**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**

**ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Фізико-технічний інститут

**КУРСОВА РОБОТА**

з дисципліни: «Безпека операційних систем та комп’ютерних мереж»

на тему: «Мережева пастка honeypot HoneyD»

**Виконав:**

Студент 4 курсу ФТІ

групи ФБ-62

Бублик О.О.

**Перевірив:**

Грайворонський М.В.

Київ

2019

ЗМІСТ

ВСТУП 3

РОЗДІЛ 1 Класифікація Honeypot 4

1.1 Тип взаємодії ..4

1.2 Тип системи ....5

1.3 Розміщення honeypot 6

1.4 Призначення ..7

1.5 Тип конфігурації ....7

1.6 Висновки ....8

РОЗДІЛ 2. Існуючі реалізації 9

2.1 Низько інтерактивні honeypot 9

2.1.1 Deception toolkit 9

2.1.2 LaBrea. 9

2.1.3 Dionaea 9

2.1.4 HoneyD 10

2.2 Високо інтерактивні honeypot 11

2.2.1 Honeywall. 11

2.2.2 Sebek 12

2.2.3 Dockpot 12

2.2.4 T-Pot 13

2.3 Середньо інтерактивні 13

2.4 Висновки 13

РОЗДІЛ 3. ПРИКЛАД ЗАСТОСУВАННЯ HONEYD ……………………......14

3.1 Висновки 21

ВИСНОВКИ 22

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ …………………………. …… 23

**ВСТУП**

**Актуальність роботи**. Поряд з традиційними методами захисту корпоративних мереж від вторгнень, такими як системи виявлення вторгнень (IDS) і системи запобігання вторгнень (IPS), існують і альтернативні підходи для забезпечення безпеки інформаційних ресурсів. Одним з таких підходів є використання технологій honeypot.

**Основною метою** курсової роботи є розгляд, практичне застосування і вивчення ефективної реалізації honeypot-систем.

**Завдання курсової роботи:**

1. Розгляд принципів функціонування honeypot-систем;

2. Огляд, характеристика honeypot системи HoneyD;

3. Розгляд конфігураційних файлів та порядка налаштування honeypot-систем;

4. Виконання практичного завдання з метою застосування HoneyD;

**Об'єкт курсової роботи:** honeypot система HoneyD.

**Предмет даної роботи:** Визначити та ініціювати події.

**Honeypot («Пастка»)** — ресурс, що є приманкою для зловмисників.

Мета Honeypot — зазнати атаки або несанкціонованого дослідження, що згодом дозволить вивчити стратегію зловмисника та визначити перелік засобів, за допомогою яких можуть бути завдані удари реально наявним об'єктам безпеки. Реалізація Honeypot може бути спеціальним виділеним сервером, або мережевим сервісом, завдання якого — привернути увагу зловмисника.

**РОЗДІЛ 1**

**КЛАСИФІКАЦІЯ HONEYPOT**

1.1 Тип взаємодії

Технологія honeypot має на увазі надання зловмисникові доступу до свідомо уразливого ресурсу. Після того як зловмисник отримає доступ, він буде робити ті чи інші дії, очікуючи відповідної реакції від системи. Залежно від того, як буде проводитися взаємодія системи і зловмисника, можна виділити два різних типи систем honeypot :

* Високо інтерактивні системи. Повністю імітують поведінку реальної системи.
* Низько інтерактивні системи. Мають обмежений функціонал.

Високо інтерактивні системи найчастіше являють собою окремий хост або пристрій, розташований всередині корпоративної мережі, але не використовується у інформаційних процесах. Таким чином, звернення ззовні до подібного ресурсу можна розглядати як порушення безпеки мережевої інфраструктури. Так як такі системи надають зловмисникові повний функціонал реальних систем, то вони можуть також бути скомпрометовані і використані в якості допоміжних засобів для проникнення в реальну мережу і інших цілей, наприклад, формування ботнетів. Тому необхідно забезпечувати ізоляцію подібних систем.

Такі системи зазвичай використовуються для отримання різної інформації про типи атак, мети порушника і можливі вразливості в корпоративній інфраструктурі.

Низько інтерактивні системи використовують різні механізми для імітації поведінки реальної системи. Це може бути, як операційна система, так і веб-сервер, сервер баз даних або навіть звичайний мережевий додаток.

Такі системи на відміну від високо інтерактивних є менш ресурсоємними, простіше в налаштуванні і не можуть бути використані порушником для здійснення атаки на реальні ресурси корпоративної мережі.

З іншого боку, вони легко виявляються порушником, через що такі системи більше підходять для протидії різним автоматизованим засобам, що використовуються порушниками, ніж для отримання будь-яких відомостей про них

Наведемо основні переваги та недоліки кожного типу у вигляді таблиці (Таблиця 1)

Таблиця 1 – Порівняння типів систем

|  |  |  |
| --- | --- | --- |
|  | Низько інтерактивні | Високо інтерактивні |
| Переваги | * Простота налаштування * Безпека | * Повний функціонал * Важкість виявлення |
| Недоліки | * Легкість виявлення * Мало корисної інформації | * Важкість розгортання і налаштування * Загроза взлому |

Також існують системи, які складно віднести до того чи іншого класу. З одного боку, вони надають порушнику повний функціонал того чи іншого сервісу, а з іншого не виробляють ніяких «реальних» дій. Такі системи називають середньо інтерактивними.

1.2 Тип системи

Системи honeypot можуть працювати як на реальних, так і на віртуальних

машинах. Залежно від цього виділяють фізичні та віртуальні системи honeypot.

Фізичні системи теоретично забезпечують більший рівень ізоляції системи від решти мережі, але при цьому вимагають набагато більших ресурсів для роботи.

Віртуальні ж системи вимагають менше ресурсів, а також більш зручні в налаштуванні і обслуговуванні.

Для реалізації подібних систем застосовуються добре відомі рішення в області віртуалізації, такі як VMWare, VirtualBox та інші. Також можуть використовуватися технології віртуалізації рівня ОС, наприклад, LXC, або специфічні рішення, такі як User Mode Linux. Для створення віртуального honeypot, що імітує роботу різного мережевого обладнання, на базі x86-систем можуть бути використані емулятори, наприклад, QEMU.

Віртуальні системи дозволяють в рамках однієї фізичної машини створювати кілька honeypot-систем. Така конфігурація отримала назву honeynet.

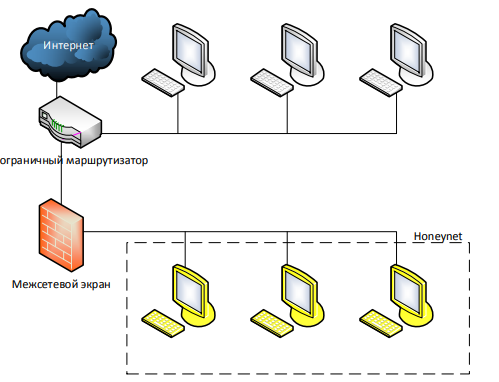


Рисунок 1 – Honeynet

Вона дозволяє розгорнути цілу мережу неправдивих комп'ютерів і дати

зловмисникові великий простір для дій, що допоможе нам отримати

більше цінної інформації. Приклад конфігурації honeynet наведено на

рисунку 1.

1.3 Розміщення honeypot

Залежно від призначення системи honeypot можуть розташовуватися:

* У локальній мережі. Такі системи збирають інформацію про атаки на

внутрішньомережеві ресурси.

* У DMZ. В такому випадку, система буде збирати інформацію про атаки

на загальнодоступні ресурси, такі як, наприклад, веб-сервер.

1.4 Призначення

Склад і функціональність систем honeypot можуть сильно змінюватися

залежно від переслідуваних цілей їх створення. Традиційно виділяють дві основні цілі існування подібних систем.

Захист корпоративної мережі - honeypot допомагає адміністраторам

організації підтримувати безпеку мережевої інфраструктури, а також бути в курсі нових методів проникнення і злому систем, використовуваних в організації. В такому випадку, honeypot (honeynet) повинен максимально правдоподібно відображати конфігурацію реальної системи.

Дослідження - honeypot допомагає виявляти нові методи проникнення в різні інформаційні системи, а також збирати статистику використання відомих методів, а також отримувати іншу інформацію, пов'язану з кіберзлочинцями і їх діяльністю. Такі системи повинні надавати порушникам максимально можливий набір потенційно уразливих ресурсів.

1.5 Тип конфігурації

Налаштування honeypot, особливо в разі високо інтерактивних систем, є складним завданням. Якщо ж подібні системи об’єднані в мережу (Honeynet), то складність конфігурації всієї системи пропорційна числу хостів, що входять до її складу.

Вручну налаштовувані системи називаються статичними. Їх конфігурація визначається адміністратором і змінюється тільки по його команді. Системи, які автоматично підлаштовуються під зміни конфігурації називаються динамічними. Загалом, до динамічних honeypot-систем можна віднести всі системи, конфігурація яких змінюється без безпосереднього втручання адміністратора.

1.6 Висновки

В даний час найбільший інтерес представляють високо інтерактивні динамічні honeypot-системи, так як з завданнями низько інтерактивних систем справляються інші елементи корпоративної мережі: мережевими скануваннями успішно запобігають міжмережевими екранами і системами виявлення та запобігання вторгнень, а також правильної конфігурацією елементів мережі, а інформація про конкретні дії зловмисника, зроблених для здійснення доступу в корпоративну мережу, надають набагато більшу цінність, ніж інформація про факт проникнення в мережу, яка також може бути отримана від міжмережевих екранів і засобів виявлення вторгнень.

Динамічні ж honeypot-системи мають великий потенціал розвитку. У таких системах можуть бути застосовані ідеї,які зарекомендували себе в інших типах систем, а також запропоновані і принципово нові підходи.

**РОЗДІЛ 2**

**ІСНУЮЧІ РЕАЛІЗАЦІЇ**

2.1 Низькоі нтерактивні honeypot

2.1.1 Deception toolkit

Deception toolkit - один з перший низько інтерактивних honeypot. Вся його функціональність полягала в тому, що відкривалися невикористовувані мережеві порти і при спробі підключення до них видавалося попередження, яке в теорії повинно було відлякати зловмисника.

2.1.2 LaBrea

Утиліта LaBrea - призначена для боротьби зі спамом, а також мережними хробаками. Система отримує список невикористаних IP-адрес внутрішньої мережі і починає приймати з'єднання на ці адреси. Використовуючи особливості протоколу TCP, система «гальмує» з'єднання. Вона може як уповільнити його, використовуючи маленькі значення Window Size, так і «Заморозити» його, встановити значення Window Size в 0, що призведе до того, що з'єднаний клієнт буде очікувати зміни цього параметра.

2.1.3 Dionaea

Dionaea являє собою модульну систему, емулює різні протоколи, такі як:

* Server Message Block (SMB) - основний протокол, підтримуваний Dionaea.

SMB має безліч відомих вразливостей і тому часто є метою зловмисників.

* Hypertext Transfer Protocol (HTTP) - Dionaea також підтримує і протокол

HTTPS, використовуючи самопідписаний сертифікат для встановлення надійного з'єднання.

* File Transfer Protocol (FTP) - Dionaea має можливість роботи як FTP-сервер, що підтримує створення каталогів і завантаження файлів.
* Trivial File Transfer Protocol (TFTP);
* Microsoft SQL Server (MSSQL) - дозволяє проводити підключення до БД і фіксує як виконуються запити.
* Voice over IP (VoIP) - Dionaea підтримує протокол Session Initial Protocol (SIP). Отримує запити, записує їх і відповідає клієнтові.

2.1.4 HoneyD

Напевно, найбільш цікавим рішенням серед низько інтерактивних honeypot є HoneyD. Ця система дозволяє створити емульовану мережу, що складається з безлічі комп'ютерів, з різними операційними системами і запущеними мережними додатками. Можлива конфігурація HoneyD зображена на рисунку 2.

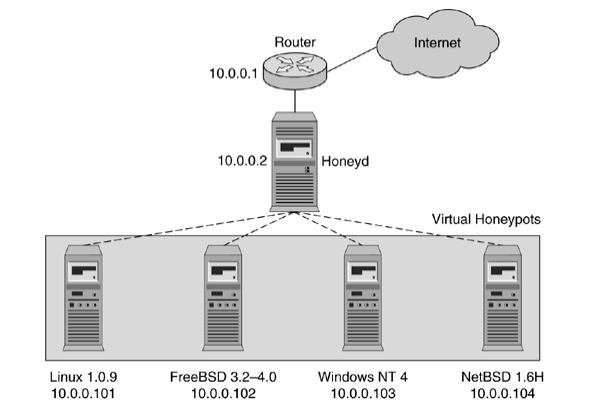


Рисунок 2 – HoneyD

До найбільш цікавих особливостей системи можна віднести:

* Можливість одночасної роботи безлічі емульованих хостів;
* Можливість налаштування програм, що запускаються при підключенні зловмисника до хосту;
* Симуляція TCP / IP стека різних ОС;
* Симуляція роботи мережевих пристроїв;
* Можливість запуску реальних додатків під час роботи.

Таким чином, за допомогою HoneyD можна побудувати honeynet, що складається з різних ОС з різними відкритими програмами.

2.2 Високо інтерактивні honeypot

Так як високо інтерактивні honeypot по суті є реальними системами, запущеними в певному середовищі, має сенс розглядати саме кошти, які допомагають забезпечити необхідні властивості цього середовища, такі як ізоляція і детальний і прихований моніторинг.

2.2.1 Honeywall

Honeywall є комплексом засобів, що здійснюють ізоляцію і моніторинг всередині системи honeynet. Архітектура honeypot системи, яка використовує honeywall зображена на рисунку 3.

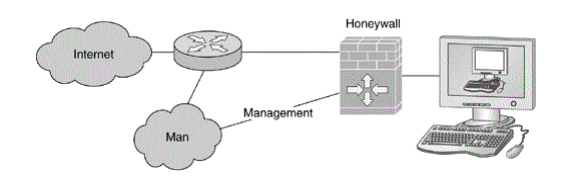


Рисунок 3 – Honeywall

Він виконує чотири основні функції:

Запис мережевого трафіку - весь трафік між зловмисником і honeynet записується для подальшого аналізу. Для цього використовуються кошти tcpdump, Wireshark та ін .;

Контроль мережевого трафіку - трафік повинен ретельно перевірятися на предмет підозрілої активності. У разі її виявлення повинні вживатися заходи щодо ізоляції honeynet від зовнішньої мережі, або по модифікації трафіку, таким чином, що він не заподіє шкоди зовнішнім системам - так звана, «стерилізація» атаки. Для цього можуть використовуватися засоби типу snort\_inline;

Аналіз мережевого трафіку - для зручності адміністратора надають різні засоби аналізу трафіку.

Запис активності користувача. Часто великий інтерес представляє саме локальна активність зловмисника - команди терміналу, запускаємі програми. Для запису такої активності використовуються утиліти типу Sebek, про яку мова піде пізніше.

2.2.2 Sebek

Sebek являє собою модуль ядра, що працює в прихованому для користувача вигляді і виконує запис всіх його дій. Sebek має клієнт-серверну архітектуру. Клієнт працює на самій машині, виконує роль honeypot. Він перехоплює системні виклики, такі як:

* read () / readv () / pread () - читання даних;
* open () - відкриття файлів;
* socketcall () - мережеві операції;
* fork () / vfork (), clone () - запуск процесів;

Контроль над цими системними викликами дозволяє фіксувати майже всі дії користувача, які потім відправляються на сервер, який обробляє і зберігає ці дані.

2.2.3 Dockpot

Dockpot представляє собою SSH-проксі, що з'єднує зловмисника з системою honeypot, реалізованої з використанням контейнерів Linux, а саме Docker. Для кожного з'єднання створюється новий контейнер, а при закритті з'єднань контейнери знищуються. Такий підхід дозволяє створювати так звану «ферму» honeypot систем, яка динамічно змінює свій розмір в залежності від числа підключень і тим самим значно економить обчислювальні ресурси, необхідні для підтримки її функціонування.

Використання SSH-проксі дозволяє розшифровувати зашифровані дані, що передаються зловмисником в систему. Використовується підхід, аналогічний атаці MITM - Dockpot встановлює два з'єднання: одне з системою, а інша зі зловмисником. Для зловмисника ж це з'єднання прозоре - він думає, що з'єднаний безпосередньо з системою.

2.2.4 T-Pot

T-Pot використовує контейнери Docker трохи інакше. Продукт має модульну структуру і складається з набору різних контейнерів, які містять засоби, що використовуються при побудові honeypot систем: glastopf, kippo, honeytrap, dionaea, а також IDS / IPS системи: suricata, elasticsearchlogstash-kibana, ewsposter. Всі ці засоби запускаються в ізольованих контейнерах, що робить можливим їх одночасну роботу, а також спрощує процес розгортання і налаштування.

2.3 Середньо інтерактивне honeypot

Прикладом середньоінтерактивної системи може послужити Kippo -

емулятор SSH. Він має наступні функції:

Імітація роботи багатьох функцій терміналу (shell);

Імітація файлової системи з можливістю створення і видалення файлів;

Можливість імітації різних даних, доступних порушнику;

Збереження завантажених порушником файлів.

2.4 Висновки

Більшість описаних систем добре зарекомендували себе, але при цьому розроб лені досить давно і не використовують ніяких нових технологій. Ідеї, використані в таких системах як Honeyd, можуть бути застосовані при побудові нових систем, що використовують всі можливості, що надаються сучасними технологіями.

**РОЗДІЛ 3**

**ПРИКЛАД ЗАСТОСУВАННЯ HONEYD**

Завдання на практичну реалізацію:

1. Описати порядок інсталювання обраної Вами мережевої пастки.

2. Визначити призначення основних конфігураційних файлів HoneyD.

3. Визначити та ініціювати події (Telnet, SSH, Nmap) в HoneyD

4. Відобразити та пояснити отримані Вами в файлах журналювання записи відносно ініційованих подій.

В цьому прикладі будемо використовувати ОС Ubuntu 12.4 (HoneyD)

а також ОС Kali Linux (Для ініціювання подій).

Реалізація:

Ми проводимо установку демона HoneyD використовуючи команду ( процес зображено на рисунку 1.)

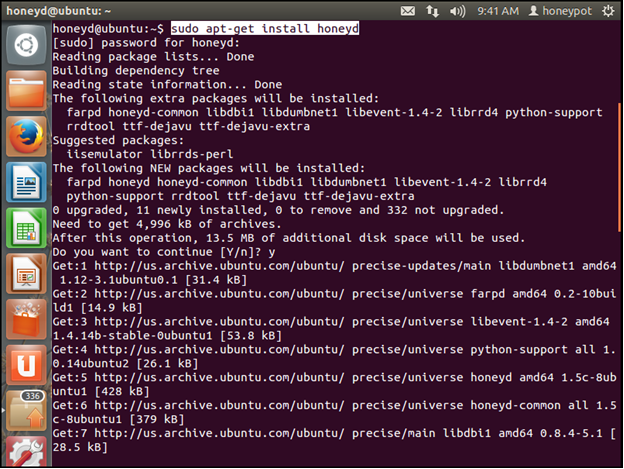


Рисунок 1 – установка HoneyD

Тепер для запуску «пастки» було створено новий конфігураційний файл,

в якому було задано параметри хоста зображені на рисунку 2.

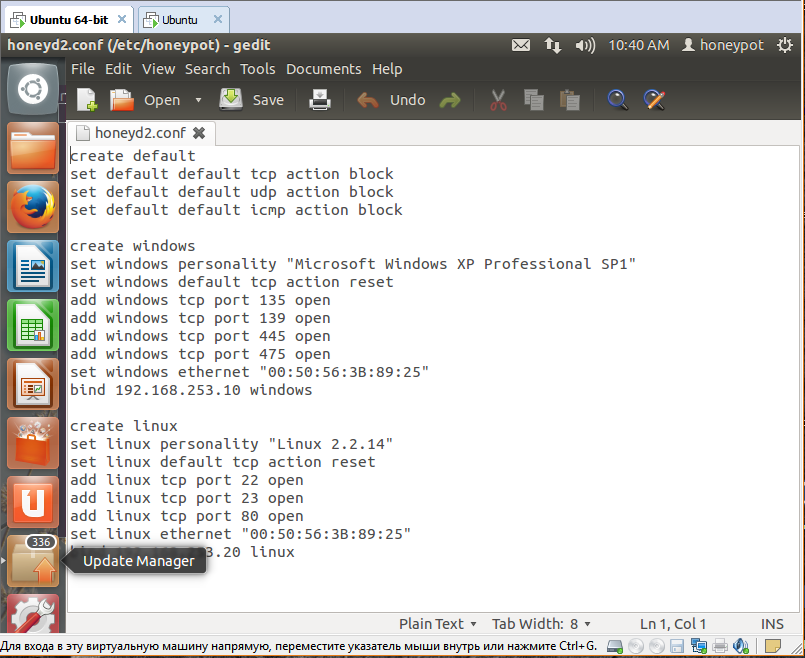


Рисунок 2 – Параметри хоста

Наступний етап запуск HoneyD.

Коректний запуск демону зі статичною IP адресою повинен дати результат який зображений на рисунку 3.

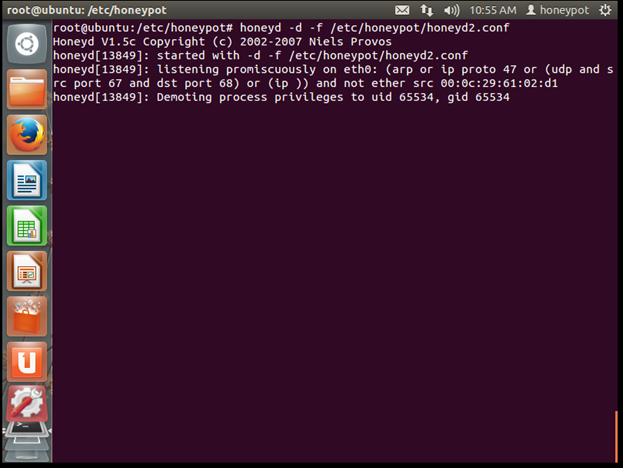


Рисунок 3 – Запуск HoneyD

Проводимо сканування хоста, який працює з honeypot

IP 192.168.253.10 адреса хоста з емульованою ОС «Windows»;

IP 192.168.253.20 адреса хоста з емульованою ОС «Linux»;

Для цього було використано Nmap інстальований на ОС Kali Linux зображено на рисунку 4.

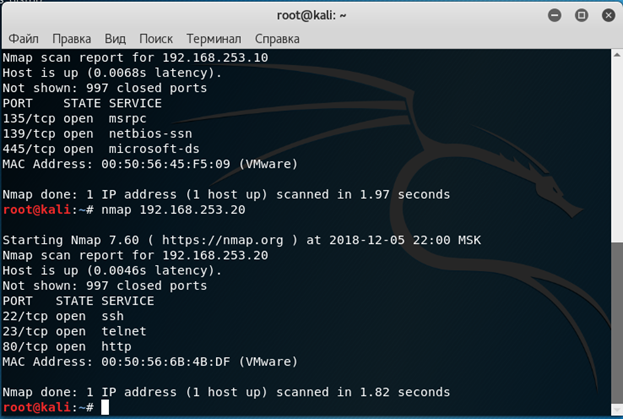


Рисунок 4 – Сканування емульованих хостів

HoneyD було зафіксовано спробу сканування емолюючого хоста (зображено на рисунку 5.)

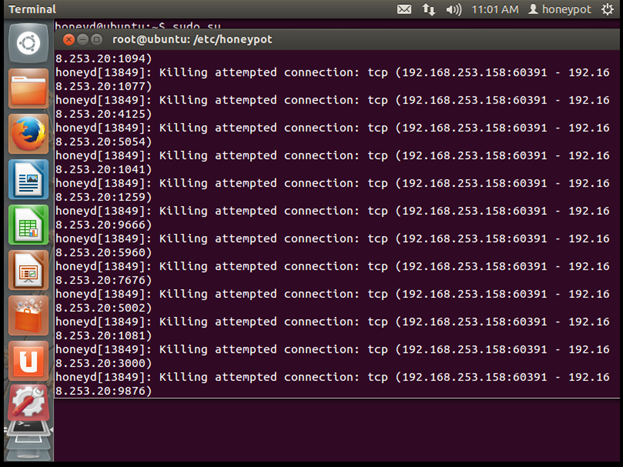


Рисунок 5 – Фіксація спроби сканування

Проведено спробу підключення до емульованого хоста за допомогою Telnet (процес зображений на рисунку 6.)

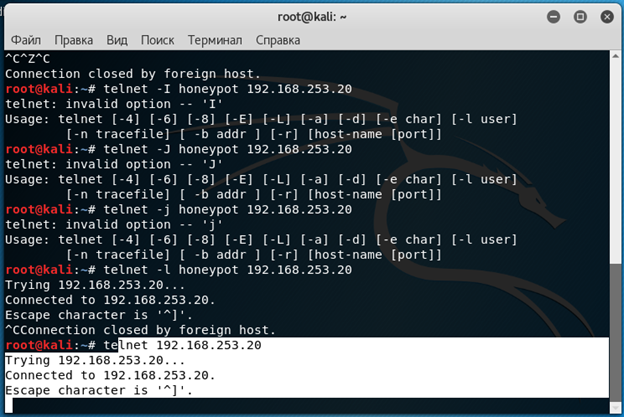


Рисунок 6 – Підключення до емульованого хоста

HoneyD було зафіксовано що була спроба підключення по Telnet (23 порт) дані зображені на рисунку 7.

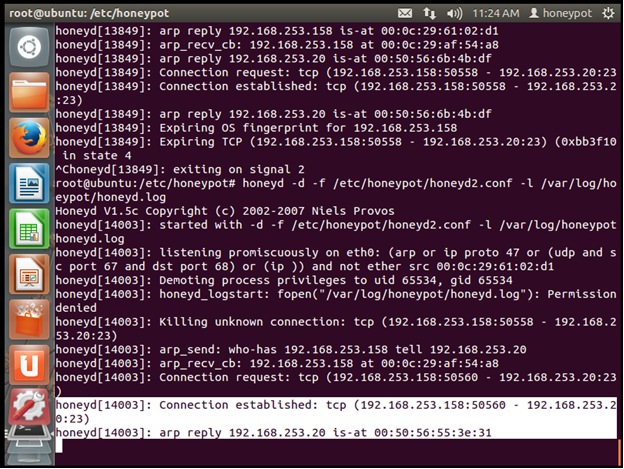


Рисунок 7 – Фіксація спроби підключення по Telnet

Було ініційоване підключення за допомогою SSH процес зображений на рисунку 8.

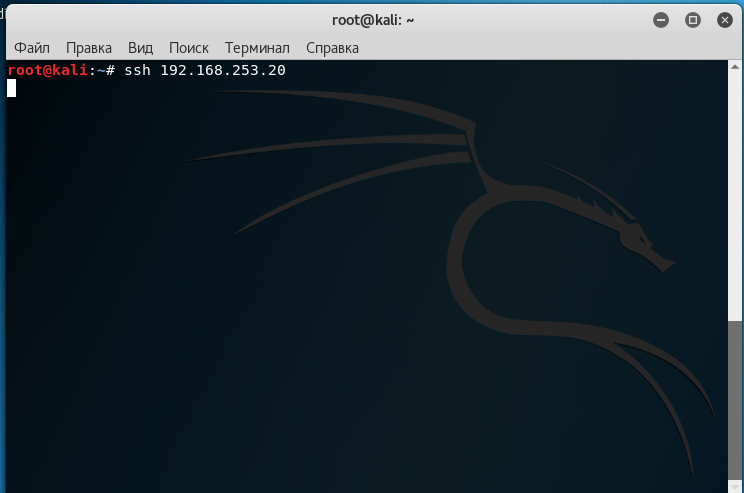


Рисунок 8 – Ініціалізація підключення по SSH

HoneyD було зафіксовано спробу підключення через SSH (22 порт) процес зображено на рисунку 9.

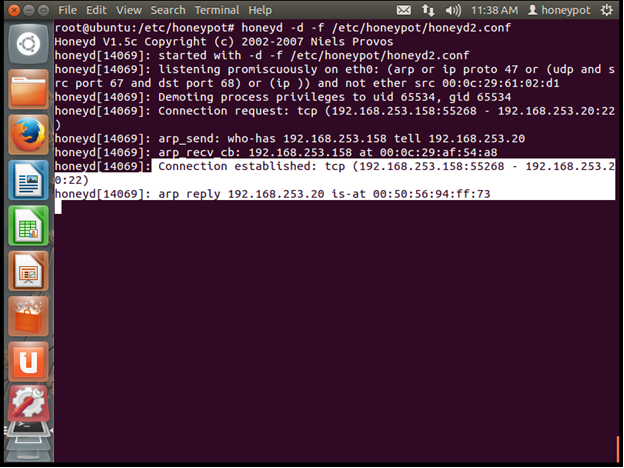


Рисунок 9 – Фіксація спроби підключення по SSH

3.1 Висновки до розділу.

В даному розділі було виконане практичне завдання з використання системи HoneyD. Було дослідженно «пастку» НoneyD, було проведено встановлення та налаштування демону, також було ініційовано атаки на емулюючі хости, проаналізовано дані отримані від НoneyD під час атак.

**ВИСНОВКИ**

В ході роботи була складена класифікація honeypot-систем і виділено типи систем, що представляють найбільший інтерес і мають високий потенціал розвитку. У термінах запропонованої класифікації динамічні високо інтерактивні системи є найбільш актуальними і корисними в сучасних реаліях інформаційної безпеки, на виконання курсової роботи ставилась мета вивчення та практичне застосування Honeypot**.**

Задля досягнення поставленої мети в ході виконання курсової роботи біло виконано наступні завдання:

1. Розглянуто принцип функціонування honeypot-систем;

2. Розглянуто характеристики honeypot системи HoneyD;

3. Розглянуто конфігураційі файли та порядок налаштування honeypot-систем;

4. Виконання практичного завдання з метою застосування HoneyD;

В ході виконання курсової роботи було досягнуто основну мету, а саме розглянуто, реалізовано і вивчення функціонування високоінтерактивної honeypot HoneyD **.**

**СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**

1. Intrusion Detection FAQ: What is a Honeypot? [Електронний ресурс]. Режим доступу: https://www.sans.org/security-resources/idfaq/honeypot3.php, вільний;

2. N. Provos, T. Holz. Virtual Honeypots: From Botnet Tracking to Intrusion Detection – Addison Wesley Professional, 2007;

3. The Honeynet Project [Електронний ресурс]. Режим доступу: https://www.honeynet.org, вільний;

4. Dynamic Honeypots [Електронний ресурс]. Режим доступу: http://www.symantec.com/connect/articles/dynamic-honeypots, вільний;

5. The Deception Toolkit Home Page and Mailing List [Електронний ресурс]. Режим доступу: http://www.all.net/dtk/dtk.html, вільний;

6. LaBrea: "Sticky" Honeypot and IDS [Електронний ресурс]. Режим доступу: http://labrea.sourceforge.net/labrea-info.html, вільний;

7. Dionaea low interaction honeypot [Електронний ресурс]. Режим доступу: https://github.com/rep/dionaea, вільний;

8. Developments of the Honeyd Virtual Honeypot [Електронний ресурс]. Режим доступу: http://www.honeyd.org, вільний;

9. Honeywall project site. Режим доступу: https://projects.honeynet.org/honeywall, вільний;

10. Sebek project site [Електронний ресурс]. Режим доступу: https://projects.honeynet.org/sebek, вільний;

11. Dockpot [Електронний ресурс]. Режим доступу: https://github.com/aabed/dockpot, вільний;

12. T-Pot: A Multi-Honeypot Platform [Електронний ресурс]. Режим доступу: http://dtag-dev-sec.github.io/mediator/feature/2015/03/17/concept.html, вільний; 76

13. Kippo [Електронний ресурс]. Режим доступу: https://github.com/desaster/kippo, вільний;