МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»
В.о. завідувача кафедри
_____ Оксана ШОВКОПЛЯС
(підпис)
6 грудня 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки»,
освітньо-професійної програми «Інформатика»
на тему: Інформаційна технологія створення додатку для навчання та тренування з кібербезпеки
здобувача групи ІН.м-32 Бовкуна Дмитра Олексійовича

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

_____ Дмитро БОВКУН
(підпис)

Керівник,
завідувач кафедри кібербезпеки,
кандидат технічних наук, доцент Володимир ЛЮБЧАК _____
(підпис)

Суми – 2024

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»
В.о. завідувача кафедри
_____ Оксана ШОВКОПЛЯС
(підпис)

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки», освітньо-професійної програми «Інформатика»
здобувача групи ІН.м-32 Бовкуна Дмитра Олексійовича

1. Тема роботи: «Інформаційна технологія створення додатку для навчання та тренування з кібербезпеки»
затверджую наказом по СумДУ від «3» грудня 2024 р. №1257-VI
2. Термін здачі здобувачем кваліфікаційної роботи до 6 грудня 2024 року
3. Вхідні дані до кваліфікаційної роботи _____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) *1) Аналіз проблеми предметної області, постановка й формування завдань дослідження. 2) Огляд технологій, що використовуються для реалізації додатку. 3) Моделювання та проєктування об'єкту досліджень: визначення основного функціоналу, який необхідно реалізувати згідно завдання. 4) Вибір інструментів для реалізації завдання. 5) Розробка додатку для навчання та тренування з кібербезпеки. 6) Аналіз результатів.*
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)_____
6. Консультанти до проєкту (роботи), із значенням розділів проєкту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «____» _____ 20 ___ р.

Завдання прийняв до виконання _____ Керівник _____
(підпис) (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	*Аналіз проблеми предметної області, постановка й формування завдань дослідження*		
2	*Огляд технологій, що використовуються для реалізації додатку та їх ролі у системі*		
3	*Моделювання та проєктування об'єкту досліджень: визначення основного функціоналу, який необхідно реалізувати згідно завдання*		
4	*Вибір інструментів для реалізації завдання*		
5	*Розробка додатку для навчання та тренування з кібербезпеки*		

6	*Аналіз отриманих результатів*		
7	*Оформлення пояснювальної записки до кваліфікаційної роботи*		

Здобувач вищої освіти _____ Керівник _____

(підпис) (підпис)

АНОТАЦІЯ

Записка: 74 стор., 42 рис., 6 табл., 4 додатки, 26 використаних джерел.

Обґрунтування актуальності теми роботи – Тема кваліфікаційної роботи є актуальною, оскільки присвячена інформаційній технології створення та використання додатку для навчання та тренування з кібербезпеки з її повним контролем з боку адміністратора системи.

Об'єкт дослідження — процес розробки, налаштування та експлуатації тренувальної платформи для виконання вправ з кібербезпеки, що забезпечує автономність, персоналізацію та доступність для користувачів.

Мета роботи — розробка інформаційної технології створення додатку з кібербезпеки для підвищення рівня фаховості, компетентності та обізнаності осіб, що навчаються, з питань кібербезпеки.

Методи дослідження — огляд існуючих аналогів, розробка моделі/сценарію функціонування системи та створення пілотного зразку платформи для навчання та тренування з кібербезпеки на основі описаної у роботі інформаційної технології.

Результати — розроблено додаток, що дає можливість виконувати персоналізоване автоматизоване розгортання окремих робочих мереж та вразливих машин, які розгортаються у даних мережах, з можливістю віддаленого доступу до них для цільових клієнтів мережі.

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ, ВІРТУАЛІЗАЦІЯ, КІБЕРБЕЗПЕКА, VIRTUAL BOX, АВТОМАТИЗАЦІЯ, УПРАВЛІННЯ ВІРТУАЛЬНИМИ МАШИНАМИ, JAVA, JAVAFX, BASH

ЗМІСТ

ВСТУП

Актуальність. З кожним роком значення кібербезпеки для компаній, урядових організацій і приватних осіб стає все більш важливим. В усьому світі та Україні спостерігається постійне зростання кількості кіберзлочинів, у тому числі атаки на фінансові системи, викрадення особистих даних, атаки на об’єкти критичної інфраструктури. Хоча на даний момент ефективність роботи засобів захисту та методів протидії кіберзагрозам значно ефективніша ніж раніше, все ж різноманітність програмного забезпечення, складність сучасних систем та зростаюча кількість та різноманітність пристроїв, що підключені до Інтернету, створюють постійні загрози для захисту інформації. Успішна протидія таким загрозам вимагає не лише теоретичних знань, але й практичних навичок. На сьогодні традиційні підходи до навчання з кібербезпеки не завжди відповідають потребам швидкої адаптації до нових видів атак, оскільки вони часто не забезпечують достатнього рівня гнучкості.

Дослідження і розробка доступних тренувальних платформ для вправ з кібербезпеки є актуальними, оскільки поширення їх використання дозволить підвищити обізнаність фахівців у сучасних кіберзагрозах та способах до їх протидії.

Об'єктом дослідження дипломної роботи магістра є процес розробки, налаштування та експлуатації тренувальної платформи для виконання вправ з кібербезпеки, що забезпечує автономність, персоналізацію та доступність для користувачів. Платформа включає персоналізовані віртуальні мережі для кожного студента та унікальні хеш-коди для розв’язування кожної окремої машини, що забезпечить їх безкоштовну автономну неперервну самостійну роботу зі зручним підключенням та без необхідності завантаження ресурсів на їх власний пристрій.

Предметом дослідження є методи і технології розробки та функціонування інформаційної технології створення тренувальної платформи для вправ з кібербезпеки.

Мета дослідження. Розробка інформаційної технології створення тренувальної платформи вправ з кібербезпеки для підвищення рівня фаховості, компетентності та обізнаності осіб, що навчаються, з питань кібербезпеки.

Основними задачами досягнення мети кваліфікаційної роботи є:

1) проведення аналізу предметної області та огляд існуючих додатків-аналогів;

2) розробка структури роботи платформи, визначення способів реалізації основного функціоналу системи;

3) написання основних сценаріїв роботи програми;

4) розробка користувацького інтерфейсу адміністратора платформи.

Гіпотеза дослідження. Засобами Virtual Box можна розгорнути середовище для колективного або групового виконання вправ з кібербезпеки, яким централізовано може керувати адміністратор системи.

Новизна. На відміну від існуючих аналогів інформаційних систем, описаний у даній роботі додаток базується на популярному серед користувачів гіпервізорі Virtual Box і надає можливість розгортання персоналізованих середовищ користувачів та груп користувачів платформи (розгортання власних робочих мереж та вразливих машин) адміністратором системи. Також існує можливість додавання до платформи образів машин як власних (зі знанням логіну та паролю користувача) так і запозичених з інших тематичних платформ (коли дані автентифікації файлу образу ova гіпервізору Virtual Box невідомі).

Практична цінність створеного продукту полягає в допомозі у вивченні різноманітних сценаріїв атак на вразливі системи для розуміння, як ці атаки відбуваються та які заходи можуть бути ефективно застосовані для їх запобігання і нейтралізації.

Структура. Дана робота складається зі вступу, аналізу існуючих платформ та інструментів для тренування та навчання з кібербезпеки, постановки задачі дослідження, опису використаних інструментів та їх ролі у вирішенні поставленої задачі, опису реалізованих програмних модулів додатку, висновків, списку використаних джерел та додатків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Геймифікація як інноваційний метод сучасного навчання

Завдяки прискоренню темпів розвитку цифрових технологій та цифровізації суспільства зростає потреба у підготовці висококваліфікованих фахівців, здатних і готових вільно орієнтуватися в інформаційному просторі, використовувати цифрові технології задля ефективного виконання поставлених завдань, а також застосовувати здобуті знання, уміння та навички задля досягнення особистих і професійних цілей. Одним із дієвих засобів підвищення мотивації навчання є гейміфікація [1].

Гейміфікація впроваджує ігрову тактику в освіту, тоді як ігрове навчання використовує гру як частину навчального процесу. Гейміфікація використовується в усіх аспектах нашого життя, щоб збільшити задоволення та зацікавленість. Існує два основних типи гейміфікації: структурна гейміфікація та контентна гейміфікація. Гейміфікація має два основні підходи: структурну гейміфікацію, яка інтегрує ігрові елементи у вже існуючу структуру навчального процесу, і контентну гейміфікацію, яка перетворює навчальний матеріал на ігровий контент [2].

Гейміфікація має суттєві переваги для освітнього процесу: вона сприяє розвитку критичного мислення, підвищує мотивацію та залученість студентів, формує прагнення досягати поставлених цілей та дає змогу застосовувати знання на практиці.

Як приклад гейміфікації у кібербезпеці можна вважати платформи для кібер-змагань (такі як Capture The Flag або CTF). На таких платформах учасники змагаються у вирішенні реальних кібербезпекових задач, які імітують загрози та вразливості, що можуть виникнути у реальному середовищі. Учасникам пропонуються різні категорії завдань, такі як злом вебдодатків, аналіз шкідливого програмного забезпечення, криптографія, реверс-інжиніринг та інші. Кожне успішно вирішене завдання приносить бали, що підвищує рівень зацікавленості і мотивує учасників вдосконалювати свої знання та навички.

1.2 Платформи для навчання та практики в галузі кібербезпеки

Для навчання та підвищення кваліфікації фахівців у сфері кібербезпеки застосовуються різноманітні методики, включно з традиційними освітніми

технологіями, такими як лекції, семінари та практичні заняття, а також електронні засоби навчання, які значно розширюють можливості для здобуття теоретичних знань та практичних навичок. Проте, окрім традиційних методів, все більшу популярність набувають сучасні тренувальні платформи та віртуальні симулятори, які спеціально розроблені для відпрацювання навичок кібербезпеки. Ці платформи дозволяють імітувати складні сценарії кіберзагроз, включаючи атаки, які часто зустрічаються в реальному середовищі, і розробляти ефективні стратегії їхнього подолання.

Такі тренувальні платформи забезпечують користувачам можливість практичного досвіду у вирішенні реальних проблем без ризику для реальних інформаційних систем. Це дозволяє спеціалістам з кібербезпеки безпечно експериментувати з новими методами захисту, тестувати уразливості та відпрацьовувати навички протидії атакам у симульованих умовах. Це особливо важливо, оскільки кіберзагрози постійно еволюціонують, що вимагає від фахівців у цій сфері не лише знань сучасних технологій, але й навичок швидкої адаптації до нових викликів.

Підготовка кадрів за допомогою подібних платформ не лише підвищує рівень кіберобізнаності, але й дозволяє компаніям значно скоротити час та витрати, необхідні для навчання працівників. Це, своєю чергою, допомагає організаціям бути більш підготовленими до можливих кіберзагроз, зменшити ризики, пов’язані з потенційними кіберінцидентами, і створити ефективну команду фахівців, здатних швидко та ефективно реагувати на атаки.

У сучасному світі, де кіберзагрози можуть завдати значної шкоди як компаніям, так і державним установам, наявність професійної команди з кібербезпеки є важливим елементом загальної стратегії безпеки організації. Тому використання навчальних платформ з кібербезпеки є ключовим інструментом для постійного підвищення кваліфікації фахівців, що відповідають за безпеку в інформаційних системах.

1.3 Аналіз існуючих рішень.

На сьогодні існує достатня кількість платформ та тренажерів для навчання та практики в галузі кібербезпеки. Нижче розглянемо кілька найпопулярніших прикладів таких платформ, а також переваги та недоліки їх використання у навчанні.

1. Hack The Box (https://www.hackthebox.com/) – це онлайн-навчальна платформа з кібербезпеки, що дозволяє людям та компаніям підвищувати свої навички пентестінгу в ізольованому, ігровому навчальному середовищі.

Серед переваг даної платформи можна виділити наступні:

1. Широкий вибір завдань — Hack The Box пропонує великий набір завдань різної складності та тематики — від базових тестів на знання мережевих протоколів до складних сценаріїв експлуатації уразливостей. Це дозволяє користувачам поступово покращувати свої навички, відшліфовувати їх і рухатися від простих до складних задач. Такий підхід підходить як для новачків, так і для досвідчених спеціалістів, які прагнуть підтримувати форму і постійно розвиватися.

2. Реалістичні сучасні сценарії — завдання, запропоновані на Hack The Box, максимально наближені до реальних загроз і ситуацій, з якими можна зіткнутися у професійному житті. Це можуть бути завдання з обходу систем захисту, аналізу складних конфігурацій або пошуку вразливостей в сучасних програмних продуктах. Такий підхід сприяє розвитку навичок, які стануть у пригоді в роботі, а також допомагає вчасно реагувати на сучасні тенденції у сфері кіберзахисту.

3. Активна спільнота — Hack The Box має велику та дружню спільноту, яка готова ділитися досвідом та допомагати іншим користувачам. Багато учасників обмінюються підказками, інсайтами і рішеннями завдань, не порушуючи політики конфіденційності платформи. Це сприяє спільному навчанню і створює підтримуюче середовище, де можна задавати питання і отримувати відповіді навіть на складні теми.

Недоліки платформи наступні:

1. Вартість деяких функцій — хоча основний доступ до платформи безкоштовний, Hack The Box має ряд платних функцій, які можуть стати корисними для серйозних користувачів або корпоративних команд. Наприклад, можливість

проходження неактивних машин та комплексних сценаріїв потребує придбання платної підписки.

2. Складність для початківців — платформа часто вимагає досить високого рівня знань, і новачки можуть зіткнутися з труднощами, коли не мають достатнього досвіду в базових техніках пентестінгу чи налаштуванні мережі. Для початківців відсутність покрокових інструкцій або доступних підказок може ускладнити процес навчання, тому вони можуть відчувати розчарування або втратити мотивацію.

3. Незручність через скрипти відновлення системи — Hack The Box використовує скрипти для відновлення систем через певні проміжки часу. Це може бути незручним для користувачів, оскільки певні дії, які були зроблені під час роботи над завданням можуть бути скинуті. Для користувачів платформи це означає, що процес доведення сценарію до кінця може іноді перериватися, що робить експлуатацію складнішою і менш гнучкою, особливо якщо користувач хоче повернутися до попереднього етапу або протестувати іншу методику з деякими попередніми діями.

2. Try Hack Me (https://tryhackme.com/) – це безкоштовна онлайн-платформа для вивчення кібербезпеки за допомогою практичних вправ і лабораторних робіт.

Переваги:

1. Дружній інтерфейс — Try Hack Me має зручний та інтуїтивно зрозумілий інтерфейс, який сприяє легкому початку навчання. Це особливо корисно для новачків, які можуть швидко освоїтись у системі без потреби у глибоких знаннях програмування чи мережевих технологій. Навігація платформою проста, а завдання чітко структуровані, що дозволяє користувачам зосередитися на навчанні замість пошуку необхідних інструментів чи функцій. Дружній інтерфейс робить платформу доступною для широкого кола користувачів, незалежно від їхнього рівня підготовки.

2. Широкий каталог навчальних ресурсів — Try Hack Me надає великий вибір матеріалів: від базових тем, таких як мережевий захист та криптографія, до більш складних тем, як експлуатація вебуразливостей чи аналіз шкідливого ПЗ. Ці матеріали часто представлені в інтерактивному форматі, що полегшує засвоєння інформації. Крім того, платформа регулярно оновлюється, додаючи нові лабораторні завдання та

кімнати, що допомагає користувачам залишатися в курсі сучасних кіберзагроз та методів їхньої нейтралізації.

3. Можливість створення власних кімнат із завданнями — Try Hack Me дозволяє створювати власні «кімнати» із завданнями, що дозволяє більш досвідченим користувачам створювати унікальні сценарії або підготувати практичні завдання для командних тренінгів та змагань. Це забезпечує додаткову персоналізацію та можливість розвивати свої навички не лише в контексті наданих завдань, але й у створенні нових викликів. Ця функція особливо корисна для викладачів, які можуть адаптувати навчальні матеріали під конкретні цілі та рівень знань студентів.

Недоліки платформи:

1. Обмеження безкоштовного доступу для деяких завдань і функцій — хоча основний функціонал Try Hack Me є безкоштовним, доступ до деяких завдань та додаткових можливостей обмежений у безкоштовній версії. Наприклад, можливість створювати власні кімнати з завданнями, яка була згадана вище, доступна лише користувачам платної версії, що може бути розчаруванням для користувачів, які бажають розширити свої можливості без додаткових витрат. Тож для тих, хто планує поглиблено навчатися, безкоштовних матеріалів може бути недостатньо, що створює певні фінансові бар'єри для користувачів.

2. Обмежена складність для більш просунутих користувачів — Try Hack Me здебільшого орієнтована на новачків і користувачів із середнім рівнем підготовки, а тому завдання та матеріали можуть бути недостатньо складними для досвідчених фахівців у сфері кібербезпеки. Хоча платформа пропонує завдання різної складності, частина контенту може здатися простим для просунутих користувачів, які потребують глибшого занурення та складніших викликів. Це може обмежити цікавість досвідчених пентестерів або фахівців із захисту інформації, які прагнуть продовжувати розвиватись і потребують завдань із вищим рівнем складності.

3. DVWA (https://github.com/digininja/DVWA/) – відкритий проєкт, що пропонує безкоштовні сценарії для тестування вразливостей вебдодатків.

Переваги:

1. Повністю безкоштовні сценарії відомих вразливостей — DVWA надає користувачам відкритий доступ до сценаріїв виявлення та експлуатації поширених вебуразливостей без будь-яких витрат. Це особливо важливо для тих, хто тільки розпочинає навчання у сфері кібербезпеки, адже дозволяє працювати з реалістичними прикладами та експериментувати з різними техніками атак у безпечному середовищі. Сценарії включають практику роботи з відомими уразливостями, такими як SQL-ін'єкції, XSS (міжсайтовий скриптинг) та CSRF (підробка запитів між сайтами).

2. Розгортання конкретних вразливостей у власній системі — використовуючи DVWA, користувачі мають можливість розгортати й експлуатувати уразливості безпосередньо у своїй системі. Це дозволяє глибше зрозуміти, як певні уразливості впливають на безпеку вебдодатків, та попрактикуватися у їхньому виправленні. Можливість налаштування рівнів безпеки уразливостей допомагає зрозуміти, як базові та просунуті налаштування безпеки змінюють поведінку вебдодатку.

3. Повністю підходить для новачків — DVWA створене з метою навчання, а тому не вимагає від користувачів попередніх знань про складні атаки або захисні механізми. Додаток містить уразливості, пояснення яких можна легко знайти в документації, що дозволяє зрозуміти основи кібербезпеки у зрозумілому форматі. Цей аспект робить DVWA доступним для тих, хто лише починає знайомитись з кібербезпекою вебдодатків і хоче навчитися базових технік пентестингу.

Недоліки:

1. Фокус тільки на вебуразливостях — DVWA обмежений у своїй тематиці та фокусується виключно на вебуразливостях. Це робить його менш корисним для тих, хто прагне вивчати ширший спектр кіберзахисту, включаючи мережеві атаки, аналіз шкідливого ПЗ або методи соціальної інженерії. Користувачам, які хочуть розширити свої знання за межі вебдодатків, доведеться звертатися до інших інструментів і платформ для всебічної практики.

2. Обмежена складність для просунутих користувачів — хоча DVWA є корисним для початківців, його складність може бути недостатньою для досвідчених фахівців у сфері кібербезпеки. Навіть з можливістю змінювати рівні безпеки, завдання можуть здатися надто простими для тих, хто вже має досвід у виявленні та експлуатації

складних уразливостей. Це може обмежити ефективність використання DVWA для професіоналів, які шукають більш складні і реалістичні сценарії.

3. Розгортання вимагає встановлення вебсерверу — DVWA потребує налаштування середовища розгортання, зокрема встановлення вебсервера, PHP та бази даних MySQL. Для початківців цей процес може бути додатковою складністю, особливо якщо вони не мають досвіду налаштування серверів. Хоча процес налаштування не є надто складним для більш досвідчених користувачів, він може зайняти час та вимагає знань про серверні середовища, що може бути бар'єром для швидкого початку роботи з платформою.

4. Vulnhub (https://www.vulnhub.com/) – платформа, що надає різноманітні машини, що імітують різні сценарії атак та вразливостей.

Переваги:

1. Повністю безкоштовна платформа – Vulnhub забезпечує користувачів вільним доступом до великої кількості вразливих машин без жодних фінансових обмежень. Це робить платформу доступною для широкого кола користувачів, включаючи студентів, ентузіастів та професіоналів, які прагнуть удосконалювати свої навички без додаткових витрат. Повна безкоштовність платформи є значною перевагою, оскільки дозволяє користувачам навчатися в своєму темпі і без необхідності вкладень у навчальні матеріали.

2. Розгортання персоналізованих машин у власній системі – Vulnhub дозволяє завантажувати і розгортати вразливі машини безпосередньо на локальних гіпервізорах користувачів. Це забезпечує додатковий контроль над середовищем навчання, даючи змогу налаштовувати інфраструктуру за власним бажанням. Розгортання на локальному обладнанні також дозволяє користувачам створювати та зберігати власні образи віртуальних машин, працюючи з ними у власних умовах і при цьому не ризикуючи безпекою основної мережі.

3. Можливість впровадження у платформу користувацьких образів машин – Vulnhub підтримує можливість завантаження та обміну власними образами віртуальних машин, що дозволяє спільноті ділитися своїми створеними сценаріями та завданнями. Це дає платформі значну перевагу в плані різноманітності контенту та

підтримки нових сценаріїв, адже користувачі можуть створювати і публікувати власні унікальні виклики. Такі можливості сприяють співпраці в межах спільноти, допомагаючи створювати більш комплексні завдання та адаптувати їх до потреб користувачів із різним рівнем підготовки.

4. Широкий спектр різноманітних вразливих машин – Vulnhub пропонує великий вибір віртуальних машин, що містять уразливості різних типів, включаючи старі та сучасні сценарії атак. Це дозволяє користувачам навчатися різним технікам атаки та експлуатації вразливостей, охоплюючи широкий спектр методів захисту інформації. Різноманітність машин допомагає покращити знання у сфері пентестингу та кібербезпеки, адже кожна з машин може мати індивідуальні особливості та складності, що дозволяє тренуватись на реалістичних прикладах.

Недоліки:

1. Наявність гіпервізору у системі для розгортання машин – для роботи з Vulnhub користувачам необхідно встановити гіпервізор (наприклад, VirtualBox чи VMware), що може стати бар’єром для деяких користувачів, особливо для новачків, які не мають досвіду з віртуалізацією. Налаштування гіпервізору також може вимагати певних технічних знань і часу на конфігурацію, що ускладнює початковий процес. Додатково, необхідність використовувати віртуальні машини на локальному обладнанні може створити обмеження для користувачів з низькою обчислювальною потужністю.

2. Мінімальна документація розгорнутих машин – Vulnhub не надає детальної документації для кожної віртуальної машини, і користувачі мають самостійно досліджувати їхні особливості та вразливості. Це ускладнює процес навчання, оскільки новачкам може бути складно зрозуміти призначення та способи вирішення задач без додаткових підказок. Відсутність супровідної інформації також може затримувати прогрес, особливо коли користувач стикається з новими або складними вразливостями, для яких необхідно провести ґрунтовне дослідження.

3. Обмежена підтримка – Vulnhub не має розвиненої служби підтримки чи активної команди розробників, що може обмежити можливості швидкого вирішення проблем або отримання порад від офіційної команди. У разі виникнення технічних

труднощів або непередбачених помилок користувачам, швидше за все, доведеться звертатися до спільноти або самостійно вирішувати питання. Обмежена підтримка може стати проблемою для користувачів, які потребують додаткових роз'яснень чи допомоги при вирішенні певних завдань на платформі.

Для аналізу даних платформ було використано наступні джерела інформації [3–5], а також власні враження від використання поданих платформ.

Аналізуючи описані вище інструменти було прийнято рішення про необхідність розробки власної персоналізованої платформи для навчання та тренування з кібербезпеки, яка забезпечить ізольований доступ користувача або груп користувачів до їх персональних ізольованих мереж та машин у них, оскільки така можливість або зовсім відсутня, або потребує платної підписки для її використання на розглянутих платформах.

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Мета та задачі дослідження

В роботі була поставлена мета розробити інформаційну технологію створення тренувальної платформи вправ з кібербезпеки для підвищення рівня фаховості та компетентності обізнаності осіб, що навчаються, з питань кібербезпеки. Для досягнення мети необхідно виконати наступні задачі:

1) обрати моделі для реалізації платформи:

2) розробити інтерфейс користувача;

3) розробити автоматизоване програмне створення та керування ізольованими віртуальними приватними мережами;

4) розробити повнофункціональне створення та додавання вразливих машин для створених ізольованих мереж;

5) реалізувати автоматизоване створення файлів підключення до віртуальних мереж;

6) реалізувати можливість підключення клієнтів до платформи з віддалених мереж;

7) реалізувати можливість перегляду результатів щодо виконаної користувачами роботи у системі.

2.2 Вибір засобів реалізації

Центральним засобом реалізації даного проєкту є гіпервізор, оскільки його роль у реалізації – це створення віртуальних машин та управління середовищем. Для виконання роботи було обрано популярний відкритий гіпервізор VirtualBox, оскільки його досить легко встановити на користувацьку операційну системи і даний інструмент дозволяє створювати та керувати віртуальними машинами на різних операційних системах [6].

У якості основної мови програмування обрано Java – це потужна мова програмування для створення комплексних додатків. Також у репозиторії Java є модуль для взаємодії з VirtualBox API. Роль у реалізації - розробка автоматизованих сценаріїв розгортання та інтерфейсу адміністратора системи. Додатково для реалізації

платформи застосовано скрипти Linux оболонки bash для автоматизації виконання команд на shell.

Для реалізації користувацького інтерфейсу додатку було обрано JavaFX – платформи для створення Java додатків з графічним інтерфейсом, оскільки JavaFX на відміну від інших місцями популярніших аналогічних платформ вважається кращою для сучасних програм завдяки своїм розширеним функціям, таким як компоненти інтерфейсу користувача, стилі CSS, FXML для декларативного дизайну інтерфейсу користувача та краща підтримка анімації [7]. Також для автоматизації розташування елементів JavaFX був використаний інструмент для графічного створення інтерфейсів користувача для JavaFX додатків – Scene Builder.

У якості програми для забезпечення VPN з'єднань між сервером, мережею вразливих машин та віддаленими пристроями застосована технологія OpenVPN – це відкрите VPN-рішення, яке може розгортатися на широкому спектрі платформ, включаючи поширені операційні системи та публічні хмарні платформи. Для криптографії OpenVPN використовує OpenSSL, що забезпечує підтримку TLS тим самим це дозволяє OpenVPN інтегруватися з багатьма існуючими системами безпеки [8]. Також обране рішення є зручним та гнучким для налаштування (зокрема для створення повноцінних віртуальних приватних мереж для взаємодії хостів між собою, що і є першочерговою задачею інструменту) [9].

Оскільки, робота передбачає створення ізольованих віртуальних мереж для клієнтів, то існує два рішення для її побудови – створення однієї "глобальної" vpn мережі, а потім її сегментація за допомогою міжмережевого екрану, або створення окремих мереж та їх об'єднання до одного локального порту машини. У роботі був обраний другий метод вирішення завдання задля спрощення таблиць з правилами брандмауеру та забезпечення максимальної ізольованості створених мереж.

Створені мережі зводяться до одного інтерфейсу за допомогою nginx –це вебсервер, який також можна використовувати як зворотний проксі-сервер, поштовий проксі та кеш HTTP, який у даній програмі виступає як зворотній проксі, який приймає вхідні з'єднання та передає їх до цільових мереж. Також варто відмітити, що за допомогою цього існує можливість відкрити лише один порт у інтернеті, щоб

користувачі мали доступ до усіх створених на платформі мереж. Дана особливість стане в нагоді, оскільки для запобігання ручного додавання деяких правил для доступу з Інтернету до локальної мережі існує можливість застосування інструменту для тунелювання трафіку, за допомогою якого відбудеться зручне надання доступу до цільового локального сервісу з Інтернету.

Тунелювання трафіку з локальної мережі до зовнішнього сервера забезпечується за допомогою сервісу ngrok – це вебсервіс тунелювання, який дозволяє встановлювати безпечний зовнішній доступ до локальних вебсерверів та локальних розробок. Він працює як проксі-сервер між локальним сервером та Інтернетом, дозволяючи зовнішнім користувачам отримувати доступ до локального вебдодатку через Інтернет. Ngrok забезпечує публічну URL-адресу, яка відображається на зовнішньому Інтернеті та перенаправляє запити на локальний сервер, де запущений додаток. Це дозволяє легко демонструвати, тестувати та спілкуватися з додатками, які виконуються локально на комп'ютері розробника або в локальній мережі [10].

3 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ОБ'ЄКТА ДОСЛІДЖЕННЯ

3.1 Структурно-функціональне проєктування

IDEF0 діаграма є потужним інструментом для опису процесів і систем, що використовує структурований підхід для представлення функціональних аспектів будь-якої організації чи системи. Вона складається з блоків діяльності, які взаємодіють через стрілки, що представляють різні види інформації (вхід, вихід, контроль і механізм).

IDEF0 використовує кілька основних компонентів для побудови моделей:

1) функція/процес (Function name) – це сутність, що необхідно для реалізації процесу;

2) вхідні дані (Inputs) – це дані або об'єкти, які перетворюються або споживаються функцією для створення вихідних даних;

3) вихідні дані (Outputs) – це дані або об'єкти, які функція створює як результат її виконання;

4) контроль (Controls) – це умови або параметри, що визначають правильність виконання функції;

5) механізми (Mechanism) – це ресурси або засоби, які підтримують виконання функції;

6) виклик (Call) – стрілки виклику дозволяють обмінюватися деталями між моделями або частинами однієї моделі.

Базова структура використання компонентів IDEF0 діаграми відображена на рисунку 3.1 [11].

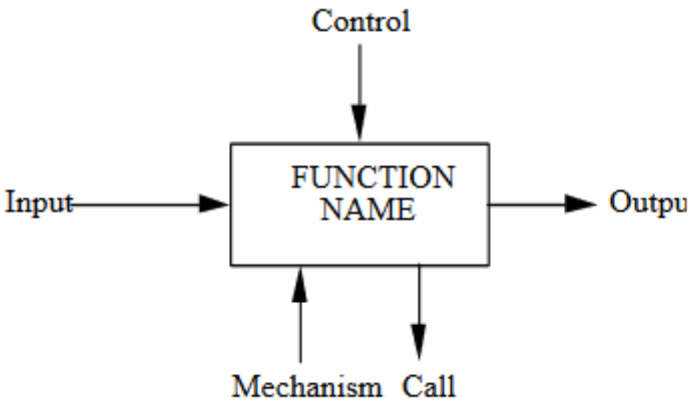


Рисунок 3.1 – Базовий компонент IDEF0 діаграми

Формат діаграми дозволяє створювати чітку та структуровану модель, яка може бути використана для аналізу, поліпшення або зміни системи, а також для управління її конфігурацією.

IDEF0 діаграма проєкту зображена на рисунку 3.2. Функцією для діаграми IDEF0 є безпосередньо розгортання та керування тренувальною платформою з елементами персоналізації для вправ з кібербезпеки. Входами в об’єкт є .json файли, які включають інформацію про головний сервер, що забезпечує віддалене підключення, а саме файл конфігурації vpn сервера та середовища для роботи з Virtual Box, файл, який зберігає інформацію про робочі машини середовища та файл для збереження образів. впровадженими в систему ova образами для розгортання машин та безпосередньо поточними розгорнутими мережами та машинами. До механізмів контролю відносяться програмне й апаратне забезпечення та відповідний користувач (адміністратор системи). Керується процес за допомогою Virtual Box API та VirtualBox Manager. На виході роботи процесу отримуємо робочий сервер (у якості однієї з машин гіпервізора), розгорнуті машини у гіпервізорі та файли підключення до створених на сервері мереж OpenVPN.

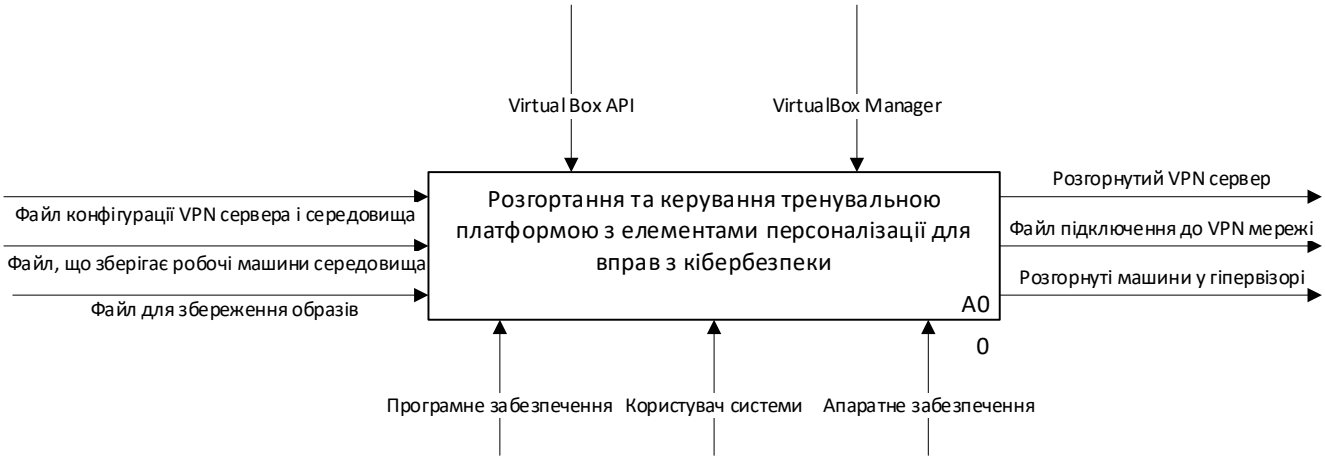


Рисунок 3.2 – IDEF0 діаграма проєкту

В декомпозиції нижнього рівня діаграми (рис 3.3) було розбито основний процес проєкту на п’ять різних підпроцесів, а саме:

1) розгортання та керування робочим сервером (запуск і конфігурація серверу та середовища Virtual Box);

2) створення та керування окремої віртуальної приватної мережі;

3) додавання образу машини до робочого простору програми;

4) додавання окремої машини та керування нею;

5) отримання окремого файлу підключення до віртуальної приватної мережі;

6) отримання результатів щодо виконаної користувачами роботи у системі.

Рисунок 3.3 відображає взаємодію цільових підпроцесів у системі. Рисунки опису кожного з підпроцесів окремо представлено на рисунках 3.4-3.9.

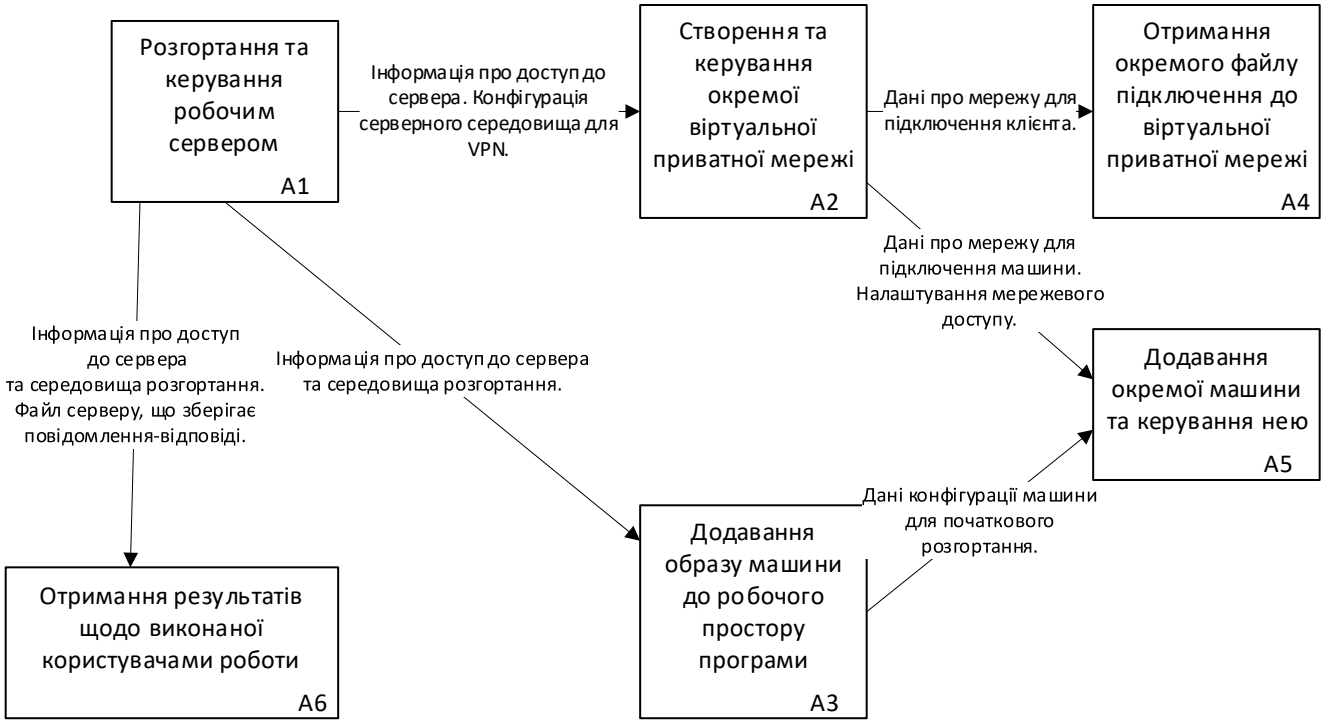


Рисунок 3.3 – Взаємодія цільових підпроцесів у системі

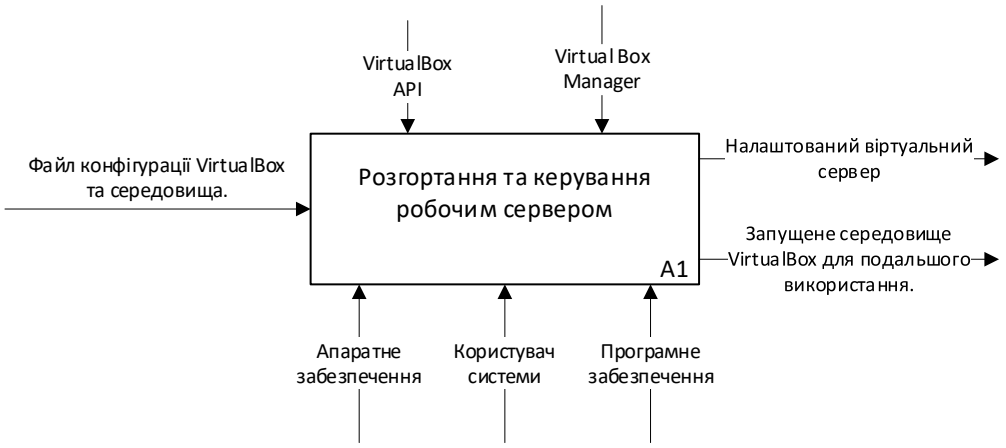


Рисунок 3.4 – IDEF0 діаграма для відображення підпроцесу розгортання та керування робочим сервером

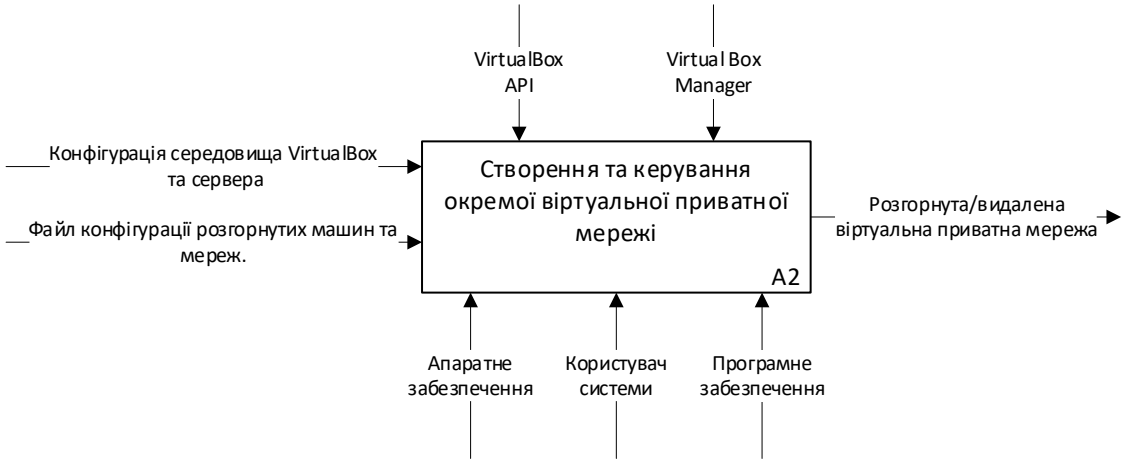


Рисунок 3.5 – IDEF0 діаграма для відображення підпроцесу створення та керування окремої віртуальної приватної мережі

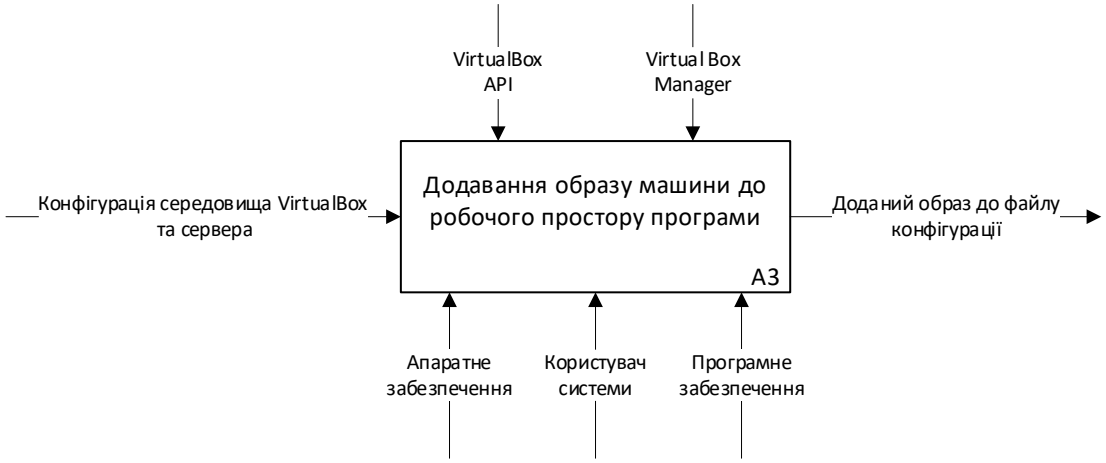


Рисунок 3.6 – IDEF0 діаграма для відображення підпроцесу додавання образу машини до робочого простору програми

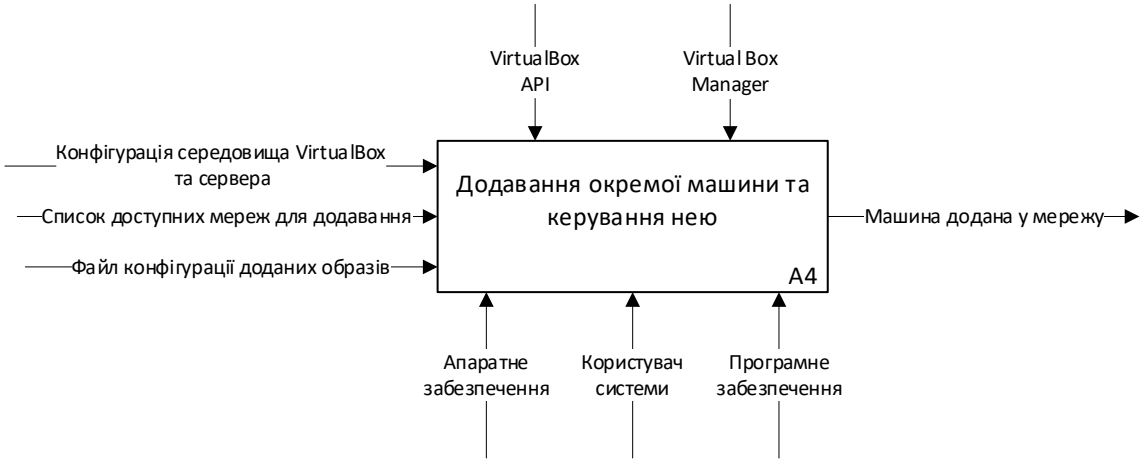


Рисунок 3.7 – IDEF0 діаграма для відображення підпроцесу додавання окремої машини та керування нею

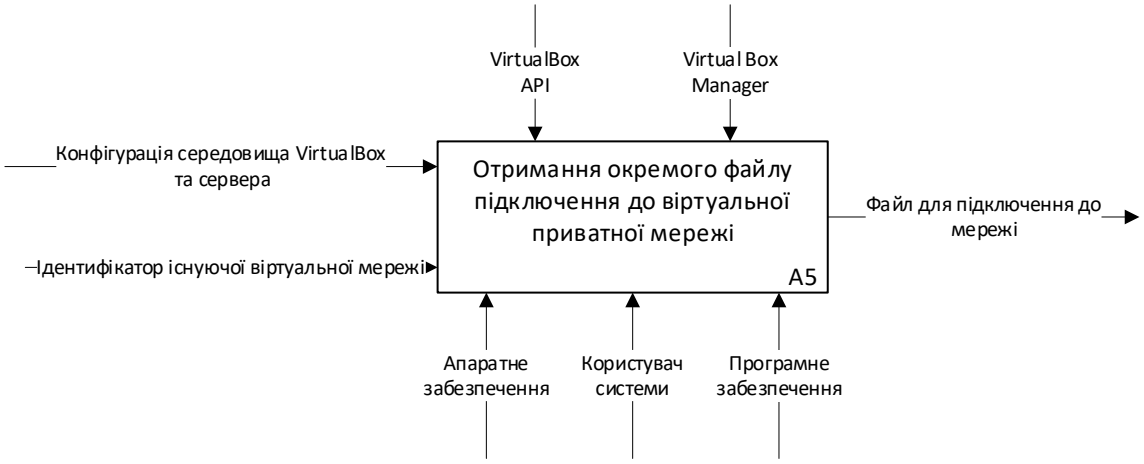


Рисунок 3.8 – IDEF0 діаграма для відображення підпроцесу отримання окремого файлу підключення до віртуальної приватної мережі

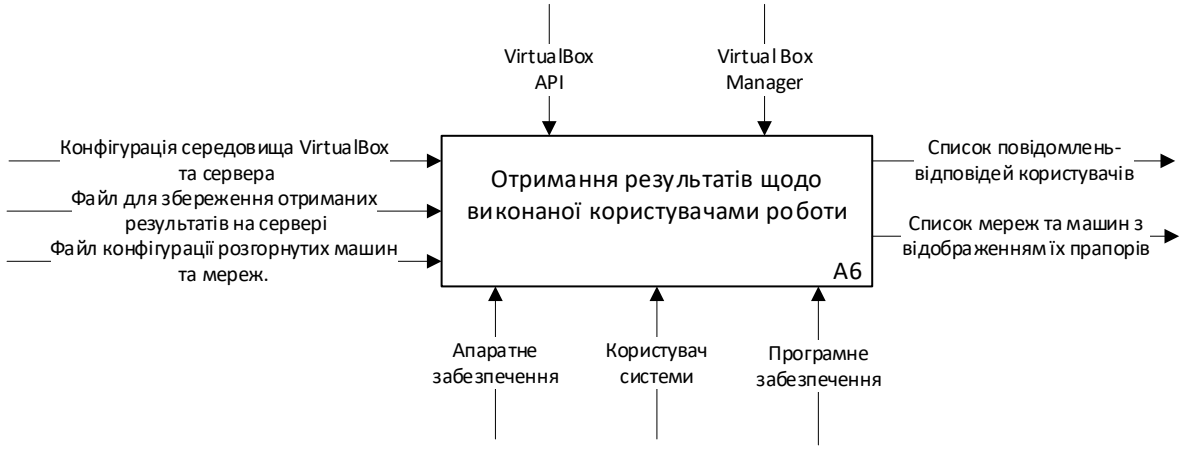


Рисунок 3.9 – IDEF0 діаграма для відображення підпроцесу відображення результатів щодо виконаної користувачами роботи у системі

3.2 Моделювання варіантів використання

UML (Unified Modeling Language) — уніфікована мова моделювання, яка дозволяє розробникам та замовникам переглядати програмну систему з різних сторін та на різних рівнях абстракції. Одна з основних причин, чому UML стала стандартом у моделюванні, полягає в її незалежності від мови програмування, що робить її універсальною і застосовною в будь-яких проєктах.

UML розділяється на два основні типи діаграм: структурні та поведінкові. Структурні діаграми представляють статичну сторону системи та визначають компоненти, які повинні бути присутніми, що робить їх корисними для

документування архітектури. Поведінкові діаграми, навпаки, показують динамічну сторону системи — вони описують, що має відбуватися у процесі роботи системи, і є ключовими для розуміння функціональності програмного забезпечення. Таким чином, UML є не тільки потужним інструментом для проєктування, а й ефективним засобом для комунікації між технічними фахівцями та бізнесом.

Таким чином UML діаграми є важливим засобом моделювання та візуалізації структури та поведінки програми. Вони допомагають порозумітися всім учасникам процесу створення програмного продукту від стейкхолдерів до розробників. Для більш ефективного розуміння процесів, що відбуваються в розробленому програмному продукті на рисунку 3.10 відображена створена UML діаграма [12].

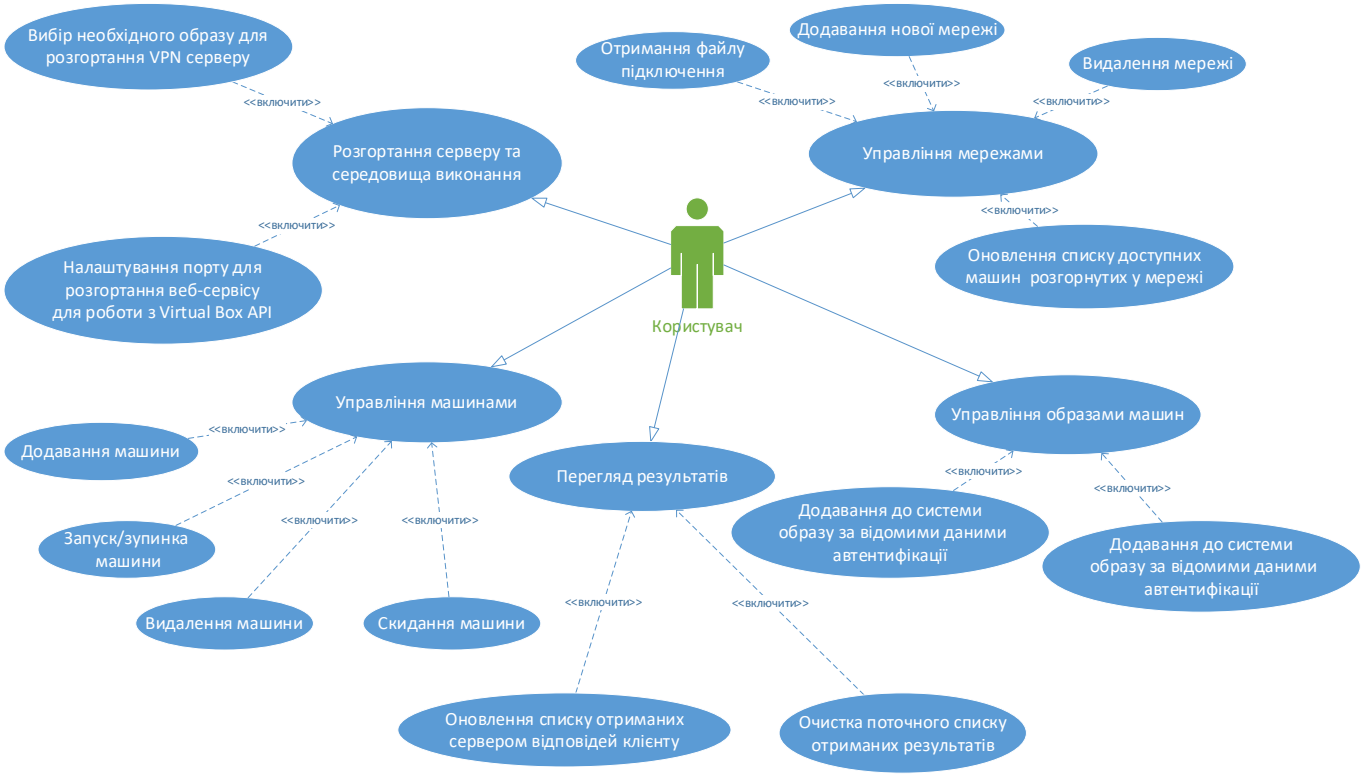


Рисунок 3.10 – UML діаграма проєкту

3.3 Розробка моделей зберігання даних

При виконанні завдання було прийнято рішення не застосовувати бази даних з цілю простішого розгортання даної платформи на персональному комп’ютері. Замість цього сутності створені під час роботи додатку будуть зберігатися у файлах формату json з цілю легкого доступу до них та їх легкого редагування/копіювання за потреби.

Якщо ж говорити безпосередньо про цільові сутності проєкту, то вони наступні:

1) сервер віртуальних мереж – центральний елемент системи, за його допомогою розгортаються цільові віртуальні мережі та до нього підключаються клієнти та машини з якими вони працюють;

2) мережа – елемент системи, який відповідає створеній на платформі мережі, мережа має включати список машин, що підключені до неї і з якими працюють клієнти;

3) образ машини – елемент системи, що необхідно обрати як основу для створеної машини, включає файл для імпорту стартової системи машини;

4) машина – елемент системи, що відповідає створеній машині на платформі у певній мережі, містить у своєму складі власні згенеровані прапори для підтвердження її проходження.

Нижче наведена таблиця, яка включає сутність та опис її окремих властивостей.

Таблиця 3.1 Властивості сутностей системи

Сутність	**Властивість**	**Зміст властивості**
Сервер	vboxFolderPath	Шлях до встановленого у системі Virtual Box
	serverName	Назва машини у системі Virtual Box
	serverPath	Шлях до віртуального жорсткого диску (.vdi) серверу
	deployManagerPort	Порт для розгортання Virtual Box Web Server для використання API
Мережа	networkName	Ім’я мережі на платформі
	networkID	Унікальний ідентифікатор мережі у системі
	networkIPRange	Діапазон адрес, які призначає клієнтам задана мережа
	localNetworkIPAddr	Локальна адреса до якої “прив’язується” конкретна віртуальна приватна мережа
	uploadMachineItems	Список, який включає підключені до мережі вразливі машини

Продовження Таблиці 3.1

Машина	machineName	Ідентифікатор машини у системі
	machineMachineFilePath	Шлях до директорії, де зберігаються ресурси розгорнутої машини
	md5UserFlag	Згенерований користувацький прапор машини
	md5RootFlag	Згенерований системний прапор машини
	machineNetworkId	Ідентифікатор мережі у якому запущена машина
	machineImageItem	Образ для запуску машини
Образ машини	machineImageName	Назва доданого до платформи образу машини
	machineImagePath	Шлях до доданого образу
	machineUsername	Логін привілейованого sudo користувача
	machinePassword	Пароль привілейованого sudo користувача
	machineUserFlagFilePath	Шлях у системі машини, де зберігається користувацький прапор
	machineRootFlagFilePath	Шлях у системі машини, де зберігається прапор привілейованого користувача

Представлення даних сутностей у файлах відображене у наступному розділі даної роботи.

4 РОЗРОБКА ПЕРСОНАЛІЗОВАНОЇ ПЛАТФОРМИ ДЛЯ НАВЧАННЯ ТА ТРЕНУВАННЯ З КІБЕРБЕЗПЕКИ

4.1 Реалізація користувацького інтерфейсу

Перед реалізацією всіх необхідних функцій для роботи платформи було створено користувацький інтерфейс для роботи адміністратора системи з усім необхідним функціоналом, а саме: вікно головного меню програми (рис 4.1), спливаючих вікон для додавання мереж(рис 4.2), машин (рис 4.3), образів машин (рис 4.4), інструкцій про порядок дій при завантаженні не готового до експлуатації образу машин (рис 4.5), налаштування серверу та середовища його роботи (рис. 4.6), встановлення та оновлення ключів для роботи служби ngrok (рис. 4.7) та перегляду результатів (рис. 4.8) [13,14].

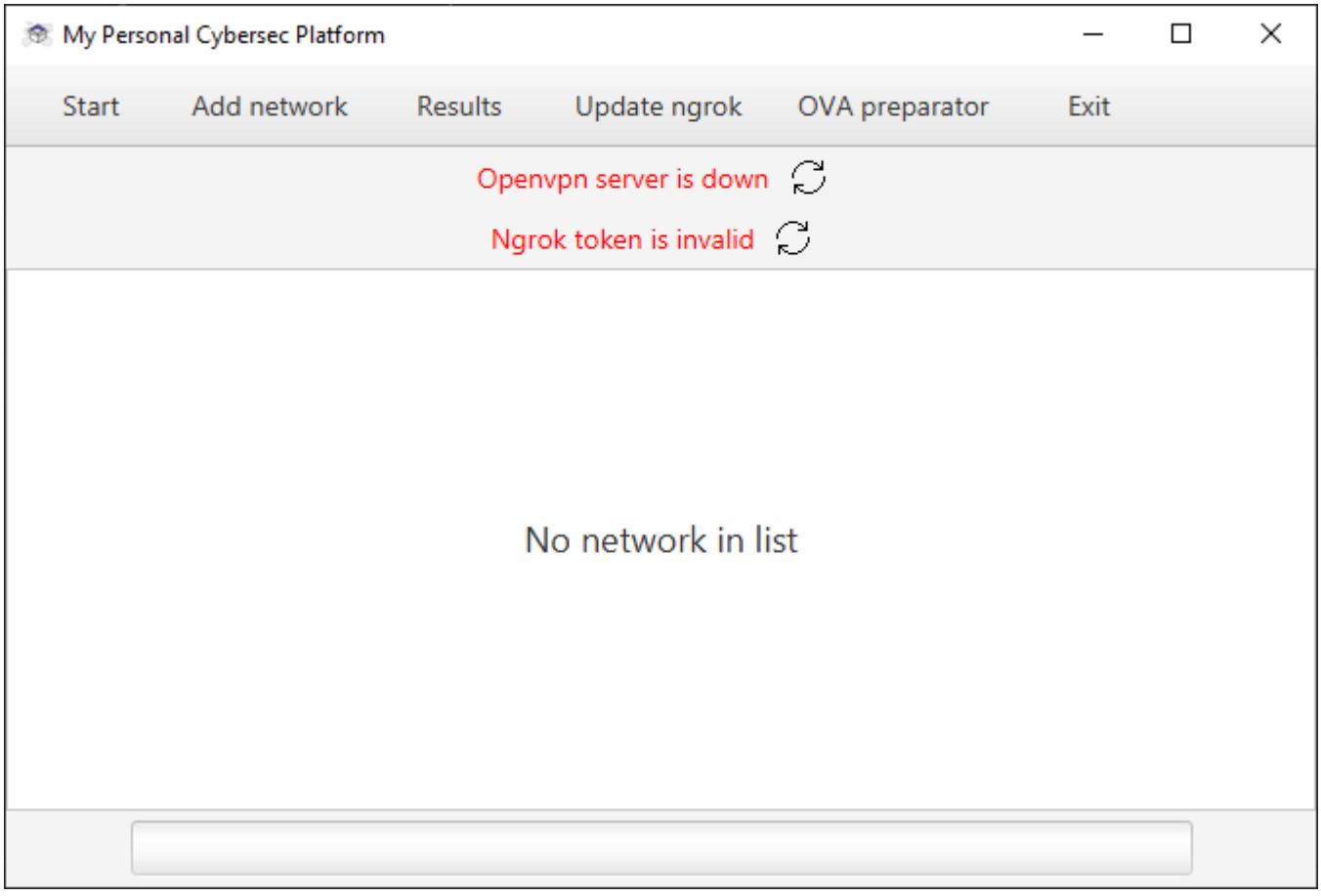


Рисунок 4.1 – Головне меню програми

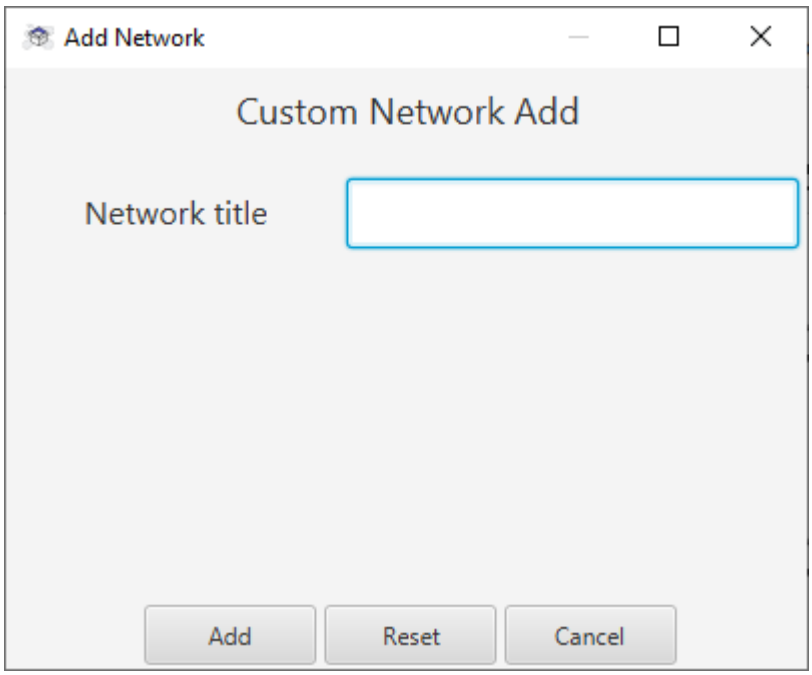


Рисунок 4.2 – Модальне вікно для додавання мережі

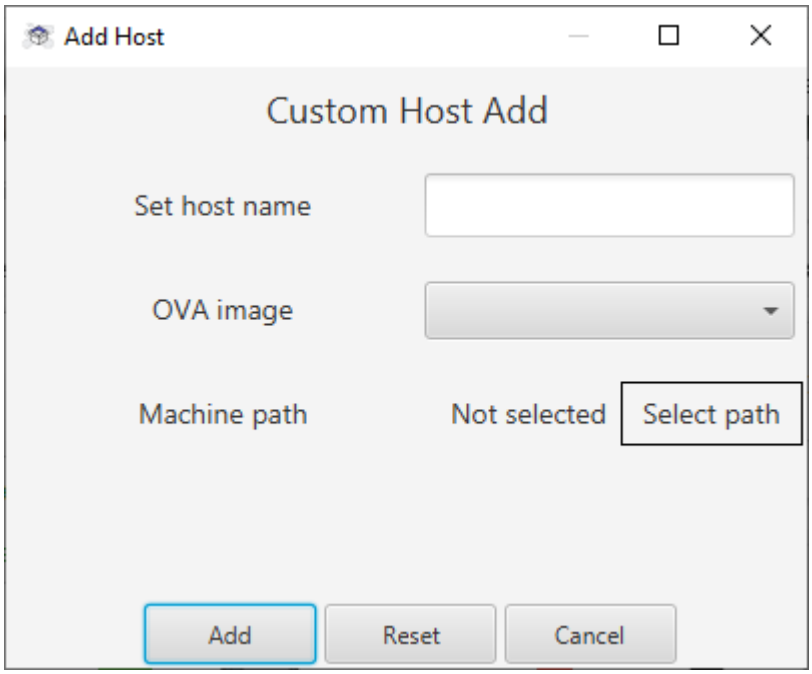


Рисунок 4.3 – Модальне вікно для додавання машини до мережі

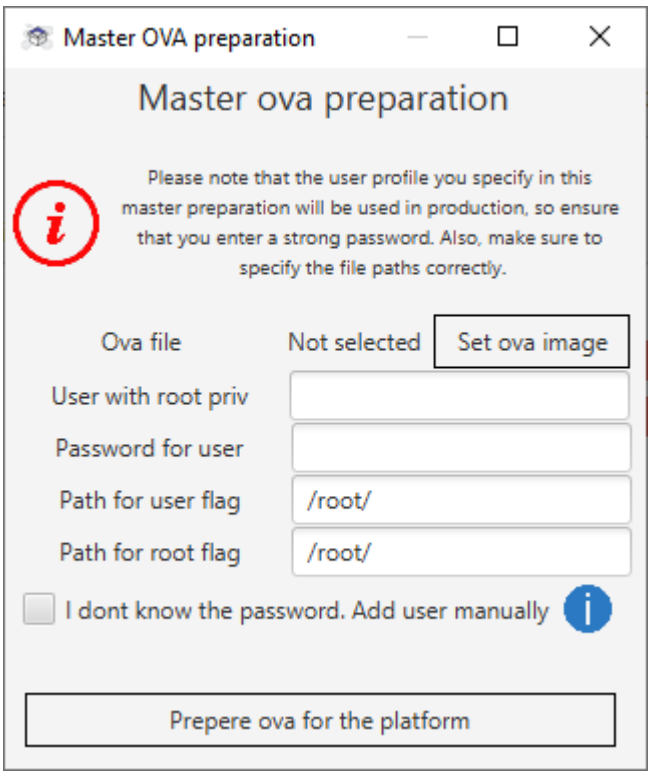


Рисунок 4.4 – Модальне вікно для додавання образу машини

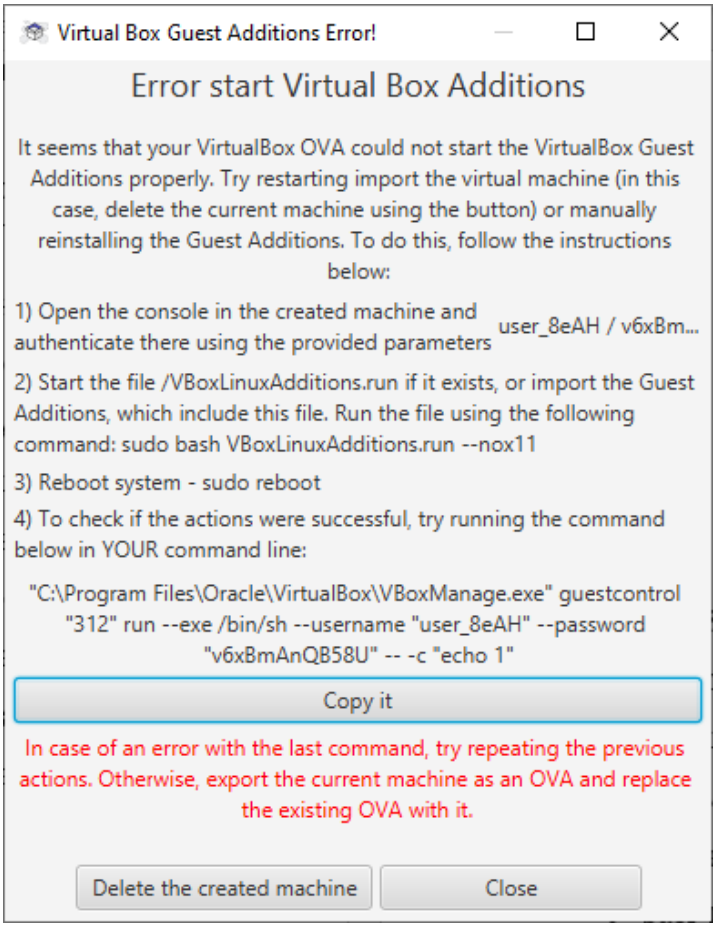


Рисунок 4.5 – Інструкція про порядок дій при завантаженні не готового до експлуатації образу машин

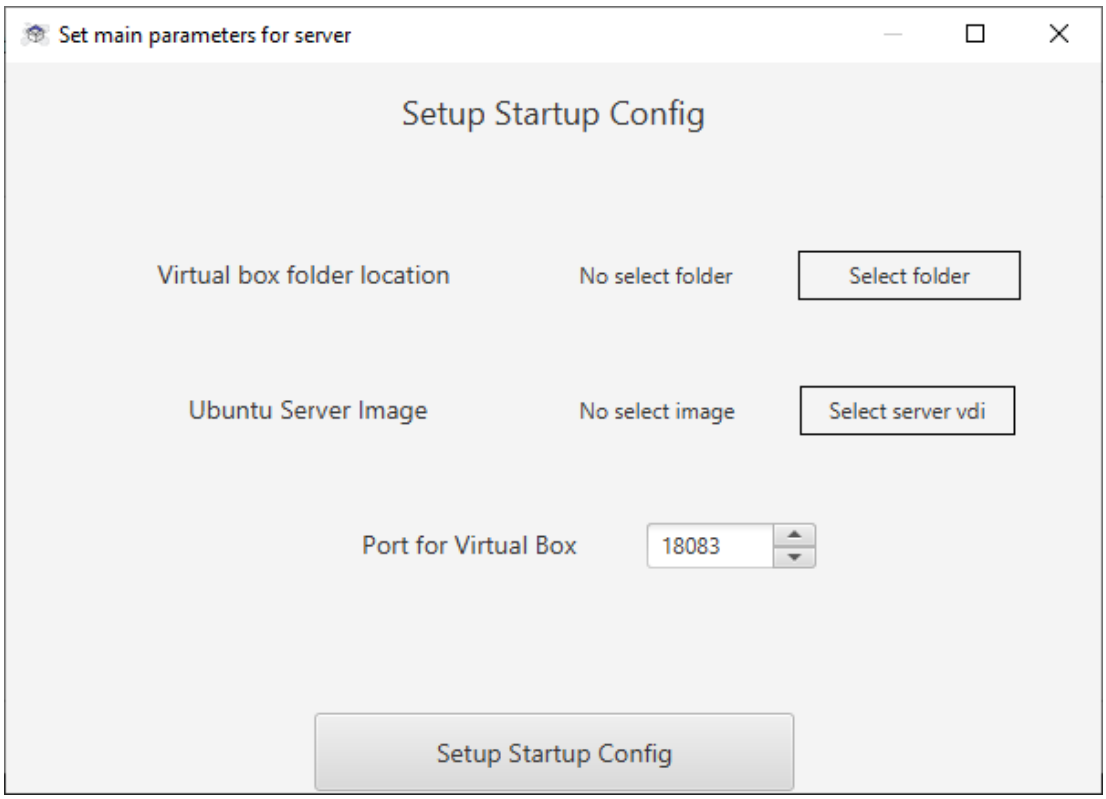


Рисунок 4.6 – Налаштування серверу та середовища його роботи

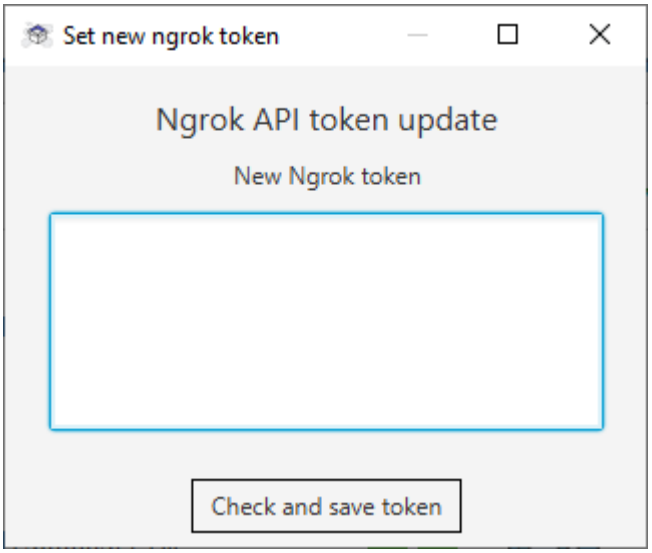


Рисунок 4.7 – Модальне вікно для функції додавання/оновлення токену ngrok

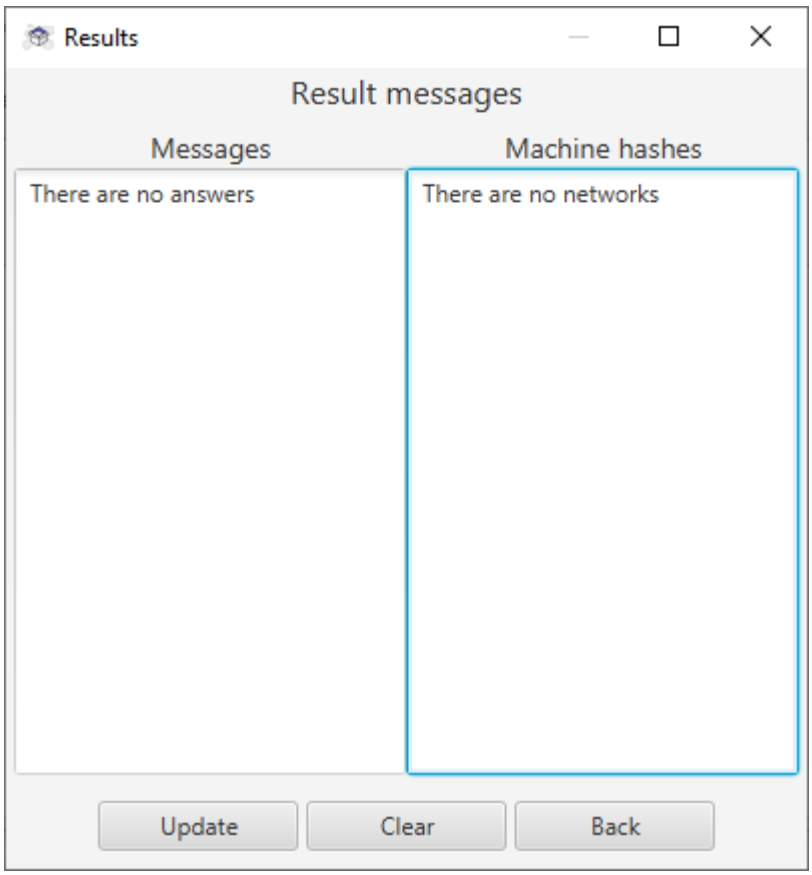


Рисунок 4.8 – Модальне вікно для функції перегляду результатів відправлених підключеними до мереж віддаленими клієнтами

4.2 Реалізація та особливості роботи системи запуску середовища виконання та робочого серверу

Для коректної роботи платформи при першому запуску необхідно натиснути на кнопку запуску серверу. Після натискання на дану кнопку можна побачити меню, де адміністратору платформи пропонується ввести папку, де розташований Virtual Box, образ машини та порт для розгортання сервісу на якому буде працювати Virtual Box Web Server. Важливо відмітити, що на даний момент платформа працює лише з образами сайту osboxes.org.

При правильному введенні усіх необхідних опцій, після першого запуску платформи при якому відбудеться налаштування необхідних програмних сервісів, інформація про конфігурацію буде збережена до файлу з назвою main_config.json. Який буде мати вигляд як на рисунку 4.9.

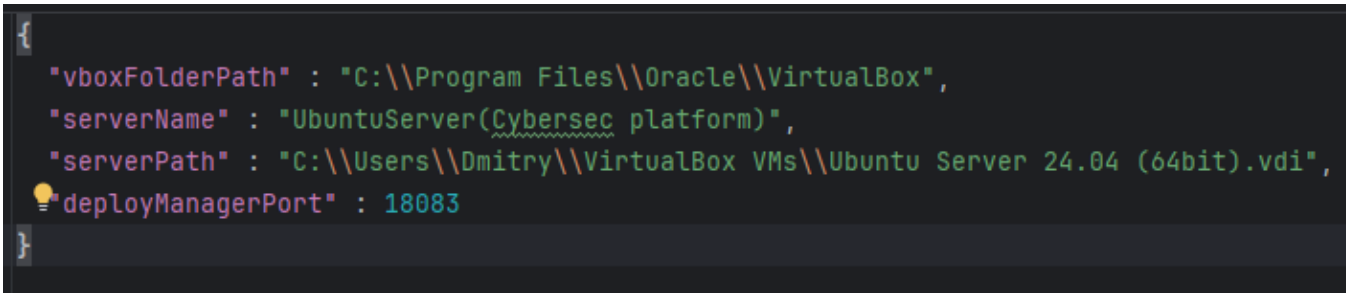


Рисунок 4.9 – Файл конфігурації середовища

У разі відсутності цього файлу, а також машини зі стандартною назвою UbuntuServer(Cybersec platform) дану процедуру можна буде переініціалізувати тим самим способом.

У інших випадках запуск і зупинка серверу відбуваються стандартними методами Virtual Box в залежності від стану машини. Якщо машина запущена є опція її зупинки (рисунок 4.10), якщо ні – її запуску (рисунок 4.11).

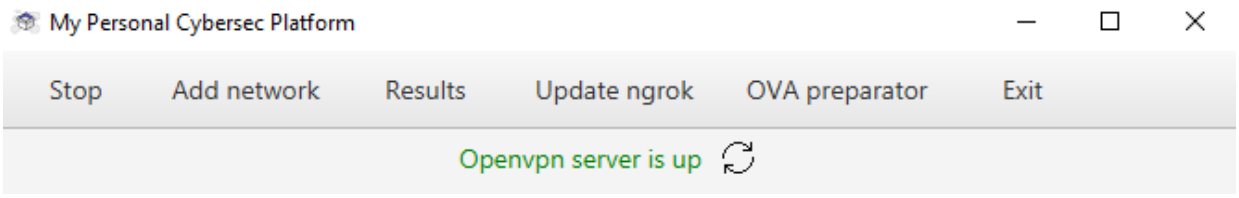


Рисунок 4.10 – Опція зупинки, коли сервер запущений

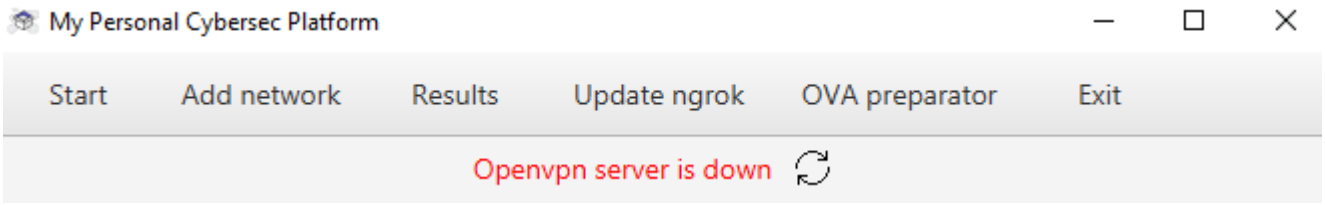


Рисунок 4.11 – Опція запуску коли сервер не запущений

Також бажано періодично перевіряти стан серверу за допомогою кнопки оновлення з правої сторони від напису, оскільки сервер можна вимкнути вручну за допомогою додатку Virtual Box.

Врато відмітити, що процес ввімкнення проходить не одразу, а, оскільки, сервер є центральним елементом платформи необхідно дочекатися його повного ввімкнення. Дана процедура виконується періодичним “пінгуванням” серверу echo запитами від VirtualBox Manger доти доки сервер не відповість клієнту. Поки відбувається дана процедура клієнт бачить вікно очікування (рисунок 4.12).

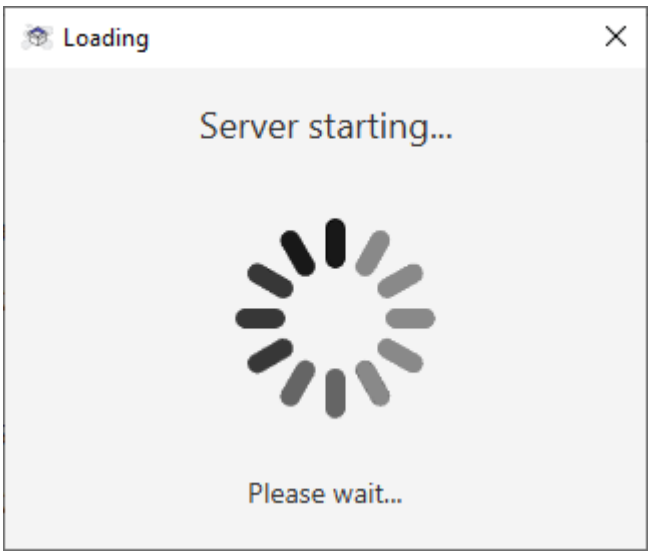


Рисунок 4.12 – Вікно очікування запуску сервера

Для першого запуску серверу необхідно встановити деякі налаштування на сервері даний процес проходить автоматизовано за допомогою запуску скриптів описаних у таблиці 4.1.

Таблиця 4.1 Робочі скрипти для завантаження необхідних служб для роботи платформи

Назва скрипту	**Роль у функціонуванні системи**
nginx-install.sh	Встановлення nginx як зворотного проксі для можливості підключення до декількох ovpenvpn сервісів використовуючи один сокет.
nginx-config.sh	Оскільки, стандартна версія nginx не підтримує потоки збірка відбувається через вихідні файли. Даний скрипт забезпечує створення системної служби з встановлених у попередньому скрипті файлів запуску. Також даний скрипт додає необхідні для функціонування платформи налаштування у конфігураційний файл nginx.
ngrok-install.sh	Скрипт для встановлення ngrok до цільової системи.

Продовження Таблиці 4.1

ngrok-config.sh	Скрипт для запуску системної служби ngrok з необхідними параметрами тунелювання.
iptables-setup.sh	Скрипт для встановлення та налаштування iptables – брандмауеру, який забезпечує відкритість портів необхідних для роботи та заборону до підключення інших
network-local-addr.sh	Скрипт для додавання локального інтерфейсу внутрішньої мережі платформи.
dhcp-server-install.sh	Скрипт для встановлення служби для роздачі динамічних локальних адрес машинам у внутрішній мережі.
auto-sudo.sh	Скрипт, що додає можливість виконання команд для користувача від імені адміністратора системи без введення паролю.
check-sudo.sh	Скрипт, що перевіряє можливість користувача вводити команду без введення паролю.

Разом з сервером запускається Virtual Box Web Server у паралельному потоці виконання програми. Даний компонент необхідний для взаємодії з Virtual Box API. При завершенні програми даний процес завершується.

Щодо особливостей реалізації екземпляру сервер у системі, то даний компонент системи має бути один, а також він має бути видимим для усіх його компонентів, тому для його реалізації був застосований Singleton Design Pattern. Це гарантує те, що у класу буде лише один екземпляр і надає глобальну точку доступу до нього, що повністю відповідає ідеї про повну доступність єдиного екземпляру серверу при виконанні роботи програми [15].

4.3 Реалізація процесу очікування інших елементів на запуск серверу

Як вже було згадано раніше сервер – центральна частина платформи від якої залежать інші елементи. Тож задля запобігання створення нових мереж, файлів підключення та машин необхідно створити умову для цих функціональних об’єктів завдяки якій вони не зможуть виконувати свої функції, якщо Virtual Box API або сервер не запущені.

Для цього стає в нагоді базова функція перевірки стану серверу. Відображення функції на рисунку 4.13:

```
11 usages
public static boolean isVMRunning() {
    String vmName = ServerItem.getInstance().getServerName();
    Integer port = ServerItem.getInstance().getDeployManagerPort();
    VirtualBoxManager mgr = VirtualBoxManager.createInstance( s: null);
    try {
        mgr.connect( s: "http://localhost:" + port, s1: null, s2: null);
        IVirtualBox vbox = mgr.getVBox();
        IMachine machine = vbox.findMachine(vmName);
        String state = machine.getState().toString();
        return state.equals("Running");
    } catch (VBoxException e) {
        return false;
    }
}
```

Рисунок 4.13 – Перевірка на те що сервер запущений

Якщо умова не виконується інші функції стають недоступними і виводиться інформаційне повідомлення про необхідність запуску робочого серверу та вебсервісу VirtualBox (рисунок 4.14).

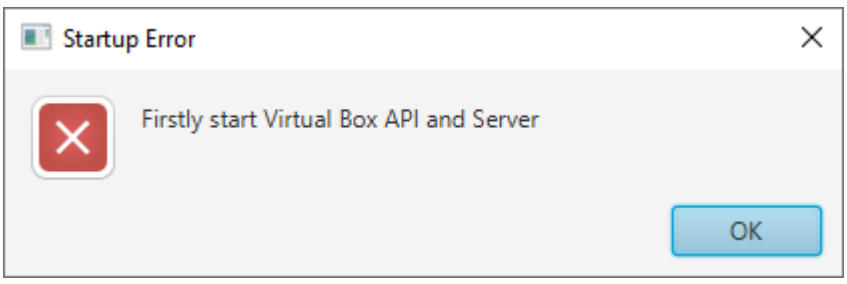


Рисунок 4.14 – Вікно для відображення повідомлення про необхідність запуску робочого серверу та вебсервісу VirtualBox

4.3 Реалізація функції керування окремими віртуальними мережами.

На платформі можна додавати і видаляти робочі мережі, також можна отримати згенерований файл для підключення до створеної мережі і оновити список активних хостів.

Розглянемо роботу кожної з описаних функцій окремо.

Для додавання мережі необхідно присвоїти їй певне ім’я, імена мереж можуть повторюватися, оскільки ідентифікатором мережі є рядок з назвою мережі та секундного еквіваленту поточного дати і часу. Перевірка на те що назва мережі не є порожнім значенням відбувається безпосередньо у модальному вікні створення (рисунок 4.15).

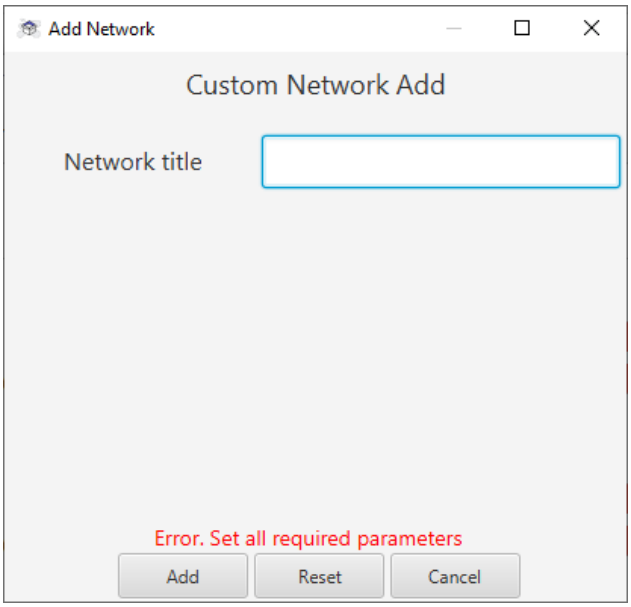


Рисунок 4.15 – Помилка при пустоті значення у полі назви мережі

Варто зазначити, що такі перевірки відбуваються на кожному кроці виконання програми. Усі вони залежать від специфіки роботи кожної з функцій та особливостей розгортання кожного окремого елемента платформи.

Додана мережа виглядає наступним чином без інформації про запущені на ній вразливі машини (рисунок 4.16) та з цією інформацією (рисунок 4.17):

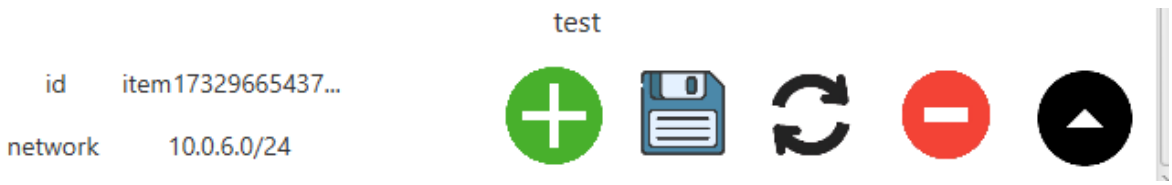


Рисунок 4.16 – Мережа без інформації про запущені вразливі машини на платформі

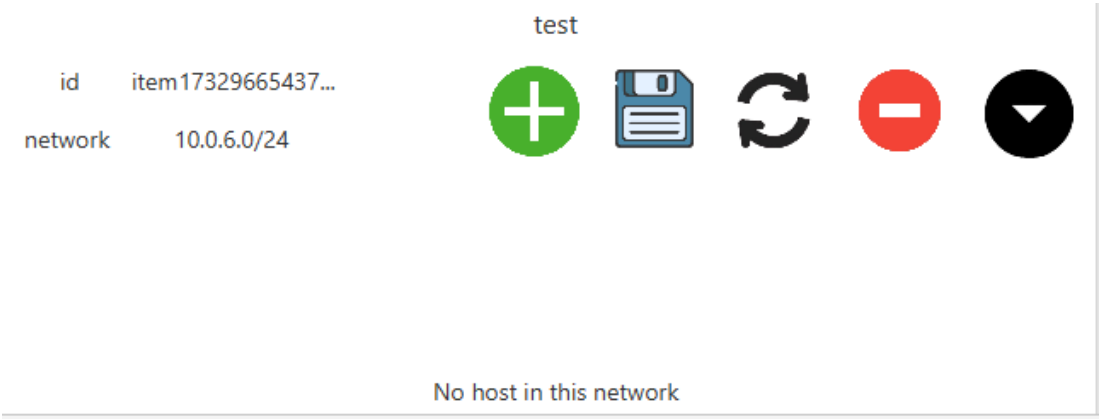


Рисунок 4.17 – Розгорнута інформація про мережу та про розгорнуті в ній вразливі машини

Підключення до даної мережі можливе за умови генерації необхідного файлу для підключення з введенням правильної кінцевої адреси ngrok у отриманому файлі. Кроки цього процесу продемонстровані на рисунках 4.18-4.22.

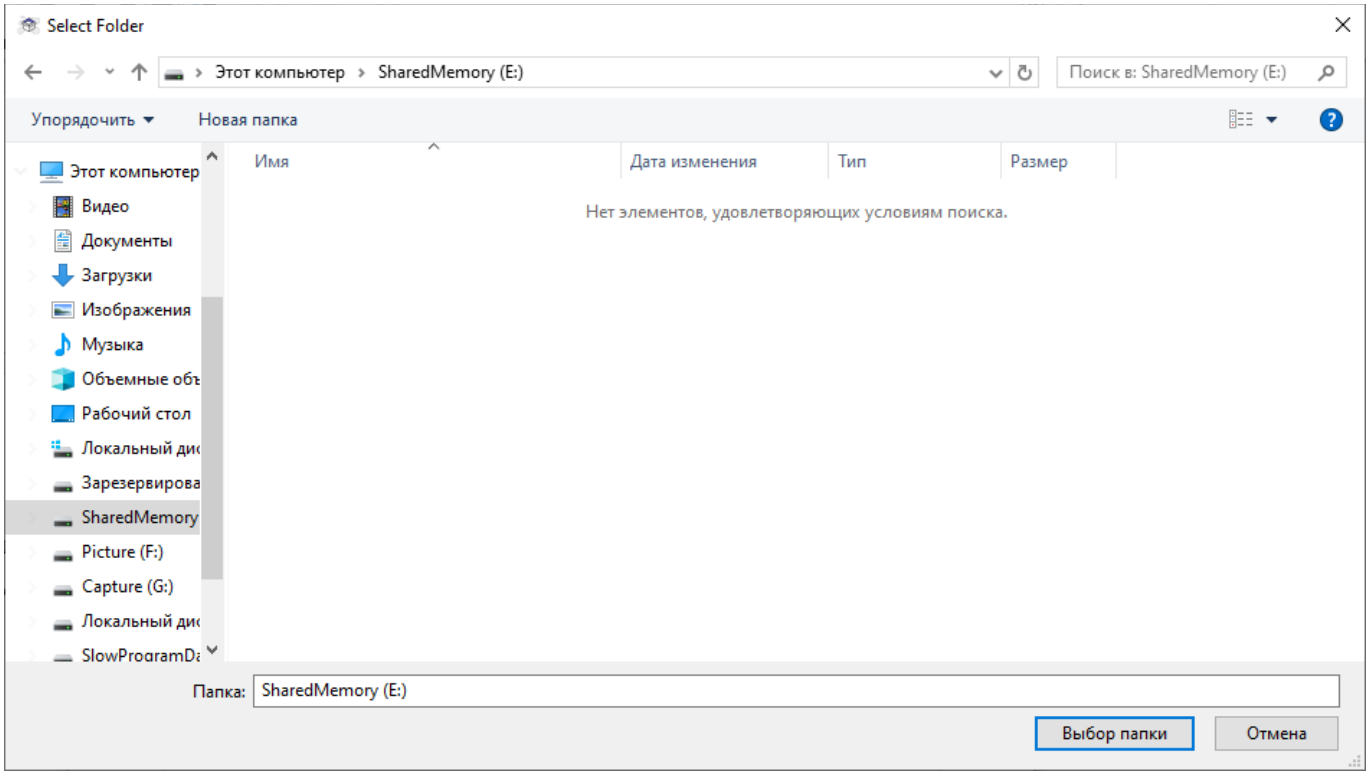


Рисунок 4.18 – Отримання файлу підключення до мережі openvpn

Рисунок 4.19 – Отримання ngrok адреси серверу з панелі керування адміністратора

```
client
dev tun
proto tcp
remote 2.tcp.eu.ngrok.io 13335
resolv-retry infinite
nobind
persist-key
persist-tun
remote-cert-tls server
auth SHA256
cipher AES-256-GCM
tls-auth ta.key 1
verb 3
```

Рисунок 4.20 – Редагування отриманого файлу

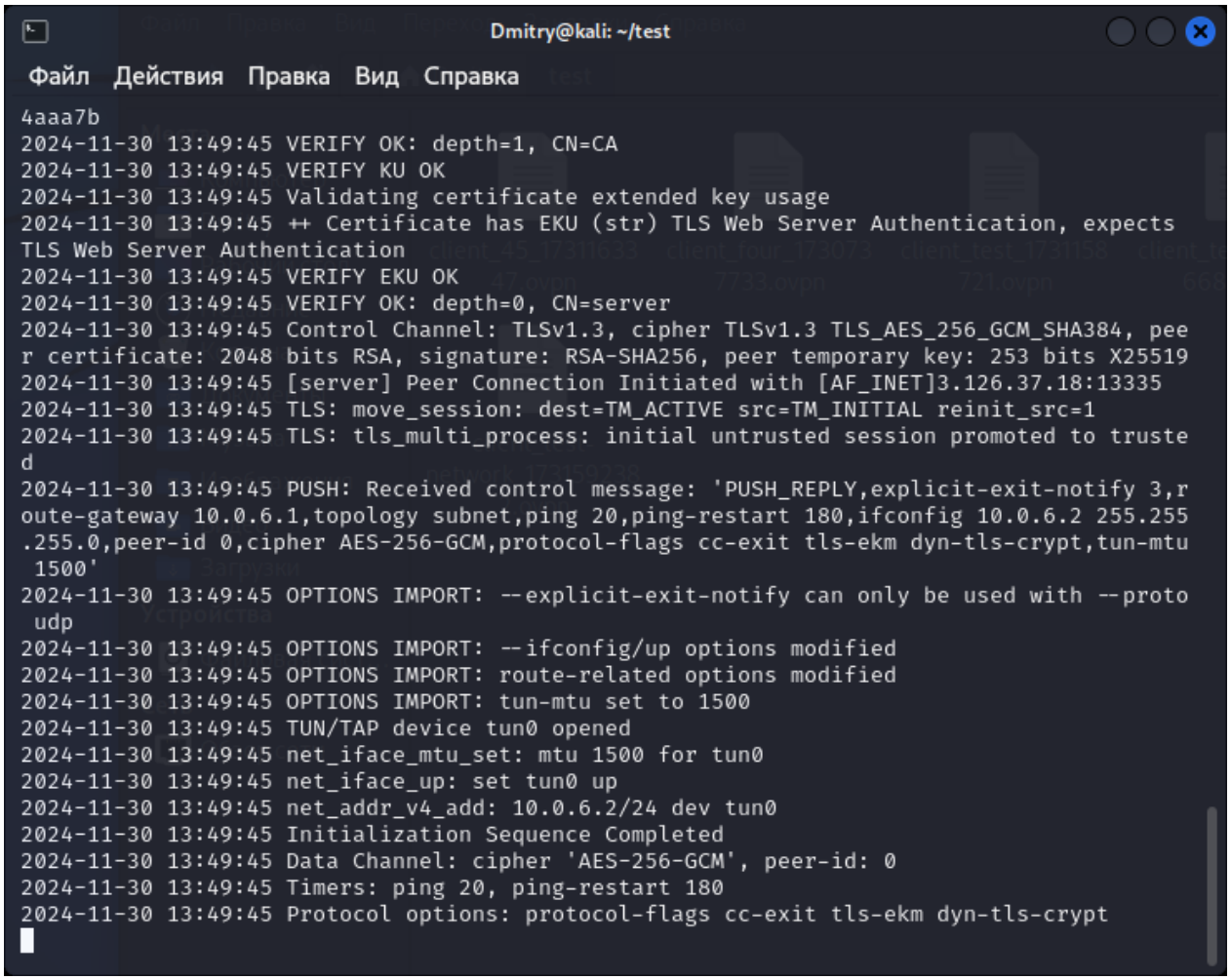


Рисунок 4.21 – Підключення до створеної мережі

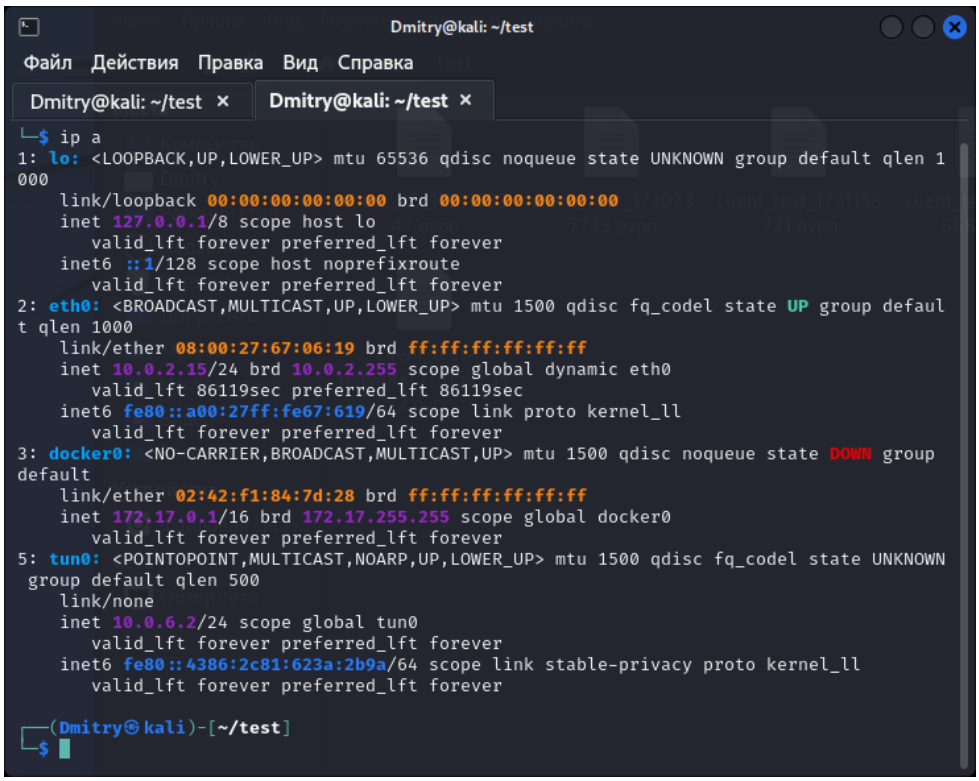


Рисунок 4.22 – Успішне підключення до створеної мережі клієнту системи

Видалення мережі відбувається за допомогою кнопки розташованої на панелі керування машиною. Видалення мережі та спроби підключення до видаленої мережі можна побачити на рисунках 4.23 та 4.24 відповідно.

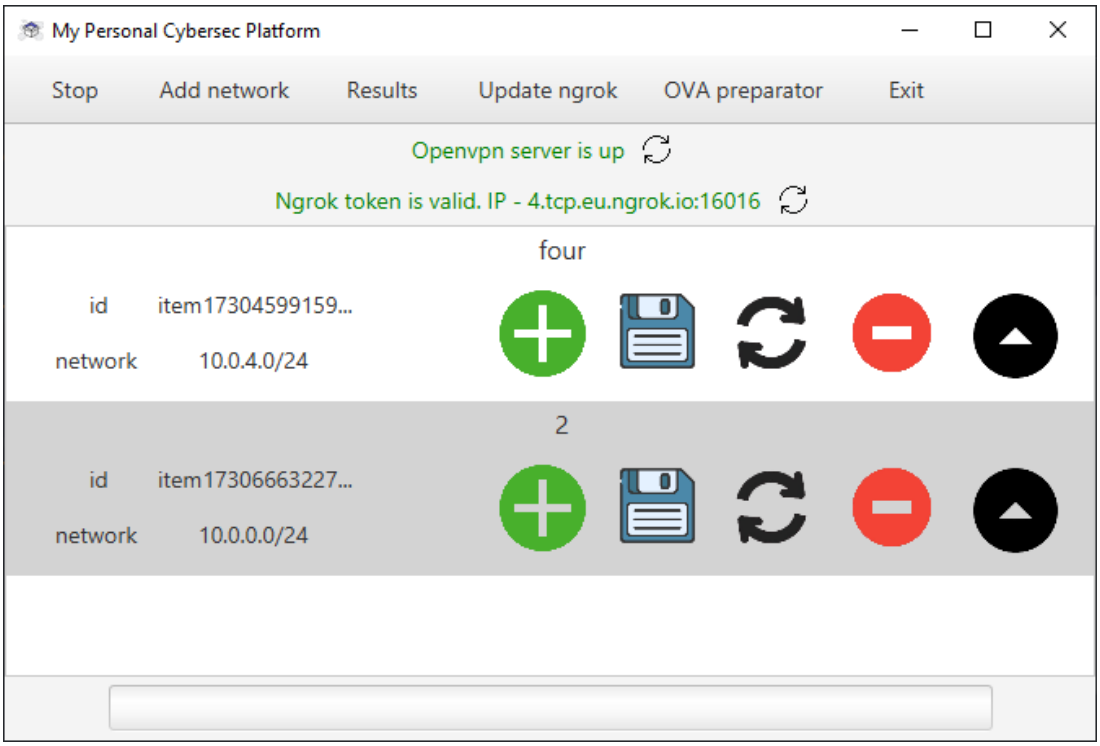


Рисунок 4.23 – Успішне видалення машини

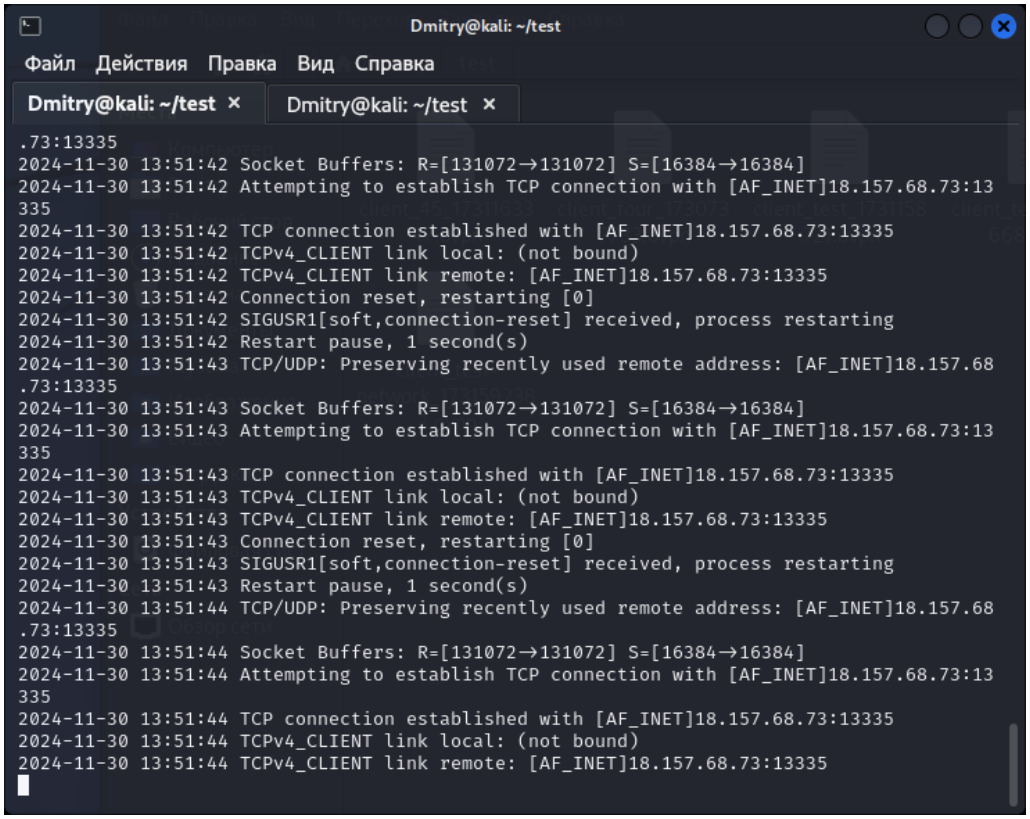


Рисунок 4.24 – Помилка при підключенні до видаленої машини після її видалення

Автоматизація взаємодії компонентів для керування мережею відбувається за допомогою написаних скриптів командної оболонки Linux, що виконується безпосередньо на робочому сервері. Робочі скрипти для керування мережею описані у таблиці 4.2.

Таблиця 4.2 Робочі скрипти для керування мережею

Назва скрипту	**Роль у функціонуванні системи**
create-network.sh	Створення мережі за ідентифікаторам мережі, адресу локальної мережі, маска локальної мережі та локальна адреса до якої закріплюється мережа.
delete-network.sh	Видалення мережі за її ідентифікатором та локальної адреси до якої закріплюється мережа.
get-available-ovpn-network.sh	Скрипт для отримання доступної адреси для VPN адреси.

Продовження Таблиці 4.2

get-local-available-ip.sh	Скрипт для отримання доступної локальної адреси.
get-openvpn.sh	Скрипт для генерації файлу підключення клієнта за ідентифікатором мережі, назвою користувача та ip сервера для підключення.

Основний скрипт – скрипт створення мережі відображений у Додатку А. Даний скрипт відображає процес створення віртуальної мережі, а саме файлів конфігурації безпосередньо самої мережі, а також середовища у якій вона розгортається.

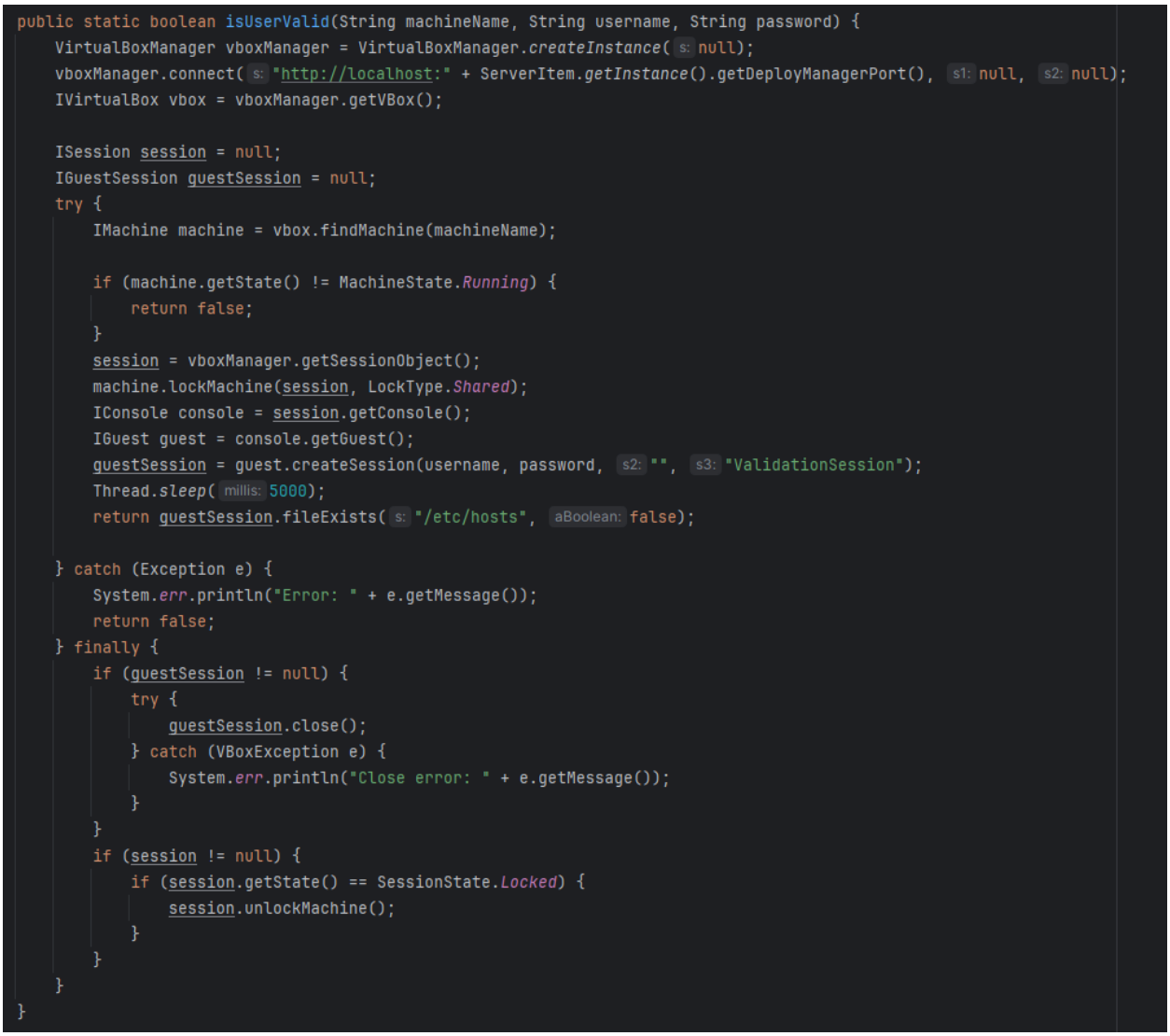
Для забезпечення незалежності мереж було використано наступний підхід – мережа, яка була створена під певним ідентифікатором у своєму файлі налаштувань “прив’язується” до певного сокету (комбінації ip адреси та порту). Для запобігання колізії та забезпечення одноманітності для відображення мереж було визначено, що адреса – це змінна величина, а порт статична, тобто при створенні мережі визначається вільна локальна адреса діапазону 127.0.0.0/8 на якій у комбінації з портом 1194 і розгортається служба створеної віртуальної мережі. Створені мережі як вже і було згадано раніше зводяться за допомогою nginx до однієї адреси і є доступними з Інтернету (за допомогою ngrok) та локальної внутрішньої мережі Virtual Box зі встановленими вразливими машинами [16–21].

4.4 Реалізація функції додавання образів машин до системи.

Модальне вікно додавання образів машин можна побачити на рисунку 4.3. Для додавання нового образу машини необхідно обрати цільовий образ у системі, вказати дані автентифікації для користувача, який має sudo права у системі образу, також необхідно вказати шляхи за якими будуть розміщені прапори для користувача та супер користувача (за замовченням вказаний /root/ шлях).

У процесі додавання відбувається перевірка даних автентифікації, якщо вони вірні, а також за умови встановленого у образі Virtual Box Guest Additions. Перевірка даних відбувається шляхом створення сесії у попередньо імпортованій машині образу

з введеними логіном та паролем та спроби перевірки за даними інснування стандартного файлу /etc/hosts стандартного текстового файлу Linux, що містить базу даних доменних імен і використовується при їх трансляції до мережних адрес вузлів. При успішній перевірці дані образу будуть додані до файлу volume_items.json. Рисунки 4.25 та 4.26 демонструють фрагменти коду для перевірки коректності даних автентифікації користувача та налаштування середовища для перевірки відповідно.



```
public static boolean isUserValid(String machineName, String username, String password) {
    VirtualBoxManager vboxManager = VirtualBoxManager.createInstance( s: null);
    vboxManager.connect( s: "http://localhost:" + ServerItem.getInstance().getDeployManagerPort(), s1: null, s2: null);
    IVirtualBox vbox = vboxManager.getVBox();

    ISession session = null;
    IGuestSession guestSession = null;
    try {
        IMachine machine = vbox.findMachine(machineName);

        if (machine.getState() != MachineState.Running) {
            return false;
        }
        session = vboxManager.getSessionObject();
        machine.lockMachine(session, LockType.Shared);
        IConsole console = session.getConsole();
        IGuest guest = console.getGuest();
        guestSession = guest.createSession(username, password, s2: "", s3: "ValidationSession");
        Thread.sleep( millis: 5000);
        return guestSession.fileExists( s: "/etc/hosts", aBoolean: false);

    } catch (Exception e) {
        System.err.println("Error: " + e.getMessage());
        return false;
    } finally {
        if (guestSession != null) {
            try {
                guestSession.close();
            } catch (VBoxException e) {
                System.err.println("Close error: " + e.getMessage());
            }
        }
        if (session != null) {
            if (session.getState() == SessionState.Locked) {
                session.unlockMachine();
            }
        }
    }
}
```

Рисунок 4.25 – Функція для перевірки даних автентифікації користувача

```java
if (importProgress.getCompleted()) {
    System.out.println("Machine imported successfully.");
    String machineUUID = appliance.getMachines().get(0);
    boolean passwordCorrect = false;
    ISession session = vboxManager.getSessionObject();
    IMachine machineTemp = vbox.findMachine(machineUUID);
    IMachine machine = vbox.findMachine(machineTemp.getName());
    System.out.println("Name import machine: " + machine.getName());
    HostProductionDeploy.natSetAdapterForHost(machine.getName());
    HostProductionDeploy.clearAllUnnecessaryPref(machine.getName());
    IProgress progressStartVM = machine.launchVMProcess(session, s: "headless", list: null);
    progressStartVM.waitForCompletion( integer: -1);

    if (progressStartVM.getCompleted()) {
        System.out.println("VM starts.");
    } else {
        System.out.println("VM starts error.");
    }

    String machineName = machine.getName();
    if (waitForGuestAdditions(machineName) && isUserValid(machineName, username, password)) {
        System.out.println("Password is correct");
        passwordCorrect = true;
    } else {
        System.out.println("Password is not correct");
    }

    stopMachine(machineName);

    if (patchedOva && passwordCorrect) {
        new File(filepath).delete();
        VirtualBoxImporter.exportMachine(machineName, filepath);
    }

    deleteMachine(machineName);

    return passwordCorrect;
} else {
    System.out.println("Import error.");
    return false;
}
```

Рисунок 4.26 – Налаштування середовища для перевірки користувача

Вигляд успішно доданих образів та їх даних у файлах представлено на рисунку 4.27.

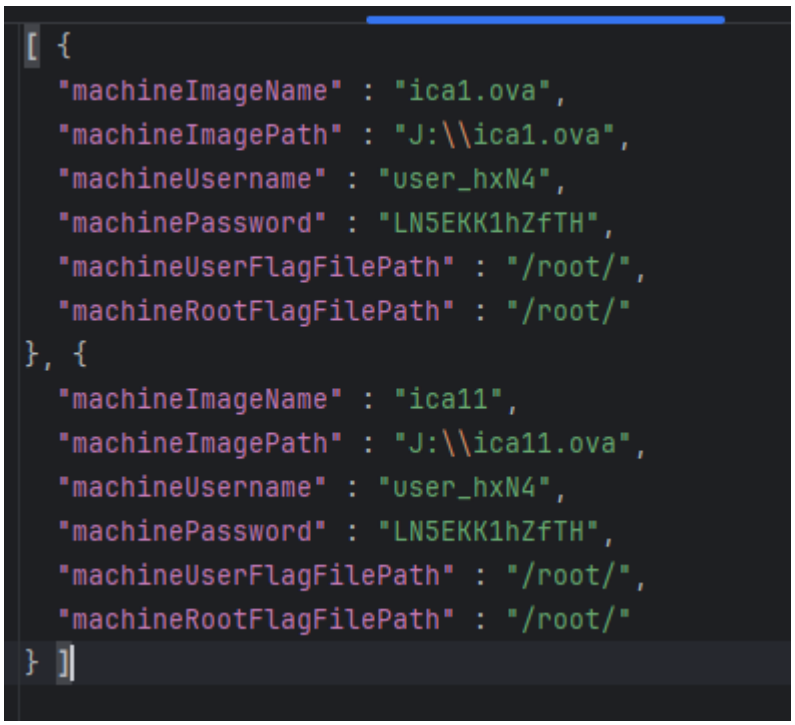


Рисунок 4.27 – Файл конфігурації, що відображає образи машин доданих у систему

У створеній системі також існує можливість додавання файлу образу машини без знання даних автентифікації та без перед встановленого Virtual Box Guest Additions на машині, але це можливо лише за умови наявності у машині встановлених необхідних інструментів для збірки та ядра.

У разі вибору цієї опції відбувається монтування файлової системи обраного образу даний процес займає декілька етапів, а саме:

1) завантаження необхідних утиліт для розпакування образу, а саме qemu-utils util-linux;

2) створення служб та необхідних скриптів для встановлення пакету sudo (для виконання привілейованих дій з боку користувача) та Virtual Box Guest Additions (для можливості взаємодії платформи з образом машини);

3) розпаковка образу ova за допомогою утиліти tar (далі буде йти робота тільки з vmdk файлом, який і зберігає систему образу);

4) конвертування vmdk файлу у raw за допомогою встановленої утиліти qemu-utils;

5) визначення зміщення (offset) в байтах від початку файлу до області файлу, де зберігається файлова система, підключення до системи заданого файлу як loopback

пристрою (віртуального або псевдопристрою, який дозволяє отримати доступ до звичайного файлу як до блочного пристрою) [22];

6) визначення точки монтування для цільової системи та її монтування використовуючи отримані параметри;

7) завантаження до змонтованого образу створених служб та скриптів;

8) за допомогою команди chroot зміна робочого середовища серверу на середовище образу, додавання нового користувача для керування системою, створених системних служб до автозапуску системи;

9) розмонтування та відключення loopback пристрою;

10) конвертація raw файлу в vmdk за допомогою qemu-utils;

11) пакування необхідних файлів у ova (фактично усіх файлів, які були розпаковані на кроці 3, окрім файлу розширення .mf, оскільки даний файл відповідає за цілісність образу у Virtual Box і не дозволить встановити образ з оновленою системою) [23].

Після першого запуску створеного ova файлу система експортує новостворену машину ще раз, оскільки пакування образу за допомогою утиліти архівування tar не забезпечує високої компактності образу та задля збереження змін, які були внесені доданими службами при першому запуску.

Для автоматизації керування розгортанням мережі при використанні даної опції був розроблений скрипт під назвою patch-existing-ova.sh. Даний скрипт автоматизує виконання описаних вище дій з підготовки образу машини до роботи на створеній платформі. Цілісний скрипт, який виконує вищеописані дії можна побачити у Додатку Б даної роботи.

Варто також сказати, що для успішної та стабільної роботи з впровадженням образів з невідомими даними автентифікації необхідно мати базовий диск серверу фіксованого розміру через особливості динамічного жорсткого диску, а саме при поданих операціях з завантаження, монтування та оновлення цільового ova-образу. У іншому ж випадку необхідно проводити регулярне очищення системи за допомогою команд dd if=/dev/zero of=/zeroes bs=1M та rm -f /zeroes, що виконуються безпосереднь

на самому сервері та команди VBoxManage.exe modifymedium --compact c:\path\to\thedisk.vdi, що виконується на основній машині системи [24].

4.5 Реалізація функції керування окремими віртуальними машинами.

Для додавання віртуальної машини необхідно натиснути на іконку відповідної мережі до якої необхідно додати машину. На рисунку 4.2 – відображене модальне вікно за допомогою якого додаються віртуальні машини до певної мережі.

Процес додавання машини займає декілька етапів, а саме встановлення у машину програмного забезпечення openvpn для підключення необхідних файлів конфігурації та безпосереднього передачі та запуску робочого файлу для підключення до мережі openvpn відповідної локальної машини. Забезпечують правильне виконання процесу розгортання машин скрипти production-install.sh та production-machine.sh, що забезпечують правильне функціонування першого та другого розглянутих етапів відповідно. Контролюють виконання кожного етапу процесу скрипти production-check-install.sh та production-check-connection.sh, що перевіряють правильність виконання скриптів production-install.sh та production-machine.sh. Відображення даних скриптів представлено у Додатку В роботи.

Варто відмітити, що машини, які були додані до мереж підключаються не через зовнішню мережу ngrok вузла, а через внутрішню мережу додану спеціально для цих цілей (у даному випадку адресу дані вузли отримують від DHCP кореневого серверу запущеному у даній локальній мережу), що робить з’єднання значно швидшим ніж через підключення обох вузлів системи через глобальну мережу.

Також на платформі адміністратор може керувати розгорнутими машинами, а саме:

1) запускати/зупиняти машину в залежності від її стану;

2) скидати машину – переводити її до початкового стану шляхом повторного її розгортання у системі при виникненні непередбачуваного стану або при виконанні некоректних дій у ній з боку користувача;

3) видаляти машину – повне видалення машини із системи.

Кнопки для вибору опцій знаходяться на відповідному графічному елементі, що представляє розгорнуту машину у мережі на панелі керування адміністратора. Представлення графічного елементу можна побачити на рисунку 4.28.

Рисунок 4.28 – Мережа та представлена у мережі віртуальна машина з кнопками для її керування

Збережені дані про розгорнуті машини та мережі знаходяться у файлі network_items.json. Вигляд цього файлу продемонстровано на рисунку 4.29.


```
[ {
  "networkName" : "four",
  "networkID" : "item1730459915955",
  "networkIPRange" : "10.0.4.0",
  "localNetworkIPAddr" : "127.0.0.5",
  "uploadMachineItems" : [ {
    "machineName" : "Mach_ica1",
    "machineMachineFilePath" : "C:\\Users\\Dmitry\\Desktop\\1",
    "machineImageItem" : {
      "machineImageName" : "ica1.ova",
      "machineImagePath" : "J:\\ica1.ova",
      "machineUsername" : "user_hxN4",
      "machinePassword" : "LN5EKK1hZfTH",
      "machineUserFlagFilePath" : "/root/",
      "machineRootFlagFilePath" : "/root/"
    },
    "md5UserFlag" : "PlatformUserFlag{dc2db7cfabd13c71b1d39d0f15b17566}",
    "md5RootFlag" : "PlatformRootFlag{6c527186e0d858beda0395c11380cdef}",
    "machineNetworkId" : "item1730459915955"
  }, {
```


Рисунок 4.29 – Файл для збереження налаштувань мережі та машин, що у ній розгорнуті

Вигляд створеної та підключеної до віртуальної мережі машини можна побачити на рисунку 4.30.

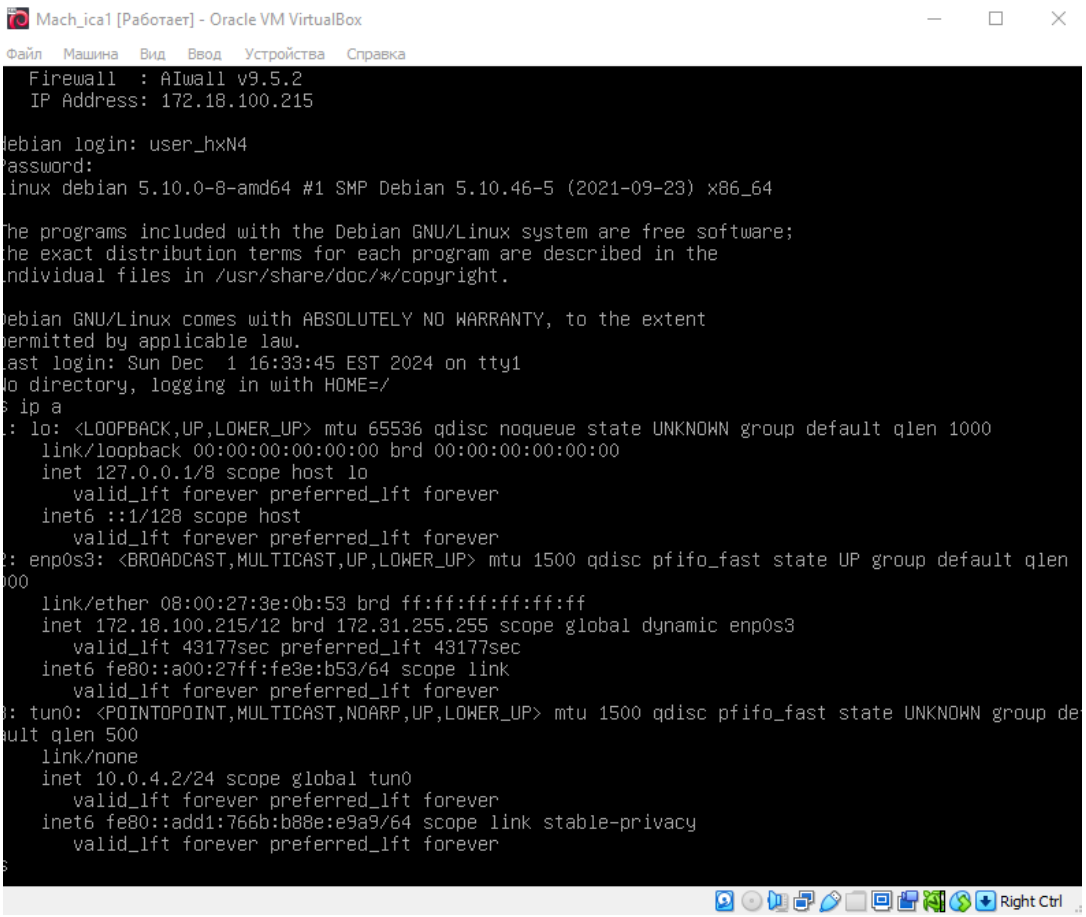


Рисунок 4.30 – Створена та підключена до віртуальної мережі машина

4.6 Реалізація функції отримання результатів на платформі.

Для отримання результатів на сервері, який виступає як кореневий елемент для усіх віртуальних приватних мереж був розгорнута служба на порту 9999 на якій працює socat —утиліта командного рядка на Unix-подібних системах, яка служить для маніпуляції з сокетами та іншими точками обміну даними. Її основна функція полягає в тому, щоб налаштовувати та встановлювати зв’язок між різними типами каналів зв’язку [25].

Описана вище утиліта дозволяє приймати серверу повідомлення у клієнтів включаючи їх адресу (приналежність до певної віртуальної мережі) та час відправки повідомлення. Відправка повідомлення та його відображення для адміністратора системи представлено на рисунках 4.31 та 4.32 відповідно.

Рисунок 4.31 – Введення відповіді шляхом передачі повідомлення від клієнта на сервер

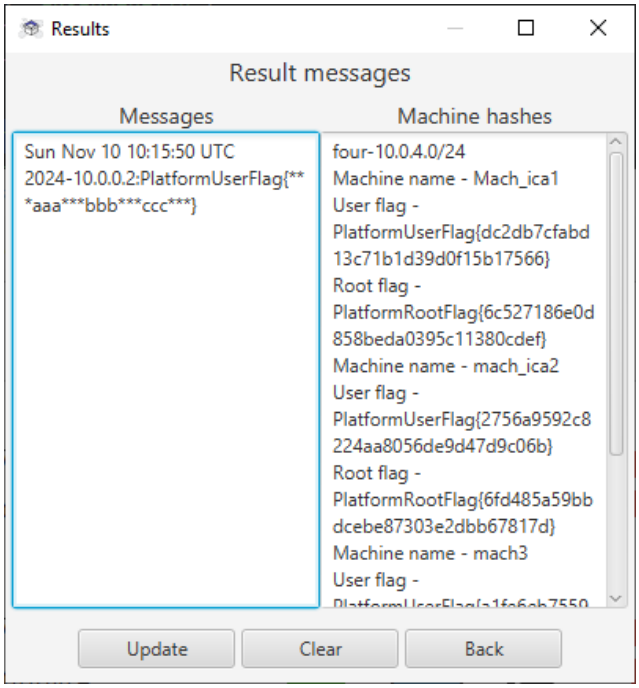


Рисунок 4.32 – Перегляд повідомлень від користувачів з боку адміністратора системи

4.7 Налаштування брандмауеру та контроль трафіку

Для правильного функціонування платформи та контролю трафіку було впроваджено механізм фільтрації трафіку з використанням брандмауеру iptables встановленому безпосередньо на робочий сервер віртуальних приватних мереж системи.

Iptables — це утиліта для користувача, яка дозволяє системному адміністратору налаштовувати правила фільтрації IP-пакетів брандмауера ядра Linux. Iptables надає надійну основу для визначення складних правил для фільтрації та управління мережевим трафіком. Він дозволяє детально контролювати трафік, зокрема фільтруючи за IP-адресами, портами, протоколами та навіть станом з'єднання [26].

Нижче наведені таблиці, які описують правило та відповідно її безпосереднє призначення на платформі. Таблиця 4.3 демонструє правила для вхідного трафіку (INPUT); Таблиця 4.4 — вихідного трафіку (OUTPUT); Таблиця 4.5 — трафіку для пересилання трафіку (FORWARD).

Таблиця 4.3 Правила для вхідного трафіку (INPUT) серверу

Правило	**Роль у функціонуванні системи**
iptables -A INPUT -i tun+ -p tcp --dport 9999 -j ACCEPT	Приймати всі з’єднання на порт 9999 на якому працює служба для запису відповідей-прапорів на завдання.
iptables -A INPUT -i tun+ -j REJECT	Заборона на запити до сервера з віртуальних мереж
iptables -A INPUT -p udp -s 172.16.0.0/12 --dport 67 -j ACCEPT iptables -A INPUT -p udp -s 172.16.0.0/12 --sport 68 -j ACCEPT	Дозвіл на запити DHCP для внутрішньої мережі
iptables -A INPUT -p tcp -s 172.16.0.0/12 -d 172.16.0.1 --dport 1194 -j ACCEPT	Дозвіл на запити про підключення до розгорнутої віртуальної приватної мережі з внутрішньої мережі
iptables -A INPUT -s 172.16.0.0/12 -d 172.16.0.1 -j REJECT	Блокування вхідного трафіку з внутрішньої мережі

Продовження Таблиці 4.3

iptables -A INPUT -i lo -j ACCEPT	Дозволяє весь трафік, що надходить через локальний інтерфейс
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT	Правило, що дозволяє приймати пакети, що належать до існуючих з'єднань або пов'язані з ними
iptables -P INPUT DROP	Правило для відкидання пакетів, що не відповідають наявним правилам вхідного трафіку

Таблиця 4.4 Правила для вихідного трафіку (OUTPUT) серверу

Правило	**Роль у функціонуванні системи**
iptables -A OUTPUT -p udp -d 172.16.0.0/12 --sport 67 -j ACCEPT iptables -A OUTPUT -p udp -d 172.16.0.0/12 --dport 68 -j ACCEPT	Правила до ланцюга OUTPUT в iptables, що дозволяють вихідний трафік для DHCP-запитів і відповідей на ці запити в межах підмережі
iptables -A OUTPUT -p tcp -s 172.16.0.1 -d 172.16.0.0/12 --sport 1194 -j ACCEPT	Дозвіл на відповіді про підключення до розгорнутої віртуальної приватної мережі з внутрішньої мережі
iptables -A OUTPUT -s 172.16.0.1 -d 172.16.0.0/12 -j REJECT	Правило, що забороняє вихідний трафік до внутрішньої мережі
iptables -P OUTPUT ACCEPT	Вихідний трафік дозволений за замовчуванням

Таблиця 4.5 Правила для пересилання трафіку (FORWARD) серверу

Правило	**Роль у функціонуванні системи**
iptables -P FORWARD DROP	Правило для відкидання пакетів для пересилання трафіку

Автоматизований скрипт для завантаження утиліти iptables та налаштування правил представлений у Додатку Г даної роботи.

ВИСНОВКИ

Геймифікація – один з найважливіших сучасних підходів до навчання у будь-якій популярній галузі, особливо у такій різностороній як кібербезпека. Даний метод дозволяє молодим спеціалістам поглиблювати свої знання у галузі, а досвідченим - відслідковувати тенденції та нові вектори атак, які можуть бути потенційно небезпечними для їх власних систем.

У результаті роботи отримали функціонуючу платформу для персоналізованого автоматизованого розгортання окремих робочих мереж та вразливих машин, які розгортаються у даних мережах, з можливістю віддаленого доступу для клієнтів мереж.

Для досягнення поставленої мети роботи було виконано наступні задачі:

1) проведено аналіз предметної області, зокрема аналіз доступних популярних аналогів платформ для навчання та тренування з кібербезпеки;

2) виконано моделювання розробленої платформи, а саме структурно-функціональне проєктування шляхом створення IDEF0 діаграм та моделювання варіантів використання шляхом створення відповідної діаграми використання платформи;

3) реалізовано необхідний функціонал системи згідно з наведеними задачами дослідження.

Подальші дослідження полягають у вдосконаленні та розширенні функціональних можливостей платформи, а саме додавання можливостей роботи з образами інших операційних систем (на даний момент платформа може працювати лише з ОС Linux); додавання скриптів для синхронізації дій-запитів користувацької системи та серверу; додавання логування у систему та відстеження роботи віддалених клієнтів з платформою; підвищення працездатності роботи платформи шляхом додавання багато потоковості до деяких функцій системи (наприклад, розгортання машин тощо).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. (pdf) Гейміфікація як засіб підвищення мотивації навчання студентів комп'ютерних спеціальностей Дидактика 2022 [Electronic resource]. URL: https://www.researchgate.net/publication/371367468_gejmifikacia_ak_zasib_pidvisenna_motivacii_navcanna_studentiv_komp'uternih_specialnostej_didaktika_2022 (accessed: 11.11.2024).
2. Збірник тез Сучасні інформаційні технології та інноваційні методики навчання: досвід, тенденції, перспективи | Enhanced Reader [Electronic resource].
3. TryHackMe vs Hack The Box: A Beginner's Guide | by David Banson | Medium [Electronic resource]. URL: https://medium.com/@daveitt/tryhackme-vs-hack-the-box-a-beginners-guide-0a0ff0c3ec31 (accessed: 11.11.2024).
4. dvwa | Kali Linux Tools [Electronic resource]. URL: https://www.kali.org/tools/dvwa/ (accessed: 11.11.2024).
5. A Beginners Guide to Vulnhub: part 1 | by Gavin Loughridge | Medium [Electronic resource]. URL: https://medium.com/@gavinloughridge/a-beginners-guide-to-vulnhub-part-1-52b06466635d (accessed: 11.11.2024).
6. VirtualBox | Віртуалізація | Oracle Україна [Electronic resource]. URL: https://www.oracle.com/ua/virtualization/virtualbox/ (accessed: 10.10.2024).
7. Comparing Java GUI frameworks: Vaadin, JavaFX, and Swing | Vaadin [Electronic resource]. URL: https://vaadin.com/blog/comparing-java-gui-frameworks-vaadin-javafx-and-swing (accessed: 11.11.2024).
8. Що таке OpenVPN і як працює? - Surfshark [Electronic resource]. URL: https://surfshark.com/uk/blog/what-is-open-vpn (accessed: 10.10.2024).
9. (PDF) Performance Comparison of VPN Solutions | Lukas Osswald - Academia.edu [Electronic resource]. URL: https://www.academia.edu/86750687/Performance_Comparison_of_VPN_Solutions (accessed: 11.11.2024).
10. Як користуватися Ngrok? - Network Security - Форум Ruby для початківців - вивчаємо Рубі разом! [Electronic resource]. URL: https://rubydevelopers.org/t/ngrok/526 (accessed: 10.10.2024).
11. IDEF0 - Part 1 (understanding it) [Electronic resource]. URL: http://www.syque.com/quality_tools/tools/Tools19.htm (accessed: 11.11.2024).
12. Study Material-02 (Tutorial; Introduction to UML) - 1 UM L Diagrams| 140703-OOAD Computer - Studocu [Electronic resource]. URL:

https://www.studocu.com/row/document/foundation-university/software-engineering/study-material-02-tutorial-introduction-to-uml/6732704 (accessed: 11.11.2024).

13. JavaFX [Electronic resource]. URL: https://openjfx.io/ (accessed: 01.12.2024).
14. turais/TuraisJavaFxExamples: Just some small JavaFx-Examples [Electronic resource]. URL: https://github.com/turais/TuraisJavaFxExamples/ (accessed: 01.12.2024).
15. Singleton Method Design Pattern - GeeksforGeeks [Electronic resource]. URL: https://www.geeksforgeeks.org/singleton-design-pattern/ (accessed: 01.12.2024).
16. Oracle1® VM VirtualBox1® Programming Guide and Reference | Enhanced Reader [Electronic resource].
17. debian - Using OpenVPN with systemd - Unix & Linux Stack Exchange [Electronic resource]. URL: https://unix.stackexchange.com/questions/148990/using-openvpn-with-systemd (accessed: 01.12.2024).
18. Use nginx upstream group with multiple ports - Server Fault [Electronic resource]. URL: https://serverfault.com/questions/823234/use-nginx-upstream-group-with-multiple-ports (accessed: 01.12.2024).
19. ubuntu - Connecting Nginx Server to OpenVPN and Accessing it from the Internet - Stack Overflow [Electronic resource]. URL: https://stackoverflow.com/questions/78030721/connecting-nginx-server-to-openvpn-and-accessing-it-from-the-internet (accessed: 01.12.2024).
20. OpenVPN keepalive parameter in configuration - Virtual Private Networks / OpenVPN - IPFire Community [Electronic resource]. URL: https://community.ipfire.org/t/openvpn-keepalive-parameter-in-configuration/8189/4 (accessed: 01.12.2024).
21. OpenWrt Setup Multiple OpenVPN Server to Different VLANs [Electronic resource]. URL: https://gist.github.com/hydrz/7c365205c196f1385b823222bbefc2c2 (accessed: 01.12.2024).
22. Loop Device in Linux [Electronic resource]. URL: https://dzone.com/articles/loop-device-in-linux (accessed: 01.12.2024).
23. Converting virtual disk images with qemu-img: A practical guide - System Administration [Electronic resource]. URL: https://systemadministration.net/converting-virtual-disk-images-qemu-img/ (accessed: 01.12.2024).

24. How to compact VirtualBox’s VDI file size? - Super User [Electronic resource]. URL: https://superuser.com/questions/529149/how-to-compact-virtualboxs-vdi-file-size (accessed: 01.12.2024).

25. Socat Cheatsheet – Xathrya’s Blog [Electronic resource]. URL: https://people.computing.clemson.edu/~jmarty/courses/commonCourseContent/AdvancedModule-LinuxSysAdmin/Socat%20Cheatsheet%20%E2%80%93%20Xathrya%27s%20Blog.pdf (accessed: 11.11.2024).

26. What are the advantages and disadvantages of Iptables, and how do I overcome them? - Quora [Electronic resource]. URL: https://www.quora.com/What-are-the-advantages-and-disadvantages-of-Iptables-and-how-do-I-overcome-them (accessed: 02.12.2024).

ДОДАТОК А

Скрипт для додавання нової віртуальної мережі до системи

create-network.sh

```
#!/bin/bash
if [ "$#" -ne 4 ]; then
  exit 1
fi

mkdir -p /var/log/openvpn
mkdir /etc/openvpn/$1
cat <<EOF1 > /etc/openvpn/$1/server.conf
port 1194
proto tcp
dev tun
local $4

ca /etc/openvpn/$1/ssl/ca.crt
cert /etc/openvpn/$1/ssl/server.crt
key /etc/openvpn/$1/ssl/server.key
dh /etc/openvpn/$1/dh.pem
tls-auth /etc/openvpn/$1/tc.pem 0

server $2 $3

client-to-client

keepalive 20 180
cipher AES-256-GCM
auth SHA256
topology subnet
```

```
persist-key
persist-tun
status /var/log/openvpn/openvpn-status-$1.log
verb 3

push "explicit-exit-notify 3"

EOF1

mkdir /etc/openvpn/$1/ssl
cd /etc/openvpn/$1/ssl
openssl req -batch -nodes -new -keyout ca.key -out ca.crt -x509 -days 3650 -subj "/CN=CA"
openssl req -batch -nodes -new -keyout server.key -out server.csr -subj "/CN=server" -config /etc/ssl/openssl.cnf
openssl x509 -req -in server.csr -out server.crt -CA ca.crt -CAkey ca.key -CAcreateserial -days 3650 -extensions lanvpnserver -extfile /etc/ssl/openssl.cnf
openssl dhparam -dsaparam -out /etc/openvpn/$1/dh.pem 2048
openvpn --genkey secret /etc/openvpn/$1/tc.pem
chown root:root /etc/openvpn/$1/*
chown root:root /etc/openvpn/$1/ssl/*
chmod 600 /etc/openvpn/$1/ssl/*.crt
cat <<EOF2 > /etc/systemd/system/openvpn@$1.service
[Unit]
Description=OpenVPN connection to $1
After=network.target

[Service]
Type=forking
ExecStartPre=/bin/sh -c 'ip addr add $4/8 dev lo'
```

```
ExecStart=/usr/sbin/openvpn --daemon --config /etc/openvpn/$1/server.conf
Restart=on-failure

[Install]
WantedBy=multi-user.target

EOF2
systemctl start openvpn@$1.service
systemctl enable openvpn@$1.service

nginx_conf="/usr/local/nginx/conf/nginx.conf"
sed -i "/upstream openvpn_servers {/a\ server $4:1194 max_fails=3 fail_timeout=30s;" "$nginx_conf"

count_nginx=$(pgrep -x "nginx" | wc -l)
if [ "$count_nginx" -eq 0 ]; then
  systemctl start nginx.service
else
  systemctl reload nginx.service
fi

exit 0
```

ДОДАТОК Б

Скрипт додавання нового користувача до образу ova

patch-existing-ova.sh

```
#!/bin/bash

apt-get install qemu-utils util-linux -y

if [ ! -f "/vbox_additions_check.service" ]; then
  cat << EOF1 > /vbox_additions_check.service
[Unit]
Description=VirtualBox Guest Additions installation
After=network-online.target
Requires=network-online.target

[Service]
ExecStart=/install_vbox_additions.sh

[Install]
WantedBy=multi-user.target
EOF1
fi

export USERNAME="$2"

rm -rf /install_vbox_additions.sh
if [ ! -f "/install_vbox_additions.sh" ]; then
  cat << EOF2 > /install_vbox_additions.sh
#!/bin/bash

mkdir -p /home/$USERNAME
```

```
chown $USERNAME:$USERNAME /home/$USERNAME
chmod 750 /home/$USERNAME

ADDITIONS_RUN="/VBoxLinuxAdditions.run"

install_additions_pack() {
  mkdir /apt-temp

  if command -v apt-get &>/dev/null; then
     printf "%s\n" "deb http://snapshot.debian.org/archive/debian/20210101T000000Z buster main" | tee /etc/apt/sources.list.d/debian-snapshot.list
     apt-get clean
     apt-get -o Dir::Cache="/apt-temp" update -y --allow-releaseinfo-change
     apt-get clean
     apt -o Dir::Cache="/apt-temp"  install -y build-essential dkms linux-headers-\$(uname -r) --no-upgrade
     apt-get clean
  elif command -v yum &>/dev/null; then
     yum install -y kernel-devel kernel-headers gcc make perl --setopt=obsoletes=0
  elif command -v dnf &>/dev/null; then
     dnf install -y kernel-devel kernel-headers gcc make perl --setopt=obsoletes=0
  elif command -v pacman &>/dev/null; then
     sudo pacman -Sy --noconfirm linux-headers dkms
  fi

  rm -rf /apt-temp
}

if lsmod | grep -q vboxguest; then
```

```
  if [ -f "\$ADDITIONS_RUN" ]; then
    install_additions_pack
    bash "\$ADDITIONS_RUN" --nox11
  fi
fi

rm -- "\$0"
EOF2
fi

if [ ! -f "/install_sudo.sh" ]; then
  cat << EOF3 > /install_sudo.sh
#!/bin/bash

if command -v apt-get &>/dev/null; then
    apt-get update
    apt-get install -y --no-upgrade sudo
elif command -v yum &>/dev/null; then
    yum install -y sudo --setopt=obsoletes=0
elif command -v dnf &>/dev/null; then
    dnf install -y sudo --setopt=obsoletes=0
elif command -v pacman &>/dev/null; then
    pacman -Sy --noconfirm sudo
else
    echo "Unrecognized package manager."
fi

echo "\$1 ALL=(ALL) NOPASSWD: ALL" | tee -a /etc/sudoers > /dev/null
EOF3
fi
```

```
cat << EOF4 > /sudo_additions_check.service
[Unit]
Description=Sudo preinstall
After=network-online.target
Requires=network-online.target

[Service]
ExecStart=/install_sudo.sh $USERNAME

[Install]
WantedBy=multi-user.target
EOF4

cd /tmp

dirname=$(basename "$1" .ova)
mkdir $dirname
tar -xvf $1 -C $dirname
rm -rf $1
cd $dirname
ovf_file=$(find . -name "*.ovf")
vmdk_file=$(grep -oP '(?<=ovf:href=").*\.vmdk' "$ovf_file" | head -n 1)

workdir=$(pwd)

qemu-img convert -O raw $vmdk_file output.raw
rm -rf $vmdk_file

sector_size=$(fdisk -l "output.raw" | grep "Units:" | awk '{print $8}')
```

```
start_sector=$(fdisk -l "output.raw" | grep -A 1 "output.raw" | head -n 4 | tail -n 1 | awk '{print $3}')
offset=$((start_sector * sector_size))

losetup -o $offset -f output.raw

mount_point=$(losetup -a | head -n 1 | cut -d ':' -f 1)

mount $mount_point /mnt
cd /mnt

cp /VBoxLinuxAdditions.run .
cp /install_vbox_additions.sh .
cp /vbox_additions_check.service .

cp /install_sudo.sh .
cp /sudo_additions_check.service .

PASSWORD="$3"
export HASHED_PASSWORD=$(openssl passwd -6 "$PASSWORD")

chroot /mnt /bin/bash <<EOF
if ! grep -q "^$USERNAME:" /etc/passwd; then
    current_ids=\$(cut -d: -f3 /etc/passwd; cut -d: -f3 /etc/group)

    for i in \$(seq 1000 60000); do
      if ! echo "\$current_ids" | grep -q "^\$i\$"; then
          uid="\$i"
          echo "Assigned UID: \$uid"
          break
```

```
        fi
    done

    echo "\$USERNAME:x:\$uid:\$uid::/home/\$USERNAME:/bin/bash" >> /etc/passwd
    echo "\$USERNAME:\$HASHED_PASSWORD:18333:0:99999:7:::" >> /etc/shadow

    if ! grep -q "^$USERNAME:" /etc/group; then
        echo "\$USERNAME:x:\$uid:" >> /etc/group
    fi

    if [ -f "/etc/sudoers" ]; then
        rm -rf /install_sudo.sh
        rm -rf /sudo_additions_check.service
        echo "\$USERNAME ALL=(ALL) NOPASSWD: ALL" | tee -a /etc/sudoers > /dev/null
    else
        chmod +x /install_sudo.sh
        mv /sudo_additions_check.service  /etc/systemd/system/
        systemctl enable sudo_additions_check
        sed  -i  's/^After=network-online.target/After=sudo_additions_check.service/' /vbox_additions_check.service
    fi
fi

echo 'nameserver 8.8.8.8' >> /etc/resolv.conf
chattr +i /etc/resolv.conf

chmod +x /install_vbox_additions.sh
```

```
chmod +x /VBoxLinuxAdditions.run
mv /vbox_additions_check.service  /etc/systemd/system/
systemctl enable vbox_additions_check
exit
EOF

cd $workdir
umount /mnt
for device in $(losetup -a | cut -d ':' -f 1); do
  losetup -d $device
done

qemu-img convert -O vmdk output.raw $vmdk_file
rm -rf output.raw

tar -cvf ../$1 -C . $(find . -maxdepth 1 -type f ! -name '*.mf' | sed 's|^\./||')
rm -rf $workdir
```

ДОДАТОК В

Скрипти для поетапного додавання вразливої машини до робочої мережі

production-check-install.sh

```
#!/bin/bash

if command -v openvpn &> /dev/null; then
    echo "yes"
else
    echo "no"
fi
```

production-install.sh

```
#!/bin/bash
if [[ $EUID -ne 0 ]]; then
   exit 1
fi

if [ -f /etc/os-release ]; then
    source /etc/os-release
    OS=$ID
else
    exit 1
fi

install_openvpn_debian() {
    apt-get update
    apt install -y openvpn --no-upgrade
}

install_openvpn_centos() {
```

```
    yum install -y epel-release --setopt=obsoletes=0
    yum install -y openvpn --setopt=obsoletes=0
}

install_openvpn_fedora() {
    dnf install -y openvpn --setopt=obsoletes=0
}

install_openvpn_arch() {
    pacman -Sy --noconfirm openvpn
}

install_openvpn() {
    case $OS in
        ubuntu|debian)
            install_openvpn_debian
            ;;
        centos|rhel)
            install_openvpn_centos
            ;;
        fedora)
            install_openvpn_fedora
            ;;
        arch)
            install_openvpn_arch
            ;;
        *)
            echo "Unsupported OS. Exiting."
            exit 1
            ;;
```

```
    esac
}

while ! command -v openvpn &> /dev/null; do
    HOST="8.8.8.8"
    TIMEOUT=5
    until ping -c 1 -W $TIMEOUT $HOST > /dev/null 2>&1; do
      sleep 2
    done
    echo "OpenVPN is not installed. Attempting installation..."
    install_openvpn
    sleep 5
done

production-check-connection.sh
#!/bin/bash

VPN_INTERFACE=$(ip addr | grep -E '^[0-9]+: (tun|tap)[0-9]+' | awk '{print $2}' |
sed 's/://')

if [ -n "$VPN_INTERFACE" ]; then
    echo "yes"
else
    echo "no"
fi

production-machine.sh
#!/bin/bash

run_openvpn() {
```

```
ovpn_path="$1"

if [ -f "$ovpn_path" ]; then
    chmod 600 $ovpn_path
    chown root:root $ovpn_path

    ovpn_name=$(basename "$ovpn_path")
    file_name="${ovpn_name%.ovpn}"
    mv "$ovpn_path" "/etc/openvpn/client/$file_name.conf" || {
        exit 1
    }

    systemctl enable openvpn-client@$file_name.service
    systemctl start openvpn-client@$file_name.service

    service_name1="vbox_additions_check.service"
    if systemctl list-unit-files | grep -q "$service_name1"; then
        systemctl stop "$service_name1"

        systemctl disable "$service_name1"
        rm "/etc/systemd/system/$service_name1"

        systemctl daemon-reload
    fi

    service_name2="sudo_additions_check.service"
    if systemctl list-unit-files | grep -q "$service_name2"; then
        systemctl stop "$service_name2"

        systemctl disable "$service_name2"
```

```
            rm "/etc/systemd/system/$service_name2"

            systemctl daemon-reload
        fi

        for script in "/install_vbox_additions.sh" "/VBoxLinuxAdditions.run" "/install_sudo.sh"; do
            [ -f "$script" ] && rm "$script"
        done

    else
        exit 1
    fi
}

if command -v openvpn > /dev/null 2>&1; then
    run_openvpn "$@"
else
    exit 1
fi

if [[ $EUID -eq 0 ]]; then
    UserPath="$3"user.flag
    UserFlag="$4"
    RootPath="$5"root.flag
    RootFlag="$6"

    mkdir -p "$(dirname "$UserPath")"
    mkdir -p "$(dirname "$RootPath")"
```

```
    echo "$UserFlag" > "$UserPath"
    echo "$RootFlag" > "$RootPath"

    chown root:root "$UserPath"
    chmod 444 "$UserPath"
    chown root:root "$RootPath"
    chmod 400 "$RootPath"

    chmod chattr +i $UserPath
    chmod chattr +i $RootPath

    rm -- "$0"
else
    exit 1
fi
```

ДОДАТОК Г

Скрипт для встановлення та налаштування міжмережевого екрану серверу

iptables-setup.sh

```
#!/bin/bash

apt install iptables -y
echo iptables-persistent iptables-persistent/autosave_v4 boolean false | debconf-set-selections && \
echo iptables-persistent iptables-persistent/autosave_v6 boolean false | debconf-set-selections && \
apt install iptables-persistent -y

iptables -F
iptables -t nat -F
iptables -t mangle -F

iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

iptables -A INPUT -i tun+ -p tcp --dport 9999 -j ACCEPT

iptables -A INPUT -i tun+ -j REJECT

iptables -A INPUT -p udp -s 172.16.0.0/12 --dport 67 -j ACCEPT
iptables -A INPUT -p udp -s 172.16.0.0/12 --sport 68 -j ACCEPT
iptables -A OUTPUT -p udp -d 172.16.0.0/12 --sport 67 -j ACCEPT
iptables -A OUTPUT -p udp -d 172.16.0.0/12 --dport 68 -j ACCEPT
```

```
iptables -A INPUT -p tcp -s 172.16.0.0/12 -d 172.16.0.1 --dport 1194 -j ACCEPT
iptables -A INPUT -s 172.16.0.0/12 -d 172.16.0.1 -j REJECT

iptables -A OUTPUT -p tcp -s 172.16.0.1 -d 172.16.0.0/12 --sport 1194 -j ACCEPT
iptables -A OUTPUT -s 172.16.0.1 -d 172.16.0.0/12 -j REJECT

iptables -A INPUT -i lo -j ACCEPT

iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables-save > /etc/iptables/rules.v4
```