Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Система виявлення атак на Інтернет речей з використанням технології honeypot

Назва теми

КвРКІ.190187.17.03.09 ПЗ
Шифр

Галузь знань *12 «Інформаційні технології»*
Шифр, назва

Спеціальність *123 «Комп'ютерна інженерія»*
Шифр, назва

Освітня програма *«Комп'ютерна інженерія»*
Назва

Виконав: студент III курсу, група КІ2с–19–1 _____ О. Ю. Круцюк
Підпис, Ініціали, прізвище

Керівник _____ В.В. Яцків
Підпис, дата, Ініціали, прізвище

Нормоконтролер _____ С.М. Лисенко
Підпис, дата, Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
Інженерії та системного
Програмування _____ Т.О. Говорущенко
Підпис, Ініціали, прізвище

« 16 » червня 2022 р.

Хмельницький 2022

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Комп'ютерної інженерії та інформаційних систем

Освітній рівень бакалавр

Галузь знань 12 Інформаційні технології

Спеціальність 123 Комп'ютерна інженерія

Освітня програма освітня програма «комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Зав. кафедри КІІС

Т.О.Говорущенко

" 11 " 01 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Круцюку Олександру Юрійовичу
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Система виявлення атак на Інтернет речей з використанням технології honeypot

Керівник проекту (роботи) Яцків В.В. д.т.н., проф.
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 06.01.2022 р. № 1

2. Строк подання студентом проекту (роботи) на кафедру 07.06.2022 р.
3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
   Дослідження предметної області та постановка задачі
   Налаштування віртуального сервера
   Тестування роботи та вивід результатів моніторингу подій
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)
   Структура системи виявлення загроз
   Структурна схема системного модуля honeypot Cowrie
   Діаграма HoneyPort

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання при
Нормоконтроль	Лисенко С.М., професор кафедри КІІП		
Антиплагіат	Нічепорук А.О., доцент кафедри КІІП		

7. Дата видачі завдання «__06__»_____09_____2021 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	При
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	11.01.2022	вико
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2022	вико
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2022	вико
4	Робота над розділом 2 – Проектування структури honeypots	01.04.2022	вико
5	Робота над розділом 3 – Налаштування віртуального сервера Тестування роботи та вивід результатів моніторингу подій	30.04.2022	вико
6	Оформлення пояснювальної записки згідно вимог	31.05.2022	вико
7	Попередній захист ВКР		вико
8	Захист ВКР на засіданні ЕК	02.06.2022	вико
		Червень 2021 року	

Студент

Керівник проекту (роботи) _____ Підпис Круцюк О.Ю

формат	Позначення	Найменування	Кіл. листів	№ скз	Примітка
		Текстові документи			
	КвРКІ 190187.13.01.12 ПЗ	Пояснювальна записка	63		
		Графічні матеріали			
	КвРКІ 190187.13.01.12 Е8	Структура системи виявлення загроз	1		
	КвРКІ 190187.13.01.12 Е8	Структурна схема	1		
		Системного модуля			
		Honeypot Cowrie			
	КвРКІ 190187.13.01.12 Е2	Діаграма Honeyport	1		

Арк	№ докум	Підпис	Дата	КвРКІ 190187.13.01.12 ПЗ		
Розробив	Крушко			Відомість проекту	Літера	Аркуш	Аркушів
Керівн.	Яцків				У	1	1
Н.контр.	Лисенко				ХНУ, КІ2с-19-1		
Затв.	Самородченко						

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Система виявлення атак на Інтернет речей з використанням технології honeypot».

Автор роботи: Круцюк Олександр Юрійович

Керівник роботи: Яцків Василь Васильович.

Пояснювальна записка: 63 с., 14 рис., 3 табл., 3 дод., 23 джерел.

Графічна частина: 7 презентаційних слайдів.

Метою роботи є аналіз існуючого стану та тенденції розвитку Інтернету речей, аналіз класичних honeypots.

Об'єктом дослідження є приманки для виявлення вразливостей безпеки Інтернету речей.

Практичне значення має налаштування віртуального сервера, встановлення приманки Cowrie та тестування роботи та вивід результатів моніторингу подій.

_____ 02.06.2022

Підпис студента Дата

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

IoT -– Internet of things

SSH - Secure SHell

IIS - Internet Information Services

SMTP - Simple Mail Transfer Protocol

EC2 - Amazon Elastic Compute Cloud

GCP - Google Cloud Platform

DoS - Відмова в обслуговуванні

OWASP - Open Web Application Security Project

ВСТУП

Сучасні інформаційні технології, в тому числі мультимедіа, відкривають доступ до нетрадиційних джерел інформації, дозволяють реалізувати принципіальні нові форми захисту інформації. Розроблена класифікація honeypot, яка враховує їх призначення, ролі, рівні взаємодії, масштабованість, рівні ресурсів, доступність вихідного коду та застосування.

Досліджено найбільш поширені типи атак на Інтернет- речей., зокрема, фізичні атаки, атаки шифрування, DoS (відмова в обслуговуванні), викрадання прошивки, ботнети, та інші.

В цій роботі проведено аналіз класичних honeypots, їх класифікація. Дослідимо наскільки саме небезпечне інтернет атаки, та кіберзагрози.

З такою кількістю технологій та пристроїв, які ми зараз маємо, та з вільним доступом до Інтернету, вони стають вразливими пристроями IoT, якими можуть скористатися зловмисники. Без правильного захисту та оновлення мікропрограмного забезпечення такі пристрої, як розумні холодильники, годинники, відеокамери можуть стати жертвами ботнету, надаючи зловмисникам можливість здійснювати DDoS-атаки.

Достатньо довго в протистоянні «напад-захист» практикувалася своєрідна покрокова стратегія – зловмисники користувалися однією «діркою» захисту і з часом її закривали, тоді вони шукали іншу – її згодом також закривали і т.п. Такий аналог гри в шахи, де партія може тривати як завгодно довго, вимагає від захисту колосальних витрат часу і ресурсів, тим паче в нападника майже завжди є можливість адекватно відреагувати на захисні заходи.

Зважаючи на це, сторона захисту повинна «грати на попередження», цим самим мінімізуючи ризик вторгнення. Саме реалізація такої ідеї лежить в основі використання віртуальних приманок – так званих, honeypot-систем (від англ. – «горщик з медом»).

Мета їх функціонування – бути атакованими або сканованими зловмисниками для вивчення стратегії останніх, визначення кола їх засобів, за допомогою яких можуть бути нанесені удари по реальних об’єктах безпеки. Метод реалізації віртуальної приманки не принциповий – це може бути як спеціально розгорнута цілісна мережа так і один єдиний емульваний мережевий сервіс, основним і першочерговим завданням якого є зацікавлення (привернення уваги) порушника.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз існуючого стану та тенденції розвитку Інтернету речей

Інтернет речей (IoT) – це термін, який описує все більш складні екосистеми підключених онлайн-пристроїв.

Сьогодні майже будь-який пристрій, який ми використовуємо в наших будинках, офісах, на заводах або просто носимо на своєму тілі, може бути онлайн та підключеним, отже, Інтернет «речей».

Автомобілі з автономним керуванням, роботи автономного виробництва та дистанційні медичні пристрої, які дозволяють лікарям діагностувати пацієнтів і навіть проводити хірургічні операції, – все це можливо завдяки цим мережам підключених речей.

Інтернет речей (IoT) зріс у геометричній прогресії за дуже короткий період часу. Розумні гаджети, хоча і є відносно новими, повсюдно присутні в сферах бізнесу та споживачів, і вони не мають ознак уповільнення.

Хоча розширення Інтернету речей може залишатися незмінним у 2022 році, воно буде коливатися та адаптуватися.

Проаналізуємо основні тенденції, які вплинуть на Інтернет речей цього року [1].

1. IoT в охороні здоров'я. IoMT продовжує розширюватися. Зі збільшенням попиту на рішення для охорони здоров’я без використання рук, пандемія COVID-19 прискорила впровадження Інтернету медичних речей (IoMT). У 2021 році 64% домогосподарств у США повідомили, що користуються цими послугами, а 43% мають намір продовжити користуватися ними після епідемії. Ці показники призводять до постійного зростання IoMT у 2022 році. Наприклад, моніторинг стану здоров’я на людині стане все більш поширеним, щоб розширити лікування вдома. З’єднання IoT використовуватиметься лікарнями для відстеження ресурсів та виконання віддалених зустрічей. Навіть після того, як епідемія пройде, ці моделі продовжуватимуть збільшуватися, щоб зробити медичну допомогу доступнішою.

2. Безпека. Оскільки Інтернет речей стає все популярнішим, його недоліки безпеки стають все більш важливими. Порушення підключених автомобілів може призвести до близько 3000 смертей у годину пік. Оскільки виробляється все більше пристроїв IoT і зростає кіберзлочинність, безпека стане пріоритетом. Більш вбудовані засоби безпеки, такі як секретні обчислення та механізми перевірки для оновлення по повітрю, будуть включені виробниками IoT. Щоб доповнити це оновлення, постачальники безпеки нададуть додаткові послуги, пов'язані з Інтернетом речей. Уразливості не зникнуть повністю, але їх буде зменшено.

3. 5G сприяє розвитку Інтернету речей. Одним з найбільш потенційних застосувань цих технологій є промисловий IoT (IIoT). Водночас низька пропускна здатність існуючих мереж і надмірна затримка перешкоджають їх розгортанню. Поява 5G у 2022 році все змінить.

Наразі існує 48 мільярдів гаджетів, підключених до Інтернету, що навантажує нинішні мережі. Завдяки високій швидкості 5G, зменшеній затримці та більшій ємності промислові об'єкти зможуть розповсюдити IIoT на додаткові місця. Цілі фабрики можуть бути перетворені в уніфіковані, взаємопов'язані одиниці.

4. IoT забезпечує стійкість ланцюга постачання Підприємства постраждали від перерв у ланцюжках поставок у 2020–2021 роках. У результаті все більше компаній інтегруватимуть підключення IoT у свої логістичні операції. Віддалене відстеження забезпечить прозорість, необхідну підприємствам для виявлення порушень і реагування на них до того, як вони виникнуть. Крім того, дані, зібрані цими пристроями, можуть бути подані в складні алгоритми, які потім можуть надати корисну інформацію. Ці вдосконалення зроблять ланцюжки поставок значно більш надійними, але їх буде неможливо досягти без більшого використання Інтернету речей. Як наслідок, ланцюг поставок IoT може зрости.

5. Розвиток граничних обчислень. Оскільки мережі IoT розширюються, вони відкриють двері для ще однієї невідворотної інновації: граничних обчислень. Граничні обчислення мають потенціал для того, щоб зробити самокеровані

автомобілі більш практичними, а також вирішити багато сучасних проблем пропускної здатності, безпеки та надійності хмари. Незважаючи на ці переваги, сьогодні це нішева технологія, але це зміниться в міру розширення Інтернету речей.

6. Носимі пристрої. Пристрої для носіння є одними з найпопулярніших споживчих сфер IoT, і очікується, що ця тенденція збережеться і після 2022 року. Зараз серед носних пристроїв переважають розумні годинники та браслети, але в майбутньому вони стануть значно диверсифікованими. Розумні кільця, розумні окуляри, пов'язана тканина та мітки IoT ID вже є на ринку і з'являться. У період з 2016 по 2019 рік кількість підключених носимих пристроїв зросла більш ніж у чотири рази, і ця тенденція буде прискорена завдяки більш різноманітним пропозиціям. Пристрої для носіння допоможуть покращити доступність для людей з вадами, розширять додатки доповненої реальності (AR), а також допоможуть компаніям зменшити кількість травм на виробництві.

Статистика IoT – основні висновки [1, 2]:

– у 2021 році було понад 10 мільярдів активних пристроїв IoT;

– кількість активних IoT-пристроїв у 2030 році перевищить 25,4 мільярда;

– до 2025 року 152 200 пристроїв IoT підключатимуться до Інтернету на хвилину;

– до 2025 року рішення IoT можуть створити 4-11 трильйонів доларів економічної вартості. 83% організацій підвищили свою ефективність за рахунок впровадження технології IoT;

– гобальні витрати на Інтернет речей становитимуть 15 трильйонів доларів за шестирічний період з 2019 по 2025 рік;

– споживчий ринок IoT досягне 142 мільярдів доларів до 2026 року при CAGR 17%. 94% роздрібних продавців погоджуються, що переваги від впровадження IoT переважають ризики;

– очікується, що до 2025 року обсяг даних, які генеруються пристроями IoT, досягне 73,1 ЗБ (зеттабайт).

1.2 Принципи функціонування приманок

Протягом останніх кількох років кількість атак зросла в тому числі і на пристрої IoT. Зростає складність атак, оскільки їх вектори постійно змінюються.

Незважаючи на те, що кількість атак IoT зросла, було проведено дуже мало досліджень, щоб визначити, наскільки ці загрози ефективні та наскільки широка їхня сфера. Щоб виправити вразливості пристроїв IoT, дослідники повинні зрозуміти причини та методи, за допомогою яких ці зловмисники атакують пристрої. У цьому випадку приманки виявилися важливим інструментом у процесі розслідування.

Honeypot – це комп'ютерна система або програма, створена для залучення зловмисників, які намагаються атакувати комп'ютерні мережі за допомогою спаму, фішингу, DDoS чи інших підступних методів. Як тільки зловмисник потрапляє в цю пастку, honeypot дозволяє адміністраторам отримати цінні дані про тип зловмисника, діяльність, яку він намагався, і в багатьох випадках навіть ідентифікувати зловмисника.

Основна мета всіх honeypots – виявляти нові атаки на різні типи програмного забезпечення та збирати звіти для аналізу та створення розвідувальних даних, які згодом будуть використовуватися для створення методів запобігання мережевим загрозам.

Це сервери-приманки, які використовуються для збору інформації про зловмисників або користувачів, які отримують доступ до інформаційних систем без авторизації. Honeypot зазвичай може бути комп'ютером, який здається частиною мережі, або будь-яким сервером, на якому розміщуються дані (рисунок 1.1).

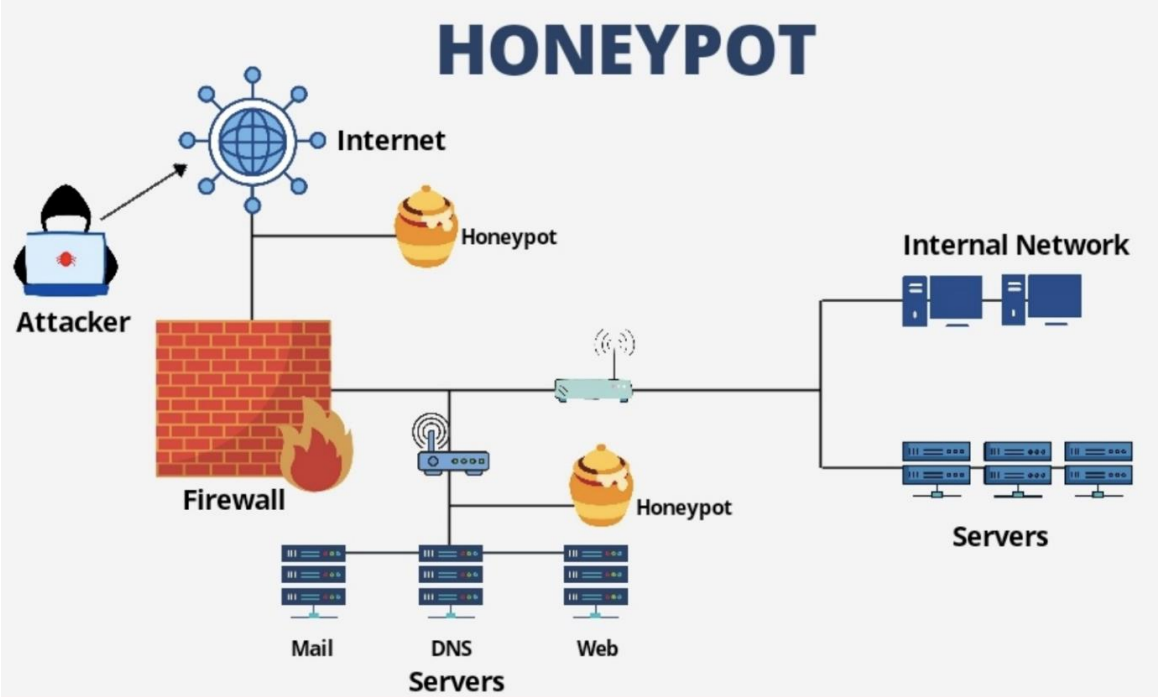


Рисунок 1.1 – Структура honeypot

Є два різних типи приманок:

– дослідницький honeypot: цей тип приманки використовується розробниками, системними адміністраторами та менеджерами синіх команд, які працюють в таких установах, як університети, коледжі, школи та інші пов'язані асоціації;

– виробничий honeypot: це використовується приватними та державними установами, компаніями та корпораціями для розслідування поведінки та прийомів хакерів, які намагаються атакувати мережі в Інтернеті.

По суті, honeypot дозволяє отримувати цінні дані, щоб ви могли працювати над різними стратегіями зменшення поверхні атаки.

Отже приманка – це система пасток. Ці системи-пастки часто встановлюються на віртуальній машині або хмарному сервері, підключеному до мережі, але ізольовані та суворо контролюються системними та мережевими командами.

Щоб допомогти їм бути поміченими зловмисниками, honeypots розроблені так, щоб бути навмисно вразливими, зі слабкими сторонами, які зловмисник

виявить і спробує використати. Ці слабкі сторони можуть бути частиною прогалини в безпеці програми або системної вразливості, наприклад, непотрібних відкритих портів, застарілих версій програмного забезпечення, слабкого пароля або старого невиправленого ядра. Як тільки зловмисник знайшов свою вразливу ціль, він спробує запустити атаку та підвищити привілеї, поки не отримає певний контроль над скринькою або програмою.

Більшість із них не знає, що адміністратор honeypot уважно стежить за кожним їхнім кроком, збираючи дані від зловмисника, які насправді допоможуть посилити поточну політику безпеки. Адміністратор також може негайно повідомити про інцидент до юридичних органів, що часто трапляється з високоякісними корпоративними мережами.

Більшість honeypots працюють як пастки, які відволікають зловмисників від критичних даних, які розміщені в реальних мережах. Інша спільна риса полягає в тому, що майже всі спроби підключення до honeypot можна розглядати як ворожі, оскільки існує небагато причин, якщо такі є, які можуть спонукати законного користувача підключитися до таких систем.

Налаштовуючи honeypot, ви повинні знати про рівень складності злому, який ви хочете надати зловмиснику. Якщо зламати надто легко, вони, ймовірно, втратять інтерес або навіть зрозуміють, що мають справу не з справжньою системою виробництва. З іншого боку, якщо система занадто жорстка, ви фактично перешкодите будь-яким атакам і не зможете збирати будь-які дані. Отже, з точки зору складності, найкращим варіантом для моделювання реальної системи є заманити зловмисника чимось середнім між легким і важким.

Користувачі з високим рівнем технічних знань можуть розпізнати кілька ознак того, що вони входять у приманку. Навіть нетехнічні користувачі можуть виявити honeypot за допомогою автоматизованих детекторів honeypot, таких як Honeyscore Shodan, який дає вам можливість ідентифікувати IP-адреси honeypot.

Щоб досягти успіху, honeypots розроблені з єдиною метою спокусити зловмисників скомпрометувати фіктивну систему. Такі пристрої налаштовані на роботу в ізольованій і окремій мережі, тому вони виглядають як мережеві

пристрої. Крім того, honeypots фактично не надають жодних корисних послуг організаціям безпосередньо, тому доступ до них можна розглядати як акт злісного наміру.

Honeypot збирає конкретну інформацію про спроби зловмисника, що перебуває на honeypot, щоб ідентифікувати зловмисника. Метою цієї технології є виявлення слабких місць у системі та виявлення інструментів і методів, які використовувалися для її компрометації, щоб пом'якшити та запобігти майбутнім атакам.

IoT Honeypot - додаток для моніторингу атак Інтернету речей. Нещодавнє дослідження було проведено компанією, яка протягом більше року розгорнула 50 honeypots по всьому світу для збору цінних даних про атаки IoT. Щоб збільшити збір даних і в кінцевому підсумку отримати краще розуміння, їхні прихильники використовували різні рівні взаємодії. Крім того, вони постійно перебирали IP-адреси, щоб мітки не позначалися як такі, зменшуючи таким чином кількість атак і кількість корисної інформації, яку можна було зібрати. У середньому кожні 15 хвилин відбувається приблизно 20 тисяч інфікованих сеансів [7].

Згідно зі статистикою пристрої Telnet були атаковані понад 105 мільйонів разів із понад 276 000 унікальних IP-адрес. З вищесказаного видно, що одна і та ж IP-адреса використовується кілька разів, що свідчить про те, що пристрої IoT постійно націлені на зараження та атаку. Було виявлено, що зловмисне програмне забезпечення Mirai є найпоширенішою загрозою для цих пристроїв IoT. Через те, що зловмисне програмне забезпечення було вільно доступним протягом значного періоду часу, а також код, здатний об'єднувати ботнети різного рівня складності, це сталося. Близько половини атак, здійснених проти IoT honeypots [7], було здійснено за допомогою бекдорного варіанту під назвою «Backdoor.Linux.Mirai.c».

Безсумнівно, SSH, Telnet і веб-сервери є одними з найбільш часто використовуваних і доступних сервісів у сфері Інтернету речей, що робить їх привабливою мішенню для зловмисників.

На додаток до цього також важливо пам’ятати, що пристрої IoT зазвичай використовують безліч обчислювальних архітектур, які значно відрізняються від традиційних комп’ютерів. Саме через це зловмисники з більшою ймовірністю запускають своє шкідливе програмне забезпечення, коли мають доступ до honeypot, і вони не перевіряють, яку архітектуру вони використовують. По суті, це працює, тому що за раз можна правильно виконати лише один рядок. Завдяки цьому дослідники можуть відстежити джерела інструментів атаки, які використовуються зловмисниками, що дозволяє їм набагато ефективніше вивчати їх пізніше.

1.3 Аналіз класичних honeypots

Все ще необхідно розрізняти honeypots, які розроблені для додатків загального призначення, і ті, які розроблені спеціально для додатків IoT.

Honeypots IoT успадковують деякі характеристики від honeypots загального застосування, включаючи здатність реагувати на події, коли вони виникають. Незважаючи на те, що ці honeypots не розроблені спеціально для IoT, наразі вони використовуються як дослідження для IoT honeypots. У таблиці 1.1 наведено список деяких загальних приманок для IoT.

Таблиця 1.1– Список загальних Honeypots Internet of Thing

Назва Honeypot	Рівень взаємодії	Симуляція сервісів
HoneyD	Low	FTP, SMTP, Telnet
Dionaea	Medium	FTP, HTTP, MQTT, etc.
Adaptive Honeypot Alternative	Low/High	SSH

Cowrie	Medium	SSH

HoneyD [8] – програма з відкритим кодом, яка використовується для створення масштабованих honeypot з низьким рівнем взаємодії. Honeyd не тільки дозволяє створювати віртуальні меді, але також дозволяє інтегрувати машини. Цей honeypot підтримує кілька протоколів, зокрема: UDP, TCP, FTP, SMTP, Telnet, IIS, POP і telnet. Різні дослідження досліджували, чи можна використовувати HoneyD для створення ефективних приманок, які приваблюють зловмисників.

Використовуючи пристрої IoT для моделювання honeypots, дослідники у своєму дослідженні порівняли їх із реальними пристроями IoT. Незважаючи на схожість між вмістом, який обслуговують honeypots, і реальними пристроями, середній час відповіді на запити та сканування Nmap сильно відрізнявся.

Dionaea – це програмне забезпечення з відкритим вихідним кодом, яке дозволяє користувачам створювати маніпулятори середньої взаємодії, які імітують різноманітні послуги (наприклад, FTP, HTTP, MQTT тощо). Ця програма націлена на зловмисників, які атакують хости в Інтернеті за допомогою вразливих служб. Оскільки зловмисники намагаються встановити зловмисне програмне забезпечення в інструмент Dionaea, цей інструмент надає дослідникам можливість аналізувати зловмисне програмне забезпечення та отримати його копію [9].

Адаптивна альтернатива honeypot (AHA) – це самоадаптивний honeypots, який можна використовувати як honeypot з низьким рівнем взаємодії, так і honeypot з високою взаємодією для збору даних. Даний Honeypot SSH використовує теорію ігор і техніку машинного навчання (ML).

Приманку AHA можна використовувати як у середовищі IoT, так і інших класичних середовищах. При цьому було виявлено, що зловмисники реагували на адаптивні honeypots, виконуючи втричі більше взаємодій, реагуючи на настроювані інструменти honeypot. Це демонструє важливість адаптивних honeypots у дослідженні honeypot [10].

За допомогою Cowrie ви можете створювати віртуальні маніпулятори з можливістю масштабування, від середнього до високого рівня взаємодії, які можуть контролювати та контролювати різноманітні поведінки.

Як засіб взаємодії із середовищем, він реєструє взаємодію оболонки зловмисника в симульованій системі UNIX за допомогою емуляції кількох команд. Будучи пристроєм високої взаємодії, він є проксі для SSH і Telnet для спостереження за взаємодією зловмисника в іншій системі. По суті, він діє як проксі-сервер між зловмисником і пулом віртуальних машин, які налаштовані на серверному сервері, що дозволяє гнучко конфігурувати.

Honeypot Cowrie був відокремлений від Honeypot Kippo і імітує послуги SSH, Telnet, SFTP тощо. [11]

HoneyThing створений для Інтернету з підтримкою TR-069 послуг, цей honeypot працює як повноцінний модем/маршрутизатор, на якому працює веб-сервер RomPager і підтримує протокол TR-069 (CWMP). Цей honeypot IoT здатний емулювати популярні вразливості для Rom-0, Misfortune Cookie, RomPager тощо. Він пропонує підтримку протоколу TR-069, включаючи більшість його популярних команд CPE, таких як GetRPCMethods, Get/Set параметрів, Download тощо.

На відміну від інших, цей honeypot пропонує простий та відшліфований веб-інтерфейс. Нарешті, всі важливі дані реєструються у файлі під назвою honeything.log Kako: Конфігурація за замовчуванням запускає ряд симуляцій служби, щоб захопити атакуючу інформацію з усіх вхідних запитів, включаючи весь текст. Він включає в себе сервери Telnet, HTTP і HTTPS.

Для належної роботи Kako потрібні наступні пакети Python: Click, Boto3, Requests і Cerberus. Коли ви отримаєте необхідні пакети, ви можете налаштувати цей honeypot IOT за допомогою простого файлу YAML під назвою kako.yaml. Усі дані записуються та експортуються в AWS SNS і плоский файл у форматі JSON.

1.4 Обґрунтування вибору апаратних ресурсів, мови програмування

Для встановлення приманок можна використовувати різні програмно-апаратні засоби, зокрема: віртуальні сервери (Amazon Elastic Compute Cloud (Amazon EC2); *Google Cloud* Platform, Oracle Cloud).

З використанням програма віртуалізації для операційних систем VirtualBox або VMware та інших.

Розглянемо детальніше переваги та недоліки вказаних способів розгортання приманок.

1) Amazon Elastic Compute Cloud (Amazon EC2) забезпечує масштабовані обчислювальні потужності в хмарі Amazon Web Services (AWS). Використання Amazon EC2 позбавляє вас від необхідності інвестувати в апаратне забезпечення, тож ви можете швидше розробляти та розгортати програми. Ви можете використовувати Amazon EC2, щоб запустити стільки віртуальних серверів, скільки вам потрібно, налаштувати безпеку та мережу, а також керувати сховищем. Amazon EC2 дає змогу збільшувати або зменшувати масштаб, щоб впоратися зі змінами вимог або сплесками популярності, зменшуючи потребу в прогнозуванні трафіку.

2) Google Cloud Platform (GCP) — це набір послуг хмарних обчислень, які працюють на тій самій інфраструктурі, що Google використовує внутрішньо для своїх продуктів для кінцевих користувачів, таких як Пошук Google, Gmail, Google Drive і YouTube. Поряд із набором інструментів керування, він надає ряд модульних хмарних сервісів, включаючи обчислення, зберігання даних, аналітику даних та машинне навчання. Для реєстрації потрібні дані кредитної картки або банківського рахунку.

Google Cloud Platform надає інфраструктуру як послугу, платформу як послугу та безсерверні обчислювальні середовища.

У квітні 2008 року Google анонсувала App Engine, платформу для розробки та розміщення веб-додатків у центрах обробки даних, якими керує Google, яка стала першою службою хмарних обчислень від компанії. Служба стала загальнодоступною в листопаді 2011 року. З моменту анонсу App Engine Google додала до платформи кілька хмарних сервісів.

Google Cloud Platform є частиною Google Cloud, яка включає загальнодоступну хмарну інфраструктуру Google Cloud Platform, а також Google Workspace (G Suite), корпоративні версії ОС Android і Chrome, а також інтерфейси програмування програм (API) для комп'ютера. навчання та картографічні послуги підприємства.

3. Oracle Cloud — це загальнодоступна хмарна послуга компанії баз даних. Oracle позиціонував і адаптував свої хмарні сервіси як найкращий варіант для запуску баз даних. Він є конкурентом лідируючого на ринку конкурента, Amazon. Oracle має глобальну мережу керованих центрів обробки даних, налаштованих для використання своїх хмарних пропозицій. Компанія використовує послуги Oracle Cloud для надання хмарних додатків, серверів, зберігання та обробки даних у цій глобальній мережі. Сама мережа побудована на 25Gb Ethernet; жоден вузол не відокремлений від іншого вузла більш ніж одним переходом із плоскою топологією мережі. Ця мережа дає Oracle впевненість у тому, що вона пропонує міцну угоду про рівень обслуговування (SLA).

Інфраструктура, платформа, програмне забезпечення та послуги даних (IaaS, PaaS, SaaS, DaaS) можна використовувати для розширення або інтеграції програм компанії в хмару Oracle або для створення та розгортання нових. Як віртуалізоване розгортання з кількома орендарями, так і чисті обчислювальні служби можуть бути підключені до одного інтерфейсу прикладної програми (API).

За словами Oracle, їхнє пізніше вступ у хмарні обчислення дозволило їм як навчитися зусиль конкуренції, так і краще спланувати власний вихід.

4. VirtualBox — це програмне забезпечення з відкритим вихідним кодом для віртуалізації обчислювальної архітектури x86. Він діє як гіпервізор, створюючи віртуальну машину (віртуальну машину), де користувач може запускати іншу ОС (операційну систему).

Операційна система, де працює VirtualBox, називається ОС «хост». Операційна система, що працює у ВМ, називається «гостьовою» ОС. VirtualBox підтримує Windows, Linux або macOS як хост-ОС.

Під час налаштування віртуальної машини користувач може вказати, скільки ядер ЦП, а також скільки оперативної пам’яті та дискового простору має бути виділено для віртуальної машини. Коли віртуальна машина працює, її можна "призупинити". Виконання системи на цей момент призупинено, і користувач може відновити її використання пізніше.

VirtualBox спочатку був розроблений компанією Innotek GmbH і випущений 17 січня 2007 року як пакет програмного забезпечення з відкритим вихідним кодом. Пізніше компанія була придбана Sun Microsystems.

27 січня 2010 року Oracle Corporation придбала Sun і взяла на себе розробку VirtualBox.

5. VMware - це програмне забезпечення для віртуалізації, розроблене VMware, Inc. Компанія VMware базується в Каліфорнії, США та була заснована в 1998 році, хоча зараз вона належить корпорації EMC.

Настільні версії VMware (VMware Workstation, VMware Fusion та VMware Player) можна запускати в Windows, Linux та Mac OS X. Однак версії сервера VMware (VMware ESX та VMware ESXi) можуть працювати безпосередньо на апаратному забезпеченні сервера, не вимагаючи операційної системи, оскільки вони використовують гіпервізорну технологію (яка відображає апаратне забезпечення хоста безпосередньо на ресурси віртуальної платформи). VMware Workstation дозволяє працювати з декількома операційними системами x86 або x86-64.

VMware Fusion - подібний продукт, призначений для користувачів Intel Mac. VMware Player - це безкоштовне програмне забезпечення, схоже як на VMware Workstation, так і на VMware Fusion.

Програмне забезпечення VMware забезпечує віртуалізацію адаптерів для відео / мережі / жорсткого диска. Провідних драйверів надає хост для USB-портів та послідовних / паралельних портів.

Отже, віртуальні машини, що працюють на VMware, надзвичайно портативні, дозволяють системним адміністраторам робити паузу на одній

машині, переміщувати її на іншу машину та відновлюватись там, де вона була призупинена.

1.5 Висновки

Пристрої Інтернету речей становлять виклик, коли справа доходить до створення honeypots, якщо дослідники покладаються на традиційні методи, оскільки пристрої IoT мають певні характеристики, на які потрібно звертати увагу. Щоб максимізувати шанси зловмисника необхідно знайти та використати вразливості, при цьому важливо, щоб honeypot залишався анонімним, імітуючи реальну систему, щоб запобігти легкому ідентифікації зловмисниками.

Через особливу природу пристроїв IoT, а також через неможливість повністю зрозуміти природу та діяльність зловмисника, проектування ефективного honeypot потребує іноваційного підходу.

2 ВИЯВЛЕННЯ АТАК НА ІНТЕРНЕТ-РЕЧІ З ВИКОРИСТАННЯМ ПРИМАНОК

2.1 Вразливості безпеки Інтернету речей

Незважаючи на те, що Інтернет - речей робить наше життя зручнішим і комфортнішим, він також відкриває двері для широкого спектру вразливостей безпеки. Насправді, експерти повідомляють про 1,5 мільярда кібератак IoT лише за перші шість місяців 2021 року, у порівнянні з 639 мільйонами за весь 2020 рік.

Підключені пристрої генерують, збирають та обробляють багато даних, і дані IoT дуже цінні для зловмисників. Проблему ускладнює те, що багато пристроїв з'єднані між собою, створюючи складну компромісну мережу, якщо зламати лише один пристрій. При цьому, багатьом підключеним пристроям бракує необхідних вбудованих засобів контролю безпеки для захисту від загроз. В результаті кожен підключений пристрій є потенційною мішенню для атаки.

Безпека IoT передбачає захист підключених пристроїв – і мереж, до яких вони підключені – від фізичних та кіберзагроз. Безпека IoT також включає широкий спектр методів, стратегій, протоколів, стандартів безпеки IoT і дій, спрямованих на захист, виявлення та моніторинг ризиків, допомагаючи усунути різні вразливості.

Онлайн-спільнота Open Web Application Security Project (OWASP), надає безкоштовні та відкриті ресурси, орієнтовані на безпеку, включаючи статті, методології, документацію, інструменти та технології, а також складає список із 10 найбільш поширених вразливостей, які використовують зловмисники.

Вразливості безпеки IoT, яких слід уникати під час створення, розгортання або керування системами IoT. Хоча списку OWASP зараз кілька років, уразливості залишаються в основному такими ж, як і під час створення списку [2].

Вразливість безпеки IoT № 1: слабкі паролі. Незалежні від того, чи є вони слабкими, незмінними, легко відгаданими, загальнодоступними або жорстко закодованими на пристрої за замовчуванням, погані паролі є найпростішим

способом для зловмисників скомпрометувати пристрої Інтернету речей і запустити широкомасштабні кібератаки. Фактично, уряди по всьому світу починають забороняти паролі пристроїв за замовчуванням, які легко вгадати, і вимагають від виробників повідомляти користувачам, як довго розумні пристрої, включаючи телефони, отримають оновлення безпеки. Отже, проектуючи та розвиваючи певний продукт IoT, необхідно, з самого початку вбудувати безпеку та заохочувати користувачів використовувати новий унікальний пароль для кожного пристрою.

Вразливість безпеки IoT № 2: непотрібні або незахищені мережеві послуги. Пристрої IoT, інтегровані в мережі, надсилають та отримують дані, використовуючи різні протоколи зв'язку, від BLE і ZigBee до Wi-Fi, стільникового зв'язку та Ethernet. Коли пристрій під'єднано та спілкується, будь-які непотрібні чи незахищені мережеві служби, що працюють на самому пристрої, можуть поставити під загрозу конфіденційність, цілісність чи доступність інформації або дозволити несанкціонований віддалений контроль. І кіберзлочинці використовують ці непотрібні або незахищені мережеві послуги, щоб спричинити хаос, будь то доступ до конфіденційної інформації, прослуховування приватних повідомлень чи виконання атак типу «Відмова в обслуговуванні» (DoS) і «Людина посередині» (MITM), тому важливо забезпечити щоб порти, які не підключені до жодних важливих мережевих служб, негайно закривалися.

Вразливість безпеки IoT № 3: незахищені інтерфейси екосистеми. Від веб-платформ до серверних API до хмарних або мобільних інтерфейсів – будь-який інтерфейс за межами пристрою або пов'язані з ним компоненти можуть бути скомпрометовані. З найпоширенішими проблемами, включаючи відсутність аутентифікації/авторизації, слабке або відсутність шифрування, а також відсутність фільтрації введення/виводу, занадто багато компаній не звертають увагу на політики та процедури безпеки IoT. Потужний механізм автентифікації та авторизації може допомогти забезпечити, щоб щоразу, коли сервер спілкувався з підключеним пристроєм, він міг відрізнити дійсну кінцеву точку та недійсну, примусово автентифікувати кінцеву точку.

Вразливість безпеки IoT № 4: відсутність безпечного механізму оновлення. Що стосується безпеки пристрою IoT, можливість безпечного оновлення програмного та мікропрограмного забезпечення пристрою є важливою. Несанкціоновані оновлення є основним вектором загроз для атак на пристрої IoT, і організації в різних галузях часто намагаються підтримувати свої системи IoT в актуальному стані, оскільки виробники пристроїв надають оновлені виправлення безпеки лише в поодинокі моменти.

Отже, підключені пристрої повинні мати можливість отримувати оновлення по ефірі (OTA), які дозволяють «виправляти» найновіші вразливості апаратного, програмного та мікропрограмного забезпечення пристрою через бездротову мережу.

Вразливість безпеки IoT № 5: небезпечні або застарілі компоненти. Якщо ваш пристрій використовує застарілі або незахищені програмні компоненти, бібліотеки чи фреймворки, проблеми можуть виникнути через вразливості залежностей від програмного забезпечення або застарілих систем. Стороннє програмне забезпечення, яке використовується для створення пристроїв Інтернету речей, важко відстежити, і воно вразливе до кібератак, якщо ним не керувати належним чином. На жаль, багато застарілих систем використовують традиційні протоколи оновлення програмного забезпечення, які змушують нічого не підозрюючих користувачів знаходити та виправляти будь-які вразливості. Але оскільки сьогоднішня хмарна інфраструктура безпеки стає популярнішою, важливо залишатися в курсі будь-якого мікропрограмного або програмного забезпечення, щоб забезпечити безпеку вашої системи Інтернету речей.

Вразливість безпеки IoT № 6: неналежний захист конфіденційності. Враховуючи дуже цінну природу конфіденційної особистої інформації користувача, захист конфіденційності є важливою в нашому підключеному світі. Багато підключених пристроїв збирають персональні дані, які необхідно безпечно зберігати та обробляти, щоб відповідати різним нормам конфіденційності, включаючи «Загальний регламент захисту даних Європи (GDPR)» або «Каліфорнійський закон про конфіденційність споживачів (CCPA)». Однак,

оскільки у звіті про витрати на порушення даних за 2021 рік зазначено, що 44% витоків даних містять особисту інформацію, очевидно, що відсутність захисту конфіденційності піддає приватну інформацію користувачів ризику і, що ще гірше, може призвести до суворих правових наслідків.

Вразливість безпеки IoT № 7: незахищена передача та зберігання даних. Як згадувалося кілька разів, деякі дані IoT надзвичайно цінні для поганих акторів. Коли конфіденційні дані не зашифровані або не мають контролю доступу в будь-якій частині екосистеми, незалежно від того, знаходяться вони в стані спокою, передаються чи обробляються, вони можуть бути вкрадені та використані для підступних цілей. Оскільки ці дані часто використовуються для автоматизованого прийняття рішень і для керування пристроями, дуже важливо, щоб вони залишалися захищеними, коли інформація передається по мережі або зберігається в новому місці. Отже, найкращі методи передачі та зберігання даних IoT вимагають надійних приватних і публічних ключів шифрування.

Вразливість безпеки IoT № 8: відсутність керування пристроями. Суттєвою проблемою для Інтернету речей є керування всіма розрізненими пристроями та захист периметра мережі. Іноді шахрайський або шкідливий пристрій встановлюється без дозволу на збір або зміну конфіденційної інформації. Як і у випадку з керуванням усіма IT-активами, ключові проблеми керування пристроями IoT включають надання, експлуатацію та оновлення всіх пристроїв і шлюзів.

Традиційна ідентифікація пристроїв IoT за допомогою IP-адрес і базових операційних систем не працює для IoT. Натомість вам потрібно визначити конкретні пристрої для планування вимог до доступу до мережі, тактики розгортання, оптимізації стратегії безпеки та оперативних планів. Після визначення ідентифікаційних даних пристроїв системи безпеки можуть відстежувати поведінку пристрою, щоб розпізнати шкідливі моделі зв’язку, перш ніж вони завдадуть пошкодження.

Вразливість безпеки IoT № 9: незахищені параметри за замовчуванням. Як і паролі за замовчуванням, налаштування пристрою за замовчуванням можуть бути

проблематичними. Обмежуючи операторів змінювати конфігурації або запускати підключені продукти з незахищеними налаштуваннями за замовчуванням, виробники пристроїв IoT відкривають двері для компромісу. Як тільки зловмисник порушує систему, він може використати вразливості в мікропрограмі пристрою або знайти приховані бекдори. Розпізнавання цих налаштувань за замовчуванням, які важко змінити, і недоліків у безпеці, які вони створюють, є життєво важливими для впровадження відповідних засобів керування для кращого захисту цих пристроїв.

Вразливість безпеки IoT № 10: відсутність фізичного захисту. На жаль, багато пристроїв IoT часто вразливі для атак, оскільки їм бракує необхідної вбудованої безпеки для протидії загрозам. Без заходів фізичної безпеки зловмисники мають можливість отримати доступ до конфіденційної інформації, яка може допомогти в майбутній віддаленій атаці або взяти локальний контроль над пристроєм.

Оскільки підключені пристрої розгортаються в розсіяних і віддалених середовищах, зловмисники, які можуть отримати доступ до фізичного рівня та втрутитися в нього, можуть порушити надання послуг пристрою. Хоча протоколи безпеки захищають дані під час передачі, дані, що зберігаються на самому пристрої, залишаються незахищеними, тому необхідно переконатися, що обладнання захищене від втручання, фізичного доступу, маніпуляцій та саботажу.

Незважаючи на неймовірну цінність, яку IoT надає як споживачам, так і підприємствам, неналежні методи безпеки можуть призвести до катастрофічних наслідків, які завдають значно більше шкоди, ніж користі.

Оскільки Інтернет речей стає все більш розповсюдженим, усвідомлення основних вразливостей безпеки IoT може допомогти впевнитися, що ваші пристрої та мережі не стануть жертвами шкідливої кібератаки. Експерти з наскрізної безпеки IoT, наголошують, що неможливо захистити пристрій, мережу та систему IoT від будь-яких атак, які можна уявити.

Однак, залишаючись в курсі найпопулярніших уразливостей, можна запобігти зловмисникам компрометації вашої системи. Те, яким чином буде спроектована система honeypot, залежить від завдань, які вона має вирішувати.

Якщо необхідно вивчити мотивації поводження зловмисників, методи їхніх атак і засобів – тоді потрібно побудувати складну honeypot, що надає зловмисникові повноцінну операційну систему (ОС), з якою він буде взаємодіяти, що забезпечить високий рівень протоколювання. Якщо необхідно виявити несанкціоновану активність, таку як сканування системи, то для цих цілей можна побудувати просту honeypot, що буде імітувати мінімальні можливості й операції сервісів, записуючи лише команди взаємодії зі зловмисником.

2.2 Класифікація honeypot

Класифікувати приманки можна кількома різними способами. Для початку їх можна розділити на категорії залежно від того, наскільки вони взаємодіють і яка їх мета. За своїм призначенням приманки мають два види використання: або для дослідження, або для виробництва. «Виробничі приманок використовується для захисту компанії, тоді як дослідницький маніпулятор використовується, щоб дізнатися більше про компанію» [5].

Найчастіше виробничі приманки використовуються в середовищі організації, щоб мінімізувати ризики, які зловмисник може створити для цієї організації. Таким чином, вони здатні виявляти атаки, і їх набагато легше створити та налаштувати, оскільки вони мають менше функцій, з якими можна боротися.

Honeypots, які використовуються як частина виробничого середовища, мають переваги запобігання помилкових спрацьовувань, таких як ті, які існують у традиційних системах виявлення вторгнень [6].

Метою цих пристосувань є забезпечення меншого ризику в мережі в разі компромісу, але вони надають мало інформації про зловмисника або атаку. Крім того, виробничі honeypots можна легко обслуговувати і можуть забезпечувати механізми безпеки, такі як виявлення, запобігання та реагування, які можна використовувати для захисту мережі організації. На відміну від системи запобігання, дослідницькі приманки в першу чергу призначені для збору інформації про зловмисників, а не для стримування.

Як додаткову перевагу, ця інформація може бути корисною для аналізу зловмисників, наприклад, хто є суб'єктами загрози, які їхні цілі, які інструменти чи методи були використані для атаки на систему тощо. Цей тип інформації опосередковано покращує безпеку ресурсів кількома способами.

Щоб краще зрозуміти зловмисника та його мотиви, нам потрібно щось схоже на справжній комп'ютер та операційну систему. Тому конфігурація дослідницьких приманок стає дедалі складнішою та займає багато часу.

Оскільки вони зазвичай створюються як «справжні» підроблені системи, це створює більший ризик для організації та «потенційно знижує безпеку, оскільки вони вимагають великих ресурсів та обслуговування» [5].

Рівень взаємодії також може бути використаний для категоризації honeypots. Зазвичай існує три типи взаємодії: низька взаємодія, середня взаємодія та висока взаємодія (рисунок 2.1).

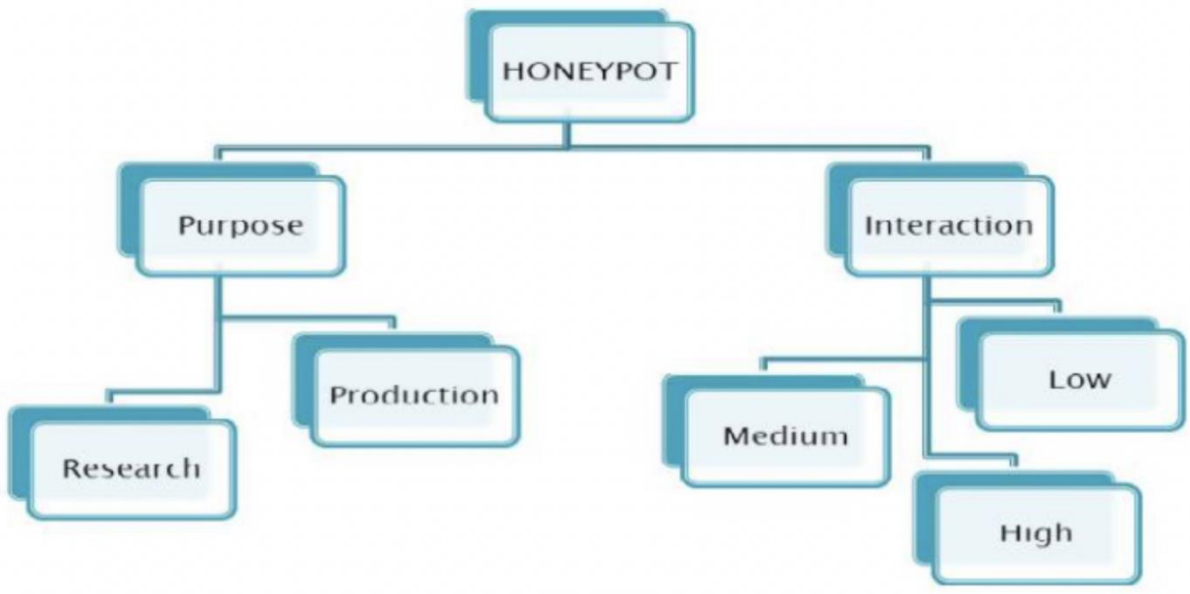


Рисунок 2.1 – Типи приманок

Honeypots з низьким рівнем взаємодії зазвичай не мають операційної системи. Таким чином, зловмисник обмежується лише спробою входу в систему. Серед служб, які можуть бути включені, є Telnet, SSH і FTP, але honeypot не пропонує нічого іншого. Подібні Honeypots зазвичай використовуються у

великому виробництві, оскільки їх легко налаштувати і навряд чи будуть повністю скомпрометовані зловмисниками.

Honeypots із середньою взаємодією також, як правило, не мають операційних систем, як і honeypots з низькою взаємодією. Хоча це може бути правдою, вони все ще мають тенденцію надавати імітовані послуги вищого рівня, які є бажаною метою для хакерів.

Крім того, ці приманки мають більш складну структуру, що ускладнює їх налаштування. Причина цього полягає в тому, що їм потрібно зібрати більше інформації та зрозуміти, як зловмисник буде вести себе під час атаки, наприклад, як система була скомпрометована та як використовуються інструменти та методи.

Найскладнішими системами є honeypots з високим ступенем взаємодії. Процес їх встановлення та підтримки може бути значно складнішим, оскільки вони забезпечують більш відкрите середовище для використання зловмисниками. Використання цих методів дозволить з часом зібрати набагато більше даних про поведінку атаки, а також про методологію. Загалом, вищі винагороди пов’язані з більшими ризиками. Через те, що ці системи реальні, вони можуть поставити під загрозу мережу організації. Щоб ці системи працювали в безпечному середовищі, зазвичай вони повинні працювати за брандмауером у контрольованому середовищі. Незважаючи на те, що зловмисник зможе отримати доступ до honeypot, брандмауер не дозволить йому отримати подальшу вигоду від нього. Зазвичай ці приманки використовуються дослідниками для цілей дослідження.

Існує ще одна класифікація, яка допомагає з’ясувати різницю між фізичним і віртуальним приманками.

Honeypots і honeynets можна класифікувати різними способами. Для того, щоб класифікувати honeypots та honeynets для IoT було використано попередні дослідження [12], [24]. Класифікація honeypots і honeynets для IoT виконана відповідно до їх призначення, ролі, рівня взаємодії, масштабованості, рівня ресурсів, доступності вихідного коду та їх застосування, як показано на рисунку 2.2.

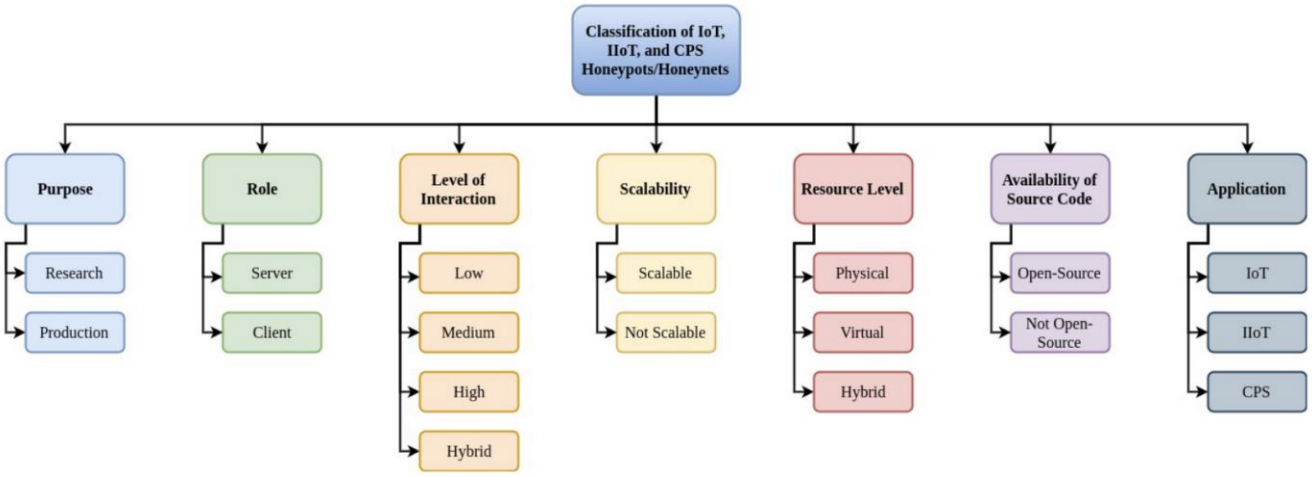


Рисунок 2.2 – Категорії класифікації honeypots і honeynets для IoT

Класифікація за призначенням: меді можна розділити на два класи залежно від мети, для якої вони були створені: дослідницькі та виробничі медники. Дослідницькі honeypots використовуються для збору та аналізу інформації про атаки, щоб розробити кращий захист від цих атак. Виробничі меді більше орієнтовані на захист. Вони є зазвичай реалізується для запобігання доступу зловмисника до актуальна система організації, яка його реалізує [13].

Класифікація за роллю: роль відноситься до того, чи приманка активно виявляє або пасивно захоплює трафік. Клієнтський honeypot може активно ініціювати запит до сервера на дослідження шкідливої програми, поки сервер honeypot чекає атак. Переважна більшість honeypots є серверними honeypots [12].

Класифікація за рівнем взаємодії: Honeypots можна класифікувати за рівнем взаємодії, який вони дозволяють зловмиснику: низька взаємодія, середня взаємодія, висока взаємодія та гібрид. Приманки з низьким рівнем взаємодії імітують одну або кілька служб з простими функціями і не надають доступу до операційної системи. Перевагами honeypots з низькою взаємодією є простота налаштування, низький ризик, низька вартість і низький рівень обслуговування. Проте, приманки з низьким рівнем взаємодії набагато легше ідентифікуються зловмисниками через їх обмеженість, а інформація, яку вони збирають, обмежена та має низьку точність [28].

Приманки високої взаємодії забезпечують набагато більше взаємодії, не тільки емулюючи послуги, але й надаючи доступ до операційна система [28]. У

той час як деякі дослідження посилаються на високу взаємодію, коли honeypot створюється за допомогою реальних пристроїв, інші роботи також включають віртуальні середовища, які емулюють повні пристрої та послуги як високу взаємодію.

Приманки з високою взаємодією збирають інформацію про всі рухи та дії зловмисника, що є перевагою приманок високої взаємодії, оскільки зібрана інформація має високу точність. Однак вони мають високий ризик, оскільки все, до чого вони надають доступ зловмисникам, використовується на реальних ресурсах для збору додаткової інформації.

Більше того, їх складніше налаштувати, вони збирають набагато більше даних, їх складніше підтримувати та запускати [28]. Як тільки вони скомпрометовані, їх перебудова стає необхідною. Крім того, зловмисники можуть скомпрометувати їх, щоб атакувати інші цілі, що створює проблеми з відповідальністю

Як видно з назви, приманки середньої взаємодії забезпечують рівень взаємодії між низьким і високим рівнем взаємодії. Хоча існують різні точки зору щодо того, чи мають вони реальну операційну систему чи емульовану операційну систему, вони імітують більше сервісів, ніж приманки з низьким рівнем взаємодії, забезпечуючи більше взаємодії, що збільшує ризик і ускладнює їх виявлення порівняно з низьким рівнем взаємодії приманок.

Поєднання приманок з різними рівнями взаємодії реалізована в одній системі називається гібридною мережею. Гібридні медові сітки здатні забезпечити кращий баланс шляхом надання переваг кожного типу меду [29].

Класифікація за масштабованістю: масштабованість відноситься до здатності приманок розвиватися і забезпечувати більше приманок. Немасштабована приманка має лише певну кількість пасток і не може бути змінений. Масштабований honeypot може розширити кількість приманок, які він розгортає та контролює [12].

Масштабованість важлива, оскільки різні honeypots, реалізовані разом в honeynet, забезпечують більший захист, послуги, збір даних і різноманітність

даних порівняно з одним honeypot. Фізичні honeypots зазвичай важче масштабувати через необхідні ресурси. Honeypots з високою взаємодією також, як правило, мають нижчу масштабованість через їх складність.

Класифікація за рівнем ресурсів: типи ресурсів, які використовуються для створення системи honeypot, можуть бути фізичними або віртуальними.

Фізична система honeypot складається з кількох honeypot, що працюють на фізичних машинах, тоді як система віртуальних honeypot складається з віртуальних honeypot, які розміщені на одній або кількох фізичних машинах.

Фізичні honeypots мають високу взаємодію та мають більшу точність захоплення даних, ніж віртуальні honeypots. Однак вони дорожчі та потребують більше ресурсів для реалізації.

Віртуальні honeypots вимагають менше ресурсів для впровадження і, отже, менш витратні. Гібридний honeynet, який використовує як фізичні, так і віртуальні honeypots, здатний краще збалансувати вартість та точність захоплення даних [22].

Фізичні honeypots будуть змонтовані в мережі організації, тоді як honeypots, які є віртуальними, будуть встановлені на хості віртуалізації за допомогою VirtualBox або VMware і переглянуть будь-який мережевий трафік, який надсилається через мережу. Перевага віртуальних приманок полягає в тому, що їх можна використовувати для розміщення кількох приманок в одній системі.

Класифікація за наявністю вихідного коду: відкритий вихідний код відноситься до вихідного коду програмного забезпечення, що випускається таким чином, що будь-хто може мати до нього доступ, змінювати його та/або поширювати. Програмне забезпечення з відкритим вихідним кодом дозволяє спільну розробку. Не всі автори honeypot і honeynet надають вихідний код своїх систем приманки. Доступ до вихідного коду дає змогу іншим дослідникам і розробникам зрозуміти й покращити існуючі honeypots та honeynets.

Класифікація за застосуванням. Додаток відноситься до передбачуваного застосування, для якого створена система honeypot.

У даній класифікації визначено системи IoT honeypot як загального користування, IoT або Smart Home IoT. Приманки загального використання − це ті, які спочатку не були створені для IoT. Однак вони є актуальними, оскільки згодом вони були використані в дослідженнях з IoT honeypots. IoT honeypots націлені на загальні додатки IoT. IoT Smart Home honeypots − це honeypots, зосереджені на програмах для використання розумного дому.

2.3 Типи атак на Інтернет-речей

Незважаючи на те, що IoT робить наше життя зручнішим і комфортнішим, він також відкриває двері для широкого спектру вразливостей безпеки. Підключені пристрої генерують, збирають та обробляють багато даних, і дані IoT дуже цінні для поганих суб'єктів.

Проблему ускладнює те, що багато пристроїв з'єднані між собою, створюючи складну компромісну мережу, якщо зламати лише один пристрій. При цьому, багатьом підключеним пристроям бракує необхідних вбудованих засобів контролю безпеки для захисту від загроз.

В результаті кожен підключений пристрій є потенційною мішенню для атаки.

Безпека IoT передбачає захист підключених пристроїв – і мереж, до яких вони підключені – від фізичних та кіберзагроз. Для підвищення ефективності приманки розглянемо найбільш поширені типи атак [10].

1. Фізичні атаки. Фізичні атаки відбуваються, коли будь-хто може фізично отримати доступ до пристроїв IoT. Оскільки більшість атак на кібербезпеку відбуваються зсередини компанії, важливо, щоб ваші пристрої Інтернету речей перебували в захищеній зоні, що часто не є варіантом. Багато атак фізичної кібербезпеки починаються з того, що зловмисник вставляє USB-накопичувач для поширення шкідливого коду, тому як ніколи важливо додати заходи безпеки на основі штучного інтелекту, щоб забезпечити захист ваших пристроїв і даних.

2. Атаки шифрування. Коли пристрій IoT не зашифрований, зловмисник може пронюхати дані та захопити їх для використання пізніше. Крім того, «як тільки ключі шифрування будуть розблоковані, кібер-зловмисники можуть встановити власні алгоритми та взяти під контроль вашу систему». З цих причин шифрування є обов'язковим у середовищі IoT як частина ваших зусиль з кібербезпеки.

3. DoS (відмова в обслуговуванні). DoS-атака відбувається, коли служба, наприклад веб-сайт, стає недоступною. Велика кількість систем атакує одну ціль через ботнет, що змушує багато пристроїв одночасно запитувати послугу. Хоча зловмисники в цьому випадку, як правило, не прагнуть захопити дані, вони серйозно впливають на бізнес, якщо послуги стають недоступними.

4. Викрадання прошивки. Якщо ви не відстежуєте оновлення мікропрограмного забезпечення IoT, ви ризикуєте атаки кібербезпеки. Обов'язково перевірте, чи надходять ваші оновлення з очікуваного джерела, інакше зловмисник може захопити пристрій і завантажити шкідливе програмне забезпечення. Ще варто пам'ятати, що більшість виробників апаратного забезпечення не підписують криптографічно вбудоване програмне забезпечення.

5. Ботнети. Розглянемо атаку на ботнет, Mirai, яка перетворила мережеві пристрої IoT на дистанційно керованих ботів, які можна використовувати як частину ботнету. Ботнети мають можливість використовувати розумні підключені пристрої для передачі конфіденційних, конфіденційних корпоративних даних, які можуть продаватися в темній мережі, або для вимкнення пристрою. Mirai продовжує залишатися проблемою сьогодні, оскільки постраждали мільйони пристроїв IoT.

6. Людина посередині. Атака «людина посередині» відбувається, коли хакер порушує зв'язок між двома окремими системами. Таємно перехоплюючи комунікації між двома сторонами, цей тип атаки змушує одержувача подумати, що він отримує законне повідомлення. Іншими словами, чоловік посередині починає спілкуватися з обома сторонами, звідси й назва. Це може виглядати як електронний лист від вашого банку з проханням увійти для виконання завдання.

Тепер фальшивий веб-сайт зловмисників збирає ваші облікові дані, тому зловмисник може завдати додаткової шкоди.

7. Програми-вимагачі. Програми-вимагачі – це тип зловмисного програмного забезпечення, яке блокує доступ до файлів, шифруючи їх. Потім зловмисники продають вам ключ дешифрування, щоб знову отримати доступ до ваших файлів. Звичайно, цей тип атаки може порушити повсякденну роботу, а ключ шифрування часто коштує дуже дорого. Уявіть собі, якби хакери змогли отримати доступ до електромережі і протягом кількох днів відмовлялися віддавати ключі. Позначте затемнення.

8. Підслуховування. У цьому типі атаки хакер перехоплює мережевий трафік, щоб вкрасти конфіденційну інформацію через ослаблене з'єднання між пристроєм IoT і сервером. Підслуховування, як правило, здійснюється шляхом прослуховування цифрового або аналогового голосового зв'язку або шляхом перехоплення перевірених даних. Знову ж таки, у цьому випадку зловмисник забирає конфіденційні корпоративні дані.

9. Підвищення привілеїв. Хакери шукають помилки та слабкі місця пристроїв Інтернету речей, щоб отримати доступ до ресурсів, які зазвичай захищені програмою або профілем користувача. У цьому типі атаки хакер намагається використати свої нещодавно отримані привілеї для розгортання шкідливих програм або крадіжки конфіденційних даних.

10. Brute Force Password Attack. У цьому сценарії хакери надсилають багато паролів або парольних фраз з надією вгадати правильний, надаючи їм доступ до ваших пристроїв Інтернету речей. Або вони використовують програмне забезпечення для створення великої кількості послідовних припущень. Тепер, коли зловмисник має доступ до вашого пристрою, він може встановити зловмисне програмне забезпечення або викрасти критично важливі для бізнесу дані. Незалежно від того, чи тільки ви починаєте працювати з Інтернетом речей, чи вже впровадили пристрої, важливо регулярно проводити перевірку кібербезпеки, щоб визначити, чи потрібно вживати додаткових заходів для захисту своїх пристроїв.

Завжди треба бути пильним щодо своєї кібербезпеки, щоб бути на крок попереду зловмисників.

2.4 Проектування структури honeypots

На рисунку 2.3 зображена базова архітектура honeynet. Існує три основні елементи будь-якої мережі Honeynet: контроль даних, збір даних і збір даних.

Контроль даних передбачає контроль над потоком даних, щоб зловмисники не усвідомлювали цього знаходяться в медовій мережі і переконайтеся, що якщо медова мережа скомпрометований, він не використовуватиметься для атаки на інші системи.

Збір даних включає збирання всіх даних, що стосуються рухи та дії всередині медової мережі [16]. Дані Колекція передбачає можливість безпечного перенесення всіх отримані дані в централізоване місце [12].

Honeypots і honeynets можуть бути розгорнуті в різних місцях цій. Вони можуть бути розгорнуті в хмарних обчислювальних середовищах (наприклад, Amazon EC2), демілітаризованих зонах (DMZ) корпоративних мереж, реальних прикладних/виробничих середовищах (наприклад, у мережі IoT, IIoT або CPS) і приватних середовищах розгортання з публічною IP-адресою. адреси.

Кожен з цих варіантів розгортання має свої переваги та недоліки. Крім того, рішення середовища розгортання може вплинути на вибір найбільш підходящого типу меду або медова мережа.

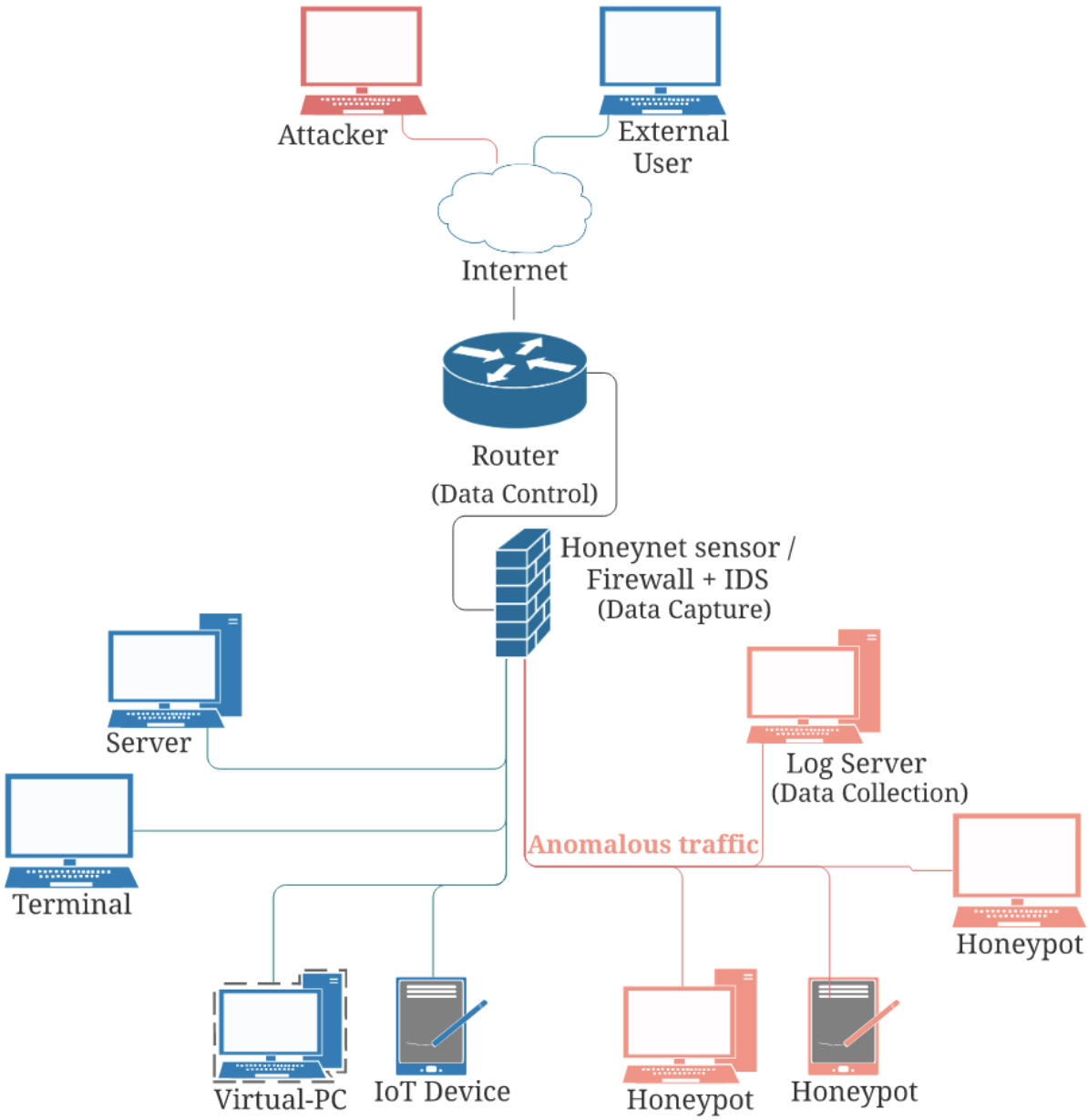


Рисунок 2.3 – Базова архітектура honeypots

Структура розробленої приманки приведена на рисунку 2.4. При реалізації приманки за основу використано honeypot Cowrie з відкритим програмним кодом [15]. Хмарний інструмент SIEM на основі аналітики – Splunk дозволяє виявляти, досліджувати, відстежувати кіберзагрози та реагувати на них. Він дозволяє вводити дані з локальних і мультихмарних розгортань, щоб отримати повну видимість у вашому середовищі для швидкого виявлення загроз.

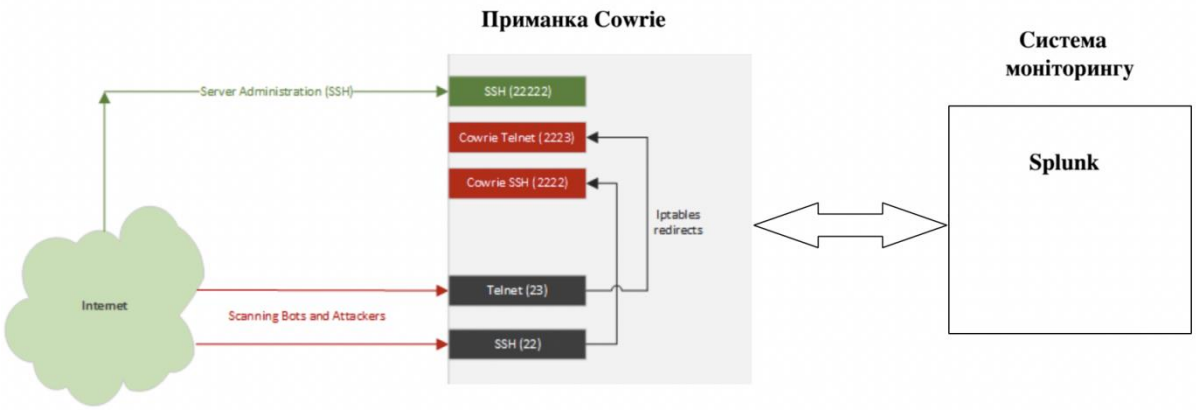


Рисунок 2.4 –Структура системи виявлення загроз

Splunk використовує машинне навчання для виявлення розширених загроз і автоматизує завдання для швидшого вирішення.

2.5 Висновки

Розкрито основні уразливості безпеки Інтернету речей, який ґрунтується на Top 10 спільноти OWASP. Усвідомлення основних вразливостей безпеки Інтернет-речей може допомогти впевнитися, що пристрої та мережі не стануть жертвами шкідливої кібератаки.

Розроблена класифікація honeypot, яка враховує їх призначення, ролі, рівні взаємодії, масштабованість, рівні ресурсів, доступність вихідного коду та застосування.

Досліджено найбільш поширені типи атак на Інтернет- речей., зокрема, фізичні атаки, атаки шифрування, DoS (відмова в обслуговуванні), викрадання прошивки, ботнети, людина посередині, програми-вимагачі, підслуховування, підвищення привілеїв, атака грубою силою.

Розроблено структуру honeypot, яка враховує особливості Інтернет-речей, зокрема способи віддаленого підключення.

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ

3.1 Налаштування віртуального сервера

Для розгортання приманки використаємо хмарний сервіс DigitalOcean [16].

DigitalOcean Droplets – це віртуальні машини (VM) на базі Linux, які працюють поверх віртуалізованого обладнання. Кожна створена користувачем Droplets – це новий сервер, який можна використовувати як самостійний, так і як частину більшої хмарної інфраструктури.

Щоб отримати доступ до панелі керування DigitalOcean і створити Droplets, потрібен мати обліковий запис DigitalOcean (рисунок 3.1). Це необхідно створити на сторінці реєстрації нового облікового запису DigitalOcean [17].

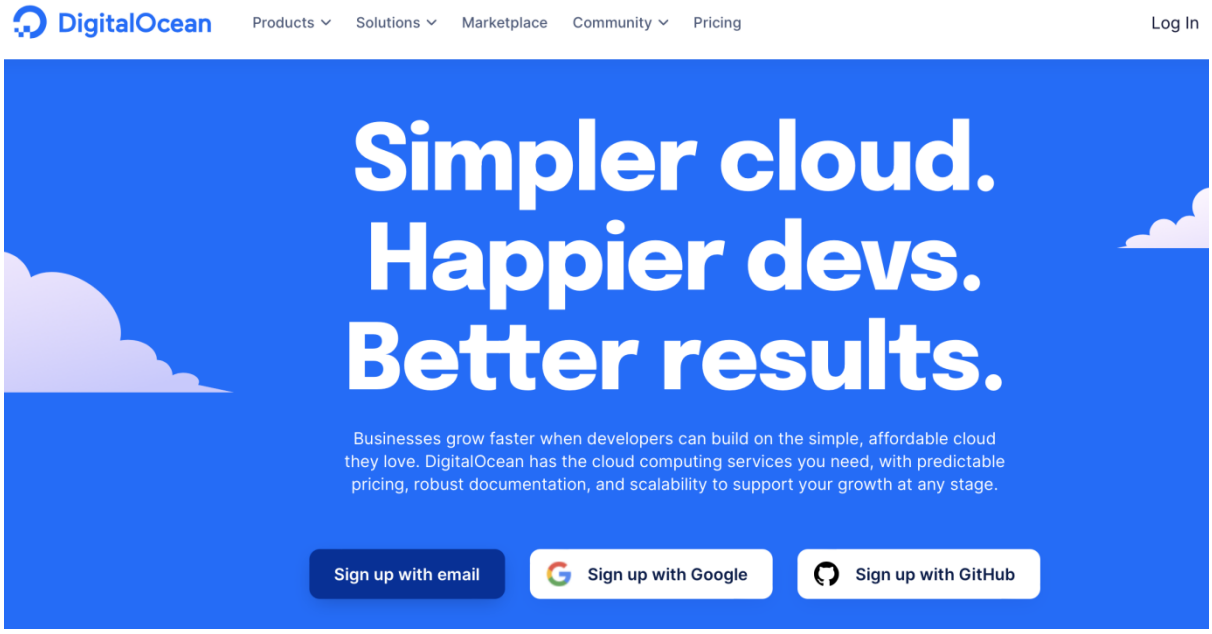


Рисунок 3.1 – Створення облікового запису

Вибираємо вхід через email і заповнюємо обов’язкові поля (рисунок 3.2).

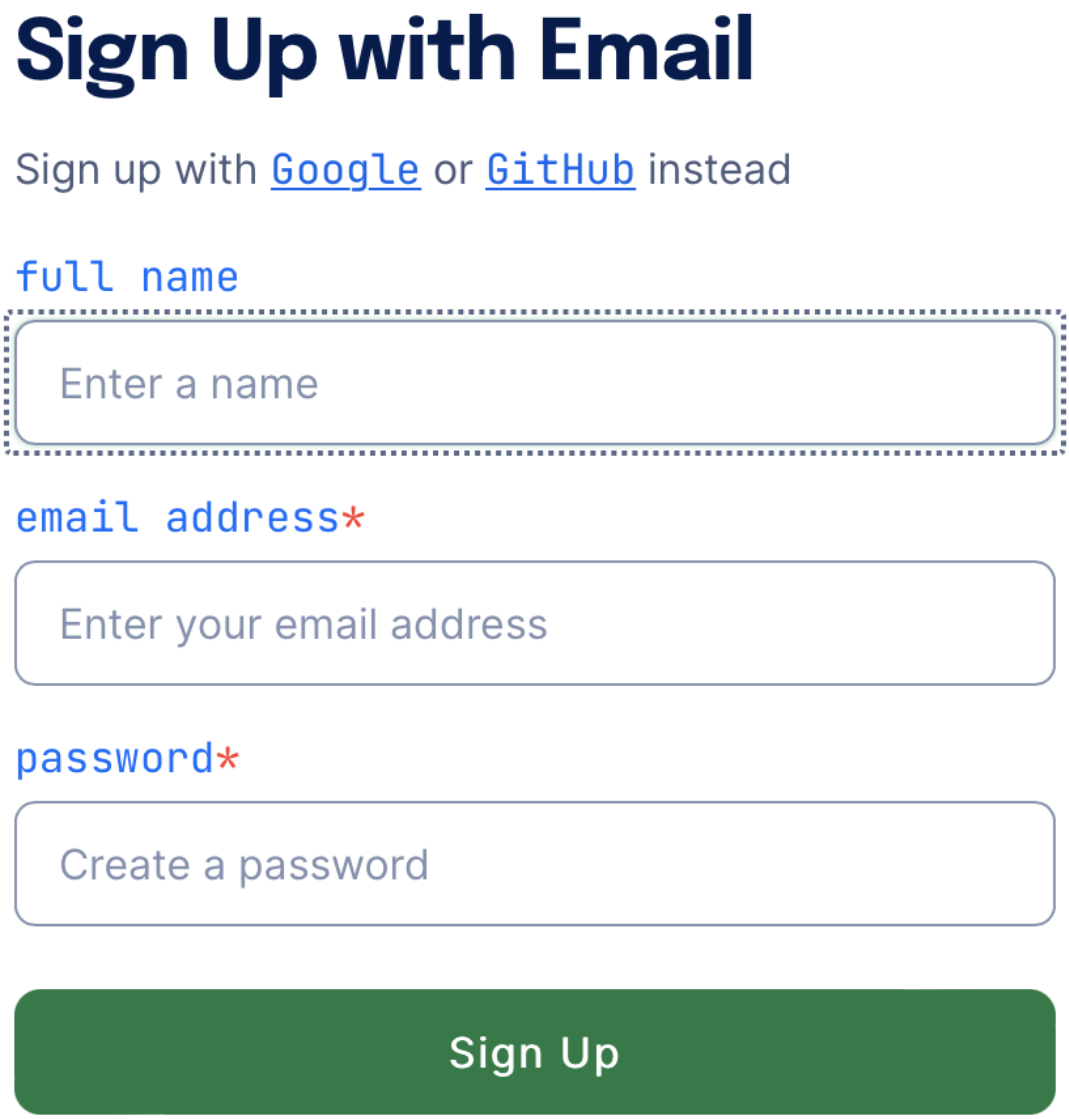


Рисунок 3.2 – Реєстрація через email

Після перевірки електронної пошти, необхідно вибрати метод оплати і ввести дані банківської картки (рисунок 3.3). Кошти, які будуть зняті в процесі реєстрації будуть миттєво повернуті на банківську картку.

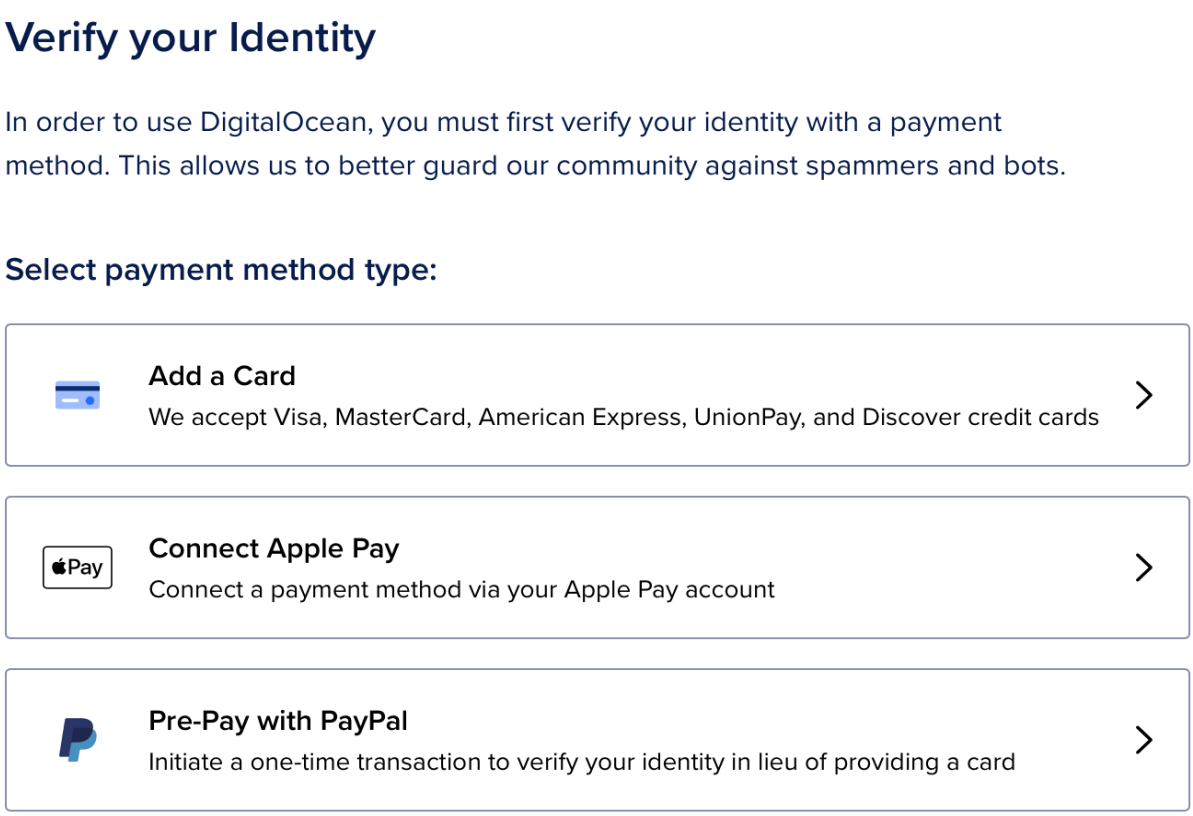


Рисунок 3.3 – Вибір способу оплати

Після успішної реєстрації переходимо до подальших налаштувань (рисунок 3.4). На сторінці створення Droplets вибираємо конфігурацію Droplets, наприклад її операційну систему, обсяг пам’яті та додатково можна увімкнути деякі функції (наприклад, резервне копіювання або моніторинг) увімкнути. Найпопулярніші параметри за замовчуванням попередньо вибрані, тому можна прокрутити сторінку донизу та негайно створити Droplets, або можна налаштувати будь-який з параметрів у кожному розділі.

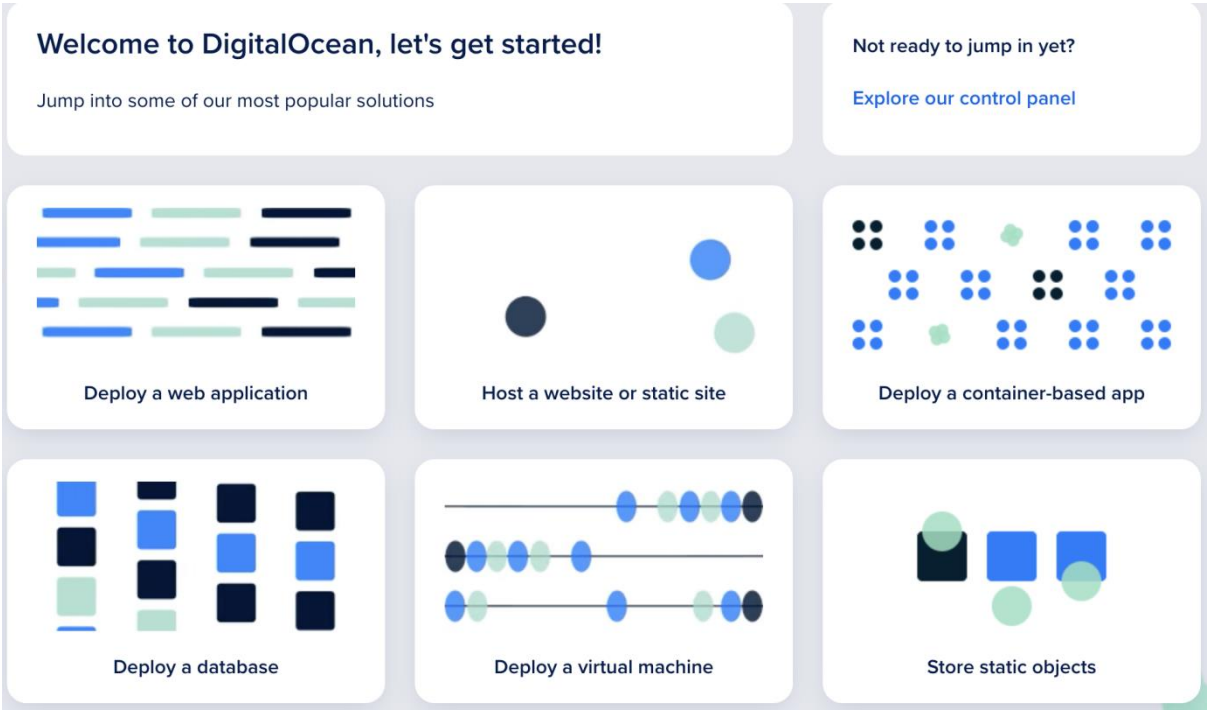


Рисунок 3.4 – Початкова панель налаштування DigitalOcean

Після входу в панель керування натисніть зелену кнопку Створити у верхньому правому куті, щоб відкрити меню створення.

У меню створення натискаємо Droplets, щоб відкрити сторінку створення Droplets. Якщо ще немає Droplets, на вкладці «Ресурси» відобразиться велика синя кнопка «Почати роботу з Droplets», яка перенаправить на сторінку створення Droplets.

У розділі «Choose an image» вибираємо образ, з якого буде створена Droplets. Спочатку можна вибрати одну з чотирьох категорій образів (рисунок 3.5).

Дистрибутиви – це базові образи, схожі на Unix, без додаткового програмного забезпечення, включаючи Ubuntu, Fedora, Debian, CentOS і FreeBSD.

Контейнерні дистрибутиви включають RancherOS.

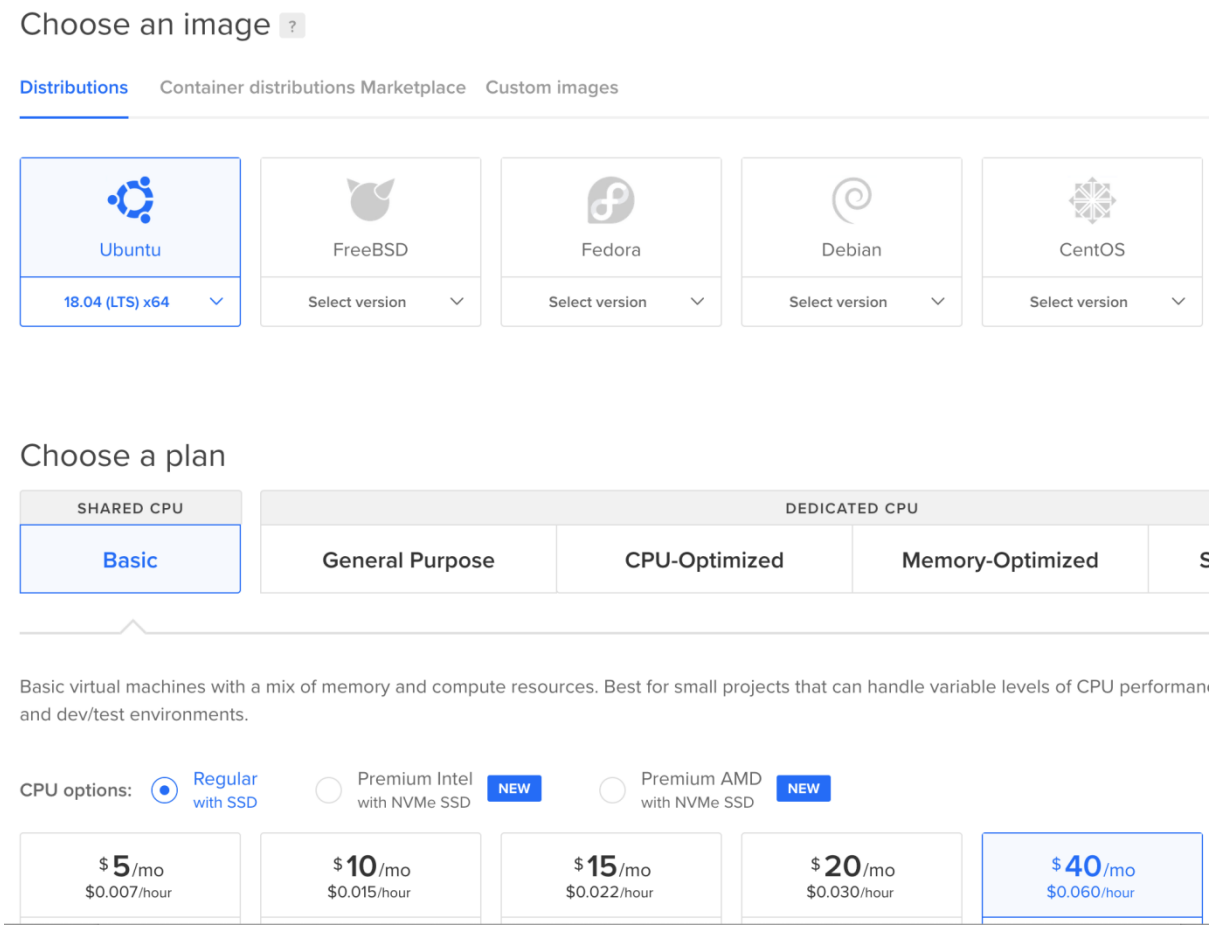


Рисунок 3.5 – Вибір операційної системи та тарифного плану

Образ Marketplace включають попередньо налаштовані програми, як-от Docker, WordPress або LAMP, щоб спростити початок роботи.

Деякі образи Marketplace, як-от WordPress, дозволяють додати кластер керованої бази даних під час створення. Облікові дані для підключення до кластеру бази даних доступні на панелі керування, у файлі на Droplet (/root/.digitalocean_dbaas_credentials і MongoDB сертифікати на /root/dbaas_ca_cert.crt) і експортуються як змінні середовища на Droplet з /etc. /навколишнє середовище.

Спеціальні образи – це зображення, подібні до Unix, які ви створюєте та завантажуєте у свій обліковий запис DigitalOcean.

Вибір тарифного плану. Існує чотири типи планів:

1. Basic Droplets, гнучкий варіант, який найкраще підходить для більшості випадків використання, як-от хостинг веб-сайтів, проміжне середовище та

потреби низької інтенсивності обчислень. Для базових Droplets також можна вибрати звичайні та преміум-процесори. Плани ЦП преміум-класу поставляються з деякими з найсучасніших процесорів Intel або AMD і SSD NVMe.

2. General Purpose Performance Droplets, надійний варіант, який підходить для широкого спектру основних виробничих додатків, які вимагають більш високого співвідношення пам’яті та ЦП. Droplets продуктивності, оптимізовані для ЦП, найкраще підходять для завдань із інтенсивним процесором та проектів, які вимагають передбачуваної продуктивності або покладаються на ЦП більше, ніж на оперативну пам’ять чи введення/виводу, як-от пакетна обробка великих наборів даних, великі збірки та кодування відео.

3. Оптимізовані для пам’яті Droplets продуктивності, найкращі для ресурсомістких бізнес-додатків, таких як високопродуктивні бази даних SQL або NoSQL, великі кеші та індекси в пам’яті, обробка великих даних у реальному часі та програми з великими вимогами до JVM.

4. Оптимізовано для зберігання, найкраще підходить для великих баз даних NoSQL (наприклад, MongoDB і Elasticsearch), баз даних часових рядів та інших сховищ даних.

Для Droplets з менш ніж 3 ГБ оперативної пам’яті рекомендується використовувати 32-розрядну операційну систему. Процеси вимагають більше пам’яті на 64-розрядних архітектурах, а на серверах з обмеженою кількістю оперативної пам’яті будь-які переваги продуктивності від 64-розрядної ОС компенсуються меншим обсягом пам’яті для буферів і кешування.

Після заповнення необхідних полів та натискання кнопки створити почнеться процес установки операційної системи. Результат приведений на рисунку 3.5.

Вибір способу аутентифікації. У розділі «Автентифікація» вибираємо метод, який будемо використовувати для входу до Droplet. Є два варіанти (рисунок 3.6):

1) SSH keys, які забезпечують більше безпеки, ніж пароль;

2) Password, який дозволяє створити власний пароль для нової Droplet.

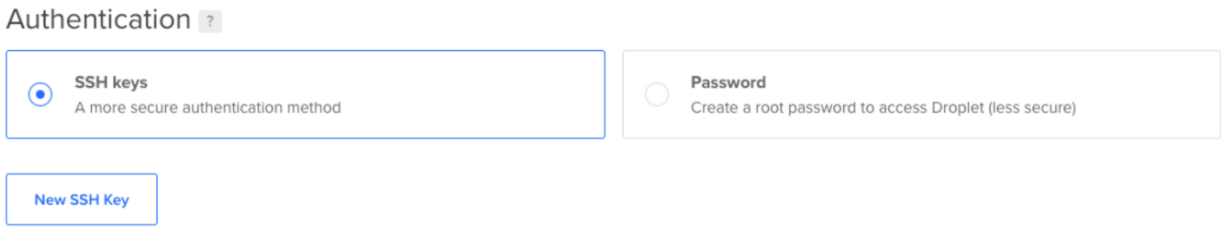


Рисунок 3.6 – Вибір способу аутентифікації

Вибір додаткових параметрів (рисунок 3.7). У розділі «Вибір додаткових параметрів» можна ввімкнути кілька додаткових служб, які додають функціональність до Droplet.

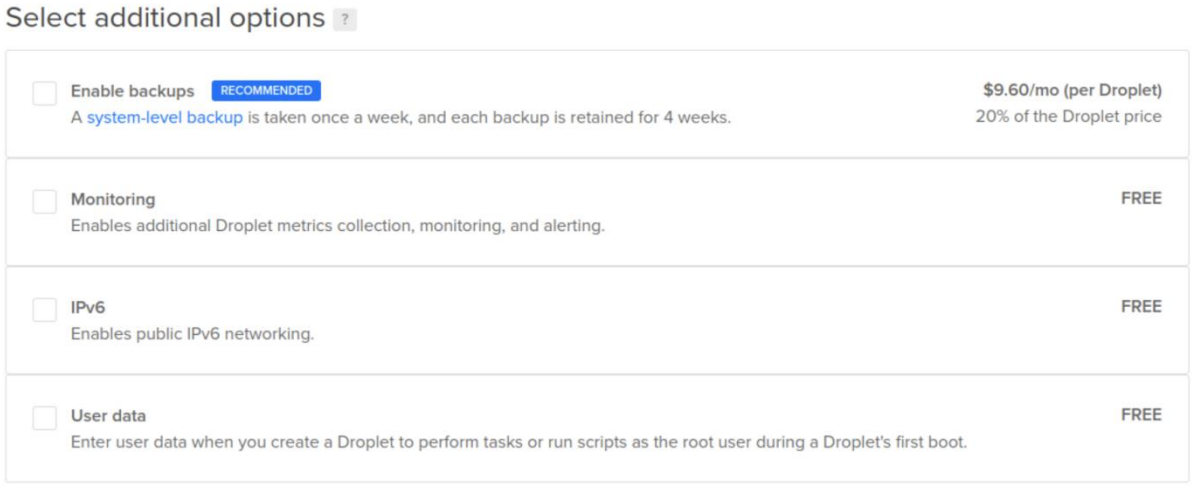


Рисунок 3.7 – Вибір додаткових параметрів

Більшість функцій, доступних у цьому розділі, є безкоштовними, а це означає, що їх увімкнення не збільшує щомісячну вартість Droplet.

Увімкнути резервне копіювання вмикає автоматичне щотижневе резервне копіювання Droplet. Це додає 20% до місячної ціни Droplet.

Якщо ви вирішите не вмикати резервне копіювання зараз, ви все одно зможете ввімкнути резервне копіювання для наявних крапель пізніше.

Моніторинг (безкоштовний) додає агент DigitalOcean для збору розширених показників і створення політики сповіщень.

IPv6 (безкоштовно) надає доступ до Ipv6 для створеного Droplet.

Дані користувача (безкоштовно) – це довільні дані, які ви вказуєте, які записуються в поле даних користувача служби метаданих DigitalOcean.
Droplet, що запускають дистрибутиви з cloud-init, можуть споживати та виконувати дані з цього поля, які зазвичай є файлами конфігурації хмари, які використовуються для початкового налаштування сервера під час першого завантаження.

Завершити та створити. У розділі «Завершити та створити» ви вказуєте кількість, назву, теги та проект для Droplet, яку ви створюєте (рисунок 3.8).

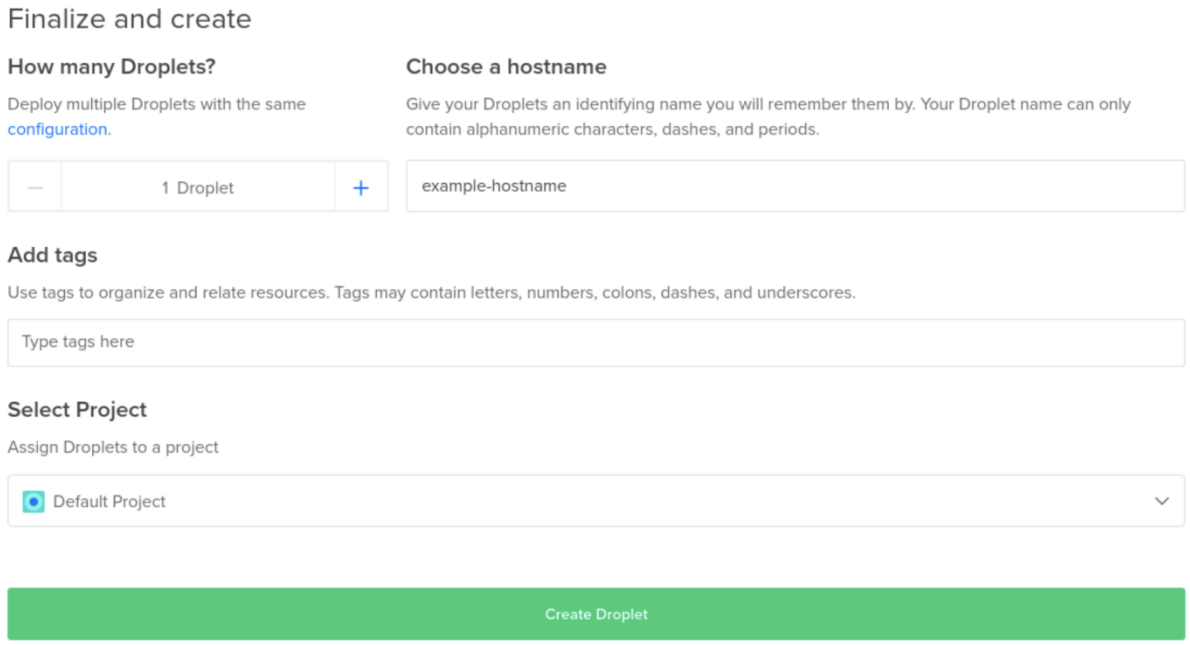


Рисунок 3.8 – Вигляд крана розділу «Завершити та створити».

Після повного налаштування Droplet на панелі керування відображається його IP-адреса (рисунок 3.9).

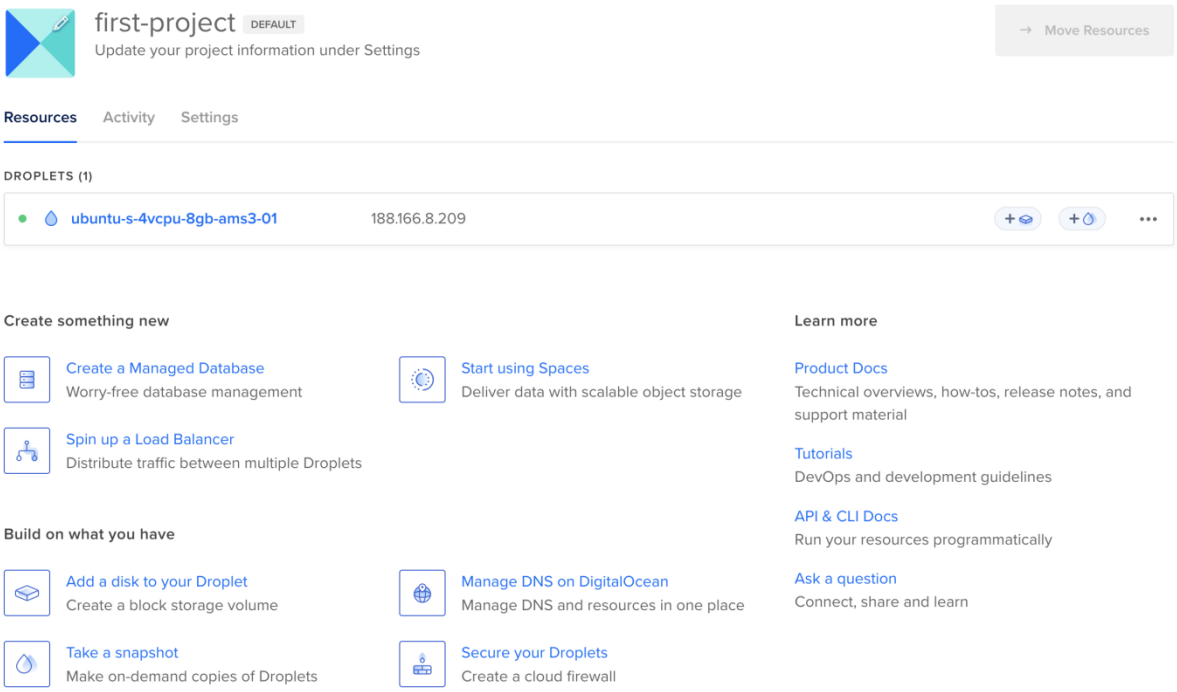


Рисунок 3.9 – Віртуальний сервер з операційною системою Ubuntu 18.04

Після того, як стала доступною IP-адреса, можна увійти до Droplet і продовжувати необхідні налаштування.

3.2 Встановлення приманки Cowrie

Спочатку необхідно встановити загальносистемну підтримку віртуальних середовищ Python та інших залежностей. Фактичні пакунки Python встановлюються пізніше.

```
sudo apt-get update
sudo apt-get install git python-virtualenv libssl-dev
libffi-dev build-essential libpython3-dev python3-
minimal authbind
```

```
Setting up libalgorithm-diff-xs-perl (0.04-5) ...
Setting up python3-virtualenv (15.1.0+ds-1.1) ...
Setting up libexpat1-dev:amd64 (2.2.5-3ubuntu0.7) ...
Setting up virtualenv (15.1.0+ds-1.1) ...
Setting up python (2.7.15~rc1-1) ...
Setting up binutils-x86-64-linux-gnu (2.30-21ubuntu1~18.04.7) ...
Setting up cpp (4:7.4.0-1ubuntu2.3) ...
Setting up python-pkg-resources (39.0.1-2) ...
Setting up libpython3.6-dev:amd64 (3.6.9-1~18.04ubuntu1.7) ...
Setting up python-virtualenv (15.1.0+ds-1.1) ...
Setting up binutils (2.30-21ubuntu1~18.04.7) ...
Setting up libpython3-dev:amd64 (3.6.7-1~18.04) ...
Setting up gcc-7 (7.5.0-3ubuntu1~18.04) ...
Setting up g++-7 (7.5.0-3ubuntu1~18.04) ...
Setting up gcc (4:7.4.0-1ubuntu2.3) ...
Setting up dpkg-dev (1.19.0.5ubuntu2.4) ...
Setting up g++ (4:7.4.0-1ubuntu2.3) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.4ubuntu1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.5) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for mime-support (3.60ubuntu1) ...
Processing triggers for install-info (6.5.0.dfsg.1-2) ...
root@ubuntu-s-4vcpu-8gb-ams3-01:~#
```

Додаємо користувача Cowrie і переходимо до нього. З метою захисту не можна запускати Cowrie з правами root:

```
sudo adduser --disabled-password cowrie
sudo su - cowrie
```

Завантажуємо код Cowrie:

```
git clone http://github.com/cowrie/cowrie
```

```
root@ubuntu-s-4vcpu-8gb-ams3-01:~# sudo su - cowrie
cowrie@ubuntu-s-4vcpu-8gb-ams3-01:~$ git clone http://github.com/cowrie/cowrie
Cloning into 'cowrie'...
warning: redirecting to https://github.com/cowrie/cowrie/
remote: Enumerating objects: 15793, done.
remote: Counting objects: 100% (112/112), done.
remote: Compressing objects: 100% (97/97), done.
remote: Total 15793 (delta 22), reused 88 (delta 14), pack-reused 15681
Receiving objects: 100% (15793/15793), 9.45 MiB | 11.90 MiB/s, done.
Resolving deltas: 100% (10945/10945), done.
cowrie@ubuntu-s-4vcpu-8gb-ams3-01:~$
```

Налаштуйте віртуальне середовище для Honeypot (підроблена ОС):

```
cd /home/cowrie/cowrie
virtualenv --python=python3 cowrie-env
source cowrie-env/bin/activate
(cowrie-env) $ pip install --upgrade pip
(cowrie-env) $ pip install --upgrade -r
requirements.txt
```

```
Requirement already satisfied: setuptools in ./cowrie-env/lib/python3.6/site-packages (from zope.interface>=4.
4.2->twisted==22.4.0->-r requirements.txt (line 14)) (59.6.0)
Building wheels for collected packages: tftpy
  Building wheel for tftpy (setup.py) ... done
  Created wheel for tftpy: filename=tftpy-0.8.2-py3-none-any.whl size=29512 sha256=70b98c4524a9141621ded799723
5b3fa56c52333946a1b9ad0b486cdd741e295
  Stored in directory: /home/cowrie/.cache/pip/wheels/b5/73/72/55aeaaa90ba95d6466da0b588021210abba3940cfc61cbb
0bf
Successfully built tftpy
Installing collected packages: pycparser, six, pyasn1, idna, cffi, attrs, zope.interface, typing-extensions, p
yasn1-modules, incremental, hyperlink, cryptography, constantly, Automat, urllib3, twisted, service-identity,
pyopenssl, charset-normalizer, certifi, requests, pyparsing, treq, tftpy, python-dateutil, packaging, configpa
rser, bcrypt, appdirs
Successfully installed Automat-20.2.0 appdirs-1.4.4 attrs-21.4.0 bcrypt-3.2.0 certifi-2022.5.18.1 cffi-1.15.0
charset-normalizer-2.0.12 configparser-5.2.0 constantly-15.1.0 cryptography-37.0.1 hyperlink-21.0.0 idna-3.3 i
ncremental-21.3.0 packaging-21.3 pyasn1-0.4.8 pyasn1-modules-0.2.8 pycparser-2.21 pyopenssl-22.0.0 pyparsing-3
.0.8 python-dateutil-2.8.2 requests-2.27.1 service-identity-21.1.0 six-1.16.0 tftpy-0.8.2 treq-22.2.0 twisted-
22.4.0 typing-extensions-4.1.1 urllib3-1.26.9 zope.interface-5.4.0
(cowrie-env) cowrie@ubuntu-s-4vcpu-8gb-ams3-01:~/cowrie$
```

Знаходимо файл cowrie.cfg.dist і копіюємо його в той самий файл, але з ім’ям cowrie.cfg, який необхідно відредагувати.

```
cp /home/cowrie/cowrie/etc/cowrie.cfg.dist /home/
cowrie/cowrie/etc/cowrie.cfg
```

Це файл конфігурації для Cowrie, якщо необхідно ввімкнути Telnet, змінити ім’я хоста сервера (щоб воно не виглядало як загальний Honeypot Cowrie), і так далі то зміни треба внести в даному файлі.

Наприклад, змінимо ім’я хоста: відредагуємо рядок ім’я хосту.. Більшість полів є зрозумілими, для зміни ім’я хоста рядок буде мати вигляд:

```
hostname = UbuntuServer5
```

Щоб Honeypot прослуховував порт 22 (за замовчуванням Honeypot слухає 2222, що пропустить багато речей), треба зробити наступні зміни:

У файлі cowrie.cfg:

```
listen_endpoints = tcp:22:interface=0.0.0.0
```

Cowrie потребує перезавантаження щоразу, після внесення змін у файл конфігурації.

Потім запускаємо наступні команди, щоб користувач, який не має права root, міг слухати порт 22 (заблокований за замовчуванням, і ми не можемо запустити Cowrie як root, тому це буде потрібно):

```
sudo apt-get install authbind
sudo touch /etc/authbind/byport/22
sudo chown cowrie:cowrie /etc/authbind/byport/22
sudo chmod 770 /etc/authbind/byport/22
```

Потім відредагуємо файл /etc/ssh/sshd_config, змінюємо рядок порту, щоб honeypot слухав справжній SSH-порт (не вибирайте 2222) і виконуємо перезапуск служби ssh:

```
nano /etc/ssh/sshd_config
service ssh restart
```

Тепер налаштуємо Cowrie під наглядом, щоб можна демонізувати його:

```
apt install supervisor
cat > /etc/supervisor/conf.d/cowrie.conf <<EOF
[program:cowrie]
command=/opt/cowrie/bin/cowrie start
directory=/opt/cowrie
stdout_logfile=/opt/cowrie/var/log/cowrie/cowrie.out
stderr_logfile=/opt/cowrie/var/log/cowrie/cowrie.err
autostart=true
autorestart=true
stopasgroup=true
killasgroup=true
user=cowrieEOF
supervisorctl update
```

```
root@ubuntu-s-4vcpu-8gb-ams3-01:~# apt install supervisor
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python-meld3
Suggested packages:
  supervisor-doc
The following NEW packages will be installed:
  python-meld3 supervisor
0 upgraded, 2 newly installed, 0 to remove and 43 not upgraded.
Need to get 287 kB of archives.
After this operation, 1580 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://mirrors.digitalocean.com/ubuntu bionic/universe amd64 python-meld3 all 1.0.2-2
[30.9 kB]
Get:2 http://mirrors.digitalocean.com/ubuntu bionic/universe amd64 supervisor all 3.3.1-1.1
[256 kB]
Fetched 287 kB in 0s (1034 kB/s)
Selecting previously unselected package python-meld3.
(Reading database ... 67627 files and directories currently installed.)
Preparing to unpack .../python-meld3_1.0.2-2_all.deb ...
Unpacking python-meld3 (1.0.2-2) ...
Selecting previously unselected package supervisor.
Preparing to unpack .../supervisor_3.3.1-1.1_all.deb ...
Unpacking supervisor (3.3.1-1.1) ...
Setting up python-meld3 (1.0.2-2) ...
Setting up supervisor (3.3.1-1.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/supervisor.service → /lib/system
d/system/supervisor.service.
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for systemd (237-3ubuntu10.53) ...
root@ubuntu-s-4vcpu-8gb-ams3-01:~#
```

Тепер перевіримо netstat, щоб переконатися, що кожен порт прослуховує правильний процес:

```
root@Cowrie:~# netstat -tanpl
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address
State PID/Program name
tcp 0 0 0.0.0.0:32328 0.0.0.0:* LISTEN 922/sshd
tcp 0 0 127.0.0.53:53 0.0.0.0:* LISTEN 639/systemd-
resolve
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 1007/python3
```

Інтеграція з VirusTotal:. Переходимо до VirusTotal і створюємо безкоштовний обліковий запис, для отримання ключа API.

Відкриваємо файл cowrie.cfg і розкоментовуємо розділ VirusTotal:

```
[output_virustotal]
enabled = true
api_key = ***paste here your API KEY***
upload = True
debug = False
scan_file = True
scan_url = True
```

Розгортання Splunk. Для розгортання Splunk нам знадобиться новий Droplet (рисунок 3.10).

Створюємо обліковий запис на Splunk і завантажуємо безкоштовну версію (безкоштовно до 500 МБ трафіку в день).

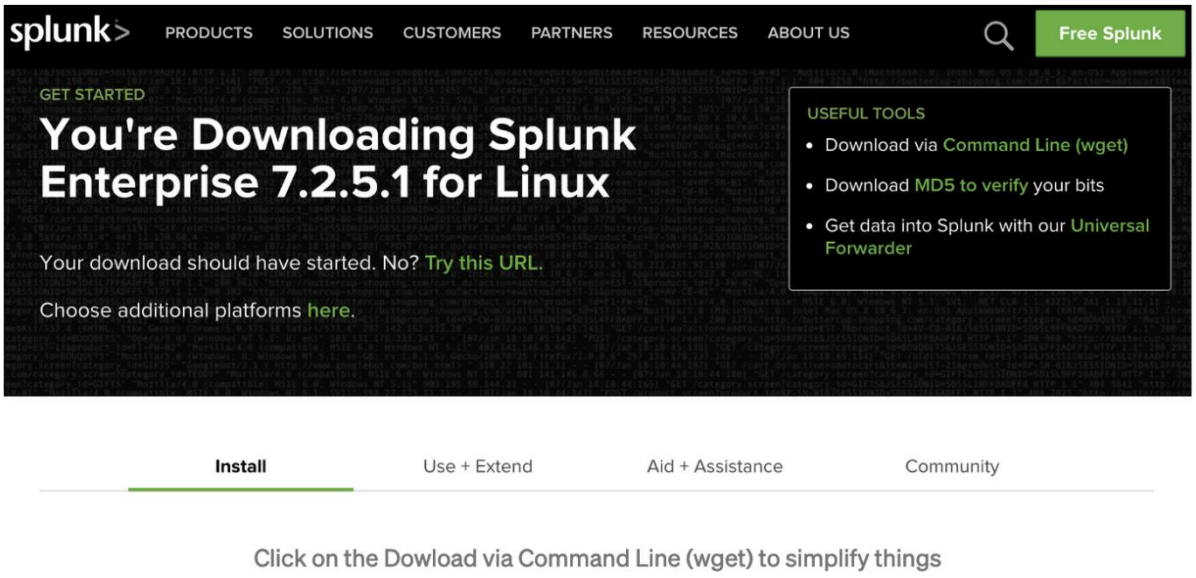


Рисунок 3.10 – Розгортання Splunk.

Завантажуємо версію Linux у форматі .tgz, і використовуємо опцію завантаження командного рядка для завантаження через wget:

```
cd /opt/
wget -O splunk-7.2..
tar -zxvf splunk-7.2...
cd /opt/splunk/bin/
./splunk start
```

Налаштовуємо user/pwd, а потім отримуємо доступ до графічного інтерфейсу, перейшовши на http://WhateverYourSplunkIpIs:8000

Підключіть Cowrie до Splunk. Щоб підключити Cowrie, перейдіть до Splunk, у верхньому меню і натисніть:

Settings> Add Data

Створюємо збірник подій HTTP, у розділі «Монітор», залишаємо все автоматично та копіюємо маркер (рисунок 3.11).

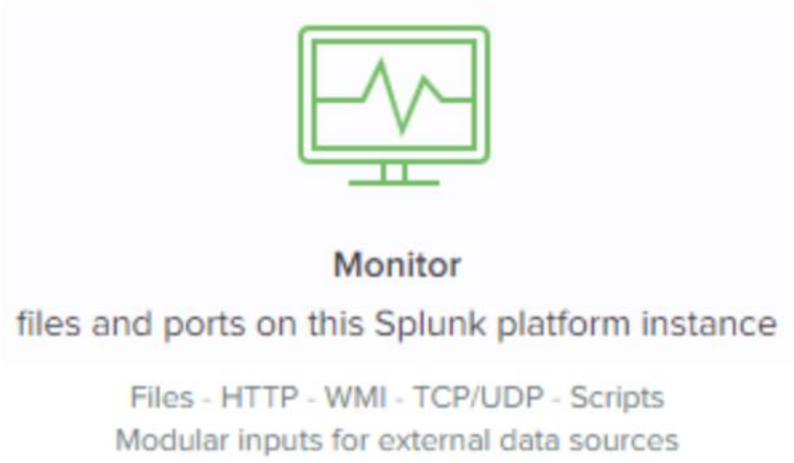


Рисунок 3.11 – Підключіть Cowrie до Splunk

У Honeypot Droplet переходимо до файлу cowrie.cfg і розкоментовуємо розділ output_splunk:

```
[output_splunk]
enabled = true
url = https://localhost:8088/services/collector/event
token = xxxxxxxxxxxxx
index = cowrie
sourcetype = cowrie
source = cowrie
```

Потім заповнюємо маркер і URL-адресу інформацією, яку отримали від Splunk.

URL є загальнодоступною IP-адресою другої Droplet, і ви отримуєте токен, розгортаючи новий збірник даних на Splunk:

Settings> Add Data > Monitor > HTTP Event Collector.

Залишаємо все автоматично і копіюємо токен, який отримали.

Після цього перезапускаємо Cowrie, і він має почати надсилати журнали.

Тепер Inside Splunk переходимо до Data>Indexes і створюємо новий індекс, просто змінюємо назву на «Cowrie».

Редагуємо розділ Data>Indexes вводимо збірник HTTP, щоб усе було надіслано до індексу «Cowrie».

Повертаємося до Data>Indexes і перевіряємо, чи Cowrie надсилає події, якщо Index “cowrie” містить деякі останні події, то налаштування всі правильні.

3.3 Тестування роботи та вивід результатів моніторингу подій

Для проведення тестування та виводу результатів роботи переходимо у верхній лівий кут програми Splunk і натискаємо «Програми»> «Керувати програмами».

Потім вибираємо «Встановити програму з файлу» та встановлюємо Manuka Honeypot:

Manuka Honeypot – це додаток Splunk, який створений спеціально для Cowrie.

Існує багато альтернативних програм (MHN, Tango, EngagedThreat…), але станом на 2022 рік усі вони здебільшого застаріли.

Залишаємо Honeypot на декілька годин і тоді побачимо перші результати.

Деякі з інформаційних панелей, приведені на рисунках (3.12 – 3.14).

Протягом 12-годинного періоду було видно очевидні докази того, що зловмисники намагалися ввести грубу силу в приманку Cowrie, зробивши понад 600 спроб підбору логіну користувача та пароля.

Подальший аналіз показав, що зловмисники намагалися завантажити «busybox», інструмент, який може надати зловмисникам оболонку для доступу та керування машиною, особливо пристроями IoT, які можуть не мати цієї функції.

Без належного захисту та оновлення мікропрограмного забезпечення такі пристрої, як розумні холодильники, можуть стати жертвами ботнету, надаючи зловмисникам можливість здійснювати DDoS-атаки.

Дані були відфільтровані за допомогою «input.keyword: exists», щоб побачити, які команди використовували зловмисники.

Команди показали, що зловмисники завантажують «ufo.apk» або «com.ufo.miner» (програму cyto-mining), встановлюють привілеї файлів і запускають програму.

Зловмисники також намагалися видалити завантаження, щоб, можливо, зберегти стабільність і уникнути виявлення. Багаторазове підрахунок може означати, що кілька зловмисників використовують одні й ті ж методи для виконання програми, що вказує на те, що атаки можуть бути частиною ботнету; і, як випливає з назви файлу, це атака криптоджекінга, яка перетворює систему як частину її ботнету для майнінгу криптовалют.

ADBhoney зміг захопити завантажені зразки .raw з пов’язаними з ним хешами SHA256. Використовуючи базу даних VirusTotal, завантажені файли (ufo.miner) були перевірені на наявність шкідливих і позначені як троянські Android Miner.

Рисунок 3.12 – Загальна інформація про під’єднання до honeypot

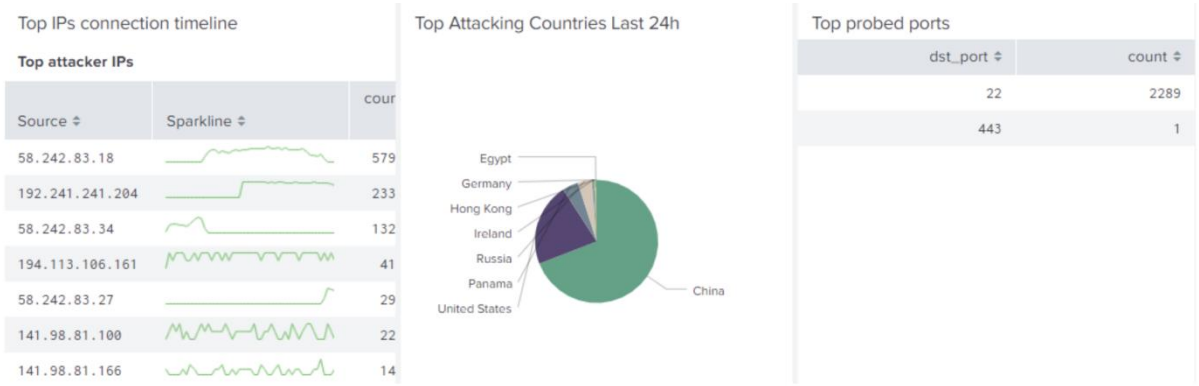


Рисунок 3.13 – Детальна інформація про події

Top User Password Combinations

Usr/Pwd Combination	Count
admin/admin	163
admin/1234	6
root/1q2w3e	5
admin/password	5
root/root	4
root/12345	4
admin/default	4
admin/changeme	4
admin/aerohive	4
admin/admin123	4

Top Usernames

username	count
root	2349
admin	214
sphinx	65
uftp	38
postgres	36
testuser	35
vps	32
ubuntu	20
ts	9
ftpuser	8

Top Passwords

password	count
admin	168
password	19
123456	17
test	12
1234	12
321	10
1q2w3e	10
123	10
qwe123	9
654321	9

Рисунок 3.14 – Імена і паролі, які використовувалися найчастіше

Cowrie SSH і Telnet приманка середньої взаємодії, яка призначена для реєстрації атак грубої сили та взаємодії оболонки, що виконується зловмисником. Cowrie також функціонує як проксі-сервер SSH і telnet, щоб спостерігати за поведінкою зловмисника в іншій системі.

Збір даних проводився протягом 12 годин. Дані були відфільтровані за допомогою «input.keyword: exists», щоб побачити, які команди використовували зловмисники.

Нижче наведено загальні команди, які використовуються зловмисниками:

Uname: роздрукувати системну інформацію;

Cat /proc/cpuinfo: збирає основну інформацію про процесор;

free -m: інформація у доступній пам'яті;

dmidecode: декодер таблиці DMI;

dmesg: друкувати або керувати кільцевим буфером ядра;

lspci: перелік усіх пристроїв pci;

ps: відобразити запущені процеси в оболонці.

Можлива причина, по якій зловмисники можуть захотіти знати про простір пам’яті системи, процесор, графічний процесор, pscі та її процес завантаження/запуску, полягає в тому, щоб визначити, чи була б це ідеальна машина, щоб бути частиною її ботнету (можливо, для DDoS або крипто-майнінгу).

3.4 Висновки

В процесі виконання цього розділу було проведено налаштування віртуального сервера на базі OS Ubuntu 18.04. Для розгортання приманки використаємо хмарний сервіс DigitalOcean. Встановлено програмне забезпечення приманки Cowrie та проведене всі необхідні налаштування.

Проведено підключення та налаштування Splunk до приманки Cowrie.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи проведено аналіз існуючого стану та тенденції розвитку Інтернету речей. Розкрито принципи функціонування приманок. Проведено аналіз класичних honeypots. Обґрунтовано вибір апаратних ресурсів та мови програмування.

Розкрито основні уразливості безпеки Інтернету речей, який ґрунтується на Top 10 спільноти OWASP. Усвідомлення основних вразливостей безпеки Інтернет-речей може допомогти впевнитися, що пристрої та мережі не стануть жертвами шкідливої кібератаки.

Розроблена класифікація honeypot, яка враховує їх призначення, ролі, рівні взаємодії, масштабованість, рівні ресурсів, доступність вихідного коду та застосування.

Досліджено найбільш поширені типи атак на Інтернет- речей., зокрема, фізичні атаки, атаки шифрування, DoS (відмова в обслуговуванні), викрадання прошивки, ботнети, людина посередині, програми-вимагачі, підслуховування, підвищення привілеїв, атака грубою силою.

В третьому розділі було проведено налаштування віртуального сервера на базі OS Ubuntu 18.04. Для розгортання приманки використано хмарний сервіс DigitalOcean.

Встановлено програмне забезпечення приманки Cowrie та проведене всі необхідні налаштування. Здійснено підключення та налаштування Splunk до приманки Cowrie.

Протягом періоду спостереження зловмисники намагалися використати грубу силу в приманці Cowrie, зробивши понад 600 спроб підбору паролю користувача. Зловмисники також намагалися завантажити «busybox», інструмент, який може надати зловмисникам оболонку для доступу та керування машиною, особливо пристроями IoT, які можуть не мати цієї функції.

З більшою кількістю технологій та пристроїв, які мають доступ до Інтернету, вони стають вразливими пристроями IoT, якими можуть скористатися

зловмисники. Без належного захисту та оновлення мікропрограмного забезпечення такі пристрої, як розумні холодильники, можуть стати жертвами ботнету, надаючи зловмисникам можливість здійснювати DDoS-атаки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. The Future of IoT: Top Trends We Expect to See in 2022. https://www.iotworldtoday.com/2022/01/10/the-future-of-iot-top-trends-we-expect-to-see-in-2022/

2. Top 6 IoT trends in 2022. https://www.analyticsinsight.net/top-6-iot-trends-in-2022/

3. The 5 Biggest Internet Of Things (IoT) Trends In 2022. https://www.forbes.com/sites/bernardmarr/2021/12/13/the-5-biggest-internet-of-things-iot-trends-in-2022/?sh=50122b975aba

4. Internet of Things statistics for 2022 - Taking Things Apart https://dataprot.net/statistics/iot-statistics/

5. 15 Michel Oosterhof. Cowrie Documentation. Release 2.3.0 https://readthedocs.org/projects/cowrie/downloads/pdf/latest/

6. DigitalOcean. Product Docs Home. https://cloud.digitalocean.com

7. Justin Ellingwood. SSH Essentials: Working with SSH Servers, Clients, and Keys. https://www.digitalocean.com/community/tutorials/ssh-essentials-working-

8. Zhang, L.; Ding, G.; Wu, Q.; Zou, Y.; Han, Z.; Wang, J. Byzantine Attack and Defense in Cognitive Radio Networks: A Survey. *IEEE Commun. Surv. Tutor.* 2015, *17*, 1342–1363.

9. Sullivan, J.E.; Kamensky, D. How cyber-attacks in Ukraine show the vulnerability of the U.S. power grid. *Electr. J.* 2017, *30*, 30–35.

10. Zhou, W.; Jia, Y.; Peng, A.; Zhang, Y.; Liu, P. The Effect of IoT New Features on Security and Privacy: New Threats, Existing

11. Solutions, and Challenges Yet to Be Solved. *IEEE Internet Things J.* 2019, *6*, 1606–1616.

12. Giraldo, J.; Sarkar, E.; Cardenas, A.A.; Maniatakos, M.; Kantarcioglu, M. Security and Privacy in Cyber-Physical Systems: A

13. Survey of Surveys. *IEEE Des. Test* 2017, *34*, 7–17.

14. Wood, A.D.; Stankovic, J.A. Denial of service in sensor networks. *Computer* 2002, *35*, 54–62.

15. Raiyn, J. A survey of cyber attack detection strategies. *Int. J. Secur. Appl.* 2014, *8*, 247–256

16. Stiawan, D.; Idris, M.Y.; Malik, R.F.; Nurmaini, S.; Alsharif, N.; Budiarto, R. Investigating Brute Force Attack Patterns in IoT Network. *J. Electr. Comput. Eng.* 2019, *2019*, 1–13.

17. "How big is iot? 20.6 billion connected devices in 2020," Apr 2021.

18. Ziaie Tabari and X. Ou, "A first step towards understanding real-world attacks on iot devices," arXiv preprint arXiv:2003.01218, 2020.

19. T. Luo, Z. Xu, X. Jin, Y. Jia, and X. Ouyang, "Iotcandyjar: Towards an intelligent-interaction honeypot for iot devices," Black Hat, pp. 1–11, 2017.

20. Vetterl and R. Clayton, "Honware: A virtual honeypot framework for capturing cpe and iot zero days," in 2019 APWG Symposium on Electronic Crime Research (eCrime), pp. 1–13, IEEE, 2019.

21. J. Cao, W. Li, J. Li, and B. Li, "Dipot: A distributed industrial honeypot system," in International Conference on Smart Computing and Communication, pp. 300–309, Springer, 2017

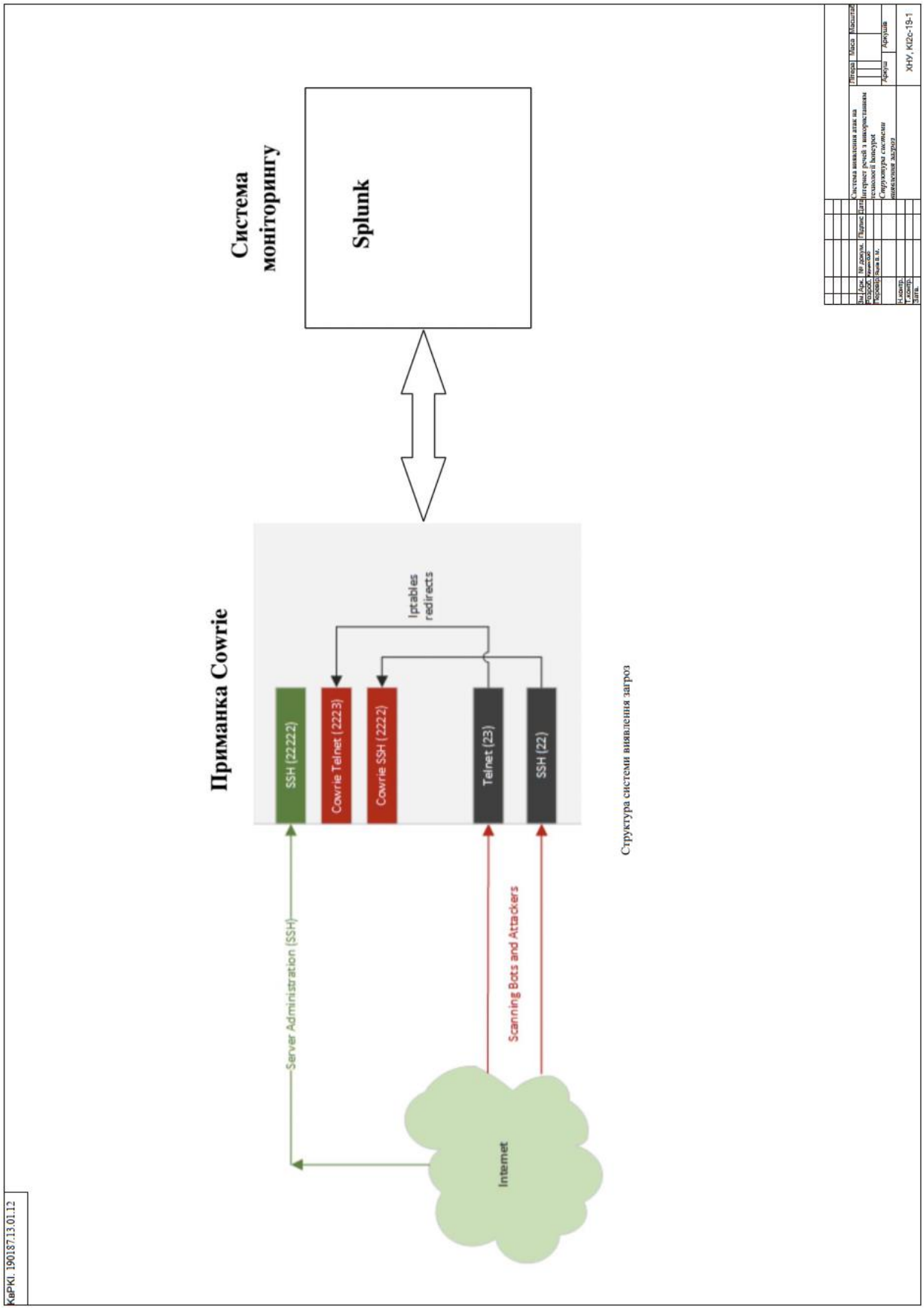
22. P. Ferretti, M. Pogliani, and S. Zanero, "Characterizing background noise in ics traffic through a set of low interaction honeypots," in Proceedings of the ACM Workshop on Cyber-Physical Systems Security & Privacy, pp. 51–61, 2019.

23. S. Lau, J. Klick, S. Arndt, and V. Roth, "Poster: Towards highly interactive honeypots for in- dustrial control systems," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1823–1825, 2016.

Додаток А

(обов’язковий)

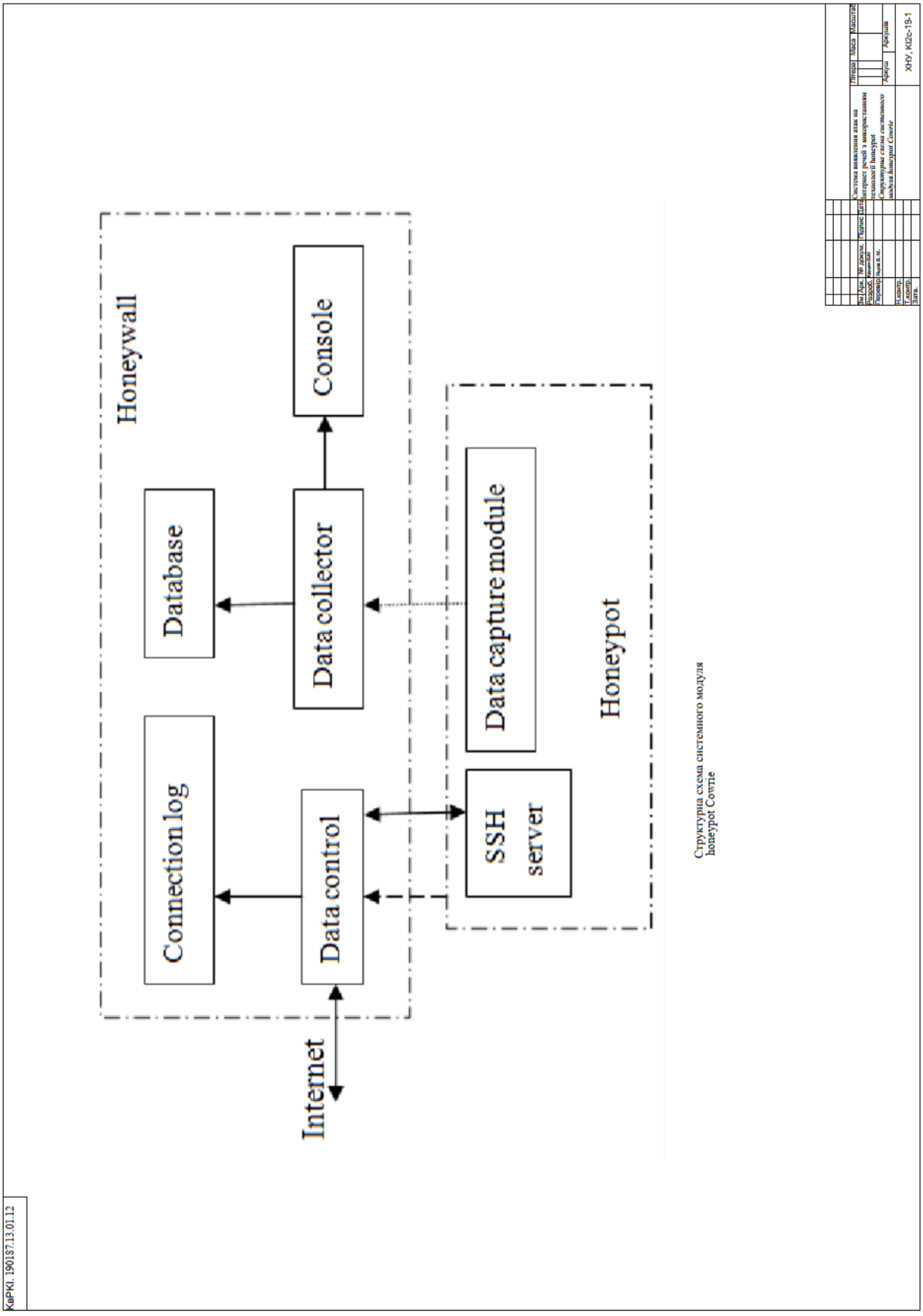
Копія креслення «Структура системи виявлення загроз»



Додаток В

(обов’язковий)

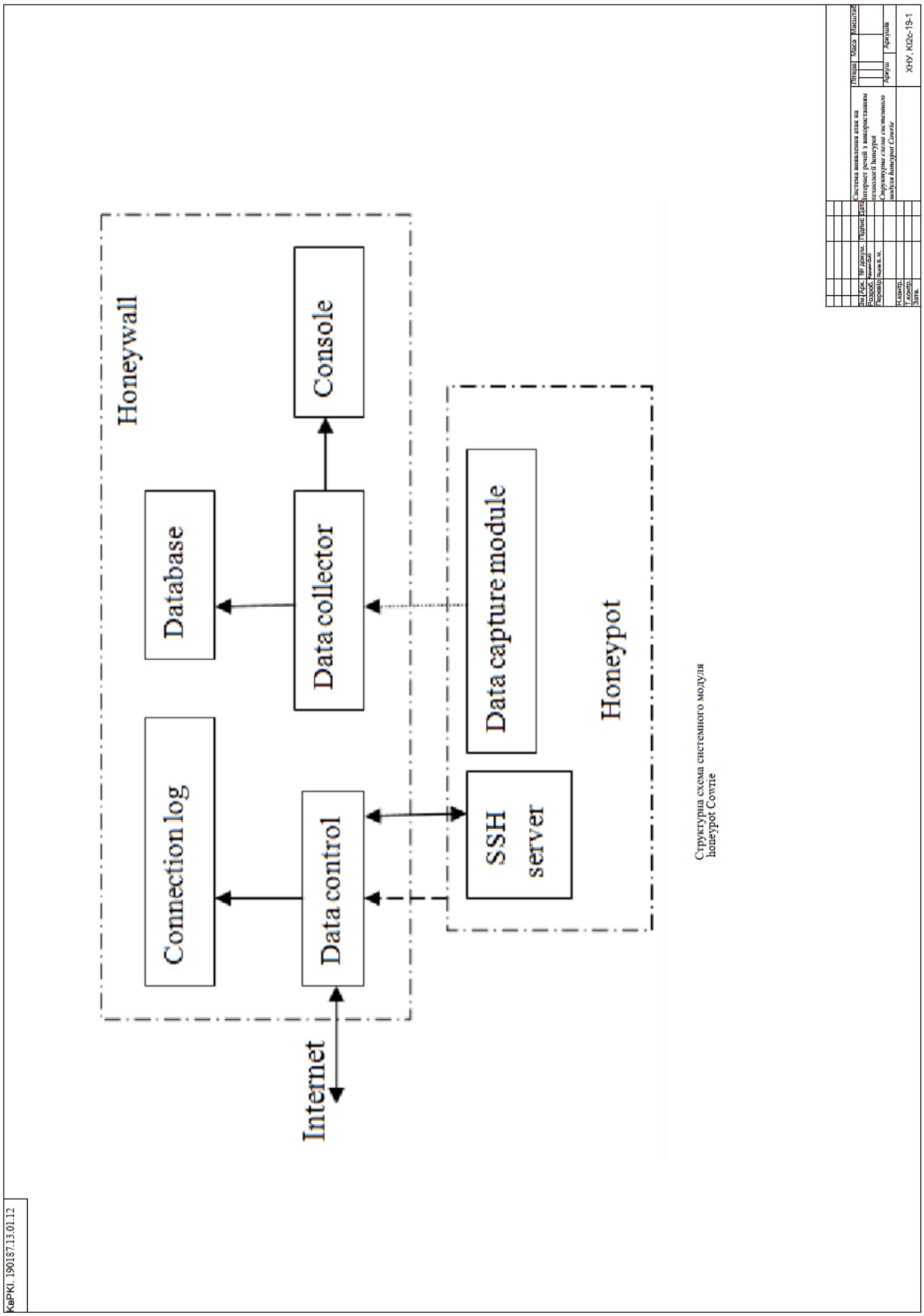
Копія креслення «Структурна схема системного модуля honeypot Cowrie»



Додаток С

(обов’язковий)

Копія креслення «Діаграма HoneyPort»



UNICHECK

Ім'я користувача:
Кафедра КІ

ID перевірки:
1011588664

Дата перевірки:
15.06.2022 18:03:01 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
15.06.2022 18:04:51 EEST

ID користувача:
100005591

Назва документа: **Круцюк_Система виявлення атак на Інтернет речей з використанням технології honeypot**

Кількість сторінок: **59** Кількість слів: **9721** Кількість символів: **77032** Розмір файлу: **4.70 MB** ID файлу: **1011457824**

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

2.05%
Схожість

Найбільша схожість: **0.96%** з джерелом з Бібліотеки (ID файлу: **1011387906**)

1.19% Джерела з інтернету	18	Сторінка 61
1.38% Джерела з Бібліотеки	94	Сторінка 61

0.73% Цитат

Цитати	2	Сторінка 62

Не знайдено жодних посилань

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	1
Підозріле форматування	12 сторінок

Wed Jun 15 17:03:37 EEST 2022, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 1.0%

Словари проверки: en_US, ru_RU, ua_UA.

Ошибок в документах: 11%

ID: 105596 Название: Система виявлення атак на Інтернет речей з використанням технології honeypot Добавлено в БД: 2022-06-15 Авторы: О. Ю. Круцюк Руководители: В.М. Яцків Консультанты: Оппоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	69437	534	711 (1%)	7 (1%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Круцюк Олександр Юрійович

Тема: Система виявлення атак на Інтернет речей з використанням технології honeypot

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 63

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є система виявлення атак на Інтернет речей з використанням технології honeypot

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі кваліфікаційної роботи проведено дослідження предметної області (проаналізовано Аналіз існуючого стану та тенденції розвитку Інтернету речей, а також Аналіз класичних honeypots) та виконано постановку задачі дослідження. В другому розділі кваліфікаційної роботи проведено моделювання та проєктування структури honeypots, а саме: виконано формалізований опис структури ; розроблено змістовну схему; переглянули вразливості безпеки інтернет речей , типи атак на інтернет.

Зробили налаштування віртуального сервера. Встановили приманку Cowrie, Тестували роботу та вивід результатів моніторингу подій

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: під час роботи не виникало негативних сторін.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Гурман Іван Васильович доцент кафедри інженерії програмного забезпечення

"16" червня 2022 р. _____ (підпис)

Завідувачу кафедри кібербезпеки
д-ру техн.наук, проф.. Говорущенко Т.О.
Круцюка Олександра Юрійовича
ПІБ здобувача вищої освіти

студента ФІТ, 3 курсу, групи КІ2с-19-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

6.06.2022
дата

підпис

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Синтез та моделювання операційного автомату на основі автомату Мура

Автор: Круцюк Олександр Юрійович

Спеціальність: 123 – Компютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Яцків Василь Васильович, д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
2) усі запозичення мають належним чином оформленні посилання;
3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 2.05% і адресується до 102 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ В.В Яцків

Гарант ОП _____ С. М. Лисенко

Завідувач кафедри КІС _____ Т. О. Говорущенко