НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«ДОСЛІДЖЕННЯ MALWARE З ВИКОРИСТАННЯМ МЕТОДІВ ВКЛАДЕНОЇ ВІРТУАЛІЗАЦІЇ»

Виконав: здобувач 4 курсу

Рівень бакалавр

Галузь знань 12 «Інформаційні технології»

спеціальність 125 «Кібербезпека»

Величканич Юрій Юрійович

Керівник: к.т.н., доцент

Бойко Віктор Дмитрович

Рецензент: к.т.н., доцент

Чепурна Олена Євгенівна

Одеса – 2024

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»

Факультет **кібербезпеки та інформаційних технологій**
Кафедра **кібербезпеки**
Галузь знань: **12 Інформаційні технології**
Спеціальність: **125 Кібербезпека**
Назва освітньої програми: **Кібербезпека**

ЗАТВЕРДЖУЮ
Завідувач кафедри кібербезпеки
професор С.А. Горбаченко
_____ «____ »_____ 20__ року

З А В Д А Н Н Я
на кваліфікаційну (бакалаврську) роботу
здобувачу Величканичу Юрію Юрійовичу

1. Тема роботи: «Дослідження Malware З Використанням Методів Вкладеної Віртуалізації»
Керівник роботи: к.т.н, доцент Віктор Дмитрович Бойко
затверджені наказом НУ ОЮА від «__» _____ 20___ року № _____
2. Строк подання здобувачем роботи «__» _____ 20___ рік
3. Дата видачі завдання«__» _____ 20___ рік

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Приміт ка

Здобувач_____
(підпис) (прізвище та ініціали)
Керівник роботи_____
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Дослідження malware з використанням методів вкладеної віртуалізації» на здобуття першого(бакалаврського) рівня вищої освіти за спеціальністю 125 Кібербезпека, освітньою програмою: Кібербезпека, містить 14 рисунків, 33 літературних посилань за переліком посилань.

Робота виконана на (45) сторінках загального тексту та (36) сторінках основного тексту

У процесі роботи було проведено огляд різноманітного malware, також було оглянуто основні методи аналізу malware та його засоби захисту від дослідження. Також було підібрано ефективне вкладене середовище віртуалізації та налаштування цього середовища. Зокрема було проведено налаштування вкладеної віртуальної машини.

Результати дослідження показують високу ефективність динамічного дослідження malware з використанням вкладеного середовища віртуалізації.

Результати дослідження можуть бути використані в подальшому при налаштуванні вкладених віртуальних середовищ котрі дозволять аналізувати та досліджувати malware та покращити сферу захисту від malware.

ABSTRACT

Qualification work on the topic "Malware research using nested virtualization methods" for obtaining the first (bachelor) level of higher education in specialty 125 Cybersecurity, educational program: Cybersecurity, contains 14 figures, 33 literary references on the list of references.

The work was done on (45) pages of general text and (36) pages of main text

During the work, a review of various malware was carried out, and the main methods of analyzing malware and its means of protection from research were also examined. An effective nested virtualization environment and configuration of this environment was also selected. In particular, the nested virtual machine was configured.

The results of the study show the high efficiency of dynamic malware research using a nested virtualization environment.

The results of the study can be used in the future when configuring nested virtual environments that will allow analyzing and researching malware and improving the scope of protection against malware.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

1. Шкідливе програмне забезпечення – ШПЗ
2. Програмне забезпечення – ПЗ
3. Операційна система – ОС
4. Віртуальна машина – ВМ
5. Центральний процесор – ЦП
6. Персональний комп’ютер – ПК
7. Головний завантажувальний запис - MBR

ВСТУП

Malware, також відоме як шкідливе програмне забезпечення, це узагальнений термін котрий використовується для опису програм, котрі створені за для того щоб завдавати шкоду користувачам, або ж ще використовуються для того, щоб отримати несанкціонований доступ до пристрою\пристроїв користувача.

Вже сьогодні Malware – це постійна загроза як для користувачів, так і для комп’ютерних систем. Зловмисники не стоять на місці, вони постійно розвивають шкідливе програмне забезпечення, та з часом вони роблять його все важчими для того щоб його помітити та знешкодити.

Багато людей та власників систем та\або серверів можуть зазнати великої шкоди від ШПЗ, тому що, якщо ці користувачі, системи та сервери не захищені належним чином, зловмисники зможуть мати доступ до певної інформації та використати її в своїх цілях, наприклад розголосити, змінити та\або продати.

За різними дослідженнями таких компаній як Cisco, Microsoft та McAfee було виявлено що порівняно з 2022 роком, у 2023 році кількість кібератак зросла аж на 39% та свідчить про те що в 2023 році кожного дня відбувалося близько 10 мільйонів кібератак різного виду. Кіберзлочинці стають все більш витонченими та мають більш гнучкі методи кібератак від котрих все важче захиститись.

В нашому світі розвиток ШПЗ стає чим балі ти більш швидшим, якщо колись віруси були примітивними та на початку майже не несли в собі ніяких загроз, то в сьогоденні це вже можна назвати повноцінним складним програмним комплексом котрі мають дуже витончені механізми захисту від виявлення, та на додаток мають дуже різноманітні методи мережевої взаємодії, що робить його все більш небезпечнішим для користувачів, та все це ускладнює протидію новому ШПЗ.

Сучасне ШПЗ обладнане засобами захисту в тому числі спрямоване для опору від аналізу (зворотної інженерії, дизасемблювання тощо) ШПЗ. Перевірки котре проводить ШПЗ може включати в себе загальну інформацію властивостей системи, таку як ім’я хоста або ж домену також може включати в себе перевірку мережевого трафіку та це означає що ШПЗ стає все більш витонченим та навчилось визначати віртуальне середовище, після чого воно може або зашифруватись або само знищитись щоб запобігти аналізу. Фахівцям в справі дослідження ШПЗ доводиться постійно використовувати нові та інноваційні методи, котрі можуть бути не так ефективні але старі методи просто не допомагають при аналізі сучасного ШПЗ.

Тому покращення інструментів аналізу ШПЗ являється дуже важливим аспектом, за для запобігання кібератак в різних його проявах, за для того щоб можна було безпечно досліджувати ШПЗ та знати як йому протидіяти.

Мета дослідження: Організація безпечного середовища, котре підходить для дослідження поведінки ШПЗ за допомогою вкладених систем віртуалізації.

Завдання дослідження: Огляд різних літературних джерел про ШПЗ, огляд способів дослідження та аналізу ШПЗ, огляд заходів котрі приймає ШПЗ як контр-міри аналізу, огляд різних можливих методів організації аналізу ШПЗ, огляд різноманітних типів засобів віртуалізації, огляд різних інструментів налаштування віртуального середовища, огляд способів організації віртуального середовища для аналізу та дослідження ШПЗ котре атакує веб-сервер, огляд організації вкладеного віртуального середовища, а також огляд автоматизації організації вкладеного віртуального середовища

Об’єктом дослідження є ШПЗ котре завдає велику шкоду користувачам, комп’ютерним системам та даним. Та в свою чергу предметом дослідження є роботи вкладеної віртуалізації, та яким чином вкладена віртуалізація впливає на ефективність дослідження ШПЗ. Воно може включити в себе порівняння різних типів вкладеної віртуалізації, а також оцінку їхніх слабких та сильних сторін.

Організація та покращення методів дослідження та аналізу ШПЗ дозволить знизити рівень затрат ресурсів та часу для отримання результатів, це дозволить пришвидшити реагування на кіберзагрози, та в цілому позитивно вплине на загальний рівень кібербезпеки

Основні висновки роботи доповідалися та обговорювалися на науковій конференції [1]

1 ЗАСОБИ ТА КОНТР-ЗАСОБИ АНАЛІЗУ

1.1 Malware

Malware – це є скорочення від (англ.) malicious software [2],[3] – це ПЗ, котре створено для «навмисного» виконання шкоди, за для задоволення намірів зловмисника котрий його створив. Aycock (2006) визначає ШПЗ як [4]«ПЗ, намір котрої забезпечує шкоду, або його ефект є шкідливим». Під час винайдення широкосмугового доступу до мережі Інтернет, прості комп'ютері віруси спочатку були написані як самі звичайні «витівки», та вже починаючи з 2003 року, більша частина вірусі та хробаків, набули дуже широкого поширення за для здійснення незаконних вчинків та діяльності на комп'ютерах користувачів, а саме викрадення чи спотворення особистих даних котрі зберігалися на приладах користувачів.

Вперше Malware з'явилося приблизно в 1970-х роках у формі вірусів, котрі заражали комп'ютери користувачів за допомогою заражених флоп-дисків, вже в 80-х роках 20-го ст. почала зростати популярність мережевих технологій то ж через це збільшилося розповсюдження нових та різних видів Malware, наприклад такі як хробаки, котрі могли без участі користувачів, самостійно поширюватись по мережі.

Зростання інтернету під кінець 20-го ст. дало змогу Malware нові шляхи через які вони могли поширюватись. З'явилось ще більше різних видів Malware, наприклад так як троянські коні, хробаки та різні шпигунські програми, котрі мали змогу завантажуватись на пристрій користувача навіть без його відому, вже не кажучи про згоду.

Останніми роками кіберзлочинці почали використовувати все більше витонченіші методи атак, котрі дуже важко вдається виявити стандартними та традиційними методами виявлення, такими як статичний аналіз та сканування сигнатур. За основу таких атак стоїть використання вразливості «нульового дня»

та інші невідомі раніше вразливості, котрі роблять їх невидимими для традиційних методів захисту.

Віруси поділяються на багато різних типів котрі класифікуються за різними ознаками:

- *Комп'ютерні віруси*: Найбільш поширені типи вірусів котрі можуть заражати різні пристрої, та в свою чергу в цих пристроях заражають системні файли та програми, частіш за все поширюються за допомогою мережі інтернет або через зовнішні носії.
- *Мобільні віруси*: Не менш поширені віруси, проте є більш молодшими від комп'ютерних вірусів. Ці віруси заражають мобільні телефони, планшети та ще низку різних пристроїв, можуть поширюватись мережею інтернет та електронну пошту за допомогою заражених файлів, також поширюються за допомогою зовнішніх накопичувальних пристроїв, тощо.
- *Мережеві віруси*: Цей тип вірусів може поширюватись за допомогою комп'ютерних мереж, також вони здібні заражати сервери, маршрутизатори, та ще різні мережеві пристрої.
- *Поліморфні віруси*: Вже віруси такого типу мають можливість змінювати свій код, за для того щоб залишатись непоміченими антивірусами або користувачами
- *Метаморфні віруси*: А цей тип шифрує свій код, теж за для того щоб залишатись непоміченими антивірусами або користувачами

Це були основні та найпоширеніші типи вірусів, також вони можуть бути скомбіновані між собою щоб мати ще більш руйнівну дію.

Також є декілька категорій вірусів, та з основних можна зазначити декілька, а саме:

- *Черви*
- *Трояни*
- *Майнери*
- *Віруси-локери*

Черви поділяються на декілька типів, а саме:

- *Мережеві черви*: цей тип поширюється через мережі, вони використовують вразливості в програмному забезпеченні. Заражають систему через мережу
- *Поштові черви*: вони можуть поширюватись за допомогою пошти через спам, та різні підозрілі файли, вони використовують вразливості в програмному забезпеченні електронної пошти. Заражають систему через відкриття заражених повідомлень в застосунках електронної пошти та\або вкладень
- *Файлові черви*: ці черви можуть заражати файли на пристрої користувача. Зараження відбувається через завантаження та\або відкриття заражених файлів з мережі інтернет та зовнішніх накопичувачів.

Прикладами відомих червів є Stuxnet та Черв’як Моріс

Черв’як Моріс - [5]він був випущений в листопаді 1988-го року, цей хробак був таємно запущений з одного з комп’ютерів в Масачусетському технологічному інституті, та поширювався на комп’ютерах, котрі були підключені до Інтернету, під управлінням BSD-версії UNIX. Черв був спроектований та розроблений таким чином, що його було майже неможливо виявити, та все ж бу недолік в його ***конструкції\дизайні*** котрий змусив його створювати дуже багато своїх копій. Черв’як Морріса не планувався як руйнівний, в тому він сповільнював роботу комп’ютерів таким чином, що він

створював багато непотрібного. навантаження на пристрій, та він не витримував під натиском ваги обсягу оброблюваних даних

[6]*Stuxnet* ще відомий як *win32/stuxnet* - це вірус хробак котрий вражав комп'ютери котрі працювали на ОС Windows, був виявлений в червні 2010 року. Є першим з відомих хробаків котрий перехоплював та модифікував інформаційний потік. Цей хробак може бути використаний як засіб несанкціонованого збору даних.

Трояни поділяються на декілька типів, а саме:

- *Трояни віддаленого* доступ: цей тип надає зловмиснику можливість користуватись пристроєм віддалено.
- *Троян-завантажувач*: цей тип може автоматично без згоди користувача на це, завантажити ще більше ШПЗ котре може зашкодити як користувачу, так і даним на пристрої.
- *Шпигунський троян*: цей тип трояну вміє відстежувати активність користувачів, збирати його дані та надсилати їх власнику ШПЗ
- *Руйнівний троян*: цей тип завдає шкоди самій системі та пристрою.

Всі ці типи можуть бути скомбіновані по різному та використовуватись у власних цілях зловмисників

Також в них є декілька шляхів поширення, а саме:

- *За допомогою заражених файлів*: якщо користувач відкриває або завантажує заражений файл з мережі інтернет або з зовнішнього носія.
- *Соціальна інженерія*: деякі зловмисники вдаються до соціальної інженерії та методом обману «змушують» користувачів завантажити та\або відкрити заражені файли.

Прикладом одного з відомих троянів є ZeuS - [7]він був випущений в 2007 році. Цей троян був розроблений задля того щоб перехоплювати паролі та дані платіжних систем та викрадення грошей. В сумі загальний збиток, котрий був заданий цим ШПЗ складає приблизно 70 мільйонів доларів США

Майнери це ШПЗ котре стало набирати популярність, вони стали дуже широко поширюваними та ним можна заразитись дуже легко, його зараз вкладають майже у всі піратські та безкоштовні софти, всі хто вміє цим користуватись

Майнери, це віруси котрі ще відомі як крипто-майнери або ж криптовалютні майнери, - це ШПЗ використовується за для того щоб видобувати крипто валюту, а для цього використовує процесор(CPU) або ж графічний процесор(GPU) зараженого пристрою користувача. Процес видобутку криптовалюти є дуже енергоємним, та свого роду руйнівним, бо воно використовує по максимуму ресурси процесорів(CPU) та графічних процесорів(GPU) користувачів

Один з відомих майнерів, а саме MinerGate був створений в 2014 році, та це була скоріш платформа для хмарного майнінгу криптовалюти, ця платформа надавала користувачам інформацію та можливості для того, щоб розпочати майнінг без володіння тими чи іншими ресурсами, як обладннаня або ж знаннями в цій галузі. Для початку потрібно було пройти реєстрацію на цій платформі та згодом завантажити ПЗ для майнінгу на свій пристрій, це міг бути як звичайний ПК, так і мобільний пристрій, після цього користувачеві пропонувалося на вибір декілька різних криптовалют, включаючи найпоширеніші як Bitcoin, Etherum, Litecoin та інші. Це програмне забезпечення використовувало обичслювальні ресурси вашого пристрою для видобутку криптовалюти, та ці кошти автоматично зараховувалися на криптогаманці де надалі користувач міг вивести цю валюту або ж обмінювати на різні інші валюти. Платформа звісно вважалась безпечною

та все ж використання такого ПЗ завжди несе ризики ля користувача. Ще в цієї платформи були певні негативні аспекти такі як: Дуже великі комісії, що означало що ви отримували тільки процент від тих коштів котрі ви здобули використовуючи це ПЗ та ресурси свого пристрою, були можливості зловживання вашими ресурсами, бо хмарні платформи часто використовують для виконання різних злодіянь, таких як розподілене відмовлення в обслуговуванні (DDoS) атаки. А також використання цієї платформи передбачає надання певних особистих даних.

Віруси-локери теж є досить актуальними в наш час, вони досить часто зустрічаються при завантажуванні ПЗ з неофіційних або підозрілих сайтів, принципом роботи цього ШПЗ є шифрування файлів та\або блокування доступу до свого приладу, натомість щоб розблокувати файли та в цілому пристрій, зловмисники просять викуп.

Алгоритм дії ШПЗ досить простий, спочатку користувач завантажує заражені файли через мережу інтернет з певних ресурсів, потім при активації цих файлів або одразу, або ж через певний час активується і сам вірус та інформує користувача про блокування доступу та\або шифрування файлів (таких як фото, відео, документи та ін.), разом з цим повідомленням ШПЗ надає інструкції що до розблокування пристрою, а точніше надає інструкцію як потрібно сплатити викуп, частіш за все це криптогаманці, тощо, також може бути повідомлення про те що якщо не сплатити викуп та\або спробувати отримати доступ самостійно, погроза того що файли будуть або безповоротно видалені або ж зашифровані назавжди.

Най відомішими вірусами цієї категорії є такі віруси як Petya та WannaCry

Petya вперше був виявлений в 2016 р. та набув великої популярності [12] наприкінці червня 2017 р. завдяки тому що про нього було повідомлено до NCCIC (National Criminal Intelligence Centr), що одразу відбувалися в декількох країнах світу. Це ШПЗ шифрувало файли з розширенням жорсткого коду (Hard-code list або ж hardcode) та якщо ШПЗ Petya вдавалось отримати доступ до прав адміністратора, то він шифрував MBR, та це в свою чергу робило прилади не придатними для того щоб їх використовували.

На основі аналізу ШПЗ було отримано інформацію що ШПЗ NotPetya шифрував файли на пристроях за допомогою динамічно згенерованого 128-бітним ключем, ще й створював кожній жертві свій особистий ідентифікатор.

В той час в результаті його атаки було завдано збитків на 8 млрд. доларів США

WannaCry – Масоване розповсюдження почалось десь приблизно в середині травня 2017 року, цей випадок атаки був глобальним що теж привело до великих збитків як звичайних користувачів так і різних підприємств та компаній. WannaCry використовував ту саму вразливість що і Petya, а саме вразливість під назвою EternalBlue (Вразливість в безпеці протоколу обміну повідомлень) за для того щоб поширюватись мережею, і та само як всі віруси-локери, це ШПЗ шифрувало файли та вимагало за них певний викуп

Як наслідок, атака цього ШПЗ принесла збитків приблизно в 1 млрд. доларів США, а також привела до порушень роботи критичної інфраструктури, також лікарень та банків

Висновок: У цьому розділі ми оглянули різні типи та категорії ШПЗ, їх властивості та вплив на пристрої. ШПЗ має досить великий спектр форм, починаючи від хробаків та закінчуючи він-локерами. ШПЗ щодня завдає великої шкоди користувачам та різним підприємствам. ШПЗ може привести до різних

наслідків, наприклад втрати даних, фінансові втрати та порушення робочого процесу. Тому важливо завжди заходів щоб захистити себе від ШПЗ

1.2 Методи аналізу malware

Можна виділити дві основні категорії методів дослідження шкідливих програм [13][14]:

Статичні методи, які досліджують ШПЗ без його виконання. Цей метод виконується шляхом аналізу коду, метаданих або інших статичних характеристик шкідливого програмного забезпечення. Перевагами такого аналізу є безпека, так як не потрібно щоб ШПЗ виконувалось, та швидкість, оскільки цей метод не потребує віртуалізації. Проте в нього є певний, і в певному значенні сильний недолік, а саме неточність, а неточним він вважається тому що під час статичного методу не можна врахувати всі можливі шляхи виконання коду ШПЗ.

Також є два основні методи статичного аналізу, а саме статичний сигнатурний та статичний евристичний

Статичний сигнатурний аналіз ШПЗ має на увазі в собі метод котрий використовує статичні сигнатури, або ж шаблони, котрі призначені для виявлення ШПЗ. Сигнатури або шаблони зазвичай базуються на вже відомих шаблонах ШПЗ, або ж на їх загальних характеристиках котрі часто зустрічаються в самому коді ШПЗ

Статичний евристичний аналіз ШПЗ використовує евристичні алгоритми або ж правила для виявлення ШПЗ. Евристика в свою чергу зазвичай базується на інформації про те як зазвичай може поводити себе ШПЗ та як воно працює, також на інформації про загальні характеристики ШПЗ.

Динамічні методи, які досліджують ШПЗ, запускаючи його в контрольованому середовищі. Це дозволяє спостерігати за поведінкою ШПЗ і збирати інформацію про його поведінку. Цей метод використовується за для спостереженням за поведінкою ШПЗ. Перевагами такого методу є те що він є максимально точним, бо він може врахувати багато можливих шляхів розвитку виконання коду ШПЗ, також він буде більш достовірним та повним через те що він може виявити більше функцій ніж статичний, також в процесі дослідження динамічним способом можна отримати більше контексту про те як ШПЗ взаємодіє з системою. Проте цей метод є більш небезпечним бо потребує виконання коду ШПЗ, та якщо бути не обачним, то це ШПЗ може вийти з під контролю та заразити основну ОС

Основним методом динамічного аналізу є відслідковування поведінки ШПЗ а саме аналіз трафіку та аналіз системи.

Аналіз трафіку включає в себе використання інструментів за допомогою яких можна відслідковувати мережеву активність, що включає в себе зв’язок з сервером за допомогою команд за для перевірки його логів, також включає в себе інструменти котрі можуть допомогти ідентифікувати протоколи котрі використовує ШПЗ, та також інструменти за допомогою яких можна виявити IP-адреси командних серверів, що в свою чергу дозволяє відстежити джерело ШПЗ.

Аналіз системи в свою чергу включає в себе можливість дослідити взаємодію ШПЗ з системою, які файли та реєстри воно змінює та які процеси воно запускає, що в свою чергу допомагає зрозуміти як працює та що робить ШПЗ.

З цього виходить що динамічний метод аналізу ШПЗ а саме аналіз мережевого трафіку є найефективніший по декільком причинам, а саме:

Він є найшвидшим та найефективнішим способом отримати велику кількість потрібної інформації про ШПЗ що дозволяє ефективніше знаходити способи протидії та ефективно йому протидіяти.

1.3 Заходи захисту malware від аналізу

[16] Частіш за все зловмисники котрі розробляють ШПЗ можуть використовувати різні системні перевірки, ШПЗ могло уникати аналізу в віртуальному середовищі, та це може нести за собою такі фактори як підміну поведінки ШПЗ якщо воно знайшло ознаки віртуального середовища. Також під час аналізу середовища може включати в себе перевірку адресів мережевих адаптерів, кількість ядер ЦП та доступний обсяг пам’яті. Також ШПЗ може включати в себе загальну перевірку запущених служб котрі можуть бути притаманні тільки для ВМ, у такій програмі як VMWare ШПЗ використовує спеціальний порт вводу та виводу для надсилання команд вихідних даних. Ще перевірка може включати в себе огляд апаратного забезпечення, тобто ШПЗ оглядає наявність вентилятора, огляд температури пристрою, та огляд аудіо пристроїв, що може вказати на віртуальне середовище якщо даних про це апаратне забезпечення не буде, або буде аномальним

З цього виходить те, що аналіз ШПЗ є актуальним методом протидії, але цей процес є досить важким тому що ШПЗ протидіє цьому тим що захищається від спроб аналізу. Це є проблемою, оскільки виникає протиріччя, з одного боку для динамічного аналізу необхідно запустити ШПЗ в середовищі з мережевим доступом – щоб зменшити до мінімуму міри захисту ШПЗ від аналізу та з іншого боку потрібно запускати ШПЗ в ізольованому та безпечному контрольованому середовищі щоб мінімізувати, або ж навіть не завдати шкоди основній системі

2. СТВОРЕННЯ ВКЛАДЕНОЇ СИСТЕМИ ВІРТУАЛІЗАЦІЇ

2.1 Можливі методи організації середовища для аналізу malware

Віртуальна машина(ВМ) – це певне ПЗ, котре може імітувати роботу звичайного фізичного пристрою, на приклад такого як комп’ютер. ВМ дозволяє створювати віртуальні середовища для різних цілей, вони можуть запускати власні ОС та ПЗ, що не залежить від системи хоста. ВМ можна вважати як окремий прилад котрий просто використовує ресурси вашого прилад для того щоб імітувати власні віртуальні процесори, центральний процесор, жорсткий диск та мережевий адаптер. Все це надає змогу запускати різні ОС такі як Windows, macOS та різні дестрибутиви Linux.

В цілому, ВМ є одним з потужних інструментів, котрі використовуються для багатьох пристроїв, до прикладу:

- *Розробка та тестування ПЗ* : ВМ може забезпечити розробнику різні умови та конфігурації котрі допоможуть йому в тестування створеного ним ПЗ для різних ОС та на різних конфігураціях, так само дозволяє проводити тестування для крос-платформового ПЗ.
- *Навчання та освітні процеси*: За допомогою ВМ можна емулювати різні ОС та вивчати різні способи роботи з ними.
- *Дослідження*: При використанні ВМ ви можете емулювати різні сценарії та досліджувати різні аспекти ОС, тестування ПЗ, а ще дослідження та аналіз ШПЗ
- *Запуск серверу*: За допомогою ВМ ви можете запускати в себе на пристрої емуляцію серверу за для того щоб заощадити на фізичному обладнанні.

В кожної ВМ є свої індивідуальні функції, котра робить їх хорошим інструментом для різних цілей. Декілька основних функцій ВМ:

- *Ізоляція та безпека*: За допомогою ВМ можна створити ізольоване середовище котре є відокремленим від основної ОС, тому ПЗ котре ви запускаєте в середині ВМ
- *Запуск кількох ОС*: На одному пристрої можна запустити паралельно різні ОС що значно економить час та кошти, це дозволяє працювати з різними ОС що значно спрощує можливість тестування та\або досліджень
- *Портативність*: Через свою особливість ВМ можна просто копіювати та перемістити з одного пристрою на інший
- *Ресурси*: Ви можете самі виділити потрібну кількість ресурсів для вашої ВМ.
- *Автоматизація*: за допомогою сценаріїв, ВМ можна автоматизувати, що може дуже заощадити час та спростити управління ВМ
- *Сумісність*: ВМ мають високу сумісність з великим спектром ОС, ПЗ та апаратного забезпечення, що робить їх універсальним рішенням котре використовується для різноманітних цілей

Вкладена віртуалізація – це метод котрий дозволяє [17] запускати ВМ в середині ще однієї ВМ. У вкладеній віртуалізації ВМ може виступати як хост для іншого рівня віртуалізації, що дозволяє їй запускати ВМ всередині як гостьову. В основі вкладеної віртуалізації лежить важлива апаратна основа, яка робить можливою цю складну технологію. Варто зазначити що гостьова ОС це система котра встановлена в середині ВМ за допомогою гіпервізора, також на різних ВМ може бути встановлена власна ОС, також є таке поняття як вкладений гіпервізор, це той гіпервізор котрий використовується у вкладених ВМ та використовує такі самі функції як звичайний гіпервізор. Вкладена гостьова система це та система котра встановлюється на вкладену ВМ та підтримується вкладеним гіпервізором.

Більшість процесорів сьогодення вже оснащені підтримкою апаратної віртуалізації. Технологія віртуалізації Intel (VT-x) і віртуалізація AMD (AMD-V) мають в своїй основі можливості вкладеної віртуалізації. Всі ці технології надають весь необхідний набір інструментів, котрі в свою чергу дозволяють процесору обробляти виконання декількох віртуальних машин.

В свою чергу апаратна основа дає гарантії на те, що ЦП, котрий є мозком вашого приладу, є достатньо обладнаний для того, щоб працювати зі складними запусками ВМ одна в одній. Якщо дивитись з іншого боку, можна зрозуміти що апаратні компоненти взаємодіють з ПЗ для досягнення захоплюючих рівнів вкладеної віртуалізації

Все це створює дуже багато різних можливостей для тестування ПЗ, дослідження ШПЗ без шкоди основній машині тому що створене багаторівневе середовище можна налаштувати таким чином що те ж ШПЗ не зможе вийти в основну ОС бо буде переконане в тому що воно там і знаходиться. В цілому можна це зрозуміти так, ВМ – це як окремий комп’ютер, а вкладена ВМ – це як ще один комп’ютер котрий працює в середині попереднього.

Механізм роботи вкладеної віртуалізації максимально простий, для початку на основний пристрій встановлюється потрібне ПЗ за для того щоб забезпечити можливість створювати ВМ, потім створюється віртуальна машина та на неї встановлюється гостьова ОС, все це працює за рахунок гіпервізора 1 котрий керує її ресурсами та ізолює її від інших систем.

[18]Гіпервізор або Монітор віртуальних машин — комп'ютерна програма або обладнання процесора, що забезпечує одночасне і паралельне виконання декількох віртуальних машин, на кожній з яких виконується власна операційна система, на одному фізичному комп'ютері (який зветься хост-машина або хост-комп'ютер, англ. host computer). Гіпервізор забезпечує взаємну ізоляцію операційних систем, що виконуються на віртуальних машинах, шляхом

розділення фізичних та логічних пристроїв між декількома віртуальними машинами.

Гіпервізор 1 – це ПЗ котре керує ресурсами основного пристрою та котрий розподіляє ці ресурси між основною ВМ та рештою. Як ми розглядали вище, гіпервізор розподіляє для ВМ такі ресурси як vCPU, vHDD, vRAM та дозволяє налаштувати віртуальний мережевий адаптер.

Для того щоб використати саме вкладену віртуалізацію в основній ВМ створюється нова ВМ котра вже використовує ресурси основної ВМ котрі були виділені з основного пристрою. Вкладена віртуалізація в свою чергу користується гіперізором 2, котрий працює так само як гіпервіор 1, тільки він бере ресурси не з основної системи, а з основної ВМ, він так само виділяє vCPU, vHDD, vRAM та мережеві налаштування.

Сценарії розвитку роботи з вкладеною віртуалізацією досить різні, воно може допомогти як з тестуванням розробленого ПЗ на різних ОС та при різних обставинах, так і для аналізу та\або дослідження ШПЗ, дослідження його поведінки, його коду та методів його поширення.

Це робить вкладену віртуалізацію дуже потужним інструментом в даному дослідженні, ось декілька прикладів яким способом можна використовувати вкладену віртуалізацію для боротьби з ШПЗ:

- Аналіз ШПЗ в ізольованих ВМ

За допомогою вкладеної віртуалізації користувачі можуть запускати ШПЗ в ізольованому від основного пристрою середовищі, що зменшує майже до мінімуму ризики заразити основний пристрій або ж на інші ВМ, що є одним з головних факторів використання саме вкладеної віртуалізації для цього, також можна створювати багато резервних копій за для дослідження динаміки всього що відбувається при дослідженні ШПЗ, також різні ОС та різні конфігурації

можуть бути налаштовані по різному за для дослідження різних сценаріїв при дослідженні ШПЗ.

- Тестування антивірусного ПЗ та різних інструментів захисту

За допомогою вкладеної віртуалізації користувачі можуть тестувати створені ними антивірусне ПЗ таким чином, що користувач під час увімкненого антивірусного ПЗ зможе завантажувати різне ШПЗ та аналізувати роботу свого антивірусного ПЗ

Також існує багато ВМ, та ось наведено декілька популярних:

- *VMware workstation pro*: Віртуальна машина, котра має великий функціонал віртуалізації, може підтримувати декілька процесорів та віртуальної пам’яті, має широкий спектр різноманітних інструментів котрі допоможуть з розробкою та тестуванням ПЗ.
- *KVM(Kernel-based Virtual Machine)*: Вона є частиною ядра Linux тому це робить її легкою та ефективною, також не потребує встановлення додаткового ПЗ, вона забезпечує досить велику продуктивність, причиною цього є пряма інтеграція з ядром Linux. Ця віртуальна машина має відкритий код та це дозволяє користувачам переглядати, а ще модифікувати код, за для комфортнішої роботи
- *Orcale VirtualBox*: Віртуальна машина котра також має відкритий код, що дозволяє користувачам модифікувати код, має дуже простий та інтуїтивно зрозумілий інтерфейс, та легка в налаштуванні, також підтримує широкий спектр різних ОС

Вибір ВМ залежить від потреб, та в деяких випадках від вашого бюджету. VMware workstation pro буде відмінним вибором для всіх кому потрібно багато різних інструментів та великий обсяг функціоналу, в свою чергу Orcale VirtualBox є чудовим безкоштовним варіантом для всіх користувачів з різними

потребами для домашнього користування або ж для невеликих досліджень та тестування.

Для того щоб розуміти як працює ВМ потрібно зрозуміти три основні аспекти, а саме:

1) Гіпервізор

Гіпервізор є ядром кожної ВМ, котре керує ресурсами фізичного пристрою та розподіляє їх між гостьовими ОС
Існує два основні типи гіпервізору:

- Hypervisor Bare-Metal

Цей тип гіпервізору встановлюється саме на апаратне забезпечення, що забезпечує максимальну продуктивність та можливість контролювати розподіл ресурсів

- Hosted Hypervisor

Цей тип гіпервізорів в свою чергу запускається в межах вже існуючої ОС, що робить їх більш простішими в налаштуванні та в свою чергу управлінні.

2) Віртуальні апаратні засоби

Всі ВМ мають індивідуальні набори віртуального апаратного забезпечення котрі імітують апаратне забезпечення фізичного пристрою, а саме ці віртуальні ресурси включають:

- Віртуальний процесор (vCPU)
- Віртуальний мережевий адаптер
- Віртуальний жорсткий диск (vHDD)
- Віртуальна оперативна пам'ять (vRAM)

3) Гостьові ОС

Гостьовою ОС називається ОС котра встановлена та запускається тільки в середині ВМ, ця ОС працю не залежно від host-OS та має власні файли, драйвери

та налаштування. Це надає змогу запускати різні ОС такі як Windows, macOS та різні дестрибутиви Linux в межах одного фізичного пристрою.

Процес створення та запуску ВМ:

1) Для початку потрібно обрати гіпервізор котрий задовільняє ваші потреби та в повній мірі покриває вимоги.

2) За допомогою обраного раніше гіпервізору потрібно створити нову ВМ виділяючи їх віртуальні ресурси котрі були зазначені визе, а саме vCPU, vHDD, vRAM та задати мережеві налаштування.

3) Після виконаних раніше дій вам порібно обрати ОС та завантажити її на віртуальний жорсткий диск вашої ВМ

4) Після всіх виконаних раніше дій, встановлюєте гостьову ОС та налаштовуєте його, та ця ВМ буде працювати як самостійний комп’ютер з власною ОС та різним ПЗ

Також існує багато різновидів віртуалізації

Та ось декілька основних типів:

Віртуалізація на рівні ОС

Цей тип віртуалізації дає нам змогу запускати декілька ОС на одному простої, також кожна ВМ має своє власне апаратне забезпечення так ОС, також вони всі можуть працювати незалежно одна від одної.

Віртуалізація апаратних засобів

Цей тип віртуалізації дає можливість віртуалізувати апаратне забезпечення, наприклад процесор, жорсткий диск або ж мережеві пристрої та диски. Цей тим вважається більш гнучким але з іншого боку воно є більш складним в реалізації та потребує більше ресурсів.

Таким способом можна емулювати QEMU – може емулювати різні процесори, до прикладу x86, ARM, PowerPC та інше, що дозволяє запускати ПЗ

котре розроблялося зовсім для інших апаратних засобів, також дозволяє емулювати ігрові консолі.

Віртуалізація процесів

Здійснюється з використанням ядра ОС хоста, котре використовується контейнером (Docker, Podman, і т.д.). В свою чергу контейнер це портативне віртуальне середовище котре містить багато потрібного за для того щоб запускати ПЗ, аналіз та дослідження ШПЗ.

Повна віртуалізація

Ця технологія віртуалізації базується на тому щоб використовувати ресурси всі ресурси основного пристрою на котрому запущене віртуальне середовище. Це означає те що ВМ має власні віртуальні апаратні засоби такі як процесор, жорсткий диск та інші апаратні пристрої.

За для того щоб ретельно дослідити та проаналізувати ШПЗ, рекомендовано запустити його в ізольованій вкладеній системі віртуалізації, що мінімізує ризики пошкодження host ОС та даних котрі знаходяться на ній, також контрольоване оточення забезпечить більш точне дослідження з різними сценаріями.

2.2 Вибір типу та засобів віртуалізації

Вибір типів та засобів віртуалізації залежать від задачі аналізу ШПЗ

Виявлення вразливостей – для цього най краще використовувати ВМ з чисто ОС, без оновлень та стороннього ПЗ. Це дозволяє детальніше визначити поведінку ШПЗ та як воно впливає на систему.

Pen-test (тестування на проникнення) – для цього типу тестування краще використовувати ВМ з самими стандартними конфігураціями котрі включають в себе оновлення та різне ПЗ котре частіш за все використовується в

звичайному середовище, що дозволяє досліджувати ШПЗ в більш типових умовах

Аналіз динамічної поведінки ШПЗ – Для цього типу аналізу ШПЗ частіш за все використовується пісочниця або ж контейнер, що дозволяє дослідити те як ШПЗ взаємодіє з системою в реальному часі.

Збіг апаратного забезпечення та ОС

Перевагами цього є стабільність, продуктивність та сумісність. Ця система буде стабільно та правильно працювати, вона буде більш продуктивною та всі драйвери будуть без проблем працювати з ПЗ. Та недоліками може бути Вартість та обмеженість вибору. Апаратне забезпечення котре буде збігатись з вимогами ОС може бути дорожчим в вартості та не всі компоненти можуть бути актуальними для тої чи іншої ОС, що означає більш обмежений вибір та функціонал.

Розбіжність апаратного забезпечення

Перевагами цього є Економія ресурсів та більш різноманітний вибір. Деяке апаратне забезпечення може бути більш дешевим та вибір є набагато більшим. Проте це може вплинути на стабільності, продуктивності та можуть бути проблеми з сумісністю. Система може працювати нестабільно через те що апаратне забезпечення не цілком сумісне з ОС, також через це понижується продуктивність та можуть виникнути проблеми з драйверами та ПЗ.

Вкладені системи віртуалізації можуть бути однотипні та різнотипні

Однотипними вкладеними система віртуалізації – це той тип котрий використовує на одному хостові створюється ВМ котра стає хостом для ще однієї ВМ того ж типу. Також такому типу вкладеної системи віртуалізації притаманна велика ефективність котра пов’язана з тим що це є економією ресурсів для основного хоста оскільки вони використовують одні й ті ж самі ресурси котрі розподілив перший гіпервізор та вже ці ресурси розподілилися між двома ВМ. Також цей тип забезпечує ізоляцію між ВМ що робить їх більш безпечними. До

прикладу можна використати дві однакові віртуальні машини по типу VirtualBox + VirtualBox.

В свою чергу різнотипна вкладена система віртуалізації – це той тип вкладеної системи віртуалізації коли на одному хості створюються дві різні ВМ. Такому типу віртуалізації притаманно те що вони використовують по різному ресурси основного хоста, що є більш ресурсоємним процесом ніж однотипний тип вкладених систем віртуалізації. До прикладу можна використати дві однакові віртуальні машини по типу VirtualBox + QEMU.

В свою чергу однотипні вкладені системи віртуалізації вважаються більш продуктивними та в свою чергу більш безпечнішими ніж різнотипна вкладена система віртуалізації.

2.3 Вибір інструментів для налаштування віртуального середовища

Створення та налаштування віртуального середовища з нуля є дуже трудомістким завданням котре вимагає знань про віртуалізацію. Тому тому щоб розподілити свій час більш раціонально потрібно автоматизувати налаштування ВМ

Щоб автоматизувати цей процес можна скористатись такими засобами інформації як:

Скрипти Bash/CMD. Bash (англ.) Bourne-Again shell – [24] це командна оболонка котра є основною за замовчуванням в ОС котрі зроблені на базі Unix та Linux та була створена в 1989 р. Ця оболонка дає можливість на автоматизацію різних задавань для робочого оточення

Перевагами використання скриптів створених за допомогою Bash є те що ці скрипти можна адаптувати о різному, в залежності від налаштувань віртуального середовища, а також вони дозволяють зекономити зусилля та час.

CLI інтерфейси засобів віртуалізації. VBox CLI або ж VBoxManage – це командний інтерфейс котрий дозволяє керувати ВМ. Цей командний інтерфейс дозволяє використовувати всі ті ж самі функції як і в графічному інтерфейсі.

взаємодіє з Main API в фоновому режимі, в свою чергу коли клієнтське ПЗ може безпосередньо використовувати Main API для керування ВМ VirtualBox. Отже ці інструменти можна використовувати одночасно, а зміни котрі були внесені за допомогою цього інструменту буде одразу видно під час використання іншого, До прикладу, зміни котрі були внесені за допомогою CLI, та це негайно відображається в графічному інтерфейсі, це працює і навпаки.[25]

Ось схематична архітектура Oracle VM VirtualBox:

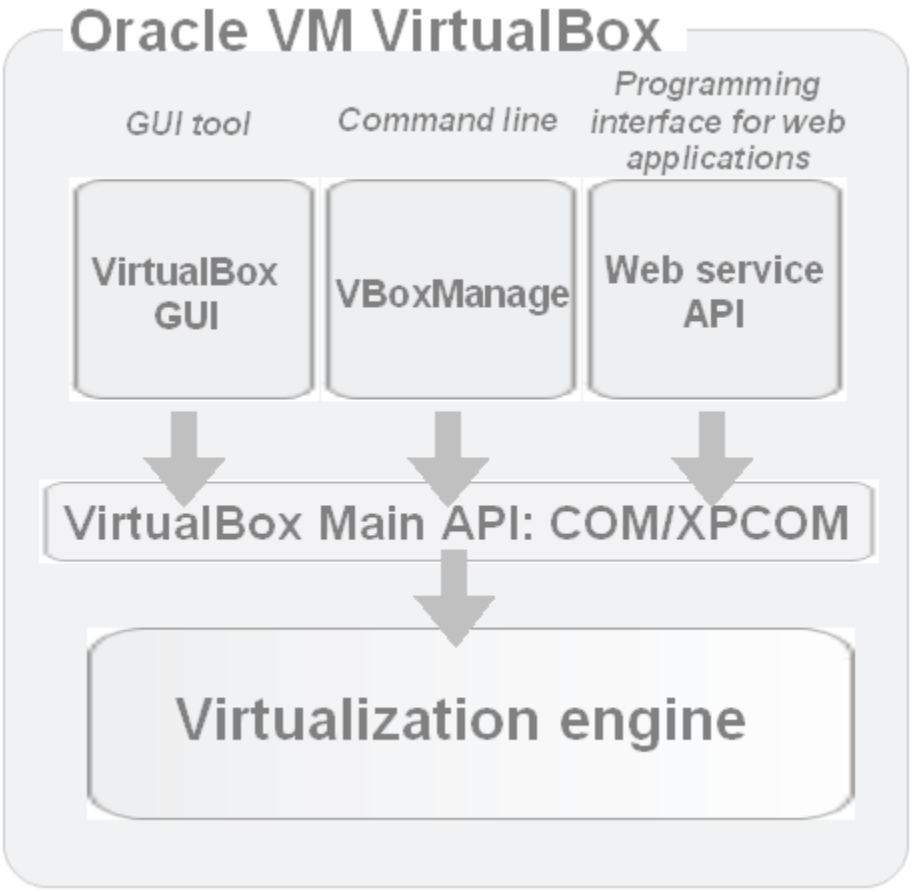

Рисунок 2.1. Інтерфейси для взаємодії з Oracle VM VirtualBox

VBox CLI дозволяє створити ВМ за допомогою команди «`createvm`», потім можна провести реєстрацію за допомогою команди «`registervm`»

Vargant– це інструмент віртуалізації з відкритим кодом котрий розробила компанія HashiCorp [26]. Цей інструмент використовується для спрощення та автоматизації керування віртуальним середовищем ВМ, він дозволяє створювати та змінювати конфігурації ВМ в одному файлі під назвою *Vargantgile*, також цей інструмент дозволяє створювати та запускати ВМ у різноманітних середовищах. Основними перевагами цього інструменту є простота в синтаксисі, його можна використовувати для запуску різні ВМ на різних платформах та ОС, також воно добре інтегрується з різними інструментами автоматизації, до прикладу той самий Ansible

Використання за допомогою Vargant відбувається таким чином, що спочатку описується конфігурація (провайдер віртуалізації, до прикладу VirtualBox, тип ОС, різноманітне ПЗ та інші налаштування) ВМ в файлі Vargantgile, після цього потрібно запустити за допомогою команди «`vargrant init`» в каталозі, в котрому знаходиться файл Vargantgile, що завантажить образ ОС та створить папку provision для сценаріїв налаштування, та після цього потрібно запустити команду «`vagrant up`» щоб в подальшому створити та запустити ВМ котра буде відповідати конфігурації, що задана в Vargantgile. Та в кінці роботи можна ввести та запустити команду «`vagrant destroy`», щоб видалити всі ресурси котрі пов’язані з нею

Ansible – [27] це один з най використовуваних інструментів з відкритим кодом котра допомагає користувачам автоматизувати, що дає змогу керувати хмарною та локальною інфраструктурою, він забезпечує крос-платформову автоматизацію, він використовує поняття керованих та контрольованих вузлів. Він підключається до керованих вузлів надсилаючи їм команди. Сам Ansible базується на мові опису котра називається YAML, зо робить його досить простим у вивченні та і в цілому у використанні. Основними можливостями Ansible є те що він використовується для автоматизації налаштування серверів,

мережевих пристроїв, також використовується для автоматизації розгортання нового ПЗ та оновлення вже існуючого, може координувати виконання завдань одразу на декількох остах та за допомогою Ansible можна легко відстежувати зміни в конфігураціях. Перевагами цього інструменту є те що він досить простий в використанні, може автоматизувати досить широкий спектр завдань та може контролювати одразу багатьма хостами.

З цього всього виходить те що за для організації комфортного робочого місця для динамічного аналізу та дослідження ШПЗ, оптимальне використання вкладених систем віртуалізації, та при цьому дуже доречне використання автоматизації конфігурування та налаштування даних систем

3 ВКЛАДЕНІ СИСТЕМИ ВІРТУАЛІЗАЦІЇ ДЛЯ АНАЛІЗУ MALWARE

3.1 Організація віртуального середовища для аналізу malware котре атакує веб-сервер

Для організації віртуального середовища котре буде максимально ефективне для аналізу ШПЗ потрібно обрати образ ОС котра працює зі стеком LAMP / LAPP / LNPP (L – Linux; A/N – Apache2 / Nginx; P/M – Postgres / MySQL; P – Perl / PHP / Python)

Для початку потрібно обрати дестрибутиву Linux, одразу можна згадати про такі як Ubuntu та Debian, ці дестрибутиви є досить надійними та популярними варіантами для віртуалізації, завдяки своїй стабільності, а також широкому набору ПЗ, в даному, потім потрібно обрати ПЗ котре буде виступати провайдером віртуалізації (В даному випадку вибір зупинився на Oracle VM VirtualBox). Наступним кроком потрібно налаштувати сам стек (в даному

випадку LAMP), для цього можна використати раніше згадувані інструменти Vargant або Ansible

Задля безпеки потрібно налаштувати мережу, а саме потрібно почати з вибору мережевого режиму, є два хороші варіанти, а саме це Bridge mode та NAT(Network Address Translation) mode, перший варіант дозволяє ВМ використовувати основну IP-адресу приладу хоста, та це є досить небезпечним при дослідженні ШПЗ, оскільки воно не забезпечує повної ізоляції, та другий варіант є більш підходящим варіантом, бо цей мережевий режим створює віртуальну мережу для ВМ зі своїм власним IP-адресом, та це вже забезпечує більшу ізоляцію, та дозволяє керувати трафік.

Налаштування правил брандмауера також дієвий спосіб захистити основний прилад, можна створити власні правила котрі, до прикладу, можуть дозволити потік тільки потрібного трафіку з середини та з зовні, тобто можна дозволити доступ до веб-серверу з пристрою хоста та заблокувати його з всіх інших пристроїв.

Також для аналізу ШПЗ нам можуть знадобитись інструменти мережевого аналізу, цими інструментів є tcpdump та Wireshark, котрі дозволяють відстежувати потік трафіку та досліджувати його, що в свою чергу допоможе ідентифікувати підозрілу активність ШПЗ.

3.2 Організація вкладених віртуальних середовищ

За для того щоб створити вкладене середовище віртуалізації потрібно завантажити та встановити на основну ВМ, провайдер віртуалізації, як і на пристрій хост та також забезпечити його всім необхідним ПЗ.

В даному випадку буде використовуватись така ВМ як Oracle VM VirtualBox. Для початку її потрібно завантажити, якщо це виконувати на Windows то цей процес є максимально постим, заходимо на офіційний сайт Oracle VM VirtualBox та завантажуємо файл котрий дозволить інсталювати ПЗ та

інсталюємо це ПЗ, але в випадку з дистрибутивною Linux, за для того щоб завантажити VirtualBox потрібно зайти в командну строку та вже в ній потрібно прописати певні команди:

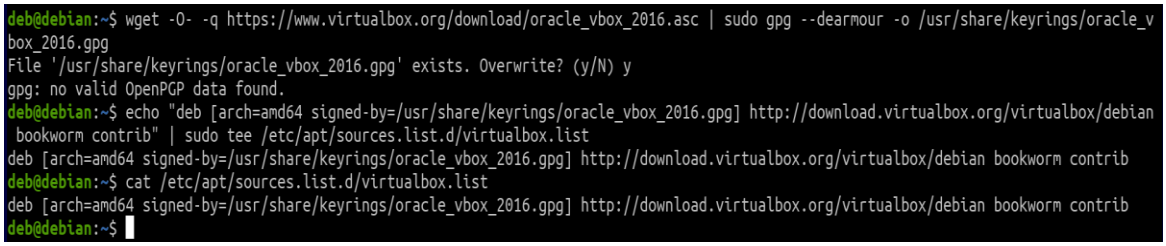

Рисунок 3.1. Приклад команди котра додає програму в «repositorium», простіше кажучи в сховище.

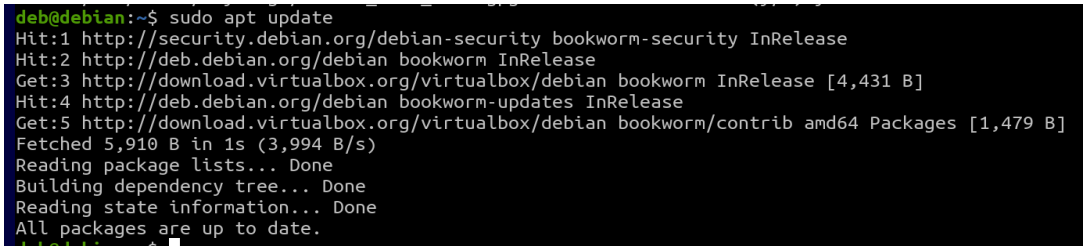

Рисунок 3.2. Оновлення списку пакетів.

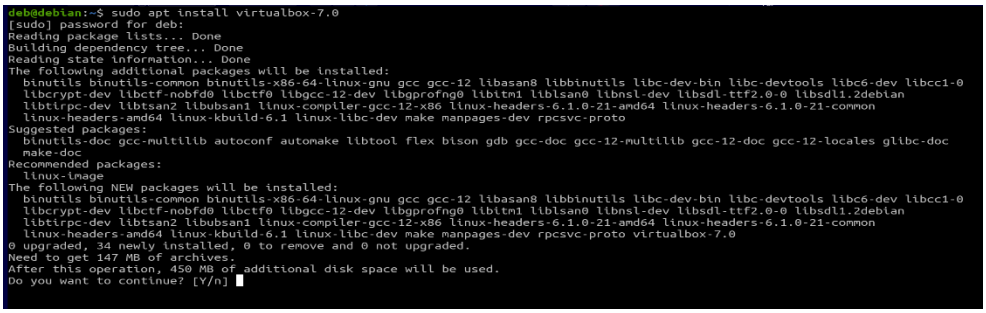

Рисунок 3.3. Інсталяція VirtualBox.

Після інсталяції ВМ потрібно встановити на неї конфігурацію її мережевих карт, а саме для першої обирається «NAT (Network Address Translation)», а для другої кращим варіантом буде «Віртуальний адаптер хоста (virtualbox0)»

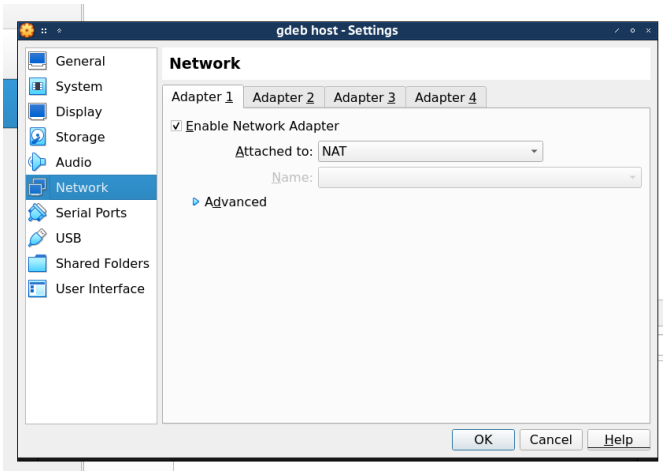

Рисунок 3.4. Мережева карта NAT.

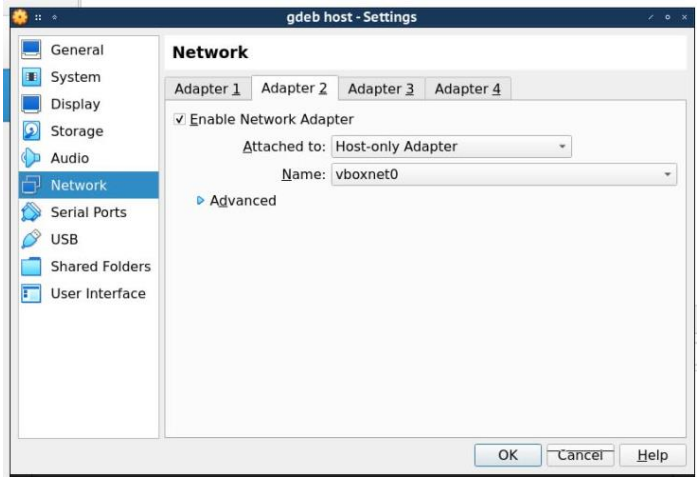

Рисунок 3.5. Мережева карта Віртуальний адаптер хоста (virtualbox0).

Після виконаних дій за для того щоб мати змогу зробити вкладене віртуальне середовище нам потрібно встановити конфігурацію вкладеної віртуалізації, тобто потрібно відмітити прапорець «Enable Nested VT-x/AMD-V» в налаштуваннях конфігурації мережі ВМ.

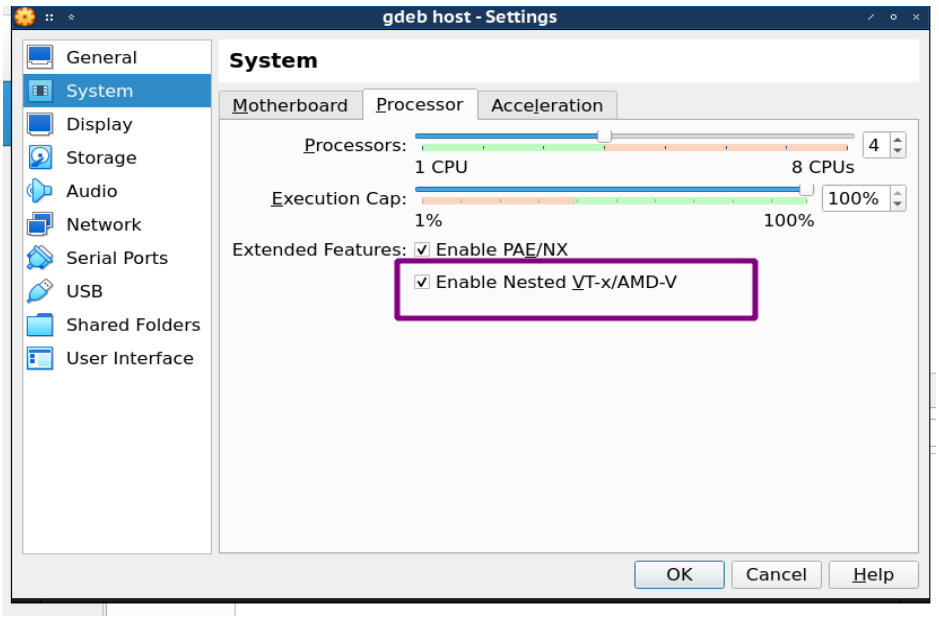

Рисунок 3.6. Увімкнення конфігурації вкладеної віртуалізації.

Коли всі перелічені дії були виконані потрібно приступити до інсталяції гостьової ОС в даному випадку було обрано ОС Debian Linux, для цього нам потрібно підключити iso-образ ОС. Після цього всього потрібно продовжити базове налаштування ОС.

Також в ВМ мережеві карти будуть відображатися як *enp0s3* та *enp0s8*, де enp0s3 – відображає налаштування мережевої карти NAT, а enp0s8 – відображає віртуальний адаптер хоста – virtualbox0.

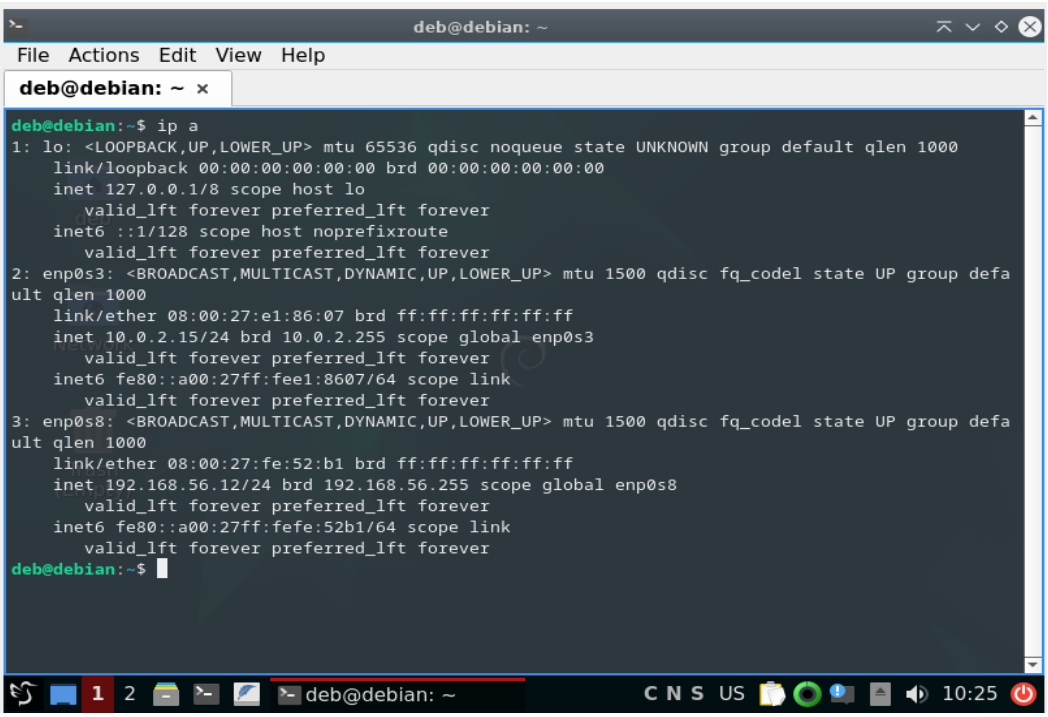

Рисунок 3.7. Огляд мережевих карт ВМ.

Після виконаних вище дій потрібно всередині ВМ інсталювати VirtualBox таким чином, за допомогою ssh потрібно підключитись до ВМ

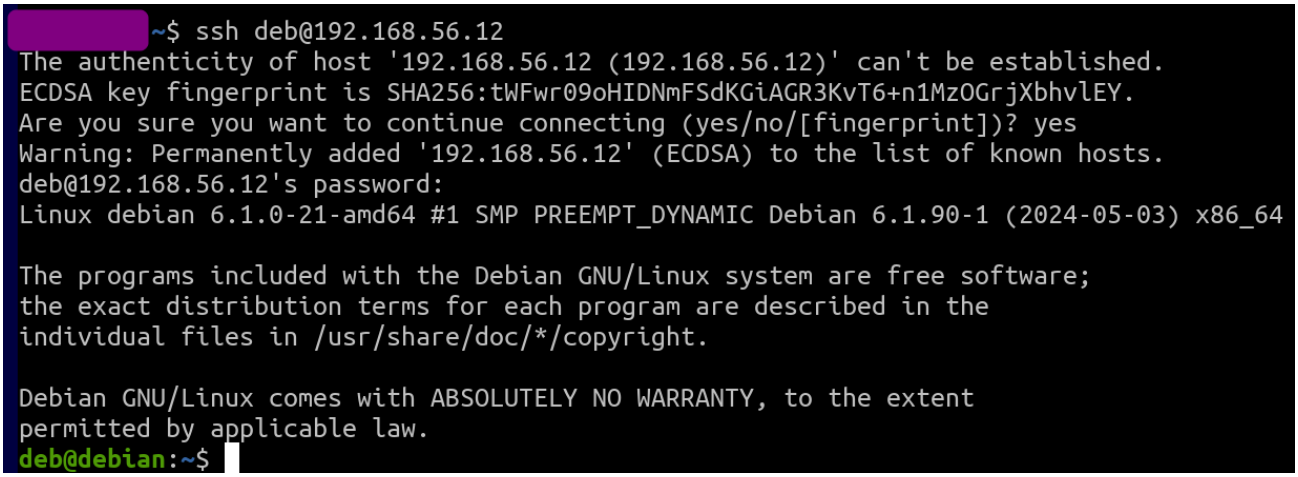

Рисунок 3.8. Підключення до ВМ за допомогою ssh.

Та також таким самим способом як вище завантажуємо VirtualBox на ВМ

На вкладеній віртуальній машині аналогічно потрібно викласти конфігурації мережевих карт такі як «NAT (Network Address Translation)» та «Віртуальний адаптер хоста (virtualbox0)» та вкладені мережеві карти гостьової вкладеної ОС також мають таке саме позначення як і в попередньої ВМ, а саме *enp0s3* та *enp0s8*, де enp0s3 – відображає налаштування мережевої карти NAT, а enp0s8 – відображає віртуальний адаптер хоста – virtualbox0.

В отриманій схемі можна зрозуміти що утворені ланцюги позначають інше:

- Ланцюжок NAT enp0s3 гостьова ОС-2 --> NAT enp0s3 гостьова ОС --> мережева карта хоста – Забезпечує доступ гостьових ОС до мережі інтернет
- Ланцюжок virtualbox0 enp0s8 гостьова ОС-2 --> virtualbox0 enp0s8 гостьова ОС --> ssh - Забезпечує доступ в систему за допомогою ssh

3.3 Розгортання альтернативних систем

В якості вкладеної гостевої ОС було обрано дестрибутиву DSL [33](Damn Small Linux - це дистрибутива Linux котра базується на Debian) за для того щоб бачити наглядну різницю.

Процес завантаження є досить простим та збігаються дещо з тим що вказано вище

Для початку потрібно запустити провайдер віртуалізації, тобто в даному випадку це VirtualBox, після цього потрібно створити оболонку ВМ а саме дати їй назву, обрати папку в котрій вона буде зберігатись, обрати тип та версію ОС, а також додати обраний та заздалегідь завантажений iso-образ потрібної ОС

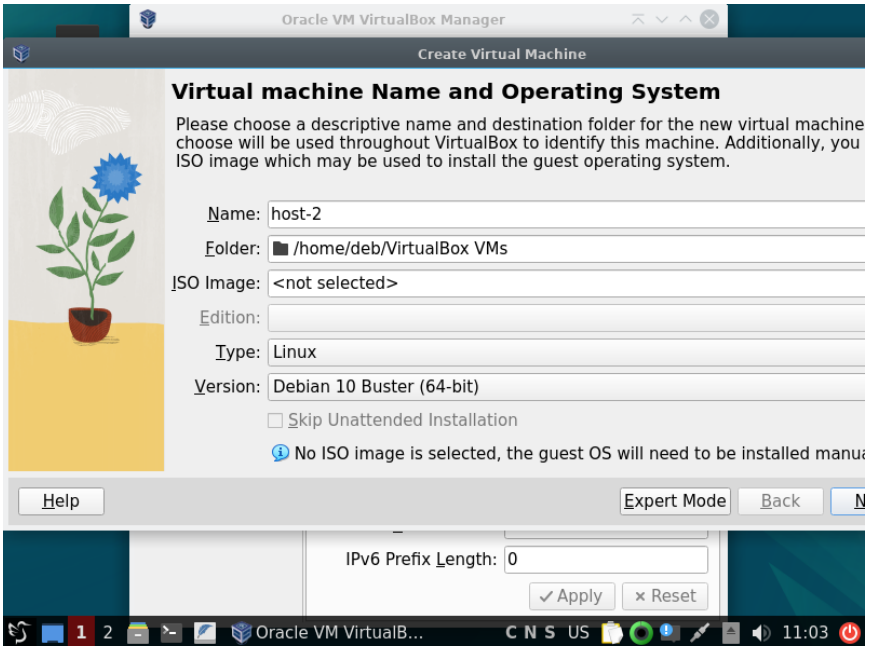

Рисунок 3.9. Створення ВМ.

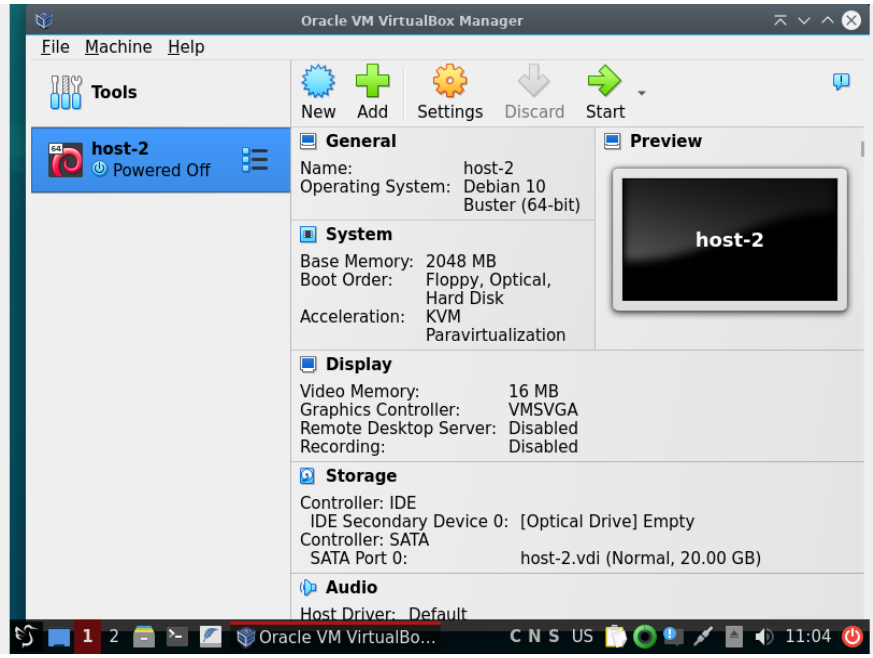

Рисунок 3.10. Створена ВМ в інтерфейсі провайдера.

Та вже після виконаних дій можна запускати ВМ

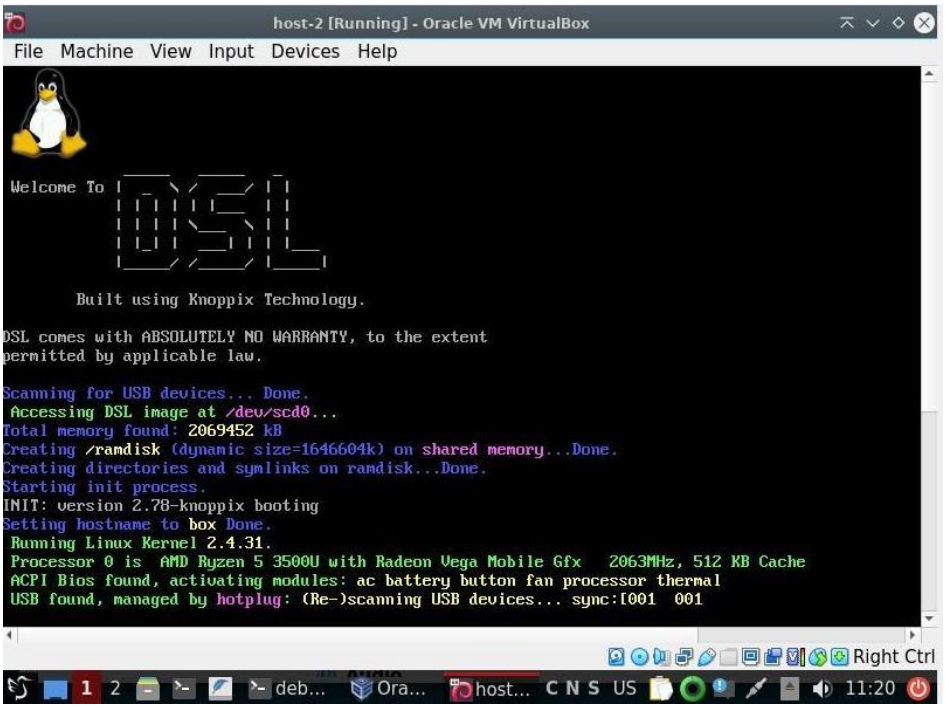

Рисунок 3.11. Запуск вкладеної гостевої ВМ.

Важливо переконатися що ОС працює справно

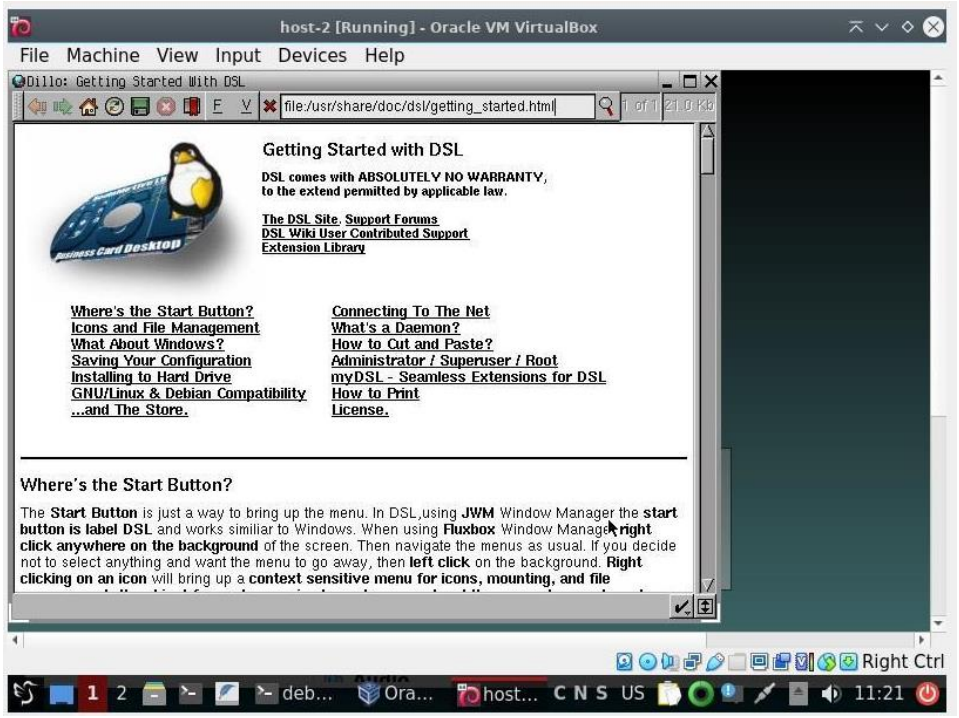

Рисунок 3.12. Огляд віртуальної ОС на працездатність.

Також можна оглянути мережеві карти

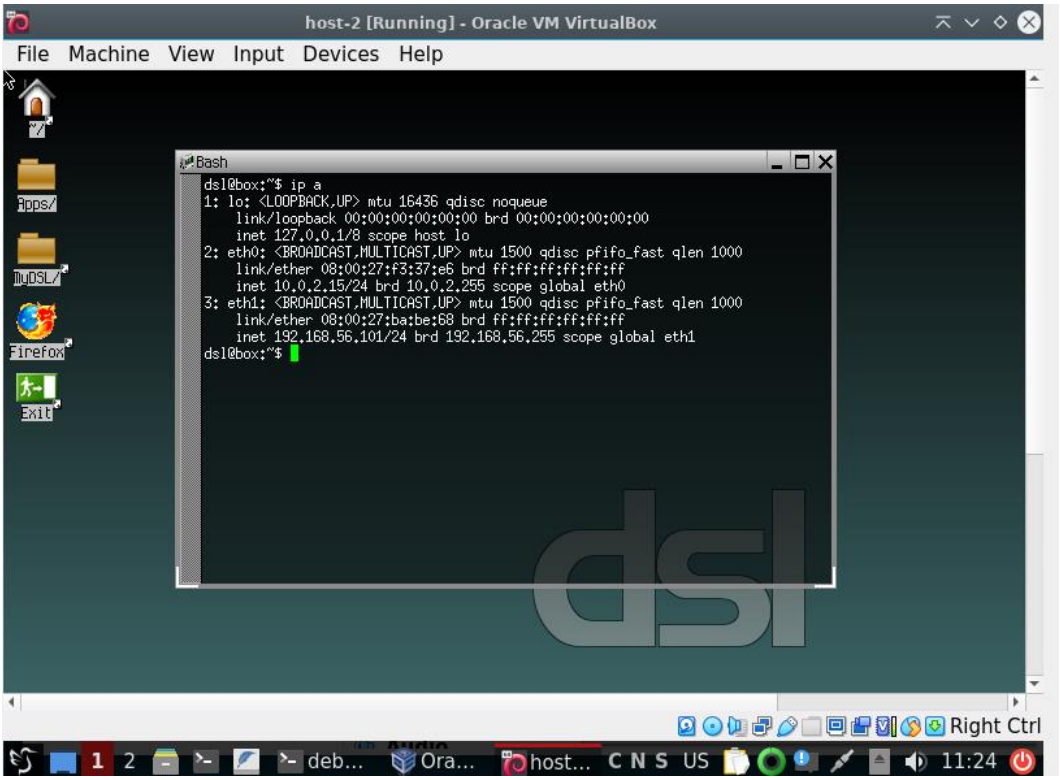

Рисунок 3.13. Огляд мережевих карт на вкладеній ВМ.

Також щоб запевнитися що вкладена ВМ дійсно має доступ до мережі інтернету можна в командній строці вписати таку команду як sudo ping 8.8.8.8 котра провіряє з'єднання з сервером Google DNS за допомогою протоколу ICMP

Таким чином було організовано середовище для аналізу ШПЗ, яка з одного боку забезпечує повний доступ мережі, та з іншого боку ізолює ШПЗ котре досліджується всередині вкладеної віртуальної машини. Навіть якщо весь процес вийде з під контролю, постраждає тільки вкладена ВМ, яку в свою чергу можна буде досить легко відновити

ВИСНОВОК

Кваліфікаційна робота містить дослідження в ході яких було вияснено що аналіз ШПЗ є актуальним методом протидії, але цей процес є досить важким тому що ШПЗ протидіє цьому тим що захищається від спроб аналізу, також було організоване середовище для динамічного аналізу та дослідження ШПЗ та оптимальне використання вкладених систем віртуалізації, котре з одного боку забезпечує повний доступ до мережі інтернет та з іншого боку ізолює ШПЗ котре потрібно досліджувати всередині вкладеного середовища віртуалізації від основного пристрою, що дозволить відновити ВМ котра може постраждати в процесі дослідження, та не зменшить ризики пошкодження основного пристрою

Ключовою вимогою для вирішення проблеми використання вкладеного середовища віртуалізації для аналізу та дослідження ШПЗ.

Також у ході дослідження було отримано практичні та теоретичні навички в роботі з вкладеним віртуальним середовищем та в сфері дослідження ШПЗ

У результаті було створено та налаштовано вкладене середовище віртуалізації для аналізу ШПЗ.

Список використаної літератури

1. Величканич Ю. Дослідження malware з використанням методів вкладеної віртуалізації. Матеріали Міжнародної науково-практично конференції «Кіберпростір в умовах війни та глобальних викликів XXI століття: теорія та практика» (м. Одеса, 24 листопада 2023 р.). Одеса, 2023. с.51-53

2. Anusmita Ray, Dr. Asoke Nath. "Introduction to Malware and Malware Analysis: A brief overview" (2016)

3. Verma, Aparna, M.S.Rao, A.K.Gupta, W. Jeberson, and Vrijendra Singh. "A Literature Review On Malware And Its Analysis." International Journal of Current Research and Review 5 (2013), 71-82.

4. Aycock J. Computer viruses and malware. – Springer Science & Business Media, 2006. – T. 22.

5. Kelty, Christopher “The Morris Worm” (2011)

6. Wikipedia URL: http://surl.li/kynkl

7. NV Бізнес URL:https://biz.nv.ua/ukr/markets/microsoft-vidmovilasja-vid-oplati-bitkoini-2444153.html

8. VMware URL:https://www.vmware.com/products/desktop-hypervisor.html

9. VitualBox URL:https://www.virtualbox.org/wiki/Downloads

10. KVM URL:https://docs.kernel.org/virt/kvm/index.html

11. CISA URL:https://www.cisa.gov/news-events/alerts/2017/07/01/petya-ransomware

12. Rami S., Khairuddin O., K. A. Z. Ariffin. Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid, and Memory Analysis URL: https://www.researchgate.net/publication/328760930_A_Survey_on_Malware_Analysis_Techniques_Static_Dynamic_Hybrid_and_Memory_Analysis

13. Dynamic Malware Analysis and Detection in Virtual Environment URL: http://surl.li/tprmi

14. Whitepaper Minergate URL: https://whitepaper.minergate.xyz/

15. MITRE ATT&CK Virtualization/Sandbox Evasion: System Checks URL: https://attack.mitre.org/techniques/T1497/001/

16. What is nested virtualization? 101 Guide URL: https://monovm.com/blog/nested-virtualization/#What-is-Nested-Virtualization?

17. Europol URL: https://www.europol.europa.eu/wannacry-ransomware

18. Wikipedia. Гіпервізор - https://uk.wikipedia.org/wiki/Гіпервізор

19. Cisco Talos Cybersecurity Report URL: https://www.cisco.com/c/en/us/products/security/cybersecurity-reports.html

20. McAfee Reveals 2024 Cybersecurity Predictions: Advancement of AI Shapes the Future of Online Scams URL: https://www.mcafee.com/tr-tr/consumer-corporate/newsroom/press-releases/press-release.html?news_id=74501198-8690-45bd-8d63-80bfce4ee18c

21. Microsoft Didgital Defense Report URL: https://www.microsoft.com/en-us/security/security-insider/microsoft-digital-defense-report-2023

22. Wikipedia URL: http://surl.li/efihq

23. Vasilev Y. Controlling VirtualBox from Command Line URL: https://www.oracle.com/technical-resources/articles/it-infrastructure/admin-manage-vbox-cli.html (Червень 2014)

24. aCode Що таке bash в Linux? Гайд по написанню bash-скриптів URL: https://acode.com.ua/bash-in-linux/#toc-4

25. VirtualBox manual Chapter 8 VBoxManage URL: https://www.virtualbox.org/manual/ch08.html

26. Dev. Vargant For Beginner: Getting Started Guide URL: https://dev.to/breakingpitt/vagrant-for-beginners-getting-started-guide-3dhd

27. Spacelift. Ansible Tutorial for Beginners: Ultimate Playbook & Examples URL: https://spacelift.io/blog/ansible-tutorial

28. Learnansible. Getting Started with Ansible: A Beginner's Guide URL: https://learnansible.dev/article/Getting_Started_with_Ansible_A_Beginners_Guide.html

29. tcpdump. TCPDUMP & LIBPCAP URL: https://www.tcpdump.org/

30. Wireshark. Wireshark Developer`s Guide URL: https://www.wireshark.org/docs/wsdg_html_chunked/

31. Chapter 6. Virtual Networking URL: https://www.virtualbox.org/manual/ch06.html

32. DigitalOcean. How to Set Up a Firewall with UFW on Ubuntu URL: https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu

33.Wikipedia. Damn Small Linux URL: https://uk.wikipedia.org/wiki/Damn_Small_Linux