Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка
Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В.о. завідувача кафедри
кібербезпеки та захисту
інформації
_____ Іван ПАРХОМЕНКО
«___» червня 2025 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)
спеціальність 125 Кібербезпека
(код і назва спеціальності)
освітній ступень бакалавр
освітня програма Кібербезпека
(назва освітньо-професійної програми)
на тему: «Розробка Honeypot-системи для виявлення та аналізу кіберзагроз для підприємств»

Виконавець: студент IV курсу, групи КБ-41

_____ **Дяків Владислав**
(підпис) (ім'я, прізвище)

	Підпис	**Ім'я, прізвище**
Керівник		Олександр ЛАПТЄВ

Нормоконтроль		Сергій ДАКОВ

Київ 2025

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«29» листопада 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

освітньої програми Кібербезпека
(назва освітньо-професійної програми)

Студенту **КБ-41** (група) **Дяківа Владислава Віталійовича** (прізвище ім'я по батькові)

Тема кваліфікаційної роботи Розробка Honeypot-системи для виявлення та аналізу кіберзагроз для підприємств

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Тема кваліфікаційної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №6 від 28.11.2024 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Дані про типові атаки, інструменти розгортання honeypot

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Необхідно Ознайомитися з теоретичними основами функціоналу Honeypot-систем, їх класифікацією та принципами роботи. Розглянути типові загрози для інформаційної інфраструктури підприємств,зокрема методи сканування, атак на сервіси та збирання облікових даних. Обрати відповідний тип Honeypot відповідно до цілей системи.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практична цінність Розроблена honeypot-система та рекомендації з її впровадження для виявлення кіберзагроз.

5. ДАТА ВИДАЧІ ЗАВДАННЯ

Дата видачі завдання: 29 листопада 2024 року

Завдання видав _____ Олександр Лаптєв
(підпис) (ім'я, прізвище)

Завдання прийняв до виконання _____ Дяків Владислав
(підпис) (ім'я, прізвище)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
1	Уточнення постановки задачі	29.11.2024 – 22.01.2025	*виконано*
2	Аналіз літератури	29.01.2025 – 11.02.2025	*виконано*
3	Обґрунтування вибору рішення	12.02.2025 – 15.02.2025	*виконано*
4	Аналіз принципів побудови Honeypot-систем	16.02.2025 – 04.03.2025	*виконано*
5	Оцінка загроз інформаційній інфраструктурі підприємств	05.03.2025 – 21.03.2025	*виконано*
6	Дослідження атак та вразливостей. виявлення Honeypot-системами	22.03.2025 – 08.04.2025	*виконано*
7	Формування рекомендацій щодо впровадження Honeypot у мережі підприємства	09.04.2025 – 10.05.2025	*виконано*
8	Оформлення пояснювальної записки	11.05.2025 – 27.05.2025	*виконано*
9	Підготовка до захисту кваліфікаційної роботи	28.05.2025 – 13.06.2025	*виконано*

Завдання видав _____ Олександр Лаптєв
(підпис) (ім'я, прізвище)

Завдання прийняв до виконання _____ Владислав Дяків
(підпис) (ім'я, прізвище)

Термін подання кваліфікаційної роботи до ЕК 13 червня 2025 року

УДК 004.056.5:004.056.53

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел, додатку, має 62 сторінки основного тексту, 3 таблиці та 27 рисунків. Список використаних джерел містить 16 найменувань і займає 2 сторінки.

Метою роботи є Розробка Honeypot-системи для виявлення та аналізу кіберзагроз у корпоративному середовищі, яка дозволяє виявляти несанкціоновану активність,імітувати вразливі сервіси,а також збирати поведінкові характеристики нападника без ризику для основної системи

Для досягнення зазначеної мети поставлено наступні завдання:

- Дослідити сучасні кіберзагрози, актуальні для підприємств, та визначити недоліки традиційних засобів виявлення атак.
- Проаналізувати принципи функціонування Honeypot-системи, їх класифікацію та архітектурні моделі.
- Визначити вимоги до безпечного розгортання Honeypot-систем у корпоративній мережі.
- Реалізувати експериментальний стенд Honeypot-системи з використанням віртуального середовища

Оцінити ефективність запропонованого рішення для виявлення різних типів атак та сформувати рекомендації щодо його впровадження

Об'єктом дослідження Є процес виявлення та аналізу кіберзагроз в інформаційній інфраструктурі підприємств.

Предметом дослідження Є методи і засоби побудови Honeypot-систем для спостереження за шкідливою активністю, збору технічної інформації про атаки та підвищення рівня обізнаності щодо поведінки зловмисників.

Практична цінність отриманих результатів Полягає у створенні Honeypot-системи яка дозволяє ефективно виявляти кіберзагрози з мінімальним

ризиком для основної інфраструктури, а також забезпечує аналітичну базу для подальшого вдосконалення захисних механізмів підприємства.

Ключові слова: honeypot, кіберзагрози, інформаційна безпека, пастка, виявлення атак, поведінковий аналіз.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

IDS — Система виявлення вторгнень

IPS — Система запобігання вторгнення

CERT — Команда реагування на комп’ютерні надзвичайні події

SSH — Захищений протокол для віддаленого адміністрування

FTP — Протокол передавання файлів

DMZ — Демілітаризована зона між внутрішньою та зовнішньою мережею

ОС — Операційна система

SIEM — Система управління подіями та інформацією безпеки

JSON — Формат обміну структурованими даними

IP — Інтернет-протокол

TCP — Протокол керування передачею

VLAN — Віртуальна локальна мережа

ПЗ — Програмне забезпечення

SOAR — Система автоматизації реагування на інциденти

ІБ — Інформаційна безпека

ВСТУП

Актуальність роботи полягає в тому, що кількість та якість загроз інформаційної безпеки зростає щодня, а технічні можливості зловмисників стають дедалі ефективнішими, автоматизованими та різноманітними. В умовах постійної цифровізації бізнес-процесів, державних систем та критичної інфраструктури підприємства опиняються під тиском складних кібератак, які не завжди можливо виявити за допомогою традиційних засобів захисту.

У цьому контексті особливої актуальності набуває впровадження honeypot-систем як ефективного інструменту для виявлення та аналізу несанкціонованої активності у мережі. Honeypot створює віртуальне середовище, яке імітує вразливі сервіси, приваблюючи зловмисників і дозволяючи фіксувати їхню поведінку без ризику для реальних ресурсів підприємства.

Такий підхід дає змогу не лише виявити факт несанкціонованого доступу, а й отримати детальну інформацію про методи, техніки та інструменти, що використовуються під час атак. Зібрані дані сприяють вдосконаленню політик безпеки, налаштуванню систем виявлення загроз, а також формуванню ефективних заходів реагування на інциденти.

Завдяки роботі в ізольованому середовищі honeypot-системи не створюють ризику для продуктивної інфраструктури навіть у разі активної взаємодії зловмисника. Це робить їх не лише ефективним інструментом безпеки, а й цінним засобом для дослідження поведінки атакуючої сторони у контрольованих умовах.

Метою роботи є розробка honeypot-системи для виявлення та аналізу кіберзагроз у корпоративному середовищі, яка дозволяє виявляти несанкціоновану активність, імітувати вразливі сервіси, а також збирати поведінкові характеристики нападника без ризику для основної системи.

Об'єктом дослідження є процес виявлення та аналізу кіберзагроз в інформаційній інфраструктурі підприємств.

Предметом дослідження є методи і засоби побудови honeypot-систем для спостереження за шкідливою активністю, збору технічної інформації про атаки та підвищення рівня обізнаності щодо поведінки зловмисників.

Для досягнення зазначеної мети поставлено такі завдання:

- дослідити сучасні кіберзагрози, актуальні для підприємств, та визначити недоліки традиційних засобів виявлення атак;
- проаналізувати принципи функціонування honeypot-систем, їх класифікацію та архітектурні моделі;
- визначити вимоги до безпечного розгортання honeypot-систем у корпоративній мережі;
- реалізувати експериментальний стенд honeypot-системи з використанням віртуального середовища;

Практична цінність отриманих результатів полягає у створенні honeypot-системи, яка дозволяє ефективно виявляти кіберзагрози з мінімальним ризиком для основної інфраструктури, а також забезпечує аналітичну базу для подальшого вдосконалення захисних механізмів підприємства.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ЗАСАДИ ЗАСТОСУВАННЯ HONEYPOT-СИСТЕМИ ДЛЯ ВИЯВЛЕННЯ ТА АНАЛІЗУ КІБЕРЗАГРОЗ В ІНФОРМАЦІЙНІЙ ІНФРАСТРУКТУРІ ПІДПРИЄМСТВА

1.1 Теоретичні аспекти застосування Honeypot-систем в інформаційній інфраструктурі підприємства

Визначення, поняття Honeypot-систем.Honeypot (англ. Горщик з медом) - це інформаційна система або її окремий компонент, спеціально створений для імітації вразливого чи привабливого для зловмисника ресурсу з метою виявлення, вивчення та аналізу від несанкціонованих дій у кіберпросторі. Головною особливістю Honeypot - систем є те, що вони не мають продуктивної цінності для реальних користувачів: будь-яка взаємодія з ними є ознакою підозрілої або ворожої активності.

Основна ідея технології полягає у створенні так званої “пастки” - сервісу або систему, що зовні виглядає як цінний об’єкт атаки, але не виконує жодних критичних бізнес-функцій. Завдяки цьому honeypot-системи дозволяють фіксувати дії зловмисників, аналізувати техніки вторгнення, збирати інформацію про підозрілу активність, перш ніж вона встигне вплинути на реальні ресурси.

По своїй суті Honeypot є пасивним елементом захисту: він не запобігає атакам, а навпаки, дозволяє їм відбутись в контрольованому середовищі. Таким чином, на відміну від міжмережевих екранів або систем виявлення вторгнення, Honeypot не блокує трафік, а створює умови для спостереження за зловмисником у реальному часі.

У галузі кібербезпеки також часто використовують поняття honeynet говорячи про більш комплексні рішення, які поєднують кілька honeypot-ресурсів у єдину логічну мережу, імітуючи цілісну IT-інфраструктуру. [1]

Таким чином, Honeypot-системи відіграють унікальну роль в інформаційній безпеці, поєднуючи елементи обману, спостереження й аналітики, що робить їх цінним інструментом для передчасного виявлення кіберзагроз та побудови ефективних механізмів реагування.

Мета та основні функції Honeypot-систем.Головна мета використання Honeypot-систем полягає у виявленні та аналізі несанкціонованих дій у мережі підприємства, а також у зборі інформації про методи, інструменти й мотиви зловмисників. Такі системи не призначені для роботи з реальними користувачами, а створюються навмисно, щоб виглядати вразливими і привертати увагу потенційних зловмисників.

До ключових функцій Honeypot-систем належать:

- Моніторинг шкідливої активності: Будь-яка взаємодія з honeypot є потенційно ворожою, що дозволяє точно ідентифікувати підозрілий трафік і поведінку.
- Вивчення технік атак: Honeypot-середовище дозволяє безпечно аналізувати дії зловмисника, отримуючи цінну інформацію для покращення реального захисту.
- Виявлення нових загроз.Завдяки своїй спостережній функції honeypot здатен фіксувати невідомі або модифіковані шкідливі дії, які можуть бути непомітними для традиційних засобів захисту.
- Передчасне попередження: У корпоративних мережах honeypot часто виконує роль системи сповіщення, що сигналізує про спробу проникнення ще до моменту досягнення критичних активів.
- Відволікання уваги нападника: Вразливий вигляд honeypot може змусити зловмисника витратити ресурси на фальшиву цілу, зменшуючи ризик пошкодження реальних систем.
- У результаті застосування honeypot дає змогу організації не лише захищатися, а й ефективно набувати знання у протидії кіберзагрозам, що є критично важливо у сучасному середовищі кібер ризиків.

Принципи роботи Honeypot-систем.Основні принципи роботи honeypot-систем ґрунтуються на залученні потенційних зловмисників шляхом імітації вразливих ресурсів. Це дозволяє не лише виявляти атаки, а й детально аналізувати їхню природу та методи, забезпечуючи захист основної інфраструктури.

Основними принципи роботи таких систем є:

- Імітація вразливостей - Honeypot створює навмисно доступні для атаки точки з відомими слабкими місцями, провокують зловмисників на взаємодію.
- Моніторинг та збір даних - Уся активність на Honeypot ретельно відслідковується для отримання інформації про типи атак, їхню поведінку та інструменти.
- Ізоляція від основної мережі - Система працює окремо від реальної інфраструктури, що мінімізує ризики поширення загроз.

Таким чином, Honeypot-системи виконують роль як пастки, так і аналітичного інструменту, підвищуючи загальний рівень кібербезпеки організації.[2]

Види Honeypot за рівнем взаємодії.Honeypot-системи класифікуються за рівнем взаємодії, який вони забезпечують з нападником.

Виділяють три такі основні типи:

Таблиця 1.1

Порівняння типів Honeypot-систем за рівнем взаємодії

Рівень взаємодії	**Характеристика**	**Переваги**	**Недоліки**	**Приклади застосування**
Низький	Імітація базових сервісів або портів, без повної ОС	Простота у використанні, низьке навантаження	Мінімальний обсяг зібраних даних	Попереднє виявлення сканування мережі

Рівень взаємодії	Характеристика	Переваги	Недоліки	Приклади застосування
Середній	Часткова емуляція ОС і сервісів	Краще розуміння тактик атакувальників	Більші ресурси на підтримку	Виявлення шкідливих дій в локальній мережі
Високий	Повноцінне середовище з можливістю реальної взаємодії	Максимальний обсяг зібраної інформації	Високий ризик компрометації, потребує ізоляції	Дослідження складних атак і поведінки зловмисників

● Honeypot з низьким рівнем взаємодії - це прості системи, які імітують базові сервіси або відкриті порти. Вони не підтримують реальної операційної системи і мають обмежену функціональність, тому мінімально впливають на продуктивність мережі, але збирають обмежений обсяг інформації про атаки.

● Honeypot з середнім рівнем взаємодії - надають більш реалістичне середовище, що частково імітує операційну систему або сервіси. Вони дозволяють спостерігати більш складну поведінку атакувальника, але потребують більшого ресурсу для підтримки.

● Honeypot з високим рівнем взаємодії - це повноцінні системи, що дозволяють зловмиснику взаємодіяти з реальними або віртуальним середовищем. Вони збирають максимальну кількість даних про методи атак і

поведінку хакерів, але потребують найсуворішої ізоляції та контролю через потенційний ризик компрометації.

Правильність вибору певного виду Honeypot залежить в першу чергу від цілей організації, наявних ресурсів і рівня допустимого ризику. [1]

Переваги застосування Honeypot-систем у корпоративному середовищі. Впровадження Honeypot-систем у корпоративній мережі має низку суттєвих переваг:

- Раннє виявлення загроз - Honeypot може виявити нові і нестандартні атаки,які традиційні системи безпеки можуть пропустити.
- Збір детальної інформації - системи збирають глибокі дані про поведінку атакувальників, що допомагає вдосконалювати стратегії захисту.
- Відволікання зловмисників - Honeypot відволікає увагу хакерів від реальних цілей, знижуючи ризик компрометації основних ресурсів.
- Оцінка ефективності захисних заходів - аналіз атак на Honeypot допомагає тестувати і покращувати існуючі засоби безпеки.
- Низькі експлуатаційні витрати - порівняно з іншими засобами безпеки, Honeypot відносно недорогі у розгортанні і підтримці.
- Завдяки цим перевагам Honeypot - системи є важливим компонентом багаторівневої стратегії захисту корпоративної інфраструктури.

1.2 Основні поняття та принципи функціонування Honeypot-систем

Компоненти Honeypot-систем. Сучасні Honeypot-системи є багатокомпонентними рішеннями, кожен елемент яких виконує чітко визначену функцію в межах виявлення та аналізу кіберзагроз. Основними компонентами такої системи є:

- Пастка (Decoy) - головний елемент Honeypot, який симулює реальні сервіси, операційні системи або вразливі програми. Цей компонент має бути достатньо переконливим, щоб зацікавити зловмисника і викликати взаємодію.

Залежно від рівня деталізації, пастка може реалізовуватись як емульована система або повноцінна віртуальна машина.

- Засоби моніторингу (Monitoring Tools) - забезпечують реєстрацію всіх дій, що відбуваються в системі Honeypot. Це можуть бути утиліти для фіксації мережевих з'єднань, запуску процесів, змін у файловій системі тощо. Наприклад, у системах з високим рівнем взаємодії застосовуються інструменти на кшталт Sebek, які дають змогу непомітно контролювати дії зловмисника.
- Аналітичний модуль (Analysis Engine) - відповідає за обробку та інтерпретацію зібраних даних. Цей компонент може бути інтегрований із системами виявлення вторгнень (IDS), SIEM-платформами або працювати автономно, виконуючи класифікацію та кореляцію подій.
- Інтерфейс управління (Control Interface) - дозволяє адміністраторам системи керувати параметрами роботи Honeypot, встановлювати сценарії поведінки, а також переглядати результати аналізу.
- Злагоджена взаємодія цих компонентів забезпечує ефективне функціонування Honeypot-системи, її стійкість до виявлення та здатність до збору цінової інформації про атаки. [1]

Механізми симуляції реального середовища. Ключовим елементом успішного впровадження Honeypot-системи є здатність правдоподібно імітувати реальне середовище, у якому функціонує справжня інформаційна інфраструктура. Ця симуляція може здійснюватися на кількох рівнях:

- Мережевий рівень - Honeypot може симулювати відкриті порти, служби, конфігурацію мережі, а також типову взаємодію з іншими вузлами, що робить його схожим на звичайний робочий сервер.
- Прикладний рівень - використовується емуляція програмного забезпечення, яке зазвичай є цілями атак, наприклад, веб-додатків, баз даних або CRM-систем. У деяких випадках застосовують реальні програми зі зміненими конфігураціями безпеки, щоб зробити їх привабливими для атак.
- Повноцінне віртуалізоване середовище - у Honeypot з високим рівнем взаємодії створюється віртуальна або фізична копія системи, де зловмисник

отримує можливість виконувати дії, як у справжньому середовищі. Це дозволяє зафіксувати навіть складні сценарії атак, включаючи переміщення по мережі.

- Правдоподібні симуляція суттєво знижує шанси виявлення Honeypot як фальшивого об’єкта, підвищуючи його ефективність як інструменту обману. [3]

Принципи збору, обробки та аналізу даних про вторгнення .Принципи збору обробки та аналізу даних про вторгнення ефективне функціонування honeypot-системи визначається не лише якістю симуляції, а й здатністю фіксувати та інтерпретувати вторгнення. Збір, обробка та аналіз даних здійснюється за кількома ключовими етапами:

- Реєстрація подій - уся активність пов'язана з підозрілою взаємодією з honeypot фіксується системами логування від мережевих. Запитів до команд введених зловмисником, часто використовуються інструменти на кшталт: Wireshark, TCPdump або спеціалізовані системи аудиту.

- Узагальнення та фільтрація: Сирі дані структуризуються фільтруються від шуму та класифікуються за типами Атак (наприклад сканування експлуатація вразливості зловмисне програмне забезпечення).

- Поглиблений аналіз здійснюється із застосуванням механізмів поведінкового аналізу машинного навчання або порівняння з існуючими сигнатурами загроз. на цьому етапі можливо виявити нові або модифіковані техніки атаки які не ідентифікуються класичними IDS.

- Експорт до зовнішніх систем - результати аналізу можуть автоматично передаватись до SIEM-систем або систем реагування на інциденти (SOAR) для подальшого реагування. Завдяки цим процесам ханіпот трансформується з пасивного засобу спостереження потужний інструмент кіберрозвідки.

Взаємозв’язок Honeypot з іншими засобами захисту інформації.Системи не функціонують ізольовано вони є частиною багаторівневої архітектури кіберзахисту й ефективної взаємодії з іншими елементами інформаційної безпеки:

- З системами виявлення вторгнень (IDS) - дані зібрані honeypot можуть використовуватись для створення нових сигнатур атак що дозволяє IDS виявити аналогічні загрози в реальному середовищі.
- З SIEM-платформами - honeypot-системи виступає джерелом якісної інформації для аналітики подій безпеки зібрані події можуть підвищити точність кореляції інцидентів і прискорити їх виявлення.

Засобами управління властивостями - дані ханіпот можуть стати основою для запуску автоматизованих сценаріїв реагування: блокування ip-адрес оновлення політик безпеки інформування аналітиків.

- Таким чином ханіпод системи виступають не конкурентами, а доповненням до існуючих систем захисту забезпечуючи раннє попередження про загрози зменшення хибних спрацювань та покращення стратегій кібербезпеки загалом. [5]

1.3 Класифікація Honeypot-систем

Ханіпод системи поділяються на кілька типів залежно від їхнього призначення рівні взаємодії за такуючим способу розгортання та технологічної реалізації така класифікація дозволяє краще розуміти сферу застосування кожного типу систем і обирати найбільш відповідний інструмент для конкретної інформаційної структури.

Поділ за рівнем взаємодії.Використання цього критерія є одним з найважливіших оскільки визначає глибину взаємодії між ханіпод системою та зловмисником.

Honeypot з низьким рівнем взаємодії (low- interaction Honeypot) - Імітують базову поведінку служб або протоколів без повної реалізації функціоналу їхня головна перевага висока безпека простота розгортання проте ці системи виявляють переважно прості атаки такі як автоматизоване сканування чи спроби використання відомих вразливостей прикладом є honeypot.

Honeypot із середнім рівнем взаємодії - надають обмежений але реалістичний функціонал для взаємодії зловмисника вони можуть імітувати поведінку операційної системи або окремих сервісів більш глибокого ніж системи з низькою взаємодією але не дозволяють повного контролю над системою.

Honeypot з високим рівнем взаємодії - створюють повноцінне середовище з реальними операційними системами та сервісами що дозволяють атакуючому вільно діяти. Цей тип honeypot найбільш інформативний оскільки дає змогу досліджувати складні техніки атак і зловмисну поведінку однак такі системи потребують ретельної ізоляції та контролю щоб уникнути ризику компрометації реального середовища. Наприклад система Cowrie або віртуалізоване середовище.

Класифікація за функціональним призначенням.Honeypot системи також класифікуються залежно від завдань які вони виконують у системі безпеки.

Дослідницькі (research Honeypots) - Використовується для глибокого аналізу нових типів атак вивчення тактик і технік і процедур (TTP) зловмисників. Такі Системи часто застосовуються в наукових дослідженнях, CERF-організаціях і центрах кіберзахисту для вивчення поведінки ботнетів.

Виробнича або промислові - Впроваджується в корпоративному середовищі з метою виявлення та стримування реальних атак вони інтегруються з іншими компонентами безпеки і служить додатковим шаром захисту зокрема в критичній інфраструктурі наприклад системи Modern Honey Network дає змогу централізовано розгортати й управляти великою кількістю Honeypot.

За типом цілей та сценаріїв використання.Honeypot системи класифікуються і за типом імітованих об'єктів або рівнем моделювання:

Мережеві Honeypot - імітують мережеву службу відкриті порти або пристрої націлені на виявлення сканування атак на протоколи або спроби проникнення ззовні.

Хостові Honeypot імітуюча конкретні хости або сервери з повним програмним стеком. Такі Honeypot можуть імітувати вразливі веб-додатки бази даних чи файлові сервіси.

Клієнтські Honeypot розроблені для активного виявлення шкідливих серверів вони самі ініціюють з'єднання з підозрілими ресурсами наприклад сайтами які можуть розповсюджувати шкідливе ПЗ.

Сервісні фокусуються на конкретних службах таких як SSH, FTP або SMTP. Вони часто використовуються для моніторингу спроб несанкціонованого доступу або збору облікових даних.

Порівняльна характеристика типів. Існують також класифікації за способом реалізації honeypot у мережевій інфраструктурі:

Фізичні honeypot - реалізуються на реальних фізичних машинах що забезпечує найвищу ступінь реалістичності але значно ускладнює масштабування і управління.

Віртуалізовані honeypot - створюються на основі віртуальних машин або контейнерів що забезпечує більшу гнучкість або можливість централізованого керування.

Хмарні honeypot розгортаються в середовищах публічних або приватних хмар і можуть імітувати хмарні сервіси API сховище тощо. Вони особливо актуальні в умовах цифрової трансформації та переходу до хмарних архітектур

1.4 Архітектура побудови Honeypot-систем у корпоративному середовищі

Ефективність honeypot-системи у корпоративному середовищі значною мірою залежить від правильного спроектованої архітектури архітектура повинна враховувати як технічну так і організаційні аспекти включаючи безпеку масштабованість можливість інтеграції з іншими компонентами системи кіберзахисту та мінімізацією ризику компрометації.

Структурні елементи архітектури Honeypot. Архітектура honeypot системи побудована на взаємодії кількох логічних відокремлених компонентів які забезпечують коректне функціонування безпечної ізоляцію та ефективний збір інформації про атаки. До основних елементів архітектури належать:

Емуляційне середовище - це основна складова яка створює віртуальне або фізичне середовище що приваблює зловмисників у ньому можуть бути розгорнуті фальшиві служби хвостові системи або мережеві топології що виглядають привабливими для Атак.

Логер фіксації подій - елемент відповідальний за непомітне для зловмисника фіксування всіх дій у системі Це може бути як мережевий сніфер так і внутрішній модуль що перехоплює системні виклики файли мережеві пакети команди тощо. Надзвичайно важливо що блогер був ізольований від декоя та стійкий до виявлення

Транспортний механізм передачі даних - канал що забезпечує безпечне транспортування журналів подій до зовнішнього сховища або централізованої системи аналізу. Зазвичай передбачає шифрування даних а також використання окремої захищеної мережі або VPN.

Аналітичний сервер - незалежний компонент що виконує попередню Обробку отриманих даних Фільтрацію фальшивих або не цікавих подій агрегацію логів та формування звітів.

Керуючий інтерфейс адміністративний модуль що дозволяє керувати конфігурацією системи моніторити її стан запускати чи зупиняти окремі вузли у безпечних реалізаціях - це інтерфейс ізольований фізично або віртуальну від декоя.

Завдяки модульній структурі honeypot система може бути адаптована до різних середовищ від невеликих локальних мереж до складних корпоративних інфраструктур з географічно розподіленими вузлами.

Вимоги до безпечного розгортання Honeypot-систем.Щоб honeypot система не становила загрозу для внутрішньої мережі її слід розгортати в ізольованому середовищі важливо обмежити її доступ до зовнішніх ресурсів

контролювати вихідний трафік і забезпечити фільтрацію мережевих з'єднань передавання логів має здійснюватись через захищені канали бажано односторонньому режимі також рекомендується приховати ознаки віртуалізації використовувати механізми автоматичного відновлення стану системи після інциденту та постійно оновлювати фальшиві сервіси для підтримання достовірності середовища.

1.5 Можливості Honeypot-систем у виявленні та аналізі кіберзагроз

Системи honeypot виступають не лише як пастки для зловмисників, а як потужний аналітичний інструмент здатний надати цінну інформацію про реальні загрози з якими стикається інформаційна інфраструктура підприємства на відміну від традиційних засобів захисту дозволяють дослідити поведінку атакувальника у контрольованому середовищі без ризику для основних активів завдяки цьому такі системи відкривають широкі можливості для виявлення класифікації глибокого аналізу кіберзагроз для різних рівнів - від простих автоматизованих сканерів до складних цілеспрямованих атак.

Виявлення шкідливих активностей і поведінкових шаблонів. Однією з найважливіших функцій фаніподсистеми є здатність фіксувати шкідливу активність у реальному часі завдяки тому що honeypot не повинен виконувати жодних легітимних функцій будь-який запит до нього вважається потенційно зловмисним. Це суттєво знижує рівень Активності яка зазвичай ускладнює аналіз подій у традиційних системах моніторингу і дозволяє зосередитись саме на реальних інцидентах honeypot-системи здатні виявити різноманітні типи активностей: тропи сканування портів підбір паролів експлуатацію вразливостей розгортання зловмисного коду, а також нетипові комунікації які свідчать про застосування автоматизованих Ботів або інструментів пентесту крім того накопичена інформація дозволяє будувати поведінкові шаблони що в подальшому можуть бути використані для автоматизованого виявлення загроз у реальних середовищах.

Завдяки цим можливостям honeypot може служити джерелом даних для побудови поведінкових моделей атак які суттєво доповнюють класичні методи детекції підвищуючи загальну ефективність систем виявлення вторгнення

Аналіз інструментів та методів атак зловмисників.Системи надають унікальну можливість для глибокого дослідження Арсеналу зловмисника У реальних умовах коли атакуючи здійснює взаємодію з ханібот система може детально зафіксувати не лише тип атаки а й конкретні інструменти скрипти мережеві команди завантаження додаткового зловмисного ПЗ а також параметри його запуску Це дає змогу фахівцям з кібербезпеки досліджувати тактики техніки процедури які застосовуються в актуальних атаках

Крім того honeypot може бути спеціально налаштований для імітації вразливих сервісів або програмного забезпечення що провокує зловмисника на розгортання експлойтів це створює умови для аналізу нових або модифікованих варіантів вже відомих шкідливих інструментів що дозволяють оновлюватися на тури засобів виявлення загроз і збагачувати бази знань про кіберзагрози.

Виявлення цілеспрямованих атак (APT) .На відміну від масових атак які часто використовуються автоматизованими ботами атаки типу apt характеризується високим рівнем підготовки тривалістю використанням кастомних інструментів і глибокою розвідкою щодо цілей honeypot-системи можуть бути налаштовані для виявлення. Саме таких загроз зокрема через розгортання високо взаємодіючих пасток у критично важливих сегментах інфраструктури. [7]

А під ці групи можуть спробувати отримати доступ до honeypotвважаючи його легітимною частиною системи особливо якщо вона ретельно емулює бізнес-процеси користувацький облікові записи або внутрішні сервіси.

Роль Honeypot у формуванні стратегії кіберзахисту.Дані зібрані за допомогою ханні подсистем мають значну аналітичну цінність при формуванні та коригуванні стратегій кіберзахисту підприємства завдяки реальній інформації про те які типи атак вектори доступу та техніки найбільш активно використовуються проти конкретних середовища організація може обґрунтовано

розставляти пріоритети щодо захисту активів посилення політик доступу або впровадження нових технологічних засобів крім того honeypot дозволяє верифікувати ефективність існуючих засобів безпеки тестувати поведінку зловмисників у змінених умовах та оперативно реагувати на появу нових загроз.

1.6 Аналіз ризиків та обмежень використання Honeypot-технологій

Попри велику користь у виявленні загроз впровадження honeypot-системи не є універсальним рішенням і супроводжується певними ризиками та обмеженнями їх врахування є критично важливими для безпечного. Не завжди можливо ефективного та легітимного застосування таких технологій у корпоративному середовищі.

Можливі ризики для корпоративної мережі при використанні Honeypot.Одним із головних ризиків є компрометація honeypot-системи подальше використання її зловмисниками як точки опори для атак на інші сегменти мережі особливо це стосується високо взаємодіючих honeypot, які можуть мати розширений функціонал і складні конфігурації недостатня ізоляція або помилки налаштування мережевих правил можуть призвести до того що honeypot стане плацдармом для внутрішнього поширення атак крім того є ризик того що honeypot буде виявлений нападником і замість збору корисної інформації систему почне генерувати хибні сигнали або стани об'єктом цілеспрямованого впливу з метою обману або дискредитації засобів безпеки.

Обмеження щодо ефективності та точності виявлення атак.Honeypot виявляє лише ті загрози які безпосередньо взаємодіють з ним це означає що він не покриває повністю весь спектр атак що можуть відбуватися в інших частинах інфраструктури у разі відсутності належного трафіку або інтересу з боку зловмисників Honeypot може залишатись “невидимим” не надаючи цінної інформації. також варто врахувати потенційне обмеження з неоднозначністю зібраних даних Не завжди можливо точно визначити наміри нападника або інтерпретувати його дії без контексту.

Юридичні та етиччні аспекти використання Honeypot.Розгортання ханіпод систем у відкритих або гібридних мережах може стикатися з юридичними обмеженнями пов'язаними з фіксацією персональних даних або перехопленням трафіку без згоди користувача.

Шляхи мінімізації ризиків та підвищення ефективності Honeypot.Для мінімізації технічних ризиків Honeypot має бути повністю ізольований від продуктивної мережі чітко налаштованими правилами доступу та журналюванням усіх дій рекомендується використовувати віртуалізацію або контейнеризацію а також моніторингові інструменти для контролю за станом у режимі реального часу.

Висновок до розділу 1

У даному розділі було розглянуто важливі аспекти пов'язані з кіберзагрозами для підприємств та можливостями їх виявлення й аналізу за допомогою honeypot-системи. Здійснено огляд сучасного стану інформаційної безпеки підкреслено актуальність проблеми недостатнього виявлення складних атак і потребу в додаткових механізмах реагування.

Було розглянуто основні поняття принципи функціонування та класифікації ханіпот систем. Встановлено що ханіпод системи є ефективним інструментом у виявленні несанкціонованої активності та дослідження поведінки зловмисників. Залежно від рівня інтерактивності ці системи можуть варіюватись від простих пасток до повноцінних середовищмуляцій що дають змогу отримати глибоке уявлення про методи атак.

Окрему увагу було приділено архітектурним особливостям honeypot рішень Як автономним так і інтегрованим у ширші системи безпеки визначено що правильна побудова архітектури ханіпот-систем дозволяє мінімізувати ризики для основної інфраструктури та водночас значно розширити можливості спостереження за кіберзагрозами.

Honeypot-системи, як показано у цьому розділі є важливими в елементом сучасного підходу до кіберзахисту. Вони забезпечують не лише виявлення атак а й збір цінної інформації про способи проникнення та цілі зловмисників. Це дозволяє фахівцям з кібербезпеки приймати обґрунтовані рішення щодо удосконалення захистих заходів та підвищення загальної кібер стійкості підприємства.

Таким чином honeypot-системи виступають не лише як засіб моніторингу загроз а й як активний інструмент для вивчення кібер злочинної активності що робить їх перспективним компонентом комплексної системи захисту інформаційної інфраструктури.

РОЗДІЛ 2
АНАЛІЗ ПРОБЛЕМ ВИЯВЛЕННЯ КІБЕРЗАГРОЗ У СУЧАСНИХ ІНФОРМАЦІЙНИХ ІНФРАСТРУКТУРАХ ТА ПІДХІД ДО ЇХ ВИЯВЛЕННЯ ЗАСОБАМИ HONEYPOT-СИСТЕМ

Сучасні інформаційні інфраструктури підприємств стають дедалі складнішими що зумовлено як інтенсивною цифровізацією бізнес-процесів так і зростання в кількості взаємозв'язаних систем пристроїв та хмарних сервісів. У таких умовах забезпечення належного рівня кіберзахисту стає стратегічним завданням, а виявлення кіберзагроз одним з найкритичніших напрямків діяльності фахівців з інформаційної безпеки. Однак традиційні засоби виявлення атак дедалі частіше виявляється неефективними перед новими типами загроз зокрема складними цілеспрямованими, атака нульового дня та засобами з боку інсайдерів.

У цьому контексті виникає необхідність у переосмисленні підходів до виявлення загроз, а також у провадженні нових технологій здатних забезпечити більшу спостережність адаптивність і контекстуальне розуміння поведінки зловмисників. Одним із перспективних напрямів є використання honeypot-системи, які дозволяють не лише виявляти активність порушників, а й отримувати цінну інформацію про їхні методи інструменти та тактики.

2.1. Проблеми традиційних засобів виявлення кіберзагроз

Більшість сучасних організацій застосовують традиційні засоби виявлення кіберзагроз до яких належить системи виявлення вторгнень (IDS) антивірусне програмне забезпечення системи контролю доступу та міжмережеві екрани. незважаючи на те що ці рішення є важливими компонентами захисної інфраструктури у багатьох випадках вони виявляються недостатньо ефективними для протидії сучасним типам атак.

Однією з головних проблем традиційних засобів є орієнтація на сигнатурний аналіз. Більшість IDS або антивірусних систем базується на виявленні вже відомих шаблонів атак. Такий підхід не дозволяє своєчасно виявити нові загрози, зокрема атаки нульового дня, які ще не мають фіксованих ознак. У результаті - критичні події можуть залишатися непоміченими.

Іншою проблемою є велика кількість хибно-позитивних або хибно негативних спрацювань. Надмірна кількість тривог призводить до перевантаження аналітиків, які змушені вручну перевіряти сотні подій багато з яких не становлять реальної загрози. Це, в свою чергу, знижує оперативність реагування на справді небезпечні інциденти.

Традиційні засоби також зазвичай працюють у межах дозволеного трафіку й обмежені доступом до деяких сегментів мережі або обчислювального середовища у таких умовах вони можуть не виявити загрозу яка діє в мережі або маскується під легітимну активність. [6]

Крім того недостатня адаптивність багатьох традиційних систем унеможливлює їхню ефективну роботу в умовах швидкозмінного ландшафту загроз. Відсутність механізмів машинного навчання або динамічного оновлення баз знань часто означає що система застаріває ще до того як її повноцінно розгорнуто.

Усі ці фактори обумовлюють необхідність переходу до більш гнучких та інтерактивних засобів виявлення які не лише виявляють факт атаки а й здатні імітувати вразливе середовище збирати інформацію про поведінку порушника та допомагати розробці ефективних контрзаходів.

Для кращого розуміння honeypot-системи доцільно порівняти їх із традиційними підходами до виявлення загроз. У таблиці 2.1 наведено основні відмінності між цими підходами.

Таблиця 2.1

Порівняння традиційних методів виявлення Ханіпот

Характеристика	**Традиційний підхід (IDS/SIEM)**	**Honeypot-підхід**
Основний метод виявлення	Сигнатури, евристика	Поведінковий аналіз атакувальників
Джерело даних	Увесь реальний трафік	Штучно створене середовище, що приваблює атак
Рівень шуму (false positives)	Високий	Дуже низький
Виявлення 0-day атак	Ускладнене	Можливе завдяки несподіваним діям атакувальника
Реагування	Часто потребує ручної обробки	Можна автоматизувати на основі активності
Вплив на продуктивність систем	Є (моніториться весь трафік)	Відсутній (ізольоване середовище)

2.2. Класифікація джерел загроз у корпоративних системах

У сучасних корпоративних інформаційних системах джерела кіберзагроз мають різноманітну природу. Для ефективного виявлення та протидії загрозам необхідно здійснити їх класифікацію за певними критеріями. Це дозволяє не лише впорядкувати дані про потенційні ризики але й адаптувати засоби захисту

зокрема honeypot-системи під конкретні типи загроз. одним із ключових критеріїв є походження загроз.

Згідно з практикою інформаційної безпеки джерела загроз поділяють на:

- Зовнішні загрози що походять з-за меж корпоративної мережі до них відносяться зломисники з інтернету, хакерські групи, ботнети тощо
- Внутрішні. - загрози що виникають всередині організації. Наприклад помилка адміністраторів неправильна конфігурація системи.

Інший важливий аспект класифікації - намір або мотивація загроз. Вони можуть бути:

Навмисні - Атаки здійснені цілеспрямовано з метою заподіяння шкоди або отримання вигоди як наприклад хакерські атаки.

Не навмисні - випадкові або необережні дії які призводять до порушення безпеки помилка користувача неправильні налаштування.

Також джерела загроз поділяють за рівнем технічної складності:

Технічно прості - Атаки низької складності які можна здійснити без глибоких знань наприклад фішинг або використання слабких паролів.

Середньої складності - атаки щоб потребують певних технічних навичок наприклад експлойти відомих вразливостей.

Складні цілеспрямовані атаки - високотехнологічні спеціалізовані операції використання зіродей вразливостей.

Для підприємств важливо також виділяти внутрішню організаційні джерела що виникають внаслідок:

- Неналежного управління активами.
- Неправильного налаштування систем.
- Недоліків у політиці безпеки.

Застосування класифікації дозволяє:

Формувати більше релевантні правила для систем виявлення загроз.

Підвищити ефективність ханіпот рішень шляхом орієнтації на найбільш актуальні джерела.

Прогнозувати потенційні ризики та вдосконалювати процедури реагування. (рис 2.1)

Рисунок – 2.1 Класифікація джерел загроз у корпоративних системах

2.3. Вимоги до систем виявлення загроз в інфраструктурі підприємств

Системи виявлення загроз є ключовими елементами безпекової інфраструктури сучасного підприємства. Їх завдання полягає не лише у виявленні фактів вторгнення або аномальної активності, а й у забезпеченні своєчасного реагування, коректної інтерпретації інцидентів та інтеграції з

іншими елементами захисту. Щоб така система була ефективною Вона має відповідати низці вимог які можна поділити на функціональні та нефункціональні.

1. Функціональні вимоги:

- Можливість виявлення різних типів загроз включаючи мережеві атаки несанкціонований доступ аномальні дії користувачів тощо.
- Аналіз поведінки визначення відхилень від нормального функціонування системи.
- Підтримка різних джерел логів - журналів з мережевих пристроїв, серверів, клієнтів.
- Можливість інтеграції SIEM/Honeypot-системами для обміну даними та централізованого моніторингу.
- Формування інцидентів і сповіщень - автоматична генерація повідомлень для адміністраторів або SOC-груп.
- Підтримка адаптивності - система має навчатись або дозволяти адміністратору оперативно змінювати правила виявлення.

2. Нефункціональні вимоги:

a. Масштабованість - здатність обробляти великі обсяги трафіку та подій без зниження ефективності.

b. Стійкість до навантаження та збоїв - система повинна працювати стабільно за умов пікових навантажень.

c. Зручність використання - Інтерфейс має бути інтуїтивним для адміністраторів безпеки.

d. Швидкість виявлення - Чим менше затримка між інцидентом та часом його виявлення, тим ефективніше реагування.

e. Відповідність стандартам - зокрема, ISO/IEC 27001, NIST SP 800-94, національні норми НД ТЗІ. (рис 2.2)

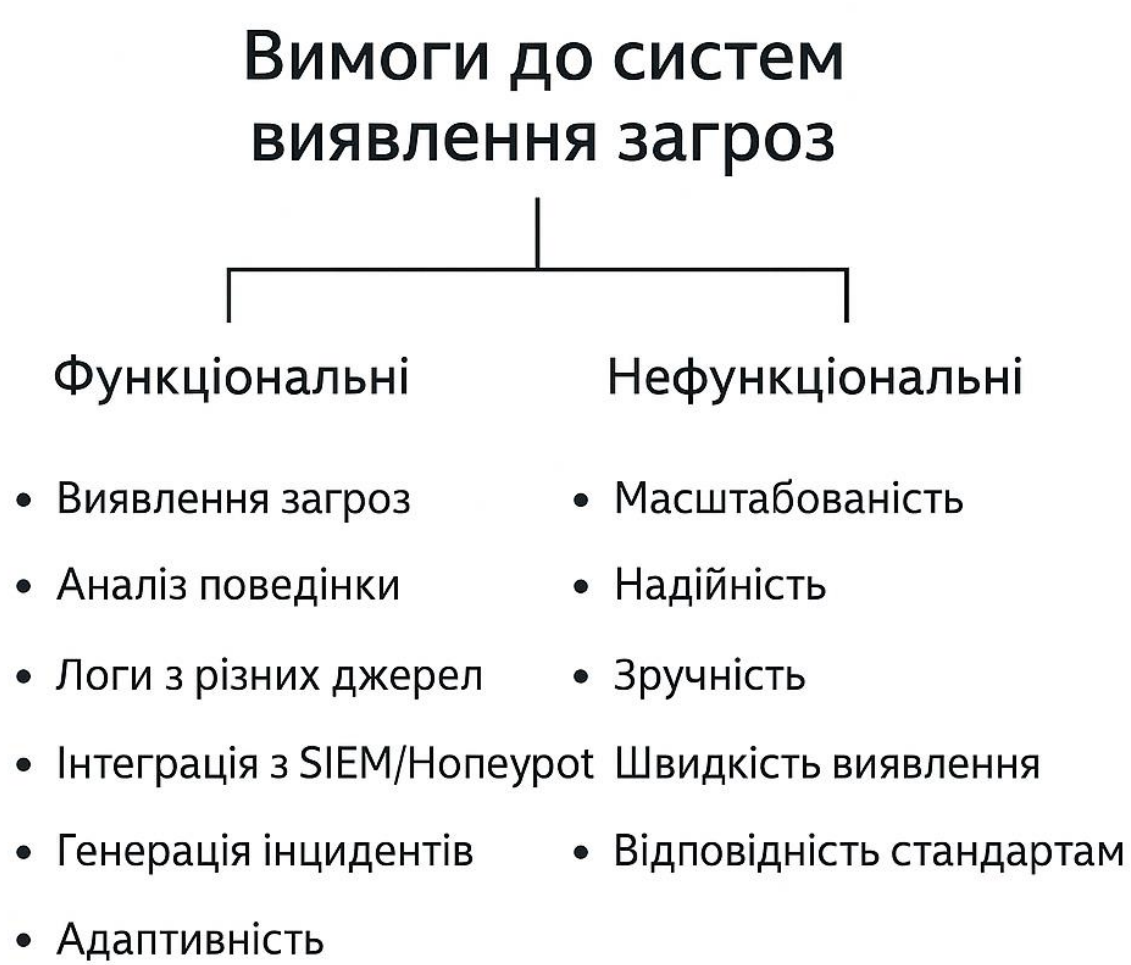

Рисунок – 2.2 Вимоги до систем виявлення загроз

2.4. Роль Honeypot-систем у вирішенні проблем виявлення загроз

У сучасних інформаційних інфраструктурах підприємств традиційні засоби виявлення кіберзагроз - такі як системи виявлення вторгнень(IDS), антивірусні рішення, міжмережеві екрани - деталі частіше виявляються недостатньо ефективними. Це пов’язано із зростанням складності та різноманітності Атак, появою цілеспрямованих, складних загроз, а також із загрозами з боку внутрішніх користувачів.

Основна роль Honeypot-систем полягає у створенні штучного привабливого середовища - яке імітує вразливу систему або сервіс і приваблює зловмисників. Таким чином, Honeypot-системи дозволяють не лише швидко виявляти а й аналізувати атаки, формувати ефективні контрзаходи та вдосконалювати політики захисту підприємств.

На відміну від звичайних засобів виявлення, honeypot не лише ідентифікує факт атаки а й дозволяє глибше зрозуміти мотивацію, стратегії та інструменти зловмисника. Це особливо важливо у випадках нових, раніше невідомих загроз, так званих зіродей вразливостей, які традиційні антивірусні рішення часто не можуть виявити через відсутність даних. Зібрана в honeypot інформація може бути використана для розробки нових правил виявлення оновлення систем виявлення вторгнень та покращення стратегій кіберзахисту.

Крім цього honeypot дозволяє виявити фази які зазвичай лишаються непоміченими - наприклад розвідку переміщення мережею спроби експлуатації вразливостей встановлення бекторів тощо.

Honeypot-системи виступають важливим доповненням до традиційних механізмів захисту, дозволяючи значно розширити можливості виявлення, аналізу та реагування на кіберзагрози. [6]

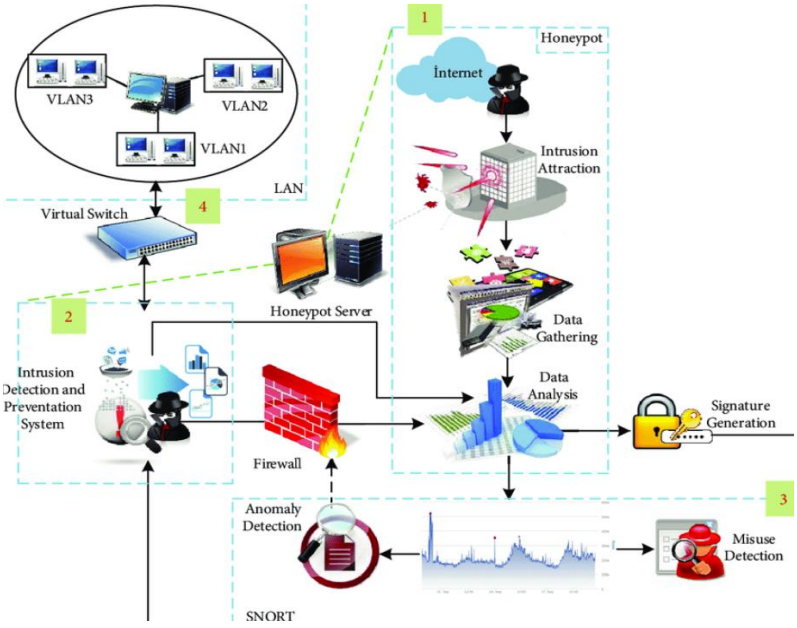

Рис 2.3 Наглядний приклад повної архітектури захисту Honeypot у корпоративній системі. [8]

1. Залучення зловмисника (Intrusion Attraction)

Архітектура починається з розгортання Honeypot-системи, яка знаходиться у відкритому або частково відкритому доступі для потенційних атак із зовнішнього середовища (Інтернету).

● Honeypot імітує вразливі сервіси (наприклад, відкриті порти, незахищені бази даних, старі версії ОС тощо).

● Зловмисник, скануючи мережу, знаходить ці сервіси та починає взаємодію з Honeypot, вважаючи його реальною мішенню.

2. Фіксація та збір даних про атаку (Data Gathering)

Усі дії атакувальника в Honeypot детально фіксуються.

● Записуються: Команди, які він виконує. Застосовані експлойти. Файли, які завантажуються або передаються. Встановлені бекдори або шкідливе ПЗ. Створюється повний журнал активності для подальшого аналізу.

3. Аналітика та виявлення загроз (Data Analysis)

Зібрана інформація надходить до аналітичного модуля Honeypot-серверу.

● Аналізуються:

Поведінкові шаблони зловмисника. Нестандартні техніки вторгнень. Ознаки атак нульового дня (zero-day). Виявляються ознаки нових або цілеспрямованих атак, які ще не мають сигнатур.

4. Генерація сигнатур (Signature Generation)

На основі проаналізованих дій формуються:

● Нові правила для IDS/IPS-систем (наприклад, Snort або Suricata). Машинно-читані сигнатури, які дозволяють розпізнавати подібні атаки в майбутньому. Це суттєво покращує здатність системи ідентифікувати загрози ще до їхнього проникнення у продуктивне середовище.

5. Виявлення вторгнень у реальному часі (Intrusion Detection System)

Згенеровані сигнатури надходять до системи виявлення вторгнень (IDPS).

● Вона аналізує мережевий трафік у реальному часі. Може як сигналізувати про підозрілу активність, так і блокувати її (у разі інтеграції з фаєрволом).

6. Виявлення аномалій (Anomaly Detection)

Паралельно з цим працює система виявлення аномалій (наприклад, SNORT).

● Вона оцінює поточну поведінку користувачів і пристроїв. Якщо зафіксовано відхилення від типових патернів (наприклад, нічна активність або нетипові порти) — система реагує. Цей рівень дозволяє виявляти невідомі загрози, навіть без чітких сигнатур.

7. Виявлення зловживань (Misuse Detection)

Останній рівень оборони — це фільтрація за вже відомими шаблонами атак.

● Використовується база даних загроз і попередньо згенеровані сигнатури.

Виявляються спроби використання відомих уразливостей, вірусів, троянів тощо.

8. Впровадження в мережеву інфраструктуру (VLAN, Switch, Firewall)

● Захист реалізується на рівні віртуального комутатора (Virtual Switch), що керує доступом між VLAN-сегментами. Трафік проходить через фаєрвол, який блокує або дозволяє з’єднання на основі правил. Вся система інтегрована в корпоративну мережу, що дозволяє безпечно вивчати загрози, не ризикуючи основною інфраструктурою.

Таким чином honeypot системи не лише підвищують загальну ефективність системи безпеки а й забезпечують підприємствам додатковий рівень захисту орієнтований на виявлення складних таргетованих або ще невідомих загроз

2.5 Переваги та неділоки Honeypot

Для ефективного використання ханіпот систем у забезпеченні кібербезпеки важливо розуміти їхні основні переваги та обмеження. Різні типи ханіпод систем мають свої особливості що впливають на рівень захисту,складність провадження та експлуатації. [6]

Таблиця 2.3

Переваги та недоліки Honeypot

Тип Honeypot	**Переваги**	**Недоліки**
Високо Деталізовані	Дає детальний аналіз атак, допомагає глибоко зрозуміти поведінку зловмисників	Вимагають багато ресурсів, складні у налаштуванні та підтримці
Низько Деталізовані	Легко встановлюються і обслуговуються, швидко збирають базову інформацію про атаки	Надають обмежену інформацію, не підходять для складних атак
Фізичні	Висока реалістичність, зловмисник не підозрює штучність	Дорогі у впровадженні, вимагають спеціального обладнання
Віртуальні	Дешевші та більш гнучкі, легкі для масштабування	Можуть бути менш переконливими для досвідчених хакерів

Висновок до розділу 2

У сучасних інформаційних інфраструктурах підприємств виявлення кіберзагроз стає все більш складним через зростання кількості та різноманіття атак, а традиційні засоби безпеки часто не забезпечують необхідного рівня захисту. Аналіз основних проблем традиційних методів показав їхню обмеженість у виявлені нових та цілеспрямованих загроз, а також складність у своєчасному реагуванні на них. У цьому контексті honeypot-системи виступають перспективним інструментом що дозволяє не лише ідентифікувати атаки але й детально аналізувати поведінку зловмисників, забезпечуючи глибше розуміння методів та тактик нападників. Це, в свою чергу, сприяє підвищенню ефективності засобів безпеки та розробці більш адаптивних контрзаходів.

Водночас, застосування honeypot-системи вимагає врахування їхніх переваг і обмежень що залежить від типу та масштабу впровадження. Розуміння цих аспектів є необхідною умовою для оптимального вибору та налаштування систем у конкретних умовах підприємства. Подальше дослідження і вдосконалення honeypot-технологій має велике практичне значення для підвищення рівня кіберзахисту особливо в умовах зростаючих складних та цілеспрямованих кіберзагроз.

РОЗДІЛ 3
РЕАЛІЗАЦІЯ ВИЯВЛЕННЯ ТА АНАЛІЗУ ВТОРГНЕННЯ ЗА ДОПОМОГОЮ HONEYPOT-СИСТЕМИ

3.1 Постановка задачі та обґрунтування вибору технології.

З кожним роком зростає складність та різноманіття кіберзагроз які спрямовані на офіційні системи підприємств. Зловмисники використовують дедалі витонченіші методи проникнення, а традиційні засоби захисту не завжди здатні ефективно їх виявити. У зв'язку з цим актуальним є застосування альтернативних підходів до моніторингу діл підозрілої активності в мережі. Одним із таких підходів є використання honeypot-систем, які імітують вразливі сервіси або пристрої й дозволяють спостерігати за діями потенційного зловмисника в контрольованому середовищі.

Метою цього етапу роботи стало встановлення, налаштування та тестування honeypot-системи, що дозволяє:

- виявити спроби несанкціонованого доступу до мережі.
- фіксувати дії атакувальника.
- аналізувати його інструменти та поведінкові моделі.
- використовувати зібрані дані для покращення загальної системи кіберзахисту.

Для Практичної реалізації було обрано Cowrie Honeypot - популярну open-source систему, яка спеціалізується на емуляції SSH та Telnet-серверів. Вона здатна записувати спробу входу, збережені паролі, команди, що використовуються зловмисником, та іншу цінну інформацію. Cowrie добре документована, активно підтримується спільнотою, та легко встановлюється у віртуальному середовищі.

Основними критеріями вибору саме цієї технології стали:

- придатність для фіксації типових атак.

- низький поріг ходу в плані налаштування.
- сумісність із інструментами аналізу логів.
- можливість інтеграції в загальну архітектуру захисту підприємства.

Окрім цього, використання Cowrie дозволяє моделювати поведінку зловмисника в умовах, максимально наближених до реального середовища, але без ризику для критичної інфраструктури.

3.2 Опис середовища тестування та інфраструктури

Для реалізації Honeypot-системи Було створено окреме тестове середовище, яке забезпечує ізоляцію від основної мережі та дозволяє безпечно проводити спостереження за потенційними вторгненнями. Це середовище має мінімальні апаратні вимоги та може бути легко масштабоване у разі потреби.

Обладнання та програмне забезпечення

усі компоненти були розгорнуті у віртуальній машині, створений за допомогою VirtualBox. В якості гостьової операційної системи використовувалась Ubuntu 22.04 LTS, [11] оскільки вона стабільна та легка.

Параметри віртуальної машини:

Операційна система: Ubuntu Server 22.04 (x64)

Процессор: 4 ядра

Оперативна пам'ять: 8 Гб

Мережевий режим: “Bridged Adapter” (Для Отримання власної IP- адреси в локальній мережі)

Диск: 25 ГБ

Зовнішні сервери: OpenSSH, Python3, Git.

У якості Honeypot-системи було обрано Cowrie - відкритий емулятор SSH/Telnet, який дозволяє фіксувати усю активність зловмисника після підключення: введені логіни, паролі, команди, IP-адреси, час підключення, тип атак, тощо.

Для початку встановимо Ubuntu Server 22.04 (x64) за допомогою VirtualBox [11] (рис 3.1)

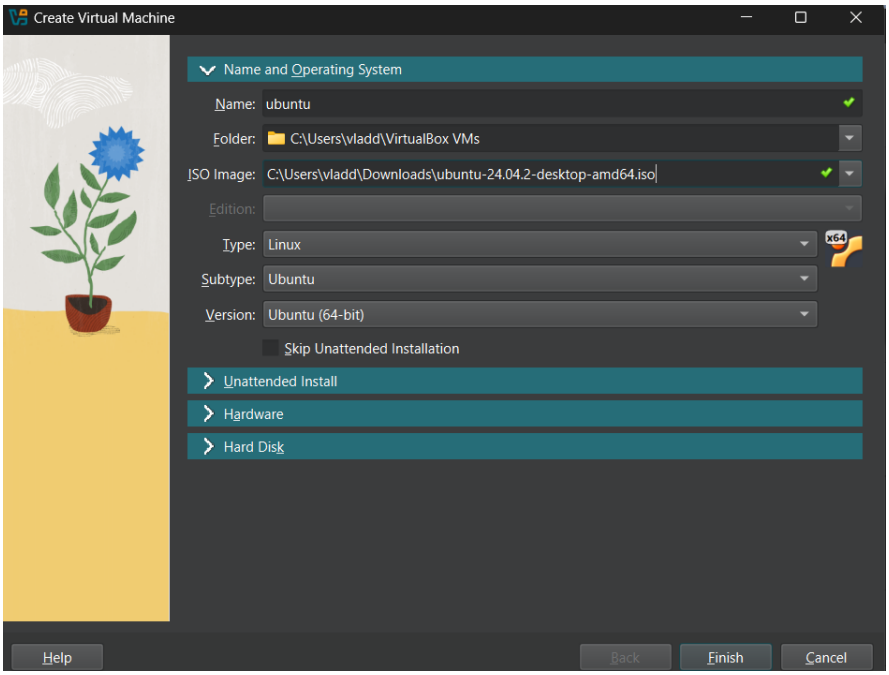

Рисунок-3.1 Створення середовища віртуальної машини

Після створимо нашого користувача, додавши username та password (рис 3.2)

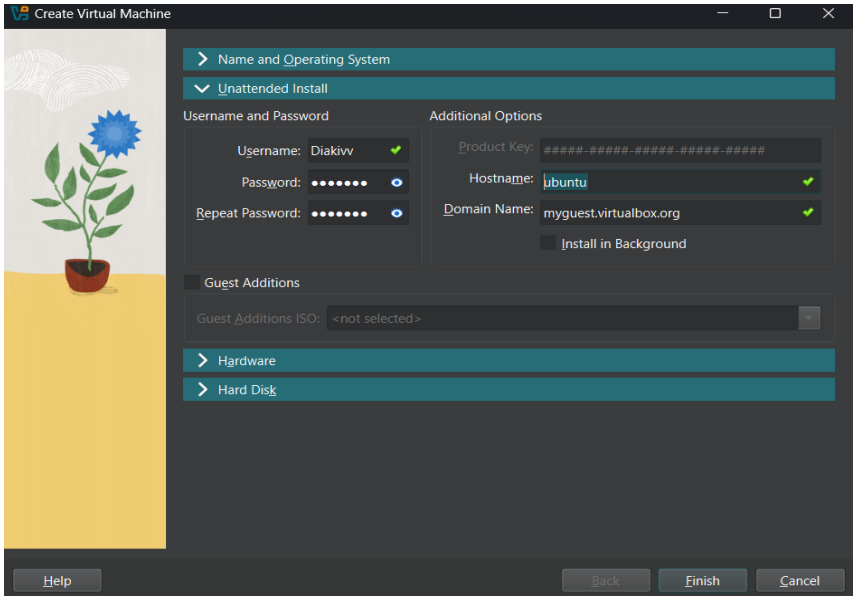

Рисунок-3.2 Створення користувача у VirtualBox

Також налаштуємо оперативну пам’ять машини та збільшимо кількість ядер в процесорі щоб підвищити ефективність роботи системи (рис 3.3)

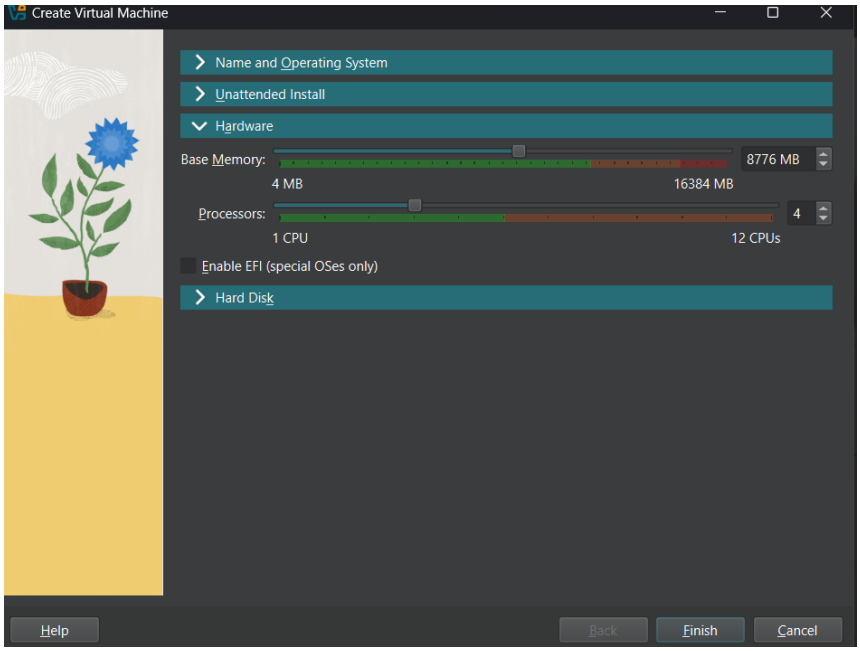

Рисунок-3.3 Налаштування Оперативної пам’яті та процесора

Створюємо віртуальний жорсткий диск, де будуть зберігатись всі наші файли (рис 3.4)

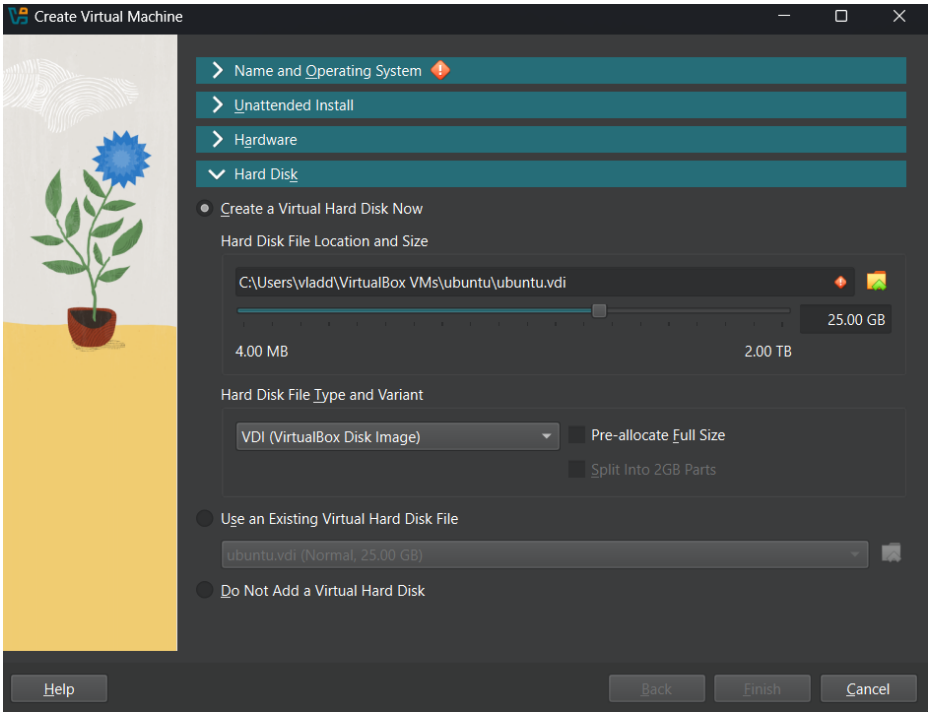

Рисунок-3.4 Створення віртуального жорсткого диску

Після успішної установки Ubuntu LTS 22.04 запускаємо нашу віртуальну машину.(рис 3.5)

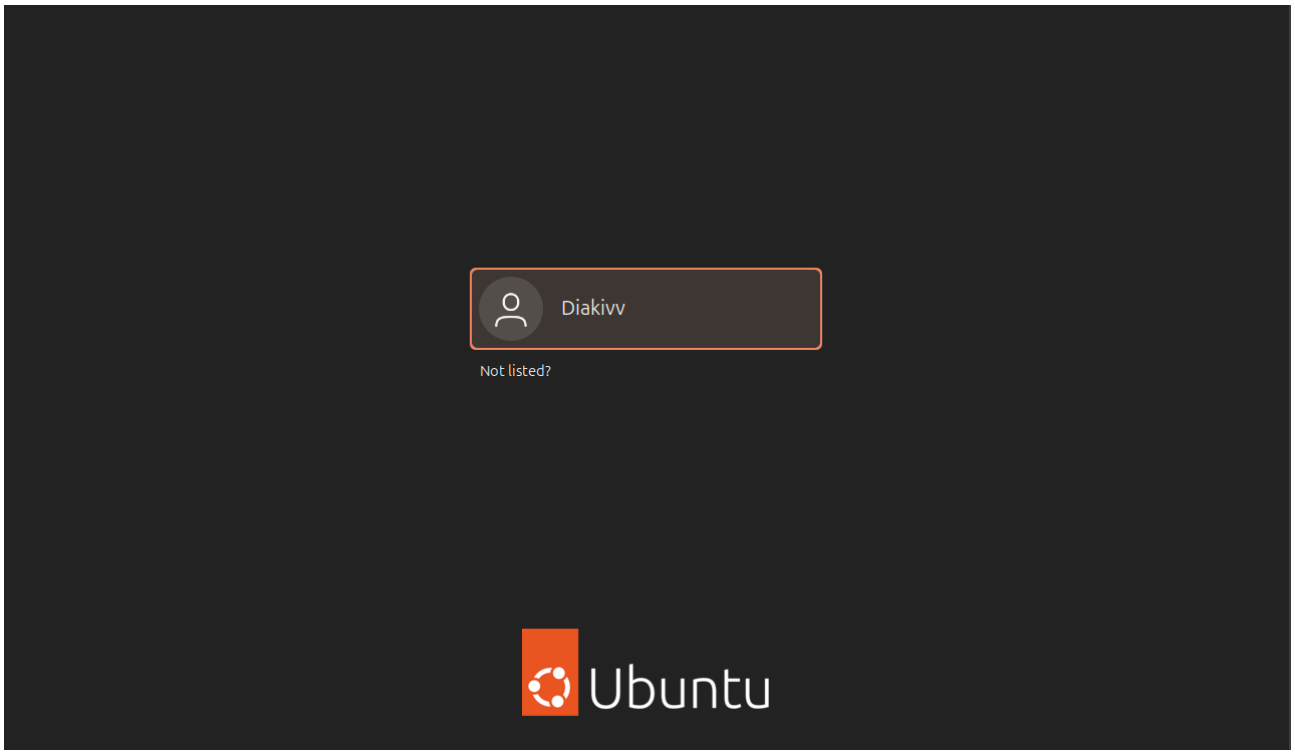

Рисунок-3.5 Запуск віртуальної машини

Віртуальну машину успішно встановлено.

Мережева структура:

Схема роботи побудована так, щоб створити враження справжнього сервера.Вхідні запити до відкритого SSH-порту(TCP 22) скеровуються на Honeypot, де запускається емуляція терміналу.

Щоб зменшити ризики, було дотримано таких умов:

Cowrie працює у непривілейованому користувачеві, в відмінному від root.

Інтернет-З'єднання обмежено лише для надсилання логів.

Логи зберігаються як локально, так і передаються через захищений канал на інший вузол для подальшого аналізу.

3.3 Налаштування та розгортання Honeypot-системи

Для реалізації Honeypot-системи було обрано Cowrie-емульований SSH/Telnet-сервер, орієнтований на фіксацію дій атакувальника у фальшивому середовищі.Його встановлення на Ubuntu не потребує складних дій і дає змогу швидко розгорнути повноцінне рішення для збору інформації про кіберзагрози.

Встановлення Cowrie Honeypot

Для початку оновимо нашу систему Командою

sudo apt update && sudo apt upgrade -y(рис 3.6)

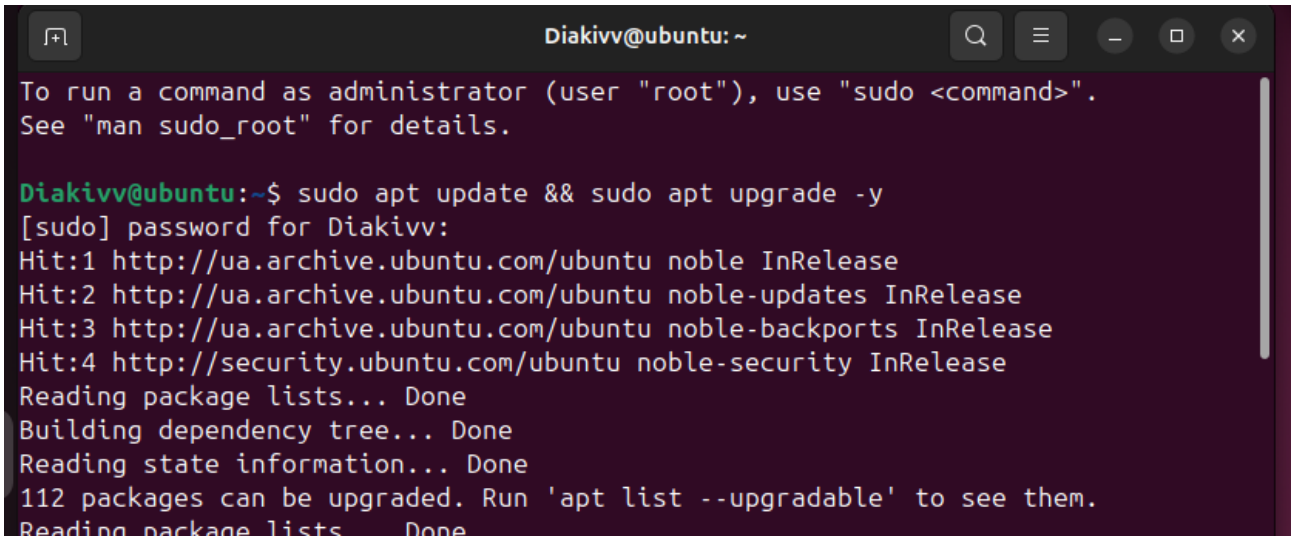

Рисунок-3.6 Повне оновлення систем перед початком встановки необхідного ПЗ

Наступна команда допомагає встановити все необхідне для роботи ПЗ

sudo apt install -y git python3 python3-venv python3-pip libssl-dev libffi-dev build-essential(рис 3.7)

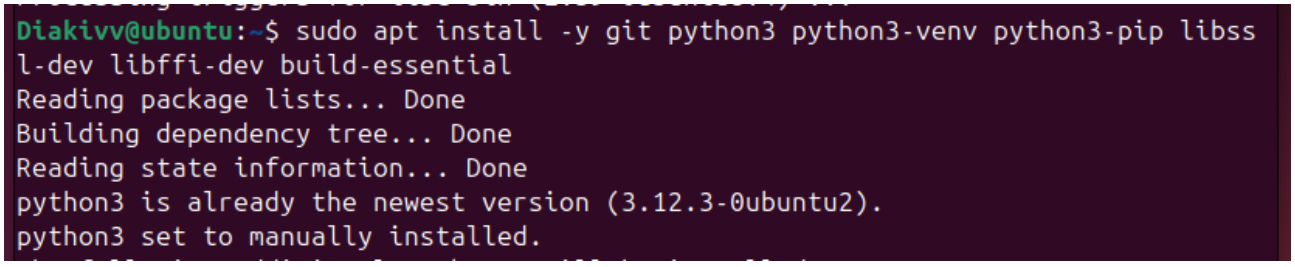

Рисунок-3.7 Встановлення необхідного ПЗ

Після нам необхідно клонувати репозиторію Cowrie з GitHub командою git clone https://github.com/cowrie/cowrie.git(рис 3.8)

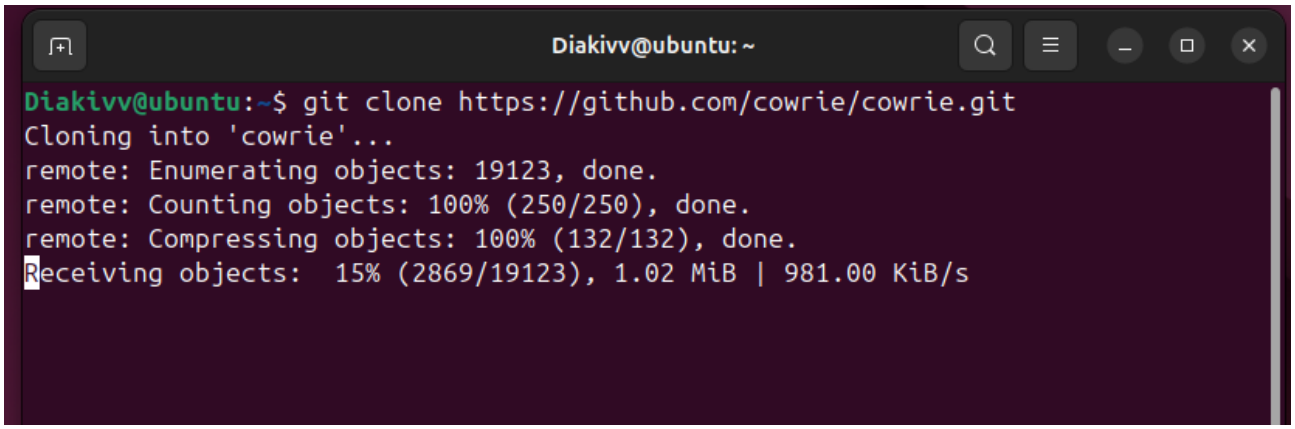

Рисунок-3.8 Клонування відбулось успішно [9]

Далі нам необхідно перейти в диск де розташоване наш Cowrie інструмент та Створити і активувати віртуальне середовище

cd cowrie

python3 -m venv cowrie-env

source cowrie-env/bin/activate [14] (рис 3.9)

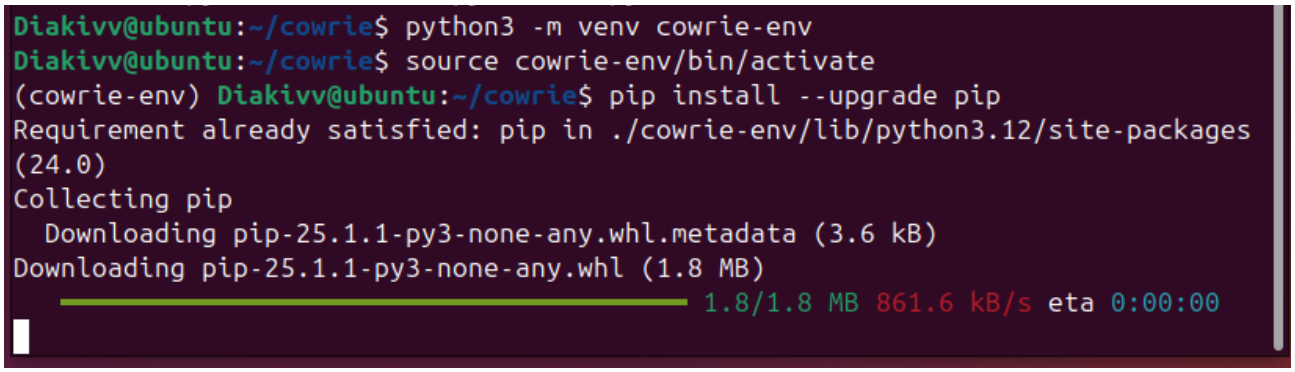

Рисунок-3.9 Створення та активація віртуального середовища

Після необхідно встановити залежності командами:

pip install --upgrade pip(рис 3.10)

pip install -r requirements.txt(рис 3.11)

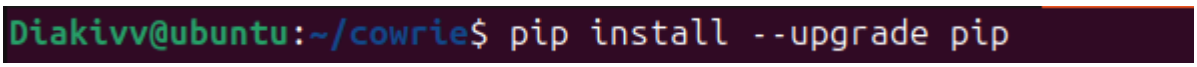

Рисунок-3.10 Команда pip install --upgrade pip

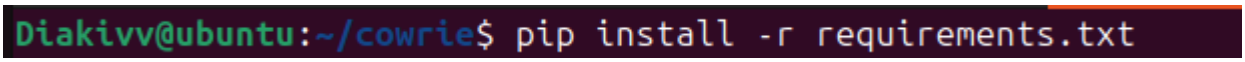

Рисунок-3.11 Команда pip install -r requirements.txt

Тепер налаштовуємо конфігурацію

cp etc/cowrie.cfg.dist etc/cowrie.cfg [14]

nano etc/cowrie.cfg(рис 3.12)

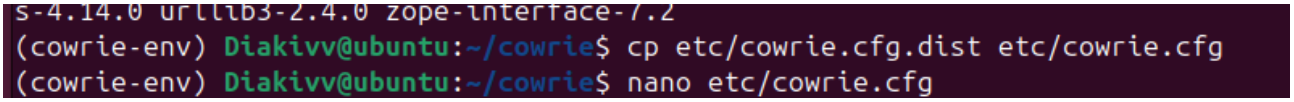

Рисунок-3.12 Налаштування конфігурації Cowrie

Запуск нашої Honeypot командою bin/cowrie start(рис 3.13)

Рисунок-3.13 Програма успішно запущена

Cowrie Починає слухати вхідні запити. Усі команди, логіни, спроби входу - фіксуються у логах

Тепер необхідно здійснити ручне підключення до Cowrie (використавши порт 2222, за замовчуванням):(рис 3.14)

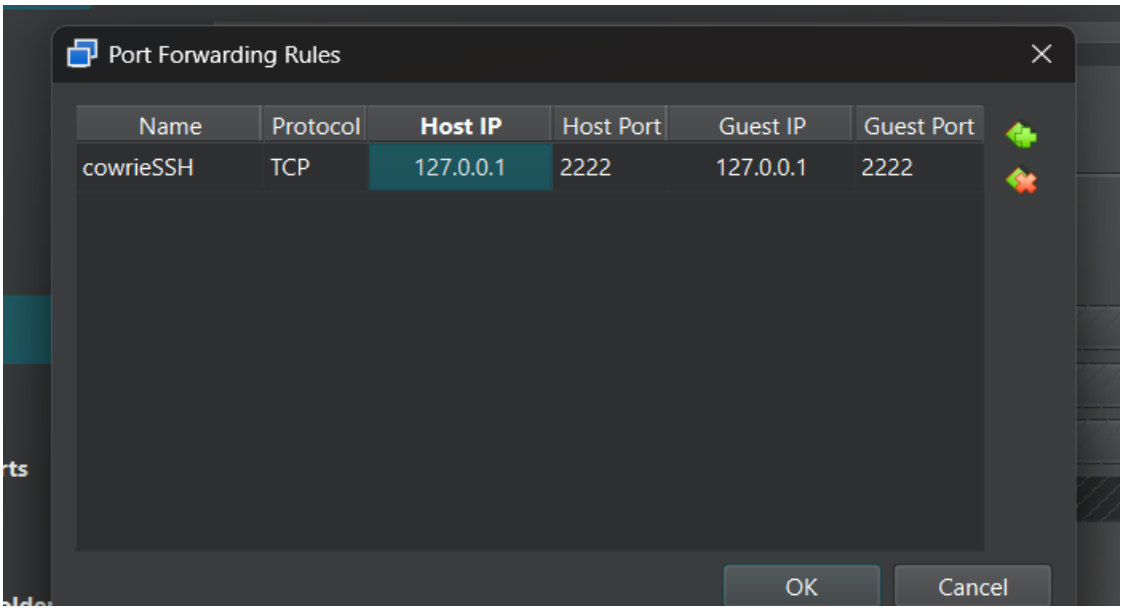

Рисунок-3.14 Налаштування port forwarding у VirtualBox [13]

Використовуючи команду ssh root@127.0.0.1 -p 2222

Повторно запускаємо Cowrie і фіксуємо результат.(рис 3.15)

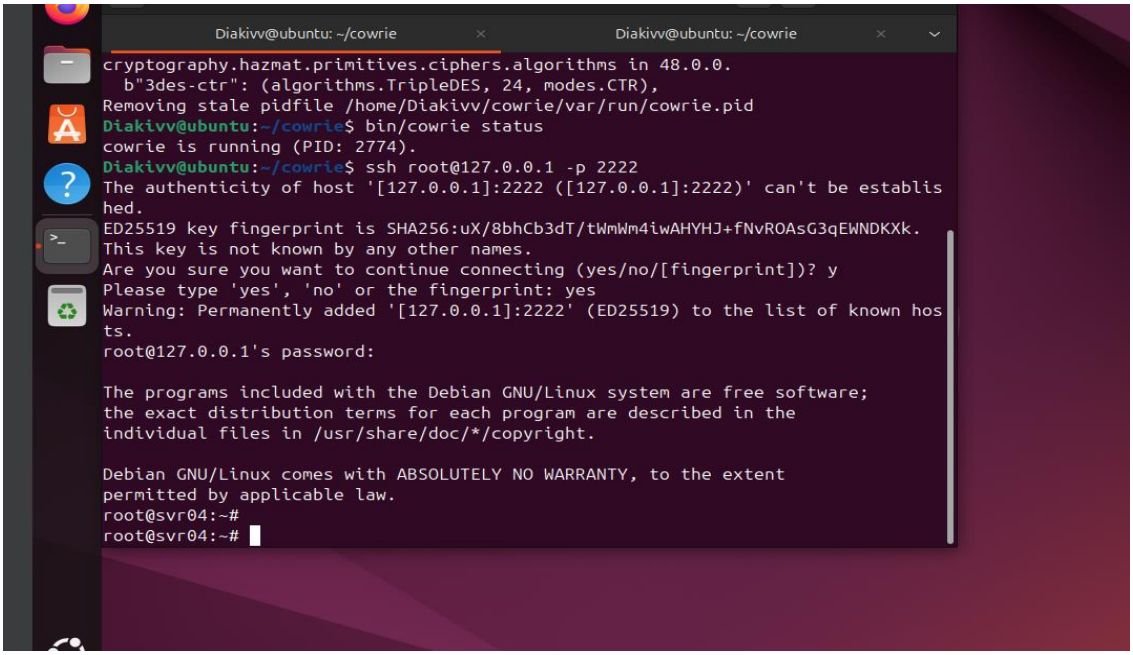

Рисунок-3.15 Бачимо що Cowrie сервіс запущений.

ssh root@127.0.0.1 -p 2222 показує що з’єднання відбулось успішно.

З’явився фейковий Debian GNU/Linux із умоввою root@svr04:~# - що вказує на успішну емуляцію від Cowrie, також бачимо команду password: - Cowrie дозволяє будь-який пароль,але фіксує його в логах.[13]

3.4 Емуляція атак та автоматичний збір даних

Після успішного розгортання Honeypot-системи на базі Cowrie було проведено серію тестових підключень з метою перевірки працездатності системи та збору даних для подальшого аналізу. Атаки імітували вручну через термінал що дозволило детально контролювати кожен етап Honeypot-сервером.

Імітація атак

Було використано звичайне SSH- підключення до Honeypot за допомогою команди:

ssh root@127.0.0.1 -p 2222

Спробуємо змоделювати поведінку справжнього хакера у нашій системі- ввівши кілька команд на прикладі:

Для розвідки системи:

- whoaim
- uname -a
- hostname
- uptime
- id

Для перегляду структури системи:

- ls -al /
- ls - al /home
- cat /etc/os-release
- df -h

Для роботи з користувачами:

- cat /etc/passwd
- cat /etc/shadow
- w

Мережеві команди:

- ifconfig
- ip a
- netstat -tuln
- ping 8.8.8.8 -c 3
- curl http://example.com [16] (рис 3.16-3.17)

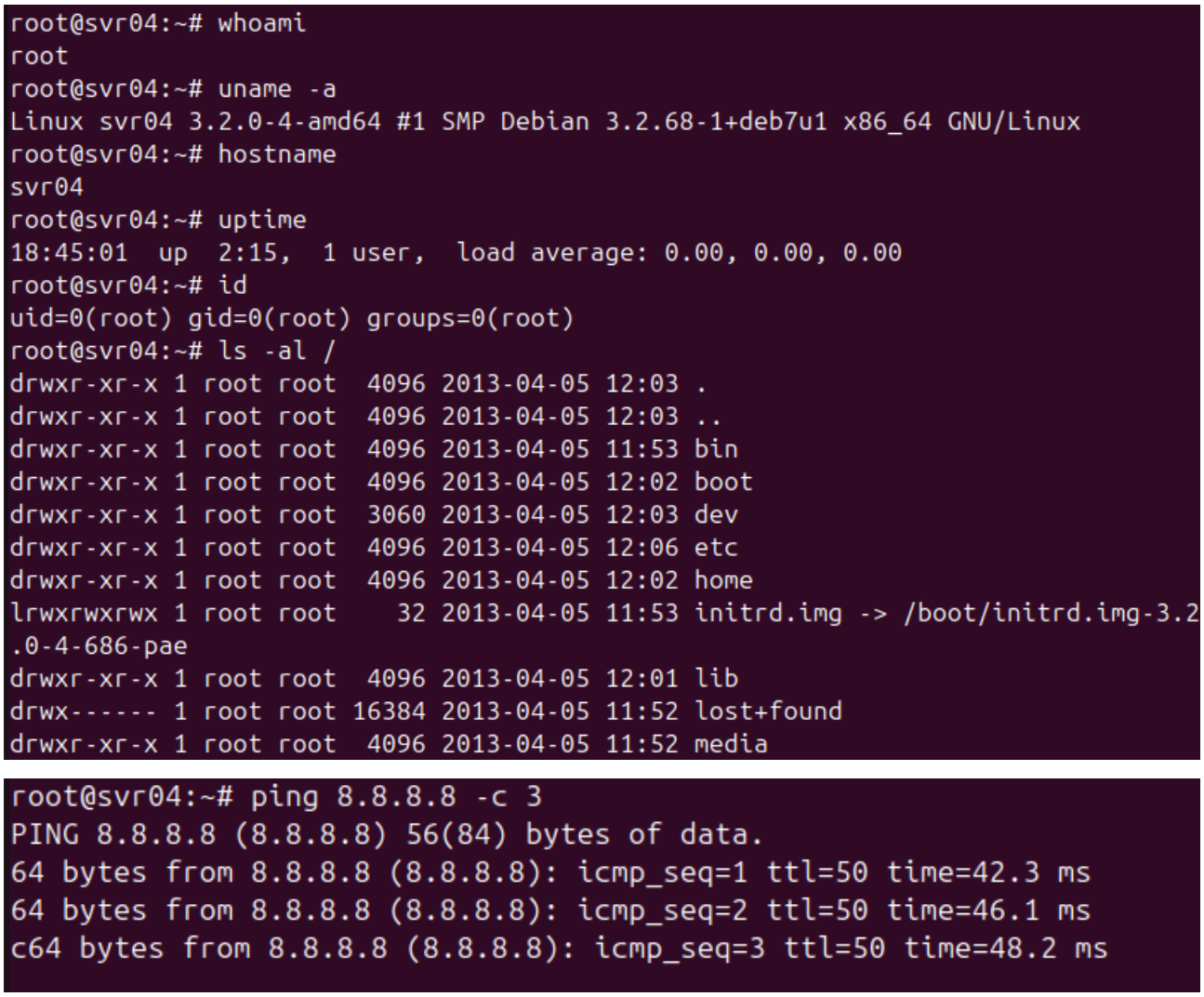

Рисунок-3.16-3.17 Приклад застосування кількох команд на стороні “хакера” [16]

І після симуляції командлю exit завершуємо нашу сесію.

3.5 Аналіз зафіксованих вторгнень

Фіксувати вторгення можна командою tail -n 30 var/log/cowrie/cowrie.log (рис 3.18-3.19)

```
Diakivv@ubuntu:~/cowrie$ tail -n 30 var/log/cowrie/cowrie.log
2025-06-02T18:45:25.795656Z [HoneyPotSSHTransport,1,127.0.0.1] CMD: cat /etc/os-
release
2025-06-02T18:45:25.796301Z [HoneyPotSSHTransport,1,127.0.0.1] Command found: ca
t /etc/os-release
2025-06-02T18:45:29.169181Z [HoneyPotSSHTransport,1,127.0.0.1] CMD: df -h
2025-06-02T18:45:29.173619Z [HoneyPotSSHTransport,1,127.0.0.1] Command found: df
 -h
2025-06-02T18:45:29.173952Z [HoneyPotSSHTransport,1,127.0.0.1] Reading txtcmd fr
om "src/cowrie/data/txtcmds/bin/df"
2025-06-02T18:45:40.476379Z [HoneyPotSSHTransport,1,127.0.0.1] CMD: cat /etc/pas
swd
2025-06-02T18:45:40.476800Z [HoneyPotSSHTransport,1,127.0.0.1] Command found: ca
t /etc/passwd
2025-06-02T18:45:46.591836Z [HoneyPotSSHTransport,1,127.0.0.1] CMD: lastlog
2025-06-02T18:45:46.594979Z [HoneyPotSSHTransport,1,127.0.0.1] Can't find comman
d lastlog
2025-06-02T18:45:46.595129Z [HoneyPotSSHTransport,1,127.0.0.1] Command not found
: lastlog
2025-06-02T18:45:51.568159Z [HoneyPotSSHTransport,1,127.0.0.1] CMD: w
2025-06-02T18:45:51.585740Z [HoneyPotSSHTransport,1,127.0.0.1] Command found: w
2025-06-02T18:46:02.752098Z [HoneyPotSSHTransport,1,127.0.0.1] CMD: ifconfig
2025-06-02T18:46:02.753351Z [HoneyPotSSHTransport,1,127.0.0.1] Command found: if
config
```

```
2025-06-02T18:46:14.979691Z [HoneyPotSSHTransport,1,127.0.0.1] Command found: ne
tstat
2025-06-02T18:46:22.619861Z [HoneyPotSSHTransport,1,127.0.0.1] CMD: ping 8.8.8.8
 -c 3
2025-06-02T18:46:22.620408Z [HoneyPotSSHTransport,1,127.0.0.1] Command found: pi
ng 8.8.8.8 -c 3
2025-06-02T18:49:25.678880Z [twisted.conch.ssh.session#info] exitCode: 1
2025-06-02T18:49:25.679401Z [cowrie.ssh.connection.CowrieSSHConnection#debug] se
nding request b'exit-status'
2025-06-02T18:49:25.680605Z [-] Closing TTY Log: var/lib/cowrie/tty/7ba6a8f1c057
2fbb02c7071c6472c7c4729f09d35de88e3e265c5946331c731a after 284.4 seconds
2025-06-02T18:49:25.681014Z [cowrie.ssh.connection.CowrieSSHConnection#info] sen
ding close 0
2025-06-02T18:49:25.683514Z [cowrie.ssh.session.HoneyPotSSHSession#info] remote
close
2025-06-02T18:49:25.683918Z [HoneyPotSSHTransport,1,127.0.0.1] Got remote error,
 code 11 reason: b'disconnected by user'
2025-06-02T18:49:25.772640Z [HoneyPotSSHTransport,1,127.0.0.1] avatar root loggi
ng out
2025-06-02T18:49:25.772856Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] con
nection lost
2025-06-02T18:49:25.772932Z [HoneyPotSSHTransport,1,127.0.0.1] Connection lost a
fter 287.6 seconds
Diakivv@ubuntu:~/cowrie$
```

Рис 3.18-3.19 Список всі застосованих команд “хакером” в нашій Honeypot

Після запуску Honeypot-систем та виконання тестових підключень було зібрано Log файли, які містять інформацію про сесії, спроби входу, введення команди, а також дії потенційного зловмисника. Для обробки цих логів було розроблено власний скрипт мовою Python, Який автоматизує аналіз подій, зафіксованих у структурованому форматі cowrie.json. [15](рис 3.20)

Рисунок-3.20 Розробка скрипта на мові програмування Python [10]

Для того щоб наш скрипт працював з нашою системою потрібно було знайти розташування json файла де зберігалась інформацію для цього я використав команду sudo find / -name "cowrie.json" 2>/dev/null(рис 3.21)

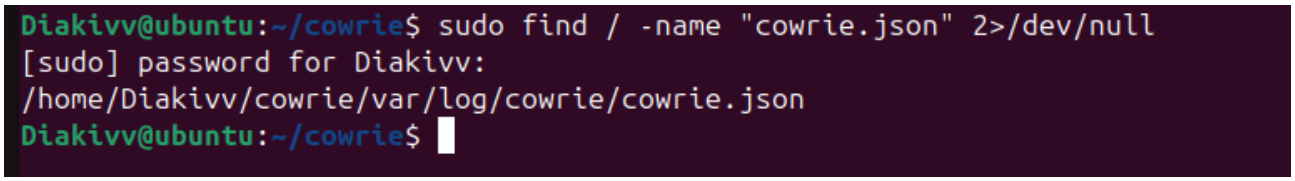

Рис 3.21 Пошук розташування json файлу для подальшого застосування у скрипті

Обробка логів.Основний файл для аналізу - cowrie.json, який містить події у форматі JSON, де кожен рядок - окремий запис: початок сесії, введена команда, завершення сесії, тощо.

Щоб уникнути ручного перегляду великої кількості записів, було створено Python-скрипт, який:

Підраховує кількість підключень за IP-адресами.

Аналізує команди, які були введені.

Фіксує тривалість сесії.(рис 3.21-3.22)

```
  GNU nano 7.2                                log.py
import json
from collections import Counter

log_path = "/home/Diakivv/cowrie/var/log/cowrie/cowrie.json"

ip_counter = Counter()
commands = []
sessions = {}

with open(log_path, 'r') as log_file:
    for line in log_file:
        try:
            event = json.loads(line)
            if event["eventid"] == "cowrie.session.connect":
                ip = event["src_ip"]
                ip_counter[ip] += 1

            elif event["eventid"] == "cowrie.command.input":
                commands.append(event["input"])
```

```
  GNU nano 7.2                                log.py
            elif event["eventid"] == "cowrie.session.closed":
                sid = event["session"]
                duration = event.get("duration", 0)
                sessions[sid] = duration

        except json.JSONDecodeError:
            continue

print(" Топ IP-адрес атакувальників:")
for ip, count in ip_counter.most_common(5):
    print(f"{ip} – {count} спроб")

print("\n Найпопулярніші команди:")
for cmd, count in Counter(commands).most_common(5):
    print(f"{cmd} – {count} разів")

print(f"\n Кількість завершених сесій: {len(sessions)}")
if sessions:
    avg = round(sum(sessions.values()) / len(sessions), 2)
    print(f" Середня тривалість сесії: {avg} секунд")
```

Рисунок-3.22-3.23 Код скрипта на мові програмування Python текстового аналізу логів Cowrie

У даній програмі ми в першу чергу бачимо підключення стандартних бібліотек

json - для зчитування .json-логів

counter - структура яка підраховує повторювані значення.

log_path = Вказує шлях до JSON-файлу з логами Cowrie.

ip_counter - підрахунок спроб підключення з кожного IP

commands - список усіх введених команд зловмисниками

session - словник сесій

Наступний код відкриває файл пострічково, оскільки у Cowrie кожен лог-окремий JSON рядок.

Після йде обробка типів подій, якщо подія підключення то беремо IP-адресу і рахуємо її, якщо подія введення команди, зберігаємо команду у список, якщо сесію завершено беремо ID та тривалість і додаємо в словник.Після є виключення в коді якщо трапилась помилка у розборі JSON коду. В результаті виконання наша програма показує IP-адреси, з яких найбільше здійснювались спроби входу, 5 найчастіше введених команд, та проводиться підрахунок всіх завершених сесій.(рис 3.24)

Запустивши даний скрипт отримуємо результат роботи системи:

```
Diakivv@ubuntu:~/cowrie$ python3 log.py
 Топ IP-адрес атакувальників:
127.0.0.1 — 2 спроб

 Найпопулярніші команди:
whoami — 2 разів
cat /etc/passwd — 2 разів
 — 1 разів
ls — 1 разів
cd — 1 разів

 Кількість завершених сесій: 2
```

Рисунок-3.24 Результати які надала система. [15]

За час роботи системи було здійснено 2 тестові спроби нападу на систему “зловмисником” Результатом даного аналізу є:

Після запуску скрипта було отримано наступну інформацію:

IP-адреса: 127.0.0.1 — симулюване підключення з хост-системи.

Середня тривалість сесії: ~22 секунд

Команди, які були введені найчастіше:

whoami - 2 рази

cat /etc/passwd -2 рази

ls - 1 раз

cd - 1 раз

Кількість завершених сесій: 2

[16]

Ці дії імітують базові кроки зловмисника після входу в систему: розвідка, перегляд користувачів, запис шкідливого файлу.

Розроблений скрипт дозволяє ефективно аналізувати вторгнення та будувати картину поведінки потенційного зловмисника. Зібрані дані можуть використовуватися для:

Формування правил IDS/IPS

Виявлення повторювальних атак

Пінтеграції з SIEM-системою

Таким чином, Honeypot-система не лише фіксує події, а й забезпечує ґрунтовну базу для аналітики, що є критично важливим для кіберзахисту підприємства.

3.6 Інтеграція Honeypot-система корпоративна інфраструктуру кіберзахисту.

Використання Honeypot-систем має значний потенціал у сфері кіберзахисту підприємств. Вони не замінюють класичні засоби захисту, але діють як додатковий рівень, орієнтований на виявлення та моніторинг активності зловмисників, які вже пройшли зовнішній периметр безпеки або діють з середини.

Архітектурна модель інтеграції

Honeypot-систему доцільно впроваджувати у відокремленій мережевій зоні (DMZ) або у віртуалізованому середовищі, яке імітує критичні сервіси підприємства.

Рекомендована архітектура:

Cowrie розгортається у віртуальній машині або контейнері, підключається до ізольованої мережі, з фільтрацією вихідного трафіку.

Логи з Honeypot передаються у централізовану систему моніторингу.

Інциденти передаються на SIEM або SOC.

Переваги для підприємства:

Раннє виявлення загроз-Honeypot фіксує перші кроки зловмисника.

Реальна аналітика - дає змогу збирати власні дані про методи атак, замість покладатися лише на зовнішні звіти.

Виявлення внутрішніх порушень - якщо хтось із локальної мережі намагається атакувати Honeypot, це прямий сигнал про інцидент.

Навчання персоналу - Honeypot може бути використана як тренажер для кіберфахівців.

Технічні особливості інтеграції:

Безпека середовища: Honeypot має відбути від'єднаний від внутрішньої інфраструктури, з обмеженим вхідним- трафіком і феєрволом на вході.

Маштабованість:

Система може бути розгорнута у вигляді кількох контейнерів або віртуальних машин у різних точках мережі.

Практичні рекомендації щодо впровадження honeypot-системи в інфраструктуру підприємства.

Розгортання в зоні DMZ:

Рекомендується встановлювати honeypot у демілітаризованій зоні(DMZ), поряд із тими сервісами, які найчастіше стають об'єктами атак - SSH,FPT. Це дозволить ефективніше виявляти спроби вторгнення до критичних точок мережі.

Централізоване логування подій.

Усі логи, згенеровані honeypot-системою, доцільно надсилати до централізованої системи збору журналів. Це дозволяє виконувати кореляційний аналіз, виявляти шаблони атак і швидко реагувати на інциденти.

Регуляторні оновлення та перевірки реалізму Honeypot.

Honeypot повинен виглядати як справжній сервер, тому його необхідно періодично оновлювати, змінювати конфігурацію, імітувати нові версії ОС або сервісів. Це допоможе уникнути виявлення системи як пастки досвідченими зловмисниками.

Налаштування сповіщень про підозрілу активність:

Доцільно реалізувати систему сповіщень, яка буде повідомляти відповідальних осіб про:

- нові підключення до Honeypot.
- спроби brute-force-атак.
- підозрілу поведінку в сесії.

Межерева сегментація та обмеження доступу:

Honeypot повинен бути суворо ізольований від продуктивної мережі. Необхідно обмежити його мережеві можливості, щоб у разі компрометації він не міг бути використаний як плацдарм для подальшого проникнення у критичну інфраструктуру.

ВИСНОВОК

У процесі виконання кваліфікаційної роботи було комплексно досліджено теоретичні, методичні та практичні аспекти застосування honeypot-систем для виявлення та аналізу кіберзагроз у корпоративному середовищі.

На основі проведеного дослідження сформульовано такі висновки:

- Досліджено сучасні кіберзагрози, актуальні для підприємств, а також проаналізовано недоліки традиційних засобів виявлення атак. Встановлено, що класичні системи IDS/IPS не завжди забезпечують ефективний моніторинг новітніх атак або дій невідомих зловмисників, що підкреслює необхідність використання додаткових засобів контролю.
- Проаналізовано принципи функціонування honeypot-систем, їх класифікацію та архітектурні моделі. Визначено, що honeypot-рішення, залежно від рівня взаємодії, можуть виконувати різні завдання — від збору базових відомостей про спроби сканування до детального аналізу технік експлуатації вразливостей та дій зловмисників.
- Визначено вимоги до безпечного розгортання honeypot-систем у корпоративній мережі. Було обґрунтовано необхідність ізоляції таких систем, ретельного налаштування журналювання, контролю мережевого трафіку та застосування віртуального середовища для мінімізації ризиків впливу на продуктивні сервіси.
- Розроблено та реалізовано експериментальний стенд honeypot-системи із використанням віртуального середовища. На практиці продемонстровано працездатність розгорнутої системи, зокрема на основі рішення Cowrie, що дало змогу фіксувати спроби несанкціонованого доступу, зібрати дані про інструменти зловмисників та їхню поведінку.

Таким чином, усі поставлені у роботі завдання виконано в повному обсязі. Запропонована honeypot-система підтвердила свою ефективність як компонент багаторівневої системи захисту. Її застосування дозволяє не лише підвищити

якість моніторингу загроз та оперативного реагування, а й накопичувати цінну інформацію для подальшого вдосконалення політик інформаційної безпеки підприємства.

Результати роботи мають практичну цінність і можуть бути використані для розгортання honeypot-рішень у реальних умовах з метою підвищення захищеності корпоративної інформаційної інфраструктури.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Spitzner, L. (2003). *Honeypots: Tracking Hackers.* Addison-Wesley Professional.

2. Kumar, D., & Sharma, M. K. (2017). Honeypot based intrusion detection systems: A review. *International Journal of Computer Applications, 169*(3), 20–25.

3. Valli, G. (2016). *A guide to honeypots.*

4. Titarmare, N., Hargule, N., & Gupta, A. (2019). An overview of honeypot systems. *International Journal of Computer Sciences and Engineering, 7*(2), 394–397. https://doi.org/10.26438/ijcse/v7i2.394397

5. Мєшков, В. І., & Віролайнен, В. О. (2015). Аналіз сучасних систем виявлення та запобігання вторгнень в інформаційно-телекомунікаційних системах. *Проблеми безпеки інформації в інформаційно-комунікаційних системах.* НТУУ КПІ РТФ. https://ela.kpi.ua/bitstream/123456789/17609/1/meshkov.pdf

6. Mittal, M., Singh, M., & Kumar, R. (2022). Honeypot-based intrusion detection systems: A comprehensive review. *Journal of Network and Computer Applications, 202,* 103362. https://doi.org/10.1016/j.jnca.2022.103362

7. Conti, M., & D'Angelo, S. (2019). Honeypot for APT detection: A survey. *Computers & Security, 85,* 230–246. https://doi.org/10.1016/j.cose.2019.05.009

8. ResearchGate. (n.d.). Architecture of the proposed honeypot-based system. https://www.researchgate.net/figure/Architecture-of-the-proposed-honeypot-based-system_fig1_345412586

9. Cowrie Project. (n.d.). *Cowrie GitHub repository.* https://github.com/cowrie/cowrie

10. GeeksforGeeks. (n.d.). Python for cybersecurity. https://www.geeksforgeeks.org/python-for-cybersecurity/

11. Canonical. (n.d.). *Install and configure Ubuntu server.* https://ubuntu.com/server/docs/install-and-configure

12. GeeksforGeeks. (n.d.). How to build a simple honeypot using Python. https://www.geeksforgeeks.org/how-to-build-a-simple-honeypot-using-python/

13. Nxnjz. (2019, January). Deploying an interactive SSH honeypot on Ubuntu 18.04. https://nxnjz.net/2019/01/deploying-an-interactive-ssh-honeypot-on-ubuntu-18-04/

14. Cowrie Project. (n.d.). *Cowrie documentation.* https://docs.cowrie.org/en/latest/README.html

15. Pittman, J. M. (n.d.). *Cowrie log analyzer GitHub repository.* https://github.com/jasonmpittman/cowrie-log-analyzer

16. GeeksforGeeks. (n.d.). Linux commands for simulating intrusions. https://www.geeksforgeeks.org/linux-commands/

ДОДАТОК

```
import json

from collections import Counter

log_path = "cowrie/var/log/cowrie/cowrie.json"

ip_counter = Counter()

commands = []

sessions = {}

with open(log_path, 'r') as log_file:

for line in log_file:

try:

event = json.loads(line)

if event["eventid"] == "cowrie.session.connect":

ip = event["src_ip"]

ip_counter[ip] += 1

elif event["eventid"] == "cowrie.command.input":

commands.append(event["input"])

elif event["eventid"] == "cowrie.session.closed":

sid = event["session"]

duration = event.get("duration", 0)
```

```
sessions[sid] = duration

except json.JSONDecodeError:

continue

print(" IP-адреси атакувальників:")

for ip, count in ip_counter.most_common(5):

print(f"{ip} — {count} спроб")

print("\n Найпопулярніші команди:")

for cmd, count in Counter(commands).most_common(5):

print(f"{cmd} — {count} разів")

print(f"\n Кількість завершених сесій: {len(sessions)}")

if sessions:

avg = round(sum(sessions.values()) / len(sessions), 2)

print(f" Середня тривалість сесії: {avg} секунд")
```