE)

Jun 28 18:56:20 monitor-server-VirtualBox systemd[1]: suricata.service: Failed with result 'exit-code'.
Jun 28 18:56:21 monitor-server-VirtualBox systemd[1]: suricata.service: Scheduled restart job, restart count=
Jun 28 18:56:21 monitor-server-VirtualBox systemd[1]: suricata.service: Start request repeated too quickly.
Jun 28 18:56:21 monitor-server-VirtualBox systemd[1]: suricata.service: Failed with result 'exit-code'.
Jun 28 18:56:21 monitor-server-VirtualBox systemd[1]: Failed to start suricata.service - Suricata IDS/IDP da
```

Рис. 3.23 - Перевірка роботи Suricata

Такі помилки є очікуваними, адже файл з правилами та самі правила ми ще не створили. Програма успішно завантажено готова до налаштування.

3.4 Встановлення Netdata та Prometheus

За допомогою офіційно документації [24] Prometheus починаємо процес встановлення.

```
monitor-server@monitor-server-VirtualBox:~$ sudo groupadd --system prometheus
monitor-server@monitor-server-VirtualBox:~$ sudo useradd -s /sbin/nologin --system -g prometheus prometheus
monitor-server@monitor-server-VirtualBox:~$ sudo mkdir /etc/prometheus
monitor-server@monitor-server-VirtualBox:~$ sudo mkdir /var/lib/prometheus
monitor-server@monitor-server-VirtualBox:~$ wget https://github.com/prometheus/prometheus/releases/download/v2.43.0/prometheus-2.43.0.linux-amd64.tar.gz
--2025-06-28 10:02:10-- https://github.com/prometheus/prometheus/releases/download/v2.43.0/prometheus-2.43.0.linux-amd64.tar.gz
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)|140.82.121.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/6838921/cdff95e-2db9-48c1-8ee8-69bd587a304a?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250628%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250628T100210Z&X-Amz-Expires=1800&X-Amz-Signature=8d4157b09c177f1e14522f924c9d84600f4e11349ad3d671bdc1b163536c7d7&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dprometheus-2.43.0.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2025-06-28 10:02:10-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/6838921/cdff95e-2db9-48c1-8ee8-69bd587a304a?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250628%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250628T100210Z&X-Amz-Expires=1800&X-Amz-Signature=8d4157b09c177f1e14522f924c9d84600f4e11349ad3d671bdc1b163536c7d7&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dprometheus-2.43.0.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.111.133, 185.199.109.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 91091544 (87M) [application/octet-stream]
Saving to: 'prometheus-2.43.0.linux-amd64.tar.gz'

linux-amd64.tar.gz 62%[=====>] 54.06M 7.63MB/s eta 6s
```

Рис. 3.24 - Створення груп, папок та завантаження Prometheus з git

Після завантаження розпаковуємо архів.

```
monitor-server@monitor-server-VirtualBox:~$ tar vxvf prometheus*.tar.gz
prometheus-2.43.0.linux-amd64/
prometheus-2.43.0.linux-amd64/LICENSE
prometheus-2.43.0.linux-amd64/consoles/
prometheus-2.43.0.linux-amd64/consoles/prometheus.html
prometheus-2.43.0.linux-amd64/consoles/node-disk.html
prometheus-2.43.0.linux-amd64/consoles/node-overview.html
prometheus-2.43.0.linux-amd64/consoles/prometheus-overview.html
prometheus-2.43.0.linux-amd64/consoles/index.html.example
prometheus-2.43.0.linux-amd64/consoles/node-cpu.html
prometheus-2.43.0.linux-amd64/consoles/node.html
prometheus-2.43.0.linux-amd64/prometheus
prometheus-2.43.0.linux-amd64/promtool
prometheus-2.43.0.linux-amd64/NOTICE
prometheus-2.43.0.linux-amd64/console_libraries/
prometheus-2.43.0.linux-amd64/console_libraries/prom.lib
prometheus-2.43.0.linux-amd64/console_libraries/menu.lib
prometheus-2.43.0.linux-amd64/prometheus.yml
monitor-server@monitor-server-VirtualBox:~$ cd prometheus*/
monitor-server@monitor-server-VirtualBox:~/prometheus-2.43.0.linux-amd64$ sudo mv prometheus /usr/local/bin
monitor-server@monitor-server-VirtualBox:~/prometheus-2.43.0.linux-amd64$ sudo mv promtool /usr/local/bin
monitor-server@monitor-server-VirtualBox:~/prometheus-2.43.0.linux-amd64$ sudo chown prometheus:prometheus /usr/local/bin/prometheus
monitor-server@monitor-server-VirtualBox:~/prometheus-2.43.0.linux-amd64$ sudo chown prometheus:prometheus /usr/local/bin/promtool
monitor-server@monitor-server-VirtualBox:~/prometheus-2.43.0.linux-amd64$
```

Рис. 3.25 - Розпакування архіву

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		61

```

monitor-server@monitor-server-VirtualBox:~/prometheus-2.43.0.linux-amd64$ sudo mv consoles /etc/prometheus
monitor-server@monitor-server-VirtualBox:~/prometheus-2.43.0.linux-amd64$ sudo mv console_libraries /etc/prometheus
monitor-server@monitor-server-VirtualBox:~/prometheus-2.43.0.linux-amd64$ sudo mv prometheus.yml /etc/prometheus
monitor-server@monitor-server-VirtualBox:~/prometheus-2.43.0.linux-amd64$ sudo chown prometheus:prometheus /etc/prometheus
monitor-server@monitor-server-VirtualBox:~/prometheus-2.43.0.linux-amd64$ sudo chown -R prometheus:prometheus /etc/prometheus/consoles
monitor-server@monitor-server-VirtualBox:~/prometheus-2.43.0.linux-amd64$ sudo chown -R prometheus:prometheus /etc/prometheus/console_libraries
monitor-server@monitor-server-VirtualBox:~/prometheus-2.43.0.linux-amd64$ sudo chown -R prometheus:prometheus /var/lib/prometheus

```

Рис. 3.26 - Робота з файлами Prometheus

```

monitor-server@monitor-server-VirtualBox:~/prometheus-2.43.0.linux-amd64$ sudo systemctl status prometheus
● prometheus.service - Prometheus
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; preset: enabled)
   Active: active (running) since Sat 2025-06-28 10:09:37 UTC; 37s ago
 Invocation: 231a45cb69e6423b95761451a24a1087
   Main PID: 9732 (prometheus)
    Tasks: 6 (limit: 6663)
   Memory: 17.5M (peak: 17.5M)
      CPU: 81ms
   CGroup: /system.slice/prometheus.service
           └─9732 /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --storage.tsdb.path /var>

Jun 28 10:09:37 monitor-server-VirtualBox prometheus[9732]: ts=2025-06-28T10:09:37.435Z caller=head.go:658 level=info>
Jun 28 10:09:37 monitor-server-VirtualBox prometheus[9732]: ts=2025-06-28T10:09:37.435Z caller=head.go:664 level=info>
Jun 28 10:09:37 monitor-server-VirtualBox prometheus[9732]: ts=2025-06-28T10:09:37.436Z caller=head.go:735 level=info>
Jun 28 10:09:37 monitor-server-VirtualBox prometheus[9732]: ts=2025-06-28T10:09:37.436Z caller=head.go:772 level=info>
Jun 28 10:09:37 monitor-server-VirtualBox prometheus[9732]: ts=2025-06-28T10:09:37.440Z caller=main.go:1026 level=info>
Jun 28 10:09:37 monitor-server-VirtualBox prometheus[9732]: ts=2025-06-28T10:09:37.441Z caller=main.go:1029 level=info>
Jun 28 10:09:37 monitor-server-VirtualBox prometheus[9732]: ts=2025-06-28T10:09:37.441Z caller=main.go:1209 level=info>
Jun 28 10:09:37 monitor-server-VirtualBox prometheus[9732]: ts=2025-06-28T10:09:37.442Z caller=main.go:1246 level=info>
Jun 28 10:09:37 monitor-server-VirtualBox prometheus[9732]: ts=2025-06-28T10:09:37.442Z caller=main.go:990 level=info>
Jun 28 10:09:37 monitor-server-VirtualBox prometheus[9732]: ts=2025-06-28T10:09:37.442Z caller=manager.go:974 level=info>
ESCOC
lines 1-21/21 (END)
monitor-server@monitor-server-VirtualBox:~/prometheus-2.43.0.linux-amd64$ sudo ufw allow 9090/tcp
Rules updated
Rules updated (v6)
monitor-server@monitor-server-VirtualBox:~/prometheus-2.43.0.linux-amd64$

```

Рис. 3.27 - Запуск Prometheus

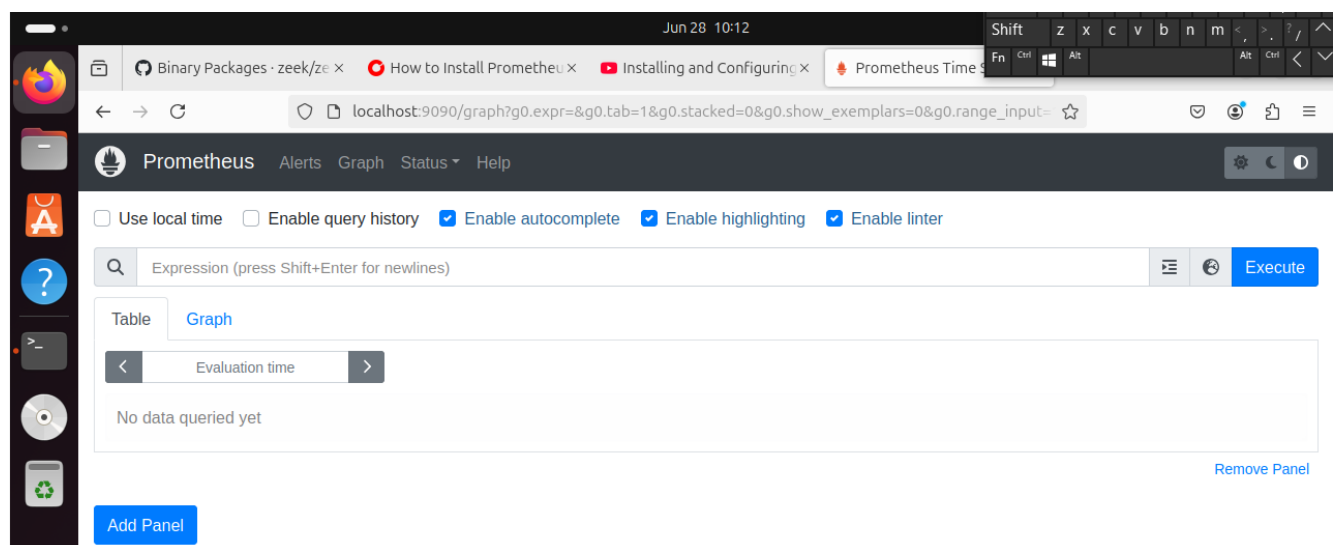


Рис. 3.28 - Перевірка роботи в браузері

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				62
Змн.	Арк.	№ докум.	Підпис	Дата		

Встановлення NetData виконуємо за інформацією з офіційного сайту[23].

```
monitor-server@monitor-server-VirtualBox: ~
monitor-server@monitor-server-VirtualBox: ~/prometheus-2.43.0.linux-amd64$ cd
monitor-server@monitor-server-VirtualBox:~$ sudo apt install netdata
[sudo] password for monitor-server:
The following packages were automatically installed and are no longer required:
linux-headers-6.11.0-8 linux-modules-6.11.0-8-generic linux-tools-6.11.0-8
linux-headers-6.11.0-8-generic linux-modules-extra-6.11.0-8-generic linux-tools-6.11.0-8-generic
Use 'sudo apt autoremove' to remove them.

Installing:
netdata

Installing dependencies:
fonts-font-awesome libjs-bootstrap netdata-core netdata-plugins-python
fonts-glyphicons-halflings libnetfilter-acct1 netdata-plugins-bash netdata-web

Suggested packages:
apcupsd lm-sensors nc fping python3-psycopp2 python3-pymysql

Summary:
Upgrading: 0, Installing: 9, Removing: 0, Not Upgrading: 311
Download size: 16.6 MB
Space needed: 63.9 MB / 13.0 GB available

Continue? [Y/n] y
Get:1 http://ua.archive.ubuntu.com/ubuntu oracular/main amd64 fonts-font-awesome all 5.0.10+really4.7.0-4.1 [51
6 kB]
Get:2 http://ua.archive.ubuntu.com/ubuntu oracular-updates/universe amd64 fonts-glyphicons-halflings all 1.009-3.4.1
+dfsg-3+deb12u1build0.24.10.1 [119 kB]
Get:3 http://ua.archive.ubuntu.com/ubuntu oracular-updates/universe amd64 libjs-bootstrap all 3.4.1+dfsg-3+deb12u1bu
ild0.24.10.1 [129 kB]
Get:4 http://ua.archive.ubuntu.com/ubuntu oracular/universe amd64 libnetfilter-acct1 amd64 1.0.3-4 [7,318 B]
```

Рис. 3.29 - Встановлення NetData

```
monitor-server@monitor-server-VirtualBox:~$ sudo systemctl start netdata && sudo systemctl enable netdata
monitor-server@monitor-server-VirtualBox:~$ sudo systemctl status netdata
● netdata.service - netdata - Real-time performance monitoring
   Loaded: loaded (/usr/lib/systemd/system/netdata.service; enabled; preset: enabled)
   Active: active (running) since Sat 2025-06-28 10:14:19 UTC; 10s ago
     Invocation: 3a5f54728f2544caba96c298bb037bfc
       Docs: man:netdata
             file:///usr/share/doc/netdata/html/index.html
             https://github.com/netdata/netdata
   Main PID: 10437 (netdata)
     Tasks: 64 (limit: 6663)
    Memory: 70.5M (peak: 70.8M)
       CPU: 1.623s
    CGroup: /system.slice/netdata.service
            └─10437 /usr/sbin/netdata -D
              └─10587 /usr/sbin/netdata --special-spawn-server
                └─10810 bash /usr/lib/netdata/plugins.d/tc-qos-helper.sh 1
                  └─10818 /usr/lib/netdata/plugins.d/apps.plugin 1
                    └─10824 /usr/bin/python3 /usr/lib/netdata/plugins.d/python.d.plugin 1
                      └─10826 /usr/lib/netdata/plugins.d/cups.plugin 1
                        └─10831 /usr/lib/netdata/plugins.d/nfacct.plugin 1

Jun 28 10:14:24 monitor-server-VirtualBox netdata[10824]: python.d INFO: plugin[main] : [samba] is disabled by defa
```

Рис. 3.30 - Увімкнення Netdata

```
monitor-server@monitor-server-VirtualBox:~$ netstat -tunlp | grep 19999
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
tcp        0      0 127.0.0.1:19999      0.0.0.0:*              LISTEN      -
monitor-server@monitor-server-VirtualBox:~$
```

Рис. 3.31 - Перевірка налаштувань NetData

3.5 Налаштування системи виявлення та протидії експільтрації ICMP-протоколу

За допомогою whereis визначаємо папки Suricata. Створюємо бекапу конфігурацію за допомогою копіювання для відновлення у разі помилок налаштування

```
monitor-server@monitor-server-VirtualBox:~$ whereis suricata
suricata: /usr/bin/suricata /usr/lib/suricata /etc/suricata /usr/share/man/man1/suricata.1.gz
monitor-server@monitor-server-VirtualBox:~$ cd /etc/suricata
monitor-server@monitor-server-VirtualBox:/etc/suricata$ cp suricata.yaml suricata.yaml.backup
cp: cannot create regular file 'suricata.yaml.backup': Permission denied
monitor-server@monitor-server-VirtualBox:/etc/suricata$ sudo cp suricata.yaml suricata.yaml.backup
monitor-server@monitor-server-VirtualBox:/etc/suricata$ ls
classification.config reference.config rules suricata.yaml suricata.yaml.backup threshold.config
monitor-server@monitor-server-VirtualBox:/etc/suricata$
```

Рис. 3.32 - Створення бекапу

```
monitor-server@monitor-server-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.106 netmask 255.255.255.0 broadcast 192.168.0.255
    ether 08:00:27:53:6a:48 txqueuelen 1000 (Ethernet)
    RX packets 695 bytes 223723 (223.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 768 bytes 108799 (108.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 317 bytes 202662 (202.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 317 bytes 202662 (202.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
monitor-server@monitor-server-VirtualBox:~$
```

Рис. 3.33 - Перевірка адресації Monitor-Server (не підключеного до схеми)

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		64

Звідси потрібно запам'ятати назву інтерфейсу – enp0s3 та його IP-адресу. Переходимо у файл suricata.yaml і редагуємо його за допомогою nano.

```

monitor-server@monitor-server-VirtualBox: /etc/suricata
monitor-server@monitor-server-VirtualBox: ~
monitor-server@monitor-server-VirtualBox: ~
GNU nano 8.1 suricata.yaml *
%YAML 1.1
---
# Suricata configuration file. In addition to the comments describing all
# options in this file, full documentation can be found at:
# https://docs.suricata.io/en/latest/configuration/suricata-yaml.html
# This configuration file generated by Suricata 7.0.6.
suricata-version: "7.0"

##
## Step 1: Inform Suricata about your network
##

vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "[192.168.3.0/29]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    #HOME_NET: "[172.16.0.0/12]"
    #HOME_NET: "any"
  
```

Рис. 3.34 - Поле з адресою підмережі

```

monitor-server@monitor-server-VirtualBox: /etc/suricata
monitor-server@monitor-server-VirtualBox: ~
GNU nano 8.1 suricata.yaml *
##
## Step 3: Configure common capture settings
##
## See "Advanced Capture Options" below for more options, including Netmap
## and PF_RING.
##
# Linux high speed capture support
af-packet:
  - interface: enp0s3
    # Number of receive threads. "auto" uses the number of cores
    #threads: auto
    # Default clusterid. AF_PACKET will load balance packets based on flow.
    cluster-id: 99
    # Default AF_PACKET cluster type. AF_PACKET can load balance per flow or per hash.
    # This is only supported for Linux kernel > 3.1
    # possible value are:
    # * cluster_flow: all packets of a given flow are sent to the same socket
  
```

Рис. 3.35 - Поле з інтерфейсом

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				65
Змн.	Арк.	№ докум.	Підпис	Дата		

```

monitor-server@monitor-server-VirtualBox: /etc/suricata
monitor-server@monitor-server-VirtualBox: /etc/suricata  x  monitor-server@monitor-server-VirtualBox: ~  x  v
GNU nano 8.1 suricata.yaml *
# This parameter has no effect if auto-config is disabled.
#
hashmode: hash5tuplesorted

##
## Configure Suricata to load Suricata-Update managed rules.
##

default-rule-path: /var/lib/suricata/rules

rule-files:
#- suricata.rules
- custom.rules

##
## Auxiliary configuration files.
##

```

Рис. 3.36 - у полі rules-files додаємо custom.rules

```

monitor-server@monitor-server-VirtualBox: /etc/suricata
monitor-server@monitor-server-VirtualBox: /etc/suricata  x  monitor-server@monitor-server-VirtualBox: ~  x  v
GNU nano 8.1 suricata.yaml *
#bpf-filter: port 80 or udp
#- interface: eth3
#threads: auto
#copy-mode: tap
#copy-iface: eth2
# Put default values here
- interface: default

# PF_RING configuration: for use with native PF_RING support
# for more info see http://www.ntop.org/products/pf_ring/
pfiring:
- interface: enp0s3
# Number of receive threads. If set to 'auto' Suricata will first try
# to use CPU (core) count and otherwise RSS queue count.
threads: auto

# Default clusterid. PF_RING will load balance packets based on flow.
# All threads/processes that will participate need to have the same
# clusterid.
cluster-id: 99

^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo

```

Рис. 3.37 - Змінюємо значення інтерфейсу на enp0s3

Тепер переходимо до var/lib та перевіряємо чи там є папка suricata. Використовуємо команду ls -l.

```
monitor-server@monitor-server-VirtualBox: /var/lib
monitor-server@monitor-server-VirtualBox: /var/lib
monitor-server@monitor-server-VirtualBox: ~
drwxr-xr-x 2 root root 4096 Jun 28 09:37 sgml-base
-rw-r--r-- 1 root root 84 Oct 9 2024 shells.state
drwxr-xr-x 24 root root 4096 Jun 28 18:56 snapd
drwxr-xr-x 2 root root 4096 Sep 4 2024 snmp
drwxr-xr-x 10 root root 4096 Oct 9 2024 sss
drwxr-xr-x 3 root root 4096 Oct 9 2024 sudo
drwxr-xr-x 3 root root 4096 Jun 28 12:07 suricata
drwxr-xr-x 4 root root 4096 Jun 28 18:51 swcatalog
drwxr-xr-x 12 root root 4096 Jun 28 09:27 systemd
drwxr-xr-x 2 tss tss 4096 Jul 29 2024 tpm
drwxr-xr-x 2 root root 4096 Oct 4 2024 ubuntu-advantage
drwxr-xr-x 2 root root 4096 Jun 28 09:27 ubuntu-drivers-common
drwxr-xr-x 2 root root 4096 Oct 3 2024 ubuntu-release-upgrader
drwxr-xr-x 3 root root 4096 Jun 28 09:14 ucf
drwxr-xr-x 2 root root 4096 Jun 28 18:57 udisks2
drwxr-xr-x 2 root root 4096 Feb 12 2024 unattended-upgrades
drwxr-xr-x 2 root root 4096 Sep 24 2024 update-manager
drwxr-xr-x 4 root root 4096 Jun 28 18:52 update-notifier
drwxr-xr-x 2 root root 4096 Jun 28 19:06 upower
drwxr-xr-x 2 root root 4096 Jul 15 2024 usb_modeswitch
drwxr-xr-x 3 root root 4096 Oct 9 2024 vin
drwxr-xr-x 2 root root 4096 Jun 28 16:03 whoopsie
drwxr-xr-x 2 root root 4096 Oct 9 2024 xfonts
drwxr-xr-x 2 root root 4096 Oct 9 2024 xkb
drwxr-xr-x 2 root root 4096 Jun 28 09:37 xml-core
monitor-server@monitor-server-VirtualBox: /var/lib$
```

Рис. 3.38 - Наявність папки suricata

Позаяк папку створювати не потрібно, перейдемо до написання правил. Створюємо файл custom-rules, який перед цим зазначили в suricata.yaml, та перезапускаємо suricata. Одна з помилок має зникнути.

```
monitor-server@monitor-server-VirtualBox: /var/lib$ cd suricata
monitor-server@monitor-server-VirtualBox: /var/lib/suricata$ ls
rules
monitor-server@monitor-server-VirtualBox: /var/lib/suricata$ sudo touch custom.rules
monitor-server@monitor-server-VirtualBox: /var/lib/suricata$ ls
custom.rules rules
monitor-server@monitor-server-VirtualBox: /var/lib/suricata$
```

Рис. 3.39 - Створення файлу

```
monitor-server@monitor-server-VirtualBox: ~
monitor-server@monitor-server-VirtualBox: /var/lib/suricata$ cd
monitor-server@monitor-server-VirtualBox: ~$ sudo suricata -c /etc/suricata/suricata.yaml -i enp0s3
[sudo] password for monitor-server:
Info: conf-yaml-loader: Configuration node 'types' redefined.
i: suricata: This is Suricata version 7.0.6 RELEASE running in SYSTEM mode
W: detect: No rule files match the pattern /var/lib/suricata/rules/custom.rules
W: detect: 1 rule files specified, but no rules were loaded!
i: threads: Threads created -> W: 2 FM: 1 FR: 1 Engine started.
```

Рис. 3.40 - Перезапуск Suricata

Редагуємо файл та вписуємо туди правило.

```
monitor-server@monitor-server-VirtualBox: /var/lib/suricata
GNU nano 8.1 /var/lib/suricata/custom.rules *
alert icmp any any -> $HOME_NET any (msg:"INCOMING ICMP PACKETS!"; sid:1000001; rev:1;)
```

Рис. 3.41 - Написання правила

```
monitor-server@monitor-server-VirtualBox:~$ sudo suricata -c /etc/suricata/suricata.yaml -i enp0s3
[sudo] password for monitor-server:
Info: conf-yaml-loader: Configuration node 'types' redefined.
i: suricata: This is Suricata version 7.0.6 RELEASE running in SYSTEM mode
E: af-packet: fanout not supported by kernel: Kernel too old or cluster-id 99 already in use.
i: threads: Threads created -> W: 1 FM: 1 FR: 1 Engine started.
```

Рис. 3.42 - Перезапуск Suricata

У випадку, коли будь який пристрій відправить на Monitor-Server ICMP-пакети, відповідне логування з'явиться у fast.log.

```
monitor-server@monitor-server-VirtualBox:~$ sudo tail -f /var/log/suricata/fast.log
06/28/2025-19:57:01.536138  [**] [1:1000003:1] INCOMING ICMP PACKETS! [**] [Classification: (null)] [Priority: 3] {ICMP
} 192.168.3.1:3 -> 192.168.3.3:1
06/28/2025-19:57:01.537281  [**] [1:1000003:1] INCOMING ICMP PACKETS! [**] [Classification: (null)] [Priority: 3] {ICMP
} 192.168.3.1:3 -> 192.168.3.3:1
06/28/2025-19:57:01.538378  [**] [1:1000003:1] INCOMING ICMP PACKETS! [**] [Classification: (null)] [Priority: 3] {ICMP
} 192.168.3.1:3 -> 192.168.3.3:1
06/28/2025-19:57:01.539320  [**] [1:1000003:1] INCOMING ICMP PACKETS! [**] [Classification: (null)] [Priority: 3] {ICMP
} 192.168.3.1:3 -> 192.168.3.3:1
06/28/2025-19:57:01.539979  [**] [1:1000003:1] INCOMING ICMP PACKETS! [**] [Classification: (null)] [Priority: 3] {ICMP
} 192.168.3.1:3 -> 192.168.3.3:1
06/28/2025-19:57:01.540620  [**] [1:1000003:1] INCOMING ICMP PACKETS! [**] [Classification: (null)] [Priority: 3] {ICMP
} 192.168.3.1:3 -> 192.168.3.3:1
06/28/2025-19:57:01.541827  [**] [1:1000003:1] INCOMING ICMP PACKETS! [**] [Classification: (null)] [Priority: 3] {ICMP
} 192.168.3.1:3 -> 192.168.3.3:1
06/28/2025-19:57:01.542521  [**] [1:1000003:1] INCOMING ICMP PACKETS! [**] [Classification: (null)] [Priority: 3] {ICMP
} 192.168.3.1:3 -> 192.168.3.3:1
06/28/2025-19:57:01.543314  [**] [1:1000003:1] INCOMING ICMP PACKETS! [**] [Classification: (null)] [Priority: 3] {ICMP
} 192.168.3.1:3 -> 192.168.3.3:1
06/28/2025-19:57:01.543844  [**] [1:1000003:1] INCOMING ICMP PACKETS! [**] [Classification: (null)] [Priority: 3] {ICMP
} 192.168.3.1:3 -> 192.168.3.3:1
```

Рис. 3.43 - Приклади записів у fast.log.

Встановлення blackbox для створення пробів ICMP-протоколу. У майбутньому при розширенні системи це допоможе інтегрувати більш складні інструменти аналізу.

```
monitor-server@monitor-server-VirtualBox:~$ cd blackbox_exporter
bash: cd: blackbox_exporter: No such file or directory
monitor-server@monitor-server-VirtualBox:~$ ls -l
total 93852
drwxrwxr-x 2 monitor-server monitor-server 4096 Feb 27 2018 blackbox_exporter-0.12.0.linux-amd64
-rw-rw-r-- 1 monitor-server monitor-server 4950885 Dec 8 2021 blackbox_exporter-0.12.0.linux-amd64.tar.gz
drwxr-xr-x 2 monitor-server monitor-server 4096 Jun 28 09:28 Desktop
drwxr-xr-x 2 monitor-server monitor-server 4096 Jun 28 09:28 Documents
drwxr-xr-x 2 monitor-server monitor-server 4096 Jun 28 09:28 Downloads
drwxrwxr-x 3 monitor-server monitor-server 4096 Jun 28 12:28 go
drwxr-xr-x 2 monitor-server monitor-server 4096 Jun 28 09:28 Music
drwxrwxr-x 2 monitor-server monitor-server 4096 Jun 28 11:24 netflow-data
drwxr-xr-x 2 monitor-server monitor-server 4096 Jun 28 09:28 Pictures
drwxr-xr-x 2 monitor-server monitor-server 4096 Jun 28 10:04 prometheus-2.43.0.linux-amd64
-rw-rw-r-- 1 monitor-server monitor-server 91091544 Mar 21 2023 prometheus-2.43.0.linux-amd64.tar.gz
drwxr-xr-x 2 monitor-server monitor-server 4096 Jun 28 09:28 Public
drwx----- 9 monitor-server monitor-server 4096 Jun 29 05:17 snap
drwxrwxr-x 5 monitor-server monitor-server 4096 Jun 28 12:29 suricata_exporter
drwxr-xr-x 2 monitor-server monitor-server 4096 Jun 28 09:28 Templates
drwxr-xr-x 2 monitor-server monitor-server 4096 Jun 28 09:28 Videos
monitor-server@monitor-server-VirtualBox:~$ cd blackbox_exporter-0.12.0.linux-amd64
monitor-server@monitor-server-VirtualBox:~/blackbox_exporter-0.12.0.linux-amd64$ sudo ./blackbox_exporter
level=info ts=2025-06-29T05:19:51.256445159Z caller=main.go:213 msg="Starting blackbox_exporter" version="(version=0.12.0, branch=HEAD, revision=4a22586cf0cf139d9b2f9cde099f0012d9fcabde)"
level=info ts=2025-06-29T05:19:51.257034918Z caller=main.go:220 msg="Loaded config file"
level=info ts=2025-06-29T05:19:51.25742204Z caller=main.go:324 msg="Listening on address" address=:9115
```

Рис. 3.44 - Завантаження та встановлення BlackBox

Для того, щоб інтегрувати BlackBox в Prometheus, змінимо конфігураційний файл.

```
monitor-server@monitor-server-VirtualBox: /etc/prometheus
GNU nano 8.1 prometheus.yml
# - "first_rules.yml"
# - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"
    static_configs:
      - targets: ["localhost:9005"]
  - job_name: "blackbox"
    metrics_path: /probe
    params:
      module: [icmp]
    static_configs:
      - targets:
        - localhost
    relabel_configs:
      - source_labels: [__address__]
        target_label: __param_target
      - source_labels: [__param_target]
        target_label: instance
      - target_label: __address__
        replacement: 127.0.0.1:9115
```

Рис. 3.45 - Зміни у prometheus.yml

У конфігураційному файлі BlackBox створюємо пробер дл ICMP-трафіку.

```

GNU nano 8.1 blackbox.yml *
query_response:
  - expect: "^+OK"
  tls: true
  tls_config:
    insecure_skip_verify: false
ssh_banner:
  probe: tcp
  tcp:
    query_response:
      - expect: "^SSH-2.0-"
irc_banner:
  probe: tcp
  tcp:
    query_response:
      - send: "NICK prober"
      - send: "USER prober prober prober :prober"
      - expect: "PING :{[^\s]+}"
      - send: "PONG ${1}"
      - expect: "^[^\s]+ 001"
icmp:
  probe: icmp
icmp_ipv4:
  probe: icmp
icmp:
  preferred_ip_protocol: ip4
  
```

Рис. 3.46 - Зміни у blackbox.yml

Система майже готова, проте нам не потрібно, щоб фіксувалися усі ICMP-пакети, а лише ті, які потенційно можуть містити ексфільтрацію, тому змінимо тестове правило на ті, які створили раніше.

```

GNU nano 8.1 custom.rules
drop icmp any any -> any any (msg:"LARGE ICMP PACKET"; dsize:>50; sid:1000001; rev:1;)
drop icmp any any -> any any (msg:"ICMP FLOOD"; threshold:type both, track by_src, count 3, seconds 1, sid:1000002; rev:1;)
drop icmp any any -> any any (msg:"POSSIBLE PASSWORD EXFIL"; content:"password"; sid:1000003; rev:1;)
  
```

Рис. 3.47 - Правила для перехоплення ICMP-трафіку з потенційною ексфільтрацією

Тепер перейдемо до налаштування правил на кінцевих пристроях. Виконання налаштування виконуємо за допомогою оболонки PowerShell. Відкриваємо оболонку із правами адміністратора та вводимо попередньо розроблені команди.

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> New-NetFirewallRule -DisplayName "Block Fragmented ICMP" -Protocol ICMPv4 -IcmpType 3,4 -Direction Outbound -Action Block

Name                : {321e003a-5418-436d-920a-d1d0f1fc698c}
DisplayName          : Block Fragmented ICMP
Description          :
DisplayGroup         :
Group               :
Enabled              : True
Profile              : Any
Platform             : {}
Direction            : Outbound
Action               : Block
EdgeTraversalPolicy  : Block
LooseSourceMapping   : False
LocalOnlyMapping     : False
Owner                :
PrimaryStatus        : OK
Status               : Правило успішно проаналізовано зі сховища. (65536)
EnforcementStatus    : NotApplicable
PolicyStoreSource    : PersistentStore
PolicyStoreSourceType : Local

PS C:\Windows\system32> New-NetFirewallRule -DisplayName "Allow Internal ICMP Only" -Protocol ICMPv4 -Direction Outbound -RemoteAddress 192.168.4.0/24 -Action Allow

Name                : {d1c8f473-dd64-4f5c-abfb-52b1eba9e6ec}
DisplayName          : Allow Internal ICMP Only
Description          :
DisplayGroup         :
Group               :
Enabled              : True
Profile              : Any
Platform             : {}
Direction            : Outbound
Action               : Allow
EdgeTraversalPolicy  : Block
LooseSourceMapping   : False
LocalOnlyMapping     : False
Owner                :
PrimaryStatus        : OK
Status               : Правило успішно проаналізовано зі сховища. (65536)
EnforcementStatus    : NotApplicable
PolicyStoreSource    : PersistentStore
PolicyStoreSourceType : Local

```

Рис. 3.48 - Застосування правил обмеження трафіку на PC3

3.6 Тестування роботи системи

Тепер змодельюємо атаку з використанням ICMP-протоколу, використовуючи віртуальну машину Kali-Linux.

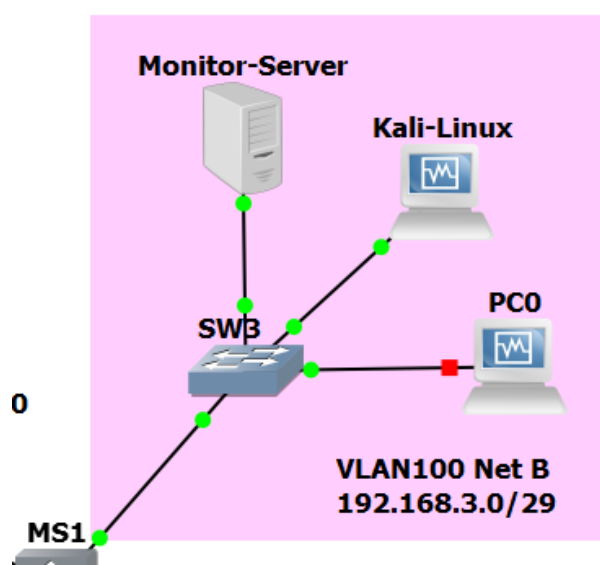


Рис. 3.49 - Підключення Kali Linux до SW3

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				71
Змн.	Арк.	№ докум.	Підпис	Дата		

Налаштуємо її адресу:

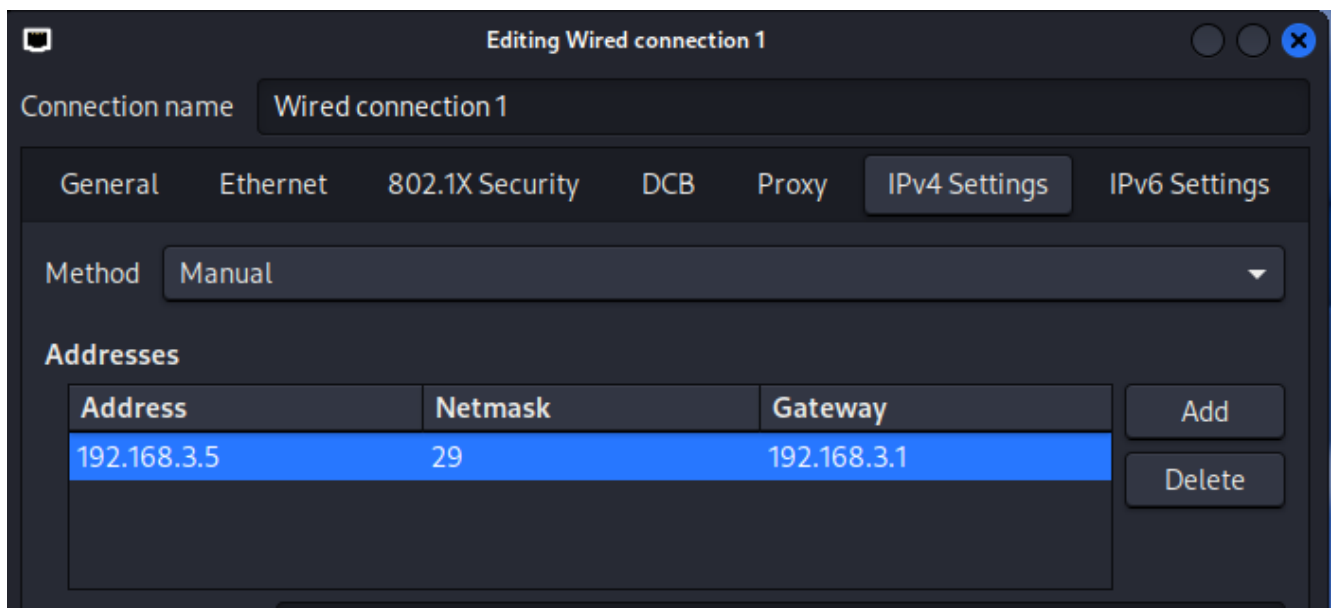


Рис. 3.50 - Налаштування адресації Kali-Linux

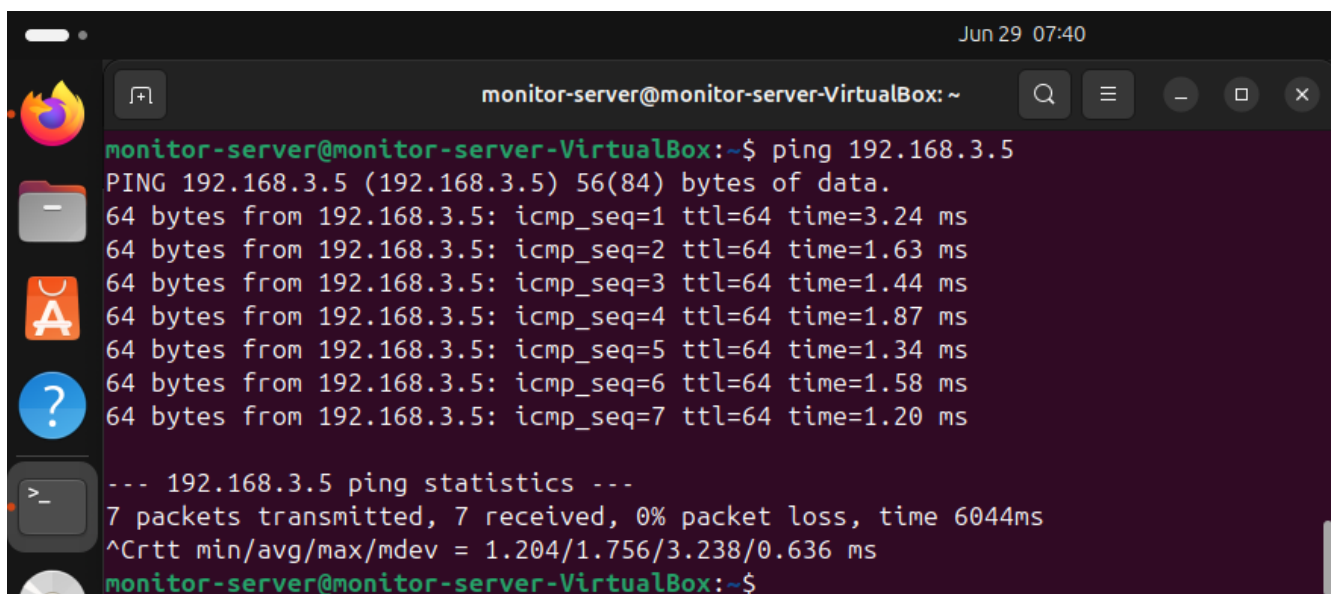


Рис. 3.51 - Перевірка зв'язку Monitor-Server з Kali-Linux

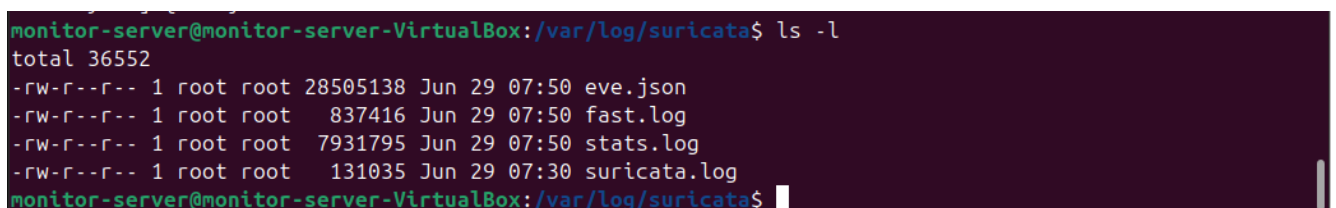


Рис. 3.52 - Переходимо в папку з логами

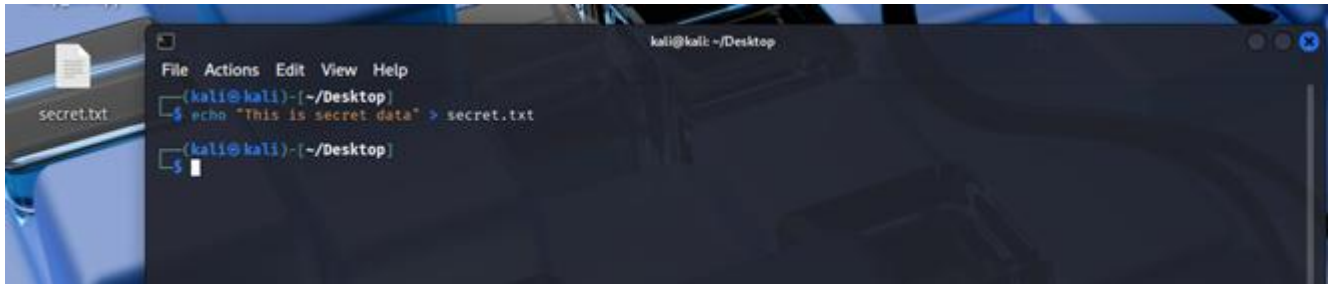


Рис. 3.53 - Створюємо файл для пересилання його вмісту

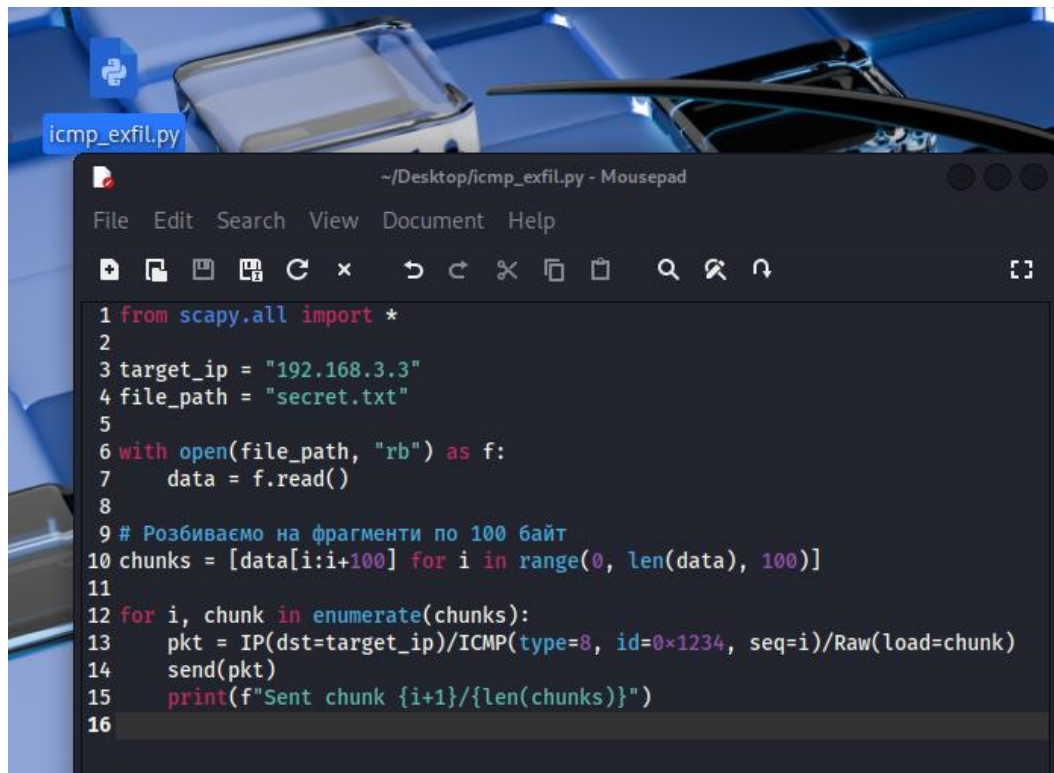


Рис. 3.54 - Скрипт атаки з відправленням пакетів великого розміру

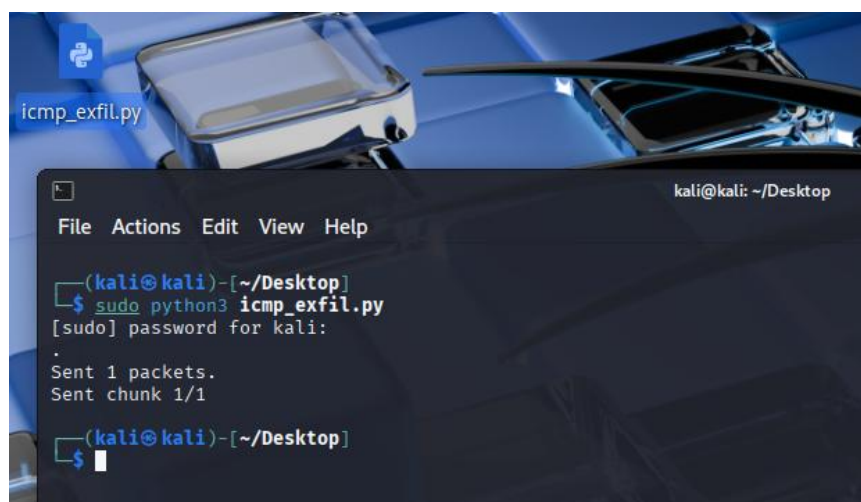


Рис. 3.55 - Ініціація атаки

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				73
Змн.	Арк.	№ докум.	Підпис	Дата		

```

Jun 29 13:38
monitor-server@monitor-server-VirtualBox: /var/log/suricata

monitor-server@monitor-server-... x monitor-server@monitor-server-... x monitor-server@monitor-server-... x monitor-server@monitor-server-... x
MP} 192.168.3.5:8 -> 192.168.3.3:0
06/29/2025-13:20:08.857639 [wDrop] [**] [1:1000001:1] LARGE ICMP PACKET [**] [Classification: (null)] [Priority: 3] {IC
MP} 192.168.3.3:0 -> 192.168.3.5:0
06/29/2025-13:20:08.885726 [wDrop] [**] [1:1000001:1] LARGE ICMP PACKET [**] [Classification: (null)] [Priority: 3] {IC
MP} 192.168.3.5:8 -> 192.168.3.3:0
06/29/2025-13:20:08.885829 [wDrop] [**] [1:1000001:1] LARGE ICMP PACKET [**] [Classification: (null)] [Priority: 3] {IC
MP} 192.168.3.3:0 -> 192.168.3.5:0
06/29/2025-13:20:08.923358 [wDrop] [**] [1:1000001:1] LARGE ICMP PACKET [**] [Classification: (null)] [Priority: 3] {IC
MP} 192.168.3.5:8 -> 192.168.3.3:0
06/29/2025-13:20:08.923428 [wDrop] [**] [1:1000001:1] LARGE ICMP PACKET [**] [Classification: (null)] [Priority: 3] {IC
MP} 192.168.3.3:0 -> 192.168.3.5:0
06/29/2025-13:20:08.961334 [wDrop] [**] [1:1000001:1] LARGE ICMP PACKET [**] [Classification: (null)] [Priority: 3] {IC
MP} 192.168.3.5:8 -> 192.168.3.3:0
06/29/2025-13:20:08.961389 [wDrop] [**] [1:1000001:1] LARGE ICMP PACKET [**] [Classification: (null)] [Priority: 3] {IC
MP} 192.168.3.3:0 -> 192.168.3.5:0
06/29/2025-13:20:08.993540 [wDrop] [**] [1:1000001:1] LARGE ICMP PACKET [**] [Classification: (null)] [Priority: 3] {IC
MP} 192.168.3.5:8 -> 192.168.3.3:0
06/29/2025-13:20:08.993599 [wDrop] [**] [1:1000001:1] LARGE ICMP PACKET [**] [Classification: (null)] [Priority: 3] {IC
MP} 192.168.3.3:0 -> 192.168.3.5:0

```

Рис. 3.56 - Повідомлення про ICMP-пакети великого розміру

```

~/Desktop/icmp_pass.py - Mousepad
File Edit Search View Document Help
icmp_exfil.py x icmp_pass.py x
1 from scapy.all import *
2
3 # Цільова IP-адреса
4 target_ip = "192.168.3.3"
5
6 # Дані для експільтрації
7 payload = "password:12345"
8
9 # Формування ICMP-пакета з даними
10 packet = IP(dst=target_ip)/ICMP()/payload
11
12 # Відправка пакета
13 send(packet)
14

```

Рис. 3.57 - Скрипт атаки з відправленням пакетів, що містять дані автентифікації

```

(kali@kali)-[~/Desktop]
$ sudo python3 icmp_pass.py
.
Sent 1 packets.

(kali@kali)-[~/Desktop]
$ sudo python3 icmp_pass.py
.
Sent 1 packets.

```

Рис. 3.58 - Ініціація атаки у командному рядку

```
06/29/2025-13:31:37.780489 [wDrop] [**] [1:1000003:1] POSSIBLE PASSWORD EXFIL [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.3.5:8 -> 192.168.3.3:0
06/29/2025-13:31:37.780533 [wDrop] [**] [1:1000003:1] POSSIBLE PASSWORD EXFIL [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.3.3:0 -> 192.168.3.5:0
monitor-server@monitor-server-VirtualBox:/var/log/suricata$
```

Рис.3.59 - Перехоплені пакети, що містять пароль

Blackbox Exporter

[Probe prometheus.io for http 2xx](#)
[Debug probe prometheus.io for http 2xx](#)
[Metrics](#)
[Configuration](#)

Recent Probes

Module	Target	Result	Debug
icmp	localhost	Success	Logs
icmp	localhost	Success	Logs
icmp	localhost	Success	Logs
icmp	localhost	Success	Logs
icmp	localhost	Success	Logs
icmp	localhost	Success	Logs
icmp	localhost	Success	Logs
icmp	localhost	Success	Logs
icmp	localhost	Success	Logs
icmp	localhost	Success	Logs
icmp	localhost	Success	Logs
icmp	localhost	Success	Logs
icmp	localhost	Success	Logs

Рис. 3.60 - Проби у BlackBox

Prometheus Time Series × monitor-server-VirtualBo × 127.0.0.1:9115/logs?id=0 × +

127.0.0.1:9115/logs?id=0

Restore Down

Logs for the probe:
ts=2025-06-29T15:17:20.152239772Z caller=main.go:116 module=icmp target=localhost level=info msg="Beginning probe" probe=icmp timeout_seconds=0.5
ts=2025-06-29T15:17:20.152309743Z caller=utils.go:42 module=icmp target=localhost level=info msg="Resolving target address" preferred_ip_protocol=ip6
ts=2025-06-29T15:17:20.155470451Z caller=utils.go:52 module=icmp target=localhost level=warn msg="Resolution with preferred IP protocol failed, attempting fallback protocol" fallback_protocol=ip4 err="address localhost: no suitable address found"
ts=2025-06-29T15:17:20.155873467Z caller=utils.go:65 module=icmp target=localhost level=info msg="Resolved target address" ip=127.0.0.1
ts=2025-06-29T15:17:20.155892392Z caller=icmp.go:71 module=icmp target=localhost level=info msg="Creating socket"
ts=2025-06-29T15:17:20.157294732Z caller=icmp.go:90 module=icmp target=localhost level=error msg="Error listening to socket" err="listen ip4:icmp 0.0.0.0: socket: operation not permitted"
ts=2025-06-29T15:17:20.15732551Z caller=main.go:129 module=icmp target=localhost level=error msg="Probe failed" duration_seconds=0.005048319

Metrics that would have been returned:
HELP probe_dns_lookup_time_seconds Returns the time taken for probe dns lookup in seconds
TYPE probe_dns_lookup_time_seconds gauge
probe_dns_lookup_time_seconds 0.003542353
HELP probe_duration_seconds Returns how long the probe took to complete in seconds
TYPE probe_duration_seconds gauge
probe_duration_seconds 0.005048319
HELP probe_ip_protocol Specifies whether probe ip protocol is IP4 or IP6
TYPE probe_ip_protocol gauge
probe_ip_protocol 4
HELP probe_success Displays whether or not the probe was a success
TYPE probe_success gauge
probe_success 0

Module configuration:
prober: icmp

Рис. 3.61 - Перегляд логу з пробу ICMP

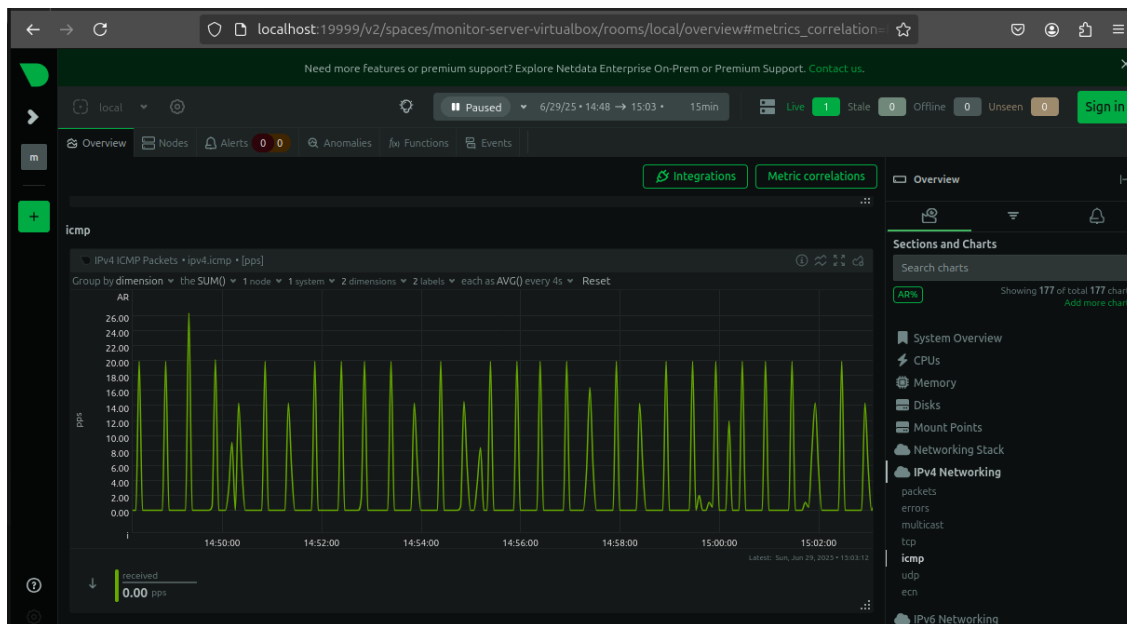


Рис. 3.62 - Дошка ICMP-пакетів у NetData

Високий рівень pps (packets per secont) може бути ознакою атак.

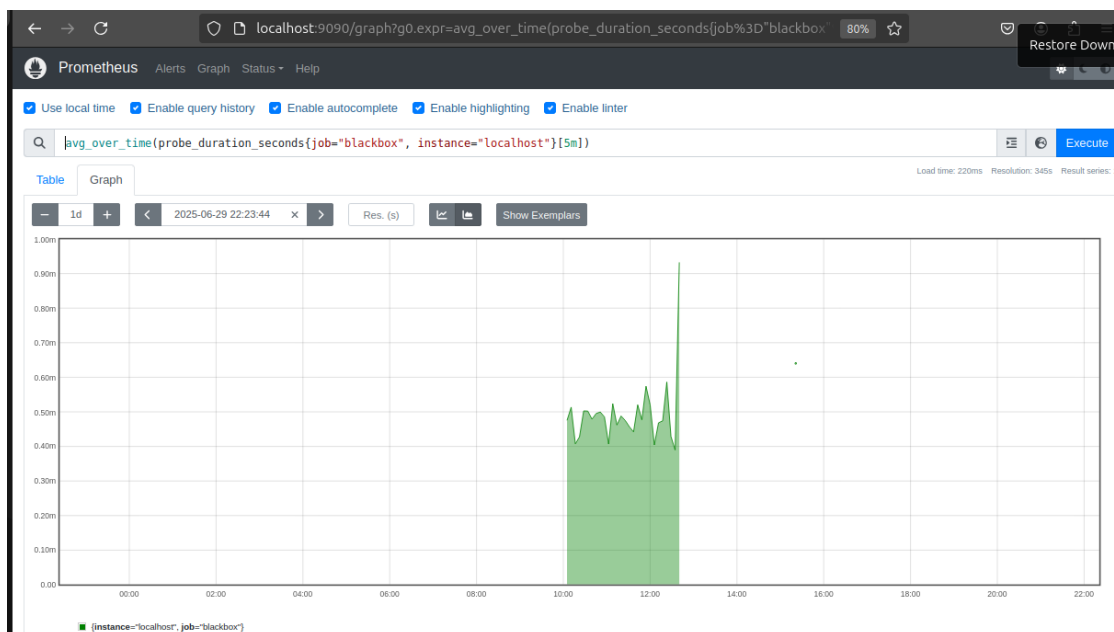


Рис. 3.63 - Статистика створення пробів у Prometheus

Відповідно до статистики NetData, велика кількість одночасно створених пробів може слугувати ознакою атаки. На рисунку найвища точка позначає момент, коли було здійснено атаку із надсиланням великої кількості пакетів великого розміру.

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				76
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до розділу

У межах цього розділу було налаштовано пристрої інформаційно-комунікаційної мережі, встановлено і налаштовано систему виявлення та протидії ексфільтрації ICMP-протоколу із використанням Suricata, Prometheus із BlackBox та Net-Data. Роботу системи було перевірено з моделюванням атаки ексфільтрації ICMP-протоколу.

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				77
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було реалізовано підсистему для виявлення та протидії ексфільтрації даних через ISMP-протокол.

Першочергово було досліджено процеси роботи ISMP-протоколу та його трафіку, проаналізовано види ISMP-повідомлень, що можуть бути використані у атаках з використанням ексфільтрації. Було досліджено існуючі системи виявлення вторгнень з відкритим кодом, що дало змогу порівняти їх функціональні можливості. Особливу увагу приділено системі Suricata, яка продемонструвала вищу ефективність у виявленні прихованих загроз у порівнянні з альтернативними рішеннями.

На етапі проектування розроблено топологію мережі, систему IP-адресації та налаштовано сервіси на кінцевих користувачах. Була реалізована симуляція атак з використанням ISMP-протоколу для перевірки ефективності створених правил виявлення. Також створено набір власних правил для Suricata, проведено інтеграцію з Netdata та Prometheus (Blackbox), що дозволило здійснювати моніторинг трафіку в реальному часі.

Під час тестування системи було перевірено її здатність виявляти спроби ексфільтрації, адаптуватися до різних сценаріїв атак та забезпечувати стійку роботу при змінних навантаженнях. Робота підтвердила, що комплексне застосування інструментів Suricata, моніторингових сервісів і правильно спроектованої мережевої архітектури дозволяє ефективно виявляти та протидіяти ISMP-ексфільтрації.

Результати дослідження можуть бути використані як основа для розгортання подібних систем в реальних мережах, а налаштування та конфігураційні файли — адаптовані для практичного застосування та розширення чи покращення в майбутньому.

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		78

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. (September 1981). [INTERNET CONTROL MESSAGE PROTOCOL - DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION](#). Network Working Group. [doi:10.17487/RFC0792](#). STD 5. [RFC 792](#). Internet Standard 5. Updates RFC [760](#), [777](#), IENs 109, 128. Updated by RFC [950](#), [4884](#), [6633](#) and [6918](#).
2. Postel, J. (ed.), "Internet Protocol - DARPA Internet Program Protocol Specification," [RFC 791](#) (<https://datatracker.ietf.org/doc/html/rfc791>), USC/Information Sciences Institute, September 1981 (дата звернення 20.06.2025)
3. Data Exfiltration: A Review of External Attack Vectors and Countermeasures November 2017 Journal of Network and Computer Applications 101(2) DOI:10.1016/j.jnca.2017.10.016 (дата звернення 20.06.2025)
4. ICMP (Internet Control Message Protocol). URL: <https://community.cisco.com/t5/networking-knowledge-base/icmp-internet-control-message-protocol/ta-p/3116636> (дата звернення 20.06.2025)
5. ICMP Router Discovery Messages. URL: <https://datatracker.ietf.org/doc/html/rfc1256> (дата звернення 22.06.2025)
6. Forouzan, Behrouz A. (2007). [Data Communications And Networking](#) (Fourth ed.). Boston: McGraw-Hill. pp. [621](#)–630. [ISBN 978-0-07-296775-3](#). (дата звернення 22.06.2025)
7. Deprecation of ICMP Source Quench Messages. URL: <https://datatracker.ietf.org/doc/html/rfc6633> (дата звернення 22.06.2025)
8. Exfiltration. URL: <https://attack.mitre.org/tactics/TA0010/> (дата звернення 22.06.2025)
9. Exfiltration Over Alternative Protocol. URL: <https://attack.mitre.org/techniques/T1048/> (дата звернення 25.06.2025)
10. Snort Official Page. URL: <https://snort.org/> (дата звернення 25.06.2025)
11. Snort 3 Rule Writing Guide. URL: <https://docs.snort.org/> ((дата звернення 25.06.2025)
12. Suricata Official Page. URL: <https://suricata.io/> (дата звернення 25.06.2025)

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				79
Змн.	Арк.	№ докум.	Підпис	Дата		

13. Suricata Rules Format. URL: <https://suricata.readthedocs.io/en/latest/rules/intro.html> (дата звернення 25.06.2025)
14. Eve JSON Output. URL: <https://suricata.readthedocs.io/en/latest/output/eve/eve-json-output.html> (дата звернення 25.06.2025)
15. Payload Keywords. URL: <https://suricata.readthedocs.io/en/latest/rules/payload-keywords.html>
16. Zeek Official Page. URL: <https://zeek.org/> (дата звернення 25.06.2025)
17. Zeek Introduction to Scripting. URL: <https://docs.zeek.org/en/current/scripting/index.html> (дата звернення 25.06.2025)
18. NetData Official Page. URL: <https://www.netdata.cloud/> (дата звернення 25.06.2025)
19. Prometheus Official Page. URL: <https://prometheus.io/> (дата звернення 25.06.2025)
20. Querying Prometheus. URL: <https://prometheus.io/docs/prometheus/latest/querying/basics/> (дата звернення 25.06.2025)
21. Suricata Quickstart guide. URL: <https://docs.suricata.io/en/latest/quick-start.html> (дата звернення 25.06.2025)
22. Rule Management 9.2. Adding Your Own Rules. URL: <https://docs.suricata.io/en/latest/rule-management/adding-your-own-rules.html> (дата звернення 29.06.2025)
23. Netdata Agent Installation. URL: <https://learn.netdata.cloud/docs/netdata-agent/installation/> (дата звернення 25.06.2025)
24. Using pre-compiled binaries, Prometheus. URL: <https://prometheus.io/docs/prometheus/latest/installation/> (дата звернення 25.06.2025)
25. Usage of Scapy. URL: <https://scapy.readthedocs.io/en/latest/usage.html> (дата звернення 25.06.2025)

ДОДАТКИ

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				81
Змн.	Арк.	№ докум.	Підпис	Дата		

Конфігураційні файли мережевих пристроїв

R1:	!	line aux 0
Current configuration : 1494	interface FastEthernet0/0	exec-timeout 0 0
bytes	ip address 192.168.3.9	privilege level 15
!	255.255.255.252	logging synchronous
upgrade fpd auto	duplex half	stopbits 1
version 12.4	!	line vty 0 4
service timestamps debug	interface Ethernet1/0	login local
datetime msec	no ip address	transport input none
service timestamps log	shutdown	transport output none
datetime msec	duplex half	!
service password-encryption	!	end
!	interface Ethernet1/1	
hostname R1	no ip address	MS1:
!	shutdown	Current configuration : 1949
boot-start-marker	duplex half	bytes
boot-end-marker	!	!
!	interface Ethernet1/2	! Last configuration change
logging message-counter	no ip address	at 15:34:15 UTC Sun Jun 29
syslog	shutdown	2025
enable secret 5	duplex half	!
\$1\$Gmgm\$zeW8oSXtrCFA	!	version 15.2
DU.ulUQD/	interface Ethernet1/3	service timestamps debug
!	no ip address	datetime msec
no aaa new-model	shutdown	service timestamps log
ip source-route	duplex half	datetime msec
no ip icmp rate-limit un-	!	service password-encryption
reachable	interface GigabitEthernet2/0	service compress-config
ip cef	no ip address	!
!	shutdown	hostname MS1
no ip domain lookup	negotiation auto	!
no ipv6 cef	!	boot-start-marker
!	ip forward-protocol nd	boot-end-marker
multilink bundle-name au-	no ip http server	!
thenticated	no ip http secure-server	logging discriminator EX-
!	!	CESS severity drops 6 msg-
username campusadmin1	no cdp log mismatch duplex	body drops EXCESSCOLL
privilege 15 secret 5	!	logging buffered 50000
\$1\$pLn4\$vELpt0.q3sY3XV	control-plane	logging console discrimina-
cNQiQEY1	!	tor EXCESS
username campustech1 privi-	gatekeeper	enable secret 5
lege 5 secret 5	shutdown	\$1\$9Awf\$1TOiQwP1e4DBg
\$1\$QOZL\$hPDKKngEUCn	!	pWy9zICJ.
QlPJN0FR6/.	line con 0	!
archive	exec-timeout 0 0	username campusadmin1
log config	privilege level 15	privilege 15 secret 5
hidekeys	logging synchronous	\$1\$XmqH\$jGgB3Z.ygG/vrI
!	login local	0xJmflp/
ip tcp synwait-time 5	stopbits 1	

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		82

```

username campustech1 privilege 5 secret 5
$1$aY5$b.m3cLdA2mLa5ubQ2F2Lb0
no aaa new-model
!
no ip icmp rate-limit unreachable
!
no ip domain-lookup
ip cef
no ipv6 cef
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
ip tcp synwait-time 5
!
interface Ethernet0/0
no switchport
ip address 192.168.3.10 255.255.255.252
duplex half
!
interface Ethernet0/1
!
interface Ethernet0/2
!
interface Ethernet0/3
switchport trunk allowed vlan 100
switchport trunk encapsulation dot1q
switchport mode trunk
!
interface Ethernet1/0
!
interface Ethernet1/1
!
interface Ethernet1/2
!
interface Ethernet1/3
!
interface Ethernet2/0
!
interface Ethernet2/1
!
interface Ethernet2/2

```

```

!
interface Ethernet2/3
!
interface Ethernet3/0
!
interface Ethernet3/1
!
interface Ethernet3/2
!
interface Ethernet3/3
!
interface Vlan1
no ip address
shutdown
!
interface Vlan10
ip address 192.168.4.1 255.255.255.0
!
interface Vlan20
ip address 192.168.5.1 255.255.255.0
!
interface Vlan100
ip address 192.168.3.1 255.255.255.248
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
exec-timeout 0 0
privilege level 15
logging synchronous
login local
line aux 0
exec-timeout 0 0
privilege level 15
logging synchronous
line vty 0 4
login local
transport input none
transport output none
!
end

SW1:

```

```

Current configuration : 2192 bytes
!
! Last configuration change at 15:41:00 UTC Sun Jun 29 2025
!
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname SW1
!
boot-start-marker
boot-end-marker
!
logging discriminator EXCESS severity drops 6 msg-body drops EXCESSCOLL
logging buffered 50000
logging console discriminator EXCESS
enable secret 5
$1$Zjsd$vJGZaHeYscuvNsvedCehv0
!
username campusadmin1 privilege 15 secret 5
$1$OSS2$Q45N9IHZ.POg4ng5H4tCb0
username campustech1 privilege 5 secret 5
$1$K3Fc$Iaa3PNS/Xbe9/SGbYik5U1
no aaa new-model
no ip icmp rate-limit unreachable
!
ip cef
no ip domain-lookup
!
no ipv6 cef
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending

```

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				83
Змн.	Арк.	№ докум.	Підпис	Дата		

```

!
ip tcp synwait-time 5
!
interface Ethernet0/0
 switchport
 switchport trunk encapsula-
tion dot1q
 duplex auto
!
interface Ethernet0/1
 switchport
 switchport access vlan 10
 switchport mode access
 duplex auto
!
interface Ethernet0/2
 switchport
 switchport access vlan 10
 switchport mode access
 duplex auto
!
interface Ethernet0/3
 switchport
 switchport access vlan 10
 switchport mode access
 duplex auto
!
interface Ethernet1/0
 no ip address
 duplex auto
!
interface Ethernet1/1
 no ip address
 duplex auto
!
interface Ethernet1/2
 no ip address
 duplex auto
!
interface Ethernet1/3
 no ip address
 duplex auto
!
interface Ethernet2/0
 no ip address
 duplex auto
!
interface Ethernet2/1
 no ip address
 duplex auto
!
interface Ethernet2/2

```

```

no ip address
 duplex auto
!
interface Ethernet2/3
 no ip address
 duplex auto
!
interface Ethernet3/0
 no ip address
 duplex auto
!
interface Ethernet3/1
 no ip address
 duplex auto
!
interface Ethernet3/2
 no ip address
 duplex auto
!
interface Ethernet3/3
 no ip address
 duplex auto
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan10
 ip address 192.168.4.2
 255.255.255.0
!
!
no ip http server
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 login local
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login local
 transport input none
 transport output none
!
end

```

```

SW2:
Current configuration : 2223
bytes
!
! Last configuration change
at 15:37:14 UTC Sun Jun 29
2025
!
version 12.2
service timestamps debug
datetime msec
service timestamps log
datetime msec
service password-encryption
!
hostname SW2
!
boot-start-marker
boot-end-marker
!
!
logging discriminator EX-
CESS severity drops 6 msg-
body drops EXCESSCOLL
logging buffered 50000
logging console discrimina-
tor EXCESS
enable secret 5
$1$2MIJ$B78HDjNKxfugtz
LeLGYqw1
!
username campusadmin1
 privilege 15 secret 5
$1$/5kD$QkWsVQfdAQTu-
baIi/zAKj1
username campustech1 privi-
lege 5 secret 5
$1$.NRR$s415Nd6i155sZf.
FfbWTZ.
no aaa new-model
no ip icmp rate-limit un-
reachable
!
ip cef
no ip domain-lookup
!
no ipv6 cef
!
spanning-tree mode pvst
spanning-tree extend system-
id
!

```

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		84

```

vlan internal allocation pol-
icy ascending
!
ip tcp synwait-time 5
!
interface Ethernet0/0
switchport
switchport trunk encapsula-
tion dot1q
duplex auto
!
interface Ethernet0/1
switchport
switchport access vlan 20
switchport mode access
duplex auto
!
interface Ethernet0/2
switchport
switchport access vlan 20
switchport mode access
duplex auto
!
interface Ethernet0/3
switchport
switchport access vlan 20
switchport mode access
duplex auto
!
interface Ethernet1/0
no ip address
duplex auto
!
interface Ethernet1/1
no ip address
duplex auto
!
interface Ethernet1/2
no ip address
duplex auto
!
interface Ethernet1/3
no ip address
duplex auto
!
interface Ethernet2/0
no ip address
duplex auto
!
interface Ethernet2/1
no ip address
duplex auto

```

```

!
interface Ethernet2/2
no ip address
duplex auto
!
interface Ethernet2/3
no ip address
duplex auto
!
interface Ethernet3/0
no ip address
duplex auto
!
interface Ethernet3/1
no ip address
duplex auto
!
interface Ethernet3/2
no ip address
duplex auto
!
interface Ethernet3/3
no ip address
duplex auto
!
interface Vlan1
no ip address
shutdown
!
interface Vlan20
ip address 192.168.5.2
255.255.255.0
!
ip default-gateway
192.168.5.1
!
no ip http server
!
control-plane
!
line con 0
exec-timeout 0 0
privilege level 15
logging synchronous
login local
line aux 0
exec-timeout 0 0
privilege level 15
logging synchronous
line vty 0 4
login local
transport input none

```

```

transport output none
!
end

SW3:
Current configuration : 1921
bytes
!
! Last configuration change
at 15:31:37 UTC Sun Jun 29
2025
!
version 15.2
service timestamps debug
datetime msec
service timestamps log
datetime msec
service password-encryption
service compress-config
!
hostname SW3
boot-start-marker
boot-end-marker
!
!
logging discriminator EX-
CESS severity drops 6 msg-
body drops EXCESSCOLL
logging buffered 50000
logging console discrimina-
tor EXCESS
enable secret 5
$1$3MeJ$YCE8f00bjKdST
TAnjPqIb.
!
username campusadmin1
privilege 15 secret 5
$1$XaLE$rUAouXWz79sv.j
kZ0uyoz1
username campustech1 privi-
lege 5 secret 5
$1$K2N6$Ape.dkylxuAKi5n
mO6IFrQ/
no aaa new-model
!
no ip icmp rate-limit un-
reachable
!
no ip domain-lookup
ip domain-name campussh-
ark
ip cef

```

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		85

```

no ipv6 cef
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
ip tcp synwait-time 5
!
interface Ethernet0/0
  switchport trunk encapsulation dot1q
  switchport mode trunk
!
interface Ethernet0/1
  switchport access vlan 100
  switchport mode access
!
interface Ethernet0/2
  switchport access vlan 100
  switchport mode access
!
interface Ethernet0/3
  switchport access vlan 100
!
interface Ethernet1/0
!
interface Ethernet1/1
!
interface Ethernet1/2
!
interface Ethernet1/3
!
interface Ethernet2/0
!
interface Ethernet2/1
!
interface Ethernet2/2
!
interface Ethernet2/3
!
interface Ethernet3/0
!
interface Ethernet3/1
!
interface Ethernet3/2
!
interface Ethernet3/3
!
interface Vlan1
no ip address
shutdown
!
interface Vlan100
  ip address 192.168.3.2 255.255.255.248
!
ip default-gateway 192.168.3.1
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
  exec-timeout 0 0
  privilege level 15
  logging synchronous
  login local
line aux 0
  exec-timeout 0 0
  privilege level 15
  logging synchronous
line vty 0 4
  login local
  transport input none
  transport output none
!
end

```

		Шарковський Р. О.			ККІКБ.КР.Б – 125 – 21 – ПЗ	Арк.
		Покотило О. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		86