ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА СИСТЕМ ТА ТЕХНОЛОГІЙ КІБЕРБЕЗПЕКИ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Рекомендації щодо захисту інформаційної системи організації за допомогою deception in depth»

зі спеціальності *125 Кібербезпека*
(код, найменування спеціальності)

освітньо-професійної програми *Інформаційна та кібернетична безпека*
(назва програми)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Максим СЕЛІВЕРСТОВ
(підпис)

Виконав: здобувач вищої освіти групи БСД-41
СЕЛІВЕРСТОВ Максим
(прізвище, ім'я)

Керівник к.військ.н., доцент ГАХОВ Сергій
(науковий ступінь, вчене звання, прізвище, ім'я)

Рецензент д. ф. МАРЧЕНКО Віталій
(науковий ступінь, вчене звання, прізвище, ім'я)

Київ 2025

ВСТУП

Актуальність теми. З кожним роком організації все частіше зазнають складних кібератак, серед яких як масові botnet-атаки, так і високоточні цілеспрямовані APT-кампанії. Відомі інциденти - SolarWinds, WannaCry, NotPetya - довели, що навіть великі компанії та державні установи не захищені від критичних порушень безпеки [1].

Традиційні засоби кіберзахисту (фаєрволи, антивіруси, IDS/IPS) демонструють обмежену ефективність щодо складних і невідомих загроз. Середній час виявлення атаки перевищує 200 днів, що створює сприятливі умови для зловмисника.

У відповідь на це все більше уваги привертають проактивні стратегії, зокрема Deception in Depth - підхід, що базується на створенні фальшивих об'єктів (honeypots, honeytokens, honeyfiles) у мережі. Такі елементи дозволяють виявити зловмисника на ранній стадії, знизити шкоду та зібрати цінну інформацію про техніки атаки. Це робить deception-технології важливою складовою сучасної кібербезпеки.

*Постановка проблеми***.** Традиційні засоби кіберзахисту - брандмауери, антивіруси, IDS/IPS, EDR/XDR та інші - залишаються основою безпеки, проте демонструють обмежену ефективність проти складних, багатовекторних та zero-day атак. Вони здебільшого орієнтовані на виявлення відомих загроз, що робить їх вразливими до нових та маскованих атак. Прикладом такої загрози є атака CrashOverride (2016), яка обійшла класичні системи захисту та порушила роботу енергомережі України. Це показало необхідність переходу до проактивних стратегій безпеки, здатних не лише виявляти, а й впливати на хід атаки.

У цьому контексті підхід Deception in Depth, що передбачає розміщення фальшивих об'єктів у мережі (honeypots, honeytokens, honeyfiles), є актуальним рішенням. Він дозволяє виявляти загрози на ранніх етапах, фіксувати дії зловмисника та зменшувати потенційні збитки через дезорієнтацію атакуючого.

*Об’єкт дослідження***.** системи виявлення та протидії кіберзагрозам.

Предмет дослідження - методика багаторівневого оборонного обману (*Deception in Depth*) та засоби її реалізації для виявлення та нейтралізації несанкціонованих дій в інформаційній системі.

Мета роботи. створити лабораторне середовище обманного захисту з відкритими засобами та розробити рекомендації щодо впровадження deception-технологій у компаніях різного масштабу.

Завдання дослідження:

Проаналізувати сучасні кіберзагрози для інформаційних систем

Дослідити концепцію багаторівневого deception-захисту

Обґрунтувати доцільність використання ручного підходу

Розробити архітектуру тестового середовища

Налаштувати ключові компоненти deception-середовища вручну

Сформулювати практичні рекомендації для впровадження системи

Методи дослідження **-** Для досягнення поставленої мети використано комплекс методів. аналіз і синтез літературних джерел з кібербезпеки, порівняльний аналіз існуючих засобів захисту та deception-рішень. Для мітації корпоративного середовища використано програмний комплекс GNS3, програмне забезпечення з відкритим кодом, яке дозволяє симулювати мережі будь якої складності, що підходить під ціль цього дослідження.

Практичне значення одержаних результатів: Можливість використання одержаних результатів при проектуванні реальних deception-систем для огранізацій будь-якого розміру та складності за допомогою відкритих програмних засобів.

1 АНАЛІЗ ПРОБЛЕМИ ЗАХИСТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОРГАНІЗАЦІЇ

1.1. Постановка проблеми захисту інформаційної системи організації

Основні загрози інформаційним системам

У сучасному цифровому середовищі інформаційні системи є основою роботи підприємств, держустанов і критичної інфраструктури, що робить їх привабливою ціллю для кіберзлочинців. Атаки стають дедалі складнішими, з використанням соціальної інженерії, нульових вразливостей та автоматизованих засобів.

Традиційні засоби захисту (фаєрволи, антивіруси, IDS/IPS) залишаються важливими, але часто недостатніми для виявлення цілеспрямованих або прихованих загроз, особливо в умовах обмежених ресурсів і відсутності повноцінної SOC/EDR/XDR-інфраструктури.

Відповідно до звіту *IBM Cost of a Data Breach Report 2023* [2], середній час виявлення і стримування порушення безпеки становить 277 днів, з яких 204 дні - це лише етап виявлення. Таким чином, зловмисник може залишатись у мережі понад пів року, маючи змогу здійснювати несанкціонований доступ, ексфільтрувати дані або завдати шкоди критичним активам.

Зменшення середнього часу виявлення загроз - так званого Mean Time to Detect (MTTD) - є критично важливим для зниження ризиків і фінансових втрат, пов’язаних з кіберінцидентами. Як зазначає компанія Palo Alto Networks [3], застосування високоточних інструментів безпеки на базі штучного інтелекту та автоматизованих рішень дає змогу значно прискорити виявлення загроз і реагування на них. Це дозволяє організаціям суттєво мінімізувати вплив атак і скоротити «вікно вразливості».

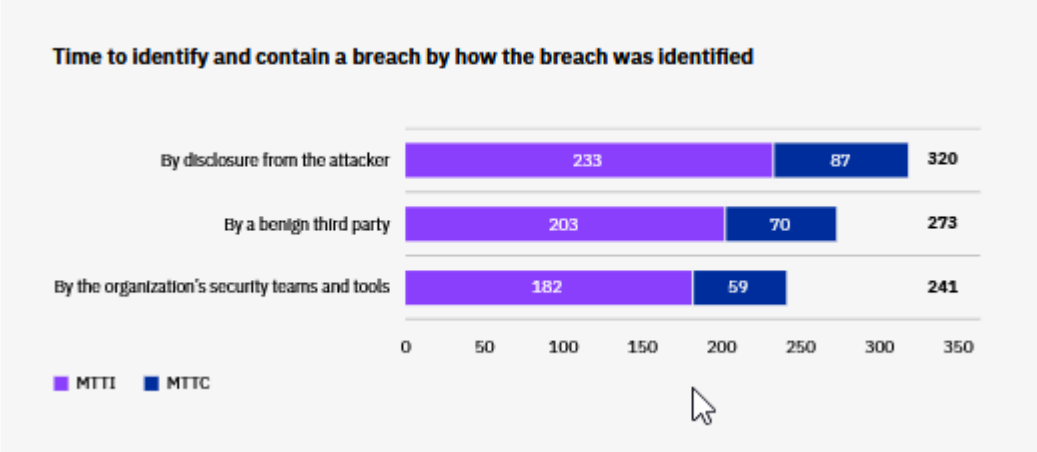

Рис. 1.1. Графік часу індентифікації зламу

На значення MTTD впливають такі чинники:

- наявність цілодобового моніторингу подій безпеки;
- використання SIEM-систем для централізованого аналізу;
- автоматизація обробки інцидентів;
- досвід та кваліфікація аналітиків;
- інтеграція поведінкових аналітичних систем (UEBA, NDR);
- застосування активних технологій виявлення, таких як Deception in Depth.

На цьому фоні підхід Deception in Depth постає як ефективне доповнення до класичних стратегій кібербезпеки, яке дозволяє не лише виявляти загрозу на ранньому етапі, а й отримувати цінну аналітичну інформацію про наміри та техніки зловмисника. Завдяки впровадженню фальшивих об'єктів - honeypots, honeytokens, honeyfiles - система безпеки набуває проактивного характеру, забезпечуючи

У відповідь на це з’являється підхід Deception in Depth - стратегія активного обману, яка передбачає розміщення в мережі фальшивих об’єктів (honeypots, honeytokens) для виявлення зловмисника на ранній стадії. Вони не впливають на продуктивність, залишаючись непомітними для легітимних користувачів.

Ця робота присвячена створенню лабораторного стенду для реалізації багаторівневого deception-захисту, з можливістю тестування на практичних сценаріях атак і оцінки ефективності виявлення загроз.

Серед основних категорій кіберзагроз можна виокремити:

Цільові атаки (APT) - тривалі, ретельно сплановані дії, спрямовані на критичні ресурси організації (сервери, БД, додатки). Виконуються спеціалізованими групами з попереднім збором інформації.

Шкідливе ПЗ - віруси, трояни, ransomware тощо. Наприклад, програми-вимагачі шифрують дані й можуть зупинити роботу підприємства до виплати викупу або відновлення систем.

Експлойти нульового дня - використовують вразливості, про які ще не відомо виробникам. Такі атаки обходять традиційні засоби захисту, бо не мають сигнатур.

Фішинг і соціальна інженерія - вплив на людей задля отримання доступу до систем. Найчастіше реалізується через електронну пошту або шкідливі посилання.

Внутрішні загрози - дії співробітників або інсайдерів, які мають легітимний доступ до систем. Вони складно виявляються через відсутність явних технічних аномалій.

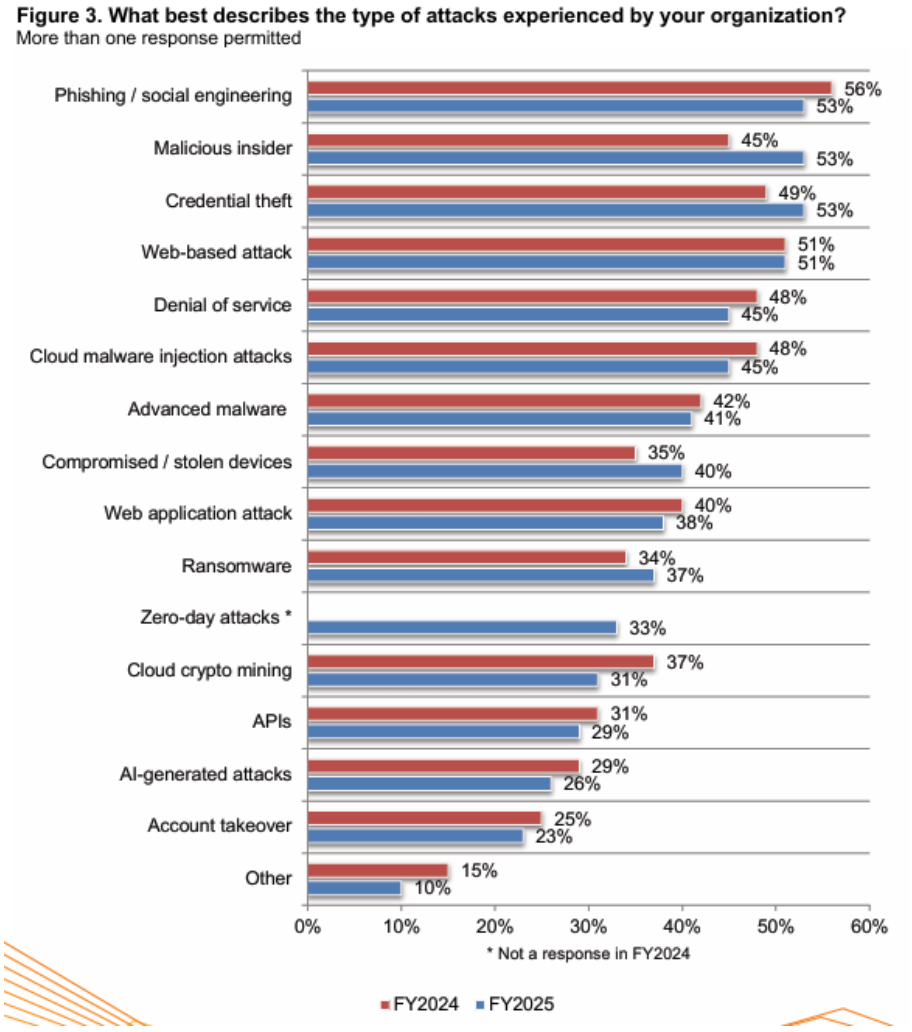

Рис. 1.2. Графік, що демонструє найпоширеніші атаки.

Основні недоліки традиційних засобів захисту

Попри широке впровадження класичних засобів захисту - таких як фаєрволи, антивірусні рішення, системи виявлення та запобігання вторгнень (IDS/IPS), засоби реагування на інциденти на рівні кінцевих точок (EDR/XDR) та міжмережеві екрани нового покоління (NGFW) - організації стикаються з дедалі більш витонченими атаками, які ці традиційні засоби не здатні ефективно стримувати. Особливо це стосується атак з використанням вразливостей нульового дня, соціальної інженерії або легітимних адміністративних інструментів, які дозволяють обійти стандартні механізми виявлення.

Класичні підходи до захисту часто демонструють низьку чутливість до невідомих загроз, високий рівень хибнопозитивних спрацювань і неадекватну видимість дій легітимних, але потенційно зловмисних користувачів (наприклад, інсайдерів). Ці проблеми значно знижують ефективність реагування команд безпеки та можуть призвести до фатальних затримок виявлення - у середньому понад 200 днів згідно з даними звіту *MixMode State of AI in Cybersecurity Report 2025* [4].

Традиційні системи виявлення зазвичай орієнтовані на вже відомі загрози, що робить їх особливо вразливими до нових, адаптивних кампаній - таких як APT-атаки. Як зазначено в огляді *Three Decades of Deception Techniques in Active Cyber Defense* (Zhang et al., 2021), класичний perimeter-based захист часто неспроможний забезпечити ефективне стримування складних інцидентів, навіть за умов багаторівневої архітектури безпеки [5].

У відповідь на ці виклики формується нова парадигма активного захисту - зокрема підхід Deception in Depth. Його суть полягає у впровадженні контрольованих обманних об’єктів - таких як honeypots, honeytokens, honeyfiles - у реальне середовище, що дозволяє виявити зловмисника вже на ранніх стадіях атаки, дезорієнтувати його, сповільнити розвиток загрози та водночас зібрати аналітику про його тактики, техніки та процедури (TTP).

Завдяки високій точності спрацювань (усі взаємодії з обманними об'єктами

апріорі є підозрілими) та відсутності залежності від сигнатур, deception-підхід значно знижує рівень інформаційного шуму в SOC. Він також дозволяє:

спостерігати за поведінкою зловмисника в реальному часі (lateral movement, privilege escalation),

зменшити ризик шкоди для критичних систем, ізолювавши зловмисника в пастці,

впливати на ухвалення ним рішень, дезорієнтуючи та підвищуючи ймовірність помилок.

Як зазначає Yuill у роботі *Deception for Computer Security Defense* (2004), головна цінність deception-підходу полягає не лише у виявленні порушника, а у впливі на його мислення та дії, що дозволяє оборонній стороні нав'язувати власний темп гри [6].

Таким чином, організація перестає бути лише об'єктом нападу, а трансформується у суб'єкта активної кіберконтроборони, здатного формувати та реалізовувати проактивну тактику у відповідь на загрози.

Перспективність deception-підходу

Deception-підхід пропонує новий рівень кіберзахисту, орієнтований не на блокування, а на введення зловмисника в оману. Створюється контрольоване середовище з фальшивими активами, де атакуючий взаємодіє з приманками, залишаючи сліди. Це дозволяє виявити загрозу до того, як вона досягне реальних систем.

Основу такого захисту складають об'єкти-імітації [7]:

honeypots - фальшиві сервери або сервіси,

honeytokens - підставні облікові дані або записи,

honeyfiles - фіктивні документи.

Будь-який доступ до них є підозрілим і викликає сповіщення.

Deception-система також може імітувати мережевий трафік і активність користувачів, створюючи переконливу картину справжньої інфраструктури. Це ускладнює розвідку, змушує атакуючого витрачати ресурси на аналіз фальшивих цілей і знижує ризики для реального середовища.

Завдяки цьому підхід Deception in Depth не лише виявляє загрози, а й впливає на дії зловмисника, збираючи цінну інформацію про його тактики, техніки й процедури (TTP) для подальшого аналізу.

Розглянемо основні аспекти методів Deception-підходу:

Проактивне виявлення. Deception дозволяє виявити зловмисника ще на ранньому етапі - під час розвідки, до початку активної атаки [7]. Наприклад, спроба входу через honeytoken відразу сигналізує про загрозу.

Зниження хибних спрацювань. Оскільки легітимні користувачі не взаємодіють із приманками, будь-який доступ до них вважається підозрілим. Це суттєво зменшує навантаження на аналітиків безпеки.

Збір аналітики. Усі дії атакуючого в середовищі deception фіксуються, що дозволяє глибоко проаналізувати його техніки (TTP) і підвищити ефективність майбутнього захисту.

Відволікання загроз**.** Зловмисник витрачає час на взаємодію з фальшивими об'єктами, що знижує ризик для реальних активів і дає додатковий час для реагування.

Сумісність з іншими системами**.** Deception не замінює класичні засоби захисту, а підсилює їх. Інтеграція з SIEM, IDS/IPS та EDR дозволяє будувати ефективну багаторівневу оборону.

Сучасний стан кіберзагроз вимагає впровадження нових підходів до захисту інформаційних систем - більш проактивних, точних і адаптивних. Однією з ключових проблем є велика кількість хибнопозитивних спрацювань, обмежена видимість внутрішньої активності зловмисників та затримки у виявленні складних атак.

У цьому контексті deception-підхід вирізняється як ефективне рішення, що створює контрольоване середовище для виявлення загроз, знижує навантаження на захисні системи й надає глибоку аналітику дій атакуючого. Завдяки точності спрацювань і можливості вивчення технік противника, deception-технології можуть стати ключовим елементом сучасної стратегії кіберзахисту - забезпечуючи не лише захист, а й активне виявлення та нейтралізацію загроз до їх ескалації.

1.2. Аналіз методів захисту

Системи виявлення вторгнень (IDS/IPS)

Системи виявлення (IDS) та запобігання (IPS) вторгнень є одними з основних технологій, які використовуються для моніторингу мережевого трафіку та виявлення потенційно небезпечних дій.

Принцип роботи - IDS/IPS базуються на сигнатурному аналізі, тобто порівнюють поточний трафік із базою відомих загроз. Також застосовуються поведінкові моделі, які дозволяють визначати аномалії - наприклад, незвичну кількість запитів до певного порту або нетипову поведінку користувача.

Переваги - Ці системи дозволяють у режимі реального часу виявляти відомі загрози, автоматично блокувати підозрілий трафік та генерувати сповіщення. IPS, крім того, можуть безпосередньо блокувати підозрілу активність, не чекаючи ручного втручання.

Обмеження - Основним недоліком є залежність від сигнатур - нові загрози або варіанти атак можуть не бути зафіксованими в базі даних, що призводить до пропуску атаки. Крім того, поведінкові алгоритми часто генерують велику кількість хибних позитивів, що створює навантаження на операторів SOC та може призводити до пропуску справжніх інцидентів через “шум” сповіщень.

Антивірусні рішення

Антивірусні системи орієнтовані на виявлення, ізоляцію та видалення шкідливого програмного забезпечення на рівні кінцевих точок.

Принцип роботи - Сучасні антивірусні рішення використовують комбінацію сигнатурного методу, евристичного аналізу, а також методи поведінкового моніторингу. Це дозволяє виявити як відомі, так і нові типи шкідливого ПЗ.

Переваги - Антивіруси здатні оперативно реагувати на загрози, блокуючи виконання шкідливих процесів і ізолюючи заражені файли. Вони активно оновлюються виробниками, що дозволяє вчасно реагувати на нові варіанти атак.

Обмеження - Як і IDS, антивіруси залежать від баз сигнатур. Зловмисники застосовують техніки поліморфізму або шифрування коду, що може обійти

сигнатурний аналіз. Крім того, у випадку складних атак, де використовується кілька етапів (наприклад, проникнення через соціальну інженерію, подальше горизонтальне розповсюдження в мережі), антивірус може виявити лише частину загрози.

Системи захисту кінцевих точок (EDR)

EDR-системи (Endpoint Detection and Response) забезпечують глибокий моніторинг активності на кінцевих пристроях (віртуальні машини, сервери, робочі станції) та реагування на загрози в режимі реального часу.

Принцип роботи - EDR постійно відстежує системні події, процеси, файлові дії, мережеві підключення та поведінку користувачів. При виявленні аномалій або відомих атак - фіксує інцидент, може блокувати активність та формувати звіт.

Переваги - Забезпечує глибоку видимість у процеси на хості, дозволяє виявляти навіть складні та багатостадійні атаки, включає інструменти аналізу інцидентів. Багато EDR-систем підтримують інтеграцію з MITRE ATT&CK [8].

Обмеження - Може потребувати значних ресурсів (CPU, RAM), потребує навченого персоналу для аналізу, часто є комерційними рішеннями з ліцензійними обмеженнями.

1.3. Аналіз засобів deception-захисту

Deception-рішення реалізують стратегію активної оборони, створюючи штучні цілі у вигляді пасток, фальшивих даних або імітованого корпоративного середовища. Це дозволяє виявити несанкціоновану активність ще до того, як вона вплине на реальні інформаційні системи. Взаємодія з такими обманними об’єктами є чітким індикатором потенційної загрози й дає можливість організаціям не лише вчасно реагувати, а й збирати цінну інформацію про інструменти, техніки та тактики атакуючого (TTP).

У цьому підрозділі розглянуто основні типи deception-компонентів, принципи їх роботи, переваги та обмеження, а також обґрунтовано доцільність їх застосування в реальних інфраструктурах.

Honeypots, Honeytokens, Honeyfiles, Honeynets

Honeypots - Ізольовані системи або сервери, які імітують реальні сервіси чи вразливі хости. Їх основне завдання - привернути увагу зловмисника, дозволити фіксувати дії в контрольованому середовищі та збирати розвіддані без ризику для реальної інфраструктури.

Honeytokens - Фальшиві облікові записи, дані, API-ключі або записи в базах даних. Вони не мають жодної практичної функції, тому будь-який доступ до них автоматично сигналізує про несанкціоновану активність або витік інформації.

Honeyfiles - Файли, що виглядають як важливі документи (наприклад, passwords.xlsx чи finance_2025.pdf). Їх відкриття або копіювання ініціює сповіщення або логування, фіксуючи факт потенційної атаки зсередини чи зовні.

Honeynets - Мережі, що складаються з декількох honeypot-систем, об’єднаних у повноцінну імітацію корпоративного середовища. Дає змогу вивчати латеральне переміщення (lateral movement) атакуючого, розкривати вектори атаки та глибше аналізувати поведінку в мережі.

Honeyports / Honeyservices - Вибірково відкриті порти або сервіси (наприклад, SSH, SMB, RDP), що створені виключно для виявлення сканування або спроб підключення. Дають змогу виявити зовнішню розвідку (reconnaissance) ще до початку реального вторгнення.

Системи виявлення аномалій у контексті deception

Системи виявлення аномалій (ADS - Anomaly Detection Systems), зокрема UEBA (User and Entity Behavior Analytics) та NDR (Network Detection and Response), ґрунтуються на побудові профілів нормальної поведінки користувачів, хостів і мережевих потоків. Такі профілі формуються за допомогою алгоритмів машинного навчання (ML), статистичного аналізу та обробки великих обсягів телеметрії.

У поєднанні з компонентами Deception in Depth, ці системи набувають додаткової точності та надійності. Аномалії, підтверджені взаємодією з обманними елементами (honeypots, honeytokens), автоматично класифікуються як події з високим ступенем довіри, що значно знижує кількість хибнопозитивних спрацювань і зменшує навантаження на команди безпеки.

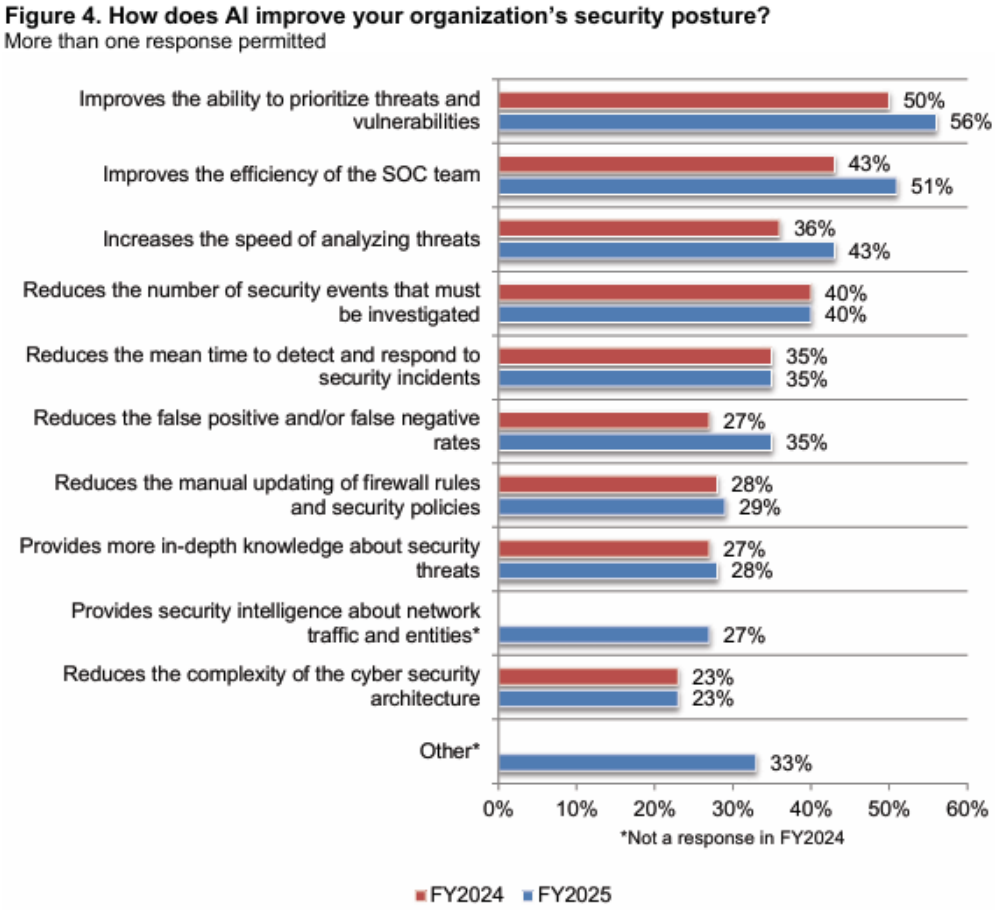

Рис. 1.3. Метрики ефективності AI

Автоматизовані open-source платформи deception-захисту

Відкриті платформи deception-захисту (open-source deception platforms) реалізують концепцію активної кібероборони шляхом розгортання фальшивих (віртуальних або програмних) ресурсів, які імітують реальні активи організації. Їх основне призначення - виявлення вторгнень, уповільнення атак, відволікання зловмисника від реальних систем та збір розвідувальних даних про техніки, тактики і процедури (TTP) атакуючих.

Ці інструменти охоплюють кілька напрямів: емуляція сервісів (honeypots), генерація фальшивих облікових даних (deceptive credentials), файлів (honeyfiles), токенів (honeytokens) та навіть побудова цілісних фальшивих мереж (honeynets).

Огляд ключових open-source рішень:

Одним із найбільш популярних легких honeypot-рішень є OpenCanary [9]. Цей інструмент призначений для емуляції поширених мережевих сервісів, таких як

SSH, HTTP, FTP, SMB, MySQL тощо, та генерує сповіщення при спробах сканування або підключення. До його переваг належать простота налаштування, сумісність із Linux, macOS, Raspberry Pi, а також підтримка syslog, JSON-формату та інтеграція з системами типу SIEM. Серед недоліків можна зазначити відсутність моделювання поведінки операційної системи та обмежений функціонал щодо глибокого аналізу атак.

Іншим прикладом є Cowrie [10] - honeypot високого рівня взаємодії, орієнтований на SSH- та Telnet-протоколи. Він імітує повноцінний доступ до Linux-системи з можливістю виконання команд і завантаження файлів. Основні переваги Cowrie полягають у високій деталізації журналювання дій зловмисника, підтримці інтеграції з аналітичними платформами ELK Stack і Splunk, а також здатності моделювати сценарії взаємодії з атакуючим. Недоліком є обмеження лише до SSH/Telnet та підвищені вимоги до ресурсів у порівнянні з low-interaction honeypot-системами.

Комплексним рішенням є платформа T-Pot [11], яка поєднує в собі кілька honeypot-модулів, зокрема Cowrie, Dionaea, Conpot, Mailoney та інші, й інтегрує їх з аналітичним стеком ELK у контейнерному середовищі Docker. Перевагами цієї платформи є наявність уніфікованого функціоналу «все в одному», зручна візуалізація атак та висока масштабованість. Разом з тим, до недоліків належать високі системні вимоги та складність початкового налаштування, що може ускладнити її використання для недосвідчених користувачів.

Ще одним прикладом є Honeyd - honeynet-движок, здатний створювати великі масиви фальшивих хостів з імітацією заданої поведінки, відкритих портів та TCP/IP-стеків. Honeyd ефективний для побудови симульованої корпоративної мережі з багатьма активами. Його основними перевагами є підтримка емуляції операційних систем та можливість роботи на одному фізичному сервері. Серед недоліків - потреба в ручному налаштуванні та відсутність регулярного оновлення проєкту, що обмежує його актуальність у довгостроковій перспективі.

Honeypot	Open source	Popular	Maintained	Linux-based	Fingerprinting
Kippo	Y	Y	Y	Y	Tool
Cowrie	Y	Y	Y	Y	Approach
(Thinkst) Open Canary	Y	Y	Y	Y	N?
Glastopf	Y	Not anymore?	N	Y	Approach
Conpot	Y	Y	N?	Y	Approach
Dionaea	Y	Y	N	Y	Y
(SANS) DShield	Y	N?	Y	Y	Unknown
MongoDB-HoneyProxy	Y	N?	Y	Y	Unknown
GasPot	Y	N?	Y	Y	Approach
Honeyd	Y	N	N	Y	Approach

Рис. 1.4. Порівняння відомих відкритих deception компонентів

Аргументація вибору, переваги та обмеження засобів deception-захисту

Вибір технологій для реалізації deception-захисту залежить від масштабів інформаційної системи, рівня технічної зрілості організації та доступних ресурсів. У той час як більшість комерційних платформ deception-класу орієнтовані на великі корпорації з розвиненими SOC-підрозділами, значна частина підприємств, зокрема малого та середнього бізнесу, не можуть дозволити собі подібні інвестиції.

У цьому контексті доцільним є впровадження ручного підходу до побудови deception-системи, що базується на використанні віртуальних машин і контейнеризованих сервісів**.** Такий підхід дає змогу:

самостійно створювати honeypot-и та інші приманки;

повністю контролювати архітектуру та поведінку фальшивих об’єктів;

адаптувати середовище до конкретних сценаріїв загроз;

розгортати deception-рішення навіть у обмеженому середовищі (локальна мережа, навчальна лабораторія, сегмент підприємства).

Переваги ручного підходу:

Низька вартість впровадження: використання безкоштовного ПЗ (VirtualBox, Docker, Cowrie, OpenCanary, T-Pot) дозволяє побудувати повноцінну deception-систему без додаткових ліцензійних витрат.

Гнучкість і контроль: адміністратор самостійно конфігурує типи пасток, поведінку, рівень деталізації логування тощо.

Освітня та дослідницька цінність: ручне створення honeypot-ів сприяє

кращому розумінню механізмів атак та побудови кіберзахисту.

Ізоляція обманних елементів: усі deception-компоненти розміщуються у віртуалізованому середовищі, що гарантує безпеку основної інфраструктури.

Обмеження та виклики:

Витрати часу на налаштування: ручна конфігурація потребує технічних знань і більше часу в порівнянні з комерційними платформами.

Обмежена автоматизація**:** оновлення, адаптація до нових загроз і ротація пасток виконуються вручну, що знижує оперативність.

Високі вимоги до компетентності**:** адміністратор має розуміти як принципи мережевої безпеки, так і основи взаємодії сервісів, контейнерів, журналювання тощо.

Масштабованість**:** у великих інфраструктурах ручний підхід потребує додаткових засобів автоматизації (наприклад, CI/CD для оновлення Docker-образів).

Висновок до розділу 1

Ручне впровадження deception-систем - це реалістичний, економічно ефективний і гнучкий шлях до створення багаторівневого обманного захисту, особливо у навчальних, експериментальних та малих виробничих середовищах. Такий підхід дає змогу імітувати реальні сценарії атак, вивчати поведінку зловмисників та формувати практичні навички побудови активного захисту. У контексті даного дослідження саме ця стратегія обрана як базова модель реалізації Deception in Depth - через віртуальну машину з контейнеризованими honeypot-сервісами, що дозволяє досягти цілей дослідження без залучення дорогих технологій.

2 ДОСЛІДЖЕННЯ МЕТОДИКИ DECEPTION IN DEPTH

2.1. Архітектура та налаштування deception-рішення

Однією з ключових переваг методології Deception in Depth є її інтеграційна природа: вона не замінює, а доповнює традиційні засоби кіберзахисту, такі як SIEM-системи, EDR, IDS/IPS, фаєрволи тощо. Це дозволяє побудувати багаторівневу, адаптивну та проактивну оборонну модель, що дозволяє не лише виявляти атаки, а й впливати на поведінку зловмисника у режимі реального часу.

Концептуальна суть Deception in Depth

Головна ідея підходу Deception in Depth полягає у створенні дезорієнтаційного середовища, в якому зловмисник взаємодіє не з реальними об’єктами інфраструктури, а з достовірно змодельованими, але фальшивими ресурсами - такими як сервери, облікові записи, мережі, файли чи ключі доступу.

Таке середовище виконує кілька критично важливих функцій. Воно уповільнює розвиток атаки, затримуючи та дезорієнтуючи зловмисника. Водночас воно відволікає увагу від реальних активів, підставляючи фіктивні об'єкти, що виглядають привабливо. Крім того, система deception дозволяє збирати детальну аналітичну інформацію про інструменти, тактики й маршрути переміщення атакуючого (TTP), а також забезпечує виявлення загрози ще до того, як буде завдано реальної шкоди.

Архітектурна побудова deception-рішення

Для реалізації Deception in Depth створюється окрема віртуалізована архітектура, яка, хоча й інтегрована з основною мережею, залишається логічно та фізично ізольованою. Це забезпечує безпеку, керованість і масштабованість deception-середовища.

Ключові характеристики архітектури:

Віртуалізація (через гіпервізори або контейнеризацію) дозволяє швидко розгортати та відновлювати компоненти deception;

Ізоляція - використання окремих VLAN, віртуальних комутаторів, або спеціальних зон безпеки (sandbox), які унеможливлюють перехід до продуктивних сегментів;

Інтеграція з SIEM/EDR - події з honeypot-ів автоматично направляються до систем моніторингу для аналізу та реагування;

Оркестрація та масштабування - використання автоматизації (Ansible, Terraform, CI/CD пайплайни) для керування фальшивими активами.

Гнучкість та керованість

Однією з ключових переваг сучасного deception-середовища є його динамічна налаштовуваність. Адміністратор системи має змогу створювати фальшиві сервери, служби, облікові записи користувачів, ключі доступу, а також задавати логіку взаємодії між ними відповідно до заданого сценарію. Крім того, підтримується налаштування часових шаблонів активності, таких як планові cron-запити чи генерація умовного «живого» трафіку, що створює ілюзію повноцінного функціонування системи.

Архітектура deception-середовища може оперативно змінюватися у відповідь на нові загрози або зміну ризик-профілю, без потреби у фізичному втручанні. Такий підхід дозволяє не лише зменшити витрати на підтримку інфраструктури, а й значно скоротити час реагування, підвищуючи адаптивність системи безпеки до реальних бізнес-процесів.

Взаємодія з основною інфраструктурою

Deception-рішення інтегрується в інформаційне середовище організації таким чином, щоб залишатися невидимим для легітимних користувачів, викликати довіру у потенційного зловмисника та не впливати на роботу бізнес-процесів і продуктивних служб.

Досягається це шляхом використання реалістичних шаблонів іменування активів, інтеграції з інфраструктурними службами, такими як Active Directory, DNS і DHCP, а також завдяки імітації типової мережевої поведінки. Зокрема, застосовуються підроблені ARP-запити, DNS-резолюції та файлові звернення, що створює переконливу ілюзію справжньої інформаційної системи для атакуючого.

Очікувані переваги

Запровадження deception-архітектури сприяє значному зменшенню загальної поверхні атаки в інформаційній системі. Такий підхід дозволяє виявляти навіть невідомі типи атак, включаючи zero-day загрози, які часто залишаються поза межами дії класичних засобів захисту. Завдяки високій точності виявлення, знижується навантаження на аналітиків центрів обробки інцидентів (SOC), оскільки кількість хибнопозитивних спрацювань помітно скорочується. Крім того, використання deception-компонентів дає змогу отримувати цінні форензічні артефакти, такі як шкідливі файли, журнали подій або сесії пам’яті, що мають високу аналітичну цінність. У сукупності це підвищує загальний рівень кіберстійкості організації та її здатність ефективно протидіяти сучасним загрозам.

Логічна структура розгортання

Архітектура системи обманного захисту (deception) умовно поділяється на три основні рівні:

1. Рівень виявлення - включає елементи, що імітують реальні системи або дані й створені для приваблення зловмисників. Сюди належать *honeypots*, *honeytokens*, *honeyservices*, доступ до яких одразу генерує сигнал безпеки.

2. Рівень збору та аналітики - охоплює інструменти логування, кореляції подій та передавання даних у централізовані платформи моніторингу (наприклад, SIEM, EDR, SOAR).

3. Рівень управління - забезпечує централізоване керування deception-інфраструктурою: створення, модифікацію, моніторинг та інтеграцію з іншими засобами кібербезпеки.

Рис. 2.1. Логічна архітектура

Центральний компонент: Віртуальна машина-ізолятор

Для підвищення гнучкості та безпеки розгортання приманок доцільно використовувати віртуальну машину-ізолятор, яка слугує базовим середовищем для створення та динамічного управління обманними активами. Така машина працює під керуванням гіпервізора, зокрема Proxmox, VirtualBox, VMware, KVM або Xen, і виконує низку ключових функцій.

По-перше, забезпечується повна ізоляція deception-компонентів від продуктивної інфраструктури, що мінімізує ризики несанкціонованого впливу на основні ресурси. По-друге, система дозволяє динамічно розгортати різні типи приманок - файлові, мережеві, облікові - у контрольованому середовищі. Також важливою є можливість масштабування deception-інфраструктури без втручання у фізичну або хмарну архітектуру підприємства. Крім того, віртуальна машина-ізолятор забезпечує централізоване логування та аналітичну обробку активності всередині ізольованого середовища, що дозволяє оперативно виявляти та аналізувати підозрілу поведінку зловмисника.

Види розміщення обманних елементів

У сучасних системах обманного захисту можна виділити три основні варіанти реалізації приманок:

1. На фізичному пристрої - приманка встановлюється безпосередньо на

робочий комп’ютер користувача або інший фізичний пристрій (наприклад, термінал, IoT-пристрій). Це дозволяє досягти максимальної автентичності, проте ускладнює адміністрування. Такі рішення застосовуються, наприклад, у системі *Attivo Endpoint Deception*, яка розміщує honeytoken-и безпосередньо на кінцевих точках.

2. На віртуальній машині - приманка запускається як повноцінна віртуальна система, що імітує реальний сервер або клієнт. Такий підхід забезпечує високий ступінь гнучкості, дозволяє використовувати технології інтроспекції гіпервізора (наприклад, LibVMI) для безагентного моніторингу активності у пам’яті, а також спрощує автоматизацію оновлення та тестування.

3. У контейнеризованому середовищі - використовується інфраструктура на базі Docker або Kubernetes, що дозволяє швидко розгортати honeypot-и, наприклад *Cowrie* (SSH-пастка) або платформи на кшталт *T-Pot*, яка об’єднує понад 20 типів honeypot-ів у контейнерах. Цей підхід відзначається високою швидкістю масштабування, але потребує додаткових засобів для симуляції повноцінного середовища операційної системи.

Переваги підходу з віртуальним ізолятором

Використання віртуального ізолятора як базового елементу архітектури deception-системи забезпечує низку важливих переваг. По-перше, це гнучкість у розгортанні приманок і зміні сценаріїв захисту відповідно до актуальних загроз. По-друге, така архітектура є масштабованою, що дозволяє швидко адаптувати її під змінні умови або розширити в разі зростання потреб.

Крім того, ізоляція середовища значно підвищує загальний рівень безпеки, оскільки виключає ризики неконтрольованого впливу на продуктивну інфраструктуру. Завдяки централізованому моніторингу та можливості точного аналізу дій атакуючого, система забезпечує ефективне виявлення загроз у контрольованому середовищі без порушення основних бізнес-процесів.

У результаті інтеграція ізолятора в ядро deception-архітектури дає змогу створити керовану, адаптивну та надійну платформу обманного захисту, яка суттєво посилює інформаційну безпеку організації.

Конфігурація та типи honeypots

Для побудови ефективної системи Deception in Depth застосовуються різні типи honeypot-ів, які відрізняються між собою за рівнем взаємодії зі зловмисником, можливостями налаштування та глибиною аналітики отриманих даних. Залежно від ступеня реалізму й функціонального навантаження, honeypot-и поділяються на два основні типи - low-interaction та high-interaction.

Low-interaction honeypot-и є легкими емуляційними системами, які відповідають на базові запити або імітують окремі сервіси, як-от SSH, HTTP чи FTP. Їхньою ключовою перевагою є мінімальне споживання ресурсів і низький ризик компрометації, оскільки зловмисник фактично не взаємодіє з повноцінною операційною системою. Такі рішення легко масштабуються, можуть працювати у контейнерах і часто використовуються на периферійних пристроях або в IoT-середовищах. Серед найбільш поширених прикладів - Honeyd (для симуляції мережевих хостів із TCP/IP-стеком), Cowrie (SSH/Telnet-пастка з логуванням команд) та Glastopf (web-імітація з уразливостями).

High-interaction honeypot-и, навпаки, створюються на основі повноцінних віртуальних машин або контейнеризованих середовищ з розгорнутими сервісами, що дозволяє зловмиснику здійснювати реальні дії. Це відкриває можливість для детального вивчення тактик атаки, включно з переміщенням мережею (lateral movement), підвищенням привілеїв або взаємодією з командно-контрольними серверами. Такі системи потребують ретельної ізоляції від продуктивної мережі та часто інтегруються з технологіями моніторингу, як-от VMI або eBPF. До типових рішень цього класу належать платформа T-Pot (яка поєднує понад 20 honeypot-сервісів), Dionaea (для виявлення шкідливого ПЗ через SMB/FTP) та Conpot (імітація інфраструктури промислових систем управління).

Такий поділ дозволяє гнучко обирати тип honeypot-ів залежно від технічних можливостей організації та цілей дослідження чи захисту.

Ключові аспекти конфігурації honeypot-ів

Для забезпечення ефективного функціонування deception-системи під час налаштування honeypot-компонентів необхідно дотримуватись низки критично

важливих вимог. Насамперед, іменування та адресація обманних елементів мають відповідати внутрішній структурі підприємства. Наприклад, назви типу «FIN-DB-01» або «HR-FS-04» можуть імітувати реальні ресурси фінансового відділу чи відділу кадрів, що підвищує довіру атакуючого до фальшивого об’єкта.

Також доцільною є інтеграція deception-компонентів з доменною структурою, зокрема Active Directory. Це дозволяє створювати фальшиві облікові записи, політики доступу та службові розклади, що значно підвищує реалістичність середовища та ймовірність взаємодії зловмисника з приманкою.

Усі дії, здійснені в межах honeypot-ів, повинні бути ретельно зафіксовані та передані до централізованих систем моніторингу, таких як SIEM або EDR. Для high-interaction honeypot-ів доцільно також використовувати журнали аудиту операційної системи та мережевий моніторинг із застосуванням спеціалізованих рішень, наприклад Zeek або Suricata.

Окрему увагу слід приділити ізоляції deception-елементів від продуктивного середовища. Жоден обманний компонент не повинен мати реального доступу до критичної інфраструктури. Ізоляція досягається шляхом VLAN-сегментації, налаштувань гіпервізора або використання віртуальних комутаторів. У разі активації пастки має бути забезпечена можливість миттєвого розриву мережевих з’єднань для запобігання потенційному ризику.

Побудова та роль honeynets

Для моделювання складного та багаторівневого середовища з метою виявлення цілеспрямованих і багатоступеневих атак використовується підхід honeynet - мережа взаємопов’язаних фальшивих хостів і служб, які спільно імітують повноцінну корпоративну IT-інфраструктуру. Honeynets виступають ядром просунутої deception-архітектури, дозволяючи моделювати не окремий хост чи сервіс, а всю логіку взаємодії в рамках організації.

Основні завдання honeynets:

Реалістичне моделювання IT-інфраструктури підприємства: Honeynet містить фальшиві сервери баз даних, поштові сервери, файлові системи, контролери домену тощо, які структурно і функціонально відповідають

справжньому середовищу організації.

Імітація логіки взаємодії між компонентами: Наприклад, web-сервер може «викликати» базу даних, файловий сервер - взаємодіяти з псевдо-користувачами через SMB, що створює правдоподібну картину мережевої поведінки.

Виявлення складних сценаріїв атак: Honeynet дозволяє фіксувати такі дії зловмисників, як lateral movement, privilege escalation, persistence, або встановлення командно-контрольного зв’язку (C2).

Технічна реалізація honeynets

Для ефективного функціонування deception-системи під час конфігурації honeypot-компонентів необхідно дотримуватись ряду ключових вимог. Перш за все, іменування та адресація приманок мають відповідати внутрішнім стандартам підприємства. Наприклад, назви на кшталт «FIN-DB-01» чи «HR-FS-04» імітують реальні активи відповідних підрозділів, що сприяє створенню довірливого середовища для зловмисника.

Важливим елементом є інтеграція deception-елементів із доменною інфраструктурою, зокрема Active Directory. Це дає змогу додавати фальшиві облікові записи, налаштовувати групові політики доступу та створювати розклади служб, що посилює реалістичність симуляції.

Усі дії, що здійснюються в середовищі honeypot-ів, повинні бути ретельно зафіксовані та передані до централізованих систем моніторингу, таких як SIEM або EDR. У випадку high-interaction honeypot-ів доцільно також залучати механізми аудиту операційної системи та засоби мережевого аналізу, зокрема Zeek або Suricata.

Особливу увагу слід приділити ізоляції обманного середовища від продуктивної інфраструктури. Приманки не повинні мати реального доступу до критичних ресурсів. Ізоляція забезпечується за допомогою VLAN-сегментації, налаштувань гіпервізора або віртуальних мережевих комутаторів. У разі виявлення активності має бути передбачена можливість негайного розриву мережевого з’єднання для мінімізації ризиків.

2.2. Основні функції та принципи роботи deception-систем

Системи обману (*deception systems*) функціонують за принципом створення віртуального середовища, яке імітує справжню інфраструктуру організації, але є ізольованим, контрольованим і цілеспрямовано вразливим для зловмисників. Їх основна мета - виявити зловмисника ще на ранніх стадіях проникнення, до того як буде завдано шкоди реальним активам**.** У цьому підрозділі розглядаються ключові функції deception-систем, рівні їх застосування, а також особливості інтеграції з іншими засобами захисту.

Основні функції deception-систем

Deception-системи виконують низку функцій, які забезпечують їхню ефективність у багаторівневому підході до кібербезпеки:

1. Імітація критичних об'єктів**:** створення фальшивих серверів, БД, користувацьких облікових записів, конфіденційних документів, API-ключів тощо.

2. Прихованість**:** ці системи не генерують мережевого трафіку, доки не відбудеться спроба взаємодії - це робить їх невидимими для легітимних користувачів і привабливими для зловмисників.

3. Алертування в реальному часі**:** будь-яка взаємодія зі штучним об’єктом (honeypot або honeytoken) трактуються як підозріла активність, і система одразу формує повідомлення про інцидент.

4. Збір розвідувальної інформації: система записує всі дії зловмисника, включаючи введені команди, методи сканування, спроби використання вразливостей, що дозволяє проаналізувати TTP (Tactics, Techniques, Procedures).

5. Відволікання та уповільнення атакуючого**:** створення "шуму" в інфраструктурі, що змушує зловмисника витрачати час і ресурси на взаємодію з фіктивними системами.

Рівні захисту deception-систем

Deception-рішення можуть бути впроваджені на різних рівнях корпоративної інфраструктури, що дозволяє формувати багатошарову модель захисту та посилювати загальну кіберстійкість системи.

На мережевому рівні застосовується симуляція фальшивих серверів, маршрутизаторів, комутаторів та інших пристроїв у локальному сегменті. Honeypot-и, розгорнуті на цьому рівні, реагують на спроби сканування портів, підключення до типових сервісів, таких як RDP, SSH або Telnet. Це дає змогу виявляти ознаки горизонтального переміщення (lateral movement) або розвідки (reconnaissance) ще на початкових етапах атаки.

На файловому рівні використовуються так звані honeytokens - спеціально підготовлені файли-приманки з назвами на кшталт «credentials.xlsx», «договір_2023.docx» або «фінзвіт.pdf». Такі файли розміщуються у загальнодоступних файлових системах і мають вбудовані тригери, що активуються при відкритті або копіюванні. Цей підхід дозволяє виявляти спроби несанкціонованого внутрішнього доступу або зловмисні дії з боку інсайдерів.

На прикладному рівні реалізується симуляція взаємодії з веб-застосунками, базами даних та API-інтерфейсами. Наприклад, можуть бути створені фіктивні вебсторінки з формами авторизації або API-ендпоінти, які реагують на запити, характерні для зловмисників. Це дозволяє виявляти спроби атак типу SQL-ін’єкцій, XSS, brute-force, directory traversal та інших прикладних загроз.

Таким чином, поєднання deception-компонентів на різних рівнях дозволяє створити глибоку, багатовекторну систему виявлення та реагування на загрози.

Взаємодія з іншими засобами безпеки

Deception-системи не є ізольованими рішеннями - їх ефективність суттєво зростає за умови тісної інтеграції з наявною інфраструктурою кіберзахисту. Одним із ключових напрямів такої інтеграції є взаємодія з системами управління подіями та інформацією безпеки (SIEM). У цьому випадку deception-компоненти надсилають журнали подій до SIEM-платформи, де ці дані проходять централізований збір, аналіз та кореляцію з іншими джерелами. Наприклад, подія взаємодії з honeypot-ом може співпадати в часі зі спробами входу в Active Directory або з переміщенням атакуючого мережею, що дозволяє реконструювати повну логіку атаки (attack chain) та ініціювати автоматизовані сценарії реагування.

Інтеграція з рішеннями класу EDR або XDR забезпечує додатковий рівень

поведінкового аналізу на кінцевих точках. Події, зафіксовані в середовищі deception, можуть слугувати тригером для запуску полювання на загрози або карантину сесій в EDR-системах. Наприклад, виконання зловмисного скрипта в honeynet середовищі може автоматично активувати відповідні дії на реальному хості.

Системи SOAR (Security Orchestration, Automation and Response) дозволяють автоматизувати реакцію на інциденти, зафіксовані deception-інструментами. У таких випадках може бути реалізовано автоматичне блокування IP-адреси, ізоляція користувацької сесії або запуск попередньо визначеного сценарію розслідування, що значно знижує навантаження на аналітиків та прискорює цикл реагування.

Ще одним важливим напрямом є інтеграція deception-компонентів із системами керування ідентичностями, такими як Active Directory або IAM-рішення. У цьому контексті honeytoken-и можуть бути вбудовані безпосередньо в структуру домену, і кожна спроба їх використання - наприклад, авторизація або запит службового квитка - вказує на намагання компрометації або наявність внутрішньої загрози.

Таким чином, глибока інтеграція deception-рішень із ключовими системами безпеки забезпечує не лише раннє виявлення загроз, а й можливість автоматизованої, контекстно-орієнтованої відповіді на інциденти.

2.3. Юридичні та етичні аспекти застосування Deception in Depth

Хоча методологія Deception in Depth здатна значно підвищити рівень кіберстійкості організації, її застосування супроводжується низкою юридичних та етичних викликів**,** які необхідно враховувати під час розгортання та експлуатації систем обманного захисту.

Законодавчі обмеження

Застосування honeypot-ів та інших обманних технологій у сфері кібербезпеки не є прямо забороненим у більшості юрисдикцій, однак їх використання супроводжується низкою правових нюансів, які можуть створювати потенційні

ризики.

Перш за все, у разі, коли в процесі взаємодії зловмисника із deception-компонентами здійснюється збір даних, що прямо або опосередковано можуть дозволити ідентифікувати особу, виникає необхідність дотримання вимог законодавства про захист персональних даних. Наприклад, у країнах Європейського Союзу діє Загальний регламент із захисту даних (GDPR), а в Україні - Закон України «Про захист персональних даних» [12]. У цьому контексті рекомендовано впроваджувати механізми анонімізації або псевдонімізації журналів подій, щоб зменшити ризики порушення прав суб’єктів даних.

Ще одним аспектом є ризик правової оцінки дій з боку органів влади як провокації або перевищення повноважень. У деяких юрисдикціях навмисне створення фальшивих ресурсів (deception assets), які вводять зловмисника в оману, може бути розцінене як спонукання до вчинення правопорушення (entrapment), особливо якщо фіксація дій відбувається без юридичного обґрунтування. У зв’язку з цим важливо обмежувати доступ до deception-систем лише внутрішніми мережами організації або використовувати попереджувальні банери про моніторинг, які забезпечують інформовану згоду користувача - це положення підтримуються також у Законі України «Про захист інформації в інформаційно-телекомунікаційних системах» [13].

Окрему увагу слід приділити питанням зберігання та аналізу зразків шкідливого програмного забезпечення. У деяких країнах чинне законодавство передбачає відповідальність не лише за розповсюдження, а й за володіння malware - зокрема, без відповідного дозволу або цільового обґрунтування. Тому обробку таких об'єктів доцільно проводити в ізольованому середовищі з технічними й адміністративними заходами захисту, що відповідають локальним нормативно-правовим актам [13].

Етичні питання застосування deception-підходу

Окрім правових аспектів, впровадження deception-рішень супроводжується низкою етичних дилем, які організація повинна ретельно враховувати при побудові системи захисту.

Першою з них є питання прозорості щодо легітимних користувачів. Наприклад, розміщення honeytoken-ів у внутрішньому середовищі - таких як фальшиві облікові записи чи доступи до баз даних - може створити ризик випадкової взаємодії з ними з боку звичайних співробітників. Це здатне спричинити непорозуміння або навіть помилкові підозри. Етичний підхід у такому випадку полягає в чіткій сегментації прав доступу, обмеженні можливості взаємодії з обманними елементами для звичайних користувачів, а також у регулярному інформуванні персоналу про політику інформаційної безпеки без розкриття конкретних деталей щодо існування deception-компонентів.

Другим важливим питанням є право на приватність, особливо коли йдеться про розслідування внутрішніх загроз. Навіть у разі підозри на зловмисну активність з боку працівника, організація повинна дотримуватися принципів мінімального втручання та пропорційності. Збір доказової інформації, що здійснюється через deception-механізми, повинен бути обґрунтованим, документованим і проводитися виключно в межах затверджених процедур реагування на інциденти.

Ще одна етична межа стосується балансу між захистом і провокацією. Хоча суть deception-технологій полягає у введенні зловмисника в оману, їхнє використання не повинно перетворюватися на активне провокування до злочинних дій. Метою впровадження є не ескалація конфлікту, а запобігання загрозам шляхом фіксації, затримки та аналізу дій атакуючого.

Окрему увагу слід приділити використанню зібраних даних. Телеметрія та аналітична інформація, отримана через honeypot-и та інші компоненти deception-системи, мають застосовуватися виключно для цілей забезпечення безпеки, навчання персоналу та проведення внутрішніх розслідувань. Навіть у випадку їх анонімізації, передача таких даних третім сторонам або публікація повинні здійснюватися лише після ретельної юридичної оцінки.

Таким чином, успішне впровадження deception-технологій потребує не лише технічної обґрунтованості, а й дотримання етичних норм, що гарантують повагу до прав користувачів та прозорість дій організації.

Висновок до розділу 2

Впровадження концепції Deception in Depth у корпоративному середовищі повинно ґрунтуватися не лише на технічній компетентності, а й на чіткому усвідомленні правових та етичних меж допустимої діяльності. Для цього доцільно ще на етапі проєктування ініціативи провести юридичний аудит, що дозволить виявити потенційні зони ризику, пов’язані з обробкою персональних даних, правом на приватність та законністю моніторингу.

Окрім того, організація має визначити внутрішні правила щодо збору, зберігання та використання даних, які генеруються під час роботи deception-систем. Обов’язковою складовою є також формалізація політики етичного застосування таких технологій, що передбачає принципи пропорційності, мінімального втручання та недопущення зловживання можливостями контролю.

Усі вищезазначені аспекти мають бути інтегровані у загальну стратегію інформаційної безпеки організації, забезпечуючи узгодженість дій між технічними, адміністративними та правовими підрозділами.

Лише за умови комплексного підходу та дотримання встановлених стандартів deception-технології можуть стати не лише ефективним, а й легітимним, етичним і соціально прийнятним інструментом кібероборони.

3 РОЗРОБКА ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ DECEPTION IN DEPTH ДЛЯ ЗАХИСТУ ІНФРМАЦІЙНОЇ СИСТЕМИ ОРГАНІЗАЦІЇ

3.1. Методика впровадження багаторівневого обманного захисту (Deception in Depth)

Ефективне застосування концепції Deception in Depth потребує не лише теоретичного розуміння, а й чітко структурованої методики її реалізації. Побудова обманного середовища повинна базуватись на системному підході до розгортання компонентів, що охоплюють різні рівні IT-інфраструктури, а також на їх інтеграції в наявну архітектуру безпеки організації.

Рівнева організація deception-компонентів

Запровадження багаторівневого обманного захисту (*Deception in Depth*) доцільно реалізовувати через три ключові функціональні напрями, що охоплюють як технічні, так і поведінкові аспекти інформаційної безпеки.

На інформаційному рівні застосовуються фальшиві логічні ресурси, зокрема облікові записи, псевдоконфіденційні файли, симульовані доступи до баз даних, які містять вбудовані маркери відстеження (honeytokens). Взаємодія з цими об’єктами дає змогу виявити несанкціоновану активність, особливо на етапі внутрішнього доступу до критичних даних.

На інфраструктурному рівні розгортаються honeypot-сервери, які імітують роботу реальних сервісів, таких як FTP, SSH або SMB. Такі обманні вузли можуть бути інтегровані у внутрішні сегменти мережі або в зону демілітаризованого доступу (DMZ). Їх використання дозволяє фіксувати дії потенційного зловмисника ще на стадії розвідки, сканування портів або спроби первинного проникнення.

На організаційному рівні реалізується моделювання цифрової поведінки співробітників, що включає генерацію умовної ділової активності - наприклад, створення фальшивих листів з псевдофайлами або повідомлень із посиланнями на документи, захищені механізмами honeytoken. У такий спосіб deception

вбудовується безпосередньо в бізнес-логіку організації, дозволяючи виявляти не лише технічні загрози, а й цільові атаки соціального спрямування.

Комплексне охоплення цих трьох напрямів забезпечує системний підхід до впровадження deception-архітектури, підвищуючи її ефективність та адаптивність у сучасному середовищі кіберзагроз.

Аналіз існуючої IT-інфраструктури та визначення вразливих точок

Першим і надзвичайно важливим етапом впровадження багаторівневої deception-архітектури є всебічний аналіз поточної IT-інфраструктури організації. Без глибокого розуміння її архітектури, розміщення активів, логіки взаємодії компонентів і потенційних векторів атаки неможливо забезпечити коректне розгортання обманних елементів та їх ефективну інтеграцію у загальну стратегію кіберзахисту.

Інвентаризація активів

Початковим кроком виступає повна інвентаризація компонентів IT-середовища. До неї входить:

кінцеве обладнання (робочі станції, ноутбуки, мобільні пристрої),

серверна інфраструктура (бази даних, файлові та прикладні сервери, DNS, DHCP),

мережеве обладнання (комутатори, маршрутизатори, міжмережеві екрани),

віртуалізовані й хмарні середовища (VM, Docker),

інформаційні активи (системи документообігу, бази даних, облікові дані користувачів).

Результатом є створення топологічної карти інфраструктури, яка відображає логічні зв’язки та можливі траєкторії переміщення в мережі.

Визначення критичних вузлів

Наступним етапом є ідентифікація критично важливих елементів, які становлять найбільший інтерес для потенційного зловмисника:

сервери, що містять конфіденційні або персональні дані (HR, фінанси, юридичний відділ),

системи керування доступом (Active Directory, LDAP),

засоби віддаленого доступу (VPN, RDP, SSH),

сервіси, відкриті до зовнішнього середовища (вебсервери, REST API, DMZ-сегменти).

Ці елементи мають стати пріоритетними точками розміщення deception-компонентів, оскільки їх компрометація призводить до найбільш суттєвих наслідків.

Аналіз потенційних векторів загроз

На основі інвентаризації та визначених критичних активів виконується моделювання можливих векторів проникнення. Аналіз охоплює:

наявні вразливості периметру (відкриті порти, застаріле ПЗ),

невідповідності у політиках доступу та надлишкові привілеї,

слабку сегментацію мережі та відсутність міжсегментного контролю.

Для виявлення технічних недоліків доцільно використовувати інструменти аудиту безпеки, такі як Nessus, OpenVAS або Qualys, а також аналіз логів SIEM-систем для виявлення підозрілої активності.

Комплексна оцінка інфраструктури є необхідною умовою для проєктування логічно обґрунтованої deception-архітектури, яка буде не лише технічно ефективною, а й адаптованою до реальних ризиків і особливостей конкретної організації.

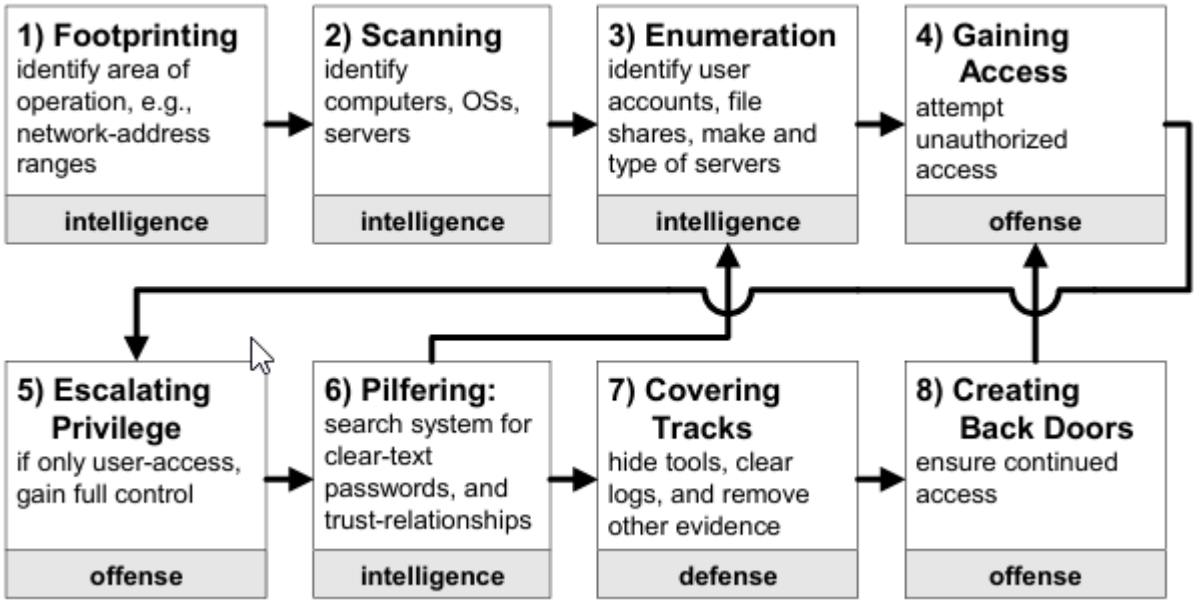

Рис. 3.1. Модель процесу атаки на організацію.

Формування deception-стратегії на основі аналізу

За результатами аналізу формується карта розміщення обманних елементів:

У внутрішніх сегментах розміщуються фальшиві облікові записи, документи, шляхи до баз даних.

У зонах периметру - легковагі honeypot-и для виявлення сканування й атак типу brute-force.

У сегментах з привілейованим доступом - високореалістичні приманки з логікою взаємодії.

Таким чином, забезпечується проактивний захист найбільш уразливих точок IT-інфраструктури, створюється «пастка» для зловмисника на його шляху до реальних активів і значно підвищується точність виявлення загроз.

Вибір типів deception-компонентів (honeypots, honeytokens, honeyfiles)

Після завершення аналізу вразливостей IT-інфраструктури наступним етапом у впровадженні стратегії *Deception in Depth* є вибір оптимального набору обманних компонентів. Залежно від характеру загроз, архітектурної моделі системи та доступних ресурсів організація може комбінувати різні типи deception-елементів, кожен з яких виконує специфічну функцію в структурі захисту.

Honeypots

Honeypot являє собою штучно створений мережевий вузол, що імітує поведінку реального пристрою або сервісу. Його основне призначення - привернення уваги зловмисника, фіксація дій, а також їх подальший аналіз.

Ці компоненти можуть бути розгорнуті як у зоні DMZ (для імітації зовнішніх сервісів), так і у внутрішній мережі (для виявлення вторгнення або дій інсайдера).

Honeytokens

Honeytoken - це логічна пастка, яка представлена у вигляді фіктивного об’єкта, що не використовується в реальних бізнес-процесах. Серед типових реалізацій: фальшиві облікові записи, API-ключі, записи в базах даних або електронні адреси. Будь-яка взаємодія з такими об’єктами автоматично вважається підозрілою.

Перевагами honeytokens є простота впровадження, відсутність потреби у

додатковому обладнанні та висока точність спрацьовування. Їх ефективність особливо проявляється у середовищах з великою кількістю внутрішніх користувачів.

Honeyfiles

Honeyfiles - це псевдодокументи, створені для виявлення несанкціонованого доступу до файлових ресурсів. Такі файли можуть містити унікальні маркери відстеження (наприклад, спеціальні URL, приховані скрипти) і автоматично генерують подію в системі моніторингу при спробі доступу.

Типові сценарії їх розміщення - загальні файлові каталоги, віртуальні файлові сервери, корпоративні мережеві диски. Ці компоненти особливо ефективні у виявленні інсайдерських загроз або атак із підвищенням привілеїв.

Критерії вибору deception-компонентів

Формування ефективної моделі обманного захисту передбачає врахування таких чинників:

характер очікуваних загроз (масові атаки, APT, інсайдери);

фізичне або логічне розміщення точки контролю (периметр, внутрішня мережа, кінцева точка);

технічні можливості для обслуговування і моніторингу;

потреба в глибокому аналізі поведінки зловмисника.

Збалансоване поєднання honeypot-ів, honeytokens і honeyfiles дозволяє охопити як зовнішні, так і внутрішні вектори атак, створивши гнучку, адаптивну та стійку до сучасних загроз систему обманного захисту.

Побудова логіки взаємодії обманних елементів у мережі

Для забезпечення повноцінного функціонування багаторівневого обманного захисту недостатньо розгорнути окремі приманки у різних сегментах IT-інфраструктури. Ефективність такого підходу досягається лише за умови створення цілісного, керованого середовища, яке імітує справжню архітектуру корпоративної мережі та відтворює типові бізнес-процеси. Мова йде про побудову так званої *обманної мережі* - логічно пов’язаної системи honeypot-ів, honeytoken-ів і допоміжних служб, що генерують переконливу симуляцію звичайної мережевої активності.

Принципи побудови логіки взаємодії

Одним із ключових елементів цієї моделі є моделювання зв’язків між окремими deception-компонентами. Наприклад, фіктивний вебсервер може регулярно надсилати запити до псевдо-бази даних, облікові записи - «мати доступ» до інших фальшивих вузлів, а файли-приманки - містити дані, що ведуть до інших об’єктів пастки. Така взаємодія створює ефект життєздатної системи, що підвищує рівень довіри атакуючого до обманної інфраструктури.

Важливим елементом є імітація активності користувачів. Регулярні логіни до системи (SSH, RDP), запуск процесів, звернення до файлових ресурсів чи внутрішніх API дозволяють створити відчуття того, що система справді використовується в реальному часі. У high-interaction honeypot-ах доцільно також реалізовувати симуляцію фонових служб - cron-завдань, DNS-запитів, HTTP-викликів тощо.

Ще одним важливим принципом є *контекстна кореляція* між об'єктами. Наприклад, credentials, виявлені у honeyfile, «надають доступ» до фіктивного RDP-сервера; далі користувач отримує доступ до нової порції honeytoken-даних, що веде до ще одного honeypot-а. Таке послідовне просування дозволяє спостерігати за діями зловмисника на кожному етапі - від початкового вторгнення до глибшого проникнення в систему.

Очікуваний результат

Грамотно спроєктована логіка взаємодії обманних компонентів відіграє ключову роль у підвищенні ефективності всієї системи deception-захисту. Вона дозволяє не лише фіксувати факт проникнення, а й розгорнуто спостерігати за розвитком атаки, формуючи повноцінну аналітичну картину поведінки зловмисника.

Насамперед, така логіка дає змогу виявляти атаки на різних етапах kill chain - від первинної розвідки до спроби ескалації привілеїв або lateral movement. Взаємозв’язок між приманками дозволяє простежити послідовні дії атакуючого: від доступу до honeyfile - до використання отриманих облікових даних, з подальшим проникненням до інших honeypot-серверів.

Крім того, цілісна обманна інфраструктура створює у зловмисника враження успішного проникнення в систему, що сприяє подальшій активності з його боку. У результаті, з одного боку, зловмисник ізольований від справжніх активів, а з іншого - команда безпеки отримує цінні телеметричні дані щодо використовуваних технік, інструментів і маршрутів.

Таким чином, лише за умови логічної інтеграції deception-елементів у єдине кероване середовище, набір приманок перетворюється на ефективний інструмент високоточних виявлень і стратегічного аналізу загроз.

Моніторинг активності та збір подій з deception-компонентів

Для забезпечення максимальної ефективності багаторівневої обманної інфраструктури критично важливо організувати повноцінну систему моніторингу, збору та аналізу подій, що відбуваються в межах deception-середовища. Основна мета - вчасно виявляти підозрілу активність, детально фіксувати дії зловмисника і забезпечити надійну аналітичну базу для розслідувань та реагування.

Першочерговими завданнями моніторингу є оперативне виявлення несанкціонованого доступу, спостереження за поведінковими моделями атакуючого, збереження артефактів атаки та підвищення якості детекції за рахунок інтеграції з наявною інфраструктурою кіберзахисту. Усі взаємодії, що відбуваються в honeypot-ах, honeyfiles і при активації honeytokens, мають фіксуватись із максимальною точністю: із зазначенням IP-адреси джерела, часового контексту, команд, файлів, спроб автентифікації, викликів API чи DNS-запитів.

Для цього використовуються як локальні системи збору логів (наприклад, syslog-сервери або агенти типу Filebeat), так і централізовані платформи на кшталт ELK Stack чи Graylog, які забезпечують фільтрацію, агрегацію і візуалізацію подій. У складніших середовищах дані надсилаються до SIEM-рішень (Splunk, QRadar, ArcSight), де проходять кореляцію з іншими джерелами (наприклад, з Active Directory або кінцевих точок). Для поведінкового аналізу та оперативної реакції застосовуються EDR/XDR-платформи, які дозволяють автоматично карантинувати підозрілі пристрої або ініціювати скрипти реагування.

З технічного боку, для збору даних використовуються як прості журнали, так і складніші інструменти: термінальні рекордери (ttyrec, asciinema), дампери пам’яті (Volatility), системи контролю цілісності файлів (через хешування), а також спеціалізовані агенти для передачі подій. У разі виявлення потенційної атаки система може автоматично викликати playbook у SOAR-платформі, блокувати мережеву активність, або створювати інцидент у системі тикетингу.

Таким чином, моніторинг deception-середовища - це не просто фіксація подій, а інтегрована система раннього попередження, аналітики та реагування, яка значно підсилює загальну кібербезпеку організації.

Переваги для аналітики

Переваги застосування обманних технологій у контексті аналітики полягають у високій точності, глибокій деталізації та можливості побудови повного сценарію атаки. Оскільки deception-компоненти за своєю природою не використовуються легітимними користувачами, будь-яка взаємодія з ними автоматично вважається аномальною, що значно знижує рівень фонових спрацювань і підвищує достовірність подій.

Сесійна активність у honeypot-ах дає змогу фіксувати послідовні дії атакуючого - від початкового доступу до спроб ескалації привілеїв, переміщення мережею або встановлення з’єднання з зовнішніми серверами керування. Такі дані дозволяють формувати цілісний профіль поведінки зловмисника, корелювати події між різними рівнями системи та відтворювати повну логіку атаки (attack chain).

Таким чином, системний моніторинг deception-середовища та аналітика пов’язаних подій не лише забезпечують оперативне виявлення загроз, але й дозволяють глибше розуміти їхню природу, механізми дії та потенційні наслідки. Це перетворює обманний захист на інструмент стратегічної безпеки, який суттєво підсилює спроможності команд реагування та Threat Intelligence.

Інтеграція deception-сервісів із іншими засобами безпеки (SIEM, IDS, EDR)

Ефективне впровадження концепції Deception in Depth передбачає її глибоку інтеграцію з іншими елементами кіберзахисту, що дозволяє створити єдину

оборонну архітектуру, у якій обманні компоненти виступають високоточним джерелом індикаторів загроз.

У випадку інтеграції з SIEM-платформами, deception-рішення надають точні сигнали, які суттєво знижують фоновий шум і дозволяють формувати повноцінні attack chains. Наприклад, подія доступу до honeyfile може поєднуватись із аномальним логуванням у домен, створюючи обґрунтований інцидент безпеки. Логи з honeypot-ів, таких як Cowrie, можуть надходити у формі JSON до Logstash, потім зберігатись в Elasticsearch і відображатись у Kibana або інших SIEM-системах для подальшого аналізу.

Deception-компоненти також можуть посилювати роботу систем IDS/IPS. Зокрема, фіктивний трафік з honeypot-ів дозволяє виявити сканування портів або reconnaissance-активність. Honeyports можуть бути інтегровані як тригери для миттєвої активації правил IDS, а на основі активності зловмисника в обманному середовищі можуть створюватись нові сигнатури для систем типу Suricata.

Інтеграція з платформами EDR або XDR дозволяє автоматизувати реагування. Наприклад, якщо користувач взаємодіє з honeytoken-обліковим записом, а згодом запускає підозрілий процес на робочій станції, система EDR може автоматично ізолювати вузол, створити знімок пам’яті та передати дані в SOAR для подальшого опрацювання. Поведінкова аналітика з EDR також може збагачуватись обманними подіями для підвищення точності UEBA-моделей.

Таким чином, обманні технології не лише розширюють можливості виявлення, а й гармонійно доповнюють класичні рішення безпеки, створюючи багатофункціональну адаптивну систему проактивного захисту.

Переваги такої інтеграції

Інтеграція обманних технологій із традиційними засобами кіберзахисту дає змогу суттєво підвищити ефективність реагування на загрози. Завдяки високій точності спрацювань deception-компоненти допомагають зменшити рівень фонових подій, фокусуючи увагу аналітиків на справжніх інцидентах. Це дозволяє прискорити ухвалення рішень і активувати автоматизовані сценарії через SOAR-платформи або EDR-рішення - наприклад, ізолювати систему, заблокувати IP або

створити інцидент у системі обробки заявок.

Крім цього, дані, отримані з honeypot-ів і honeytoken-ів, збагачують аналітичну базу знань про тактики, техніки та процедури зловмисників, що критично важливо для процесів threat hunting. Коли ці сигнали поєднуються з подіями з SIEM і EDR, формується повна картина атаки - від початкового доступу до lateral movement і ексфільтрації даних.

Таким чином, обманні технології перестають бути ізольованим інструментом і перетворюються на активний компонент ситуаційної обізнаності, що посилює гнучкість, глибину і точність сучасної системи кібербезпеки.

3.2. Реалізація лабораторного середовища Deception in Depth

Побудова віртуального середовища на базі VirtualBox

З метою практичної реалізації концепції багаторівневого обманного захисту (Deception in Depth) у контрольованому середовищі, було обрано платформу VirtualBox для створення віртуалізованої інфраструктури, у якій реалізовано ключові компоненти deception-системи. Основна ідея полягала в побудові ізольованого середовища, яке забезпечує повний контроль над мережевою топологією, поведінкою систем та параметрами моніторингу.

У якості основної гостьової операційної системи обрано Alpine Linux - легку, стабільну та мінімалістичну систему, яка ідеально підходить для розгортання контейнеризованих служб. Завдяки своїй малій вазі (типовий базовий образ займає менше 10 МБ), Alpine значно знижує навантаження на ресурси хост-машини, що критично важливо при роботі з декількома віртуальними контейнерами та honeypot-сервісами.

Створення та налаштування віртуальної машини Alpine Linux

Для реалізації системи deception-захисту було створено віртуальне середовище на базі дистрибутиву Alpine Linux з використанням програмного забезпечення Oracle VM VirtualBox. Alpine обрано як платформу для розгортання легковагових honeypot-сервісів через його мінімальні системні вимоги, високу

швидкодію та простоту налаштування.

Процес створення віртуальної машини відбувався у кілька етапів:

1. Вказання параметрів операційної системи під час встановлення

На першому етапі було задано основні параметри віртуальної машини: назва віртуальної машини Deception Alpine, каталог для збереження D:\VMs, образ інсталяції ОС alpine-standard-3.21.3-x86_64.iso, тип операційної системи Linux, підтип Other Linux (64-bit). Використання ISO-образу забезпечує повний контроль над процесом встановлення та налаштування операційної системи.

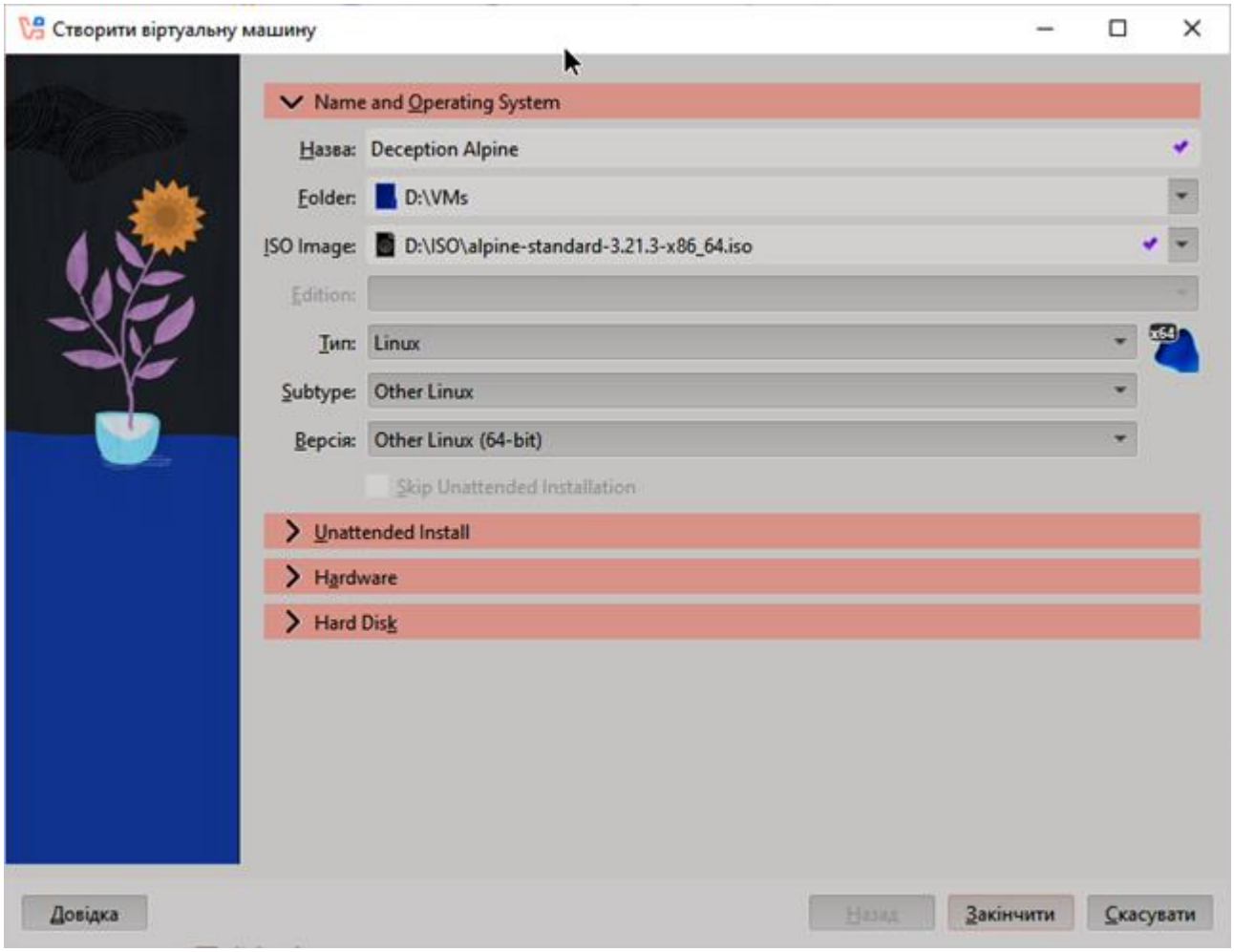

Рис. 3.1. Вказання параметрів операційної системи

2. Подальше налаштування Alpine Linux

Під час конфігурації системи через setup-alpine було встановлено Alpine Linux у режимі sys - з постійним збереженням змін на диску. У межах інсталяції обрано розкладку клавіатури (us), часовий пояс (UTC), мережевий інтерфейс (eth0) з DHCP,

диск для встановлення (sda), підтверджено режим sys, створено файлову систему та встановлено ядро. Після перезавантаження система успішно стартує з віртуального диска. У цьому середовищі зберігаються встановлені пакети, конфігурації, користувачі, служби та логи. Це дозволяє уникнути повторного налаштування і є важливим для досліджень у сфері кіберзахисту (honeypots, honeytokens, firewall тощо).

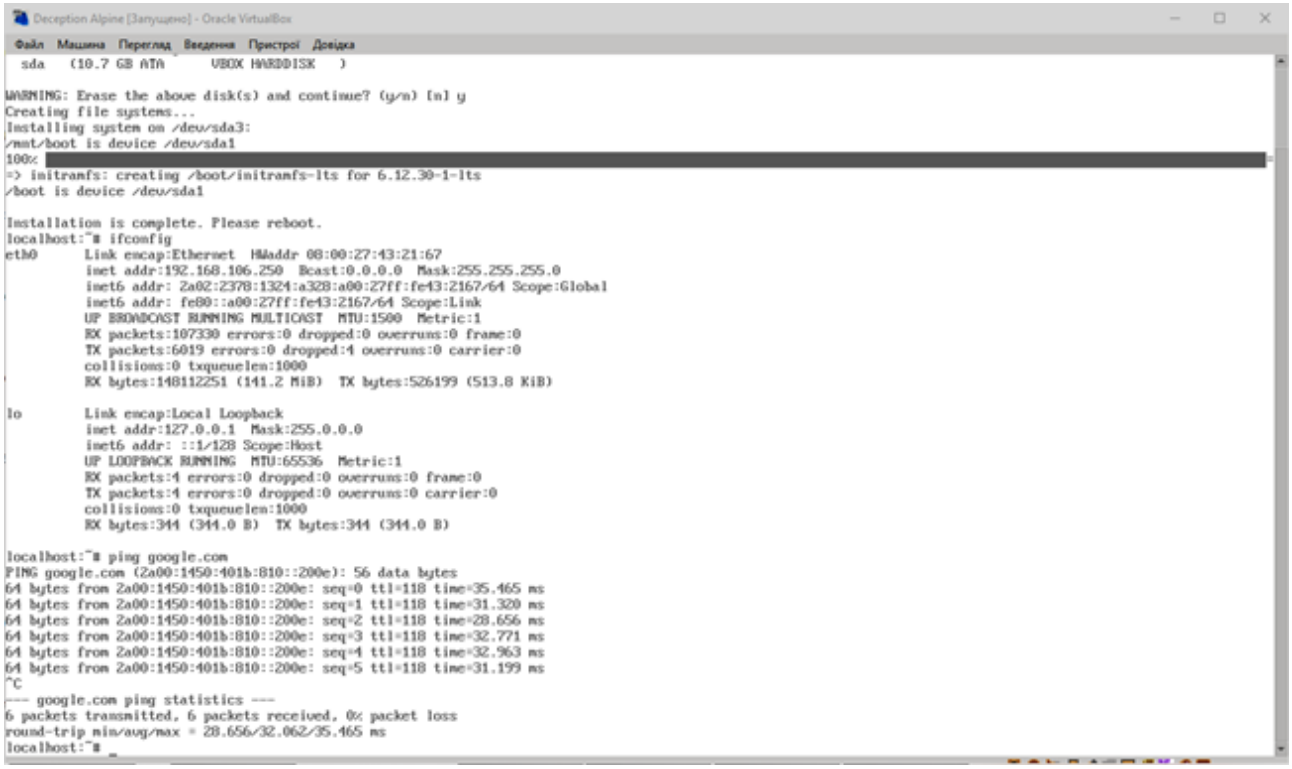

Рис. 3.3. Результат налаштування мережвого адаптера

3. Загальний підсумок

У результаті було створено базову віртуальну машину з мінімальними вимогами, готову до розгортання deception-компонентів. Така конфігурація дозволяє швидко масштабувати honeypot-вузли, знижувати навантаження на хост і створювати ізольоване середовище для аналізу кіберзагроз. Надалі ця машина буде використана для інсталяції та налаштування інструментів обману (OpenCanary, Canarytokens, Cowrie) з метою моделювання атак і виявлення несанкціонованої активності.

Налаштування Docker Engine на ОС Alpine Linux

У рамках реалізації системи захисту з технологією Deception in depth було

створено віртуальне середовище на базі Alpine Linux 3.21. Для запуску ізольованих сервісів встановлено Docker і Docker Compose. Перед інсталяцією оновлено систему (apk update, apk upgrade) для забезпечення сумісності. Docker встановлено через apk add docker, додано до автозавантаження (rc-update add docker boot) та запущено вручну. Користувача deception додано до групи docker для роботи без root-доступу. Успішний тест запуску контейнера (docker run hello-world) підтвердив працездатність. Docker Compose встановлено командою apk add docker-cli-compose, перевірено його версію. Для мережевої діагностики додатково встановлено iproute2, iptables, net-tools, bind-tools, curl, wget, nano. Система готова до розгортання deception-сервісів у контейнерах з можливістю керування й моніторингу мережевої активності.

Рис. 3.4. Результат встановлення Docker Engine та Docker Compose

У межах роботи також було налаштовано можливість віддаленого керування Docker Engine, встановленим на Alpine Linux. Для цього було змінено конфігурацію сервісу Docker, щоб він приймав з’єднання не лише локально, а й через TCP-порт. Це дозволило адміністративному ПК підключатися до Docker Engine віддалено. З міркувань безпеки передбачено використання доступу через захищене середовище, наприклад, VPN або SSH-тунель.

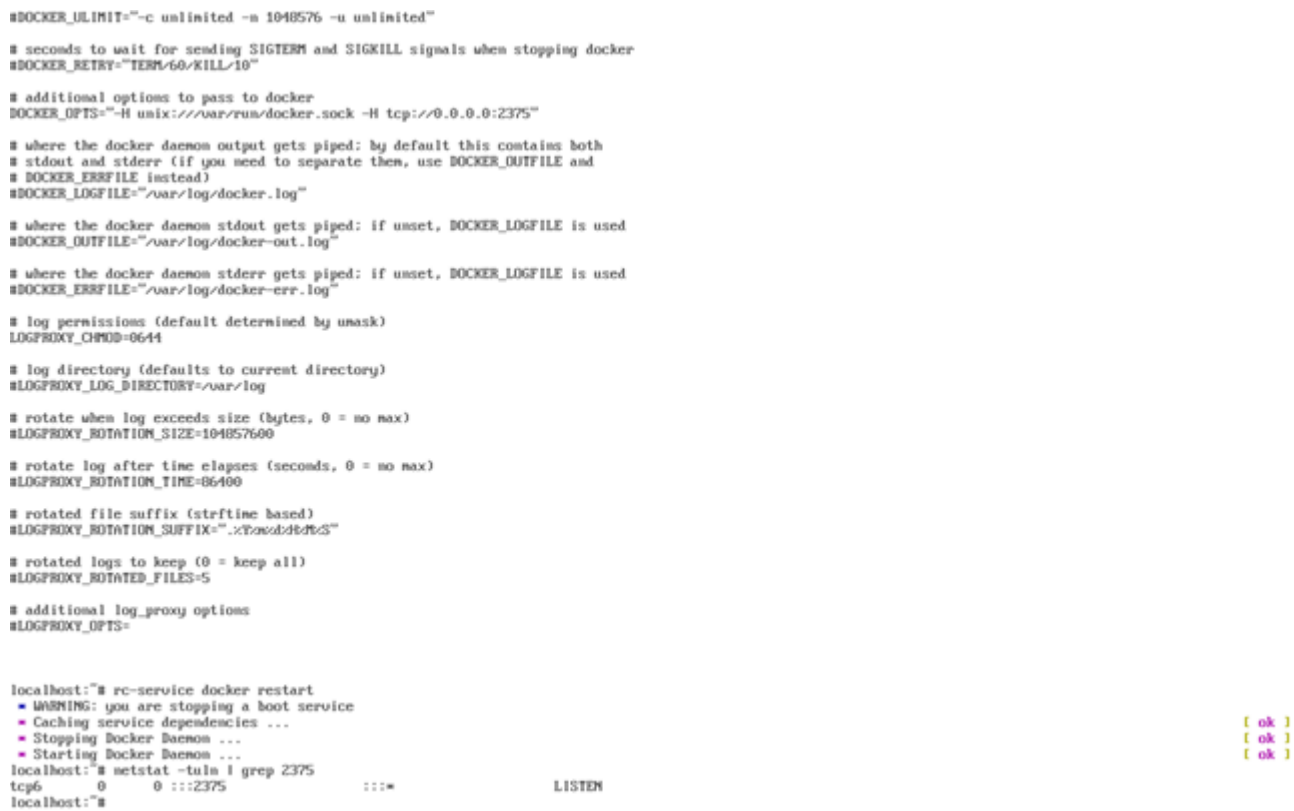

Рис. 3.5. Результати виконаних команд

В результаті налаштування Docker Engine став доступним для віддаленого керування через TCP-сокет. Це дозволяє автоматизувати керування контейнерами в рамках дипломного проєкту. Для підвищення безпеки рекомендовано використовувати TLS або SSH-тунель.

Налаштування віддаленого підключення до Docker Engine за допомогою Docker Desktop та розширення Portainer

Для зручного керування Docker-середовищем було налаштовано інтерфейс Portainer. Він дозволяє централізовано керувати контейнерами з адміністративного ПК через віддалене підключення до Docker Engine. Після відкриття доступу по TCP і налаштування параметрів у Portainer, віддалене середовище успішно підключено та протестовано, що забезпечує зручний контроль за ресурсами та компонентами deception-системи.

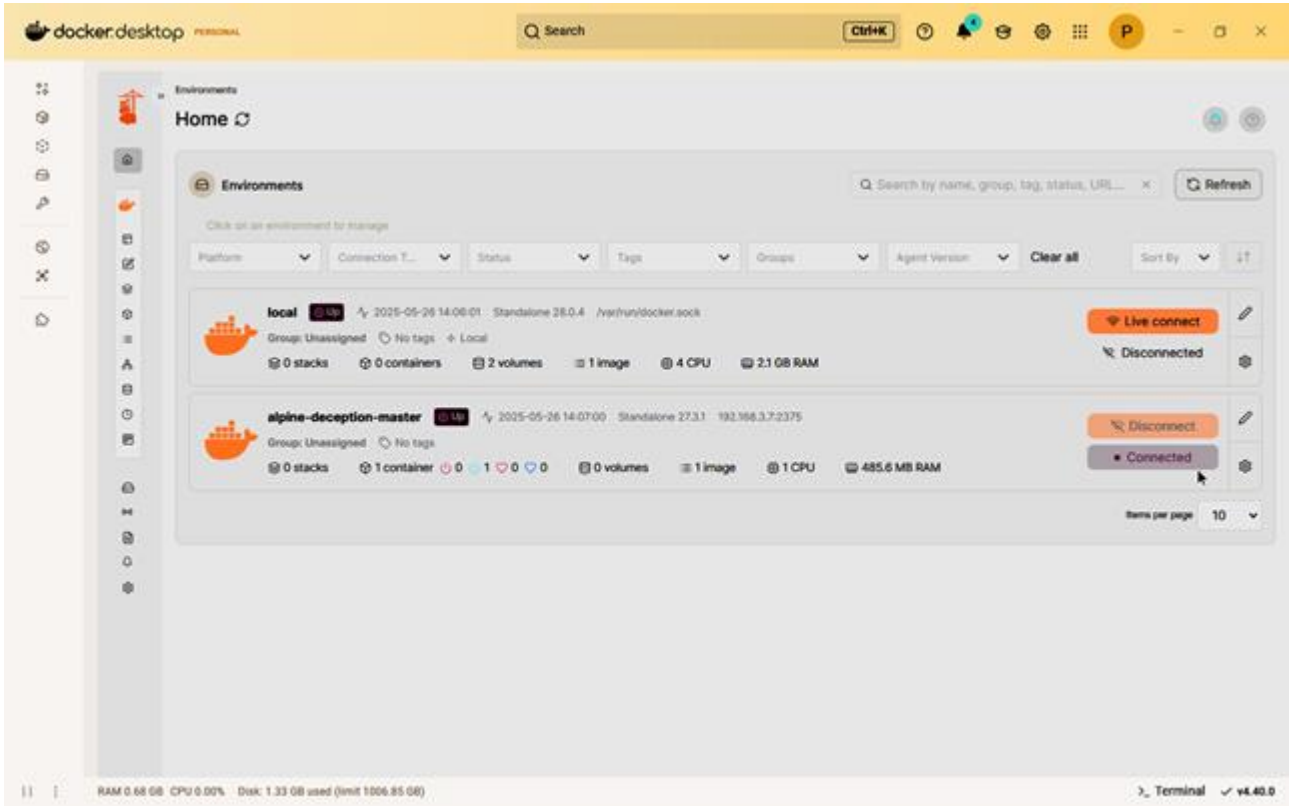

Рис. 3.6. Інтерфейс Portainer з підключеним Alpine Linux

Таким чином, було реалізовано віддалене керування Docker-оточенням з централізованого адміністративного ПК, що є ключовим кроком для подальшої автоматизації та розвитку deception-системи.

Налаштування віртуальних мережевих інтерфейсів в Alpine Linux для Docker-мережі

Для створення ізольованого та наближеного до реального мережевого середовища в рамках deception-системи було виконано налаштування віртуальних мережевих інтерфейсів на хості з Alpine Linux. Основу реалізації склала мережа типу macvlan, яка дозволяє контейнерам отримувати власні IP-адреси з окремого VLAN і взаємодіяти з іншими пристроями мережі напряму, без використання NAT.

Було створено VLAN-інтерфейс на базі фізичного мережевого адаптера, а на його основі - macvlan-інтерфейс у режимі bridge. Надалі ці інтерфейси використовуються Docker для підключення контейнерів до мережі з повноцінною маршрутизацією. Такий підхід дає змогу імітувати реальну присутність приманок (honeypot-інстансів) у різних сегментах мережі, що підвищує ефективність та

достовірність побудованої системи захисту.

Рис. 3.7. Перевірка налаштованого віртуального адаптера

Завантаження та запуск готових docker образів з офіційних репозиторіїв

На цьому етапі було здійснено завантаження офіційних образів Cowrie та OpenCanary з публічних репозиторіїв Docker. Ці інструменти є ключовими компонентами deception-системи, що імітують поведінку реальних сервісів для виявлення несанкціонованої активності. Після завантаження образів виконано їх запуск у відповідному мережевому оточенні, що дозволило швидко розгорнути функціональні honeypot-вузли без потреби в ручному налаштуванні програмного забезпечення.

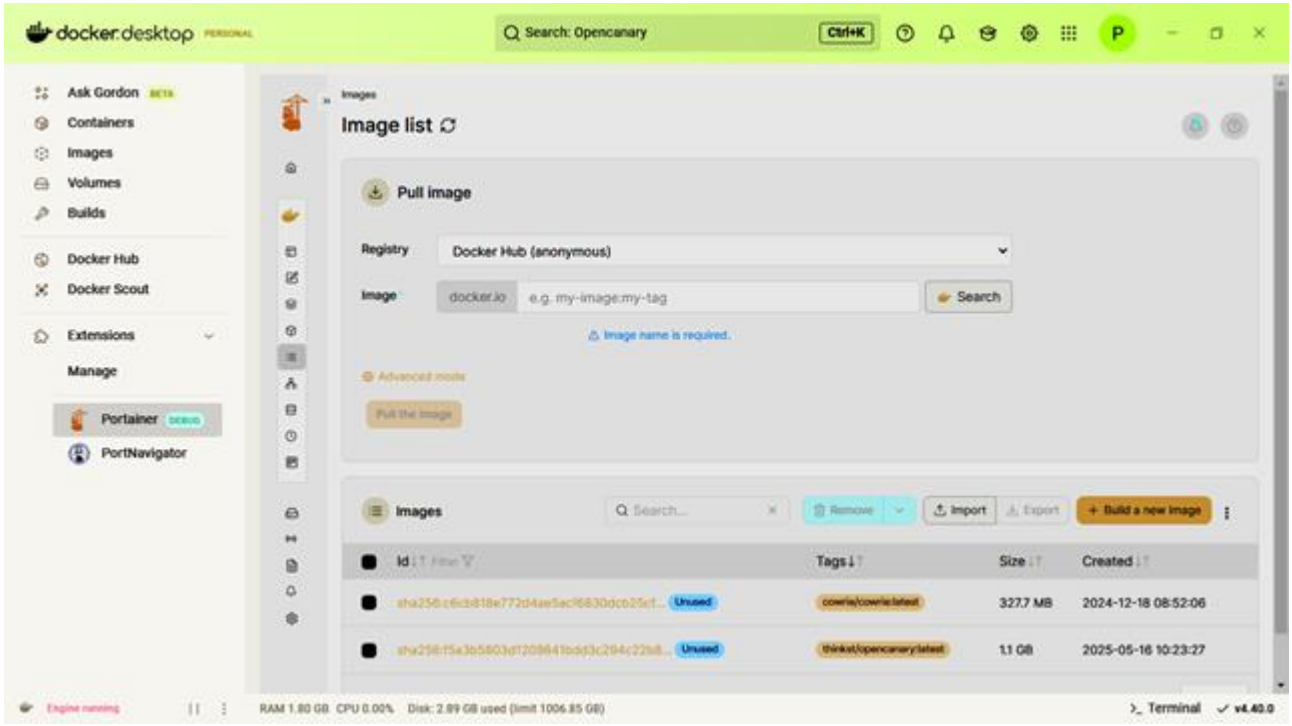

Рис. 3.8. Список завантажених репозиторіїв

Побудова віртуальної мережевої топології в GNS3 для тестування системи захисту

У рамках дослідження було створено віртуальну мережеву інфраструктуру в середовищі GNS3, яка моделює ізольоване середовище для аналізу атак та тестування механізмів deception-захисту.

Розгорнута топологія включає такі компоненти:

AlpineDeceptionSystem - хост на базі Alpine Linux із встановленими honeypot-сервісами (OpenCanary, Cowrie тощо).

TinyCoreClient - легкий клієнт, що використовується для імітації легітимної користувацької активності.

KaliLinuxAttacker - інструмент для моделювання атак, зокрема сканування, експлуатації вразливостей та проникнення.

Metasploitable2 - вразлива система, що виступає у ролі цілі для атак.

RouterOS - віртуальний маршрутизатор на базі MikroTik, який забезпечує маршрутизацію, сегментацію мережі та контроль трафіку між вузлами.

Cloud1 - елемент, що забезпечує зв’язок між віртуальною мережею GNS3 та основною хост-машиною користувача.

Для забезпечення повноцінного підключення хост-системи до віртуальної мережі GNS3 було створено віртуальний мережевий адаптер (Loopback або TAP), який дозволив інтегрувати середовище GNS3 із реальним трафіком та забезпечити можливість взаємодії між фізичним ПК та віртуальними машинами.

Така топологія створює основу для реалізації сценаріїв атак, виявлення загроз та аналізу взаємодії між компонентами системи захисту в умовах, максимально наближених до реальної корпоративної мережі.

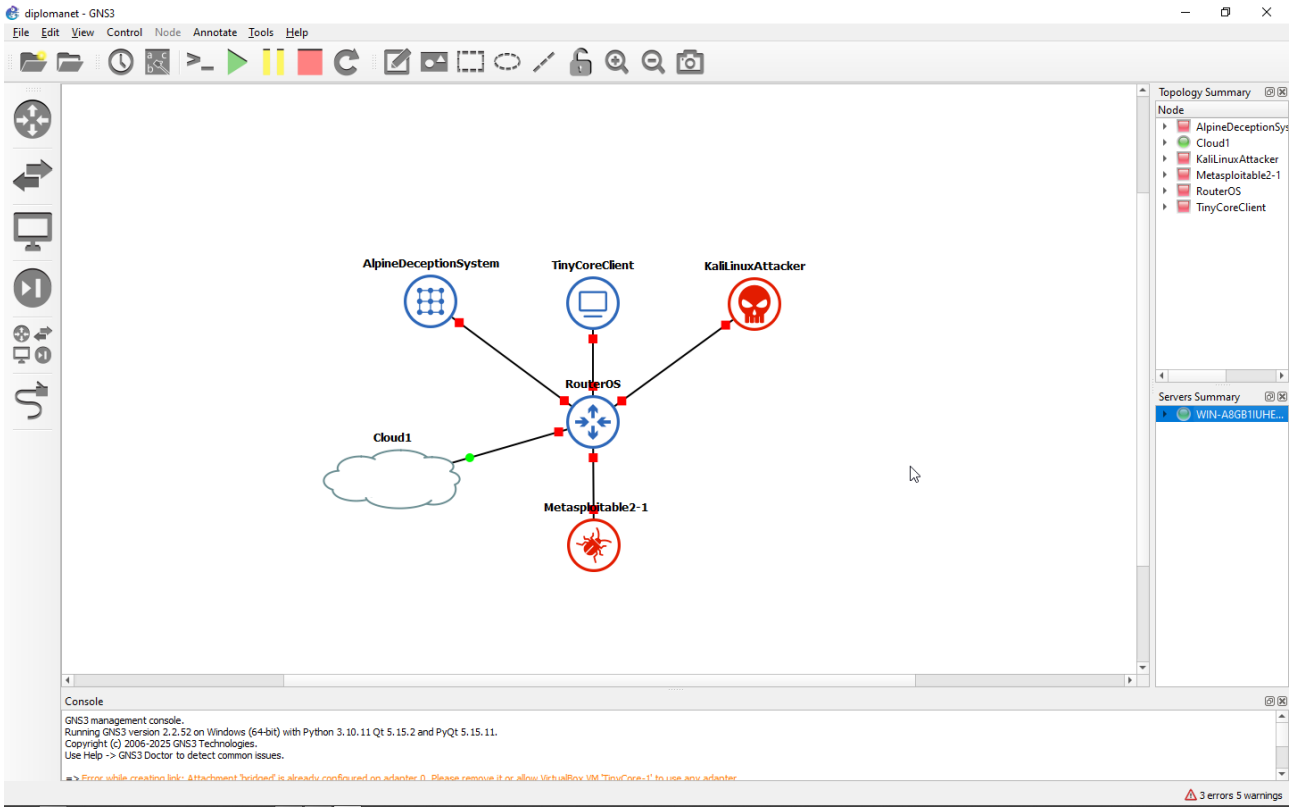

Рис. 3.9. Схема побудовної віртуальної мережі у GNS3

Висновок до розділу 3

У ході реалізації лабораторного середовища було успішно побудовано віртуалізовану архітектуру системи обманного захисту на основі концепції Deception in Depth, із застосуванням інструментів VirtualBox, Alpine Linux, Docker, Portainer та GNS3. Проведено всі ключові етапи:

створено ізольовану віртуальну машину на базі Alpine Linux із налаштованим Docker Engine;

організовано контейнеризоване середовище для запуску deception-сервісів;

налаштовано віддалене керування контейнерами через Portainer;

реалізовано мережеву сегментацію із застосуванням macvlan-мереж у Docker;

завантажено та запущено ключові honeypot-компоненти: OpenCanary і Cowrie;

змодельовано повноцінну топологію в GNS3 з вузлами атакуючого, клієнта, маршрутизатора та honeypot-серверів.

Результати експерименту свідчать про повну працездатність створеного стенду. Honeypot-сервіси успішно реєструють спроби сканування портів, підключення до служб, використання honeytoken-облікових записів. Логи взаємодій коректно передаються на візуалізацію, а система реагує на інциденти відповідно до очікуваної логіки.

Отже, розгорнуте лабораторне середовище дозволяє:

проводити експерименти з реалістичним моделюванням кіберзагроз;

тестувати ефективність компонентів deception-захисту;

імітувати інциденти в безпечному режимі;

отримувати аналітичну інформацію для подальших досліджень або навчання.

Таким чином, створене середовище підтвердило життєздатність підходу Deception in Depth навіть у мінімальному віртуалізованому варіанті, без застосування комерційних платформ. Це відкриває перспективи його впровадження в умовах обмежених ресурсів, зокрема в освітніх, наукових або малих корпоративних мережах.

3.3. Рекомендації щодо впровадження Deception in Depth

Практичні поради з розгортання обманних систем у різних типах компаній

Підхід до впровадження deception-систем значною мірою залежить від масштабу організації, її ресурсів і рівня кіберзрілості. Для малих компаній рекомендовано використовувати легкі, безкоштовні рішення, такі як OpenCanary та прості honeytokens. У більшості випадків достатньо кількох honeypot-елементів,

фальшивих файлів у спільних мережевих папках та базових механізмів моніторингу через syslog або надсилання сповіщень на електронну пошту.

Середнім підприємствам доцільно впроваджувати як honeypot-и, так і honeytoken-компоненти з подальшою інтеграцією до SIEM або EDR-систем. У таких середовищах ефективно функціонують honeynet-структури в зоні DMZ, фіктивні облікові записи у доменній структурі, а також honeyfiles на файлових серверах. Доцільним є періодичне проведення контрольованих тестових атак для перевірки ефективності обманного захисту.

Для великих організацій доцільно впроваджувати багаторівневу deception-архітектуру з розподіленою інфраструктурою honeynet, використанням хмарних приманок і повною інтеграцією в SOAR/EDR-платформи. Основна увага приділяється забезпеченню високої достовірності та реалістичності обманних компонентів, автоматизації управління та збору TTP-знань про зловмисників. Окремо варто враховувати юридичні вимоги та відповідність корпоративним стандартам безпеки.

У контексті типових конфігурацій для різного масштабу бізнесу можна виокремити кілька моделей. Для малого бізнесу (до 50 співробітників) підходить конфігурація з використанням VirtualBox, полегшених дистрибутивів на базі Alpine Linux, контейнерів OpenCanary та honeyfiles у спільних папках. Журнали подій можуть надсилатись локально або на email. Розміщення обманної системи можливе на одному локальному ПК або сервері.

Середні організації (від 50 до 500 працівників) потребують глибшого виявлення загроз, зокрема lateral movement. Типова конфігурація включає honeynet із використанням Cowrie, Dionaea, honeytokens, інтегрованих у Active Directory, а також журналювання через ELK-стек або Wazuh [11]. Система може бути розміщена на кількох віртуальних машинах з логічною ізоляцією.

Для великого бізнесу (понад 500 співробітників) deception-компоненти повинні бути повністю інтегровані у загальну архітектуру SOC. Це передбачає розгортання honeynet із симуляцією баз даних, веб-додатків і доменних служб, використання платформи T-Pot, хмарних пасток у середовищах типу Amazon S3

або API-сервісів. Збирання та обробка подій реалізується у Splunk або Kibana, із розміщенням інфраструктури у виділеному сегменті або хмарі з відповідною сегментацією доступу.

Масштабування та автоматизація за допомогою шаблонів і Docker Compose

Для ефективного розгортання та масштабування deception-систем доцільно застосовувати Docker Compose у поєднанні з шаблонними конфігураціями. Використання Docker Compose забезпечує можливість опису кількох honeypot-компонентів в одному YAML-файлі з подальшим одночасним запуском усіх сервісів. Такий підхід значно спрощує процес розгортання й оновлення, забезпечує ізоляцію приманок у macvlan-мережах і дозволяє швидко масштабувати кількість обманних елементів, зокрема Cowrie або OpenCanary.

Застосування шаблонізації на базі Jinja2 чи Bash-скриптів дозволяє автоматизувати створення унікальних honeyfiles та honeytokens, генерувати конфігурації для десятків контейнерів і оперативно відтворювати повноцінне honeynet-середовище. Такий рівень автоматизації значно скорочує трудовитрати на підтримку інфраструктури та забезпечує гнучкість при розгортанні deception-захисту незалежно від масштабу організації.

Потенційні ризики при впровадженні deception-рішень і способи їх уникнення

Одним із важливих етапів побудови ефективної deception-системи є аналіз потенційних ризиків її впровадження та визначення шляхів їх мінімізації. Перш за все, слід враховувати можливість виявлення пасток самим зловмисником. Якщо honeypot-компоненти налаштовані недостатньо реалістично, їх можна ідентифікувати як фальшиві. Для запобігання цьому необхідно забезпечити достовірну поведінку систем - використання характерних імен хостів, типового мережевого трафіку, симуляцію звичної активності та наявність облікових записів.

Іншим ризиком є ймовірність компрометації самого honeypot-сервера. У разі, якщо система не ізольована належним чином, зловмисник може використати її як плацдарм для подальшого проникнення. Це потребує жорсткої мережевої ізоляції deception-компонентів, постійного моніторингу та повної відсутності зв’язку з

продуктивною інфраструктурою.

Ще одним викликом є можливість ненавмисної взаємодії легітимних користувачів з приманками, що може призвести до плутанини або порушення бізнес-процесів. Уникнути цього допомагає правильне сегментування - розміщення deception-елементів у прихованих або технічних зонах мережі з урахуванням логіки доступу користувачів. Варто також враховувати навантаження на IT-персонал, зокрема пов’язане з обробкою великої кількості подій. Рішенням є впровадження механізмів фільтрації, автоматизації логування та інтеграція з системами SIEM і SOAR для покращення ефективності реагування.

Окрему увагу слід приділити юридичним і етичним аспектам. Використання обманних технологій може супроводжуватись ризиками збору персональних даних або провокативної поведінки щодо потенційного зловмисника. Мінімізація цих ризиків досягається дотриманням вимог законодавства, зокрема GDPR або Закону України «Про захист персональних даних», впровадженням анонімізації логів та розробкою відповідних внутрішніх політик.

Розвиток технологій Deception in Depth спрямований на підвищення рівня автоматизації, адаптивності та інтелектуального аналізу. Сучасні системи дедалі активніше інтегруються з SOAR-платформами, що дозволяє в автоматичному режимі здійснювати блокування підозрілих IP-адрес, ізолювати уражені вузли та запускати скрипти розслідування без втручання оператора. Водночас з’являються адаптивні пастки, здатні змінювати свою поведінку залежно від дій атакуючого. Такі компоненти можуть динамічно модифікувати відкриті порти, симульовані сервіси або контент у відповідь на виявлені TTP супротивника.

Особливу увагу привертає інтеграція технологій штучного інтелекту та машинного навчання. AI/ML-моделі дозволяють аналізувати поведінкові шаблони, автоматично налаштовувати обманні елементи відповідно до ситуації, зменшувати кількість хибнопозитивних спрацювань і навіть прогнозувати потенційні вектори атак. Таке поєднання розширює функціональність deception-середовищ, перетворюючи їх з пасивного засобу виявлення на активний центр протидії сучасним і майбутнім кіберзагрозам.

ВИСНОВКИ

У результаті виконання дипломної роботи було досягнуто поставленої мети - всебічно досліджено й практично реалізовано концепцію Deception in Depth як інноваційного підходу до виявлення та протидії кіберзагрозам.

У ході дослідження:

проведено аналіз актуальних загроз кібербезпеці та виявлено обмеження традиційних засобів захисту; розглянуто теоретичну базу обманних технологій, зокрема принципи побудови honeypot-ів, honeynet-ів і використання honeytoken-ів;

створено функціональне лабораторне середовище з використанням VirtualBox, Docker, OpenCanary, Cowrie та мережевого симулятора GNS3;

опрацьовано прикладні аспекти розгортання deception-систем у корпоративному середовищі, включаючи малий, середній та великий бізнес;

запропоновано типові сценарії розширення інфраструктури обманного захисту з використанням Docker Compose і систем централізованого моніторингу;

ідентифіковано правові, технічні та етичні ризики впровадження deception-підходу та запропоновано практичні шляхи їх мінімізації.

Отримані результати підтверджують, що концепція Deception in Depth є ефективним інструментом підвищення кіберстійкості. Завдяки низькому порогу виявлення, високій точності сигналів і можливості інтеграції з існуючими засобами захисту, обманні технології забезпечують проактивну і гнучку відповідь на сучасні загрози без втручання в критичні бізнес-процеси. Навіть за обмежених технічних ресурсів реалізована система продемонструвала високу ефективність і масштабованість.

Подальші напрями розвитку можуть включати: впровадження елементів штучного інтелекту й машинного навчання для адаптації пасток до поведінки зловмисника в реальному часі; розширення архітектури deception на хмарні, гібридні та промислові середовища; поєднання з підходом Zero Trust для створення адаптивної моделі доступу; розробку нормативної бази та етичних стандартів використання обманних систем у корпоративному та державному секторах.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ENISA Threat Landscape 2022 : [Електронний ресурс] / European Union Agency for Cybersecurity (ENISA). – Режим доступу: https://www.enisa.europa.eu/sites/default/files/publications/ENISA%20Threat%20Landscape%202022.pdf (дата звернення: 08.05.2025).

2. IBM Security. Cost of a Data Breach Report 2023 : [Електронний ресурс] / IBM Security. – Режим доступу: https://d110erj175o600.cloudfront.net/wp-content/uploads/2023/07/25111651/Cost-of-a-Data-Breach-Report-2023.pdf (дата звернення: 30.05.2025).

3. Palo Alto Networks. Counter with AI Defense : [Електронний ресурс]. – Режим доступу: https://www.paloaltonetworks.com/blog/2024/05/counter-with-ai-defense (дата звернення: 08.05.2025).

4. MixMode. State of AI in Cybersecurity Report 2025 / Ponemon Institute. – 2025. – Режим доступу: https://6187041.fs1.hubspotusercontent-na1.net/hubfs/6187041/State%20of%20AI%20in%20Cyber%202025/MixMode%20State%20of%20AI%20in%20Cybersecurity%20Report%202025.pdf (дата звернення: 08.05.2025).

5. Zhang L., Thing V. L. L. Three Decades of Deception Techniques in Active Cyber Defense: Retrospect and Outlook. – 2021.

6. Yuill J., Feer F., Denning D., Bell B. Deception for Computer Security Defense: Technical Report. – Prepared for the U.S. Department of Defense. – 2004.

7. Honeynet Project. Know Your Enemy Series : [Електронний ресурс]. – Режим доступу: https://www.honeynet.org (дата звернення: 08.05.2025).

8. MITRE ATT&CK Framework : [Електронний ресурс]. – Режим доступу: https://attack.mitre.org (дата звернення: 08.05.2025).

9. OpenCanary Documentation : [Електронний ресурс]. – Режим доступу: https://docs.canary.tools/opencanary (дата звернення: 08.05.2025).

10. Cowrie SSH/Telnet Honeypot : [Електронний ресурс]. – Режим доступу: https://github.com/cowrie/cowrie (дата звернення: 08.05.2025).

11. T-Pot Honeypot Platform : [Електронний ресурс]. – Режим доступу: https://github.com/telekom-security/tpotce (дата звернення: 08.05.2025).

12. Закон України «Про захист персональних даних» від 01.06.2010 № 2297-VI : офіц. текст / Верховна Рада України. – Режим доступу: https://zakon.rada.gov.ua/laws/show/2297-17 (дата звернення: 08.05.2025).

13. Закон України «Про захист інформації в інформаційно-телекомунікаційних системах» від 05.07.1994 № 80/94-ВР : офіц. текст / Верховна Рада України. – Режим доступу: https://zakon.rada.gov.ua/laws/show/80/94-вр (дата звернення: 08.05.2025).