МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет

Факультет електроніки та інформаційниї технологій

Кафедра кібербезпеки

«До захисту допущено»

Завідувач кафедри

_____ Володимир ЛЮБЧАК

(підпис) (Ім’я та ПРІЗВИЩЕ)

___ _____ 20__р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавр

зі спеціальності 125 Кібербезпека, освітньо-професійної програми Кібербезпека на тему: «Система детектування зловмисної діяльності в інформаційно-комунікаційній системі комерційного підприємства»

Здобувача (ки) групи КБ-01 Євтушенка Романа Олексійовича

(шифр групи) (прізвище, ім’я, по батькові)

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

_____ Роман ЄВТУШЕНКО

(підпис) (Ім’я та ПРІЗВИЩЕ здобувача)

Керівник ст. викладач, кафедри кібербезпеки Вадим КАЛЬЧЕНКО _____

(посада, науковий ступінь, вчення звання, Ім’я та ПРІЗВИЩЕ) (підпис)

Суми – 2024

АНОТАЦІЯ

Кваліфікаційна робота виконана на 50 аркушах та містить 51 рисунок, 20 джерел.

Об’єкт дослідження: локальна мережа, системи детектування зловмисної діяльності в локальній мережі: IDS, Honeypot, CanaryTokens.

Мета роботи: аналіз існуючих програмних рішень з виявлення спроб зламу комп’ютерних систем та побудова на їх на основі системи детектування в інтересах комерційного підприємства.

Метод дослідження: аналіз літературних джерел за обраною тематикою, дослідження та аналіз методів несанкціонованого доступу до інформаційно-комунікаційних систем та кіберзахисту від них.

Результати роботи: проведено аналіз програмних рішень, що дозволяють виявити зловмисну діяльність, досліджено практики захисту мережі та види можливих загроз, проведено аналіз технології CanaryTokens, їх види та принципи роботи. Досліджено методи реалізації технології Honeypot та принципи їх роботи. На основі CanaryTokens та Honeypot створено рішення для виявлення спроб зламу комп’ютерної системи з фіксацією даних фактів SIEM-системі Splunk.

Ключові слова: локальна мережа, кібератаки на локальну мережу, кібербезпека, CanaryTokens, Honeypot, Splunk.

ВСТУП

У сучасних умовах розвитку інформаційних технологій та їх впровадження в роботу комерційних підприємств, проблема забезпечення інформаційної безпеки стає дедалі актуальнішою. Системи детектування зловмисної діяльності в інформаційно-комунікаційних системах є одним з важливих елементів забезпечення конфіденційності, цілісності та доступності інформаційних активів компанії. Хакери постійно розробляють нові методи атак, що вимагає від підрозділів інформаційної безпеки в компанії впровадження сучасних і ефективних засобів захисту.

Актуальність дослідження зумовлена різким зростанням кількості кібератак, які призводять до значних фінансових втрат, зниження довіри клієнтів та репутаційних ризиків для комерційних підприємств. Недостатнє забезпечення інформаційної безпеки може призвести до серйозних наслідків, включаючи компрометацію конфіденційної інформації.

На сьогоднішній день існує безліч підходів до виявлення зловмисної діяльності, проте їх ефективність варіюється залежно від складності атак та технологічних можливостей зловмисників. Сучасні системи детектування базуються на різноманітних технологіях, таких як аналіз поведінки, машинне навчання, інтеграція з системами управління інформаційною безпекою (SIEM) тощо. Однак, швидке змінення кіберзагроз вимагає постійного вдосконалення цих систем та розробки нових методів детектування.

Об’єктом дослідження є інформаційно-комунікаційна система комерційного підприємства, яка включає в себе апаратні та програмні засоби, мережеву інфраструктуру. Предметом дослідження є методи та засоби детектування зловмисної діяльності в зазначеній системі, що включають аналіз мережевого трафіку та інші підходи.

Метою даної роботи є розробка системи детектування зловмисної діяльності в інформаційно-комунікаційній системі комерційного підприємства, використовуючи технологію CanaryTokens та Honeypot.

Таким чином, дослідження сприятиме підвищенню рівня інформаційної безпеки комерційних підприємств та зниженню ризиків, пов'язаних зі зловмисною діяльністю в інформаційно-комунікаційних системах.

1. АРХІТЕКТУРА ЛОКАЛЬНОЇ МЕРЕЖІ

1.1 Склад локальної мережі

Практично всі комерційні підприємства працюють за допомогою IT-інфраструктури. Мережева інфраструктура є ключовим компонентом для комерційних підприємств, оскільки вона забезпечує основу для ефективної роботи та взаємодії між співробітниками, системами та зовнішніми партнерами. Спільний доступ до ресурсів дозволяє співробітникам ділитися файлами, документами та іншими ресурсами, що сприяє продуктивності та ефективності. Використання електронної пошти, месенджерів, відеоконференцій та VoIP забезпечує швидке та ефективне спілкування між співробітниками, партнерами та клієнтами. Мережева інфраструктура може легко розширюватися для підтримки зростання підприємства, додавання нових співробітників, офісів або відділів. Також дозволяє інтегрувати бізнес-процеси з постачальниками та клієнтами, покращуючи ланцюги постачання та обслуговування клієнтів.

Загальна архітектура локальної мережі (LAN) комерційного підприємства зазвичай включає: клієнтські пристрої, мережеве обладнання, сегменти та сервери[1]. Клієнтськими пристроями можуть бути ПК, ноутбуки, принтери, сканери, IP-телефони та інші кінцеві пристрої. Мережеве обладнання потрібне для з’єднання в одну систему клієнтських пристроїв з іншими пристроями. У ролі пристроя для з'єднання клієнтських пристроїв у локальну мережу використовують комутатор(Switch). Маршрутизатори(Routers) з'єднують локальну мережу з іншими мережами, такими як Інтернет, забезпечують маршрутизацію даних на мережевому рівні. Також є безпровідні точки доступу(Access Points), які забезпечують бездротовий зв'язок для мобільних пристроїв і часто одразу інтегровані в маршрутизатор. Сегменти потрібні для забезпечення кібербезпеки підприємства, наприклад VLAN (Virtual Local Area Network) є віртуальними сегментами мережі для ізоляції та управління трафіком, підвищують безпеку та ефективність мережі. Основний сегмент для критично важливих сервісів та основних ресурсів, а гостьовий сегмент вже для

відвідувачів і некритичних пристроїв. Сервери бувають різними, відповідно під різні задачі. Програмні сервери створенні для запуску корпоративних додатків, веб-сервери для хостингу веб-додатків. Для управління електронною поштою використовують поштові сервери. Щоб забезпечити управління користувачами і їх доступом використовуються сервери автентифікації та каталогів.

1.2 Захист локальної мережі

Важливо захищати критичну інформацію, яка проходить через локальну мережу комерційного підприємства. Це можуть бути персональні дані клієнтів, інформація про технологію виготовлення продукції та таке інше. Для забезпечення безпеки використовують різні технології. Міжмережеві екрани (Firewalls) забезпечують захист від несанкціонованого доступу та загроз з Інтернету. Антивірусне програмне забезпечення створене для захисту від шкідливого ПЗ. Системи управління доступом (Access Control Systems) надає інструменти для контролю доступу користувачів до ресурсів мережі.

Системи запобігання вторгнень (IPS/IDS) потрібні для моніторингу та захисту мережі від вторгнень. Спочатку IDS класифікувалися за місцем розташування: вони могли бути орієнтовані на захист окремих вузлів (host-based або Host Intrusion Detection System - HIDS) або захищати всю корпоративну мережу (network-based або Network Intrusion Detection System - NIDS)[2].

Найбільшу увагу потрібно сконцентрувати на універсальні NIDS, які підтримують широкий набір комунікаційних протоколів та технології глибокого аналізу пакетів DPI (Deep Packet Inspection). Вони моніторять весь трафік, починаючи з канального рівня, і виявляють широкий спектр мережевих атак, а також спроби неавторизованого доступу до інформації. Часто такі системи відрізняються розподіленою архітектурою та можуть взаємодіяти з різним активним мережевим обладнанням. Потрібно зазначити, що багато сучасних NIDS є гібридними і поєднують кілька підходів. Залежно від конфігурації та

налаштувань вони можуть вирішувати різні завдання, наприклад захист одного вузла або всієї мережі.

Спочатку IDS могли тільки виявляти дії шкідливого програмного забезпечення, роботу сканерів портів, або, скажімо, порушення користувачами корпоративних політик безпеки. При настанні певної події вони повідомляли адміністратора, але досить швидко стало зрозуміло, що просто розпізнати атаку недостатньо її потрібно заблокувати. Так IDS трансформувалася на IPS (Intrusion Prevention Systems) — системи запобігання вторгнень, здатні взаємодіяти з мережевими екранами.

Гарною практикою є поєднання IDS/IPS систем з SIEM-системами. SIEM (Security Information and Event Management) системи необхідні для забезпечення комплексної безпеки інформаційних систем. Вони збирають журнали подій та інші дані з різних джерел, таких як мережеве обладнання, сервери, додатки та безпекові пристрої. На основі зібраних даних аналізують зібрані дані та корелюють події для виявлення підозрілих або шкідливих дій. Це дозволяє визначати складні атаки, які можуть бути пропущені окремими системами. SIEM-системи генерують звіти та надають аналітичну інформацію, що допомагає виявляти тенденції та покращувати безпеку. Тим самим вони допомагають організаціям відповідати нормативним вимогам шляхом забезпечення аудиту та зберігання журналів.

Взаємодія між SIEM та IDS/IPS виглядає наступним чином:

1 Етап 1 - Збір даних(IDS/IPS системи генерують велику кількість даних про мережевий трафік, події та аномалії. SIEM-системи збирають ці дані разом з іншими журналами подій).
2 Етап 2 - Кореляція та аналіз(SIEM-системи корелюють дані з IDS/IPS з іншими даними, такими як журнали подій серверів та додатків. Це дозволяє виявляти складні атаки, які можуть бути непомітними для IDS/IPS).

3 Етап 3 – Реагування на інциденти(Коли SIEM-система виявляє підозрілу активність, вона може ініціювати реагування на інциденти, яке може включати активацію політик IPS для блокування шкідливого трафіку).

4 Етап 4 - Звітування та аудит(Дані з IDS/IPS використовуються для створення звітів та проведення аудиту, що допомагає виявляти слабкі місця та покращувати захист).

Уявімо сценарій, де відбувається підозріла активність на мережі. IDS виявляє аномальну поведінку та генерує подію. Ця подія надсилається до SIEM-системи, яка також отримує інші журнали, наприклад, журнали доступу до серверів та систем аутентифікації. SIEM корелює ці події та визначає, що виявлена активність є частиною більш широкої атаки. На основі цього аналізу SIEM може відправити команду на IPS для блокування підозрілого трафіку, тим самим запобігаючи подальшому поширенню атаки. Таким чином, SIEM-системи та IDS/IPS працюють разом для забезпечення більш надійного та ефективного захисту інформаційних систем, об'єднуючи свої можливості для виявлення та реагування на загрози.

Одним з провідною системою управління інформацією та подіями безпеки є Splunk. Вона надає потужні інструменти для моніторингу, аналізу та захисту даних у реальному часі.

Splunk складається з кількох програм, кожна з яких може містити окремі набори даних, інформаційні панелі та параметри конфігурації[3]. Головний екран містить різноманітні ярлики для типових завдань у системних налаштуваннях і програмах, але для початкового ознайомлення варто почати з програми Search & Reporting, яка наведена на рисунку 1.1.

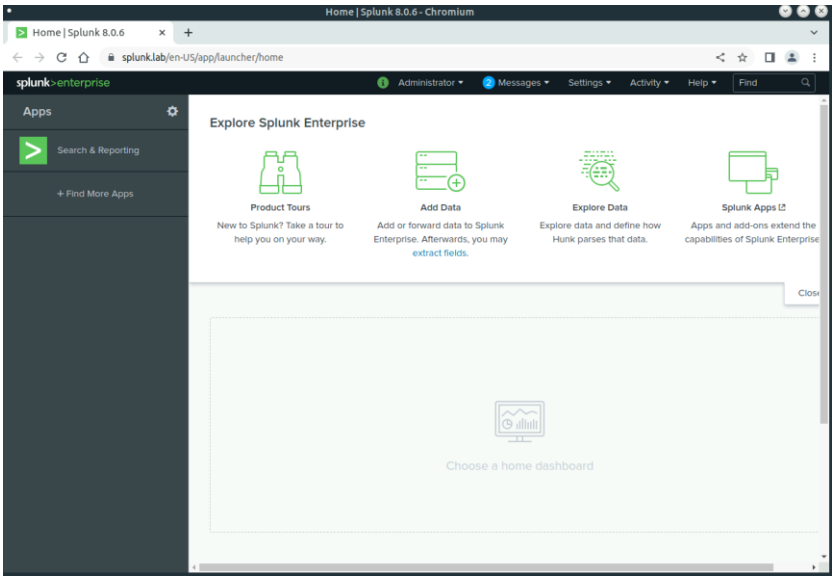

Рисунок 1.1 – Інтерфейс Splunk

Ключовими елементами екрана пошуку є основна панель пошуку(позначено «1»), селектор часу(позначено «2») та список компонентів програми (позначено «3»), ми можемо їх побачити на рисунку 1.2.

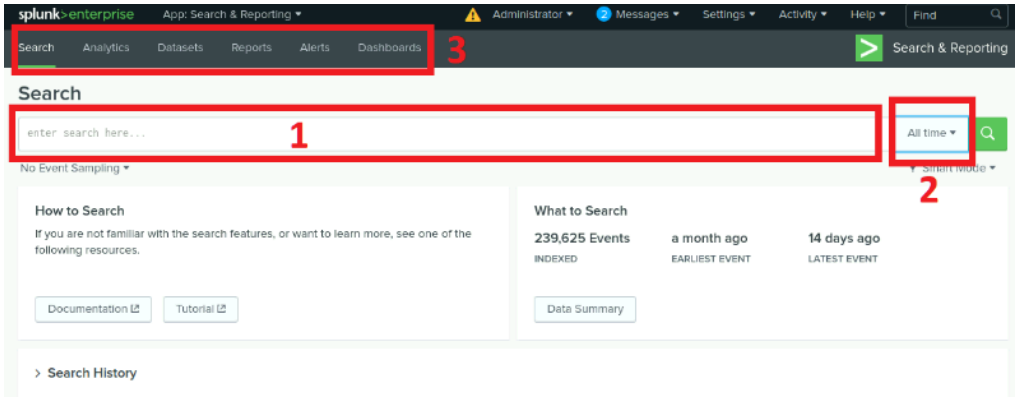

Рисунок 1.2 – Робоча область Splunk

Всередині Splunk відстежує всі події як UTC і автоматично відображатиме їх у форматі часового поясу та мови поточного користувача. Це середовище за замовчуванням має форматування en-US, але його можна змінити, відредагувавши URL-адресу як показано на рисунку 1.3.

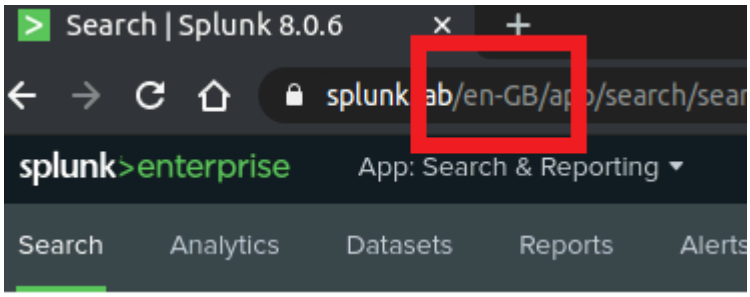

Рисунок 1.3 – URL-адреса

Пошук Splunk може бути таким же простим, як ключове слово або термін, який шукатиметься в усіх корисних навантаженнях подій[4]. Зазвичай використовується для пошуку журналів, які вказують на можливий збій компонента.

На рисунку 1.4 наведено сторінку результатів пошуку. Існують три основні компоненти сторінки результатів пошуку Splunk: хронологічна шкала, що містить підсумок розподілу подій у часі(позначено «1»); панель полів(позначено «2»); перелік основних(позначено «3»).

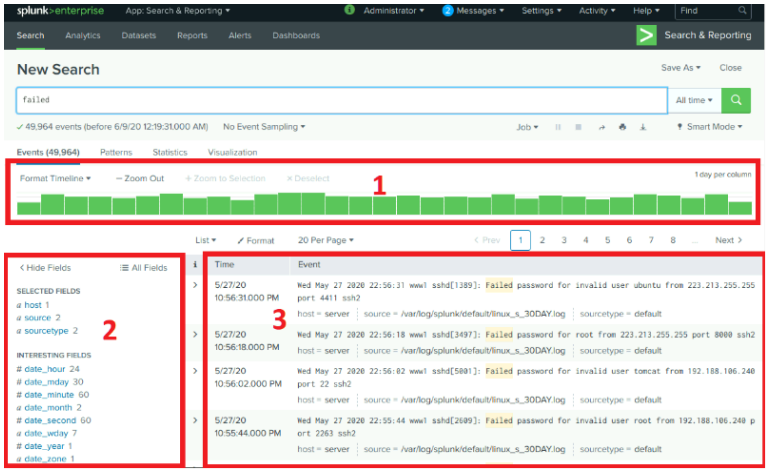

Рисунок 1.4 – Основні компоненти сторінки прошуку

Інтерфейс користувача також виділяє три поля, які можна знайти в кожній події Splunk:

- host - зазвичай ім'я хоста, IP-адреса або повне доменне ім'я хоста мережі, з якого походить подія.

- source - ім'я файлу або потоку, з якого походить подія. Для даних, які відстежуються з файлів і каталогів, це повний шлях, наприклад /archive/server1/var/log/messages.0. Для мережевих джерел даних це протокол і порт, наприклад UDP:514.
- sourcetype - формат вхідних даних, з якого вони походять, наприклад access_combined або cisco_syslog. Тип джерела визначає спосіб форматування даних.

1.2 Види кібератак

Кібератаки на мережу комерційного підприємства можуть бути різноманітними, і для кожної з них існують специфічні методи захисту. Найпоширенішими видами атаки є: фішинг, атаки на відмову в обслуговуванні(DDoS), шкідливе ПО та цільові атаки(APT)[5]. Розглянемо більш детально кожний вид атаки та методи захисту від них.

Фішинг – це атака, при якій зловмисник намагається обманом змусити користувачів розкрити конфіденційну інформацію(логіни, паролі, номери кредитних карт та інше) через електронні листи, повідомлення чи веб-сайти, які маскуються під справжні та довірені джерела. При даній атаці хакери спираються на страх та/або неуважність людини. Наприклад, користувач отримав електронний лист, який містить логотипи відомих компаній, банків, стиль оформлення листа. Або користувачу надходить повідомлення, що якщо терміново не «оновити» дані карточку, то вона буде заблокована. Необізнана людина скоріше за все поведеться на даний тип атаки. Тому для захисту від даного типу атаки потрібно навчати співробітників розпізнавати фішингові повідомлення, використовувати фільтри спамів та антивірусне програмне забезпечення.

Атака на відмову в обслуговуванні або DDoS – це вид атаки, під час якого велика кількість зловмисних запитів надсилається до сервера, веб-сайту або мережі, з метою перевантажити їх і зробити недоступними для законних

користувачів. В атаці DDoS беруть участь численні зламані комп'ютери або інші пристрої, які об'єднуються в мережу ботнет і одночасно надсилають запити до цільової системи. Основна мета DDoS-атаки — перевантажити ресурси цільової системи (сервери, мережеві канали тощо) до такої міри, що вони не зможуть обробляти законні запити від користувачів. Атака на відмову в обслуговуванні поділяється на атаки на рівні мережі(UDP-флуд, ICMP-флуд), на рівні транспортного і сеансового рівня(SYN-флуд, який націлений на відкриття великої кількості напіввідкритих з'єднань.) та на рівні додатків(HTTP-флуд, коли зловмисники намагаються перевантажити конкретний додаток або сервіс великою кількістю запитів.) Для захисту від даного типу атаки потрібно використовувати мережеві пристрої та сервіси, які можуть розпізнавати і фільтрувати зловмисний трафік. Бажано застосовувати методи розподілу трафіку, щоб перенаправити і збалансувати навантаження на мережу. Гарною практикою є впровадження хмарних рішень для масштабування і поглинання атак.

Шкідливе програмне забезпечення є програмами, які створені з метою пошкодження, втручання або отримання несанкціонованого доступу до комп'ютерних систем, пристроїв або мереж[6]. Існує багато видів шкідливого ПО, наведемо перелік розповсюдженіших атак на мережі:

- Черв'яки самостійно розповсюджуються через мережу, використовуючи уразливості в програмному забезпеченні. Вони можуть швидко заражати велику кількість пристроїв у мережі, що робить їх дуже ефективними для атак на мережеву інфраструктуру.
- Ботнети — це мережі зламаних пристроїв, які зловмисники можуть контролювати віддалено. Ці мережі часто використовуються для масових DDoS-атак, розсилки спаму або крадіжки даних. Зокрема, ботнети застосовуються для перевантаження мережевих ресурсів або атак на конкретні веб-сайти чи сервери.

- Трояни можуть використовуватися для отримання несанкціонованого доступу до мережевих ресурсів. Наприклад, трояни можуть встановлювати бекдори (backdoors), які дозволяють зловмисникам віддалено контролювати заражені системи та використовувати їх для подальших атак на мережу.
- Хоча основною метою шпигунського ПО є збір інформації, ці програми також можуть використовуватися для виявлення вразливостей у мережі, збирання паролів та інших конфіденційних даних, що потім можуть бути використані для подальших атак.
- Вимагачі все частіше використовуються для атак на мережі, особливо корпоративні. Вони можуть шифрувати важливі файли на мережевих ресурсах і вимагати викуп за їх розблокування. Такі атаки можуть призводити до серйозних перебоїв у роботі організацій.

Ці види шкідливого програмного забезпечення часто використовуються в поєднанні для досягнення більшого ефекту. Наприклад, трояни можуть встановлювати черв'яків або руткіти, які потім використовуються для розширення ботнетів або виконання DDoS-атак. Методами захисту є регулярне оновлення антивірусного ПЗ та операційних систем, обмеження прав доступу користувачів до критично важливих систем. Кращою практикою є перевірка всіх підозрілих програм через sandbox.

Останній тип атаки це цільові атаки. Вони спрямовані на конкретні організації з метою шпигунства або саботажу, використовуючи тривалі та складні атаки на мережу[7]. У спеціалізованій літературі ця атака має три важливих характеристики, яка виділяє її серед інших: Advanced – мається на увазі, що зловмисники використовують складні та часто новаторські техніки, щоб проникнути в системи і залишатися непоміченими; Persistent – означає, що атаки можуть тривати місяцями або навіть роками. У цей час хакери прагнуть залишатися непоміченими якомога довше, щоб мати можливість поступово

збирати дані або виконувати свої завдання; Targeted - тобто цілеспрямовані та зазвичай спрямовані на конкретні організації або навіть окремих осіб, які мають цінну інформацію. Це можуть бути урядові установи, великі корпорації, військові об'єкти або дослідницькі інститути. Захист від даної кібератаки потребує багато заходів захисту, таких як використання брандмауерів та систем детектування та запобігання вторгненням (IDS/IPS), обмеження доступу до критично важливих даних і ресурсів лише необхідним особам та системам та використання всіх тих методів захисту, які описані вище для кібератак.

2. ТЕХНОЛОГІЯ CANARYTOKENS

Canary Tokens – це невеликі шматочки коду або фіктивні ресурси, які створюються для того, щоб виявити та відстежити несанкціонований доступ або інші види зловмисної діяльності в системах інформаційної безпеки. При спрацьовуванні Canary Token генерує сповіщення, що дозволяє адміністраторам безпеки оперативно реагувати на загрози.

Наприклад, створюється URL-адреса, яка при нормальних умовах не повинна бути відвідана. У цьому випадку коли відвідує таку URL-адресу, адміністратору надходить сповіщення про потенційний несанкціонований доступ. Також є спеціально створені електронні листи або email-адреси. При надсилані листа на цю адресу або відкриванні листа, система генерує сповіщення. Можна замаскувати документи Microsoft Word або Excel під легальні, наприклад підписати їх passwords.docx або emails.xlsx. У разі відкриття таких документів генерується сповіщення[8].

Хакер, потрапивши в скомпрометовану систему, починає шукати всі цікаві файли. Як злодій, який проникнув до квартири, починає відкривати всі ящики, шкафи у пошуках цінностей або грошей. Кіберзлодій заздалегідь не знає, які саме дані є цінністю, але з великою ймовірністю він перевірить усі можливі варіанти. Тому важливо, щоб пастки виглядали як реальні дані та були максимально привабливими для атакуючого. Чим раніше власник помітить спрацювання пастки, тим швидше зрозуміє, що систему атакують і зможе відреагувати.

Для створення таких пасток є онлайн-сервіс https://canarytokens.org/. Він дозволяє створення власних ханітокенів для виявлення злому[9]. Також підтримує кілька варіантів тригерів і дозволяє згенерувати готовий тригер із прив'язаною поштою, куди прийде повідомлення, якщо тригер спрацював. Сервіс повністю безкоштовний, а згенерувати тригер можна без реєстрації. Існує ще self hosted версія для тих, хто хоче тримати секрети на своїй інфраструктурі.

Цікавим прикладом є ханітокен з тригером за email-адресом. Якщо на згенеровану поштову скриньку прийде будь-який лист, тригер спрацює. Корисно використовувати для моніторингу витоку баз даних email-адрес, контакт-листів.

Наприклад, можна додати цю адресу до списку своїх контактів. Якщо програма запитає доступ до списку контактів, то вона цілком може злити вашу записну книжку на телефоні. Ви дізнаєтесь про це, вам прийде сповіщення. Або можна. Або, якщо керівництво підприємства хоче дізнатися звідки витікають дані про контакти, то можна всім співробітникам додати до записників на комп'ютері, телефоні, email-клієнті різні тригерні адреси. Це дасть можливість відстежити, звідки витікають контакти.

URL-тригер спрацює у випадку, коли за посиланням був виконаний GET, POST або HEAD запит. Це спричинить спрацювання тригера. Окрім звичайного застосування, можна використовувати в скриптах та для перевірки парсерів, які переходять за посиланнями для відображення прев'ю вмісту. Така практика присутня в месенджерах: достатньо написати посилання в полі введення, що по ньому було виконано перехід із серверів месенджера.

Цікавим ханітокеном є при роботі тригера через DNS resolve. DNS resolve (або DNS розв'язання) — це процес перетворення доменного імені, яке зрозуміле людині, у відповідну IP-адресу, яку використовують комп'ютери для взаємодії в мережі[10]. DNS (Domain Name System) є системою, яка відповідає за цей процес. Тригер спрацює у випадку, якщо хтось запитає IP-адресу згенерованого піддомену, який генерується спеціально таким, щоб його не можна було випадково вгадати. Таким чином, виключаються випадкові спрацьовування.

Для боротьби з фішингом, можна використати тригер клонування сайту. Це може бути простий скрипт для веб-сторінок, що спрацьовує, якщо сторінка відкрита не з вашого домену.

Сервіс canarytokens.org підтримує інші тригери, такі як хук для SVN, веб-редирект, читання QR-коду, API-ключ Slack та інші. Всі вони використовують

схожий принцип, і за бажання можна самостійно придумати власний тригер, маючи в арсеналі DNS-ім'я, посилання з картинкою, поштову адресу та API-ключі від популярних сервісів. Механізми роботи описані у документації.

3. ТЕХНОЛОГІЯ HONEYPOT

3.1 Характеристика honeypot

Honeypot – це ресурс, який створений для заманювання хакера з метою раннього попередження атаки на ІКС та вивчення спроб проникнення в ІКС. Частіше за все, воно складається з комп’ютера або віртуальної машини, програм та даних, які імітують поведінку реального пристрою. З метою заманювання хакера, honeypot є частиною мережі, але є ізольованою від інших пристроїв в локальній мережі. Основна мета honeypot полягає у виявленні та аналізі атак, зборі інформації про методи та тактики кіберзловмисників, а також у запобіганні потенційним загрозам[11]. Це важливий інструмент в арсеналі засобів інформаційної безпеки, що дозволяє комерційним підприємствам підвищити рівень захисту своїх інформаційно-комунікаційних систем. Honeypot функціонує, залучаючи зловмисників до своєї системи, де вони можуть здійснювати атаки або спроби доступу[12]. Оскільки honeypot не виконує реальних бізнес-функцій, будь-яка взаємодія з ним вважається підозрілою. Всі дії зловмисників ретельно реєструються, аналізуються та використовуються для поліпшення системи безпеки[13]. За типом honeypot поділяють на статичні та динамічні.

3.2 Статистичний honeypot

Статичний honeypot залучає зловмисників до своєї системи, де вони можуть здійснювати атаки або спроби доступу. Всі дії зловмисників реєструються для подальшого аналізу. Враховуючи статичну природу, цей тип honeypot завжди відповідає на запити однаково, без адаптації до дій зловмисника.

Основними характеристиками статичних honeypots є фіксована конфігурація, легкість у налаштуванні. Це означає, що статичні містять незмінні налаштування та імітує конкретну версію сервісу або системи з відомими вразливостями. Тобто у відповідь на дії зловмисників не змінює свою поведінку. Імітує обмежений набір сервісів або функцій, що робить їх менш ресурсомісткими. Придатні для виявлення базових атак, таких як сканування портів або прості експлойти.

Серед переваг можна виділити простоту у використанні та мінімальні вимоги до ресурсів. Фактично вона ефективна для виявлення базових атак.

Недоліками є обмеженість у функціональності та прогнозованість та нездатність виявляти складні атаки. Це впливає на обмеженість аналізу через простоту відповідей.

Статичні honeypot є корисним інструментом для виявлення та аналізу базових атак у комерційних підприємствах. Вони забезпечують простоту у використанні, мінімальні ризики та ефективність у виявленні простих загроз. Однак, їх обмежена функціональність та прогнозованість означають, що для комплексного захисту інформаційних систем необхідно використовувати їх у поєднанні з іншими засобами безпеки та більш складними типами honeypot.

Прикладами статичних honeypot є Honeyd та Dionaea.

Honeyd — це програмне забезпечення для створення віртуальних хостів, яке дозволяє емуляцію операційних систем, мережевих сервісів та вразливостей

з метою дослідження атак і захисту мереж. Основна мета — імітація реальних хостів та служб для збору інформації про атаки і підвищення безпеки мережі. ПЗ дозволяє створювати привабливі цілі для хакерів, відволікаючи їх від реальних систем та фіксує всі спроби доступу та атаки на віртуальні хости, що дозволяє адміністраторам мережі збирати інформацію про методи, інструменти та техніки, що використовуються зловмисниками[14]. Відповідно може служити системою раннього виявлення загроз, попереджаючи адміністрацію мережі про можливі атаки. Адміністратори можуть використовувати Honeyd для тестування та оцінки ефективності існуючих систем безпеки та виявлення потенційних вразливостей. Також потужним функціоналом є можливість створювати віртуальні мережі для тестування нових програмних рішень або конфігурацій без необхідності використання фізичних ресурсів. Створюючи багато віртуальних хостів, Honeyd ускладнює зловмисникам виявлення справжніх критично важливих систем і знижує ризик успішних атак на реальні системи, відволікаючи увагу зловмисникі

Dionaea — це інструмент метою якого є виявлення та збір інформації про шкідливе програмне забезпечення та мережеві атаки. Dionaea спеціалізується на уловлюванні експлойтів, що спрямовані на уразливості у мережевих сервісах, і зберіганні зразків шкідливого ПЗ для подальшого аналізу. Вона підтримує великий набір мережевих протоколів, включаючи HTTP, FTP, TFTP, SMB, MSSQL, MySQL, SIP та інші[15]. Це дозволяє йому імітувати різноманітні мережеві сервіси, які часто стають цілями атак. Також модульна архітектура Dionaea дозволяє додавати нові протоколи та функціональності без значних змін в основному коді. Це робить його гнучким і легко розширюваним.

3.3 Динамічний honeypot

Динамічний honeypot імітує реальні системи та сервіси, дозволяючи зловмисникам взаємодіяти з ними на глибокому рівні. Цей тип honeypot адаптується до дій зловмисників, забезпечуючи більш реалістичний досвід. Всі

дії зловмисників ретельно реєструються та аналізуються для виявлення нових загроз і методів атак.

Динамічні honeypot відрізняються від статичних здатністю змінювати свою поведінку у відповідь на дії зловмисників. Виникає Імітація реальних систем та сервісів, надаючи зловмисникам більше можливостей для взаємодії. Це надає змогу збирати детальну інформацію про методи та тактики зловмисників. Але даний тип вимагає більше ресурсів для налаштування та підтримки.

Статичні та динамічні honeypot мають свої особливості, що робить їх корисними в різних контекстах інформаційної безпеки. Статичні honeypot є простими у використанні та підходять для виявлення базових атак, тоді як динамічні honeypot забезпечують глибокий аналіз і здатність виявляти складні атаки. Для ефективного захисту інформаційних систем часто рекомендується використовувати комбінацію обох типів honeypot, що дозволяє максимізувати переваги кожного з них та мінімізувати їхні недоліки.

Найбільш поширеними типами динамічних honeypot є Kippo та Cowrie.

Kippo спеціально розроблений для емуляції SSH-сервісу, з метою відстеження та аналізу атак, спрямованих на отримання несанкціонованого доступу до системи через SSH. Kippo був популярним інструментом для дослідження методів, які використовують зловмисники для вторгнення в системи, а також для збору інформації про їхні дії після отримання доступу. Хоча Kippo був популярним інструментом, він має свої обмеження, такі як відсутність підтримки нових протоколів і деякі вразливості[16]. Це привело до створення його наступника, Cowrie, який є розширеною та покращеною версією Kippo з додатковими функціями та можливостями.

Основними функціями та можливостями Cowrie є імітація SSH та Telnet сервісів, що дозволяє відслідковувати спроби підключення до системи, включаючи зловмисні дії, такі як брутфорс атаки або використання вкрадених

облікових даних; запис всіх команд, які введені зловмисниками під час сеансу, а також відповіді системи на ці команди[14]. Це дозволяє детально аналізувати поведінку зловмисників. Віртуальна файлова система дозволяє створити реалістичне середовище для зловмисників. Вона може включати фальшиві файли, папки та сервіси, які виглядають як справжні. Хакери можуть завантажувати файли до системи, що дозволяє отримати зразки шкідливого ПЗ для подальшого аналізу. Cowrie можна інтегрувати з іншими системами моніторингу та аналізу безпеки для централізованого збирання та аналізу логів[17].

Cowrie є потужним інструментом для імітації вразливих систем, який дозволяє збирати цінну інформацію про дії зловмисників і використовувати ці дані для підвищення безпеки мереж.

3.4 Рівень взаємодії

Honeypot класифікуються за рівнем взаємодії, який визначає, наскільки глибоко зловмисники можуть взаємодіяти з імітованою системою. Існують три основні рівні взаємодії: низький, середній та високий. Кожен з них має свої особливості, переваги та недоліки:

1. Низький рівень взаємодії імітує базові сервіси та відповідає на прості запити. Зловмисники можуть лише здійснювати базові дії, такі як сканування портів або спроби підключення. Через обмежену функціональність такі honeypot менш уразливі до компрометації. Перевагою є легконалаштовуваність та мінімальний запит на ресурси. Також ймовірність набагато менша того, що зловмисники зможуть використовувати honeypot для атак на інші системи. Недоліком є недостатня можливість для аналізу через просту взаємодію зловмисника. І зловмисники з досвідом можуть досить швидко зрозуміти, що це honeypot.
2. Середній рівень взаємодії вже імітує більш складні сервіси, але ще не є повноцінною операційною системою. Відповідно зловмисники можуть

виконувати більш складні дії, що призводить до збільшення слідів та даних, які можна використати для аналізу методів атаки. Недоліком є більша затратність на потужності порівняно з низьким рівнем взаємодії та час для встановлення конфігурації.

3. Високий рівень взаємодії має імітувати реальну операційну систему з сервісами, надаючи зловмисникам можливість виконувати будь-які дії, як на реальній системі. Зловмисники залишають багато даних, що дозволяє детально вивчати їх методи та тактики. Це максимально правдоподібні для зловмисників цілі, що робить їх привабливими. Але виникає підвищений ризик компрометації та можливості атаки на інші системи в локальній мережі.

4. ПРОГРАМНЕ РІШЕННЯ ТА ТЕСТУВАННЯ

4.1 Створення та налаштування honeypot

У локальній мережі комерційного підприємства буде знаходитись хост, на якому працює віртуальна машина з операційною системою Kali Linux. На цій віртуальній машині буде встановлено honeypot Cowrie та розміщені Canary Tokens в різних частинах файлової системи.

Оновимо нашу систему та встановимо необхідні пакети для роботи з Cowrie (рис. 4.1).

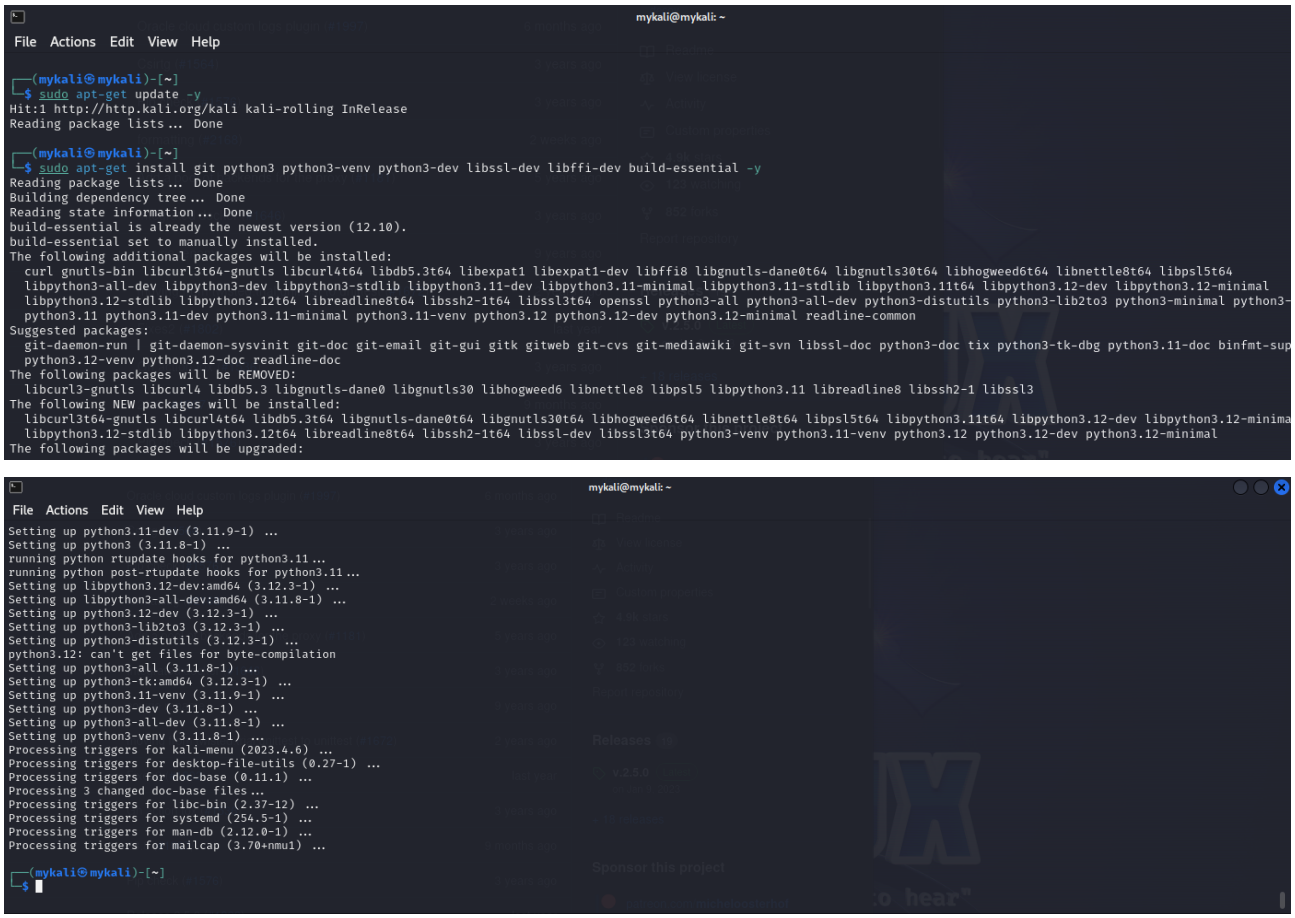

Рисунок 4.1 – Оновлення системи та встановлення необхідних пакетів

Попередньо створимо користувача cowrieрс та перейдемо в нього для подальшого налаштування. Далі завантажуємо останню версію cowrieрс з GitHub (рис. 4.2 – 2.3).

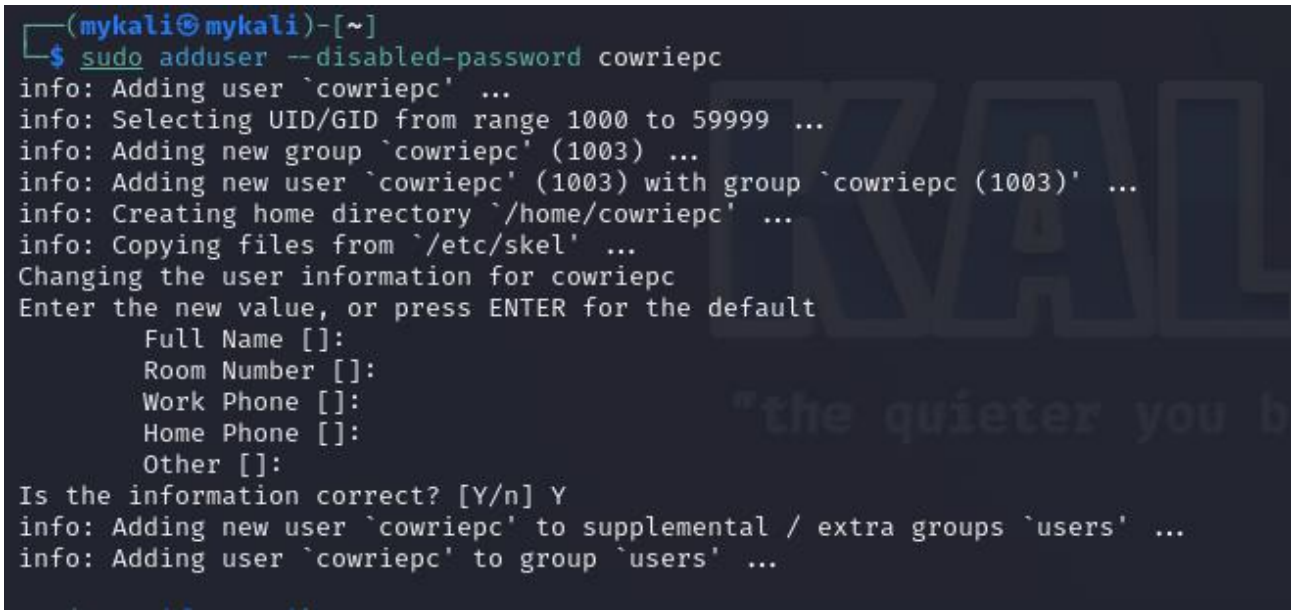

Рисунок 4.2 – Створення користувача

Рисунок 4.3 – Встановлення Cowrie

Тепер перейдемо до налаштування віртуального середовища для honeypot(рис. 4.4). Переходимо в потрібну директорію /home/cowriepc/cowrie. Далі потрібно створити нове віртуальне середовище Python у директорії cowrie-env, використовуючи інтерпретатор Python3. Віртуальне середовище дозволяє нам ізолювати залежності проєкту від інших проєктів на системі. Після чого активуємо віртуальне середовище, яке було створене на попередньому кроці. Після активації віртуального середовища, всі команди Python та pip будуть виконуватися у контексті цього середовища. Оновлюємо пакет pip до останньої

версії (рис 4.4 – 2.6). Це важливо для забезпечення сумісності та отримання останніх покращень та виправлень. Тепер необхідно встановити всі пакети, перелічені у файлі requirements.txt, у віртуальне середовище. Прапор --upgrade забезпечує, що пакети будуть оновлені до останньої доступної версії.

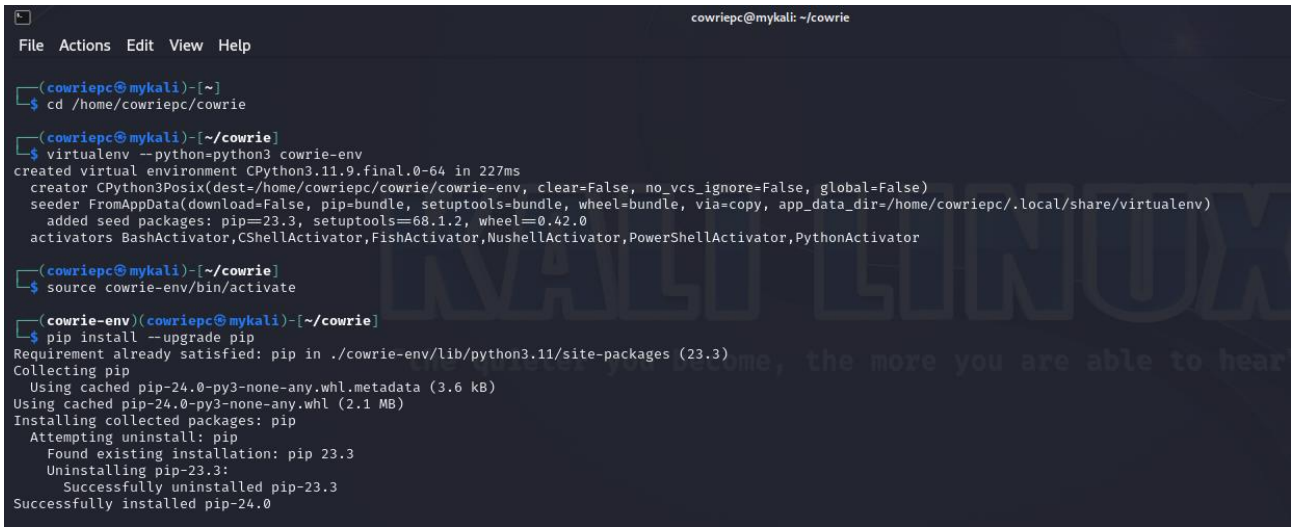

Рисунок 4.4 – Налаштування віртуального середовища

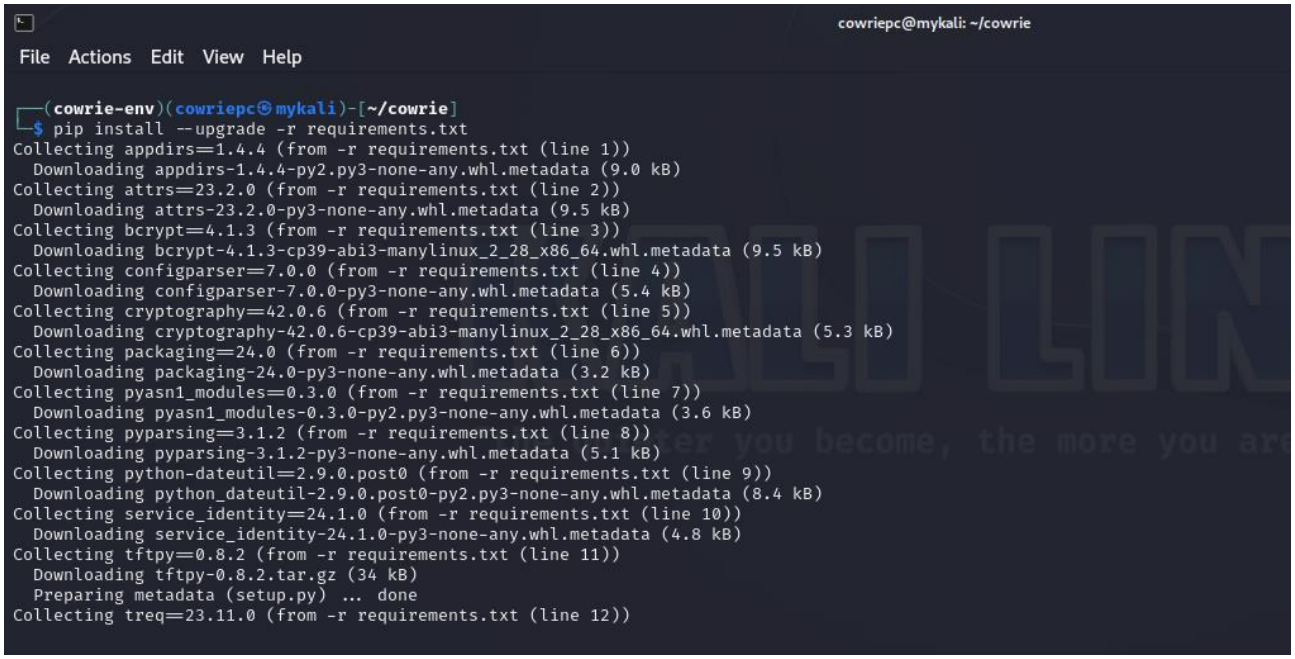

Рисунок 4.5 – Встановлення необхідних пакетів

Рисунок 4.6 – Встановлення необхідних пакетів

Робимо копію файлу налаштувань, але з ім’ям cowrie.cfg, який необхідно відредагувати (рис. 4.7).

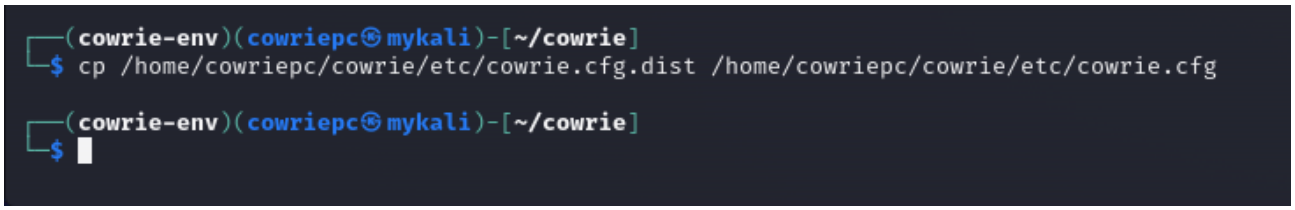

Рисунок 4.7 – Створення копії

Файл cowrie.cfg містить конфігурацію для Cowrie. Для ввімкнення Telnet чи змінити ім’я хоста сервера, потрібно вносити нові дані в даному файлі.

Наприклад, змінимо назву хоста(щоб воно не виглядало як загальний Honeypot Cowrie). Для цього відредагуємо рядок імені хосту (рис. 4.8 – 2.9). У файлі конфігурації знаходимо поле hostname та вносимо нове значення.

Рисунок 4.8 – Процес зміни назви хоста

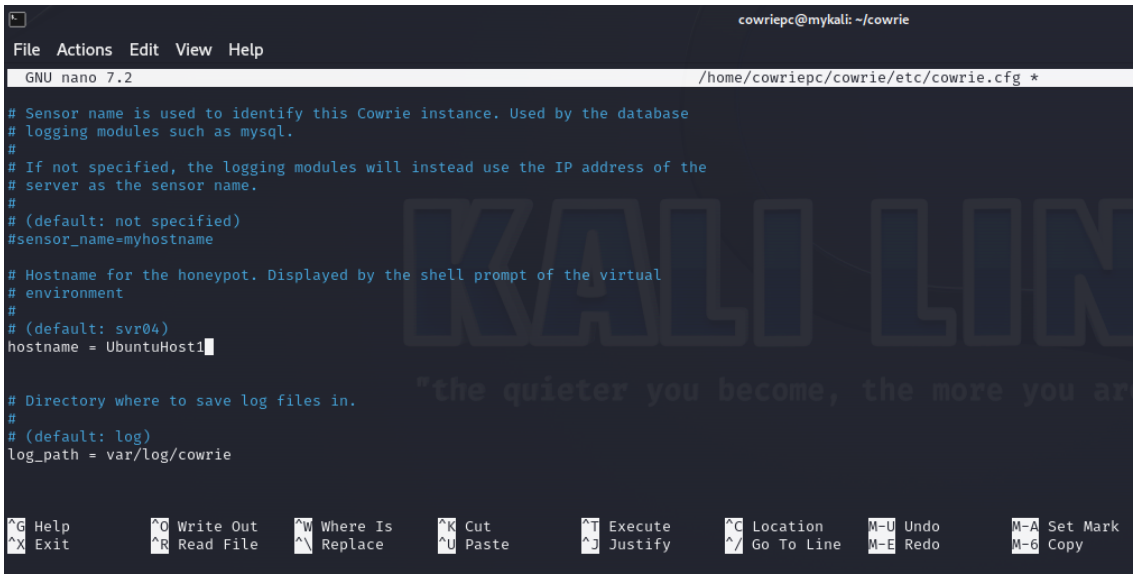

Рисунок 4.9 – Редагування файлу

Щоб Honeypot прослуховував порт, внесемо наступні зміни: знаходимо поле listen_endpoints та вносимо відповідні зміни (рис. 4.10).

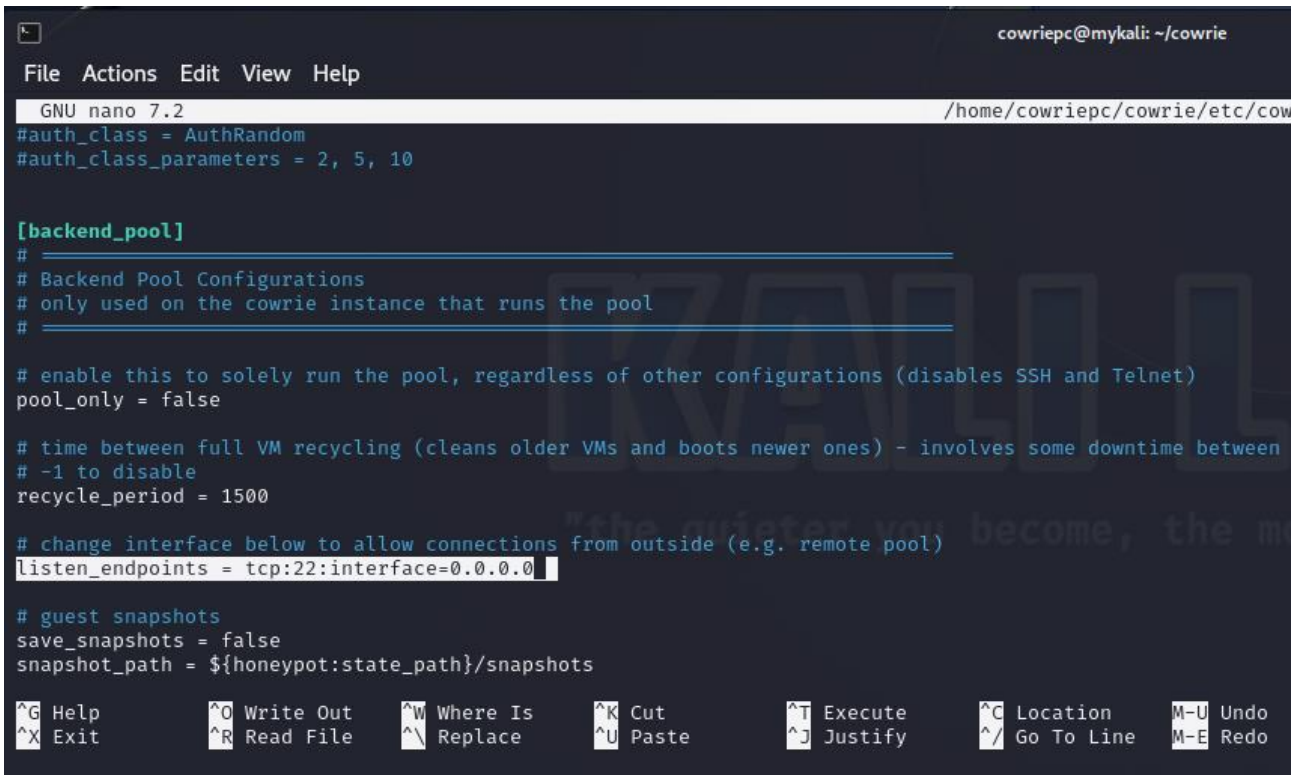

Рисунок 4.10 – Редагування файлу

Після чого встановлюємо права для користувача, який не має права root, щоб він міг слухати порт 22 (заблокований за замовчуванням, тому неможливо

запустити Cowrie як root, тому це буде потрібно). Перейдемо до встановлення пакета authbind, який дозволяє призначати некореневим користувачам права на використання зарезервованих портів. Далі створюємо порожній файл з ім'ям 22 у каталозі /etc/authbind/byport/. Ім'я файлу відповідає номеру порту, на який буде надано доступ. Наступним кроком змінюємо власника файла 22 на користувача та групу cowriepc. Це необхідно для того, щоб тільки цей користувач мав доступ до файлу. Також змінюємо права доступу до файлу 22, встановлюючи їх на 770. Тепер власник файлу та члени групи можуть читати, писати та виконувати файл, а інші користувачі не мають жодного доступу (рис. 4.11).

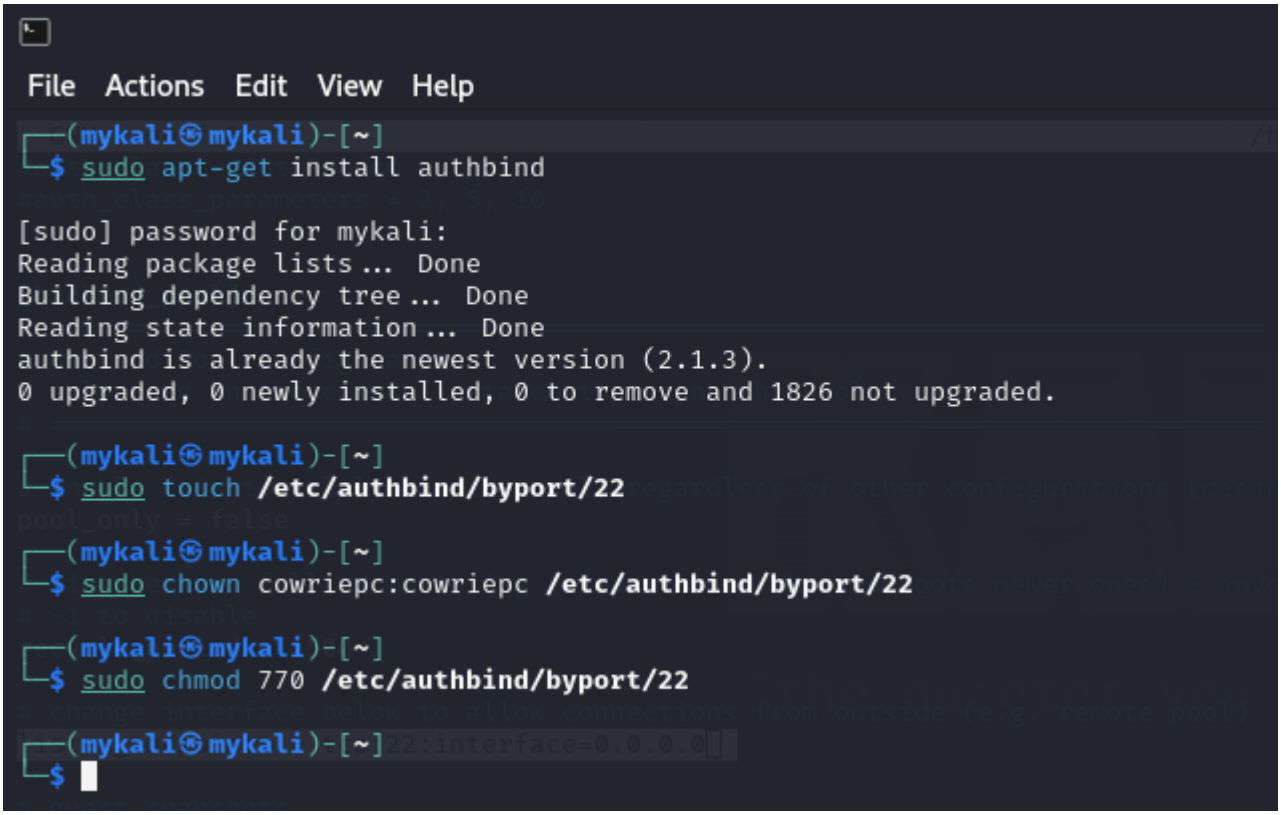

Рисунок 4.11 – Налаштування Cowrie

Також необхідно відредагувати файл /etc/ssh/sshd_config, щоб honeypot слухав справжній SSH-порт і виконуємо перезапуск служби ssh (рис. 4.12 – 2.13).

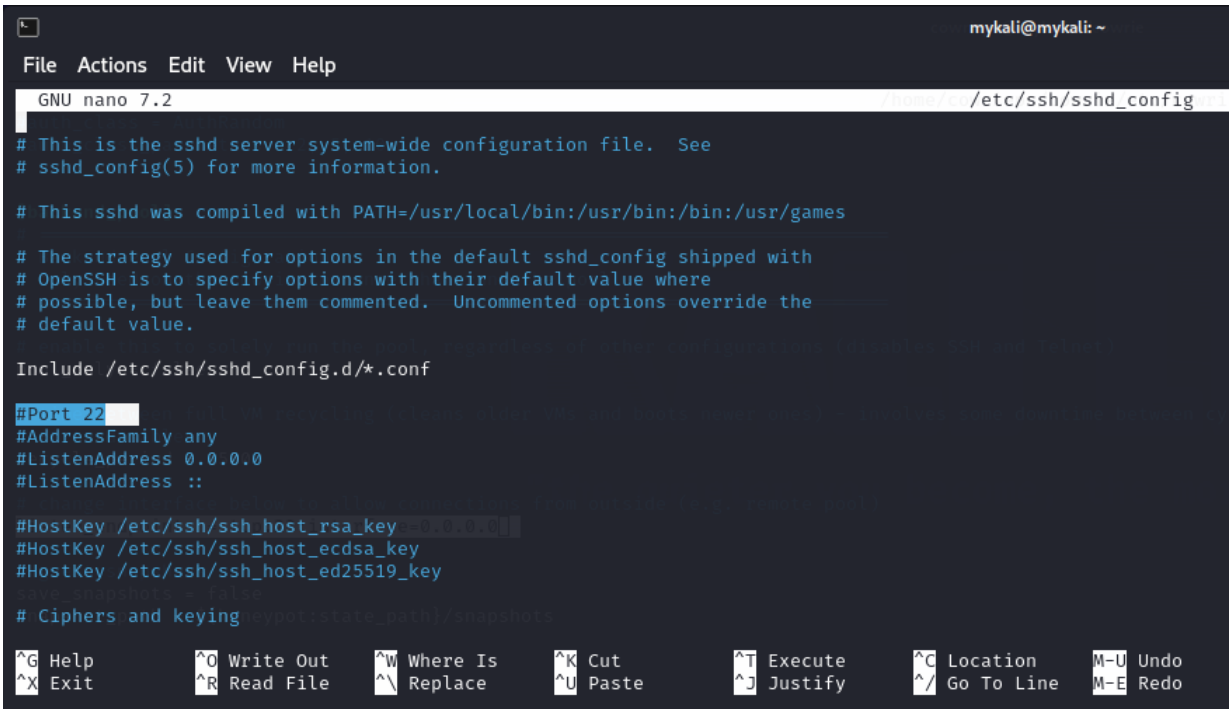

Рисунок 4.12 – Редагування файлу

Рисунок 4.13 – Перезапуск служби ssh

Далі встановлюємо Supervisor. Це інструмент для керування процесами, який дозволяє контролювати та автоматично перезапускати сервіси (рис. 4.14).

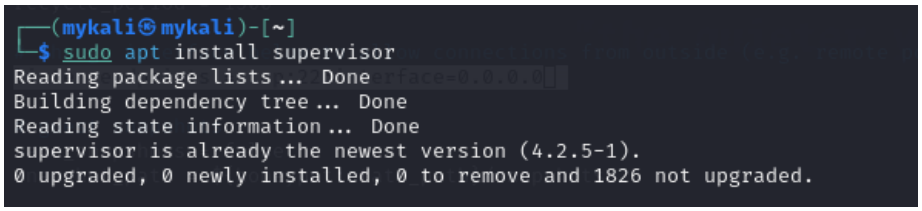

Рисунок 4.14 – Встановлення Supervisor

Тепер необхідно створити файл конфігурації для програми Cowrie в директорії Supervisor (рис. 4.15 – 2.16).

Рисунок 4.15 – Процес створення файлу

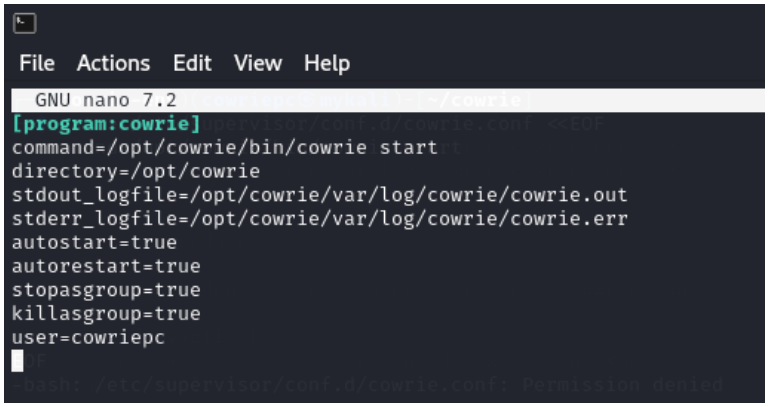

Рисунок 4.16 – Процес створення файлу

Після внесених змін перезапускаємо Supervisor і перевіряємо його статус(рис. 4.17).

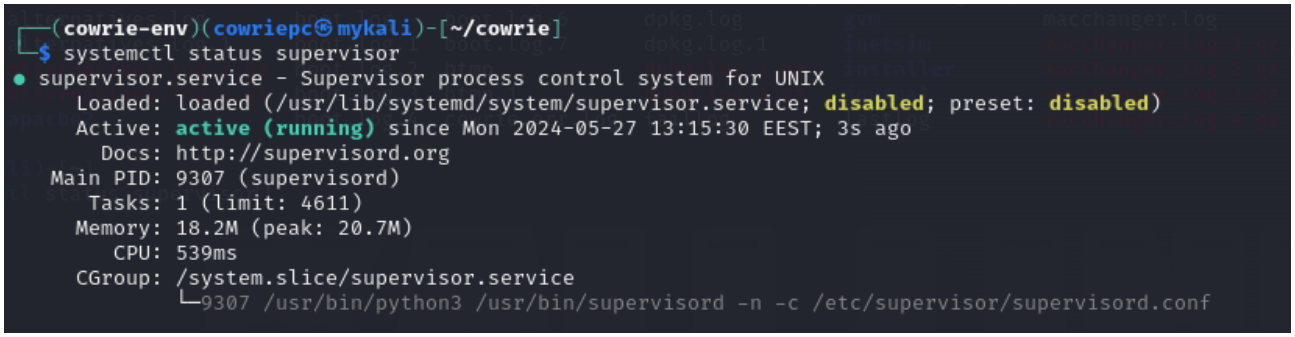

Рисунок 4.17 – Перевірка роботи Cowrie

Перевіримо, що кожен порт прослуховує правильний процес (рис. 4.18).

Рисунок 4.18 – Перегляд портів

Спочатку завантажуємо архів Splunk[3, 18] версії 8.2.6 через команду wget. Потім розпаковуємо архів (рис. 4.21).

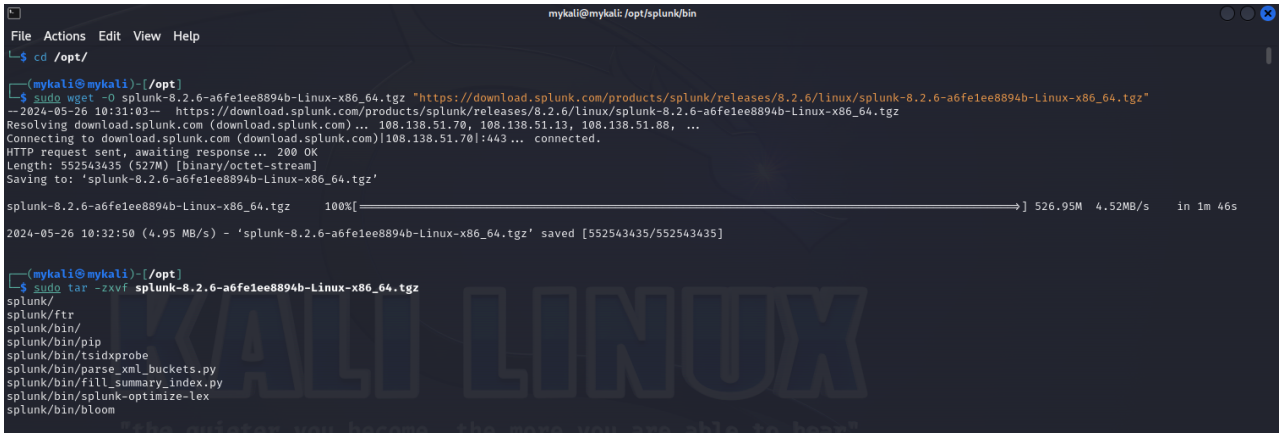

Рисунок 4.21 – Завантаження файлу та розпакування

Далі запускаємо програму та приймаємо ліцензійну угоду(рис. 4.22). створюємо акаунт (рис. 4.23).

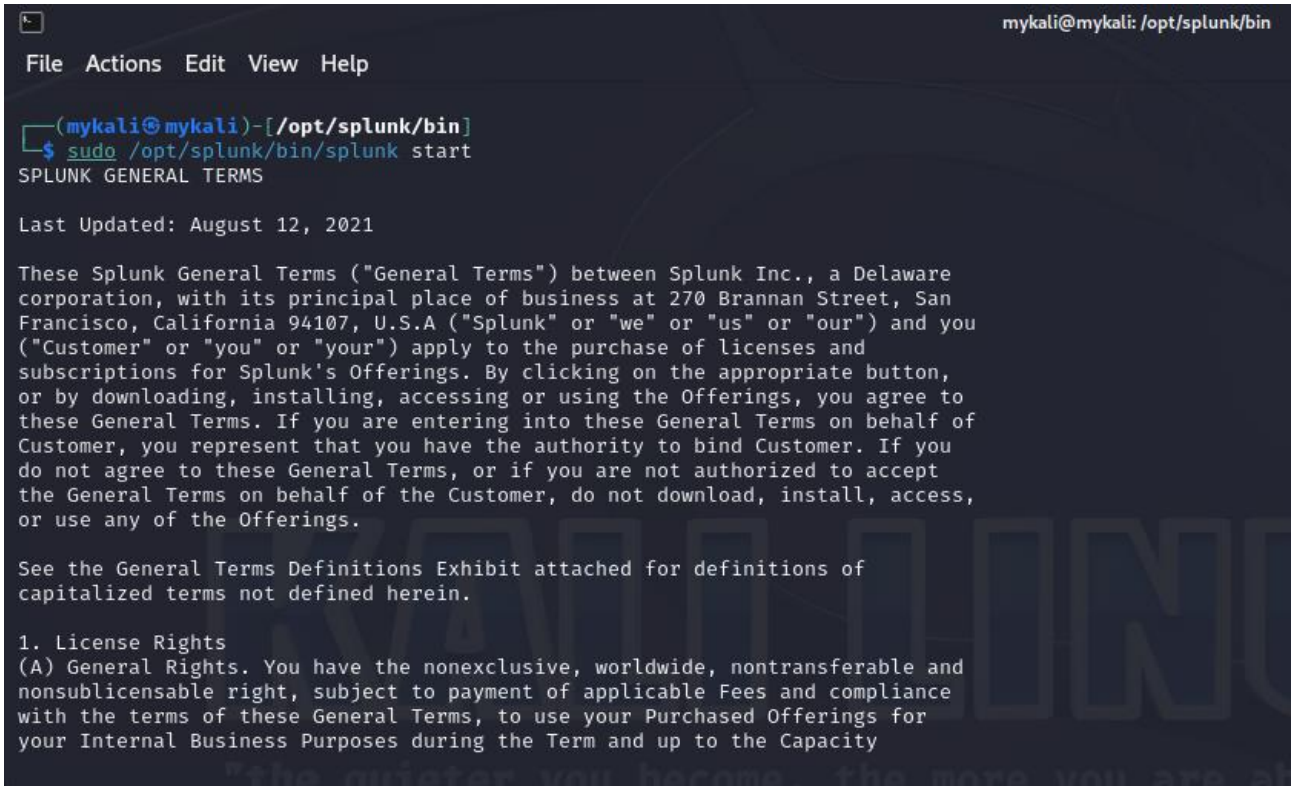

Рисунок 4.22 – Створення акаунту

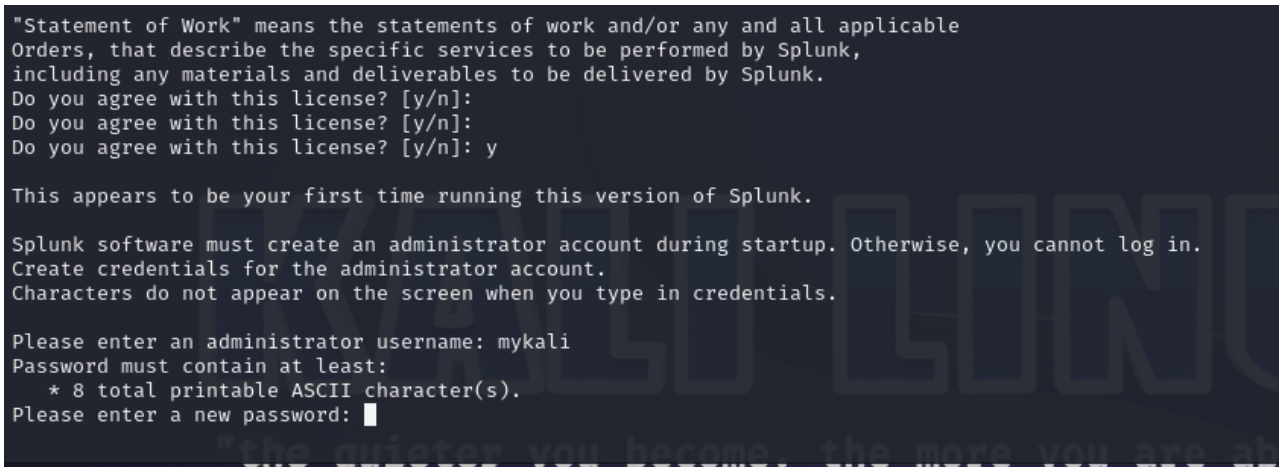

Рисунок 4.23 – Процес налаштування

Після створення акаунту, розгортається Splunk. Веб-сервер Splunk в нашому випадку знаходиться за посиланням http://mykali:8000 (рис. 4.24)

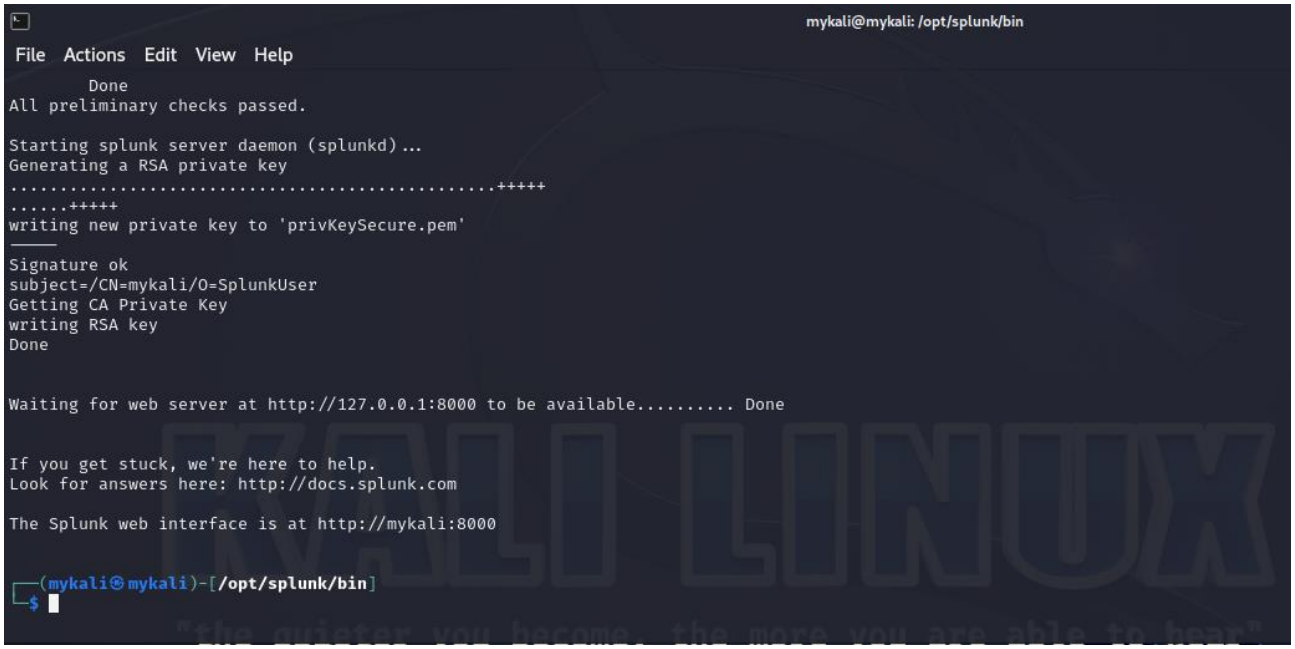

Рисунок 4.24 – Результат налаштування

Щоб отримати доступ до графічного інтерфейсу Splunk, нам потрібно перейти за посиланням та ввести логін і пароль від акаунту, який створено на попередньому кроці (рис. 4.25).

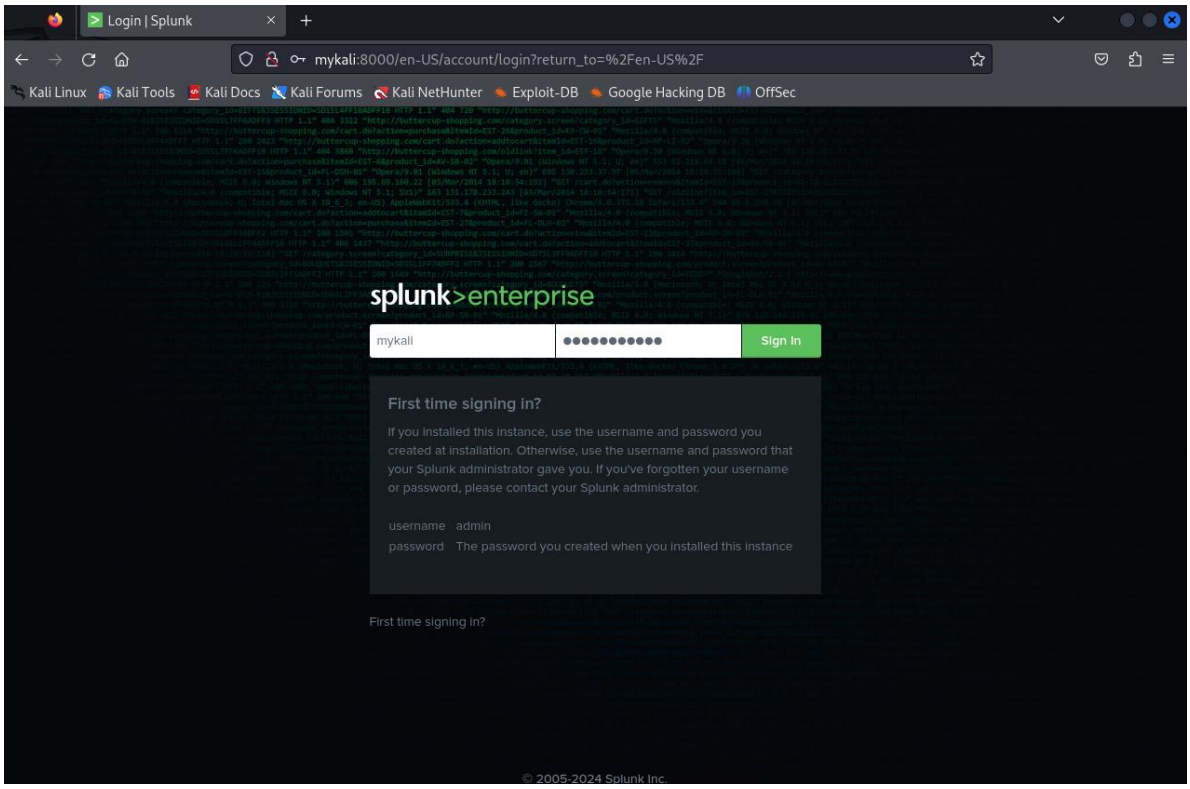

Рисунок 4.25 – Інтерфейс Splunk

Після чого переходимо на головний екран Splunk (рис. 4.26).

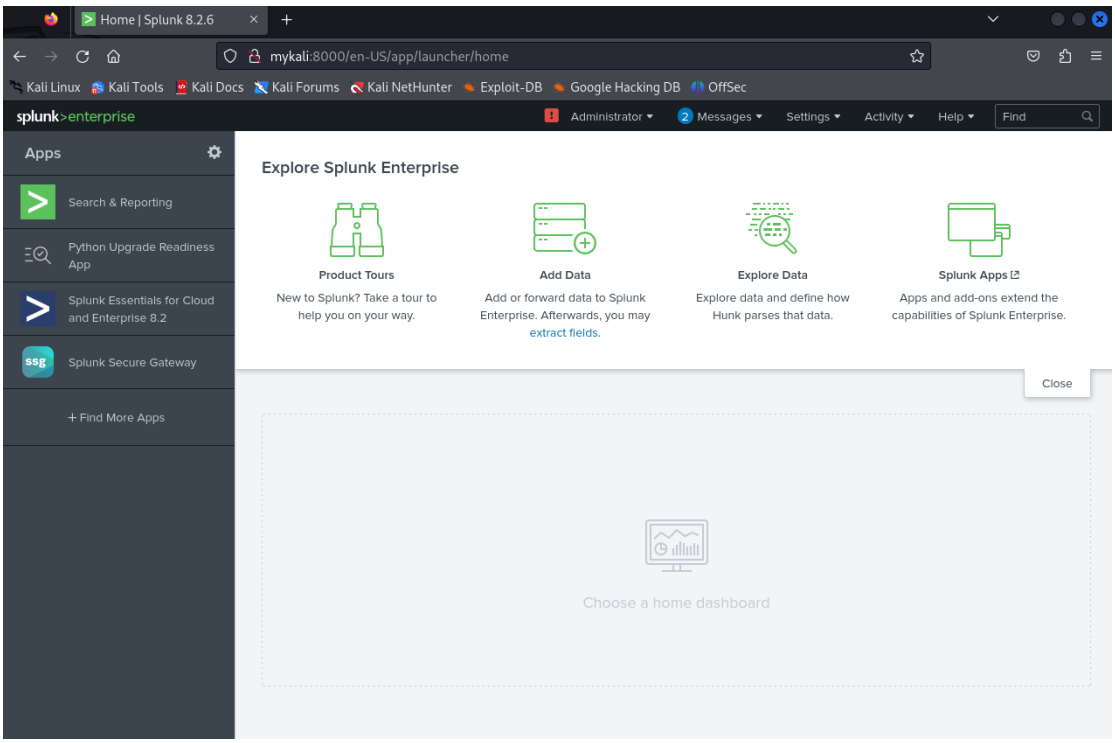

Рисунок 4.26 – Головний екран Splunk

Необхідно, щоб логи потравляли в SIEM систему. Це дозволить краще аналізувати дані. Для реалізації цієї задачі, по-перше необхідно створити монітор, куди будуть передаватися логи. Для цього переходимо Settings> Add Data (рис. 4.27).

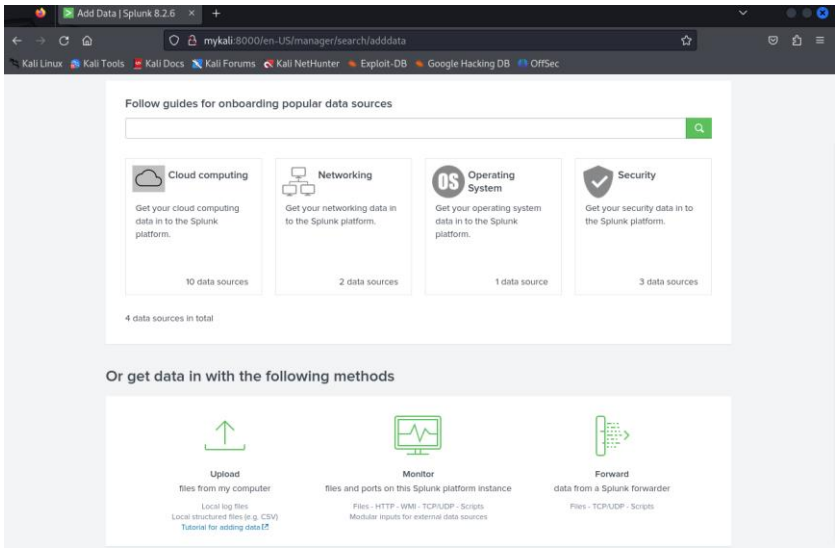

Рисунок 4.27 – Створення монітора

Вибираємо HTTP Event Controller (рис. 4.28)

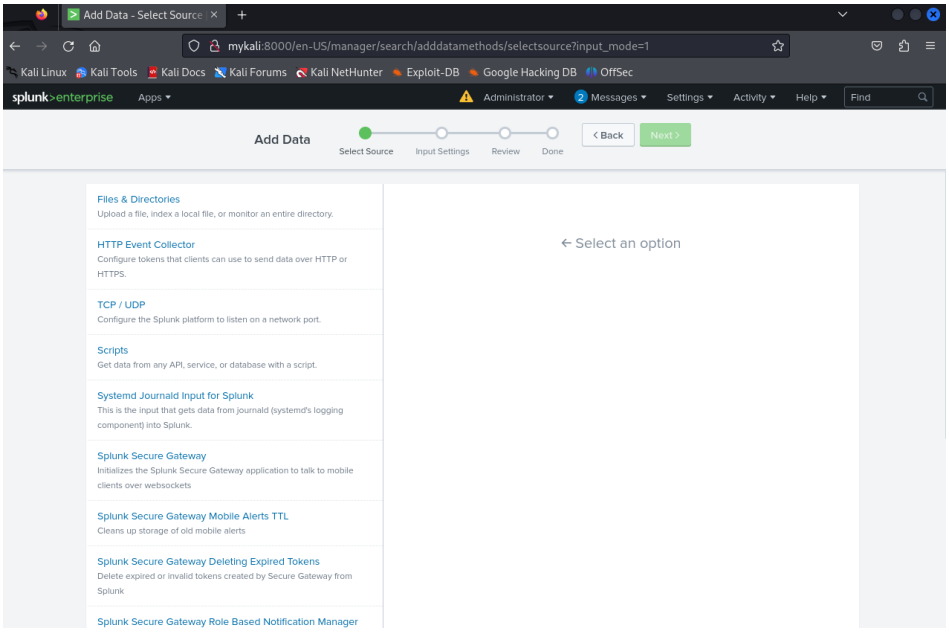

Рисунок 4.28 – Створення монітора

Після введених даних, формуються налаштування (рис. 4.29).

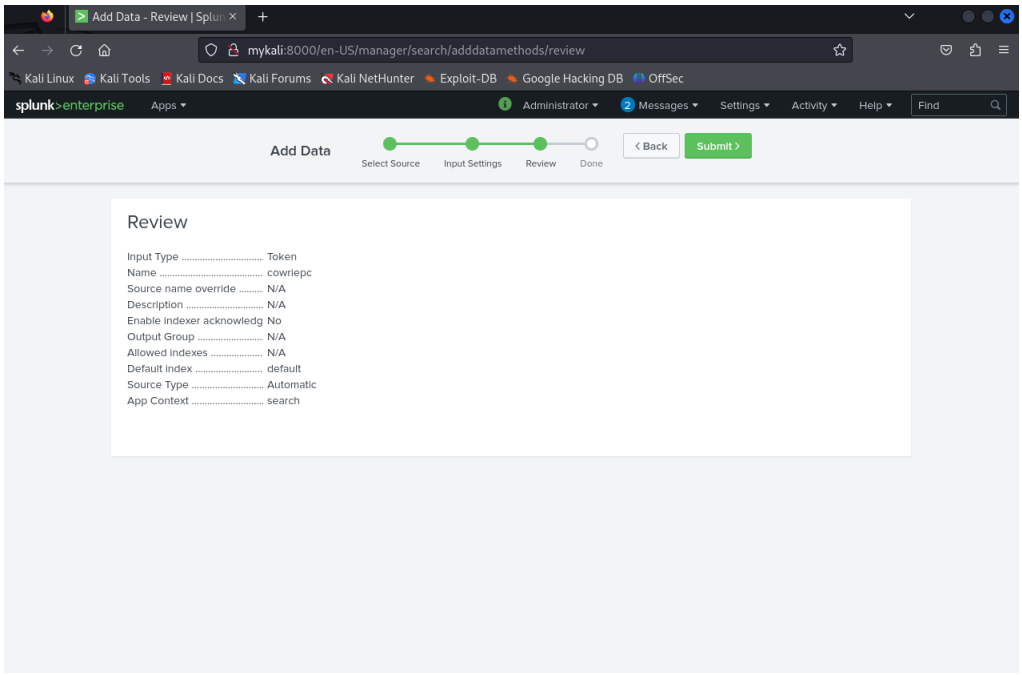

Рисунок 4.29 – Налаштування монітора

Натискаємо «Next» і отримуємо токен, який потрібно внести в конфігураційний файл Cowrie.

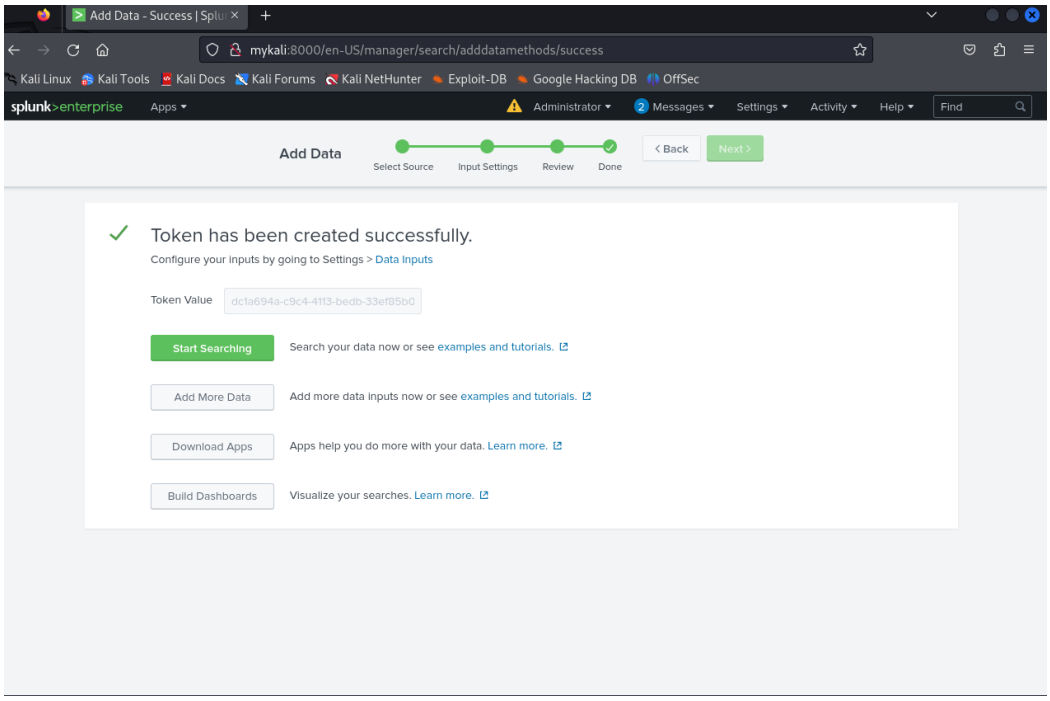

Рисунок 4.30 – Результат налаштування монітора

Переходимо до файлу cowrie.cfg і вносимо зміни у розділ output_splunk. Вводимо токен, який отримали в Splunk та змінюємо значення в enabled з false на true (рис. 4.31).

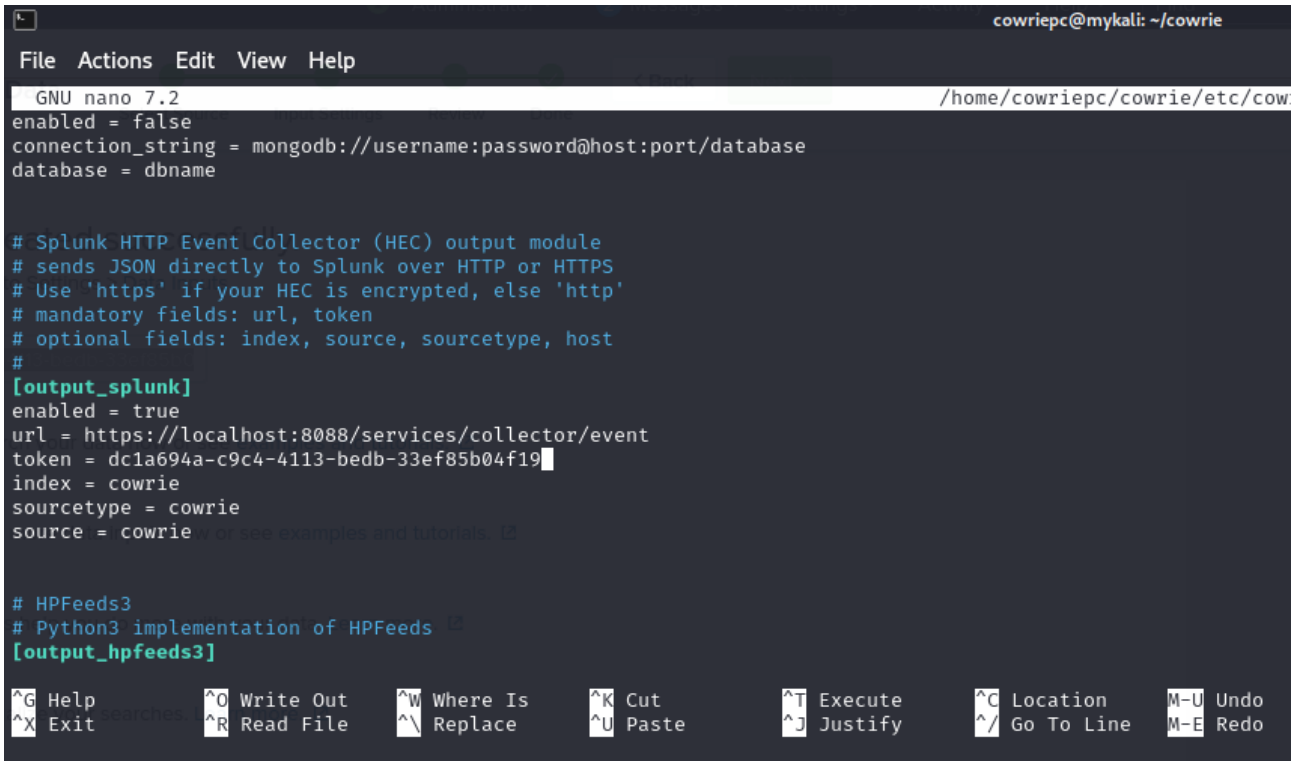

Рисунок 4.31 – Конфігураційний файл Cowrie

Тепер потрібно створити індекс, по якому будуть відсортовані логи. Для цього переходимо в «Settings» і в розділі «DATA» вибираємо «Indexes»(рис. 4.32).

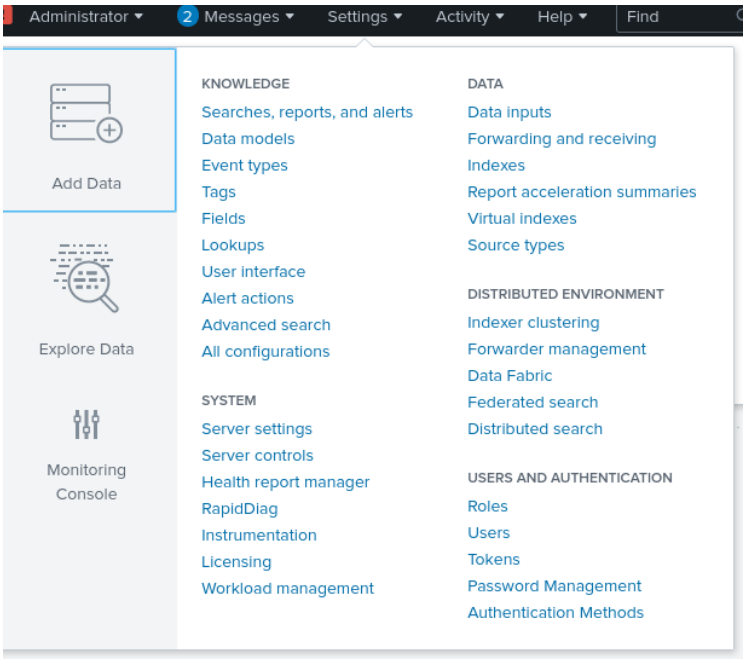

Рисунок 4.32 – Процес створення індексу

Натискаємо на кнопку «New Index». Тепер переходимо в форму створення індексу. Вводимо в поле «Index Name» назву cowrie. Переводимо в режим «Enable» і натискаємо кнопку «Save» (рис. 4.33).

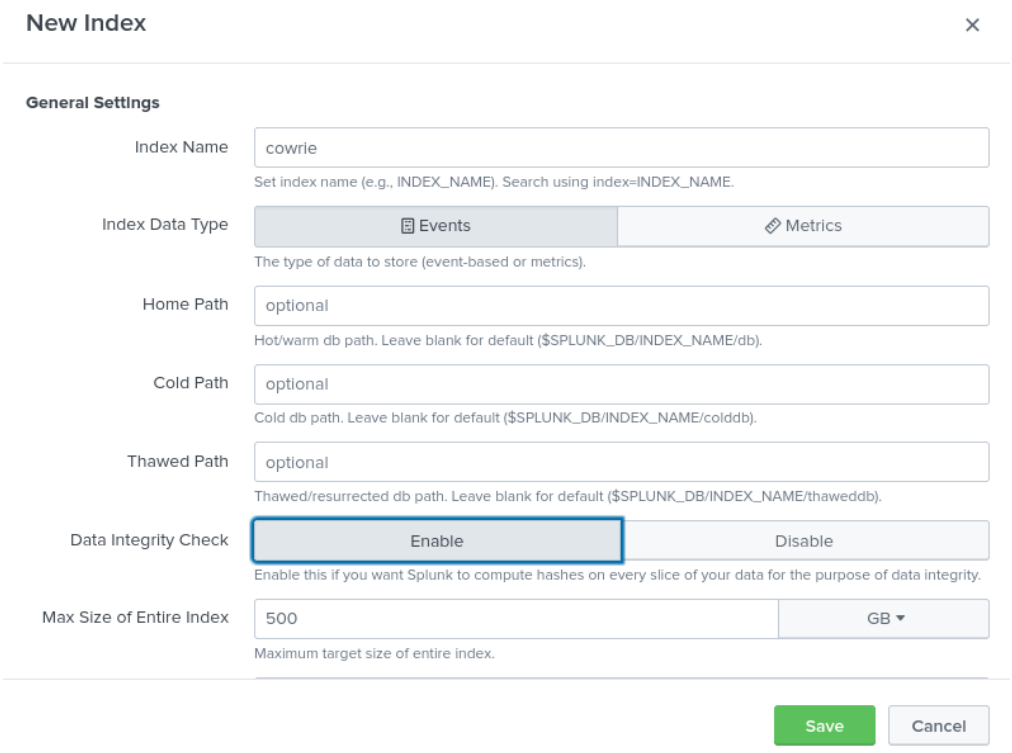

Рисунок 4.33 – Форма створення індексу

Тепер переходимо в розділ «Data Inputs» і вибираємо наш монітор. Нам потрібно додати індекс, для цього перетаскуємо його в поле «Selected indexes» і натискаємо «Save» (рис. 4.34).

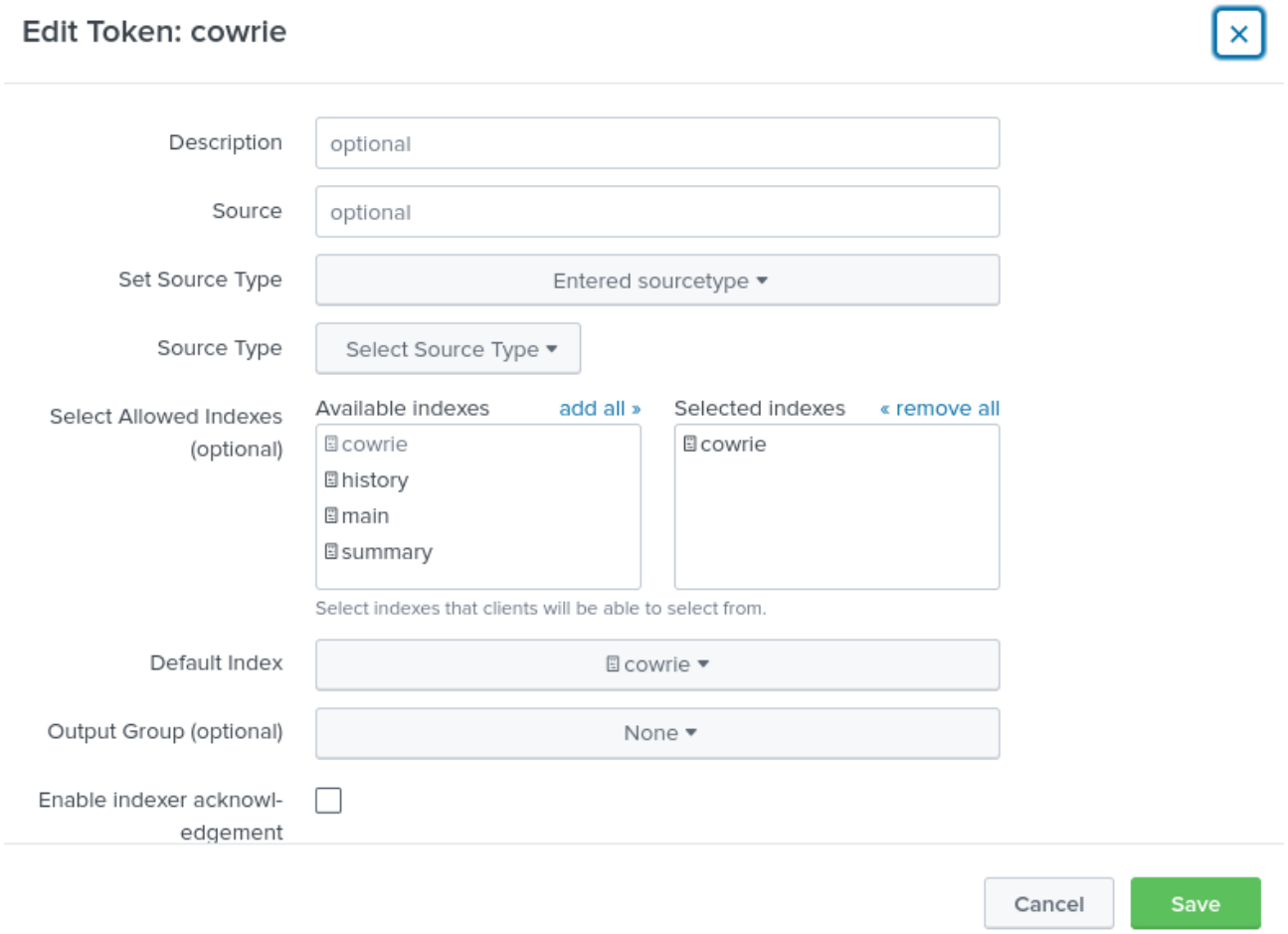

Рисунок 4.34 – Процес додавання індексу в монітор

4.2 Використання Canary Tokens

Використовуючи веб-ресурс створимо файл кенері токенс та помістимо їх у файлову систему. Вибираємо тип файлу – Microsoft Word document. Далі вводимо електронну пошту, на яку відправляється повідомлення при відкритті файлу. Натискаємо «Create my Canarytoken» (рис. 4.35).

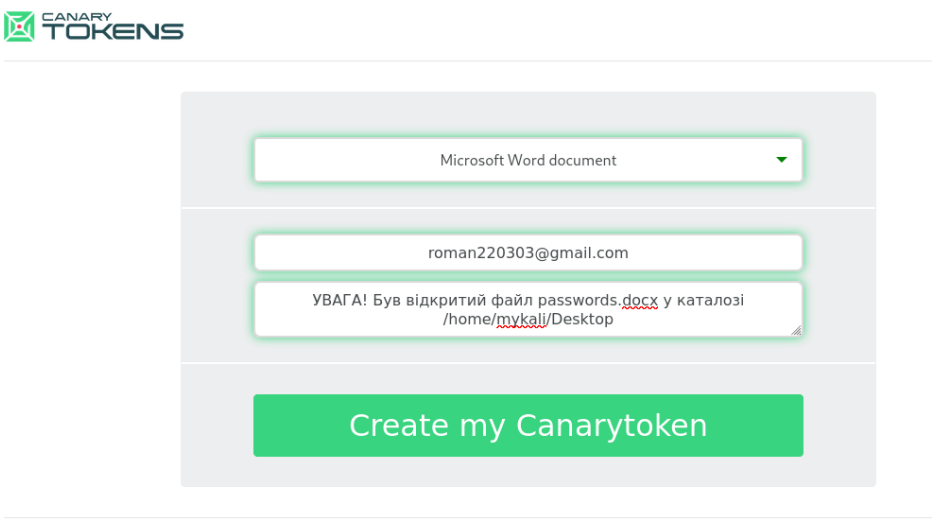

Рисунок 4.35 – Створення файлу

Завантажуємо файл, натиснувши «Download your MS Word file». Потім розміщуємо файл на робочий стіл та перейменовуємо його на passwords.docx (рис. 4.36 – 2.38).

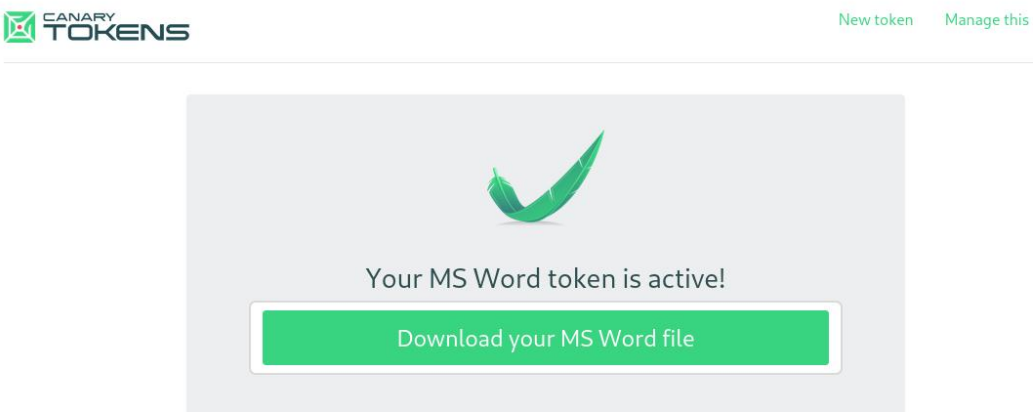

Рисунок 4.35 – Вікно веб-сайту

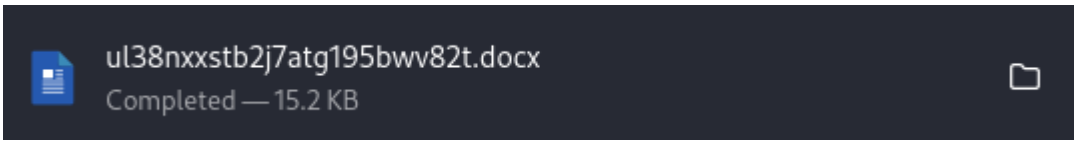

Рисунок 4.36 – Створений файл

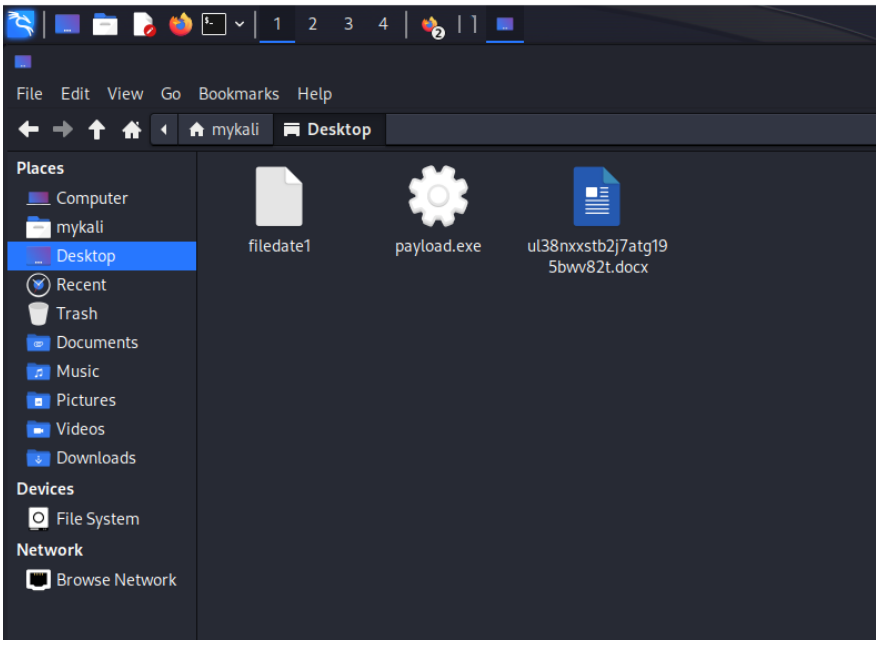

Рисунок 4.37 – Вміст робочого столу

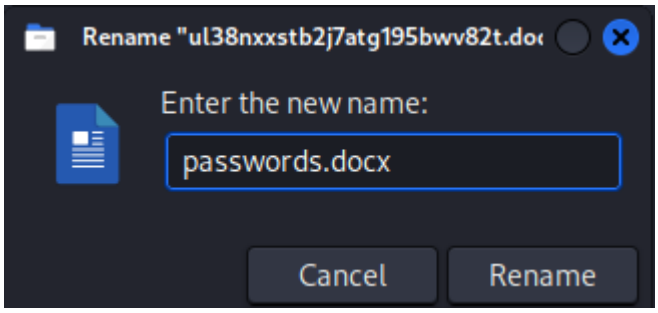

Рисунок 4.38 – Процес перейменування файлу

Натискаємо «Rename». Тепер при кожному відкритті цього файлу на пошту roman220303@gmail.com буде приходити повідомлення «УВАГА! Був відкритий файл passwords.docx у каталозі /home/mykali/Desktop». У сам файл можна вносити, будь-яку інформацію.

4.3 Тестування програмного рішення

Симулюємо ситуацію, що після сканування мережі комерційного підприємства хакер хоче під'єднатися до хоста, який є нашим honeypot Cowrie[19, 20].

Використаємо для цього програму PuTTY. Вводимо дані для підключення: IP-адресу та порт. Вибираємо тип підключення SSH та натискаємо «Open»(рис. 4.39).

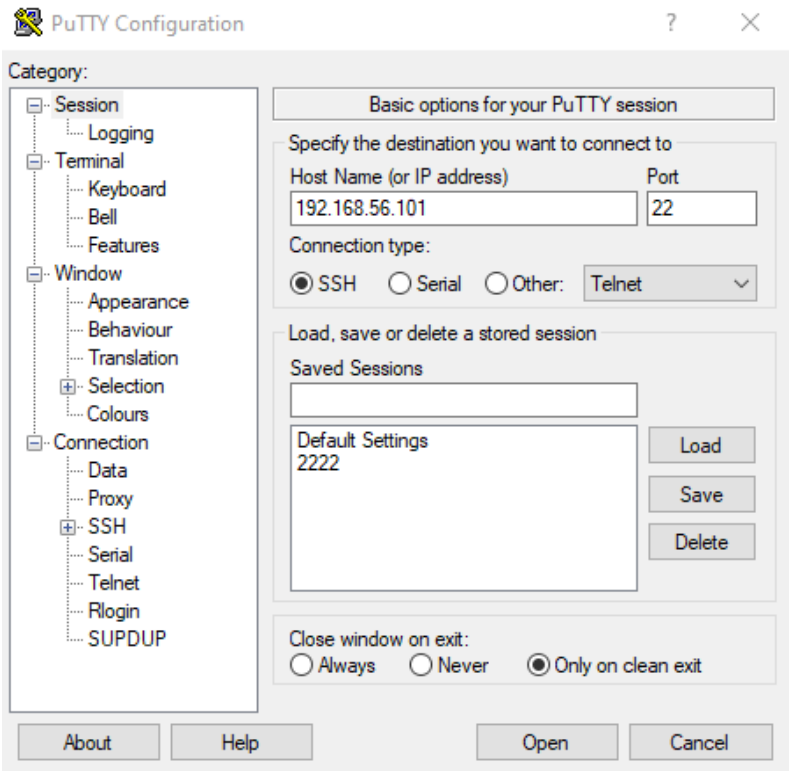

Рисунок 4.39 – Програма PuTTY

У вікно консолі, вводимо різні значення логінів та паролів (рис. 4.40).

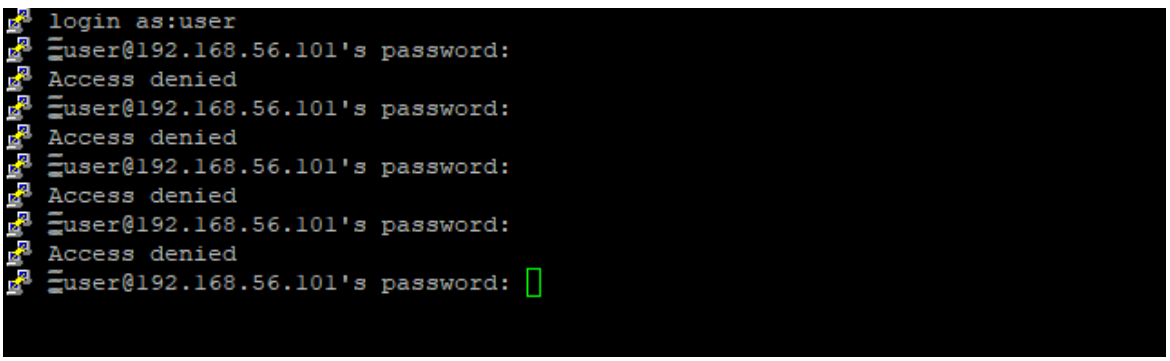

Рисунок 4.40 – Вікно консолі

Далі хакер пробує зайти під користувачем root і підбирає паролі. Після підбору пароля він отримує доступ до приманки (рис. 4.41).

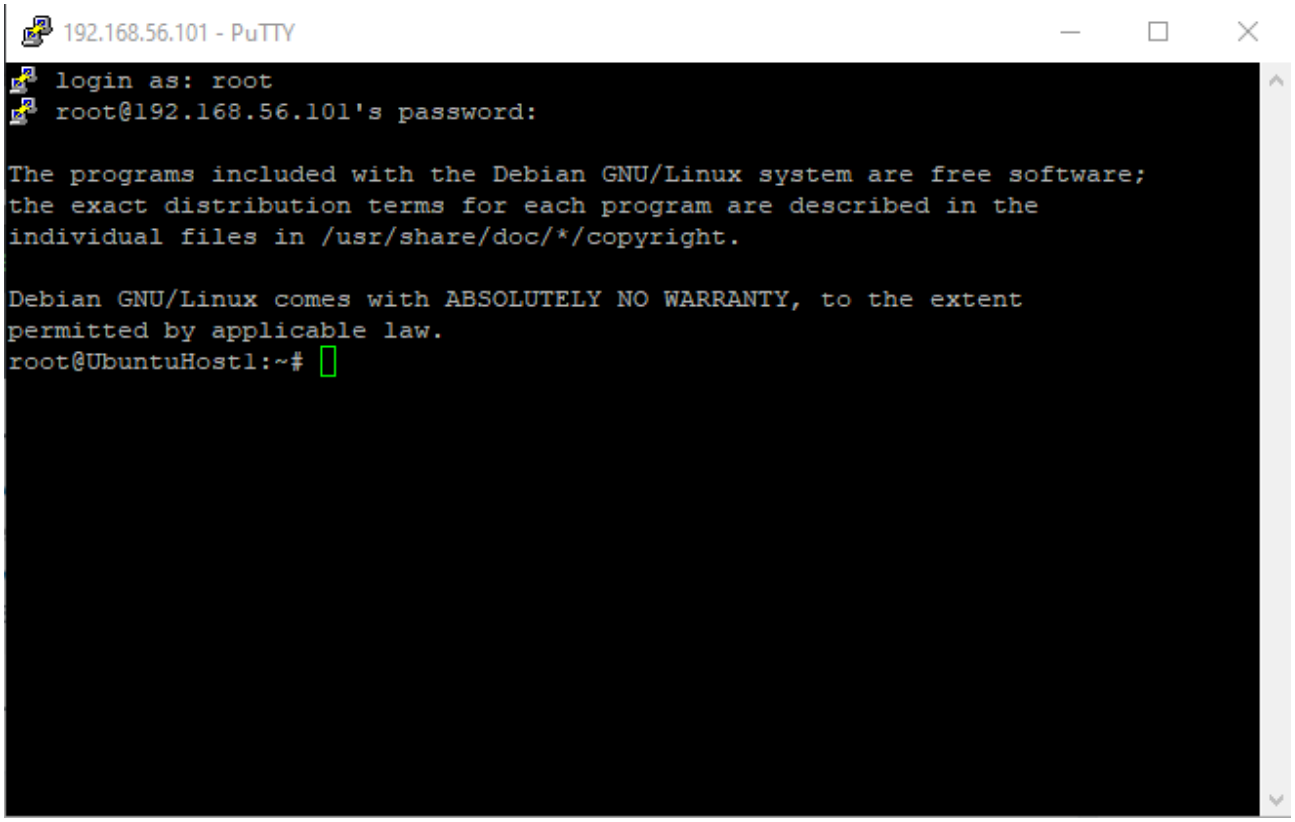

Рисунок 4.41 – Вікно консолі

Вводимо команди, які може використати хакер (рис. 4.42 – 2.43):

1. Whoami(інформація про ім'я користувача, від імені якого виконуються команди).
2. Hostname(інформація про ім'я хоста комп'ютера).
3. ls -al(виводить вміст поточної директорії з детальною інформацією).
4. cat /etc/passwd(показує список користувачів системи).
5. sudo -i(для отримання доступу до root через sudo).
6. su –(спроба переключитися на користувача root).

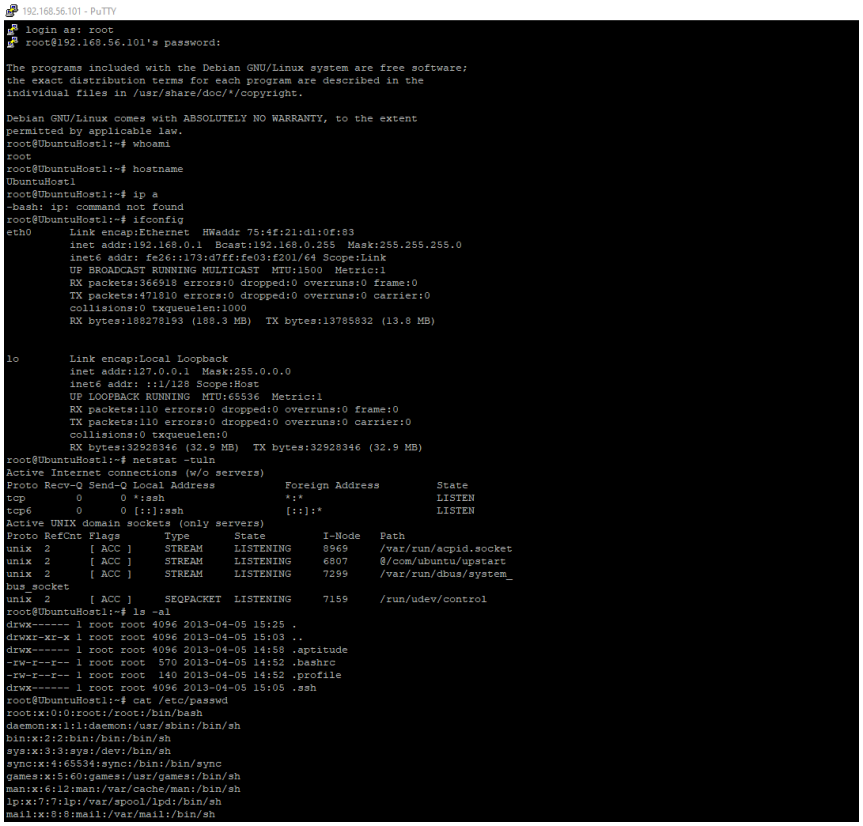

Рисунок 4.42 – Вікно консолі

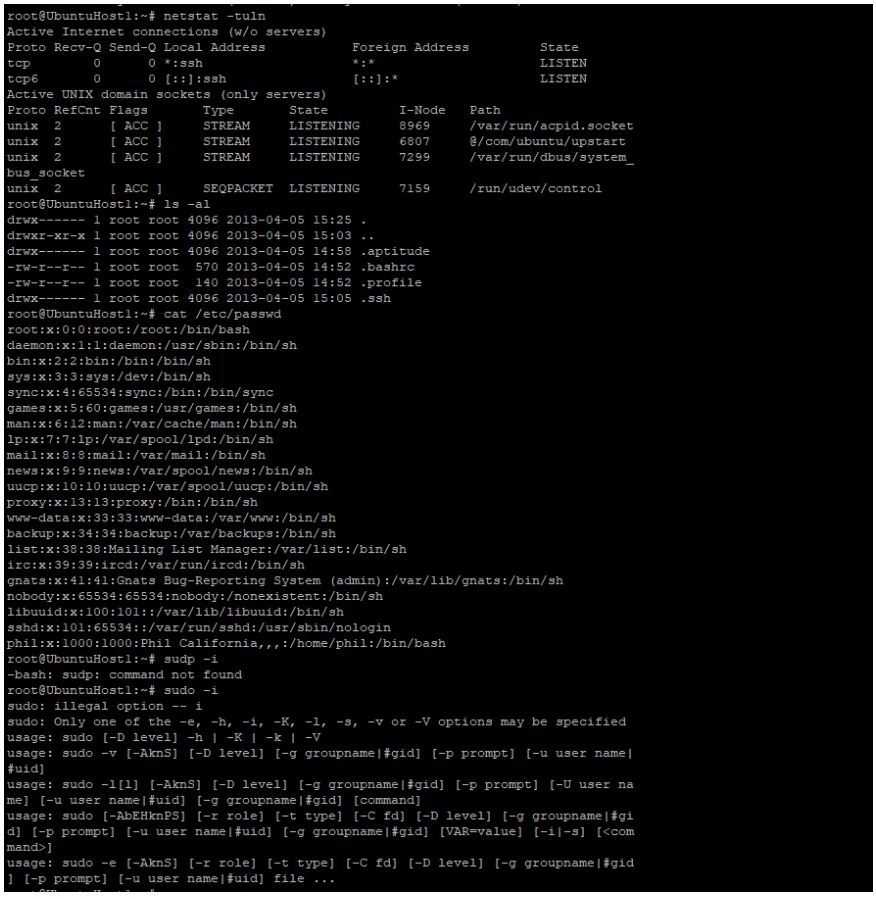

Рисунок 4.43 – Вікно консолі

Переходимо в Splunk, натискаємо «New Search» і вводимо команду «index=cowrie». Після чого виведено результат запиту. Система на основі логів, відображає інформацію про кількість подій і їх детальний вміст (рис. 4.44).

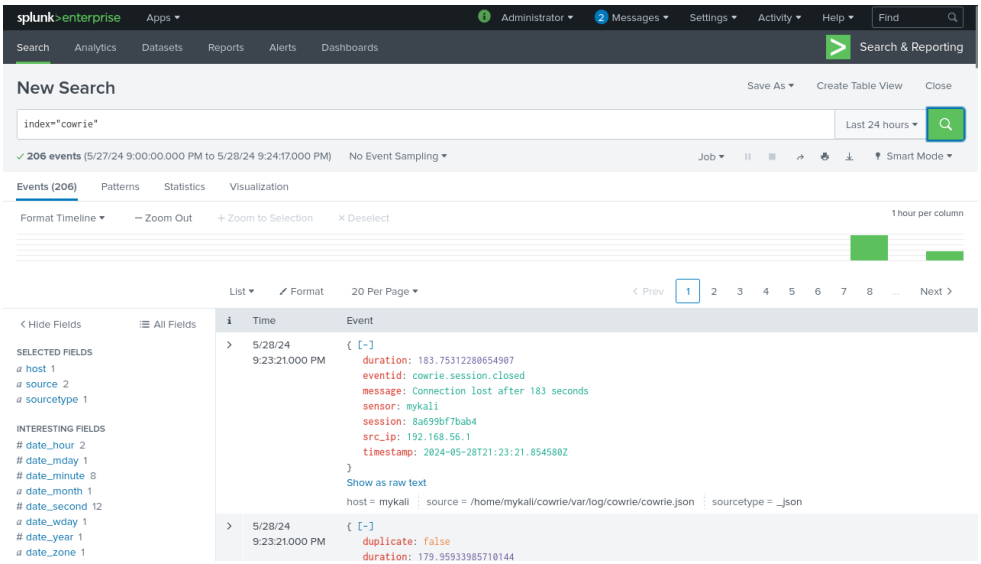

Рисунок 4.44 – Вікно Splunk

На прикладі однієї події розглянемо, що міститься в логах (рис. 4.45).

Рисунок 4.45 – Подія, яка була задетектована

«eventid» містить інформацію про тип події. У нашому випадку це означає, що користувач ввів команду в сесії. «input: cat /etc/passwd» інформую, яка команда була введена. «sensor» - це ідентифікатор або назва сенсора, який зареєстрував подію. Унікальний ідентифікатор сесії, під час якої була введена команда вказується у полі «session». Також завдяки цьому можна дізнатися IP-адресу джерела, з якого надійшло підключення до honeypot та час реєстрації події.

Перевіримо, чи коректно спрацьовує canarytoken, який створений на попередньому кроці. Для цього відкриваємо файл (рис. 4.46).

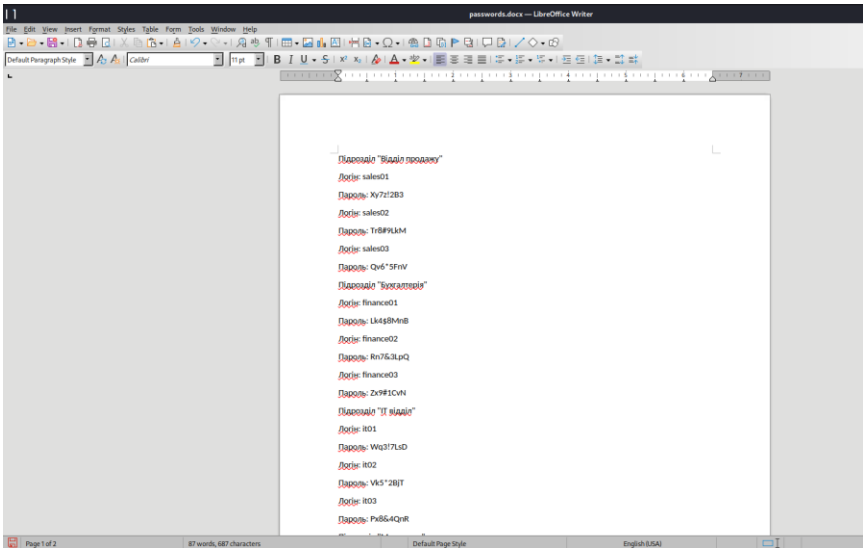

Рисунок 4.46 – Вміст файлу passwords.docx

Переходимо до електроної пошти та перевіряємо, яке повідомлення приходить (рис. 4.47).

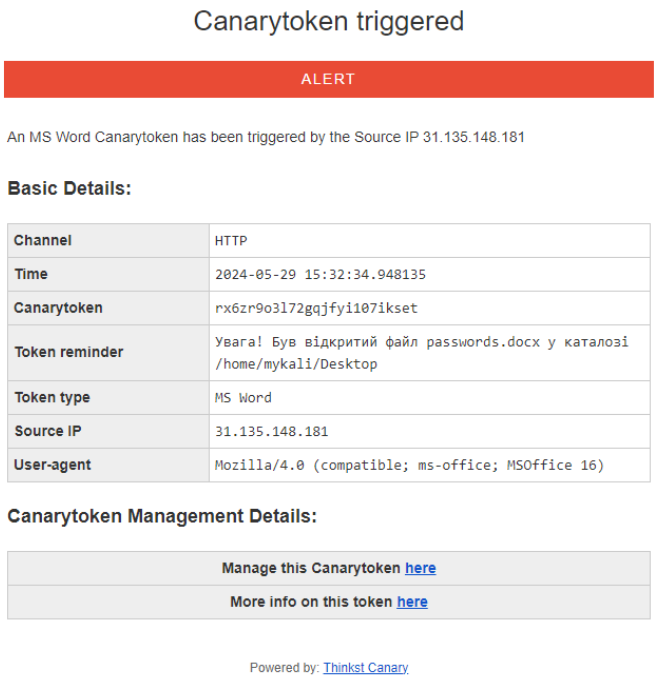

Рисунок 4.47 – Повідомлення про відкриття файлу

ВИСНОВКИ

У ході виконання роботи розглянуто загальні принципи побудови мережевої архітектури та проаналізовано вразливості, які можуть бути використані зловмисниками для атак. Під час огляду інформаційних джерел наведено детальний аналіз технологій honeypot та canary tokens. Технологія honeypot була розглянута як ефективний засіб для виявлення та аналізу атак. Досліджено різні типи honeypots, від простих емуляцій сервісів до складних інтерактивних систем, які можуть заманити атакуючих і дозволити адміністраторам отримати цінну інформацію про методи та інструменти, які використовують зловмисники. Canary tokens проаналізовано та визначено як інструмент раннього попередження про можливі компрометації системи. Крім того, приділено увагу на інтеграції цих технологій з існуючими системами моніторингу та управління безпекою, такими як SIEM (Security Information and Event Management) системи. Це дозволяє підвищити загальний рівень безпеки мережі, забезпечуючи своєчасне виявлення та реагування на інциденти. Практично реалізовано на віртуальній машині технологію Honeypot Cowrie та передавання логів до SIEM-системи Splunk. Застосовано технологію CanaryTokens. Розроблена система детектування зловмисної діяльності дозволяє своєчасно виявити спроби вторгнення в інформаційно-комунікаційну систему комерційного підприємства. Може бути використана для навчання студентів за спеціальністю 125 «Кібербезпека та захист інформації».

СПИСОК ЛІТЕРАТУРИ

1. Буров Є. В. Комп'ютерні мережі: підручник / Євген Вікторович Буров. — Львів: «Магнолія 2006», 2010. — 262 с (дата звернення: 10.04.2024).
2. Khraisat A. Survey of intrusion detection systems: techniques, datasets and challenges - Cybersecurity. SpringerOpen. URL: https://cybersecurity.springeropen.com/articles/10.1186/s42400-019-0038-7 (дата звернення: 10.04.2024).
3. Sastri S. Splunk –an overview. Medium. URL: https://sarada-sastri.medium.com/splunk-an-overview-f5473441685 (дата звернення: 21.05.2024).
4. Robbins C., Steele G. A new day for data: cisco and splunk. Cisco Blogs. URL: https://blogs.cisco.com/news/a-new-day-for-data-cisco-and-splunk (дата звернення: 12.04.2024).
5. Види мережевих атак. IT Блог Холодок. URL: https://holodoks.blogspot.com/2017/12/blog-post.html (дата звернення: 14.04.2024).
6. Network security threats and vulnerabilities. NordLayer. URL: https://nordlayer.com/learn/network-security/threats/ (дата звернення: 14.04.2024).
7. Spasojevic A. Network security threats. phoenixNAP. URL: https://phoenixnap.com/blog/network-security-threats (дата звернення: 14.04.2024).
8. Canary токени: як перетворити свою мережу на пастку для зловмисників. 10 Guards. URL: https://10guards.com/ua/articles/canary-tokens-how-to-turn-your-network-into-a-honeypot/ (дата звернення: 14.04.2024).
9. Akshantula N. Defending against website cloning attack with canary tokens. halodoc. URL: https://blogs.halodoc.io/defending-against-website-cloning-attack-with-canary-tokens/ (дата звернення: 20.04.2024).

10. What are canarytokens. Canary. URL: https://help.canary.tools/hc/en-gb/articles/4701687447325-What-are-Canarytokens (дата звернення: 20.04.2024).

11. Azizi Mohd Ariffin M. Deployment of honeypot and SIEM tools for cyber security education model in UITM. ResearchGate. URL: https://www.researchgate.net/publication/365509767_Deployment_of_Honeypot_and_SIEM_Tools_for_Cyber_Security_Education_Model_In_UITM (дата звернення: 21.04.2024).

12. A highly interactive honeypot-based approach to network threat management / X. Yang MDPI. URL: https://www.mdpi.com/1999-5903/15/4/127 (дата звернення: 23.04.2024).

13. Morgan Y., Ikuomenisan G. Meta-Review of recent and landmark honeypot research and surveys. Scientific Research. URL: https://www.scirp.org/journal/paperinformation?paperid=119340 (дата звернення: 24.04.2024).

14. Wang J., Zeng J. Construction of large-scale honeynet Based on Honeyd. ScienceDirect. URL: https://www.sciencedirect.com/science/article/pii/S1877705811021138 (дата звернення: 24.04.2024).

15. Sethia V. Malware capturing and analysis using dionaea honeypot. ResearchGate. URL: https://www.researchgate.net/publication/336953023_Malware_Capturing_and_Analysis_using_Dionaea_Honeypot (дата звернення: 24.04.2024).

16. Rawat M. Tracking attackers with a honeypot - part 2 (kippo). Infosec. URL: https://www.infosecinstitute.com/resources/incident-response-resources/tracking-attackers-honeypot-part-2-kippo/ (дата звернення: 24.04.2024).

17. Cowrie. GitHub. URL: https://github.com/cowrie/cowrie (дата звернення: 24.04.2024).

18. Documentation. Splunk. URL: https://docs.splunk.com/Documentation (дата звернення: 24.04.2024).

19. Kumar S. Advanced linux commands for the modern hacker. Medium. URL: https://blog.stackademic.com/advanced-linux-commands-for-the-modern-hacker-84b1cd4ca15f (дата звернення: 25.04.2024).

20. Iwugo D. Linux for hackers – basics for cybersecurity beginners. freeCodeCamp.org. URL: https://www.freecodecamp.org/news/linux-basics/ (дата звернення: 25.04.2024).