Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи магістра

на тему: «Методи та засоби автоматизації Ansible для адміністрування та безпеки Microsoft Active Directory»

Виконав: студент 2-го курсу групи КБм-24-1
спеціальності 125 «Кібербезпека та захист інформації»
_____ Богдан МАРКЕЄВ
Керівник: Старший викладач кафедри КІ та КБ,
_____ Ігор ФАЛЬКОВСЬКИЙ
Рецензент: Доцент кафедри КІ та КБ, к.пед.н., доцент
_____ Олена ГОЛОВНЯ

Житомир – 2025

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 125 «Кіберберзпека та захист інформації»
Освітня програма «Кіберберзпека»

ЗАТВЕРДЖУЮ
Завідувач
кафедри комп'ютерної
інженерії та кібербезпеки
Андрій ЄФІМЕНКО
6 жовтня 2025 р.

ЗАВДАННЯ

на випускну кваліфікаційну роботу магістра

Студент: Маркеєв Богдан Володимирович

Тема роботи: **«Методи та засоби автоматизації Ansible для адміністрування та безпеки Microsoft Active Directory»**

затверджена Наказом університету від 6 жовтня 2025 р. № 566/с

Термін здачі студентом закінченої роботи 22 грудня 2025 р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження): Предметом дослідження є методи та засоби автоматизованого налаштування інфраструктури та безпеки AD з використанням Ansible. Об'єктом дослідження є процес автоматизації розгортання та налаштування адміністрування та безпеки контролерів домену.

Обладнання та програмне забезпечення:

1. Гіпервізор VirtualBox
2. Windows Server 2022 Standard, Windows 10, Linux Ubuntu 22.04 з Ansible
3. LAPS.x64.msi, Mimikatz 2.2.0

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Завідувач кафедри КІ та КБ, к.т.н., доцент Андрій ЄФІМЕНКО	6.10.25	6.10.25

Студент _____ Богдан МАРКЕЄВ

Керівник _____ Ігор ФАЛЬКОВСЬКИЙ

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Остаточне затвердження теми, формування мети, завдань, структури дипломної роботи	17.09.2025–23.09.2025	Виконано
2	Збір і первинний аналіз джерел з Active Directory, Ansible, засобів безпеки, формування попереднього списку літератури	24.09.2025–07.10.2025	Виконано
3	Написання розділу 1: аналіз проблематики, загроз AD і огляд інструментів автоматизації й захисту	08.10.2025–21.10.2025	Виконано
4	Розробка концепції архітектури системи, формування вимог і технічного завдання	22.10.2025–04.11.2025	Виконано
5	Проєктування тестового стенду, мережевої структури, модулів адміністрування, безпеки, резервного копіювання	05.11.2025–18.11.2025	Виконано
6	Реалізація інфраструктури: розгортання VirtualBox-стенду, налаштування AD, Ansible, базових плейбуків	19.11.2025–02.12.2025	Виконано
7	Реалізація підсистеми безпеки, моніторингу, резервного копіювання, написання й відлагодження сценаріїв тестів	03.12.2025–10.12.2025	Виконано
8	Проведення експериментів, збір результатів, оформлення розділу з тестування й аналізом отриманих даних	11.12.2025–14.12.2025	Виконано
9	Написання висновків, остаточне оформлення звіту підготовка повного тексту до подання. Перевірка на антиплаігат та отримання відгуку керівника	15.12.2025–17.12.2025	Виконано
10	Подання закінченої роботи на кафедру	18.12.2025	Виконано

Студент _____ Богдан МАРКЕЄВ

Керівник _____ Ігор ФАЛЬКОВСЬКИЙ

АНОТАЦІЯ

Випускна кваліфікаційна робота магістра присвячена дослідженню методів та засобів автоматизації Ansible котрі пришвидшать та покращать процес розгортання контролерів домену, а також підвищать їх безпеку. Також дана тематика дозволить глибше зрозуміти тонкощі взаємодії середовища Microsoft Active Directory та засобу оркестрації Ansible.

Пояснювальна записка до випускної кваліфікаційної роботи магістра на тему «Методи та засоби автоматизації Ansible для адміністрування та безпеки Microsoft Active Directory» складається з вступу, трьох розділів, висновків, списку використаних джерел та 1 додатку обсягом 20 сторінок.

Загальний обсяг роботи займає 86 сторінок, містить 51 рисунок та 6 таблиць. Список використаних джерел містить 23 найменувань і займає 3 сторінки.

В першому розділі проведено аналіз літературних джерел за темою кваліфікаційної роботи, виділено місце автоматизації в загальній системі та виконано постановку технічного завдання, яке формує необхідні вимоги для майбутніх рішень.

В другому розділі розглянуто методи та засоби вирішення поставлених завдань, проведено проектування тестового середовища.

В третьому розділі проведено розгортання тестового стенду, налаштування систем адміністрування та захисту контролера домену. Проведено тестування на проникнення на основі поставленого технічного завдання.

Висновок містить в собі результати виконаної роботи.

У додатках до роботи наведено конфігураційні файли та лістинг плейбуків для адміністрування, захисту та взаємодії з контролером домену та клієнтськими станціями.

Ключові слова: MICROSOFT, ACTIVE DIRECTORY, VIRTUAL BOX, ANSIBLE, POWERSHELL, USERS, ПЛЕЙБУКИ, GPO, PSO, ОРГАНІЗАЦІЙНІ ОДИНИЦІ, КОМАНДЛЕТИ, АДМІНІСТРУВАННЯ, БЕЗПЕКА, OU, LAPS.

ANNOTATION

The master's thesis is devoted to researching Ansible automation methods and tools that will speed up and improve the process of deploying domain controllers, as well as increase their security. This topic will also provide a deeper understanding of the intricacies of the interaction between the Microsoft Active Directory environment and the Ansible orchestration tool.

The explanatory note to the master's thesis on ‘Ansible automation methods and tools for Microsoft Active Directory administration and security’ consists of an introduction, three chapters, conclusions, a list of references, and one appendix, totaling 20 pages.

The total volume of the work is 86 pages, containing 51 figures and 6 tables. The list of references contains 23 titles and occupies 3 pages.

The first chapter analyses the literature on the topic of the qualification work, highlights the place of automation in the overall system, and sets out the technical task that forms the necessary requirements for future solutions.

The second chapter examines methods and means of solving the tasks set and designs a test environment.

The third chapter describes the deployment of the test bench and the configuration of the domain controller's administration and protection systems. Penetration testing was performed based on the technical task set.

The conclusion presents the results of the completed research.

The appendices to the thesis include configuration files and listings of playbooks used for administration, protection, and interaction with the domain controller and client workstations.

Keywords: MICROSOFT, ACTIVE DIRECTORY, VIRTUAL BOX, ANSIBLE, POWERSHELL, USERS, PLAYBOOKS, GPO, PSO, ORGANIZATIONAL UNITS, CMDLETS, ADMINISTRATION, SECURITY, OU, LAPS.

ЗМІСТ

ВСТУП 8

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, АНАЛОГІВ ТА ІСНУЮЧИХ СИСТЕМ . 9

1.1 Аналіз сучасного стану та обґрунтування актуальності теми роботи... 9

1.2 Огляд існуючих аналогів забезпечення безпеки та автоматизації процесів Active Directory. 12

1.3 Загальна концепція архітектури 20

1.4 Постановка технічного завдання..... 21

РОЗДІЛ 2. ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА ЗАСОБІВ РЕАЛІЗАЦІЇ СИСТЕМИ 22

2.1. Архітектура тествого стенду проекту. Мережева взаємодія...... 23

2.2. Проєктування модулів адміністрування..... 26

2.3. Проєктування підсистеми безпеки, моніторингу та забезпечення безперервності..... 28

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СИСТЕМИ..... 32

3.1. Підготовка середовища та базове налаштування 32

3.2 Основні засоби адміністрування 37

3.3 Реалізація підсистеми безпеки 45

3.4 Тестування на проникнення Pass-the-Hash..... 58

ВИСНОВКИ..... 64

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 66

ДОДАТКИ..... 69

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ОС – Операційна Система.

VB – VirtualBox

MS – Microsoft.

AD – Active Directory

DHCP – Dynamic Host Configuration Protocol.

OU – Organizational Unit

WS – Work Station

GPO – Group Policy Object

PS – PowerShell.

ПЗ – програмне забезпечення

ПК – персональний комп’ютер

PtH – Pass-the-Hash

PtT – Pass-the-Ticket

TGT – Ticket Granting Ticket

TGS – Ticket Granting Service

MFA – Multi-Factor Authentication

PAM – Privileged Access Management

SIEM – Security Information and Event Management

APT – Advanced Persistent Threat

ВСТУП

Актуальність теми. У сучасному інформаційно-технологічному середовищі, корпоративні мережі стають все більш складними та вимогливими з точки зору ефективного управління та безпеки. Одним з ключових компонентів таких мереж є контролери доменів, які забезпечують централізоване управління користувачами, комп'ютерами та мережевими ресурсами на основі Active Directory.

Метою кваліфікаційної роботи є дослідження та розробка системи автоматизованого налаштування інфраструктури та безпеки Microsoft Active Directory, її контролерів домену за допомогою Ansible.

Об’єктом дослідження є процес автоматизації розгортання та налаштування мережевої інфраструктури, контролерів домену, процес створення користувачів, групових політик.

Предметом дослідження є методи та засоби автоматизованого налаштування інфраструктури та безпеки AD з використанням Ansible.

Визначена мета обумовлює наступні завдання:

- Здійснити аналіз сучасних методів автоматизації адміністрування та загроз безпеці Active Directory для визначення функціональних вимог до проєктованої системи.
- Розробити архітектуру програмного комплексу.
- Реалізувати тестовий стенд з використанням Ansible, створивши набір сценаріїв на основі поставлених завдань та вимог.
- Провести експериментальне дослідження ефективності системи.

У процесі виконання кваліфікаційної роботи було розроблено проект котрий використовує автоматизовані рішення Ansible для забезпечення розгортання та безпеки контролеру домену Active Directory.

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, АНАЛОГІВ ТА ІСНУЮЧИХ СИСТЕМ

1.1 Аналіз сучасного стану та обґрунтування актуальності теми роботи

Microsoft Active Directory (AD) є фундаментальною технологією для управління ідентифікацією, автентифікацією та авторизацією в корпоративних середовищах. Система AD забезпечує централізоване управління користувачами, пристроями та політиками безпеки, що робить її критично важливим компонентом сучасної IT-інфраструктури. [1]

В умовах цифрової трансформації компанії все більше покладаються на AD для забезпечення безпеки даних та контролю доступу до ресурсів. AD інтегрується з іншими корпоративними рішеннями, такими як Microsoft Exchange, Azure AD, Office 365, що дозволяє ефективно управляти правами доступу та політиками безпеки в гібридних середовищах. [2]

Попри численні переваги, централізація управління автентифікацією та авторизацією створює серйозні ризики, адже компрометація AD може призвести до повного контролю зловмисника над усіма системами організації. Саме тому питання захисту доменних контролерів (DC) та автоматизації процесів адміністрування є надзвичайно актуальними.

Сучасні загрози кібербезпеці спрямовані на компрометацію облікових даних, підвищення привілеїв та отримання контролю над доменними контролерами. Найпоширеніші серед них:

Атаки на облікові дані:

Pass-the-Hash (PtH) – атака, при якій зловмисник використовує хеш пароля замість пароля для автентифікації без необхідності його розшифрування. Pass-the-Ticket – використання викрадених квитків Kerberos для отримання доступу до системи. [3] Over-Pass-the-Hash– комбінована атака, що використовує NTLM-хеші для отримання квитків Kerberos. [4] Kerberoasting – отримання квитків служби Kerberos та їх розшифрування для отримання паролів службових облікових записів. [5]

Нижче наведено аналіз найнебезпечніших векторів атак на облікові дані.

Таблиця 1.1

Назва атаки	**Поширеність**	**Складність реалізації**	**Відомі приклади**	**Наслідки для бізнесу**
Pass-the-Hash (PtH)	Дуже висока(стандартна техніка для переміщення всередині мережі)	Низька(інструменти типу Mimikatz роблять її доступною навіть новачкам)	Target Breach (2013): Використовувалась для переміщення від порталу підрядника до платіжних систем.WannaCry / NotPetya: Використовували PtH для швидкого поширення.	Швидке горизонтальне переміщення. Компрометація одного ПК часто призводить до захоплення всієї мережі, якщо паролі локальних адміністраторів однакові.
Pass-the-Ticket (PtT)	Висока(серед APT-груп та у складних цільових атаках)	Середня/Висока(вимагає доступу до LSASS та розуміння Kerberos)	APT28 (Fancy Bear): Злам DNC (2016).APT29 (Cozy Bear): Кампанії кібершпіонажу.	Обхід багатофакторної автентифікації (MFA), оскільки квиток вже є підтвердженням входу. Дозволяє доступ до ресурсів без знання пароля користувача.
Over-Pass-the-Hash	Середня(використовується, коли блокується NTLM або для оновлення TGT)	Середня(вимагає наявності NTLM-хешу або AES-ключів)	MuddyWater (APT34): Іранська група, що використовує цю техніку для закріплення в мережі.	Отримання свіжого TGT-квитка Kerberos, що виглядає як легітимний вхід. Дозволяє атакувати навіть захищені мережі, де NTLM трафік моніториться або заблокований.
Kerberoasting	Дуже висока(одна з найулюбленіших технік пентестерів та хакерів)	Низька(не потребує прав адміністратора, лише будь-який доменний акаунт)	Akira Ransomware: Стандартний етап перед шифруванням.FIN7 (Carbon Spider): Використовували для ескалації привілеїв.	Отримання паролів привілейованих сервісних акаунтів (часто Domain Admins). Це прямий шлях до повного захоплення домену без необхідності експлойтів.

Атаки на доменні контролери:

Golden Ticket – атака, що дозволяє створити підроблений квиток Kerberos для отримання необмеженого доступу до AD. Silver Ticket – маніпуляція з TGS-квитками, що дає змогу отримати доступ до окремих сервісів без перевірки на рівні контролера домену. DCSync – емуляція реплікації каталогу AD, що дозволяє отримати NTLM-хеші всіх користувачів домену. DCShadow – створення фальшивого контролера домену, що може модифікувати критичні об'єкти AD.

Нижче наведено детальний аналіз найнебезпечніших векторів атак на інфраструктуру Active Directory. Для оцінки ризиків було систематизовано дані щодо їх поширеності, складності виконання та реальних наслідків.

Таблиця 1.2.

Назва атаки	**Поширеність**	**Складність реалізації**	**Відомі приклади**	**Наслідки для бізнесу**
Golden Ticket	Висока(переважно у цільових APT-атаках)	Висока(вимагає попереднього отримання прав адміністратора)	SolarWinds (Nobelium, 2020) – закріплення в мережі;Saudi Aramco (Shamoon 2.0)	Повна, часто незворотна компрометація домену. Забезпечує зловмиснику «вічний» доступ (Persistence) навіть після зміни паролів користувачів.
Silver Ticket	Середня(Lateral Movement)	Висока(вимагає глибокого розуміння протоколу Kerberos)	Часто фігурує у звітах Red Teaming як метод доступу до SQL або файлових серверів.	Непомітне викрадення конфіденційних даних або модифікація баз даних в обхід журналювання подій на контролері домену.
DCSync	Дуже висока(стандартний інструмент програм-вимагачів)	Середня(залежить від наявності прав реплікації GetNCChanges)	NotPetya (2017) – масовий збір хешів;Magic Hound (APT35)	Миттєва компрометація всіх облікових записів. Дозволяє отримати паролі адміністраторів у відкритому вигляді або створити Golden Ticket.
DCShadow	Низька(використовується елітними хакерськими групами)	Дуже висока(вимагає реєстрації підробленого сервера в конфігурації AD)	Carbon Spider (FIN7) – для обходу SIEM-систем.	Модифікація критичних об'єктів AD без створення стандартних логів безпеки. Зміни можуть залишатися непоміченими тривалий час.

Для запобігання атакам на інфраструктуру необхідно впроваджувати комплексні заходи безпеки, які охоплюють захист облікових даних, убезпечення контролерів домену, а також процеси моніторингу й аудиту. Зокрема, захист облікових даних передбачає використання технології Windows Defender Credential Guard для запобігання їх крадіжці, впровадження багатофакторної автентифікації для критично важливих акаунтів, регулярну зміну пароля системного облікового запису KRBTGT задля протидії атакам типу Golden Ticket, а також застосування систем управління привілейованим доступом (PAM) для чіткого розмежування прав адміністраторів.

Що стосується безпосереднього захисту контролерів домену, то він вимагає обмеження доступу до функцій реплікації AD для уникнення атак DCSync, використання групових політик для заборони застарілих та небезпечних протоколів автентифікації, а також моніторингу аномальної активності за допомогою рішень класу Microsoft Defender for Identity або SIEM-систем.

Паралельно з цим, ефективна система безпеки неможлива без моніторингу та аудиту, що включає детальне журналювання подій безпеки в AD, регулярний аналіз вразливостей за допомогою спеціалізованих інструментів, таких як PingCastle чи BloodHound, та автоматичне відключення або обмеження прав для облікових записів, що тривалий час не використовуються.

Водночас збільшення складності IT-інфраструктури значно ускладнює керування Active Directory та підвищує ризики виникнення помилок, які можуть бути використані зловмисниками. До основних проблем адміністрування належить високе навантаження на IT-персонал через значну кількість облікових записів і політик безпеки, що ускладнює управління та проведення аудиту. Ситуацію погіршує використання ручних процесів адміністрування, які збільшують ймовірність людських помилок і появи неконтрольованих змін. Окрім того, серйозними викликами є необхідність швидкого виявлення загроз, оскільки традиційні методи моніторингу не завжди дозволяють оперативно реагувати на інциденти, а також відсутність уніфікованої системи управління, через що компанії змушені застосовувати розрізнені інструменти, що суттєво ускладнює інтеграцію процесів та загальний контроль.

У зв’язку з цими викликами, організації все більше зацікавлені у впровадженні автоматизованих систем адміністрування AD, які б мінімізували людський фактор, забезпечували централізований контроль та підвищували безпеку інфраструктури.

1.2 Огляд існуючих аналогів забезпечення безпеки та автоматизації процесів Active Directory.

Active Directory є центральним елементом управління доступом у більшості корпоративних мереж, що робить її однією з головних цілей для зловмисників. За

даними Verizon Data Breach Investigations Report 2023, понад 80% атак на корпоративні системи спрямовані на механізми автентифікації та управління доступом.

Згідно зі звітом Mandiant (FireEye), у 2022 році 90% атак з підвищенням привілеїв використовували вразливості або помилки налаштування AD. Основні загрози включають атаки на облікові дані, компрометацію доменних контролерів та використання неправильно налаштованих прав доступу для ескалації привілеїв.

Захист AD базується на використанні вбудованих механізмів безпеки Microsoft та сторонніх інструментів, що доповнюють його можливості. Microsoft пропонує низку вбудованих механізмів захисту AD, які забезпечують базовий рівень безпеки. Вони включають групові політики, захист автентифікаційних даних та моніторинг активності в мережі.

Group Policy Objects виступають основним механізмом централізованого управління безпекою у Windows-середовищі. Завдяки цьому інструменту адміністратори мають змогу встановлювати політики складності паролів та їх примусову ротацію, обмежувати використання застарілих протоколів автентифікації, таких як NTLMv1 або LM Hash, а також налаштовувати автоматичне блокування облікових записів після серії невдалих спроб входу. Крім того, GPO дозволяють обмежувати права локальних адміністраторів і контролювати підвищення привілеїв. Водночас варто зазначити, що цей механізм не забезпечує захисту від складних атак рівня DCSync чи Golden Ticket і вимагає ретельного налаштування та регулярного аудиту конфігурацій.[6] [7]

Іншим важливим компонентом захисту є Windows Defender Credential Guard, який використовує технології віртуалізації для захисту автентифікаційних даних у пам’яті, запобігаючи крадіжці хешів паролів та квитків Kerberos. Це рішення ефективно захищає від атак типу Pass-the-Hash, Pass-the-Ticket та використання інструментів на кшталт Mimikatz, оскільки використовує ізольоване віртуалізоване середовище для зберігання чутливих даних. Проте суттєвим обмеженням технології є

те, що Credential Guard не здатний запобігти компрометації самих облікових записів через методи соціальної інженерії, зокрема фішинг, а його впровадження потребує обов'язкової підтримки віртуалізації на апаратному рівні.

Для виявлення складних загроз використовується Microsoft Defender for Identity (раніше відомий як Advanced Threat Analytics), який аналізує поведінкові шаблони користувачів та автоматично ідентифікує аномалії, що можуть свідчити про компрометацію. Цей засіб здатний виявляти такі критичні атаки, як DCSync, DCShadow, Kerberoasting та спроби нелегітимної ескалації привілеїв, забезпечуючи при цьому інтеграцію з Microsoft Sentinel та іншими SIEM-системами для автоматичного реагування на інциденти. Основними недоліками цього рішення є необхідність ретельного налаштування та оптимізації під специфіку конкретного середовища, а також висока вартість ліцензії, що часто робить його недоступним для малих підприємств.

Сторонні рішення для захисту Active Directory

Окрім вбудованих засобів Microsoft, існує багато сторонніх рішень, які розширюють можливості моніторингу, аудиту та виявлення загроз у Active Directory. Одним із таких інструментів є BloodHound, призначений для глибокого аналізу прав доступу та пошуку шляхів ескалації привілеїв. Він дозволяє виявляти небезпечні ланцюжки зв'язків між обліковими записами та групами, допомагаючи адміністраторам усувати надлишкові права та вразливі конфігурації. Втім, його недоліком є те, що цей інструмент часто використовується зловмисниками для розвідки перед атакою, до того ж він не забезпечує активного захисту, обмежуючись аналізом поточного стану.

Для автоматизованого аналізу ризиків часто використовується PingCastle – легкий інструмент, що виконує перевірку домену, ідентифікує слабкі місця в налаштуваннях та генерує звіти з рекомендаціями щодо покращення безпеки, хоча його робота обмежується статичною оцінкою без відстеження активності в реальному часі. У сфері моніторингу змін виділяється Netwrix Auditor for Active Directory,

який відстежує будь-які модифікації облікових записів, групових політик та дозволів, інтегруючись із SIEM-рішеннями для швидкого аналізу загроз. Однак впровадження цього засобу пов'язане з високою вартістю, при цьому він лише фіксує зміни після їх виконання, не запобігаючи атакам безпосередньо.

З розвитком технологій та розширенням підприємств, котрі використовують Active Directory, виникла об'єктивна потреба у рішеннях, здатних значно спростити процеси налаштування і розгортання контролерів домену та пов’язаних інструментів. На сьогоднішній день існує декілька рішень, котрі задовольняють вимоги більшості спеціалістів. Ключовими характеристиками таких систем є швидкість та ефективність розгортання, надійність та безпека, легкість у використанні, гнучкість та адаптивність, а також можливості інтеграції з існуючими системами і розвинений функціонал моніторингу та звітності.

Chef – це DevOps інструмент для управління конфігурацією, який керує інфраструктурою за допомогою написання коду, а не за допомогою ручного процесу, тому його можна дуже легко автоматизувати, тестувати і розгортати. Chef має клієнт-серверну архітектуру і підтримує різні платформи, такі як Windows, Ubuntu, Centos, Solaris тощо. Він також може бути інтегрований з хмарними платформами, такими як AWS, Google Cloud Platform, Open Stack тощо.

Керування конфігураціями відстежує всю інформацію, пов'язану з програмним і апаратним забезпеченням організації, а також ремонтує, розгортає та оновлює всю програму за допомогою автоматизованих процедур. Керування конфігураціями виконує роботу кількох системних адміністраторів і розробників, які керують сотнями серверів і додатків. Деякі інструменти, що використовуються для управління конфігурацією: Chef, Puppet, Ansible, CF Engine, SaltStack тощо.

Chef автоматизував конфігурацію, розгортання та управління додатками в мережі, навіть якщо робота відбувається в хмарному або гібридному середовищі. Chef є чудовим інструментом для прискорення доставки програмного забезпечення,

швидкість розробки програмного забезпечення відноситься до того, наскільки швидко програмне забезпечення здатне змінюватися у відповідь на нові вимоги або умови. [8]



Рис. 1.1 Використання Chef Infra для розгортання коду

Нижче наведено переваги та недоліки використання Chef.

Таблиця 1.3

Переваги Chef	Недоліки Chef
Одне з найбільш гнучких рішень для керування ОС та проміжним програмним забезпеченням.	Потребує тривалого навчання.
Розроблено спеціально для програмістів.	Початкове налаштування є складним.
Пропонує гібридні та SaaS рішення для серверів Chef Servers.	Бракує механізму "поштовху", тому немає негайних дій щодо змін. Процес витягування (pull) відбувається за визначеним графіком.
Має послідовний порядок виконання.	
Дуже стабільний, надійний і зрілий, особливо для великих розгортань як в публічних, так і в приватних середовищах.	

Ansible – це програмне забезпечення для автоматизації IT з відкритим вихідним кодом, написане мовою Python. Він може налаштовувати системи, розгортати програмне забезпечення та організовувати розширені робочі процеси для підтримки розгортання додатків, оновлення системи тощо.

Основними сильними сторонами Ansible є простота і легкість у використанні. Він також приділяє особливу увагу безпеці та надійності, маючи мінімальну кількість рухомих частин. Він використовує OpenSSH для транспортування, а також використовує зрозумілу для людини мову, яка призначена для швидкого початку роботи без тривалого навчання.

Спільний дистрибутив Ansible містить набір потужних інструментів командного рядка, які підтримуються на більшості операційних систем, де встановлено Python. Сюди входять Red Hat Enterprise Linux, Debian, Ubuntu, MacOS, FreeBSD, Microsoft Windows та інші.

Ansible виконується з керуючого вузла – наприклад, коли користувач запускає команду ansible-playbook. Керовані вузли – це пристрої, які автоматизуються, наприклад, сервер Microsoft Windows.

Для автоматизації Linux і Windows Ansible підключається до керованих вузлів і виштовхує на них невеликі програми, які називаються модулями Ansible. Ці програми написані як ресурсні моделі бажаного стану системи. Потім Ansible виконує ці модулі і видаляє їх після завершення. Ці модулі розроблені так, щоб бути ідемпотентними, коли це можливо, щоб вони вносили зміни в систему лише тоді, коли це необхідно.

Для автоматизації мережевих пристроїв та інших IT-пристроїв, де модулі не можуть бути виконані, Ansible запускається на керуючому вузлі. Оскільки Ansible не має агентів, він може взаємодіяти з пристроями, не вимагаючи встановлення додатків чи сервісів на керованому вузлі. Щоб збільшити потужність виконання для пристроїв без можливості запуску модулів, Ansible Automation Platform може розподіляти завдання автоматизації між вузлами виконання, використовуючи технологію, яка називається «сітка автоматизації». [9]

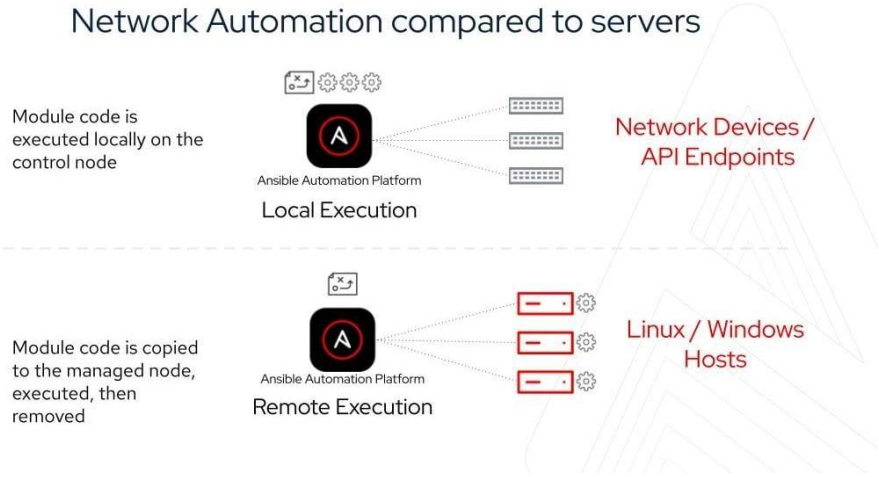


Рис. 1.2 – Порівняння Network Automation до серверів

Хоча Ansible зарекомендував себе як ефективний інструмент, його використання в середовищі Windows Server та Active Directory пов'язане з певними обмеженнями та проблемами. Насамперед, дається взнаки обмежена підтримка платформи Windows, оскільки система спочатку розроблялася як Linux-орієнтована, і попри додану сумісність, цей напрямок залишається менш розвиненим у порівнянні з Linux. Попри наявність значної кількості модулів для управління, деякі специфічні функції Windows Server та Active Directory можуть не мати повної підтримки або вимагати додаткових налаштувань. [10]

Суттєвим бар'єром також є складність початкової конфігурації. Оскільки Ansible використовує протокол WinRM для комунікації з Windows-системами, проте налаштування цього протоколу не є складним і не вимагає значних зусиль для забезпечення безпечного та стабільного з'єднання. Крім того, реалізація функцій управління Active Directory часто потребує складних налаштувань безпеки, що ускладнює використання Ansible у порівнянні з іншими інструментами.

Зрештою, фахівці можуть стикатися з обмеженими можливостями автоматизації безпосередньо для Active Directory. Наявні модулі не завжди покривають усі необхідні аспекти управління контролерами домену та об'єктами каталогу. Через це для реалізації складних сценаріїв розгортання та управління часто виникає потреба у написанні кастомних скриптів або використанні інших інструментів у поєднанні з Ansible.

Ще існують такі інструменти як MDT, який використовується для автоматизації розгортання операційних систем та базового налаштування та DSC, але усі ці рішення, як і попередні використовують у своїх оболонках можливості PowerShell. [11] Варто зазначити, що хоча DSC є нативним механізмом Windows для керування конфігураціями, йому бракує централізованої системи оркестрації "з коробки". У даному дослідженні DSC розглядатиметься не як конкурент Ansible, а як інструмент, який Ansible може використовувати для застосування складних конфігурацій. [12] Адже в свою чергу Ansible здатен деплоїти не лише командлети вже готових пакунків, а і використовувати командлети рядка PowerShell у тілі плейбуку для внесення тих чи інших змін у систему. [13]

Для наочного обґрунтування вибору інструментарію було проведено порівняльний аналіз найбільш поширених рішень автоматизації.

Таблиця 1.4

Характеристика	Ansible	Microsoft SCCM	PowerShell DSC	Chef / Puppet
Архітектура	Agentless (WinRM)	Agent-based	Agent-based (LCM)	Agent-based
Основна платформа	Linux (Control Node)	Windows	Windows	Linux
Складність впровадження	Низька (YAML)	Висока	Середня	Висока (Ruby)
Оркестрація	Висока	Середня	Низька (тільки конфігурація)	Висока
Вартість	Open Source (Безкоштовно)	Ліцензія (Висока)	Вбудовано в OS	Open Source / Enterprise

Ansible є оптимальним вибором для реалізації подальшого проекту завдяки відсутності необхідності встановлювати агенти, безкоштовній моделі розповсюдження та можливості централізованої оркестрації Windows Server та Active Directory вже з наявними пакунками.

1.3 Загальна концепція архітектури

Проєктована система автоматизації адміністрування та безпеки Active Directory з використанням Ansible є комплексним рішенням, що поєднує інфраструктурні, програмні та захисні компоненти. Основна мета полягає у підвищенні ефективності керування AD, мінімізації ручної роботи адміністраторів, зменшенні ризиків компрометації доменної інфраструктури та впровадженні автоматизованого моніторингу й відновлення.

Архітектура системи ґрунтується на принципах централізованого управління, автоматизації, інтеграції з наявними засобами безпеки, модульності, масштабованості та відмовостійкості. Централізоване управління реалізується через Ansible, який забезпечує розгортання, конфігурацію та моніторинг контролерів домену. Автоматизація спрямована на усунення рутинних ручних операцій, зокрема керування користувачами, групами, політиками безпеки та доступом до ресурсів.

Інтеграція з вбудованими механізмами Active Directory, як от LAPS та Group Policy, Kerberos, у поєднанні з кастомними сценаріями дає змогу ефективно виявляти аномалії та загрози. Модульна та масштабована архітектура спрощує розширення функціоналу й додавання нових сценаріїв автоматизації без суттєвого втручання в базову структуру. Висока доступність і відмовостійкість досягаються за рахунок використання механізмів резервного копіювання та аварійного відновлення.

Узагальнена архітектура програмного комплексу передбачає взаємодію керуючого вузла з Ansible та цільових контролерів домену й клієнтських станцій через чотири ключові модулі. Ядром виступає модуль автоматизації адміністрування, що забезпечує централізоване керування об’єктами Active Directory: створення, модифікацію та видалення облікових записів і OU, управління групами та розгортання GPO за допомогою PowerShell-скриптів і плейбуків Ansible.

Цілісність даних забезпечує модуль резервного копіювання, який виконує автоматичне створення резервних копій критичних компонентів AD, перевірку їхньої цілісності, тестування процедур відновлення та оперативне відновлення працездатності доменної інфраструктури у разі збоїв або кібератак.

Підсистема безпеки реалізована модулем захисту контролерів домену, орієнтованим на виявлення та нейтралізацію атак на кшталт Pass-the-Hash, Pass-the-Ticket та DCShadow. Він функціонує на основі контролю сесій автентифікації, застосування політик безпеки та аналізу журналів подій Windows.

Модуль моніторингу та автоматизованого тестування забезпечує безперервну оцінку стану Active Directory щодо вразливостей. Він відповідає за аудит реплікації, виявлення підозрілих змін у конфігурації та контроль складу груп привілейованих користувачів, що дозволяє проактивно реагувати на загрози цілісності системи.

1.4 Постановка технічного завдання

В результаті аналізу літератури за темою кваліфікаційної роботи було визначено актуальність розробки системи автоматизації адміністрування та безпеки MS AD з використанням Ansible.

Ця робота спрямована на розробку комплексного рішення, що об'єднує засоби автоматизації адміністрування та захисту AD, включаючи:

- управління користувачами та політиками безпеки через Ansible,
- автоматизоване резервне копіювання та аварійне відновлення,
- проактивний захист від атак на контролери домену,
- автоматизовані тести безпеки для оцінки вразливостей AD.

Основні завдання дослідження:

1. Здійснити аналіз сучасних методів автоматизації адміністрування та загроз безпеці Active Directory для визначення функціональних вимог до проєктованої системи.

2. Розробити архітектуру програмного комплексу, яка базується на агентсизному підході та включає модулі для управління конфігураціями, захисту від кібератак та забезпечення безперервності бізнесу.
3. Програмно реалізувати систему з використанням Ansible, створивши набір сценаріїв для автоматизації адміністрування, впровадження механізмів захисту та налаштування регулярного резервного копіювання.
4. Провести експериментальне дослідження ефективності системи шляхом автоматизованої симуляції атак (зокрема Pass-the-Hash) та верифікації процедур аварійного відновлення доменної інфраструктури.

Вимоги до програмного комплексу

- Функціональні вимоги

1. Автоматизація адміністрування AD
2. Моніторинг та безпека
3. Резервне копіювання та аварійне відновлення
4. Автоматизовані тести безпеки

- Нефункціональні вимоги

- Висока продуктивність та мінімальний вплив на роботу AD.
- Масштабованість та можливість адаптації під потреби різних організацій.
- Інтеграція з існуючими системами моніторингу безпеки.

Висновки до першого розділу:

У першому розділі були проведені теоретичні дослідження технологій необхідних для реалізації проекту, згідно теми кваліфікаційної роботи було визначене технічне завдання, яке формує необхідні вимоги для майбутніх автоматизованих рішень.

РОЗДІЛ 2. ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА ЗАСОБІВ РЕАЛІЗАЦІЇ СИСТЕМИ

2.1. Архітектура тествого стенду проекту. Мережева взаємодія.

Вибір технологічного стеку є фундаментальним рішенням, що безпосередньо впливає на гнучкість, безпеку та життєздатність проєкту. Тому нижче наведено основні засоби котрі будуть використовуватись в процесі виконання проекту.

В якості доступного гіпервізора котрий надає ключові для проекту функції було обрано Oracle VirtualBox. Його функція "Host-Only Network" дозволяє створити повністю ізольоване серидовище, що є обов'язковою вимогою для безпечного проведення тестів на проникнення. Варто зауважити, що повна ізоляція створює виклики для встановлення необхідного ПЗ. Для вирішення цієї проблеми на етапі початкового розгортання керуюча нода та робочі станції тимчасово підключаються до інтерфейсу NAT для завантаження пакетів з репозиторіїв, після чого зовнішній доступ вимикається для забезпечення чистоти експерименту з безпеки. [14]

Windows Server 2022 – використовується як основа інфраструктури через свою поширенність і чинну підтримку. Включає в себе сучасні функції AD DS, DNS, GPO, створення та керування користувачами, що дає змогу використати результати виконання проекту в подальшому впровадженні.

Ansible на базі Linux Ubuntu 22.04 виступає ядром оркестрації системи. Це рішення ґрунтується на його агентсизній архітектурі, що є фундаментальною перевагою при роботі з Windows-середовищем. На відміну від агент-орієнтованих систем, Ansible не вимагає встановлення та підтримки додаткового програмного забезпечення на керованих вузлах. Це не лише спрощує розгортання, але й суттєво зменшує поверхню атаки, оскільки відсутній агент, який можна скомпрометувати.

Ключовою перевагою Ansible є його ідемпотентний підхід. Система описує бажаний стан інфраструктури, а не послідовність імперативних команд. Це усуває ризик збоїв при повторному запуску сценаріїв: якщо об'єкт вже існує або налашту-

вання вже застосоване, Ansible просто пропустить цей крок. Це робить автоматизацію надійною та передбачуваною, що є критичним для адміністрування чутливого середовища Active Directory.

Зв'язок між керуючою нодою Ansible та вузлами Windows реалізується через нативний протокол WinRM. Це стандартна реалізація Microsoft протоколу WS-Management, що по суті є аналогом SSH для світу Windows. Саме використання WinRM уможливлює агентсизну архітектуру Ansible. Для цього проєкту свідомо обрано виключно захищений варіант WinRM, що працює на порту TCP/5986, на противагу незашифрованому HTTP/5985. Це рішення продиктоване вимогами безпеки: для початкового налаштування машин, які ще не введені в домен, найстабільнішим методом автентифікації є Basic. Передача облікових даних методом через незашифрований HTTP-канал є неприпустимою вразливістю. Тому початкове налаштування кожної Windows-машини включає створення самопідписного SSL-сертифіката та налаштування WinRM-слухача на порт 5986. Таким чином, весь трафік автентифікації та виконання команд надійно шифрується за допомогою TLS. [15]

Безпека самої системи автоматизації є настільки ж важливою, як і безпека інфраструктури, якою вона керує. Зберігання пароля адміністратора Windows у відкритому вигляді в файлах конфігурації є грубим порушенням безпеки, оскільки це робить весь репозиторій проєкту токсичним. Для вирішення цієї проблеми проєктом передбачено використання Ansible Vault. Цей вбудований механізм Ansible дозволяє створювати зашифровані файли, в яких безпечно зберігаються всі чутливі змінні, зокрема ansible_password. Сам зашифрований файл можна безпечно зберігати в системі контролю версій поруч із плейбуками. Під час запуску плейбука Ansible запитує у оператора "майстер-пароль" від Vault, розшифровує секрети в пам'яті, використовує їх для підключення до хостів і негайно знищує. Цей підхід реалізує принцип "Security by Design", гарантуючи, що облікові дані ніколи не опиняться у відкритому вигляді ні на диску, ні в мережі.

Робочі станції на базі Windows 10 є поширеними в сучасних інфраструктурних рішеннях. Вони дають змогу зручно перевірити якість автоматизованих налаштувань, таких як: створення користувачів, групових політик, розгортання ПЗ. Відповідно дозволяють провести тестування на загальну захищеність системи і вразливість до різного роду загроз.

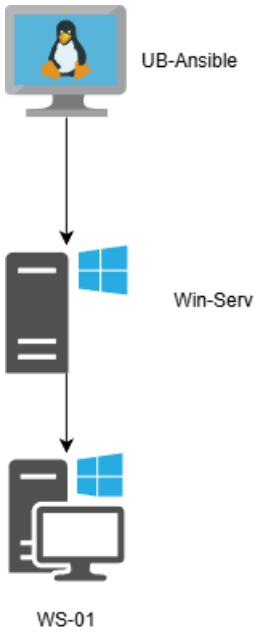


Рис. 2.1.1 Архітектура тестового стенду проекту

Для стабільного функціонування спроєктованого стенду хост-машина повинна відповідати наступним мінімальним апаратним вимогам, що обумовлені одночасним запуском чотирьох віртуальних машин:

- Процесор: підтримка віртуалізації, мінімум 4 фізичні ядра.
- Оперативна пам'ять мінімум 16 ГБ для функціонування сервера, клієнтської робочої станції та Ansible вузла
- Дисковий простір: SSD накопичувач з вільним простором не менше 60 ГБ.

Зв’язок між вузлами здійснюється за підключення до єдиного ізольованого L2-сегменту Virtual-Box “Host-Only Network”. Таке середовище ізольоване від хост-машини та зовнішніх мереж, що є критично важливим для безпеки під час за-

пуску симуляцій атак. Для стабільності та передбачуваності системи використовується статична IP-адресація. Налаштування DHCP-сервера не є релевантним для даного проєкту.

Таблиця 2.1

Віртуальна машина	IP-адреса	Маска	DNS-сервер
WIN-SERV-MRKV	192.168.56.10	/24	127.0.0.1
Ansible-MRKV	192.168.56.11	/24	192.168.56.10
WS-01	192.168.56.12	/24	192.168.56.10

2.2. Проєктування модулів адміністрування

Реалізація підсистеми адміністрування базується на принципі «Configuration as Code», де структура каталогу, облікові записи та політики безпеки описуються у вигляді декларативних сценаріїв. Це дозволяє відмовитися від ручного керування через графічні консолі, забезпечуючи стандартизацію та можливість версійного контролю змін в інфраструктурі.

Для моделювання рольової моделі доступу проєктом передбачено створення ієрархічної структури Організаційних Одиниць, що відображає типовий поділ прав у корпоративній мережі. Система автоматично розгортає три базові функціональні ролі: «Role_Users», «Role_Developers» та «Role_HelpDesk». Створення цієї структури здійснюється за допомогою плейбуків, що використовують модулі microsoft.ad.ou, microsoft.ad.user, microsoft.ad.group. [16]

Ключовим елементом проєктування є механізм управління GPO який забезпечує диференційоване розмежування доступу. Оскільки вимоги проєкту передбачають застосування обмежень не лише на основі ролі користувача, а і комп'ютера, реалізація вимагає специфічного архітектурного підходу. Ansible виступає у ролі оркестратора, що керує об'єктами GPO безпосередньо на контролері домену.

Аналіз наявних колекцій Ansible виявив недостатню гнучкість для налаштування специфічних параметрів реєстру в GPO. Тому було спроєктовано гібридний підхід: використання декларативних модулів Ansible для створення структури AD та використання імперативних PowerShell-скриптів всередині модуля win_powershell для тонкого налаштування політик безпеки.

Процес налаштування політик реалізується у кілька етапів, кожен з яких включає використання ansible.windows.win_powershell модуля. На першому етапі створюються порожні об'єкти політик для кожної ролі. Далі відбувається редагування секції «User Configuration» та «Computer Configuration» всередині об'єкта GPO, який ініціює виконання командлетів модуля GroupPolicy безпосередньо на контролері домену, [17] ін'єктуючи необхідні налаштування реєстру в об'єкт політики. Це дозволяє, наприклад, заборонити запуск PowerShell та доступ до панелі керування для звичайних користувачів, але залишити ці функції доступними для розробників. На технічному рівні Ansible реалізує це шляхом внесення змін у відповідні гілки реєстру. Наприклад, для заборони запуску конкретних додатків для групи користувачів, модуль win_regedit модифікує ключ «HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\DisallowRun» , додаючи туди виконуваний файл powershell.exe. [18]

На фінальному етапі модуль здійснює прив'язку наповнених об'єктів GPO до відповідних OU, де розташовані облікові записи користувачів. Такий підхід гарантує, що політики безпеки «слідують» за користувачем незалежно від того, на якій робочій станції він проходить автентифікацію. Це забезпечує гнучкість управління та високий рівень безпеки, унеможливлюючи використання системних інструментів для підвищення привілеїв або закріплення в системі у випадку компрометації облікового запису рядового користувача.

У таблиці 2.2 продоменстровано набір функції котрі буде обмежено для кінцевих користувачів домену за допомогою GPO.

Таблиця 2.2

Обмеження	Role_Managers	Role_Developers	Role_HelpDesk
Панель керування та налаштування	-	-	+
Доступ CMD та PowerShell	-	+	+
Доступ до regedit	-	-	+
Встановлення ПЗ (MSI)	-	-	+
Меню «Виконати»	-	+	-
USB накопичувачі	-	-	+

2.3. Проєктування підсистеми безпеки, моніторингу та забезпечення безперервності

У контексті Active Directory резервне копіювання є критичним елементом стратегії безпеки, оскільки втрата контролера домену внаслідок програмного збою або атаки типу ransomware може призвести до повної зупинки функціонування інформаційної системи. Проєктом передбачено відмову від ручного створення копій на користь автоматизованого процесу.

Реалізація базується на використанні нативного механізму Windows Server Backup, який забезпечує створення консистентної тіньової копії тома та архівування стану системи, що включає базу даних NTDS, реєстр та папку SYSVOL. Ansible у цьому процесі виконує роль оркестратора. Спеціалізований плейбук «backup_ad.yml» використовує модуль «ansible.windows.win_powershell» для ініціалізації процесу резервного копіювання на локальний або мережевий носій за допомогою інструментарію wbadmin або командлетів Backup-ADDS.

Для гарантування регулярності виконання цієї процедури використовується модуль «community.windows.win_scheduled_task». Він ідемпотентно створює за-

вдання у планувальнику Windows на контролері домену, визначаючи розклад, привілеї виконання та параметри ротації архівів. Такий підхід мінімізує людський фактор, гарантуючи наявність актуальних точок відновлення без необхідності щоденного втручання адміністратора. Важливо зазначити, що в рамках тестового стенду резервне копіювання здійснюється на окремий віртуальний диск, підключений до контролера домену. У реальному виробничому середовищі Ansible налаштовував би збереження бекапів на віддалене мережеве сховище або в хмарне сховище для захисту від фізичного виходу сервера з ладу.

Основним вектором атаки, що розглядається у проєкті, є компрометація облікових даних та подальший Lateral Movement з використанням техніки Pass-the-Hash. Для нейтралізації цієї загрози проєктується впровадження технології LAPS. [19] Оскільки ручне розгортання LAPS є трудомістким процесом, Ansible автоматизує повний цикл його імплементації: віддалене встановлення MSI пакету, розширення схеми Active Directory для зберігання атрибутів паролів, встановлення клієнтських агентів на робочі станції та налаштування відповідних GPO для примусової ротації паролів локальних адміністраторів. Після впровадження LAPS виникає потреба у легітимному отриманні паролів адміністратором. Ansible вирішує це завдання за допомогою прямих LDAP-запитів, дозволяючи авторизованому персоналу отримати актуальний пароль локального адміністратора конкретної машини безпосередньо у виводі плейбука, що значно спрощує процес адміністрування. [20]

Паралельно з превентивними заходами реалізується система моніторингу подій безпеки. Ansible використовується для стандартизації налаштувань аудиту на всіх вузлах мережі. За допомогою powershell командлетів активуються розширені політики реєстрації подій, критичних для виявлення атак, зокрема подій входу в систему, використання привілеїв.

Завершальним етапом проєктування є розробка механізму верифікації, що діє за принципом «Trust but Verify». На відміну від попередніх модулів, де Ansible налаштовує Windows-хости, у цьому сценарії керуючий вузол виступає у ролі платформи для тестування на проникнення.

Проєктується набір плейбуків «simulate_attack.yml», які оркеструють запуск спеціалізованих інструментів безпеки безпосередньо з середовища Linux проти цільової інфраструктури Windows. Ansible вузол використовує Python-бібліотеки Impacket для емуляції дій зловмисника, що дозволяє не розгортати окрему Kali Linux машину, а проводити аудит прямо з керуючого вузла. Інструмент Nmap перевіряє стан цільових для атаки портів. Сценарій валідації передбачає спробу виконання атаки Pass-the-Hash: використовуючи відомий хеш пароля локального адміністратора з WS-01 здійснюється спроба віддаленого підключення до іншої станції. NTLM хеш локального адміністратора з'ясовується за допомогою застосунку Mimikatz. [23]

Успішна реалізація захисних механізмів, описаних вище, зокрема LAPS, повинна призвести до невдачі цієї симуляції. Таким чином, Ansible дозволяє не лише побудувати захищену інфраструктуру, але й надати емпіричні докази її стійкості до реальних кіберзагроз, перетворюючи процес тестування безпеки на регулярну, автоматизовану процедуру.

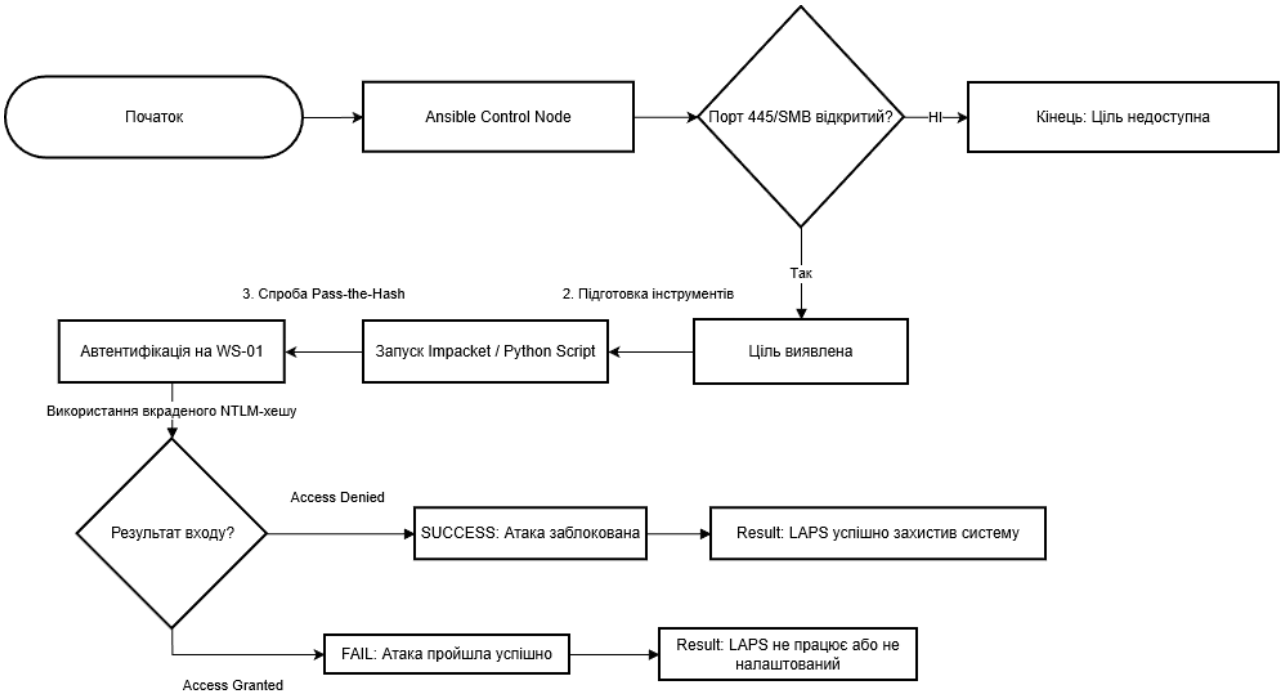


Рис 2.2.1 Схема симуляції атаки

Висновки до другого розділу

У розділі спроєктовано безпечну архітектуру системи автоматизації Active Directory на базі Ansible та протоколу WinRM HTTPS. Розроблено модель ізольованого тестового середовища та визначено логіку функціональних модулів для управління інфраструктурою, адміністрування, автоматизації резервного копіювання та впровадження політик безпеки. Запропоновано механізм автоматизованої валідації захисту шляхом симуляції атак, що дозволяє емпірично підтвердити ефективність розробленого рішення перед його практичною реалізацією.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СИСТЕМИ

3.1. Підготовка середовища та базове налаштування

Реалізація проєкту розпочалася з підготовки віртуалізованого середовища на базі гіпервізора Oracle VirtualBox, який забезпечує необхідний рівень ізоляції та керованості ресурсів. Для побудови мережевої топології було створено віртуальний адаптер типу «Host-Only Ethernet Adapter», що дозволяє організувати закритий контур обміну даними між віртуальними машинами без прямого доступу до зовнішньої мережі Інтернет. Це рішення гарантує чистоту експерименту при проведенні тестів на проникнення та унеможливлює витік трафіку за межі лабораторного стенду. Проте для забезпечення встановлення потрібних пакунків та колекцій робоча станція Linux використовуватиме NAT адаптер з доступом до інтернету. [21]

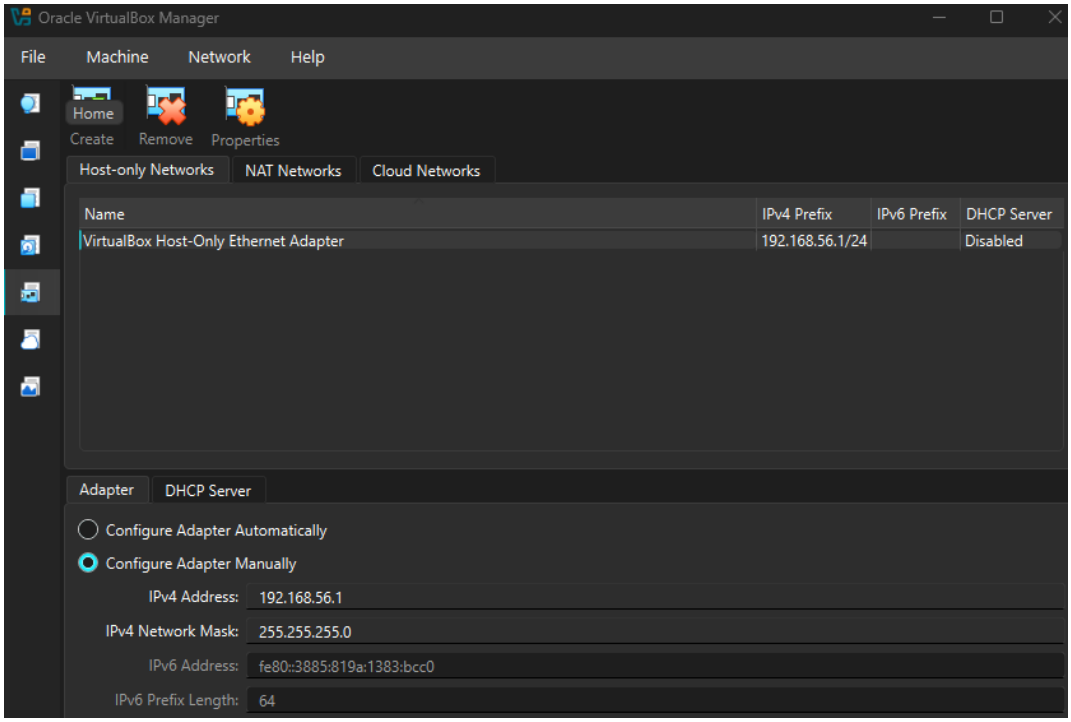


Рис 3.1.1 Створений Host-Only Ethernet Adapter

Після успішного розгортання операційних систем було проведено налаштування мережевої взаємодії та підготовку хостів до централізованого управління. Для забезпечення стабільності з'єднань та коректної роботи файлів інвентаризації Ansible усім віртуальним машинам було присвоєно статичні IP-адреси в межах підмережі 192.168.56.0/24, що виключає залежність від DHCP-сервісів та потенційні

конфлікти адрес при перезавантаженні стенду. Дані налаштування впроваджуються вручну, адже на даному етапі ще не налаштована взаємодія усіх вузлів.

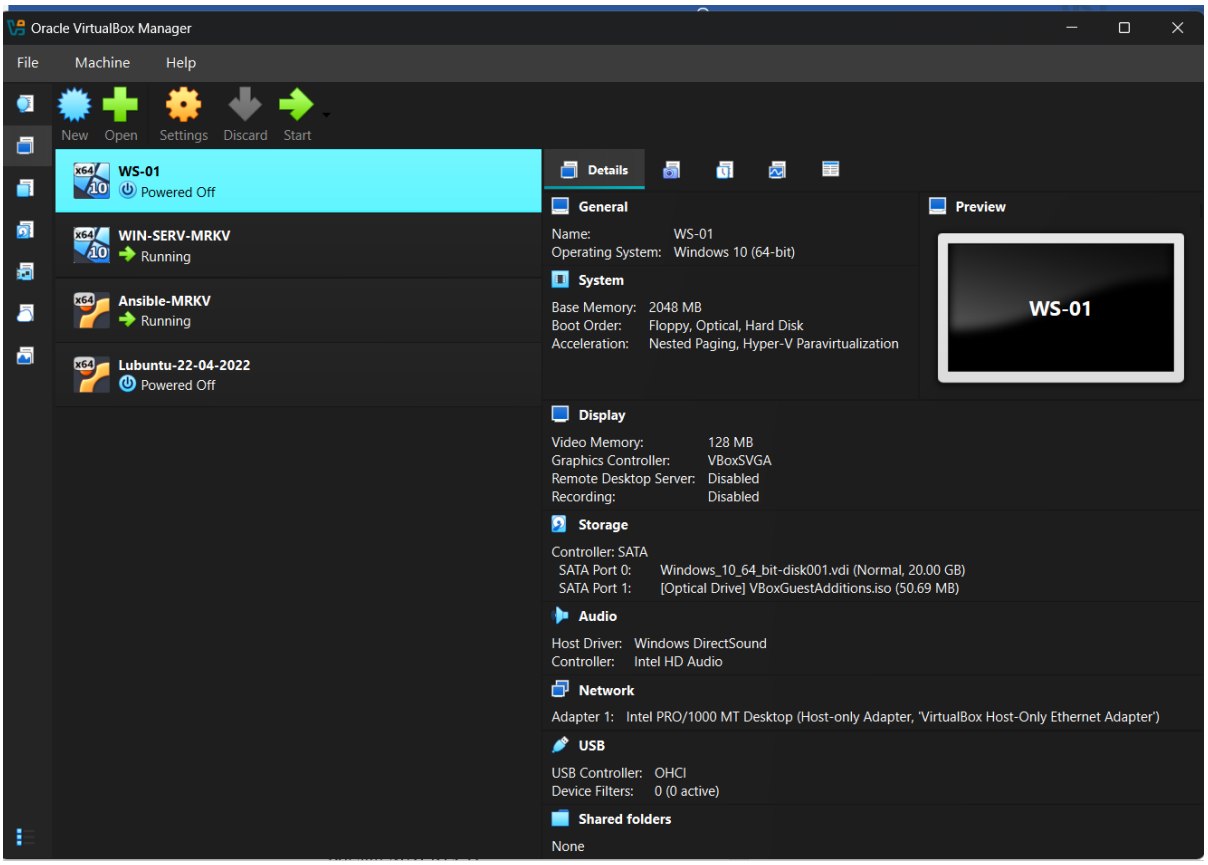


Рис. 3.1.2 Створені віртуальні машини для стенду проекту

Наступним важливим етапом стала підготовка Windows-хостів. Для забезпечення безпечного каналу управління без розгортання повноцінної PKI-інфраструктури було використано PowerShell командлети для налаштування WinRM + HTTPS. За допомогою них створюється самопідписний SSL-сертифікат, використовуючи поточне ім'я хоста, створюється Listener на порту 5986 та додаються виключення у брандмауер. Це дозволить Ansible встановлювати зашифровані з'єднання з вузлами, використовуючи Basic-автентифікацію всередині TLS-тунелю, що є безпечним та достатнім методом для початкового розгортання домену. [22]

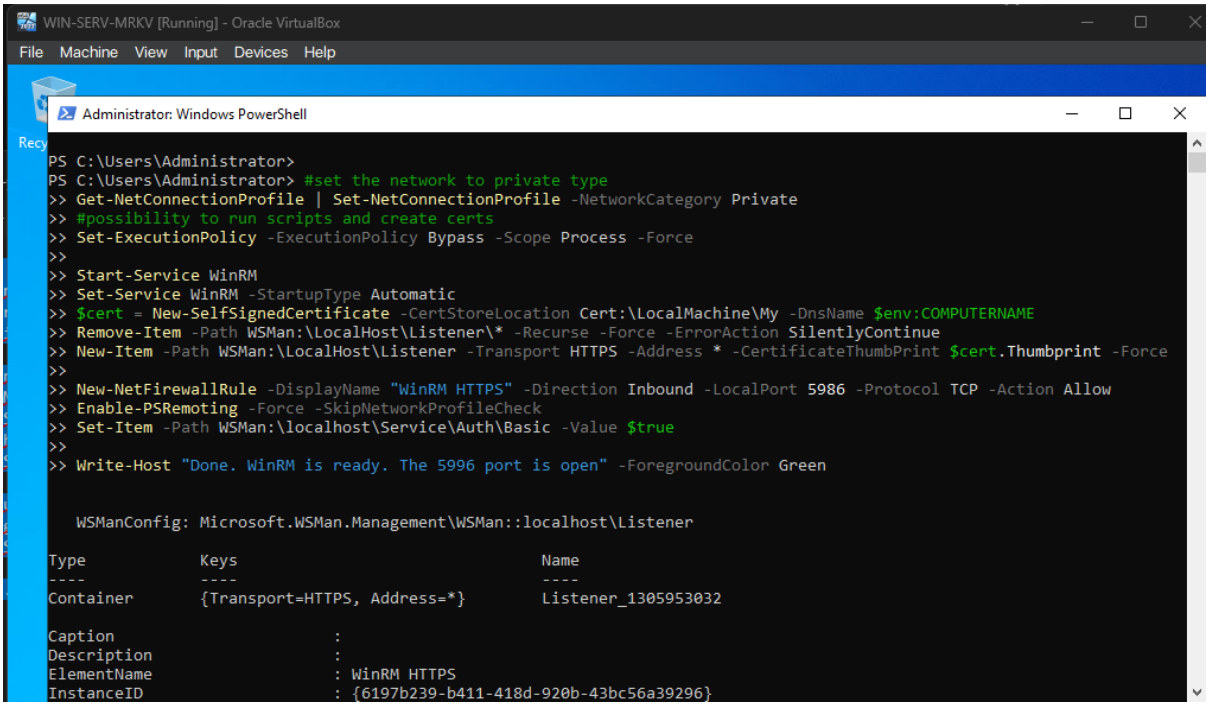


Рис. 3.1.3 Виконання командлетів для налаштування WinRM та відкриття 5986 порту на WIN-SERV-MRKV

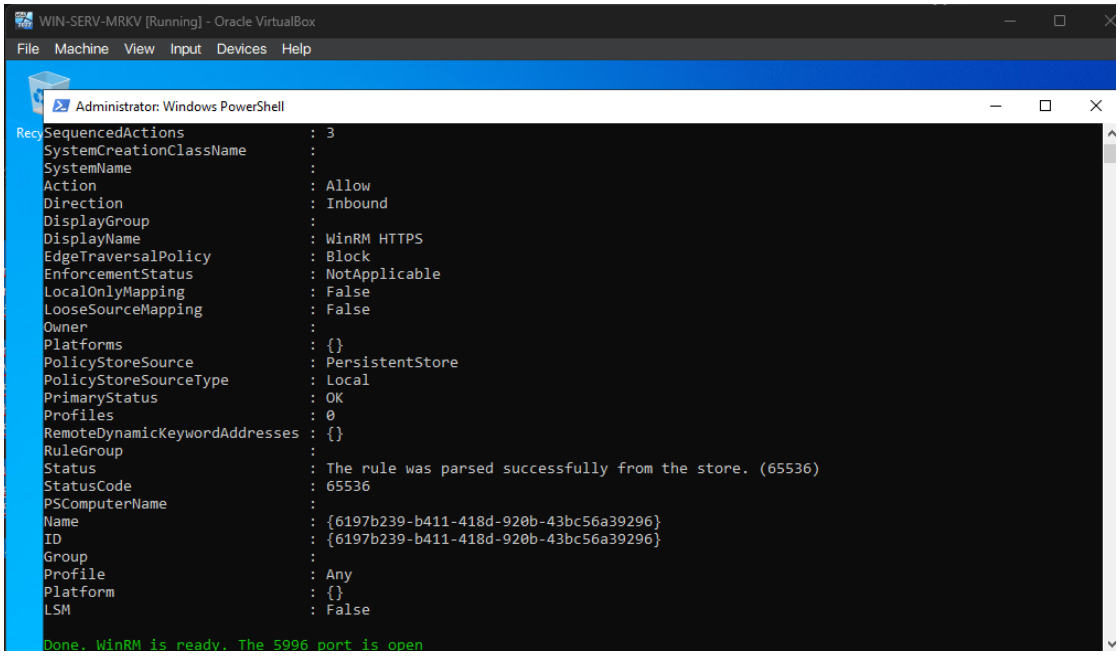


Рис. 3.1.4 Результат виконання командлетів на WIN-SERV-MRKV

Наступним етапом стало налаштування керуючої ноди. На базі ОС Ubuntu було розгорнуто середовище Ansible. Оскільки Ansible за замовчуванням не містить специфічних модулів для керування Windows-інфраструктурою, було виконано інсталяцію необхідних залежностей. Зокрема, встановлено Python-бібліотеку «pywinrm», що забезпечує реалізацію протоколу WS-Management клієнта.

Для розширення функціональних можливостей системи було завантажено та інстальовано офіційні колекції модулів із репозиторію Ansible Galaxy:

- ansible.windows / community.windows – містить базові модулі для взаємодії з ОС та її функціями, як от зміна імені комп'ютера, зміна налаштувань мережевої адресації, виконання команд рядку PowerShell
- microsoft.ad – містить модулі для реалізації адміністрування домену, налаштування його компонентів та складових;

Файлова структура проєкту організована згідно зі стандартами, де конфігураційний файл ansible.cfg визначає параметри підключення, а файл інвентаризації описує цільові хости.

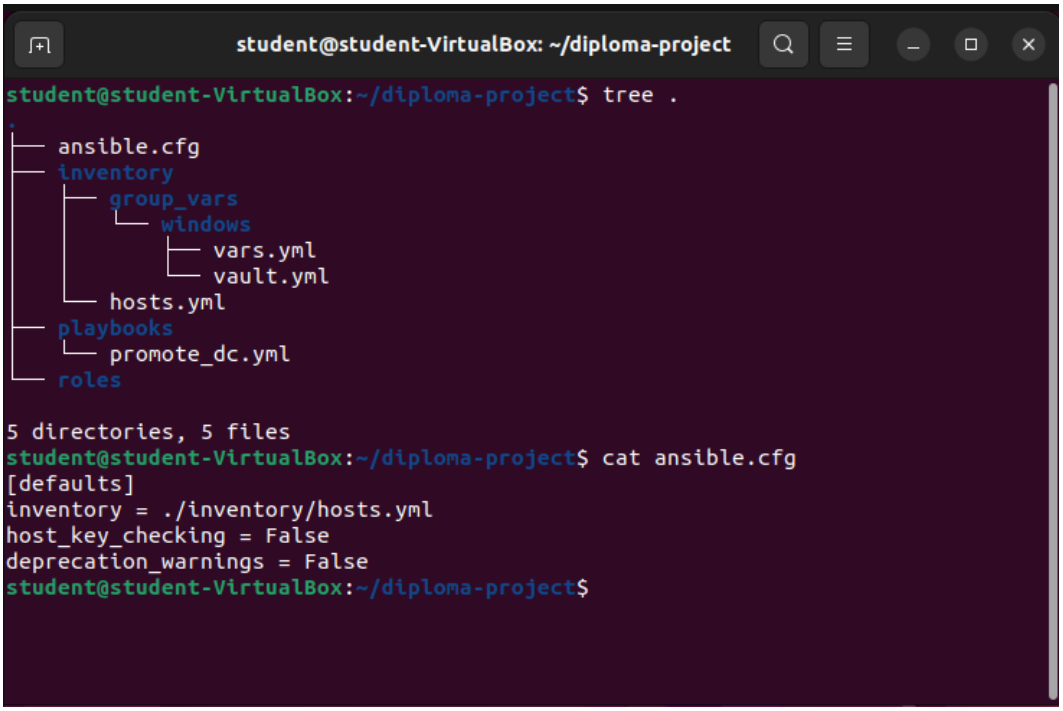


Рис. 3.1.5 Структура файлів проєкту Ansible та конфігурація інвентаризації

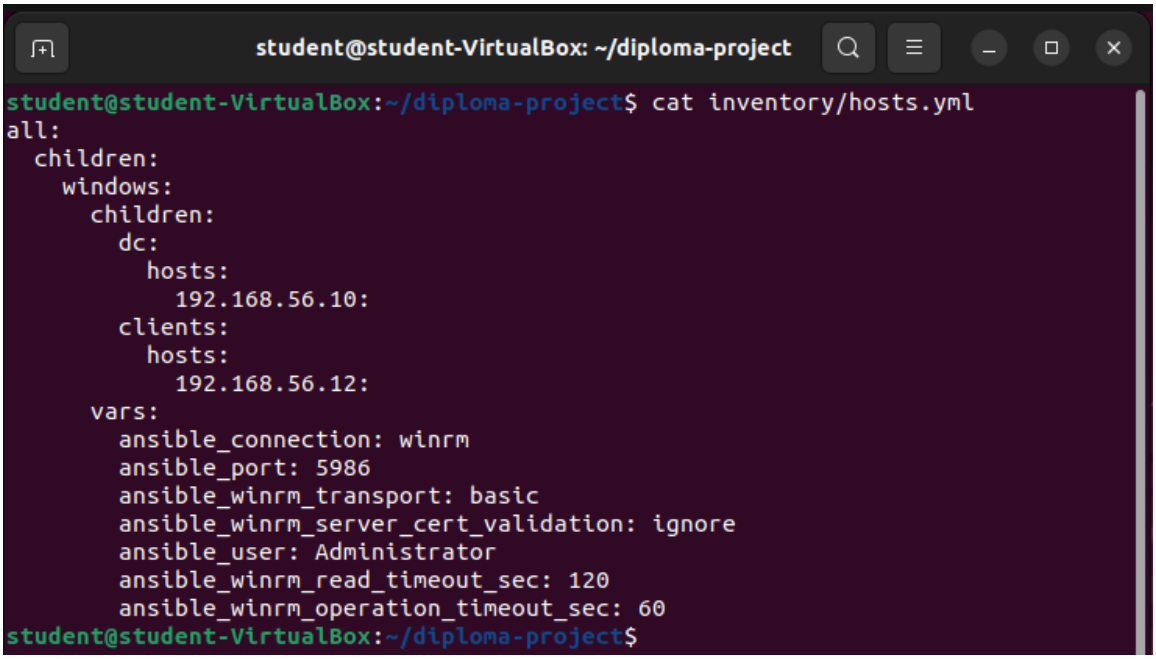


Рис. 3.1.6 Структура файлів проєкту Ansible та конфігурація інвентаризації

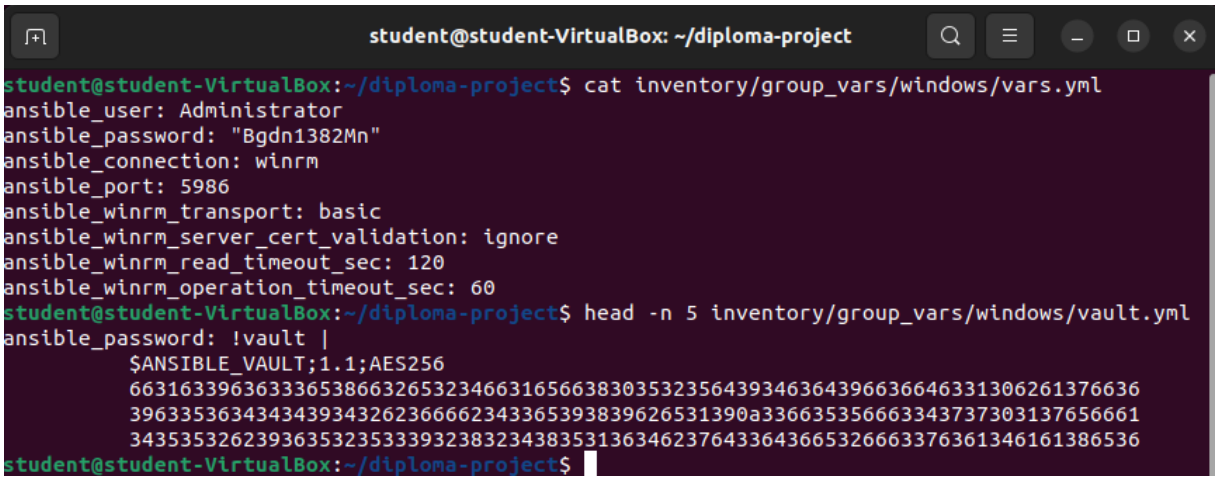


Рис. 3.1.7 Конфігурація підключення WinRM та використання Ansible Vault для захисту паролю

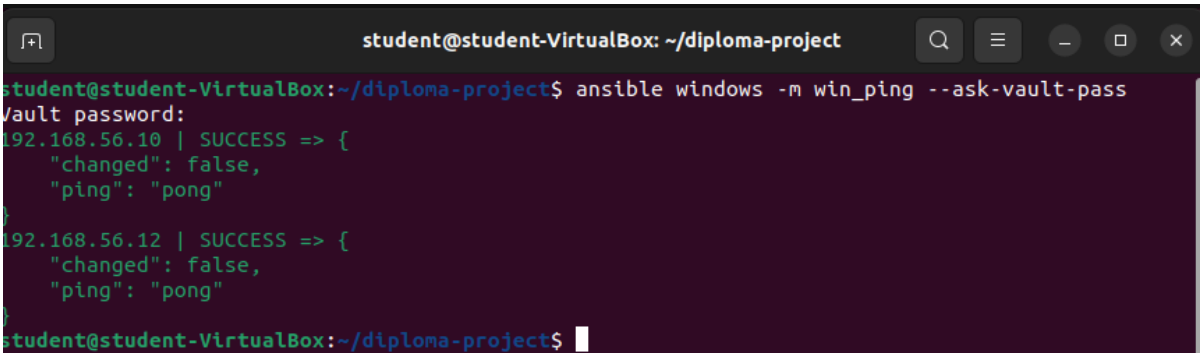


Рис. 3.1.8 Результат перевірки доступності Windows-хостів через модуль win_ping

3.2 Основні засоби адміністрування

Реалізація центрального вузла інфраструктури виконувалася відповідно до принципів Infrastructure as Code, що дозволяє гарантувати повторюваність конфігурації та уникнути помилок ручного налаштування. Враховуючи архітектурні зміни в екосистемі Ansible та припиненням підтримки застарілих модулів у нових версіях ядра, для керування службами каталогів було імплементовано офіційну колекцію microsoft.ad для забезпечення повної сумісності із Windows Server 2022 та коректної обробки специфічних параметрів лісу Active Directory.

Процес розгортання було автоматизовано за допомогою розробленого плейбука «promote_dc.yml», який застосовується до виділеної групи хостів. Алгоритм виконання сценарію складається з послідовності етапів: спершу модуль ansible.windows.win_hostname виконує стандартизацію мережевого імені сервера, встановлюючи його як SRV-MRKV. Наступним кроком модуль ansible.windows.win_feature інсталює необхідні бінарні файли ролі AD DS разом із засобами віддаленого адміністрування. Ключовим етапом є ініціалізація нового лісу за допомогою модуля microsoft.ad.domain, який створює домен corpmarkeiev.net, автоматично налаштовує інтегровані зони DNS та встановлює пароль режиму відновлення служб каталогів, використовуючи захищене значення зі сховища Ansible Vault.

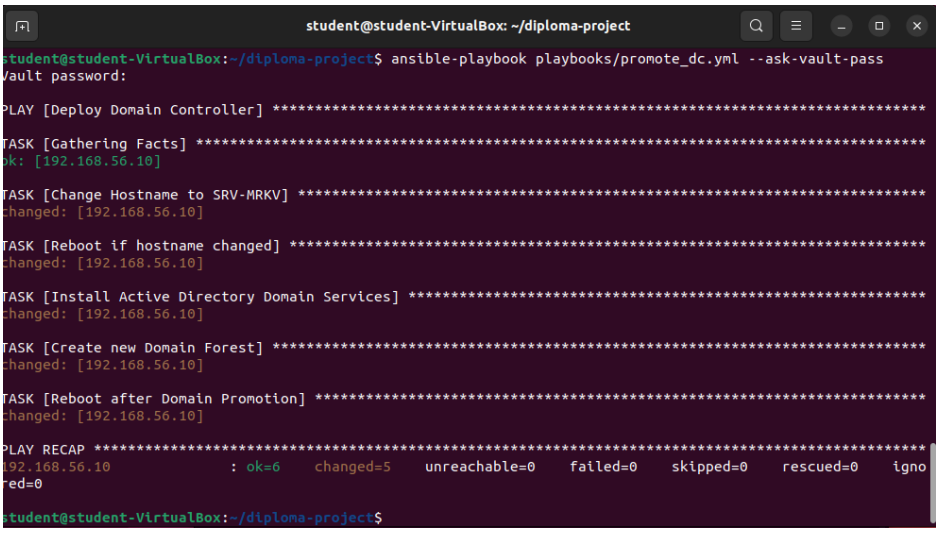


Рис. 3.2.1 Успішне виконання плейбуку promote_dc.yml

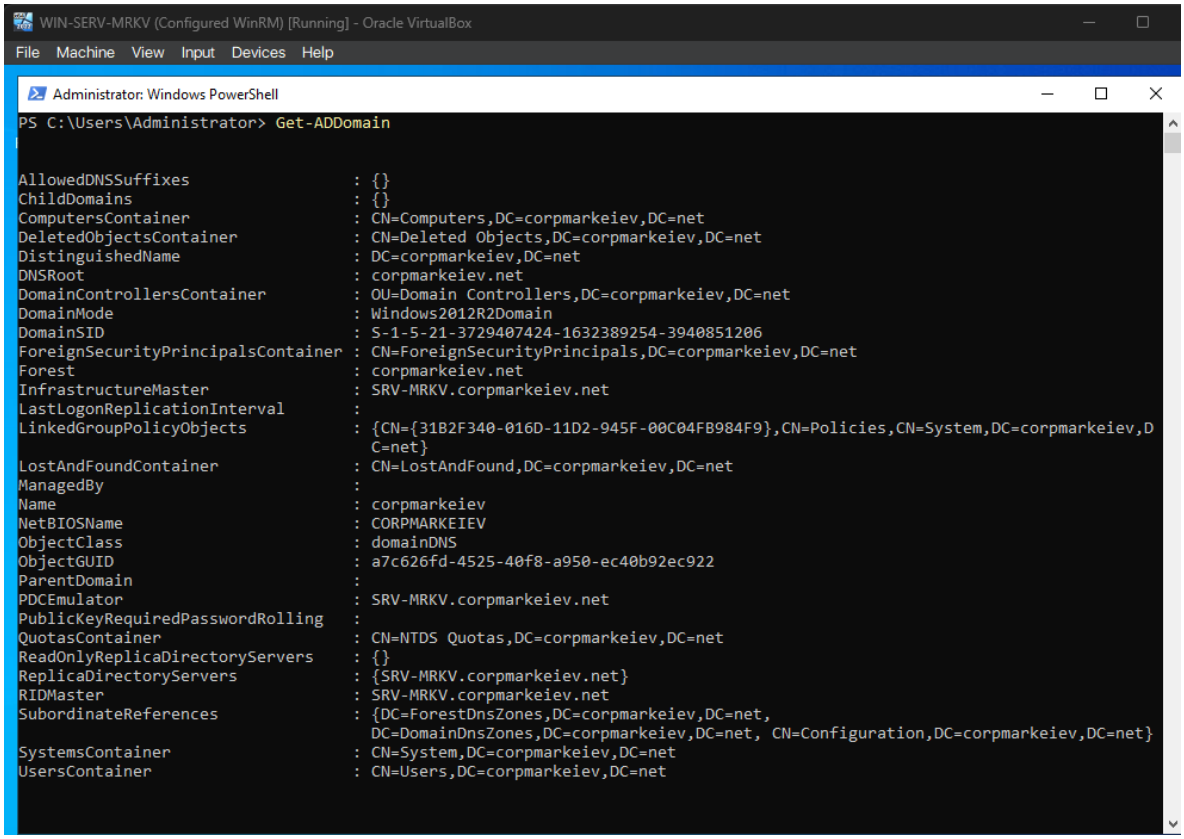


Рис. 3.2.2 Верифікація параметрів створеного домену Active Directory

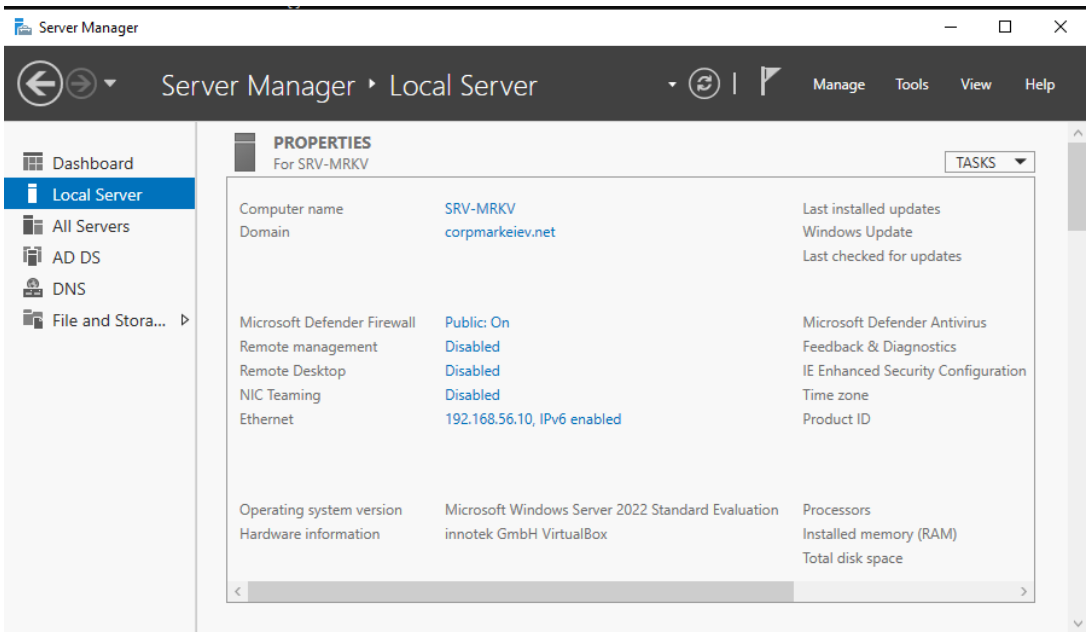


Рис. 3.2.3 Стан Windows Server 2022 після автоматизованого присвоєння ролі контролера домену

Після успішного розгортання контролера домену наступним кроком стала інтеграція клієнтських робочих станцій у єдиний простір довіри Active Directory.

Для автоматизації цього процесу використовується плейбук domain_join.yml, який застосовується до групи хостів, визначеної у файлі інвентаризації.

Алгоритм роботи плейбука включає два критичні етапи:

Налаштування DNS-резолвінгу: Оскільки коректна робота протоколів Kerberos та LDAP залежить від правильного розпізнавання SRV-записів домену, модуль ansible.windows.win_dns_client автоматично змінює налаштування мережевих адаптерів робочих станцій, встановлюючи IP-адресу контролера домену як основного DNS-сервера. Без цього кроку клієнти не зможуть виявити контролер домену в мережі.

Приєднання до домену: Модуль microsoft.ad.membership ініціює процедуру входження в домен corpmarkeiev.net. Для авторизації цієї операції сценарій використовує облікові дані адміністратора, які безпечно передаються з зашифрованого сховища Ansible Vault.

Завершальним етапом є автоматичне перезавантаження робочих станцій, що необхідно для застосування змін у системному реєстрі та зміни контексту безпеки з локального на доменний.

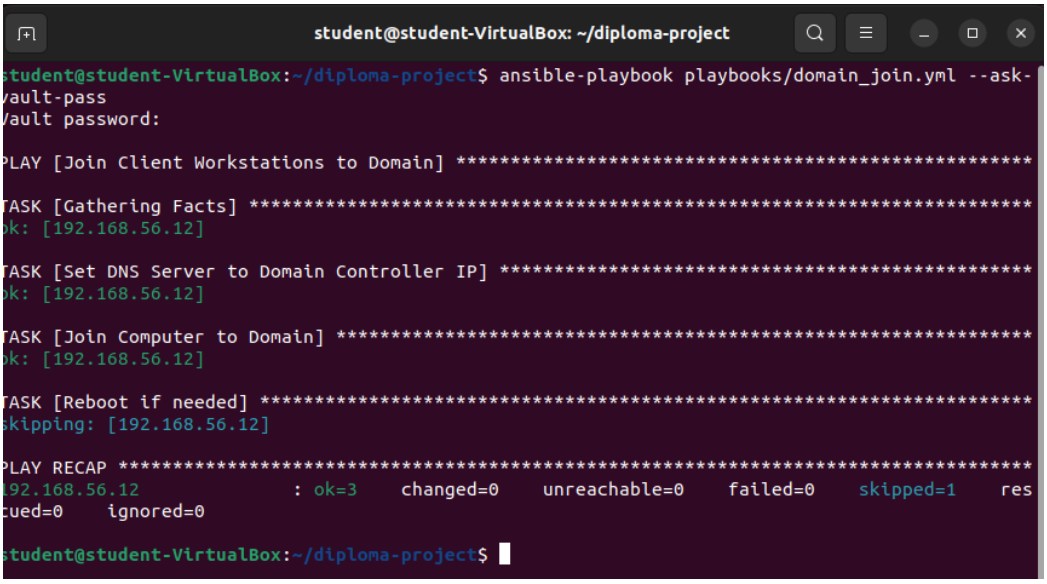


Рис. 3.2.4 Успішне виконання плейбуку domain_join.yml

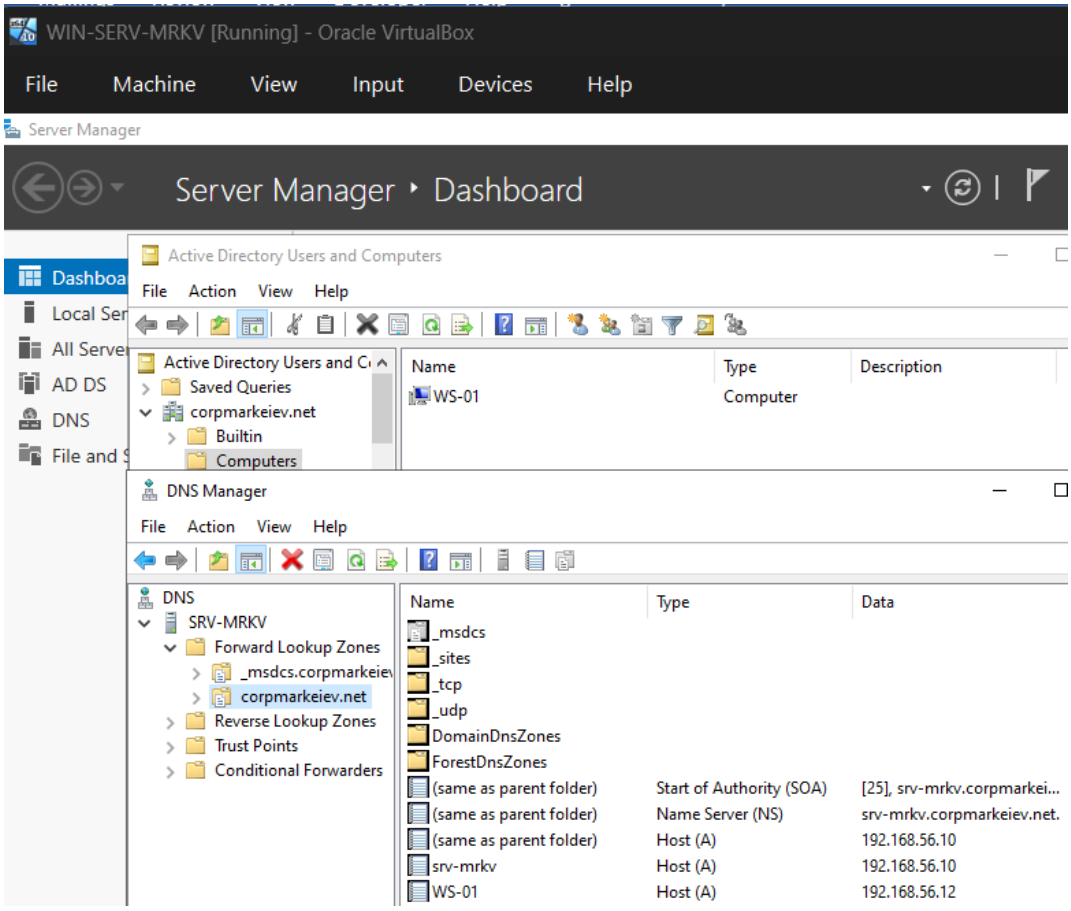


Рис. 3.2.5 Наявність доданих станції у Active Directory Users and Computers та серед записів служби DNS

Наступним кроком після розгортання контролера домену є наповнення Active Directory об'єктами: організаційними одиницями, групами безпеки та обліковими записами користувачів. Ручне створення цих об'єктів є рутинним та може призвести до помилок, тому цей процес було автоматизовано за допомогою сценарію Ansible ad_structure.yml. Він включає в себе використання одразу декількох модулів для роботи з Active Directory: microsoft.ad.ou, microsoft.ad.group, microsoft.ad.user.

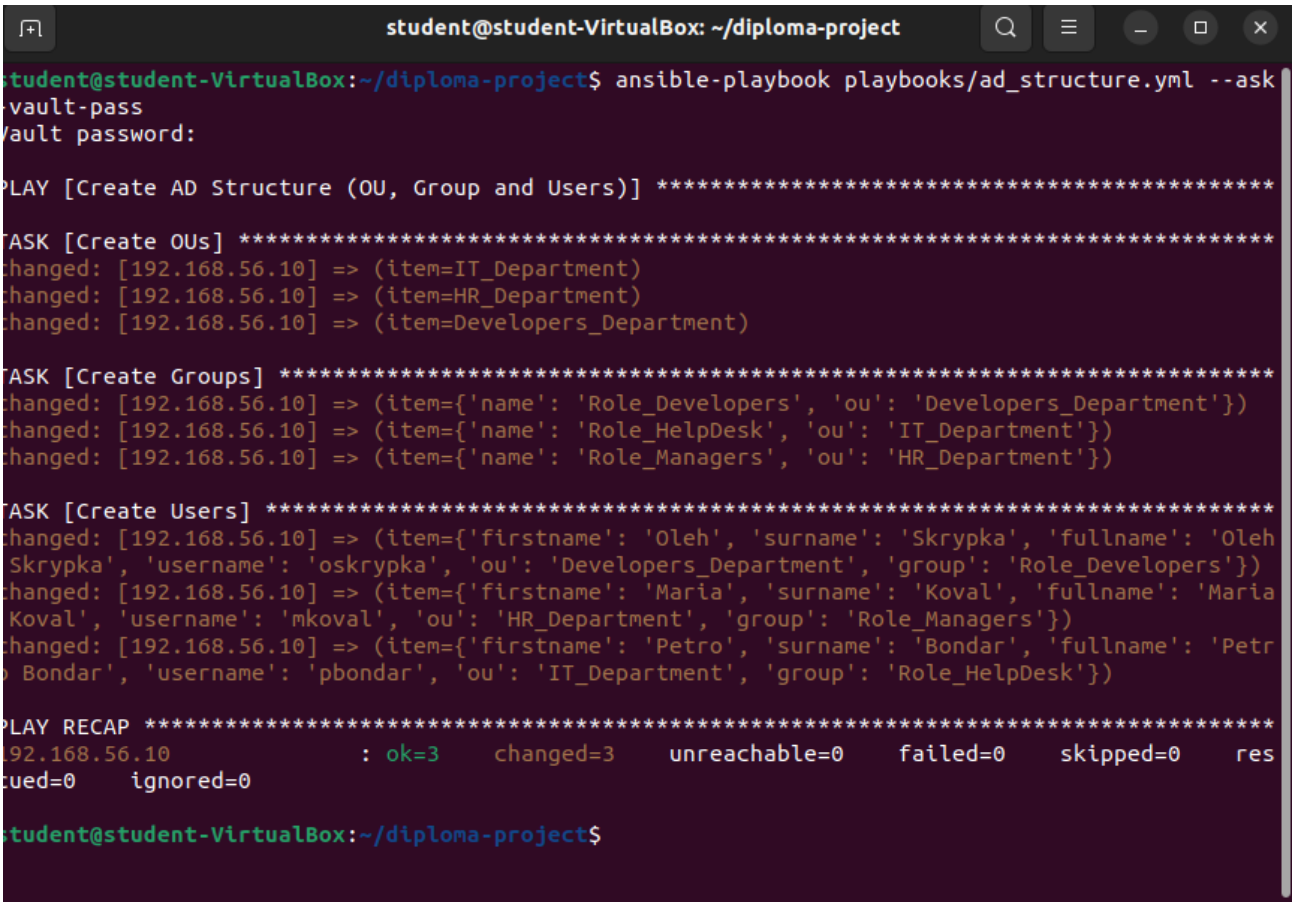


Рис. 3.2.6 Успішне виконання плейбуку ad_structure.yml

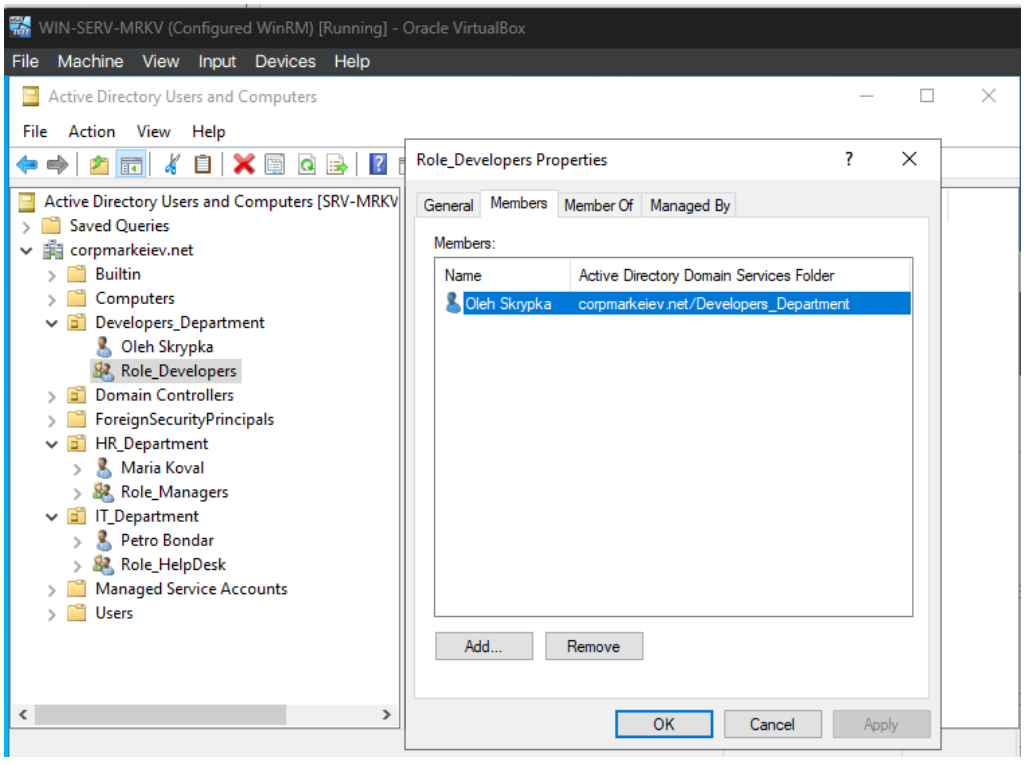


Рис. 3.2.7 Наявність створених одиниць у Active Directory Users and Computers

Після створених OU та розподілення користувачів у них було створено групові політики, котрі обмежуватимуть доступ користувачів до системних інструментів та функцій в залежності від їх приналежності до відповдних груп. Так як колекції Ansible не містять конкретного модулю для безпосередньої роботи з GPO Microsoft Active Directory, то плейбук «gpo_setup.yml» використовує модуль «ansible.windows.win_powershell» та відповідні командлети рядку PowerShell для створення та лінкування GPO до об'єктів всередині AD. Принцип їх роботи полягає в тому, аби застосовувати політики в залжності від місця робочої станції в OU та користувача котрий виконує вхід. Таким чином функції по типу доступу до командного рядка, панелі керування блокуються в залежності від користувача, а підключення USB пристроїв та встановлення ПЗ в залежності від того де знаходиться робоча станція.

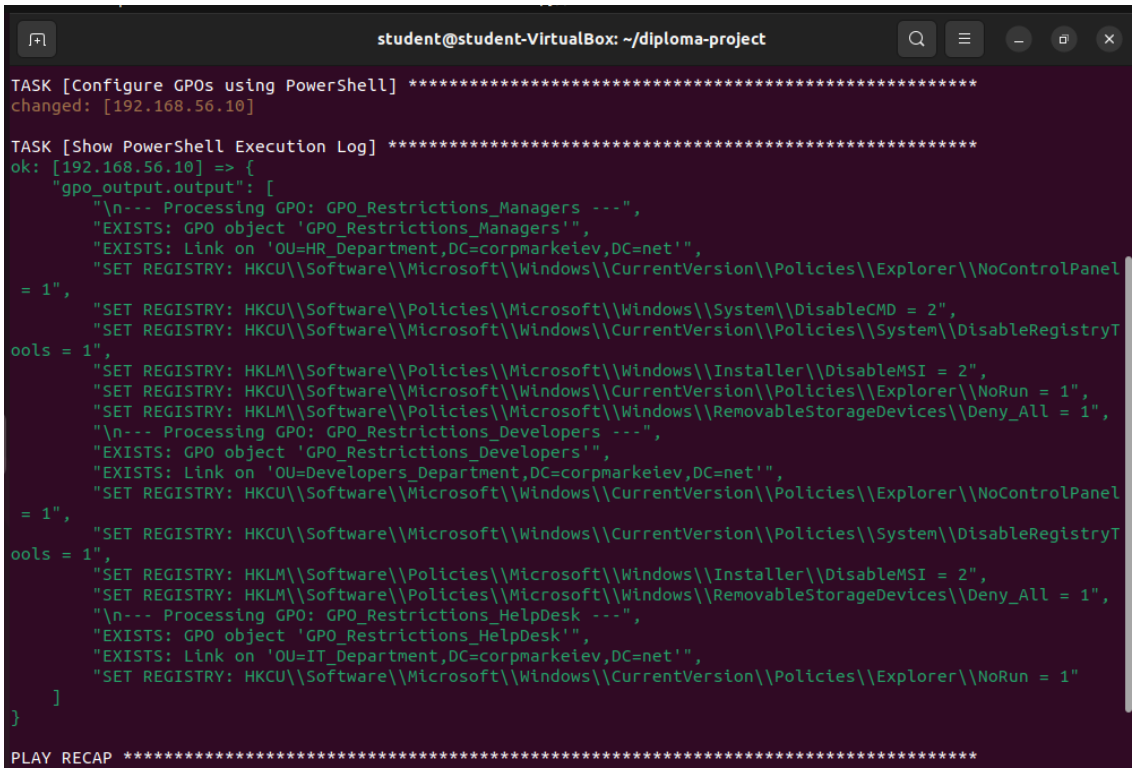


Рис. 3.2.8 Виконання плейбуку gpo_setup.yml на вузлі Ansible

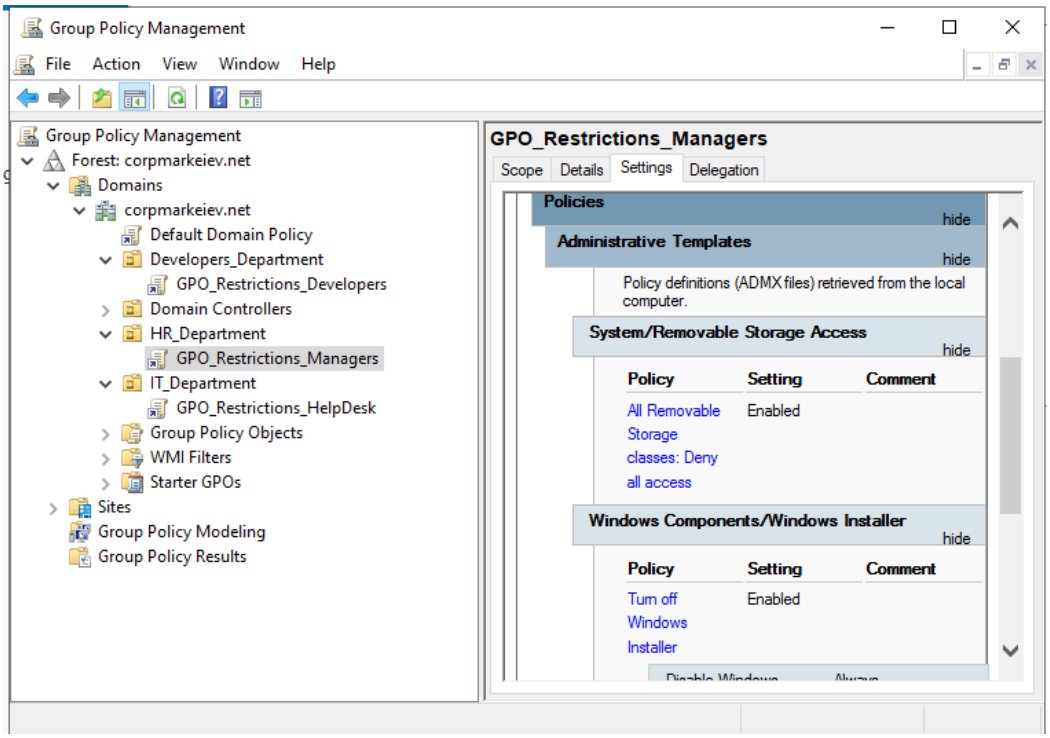


Рис. 3.2.9 Створені GPO у відповідних OU

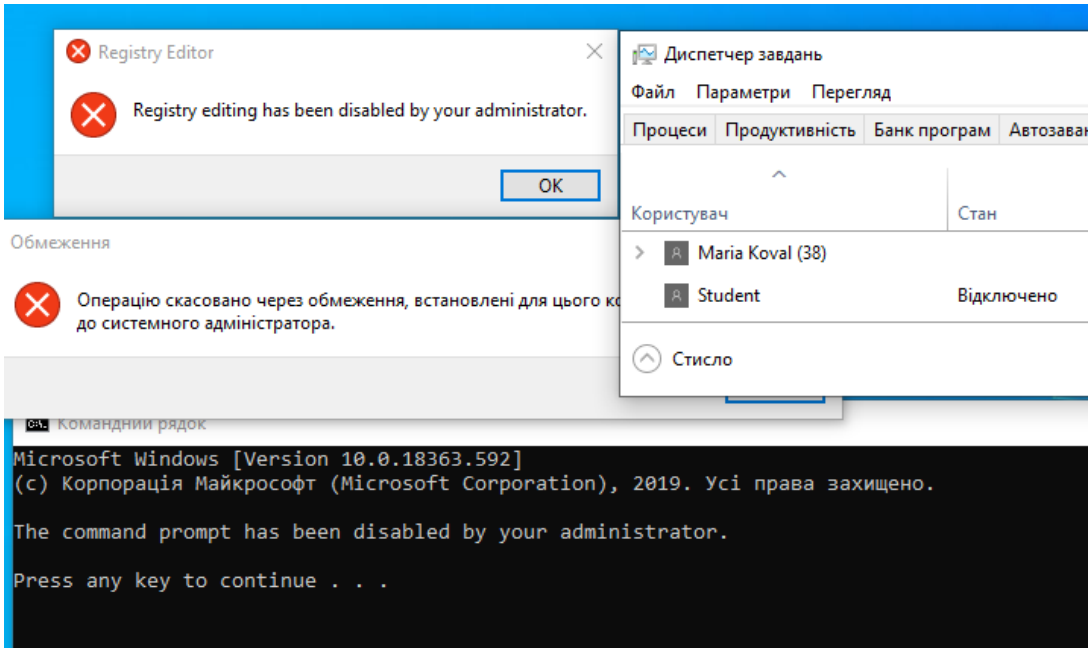


Рис. 3.2.10 Обмеження доступу до панелі керування

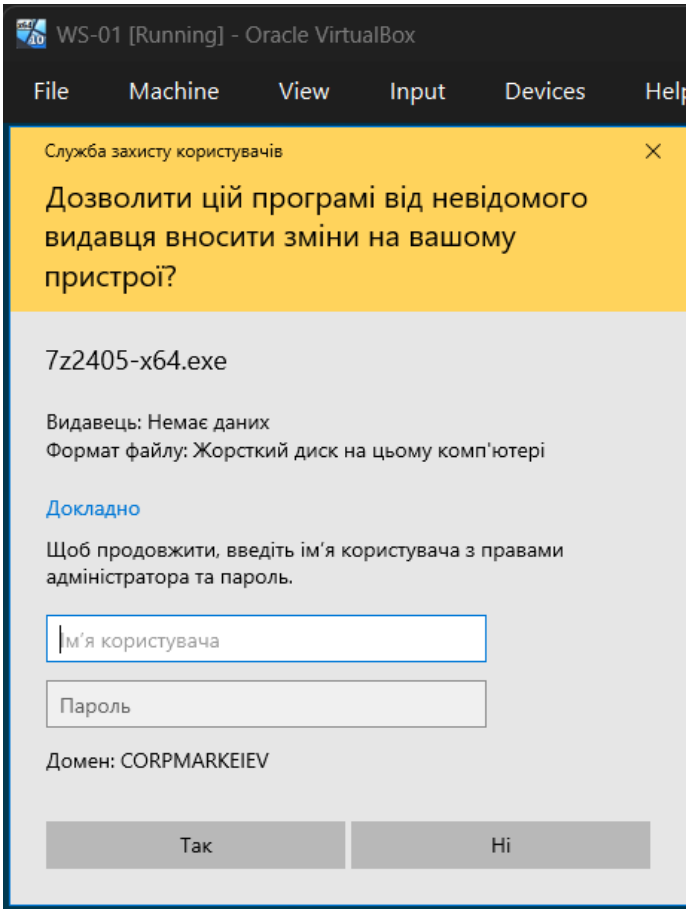


Рис. 3.2.11 Обмеження встановлення ПЗ

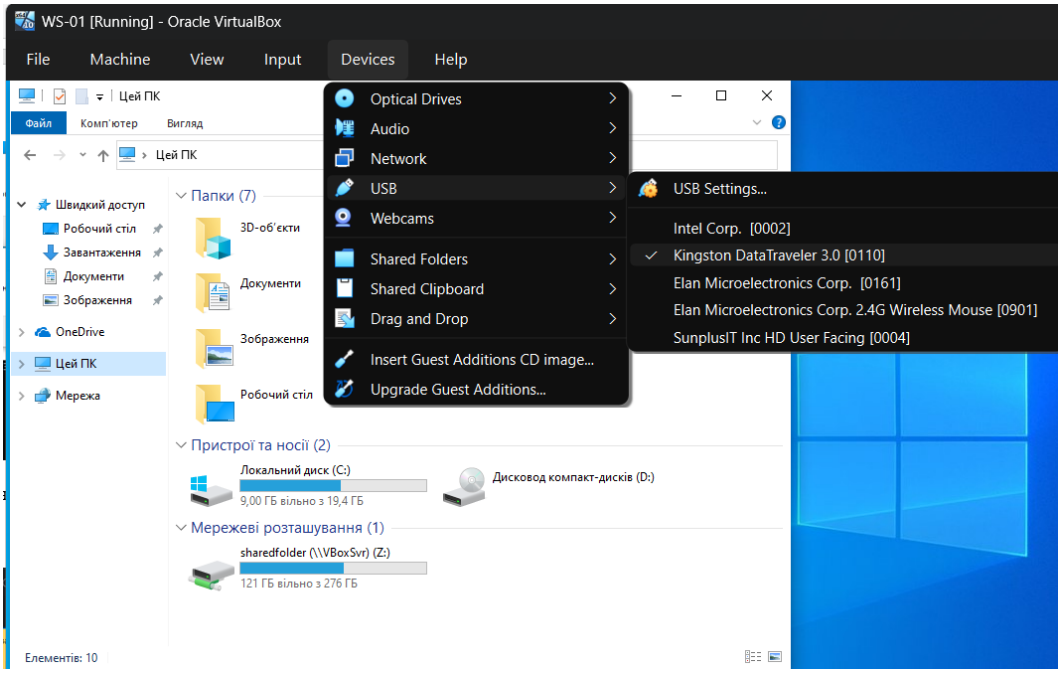


Рис. 3.2.12 Обмеження підключення USB пристроїв

3.3 Реалізація підсистеми безпеки

Плейбук win_laps.yml реалізує повний цикл автоматизованого розгортання та налаштування Microsoft LAPS у домені corpmarkeiev.net за допомогою Ansible, який працює з Linux-контролера і керує Windows-серверами та робочими станціями через WinRM. Спочатку плейбук підключається до всіх хостів групи windows, створює на них тимчасовий каталог C:\Temp, копіює туди інсталяційний пакет LAPS і виконує його встановлення. На клієнтських станціях установлюється лише клієнтський компонент LAPS, що відповідає за взаємодію з груповими політиками та запис паролів у Active Directory, тоді як на контролері домену той самий пакет інсталюється разом з інструментами адміністрування, включаючи PowerShell-модуль AdmPwd.PS та шаблони політик. Після цього на всіх робочих станціях автоматично створюється окремий локальний обліковий запис LAPSAdmin, його членство додається до локальної групи Administrators, а початковий пароль задається як тимчасовий, адже надалі ним керуватиме LAPS.

Далі робота плейбуку зосереджується на контролері домену SRV-MRKV. На час виконання інсталяційних дій та оновлення клієнтів тимчасово послаблюються раніше створені політики обмеження встановлення програмного забезпечення: у трьох політиках GPO_Restrictions_Developers, GPO_Restrictions_HelpDesk та GPO_Restrictions_Managers змінюється параметр DisableMSI у гілці реєстру політик Windows Installer, що дозволяє коректно відпрацювати розгортанню клієнта LAPS по доменному середовищу. Далі через модуль AdmPwd.PS виконується розширення схеми Active Directory для підтримки атрибутів, у яких LAPS зберігає пароль та час його зміни, а також налаштовуються необхідні делегування прав для комп’ютерних об’єктів та групи Domain Admins у підрозділах HR_Department, Developers_Department і IT_Department, що дозволяє комп’ютерам оновлювати власний пароль, а адміністраторам – його читати.

Наступний етап плейбуку полягає в автоматичному створенні групової політики GPO_LAPS_Workstations і її повному налаштуванні за допомогою PowerShell

без ручної роботи в графічному інтерфейсі. У розділі реєстру політик, призначеному для LAPS, задаються ключові параметри: увімкнення механізму керування локальними паролями, використання облікового запису LAPSAdmin як керованого локального адміністратора, інтервал ротації пароля у сім днів, довжина пароля у шістнадцять символів, вимога максимальної складності, заборона повторного використання попереднього пароля та захист від перевищення максимального строку дії. У цьому ж GPO вноситься запис у розділ Winlogon для приховання облікового запису LAPSAdmin з екрана входу в систему, що зменшує видимість службових облікових записів для звичайних користувачів. Після створення політика автоматично прив’язується до трьох організаційних підрозділів з робочими станціями, забезпечуючи єдині параметри LAPS для відповідних комп’ютерів, незалежно від підрозділу. Завершальним кроком плейбуку є відновлення вихідних обмежень на встановлення програмного забезпечення в політиках GPO_Restrictions_Developers, GPO_Restrictions_HelpDesk та GPO_Restrictions_Managers, а також виконання контрольного запиту до Active Directory для комп’ютера WS-01 з виведенням атрибутів, зарезервованих під LAPS, що дозволяє продемонструвати готовність доменної інфраструктури до централізованого керування локальними адміністративними паролями. Проте слід зазначити що через нюанси роботи LAPS у виводі результату роботи плейбуку атрибут ms-Mcs-AdmPwd не містить жодного значення, через те що на ротацію паролю потрібен час.

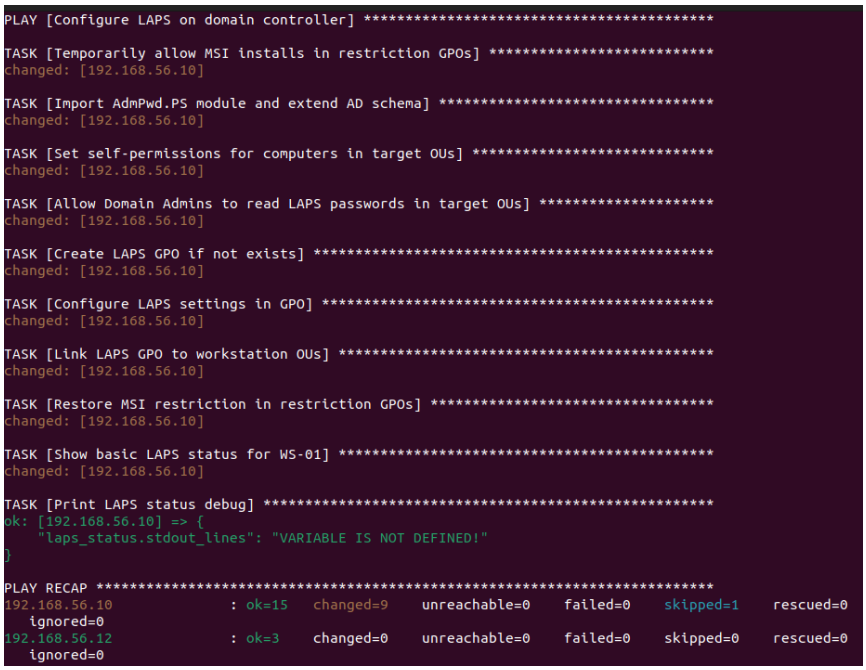


Рис. 3.3.1 Запуск плейбуку win_laps.yml

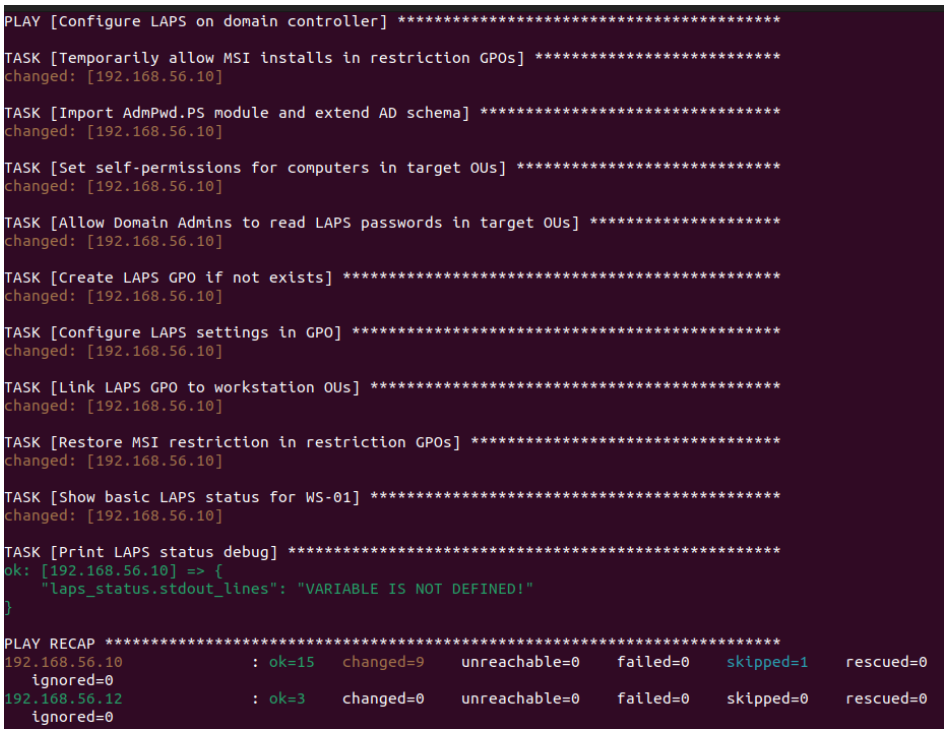


Рис. 3.3.2 Успішне виконання win_laps.yml

```
PS C:\Users\Administrator> Get-ADObject -LDAPFilter "(name=ms-Mcs-AdmPwd)" -SearchBase "CN=Schema,CN=Configuration,DC=co
rpmarkeiev,DC=net" |
>>   Select-Object Name,DistinguishedName

Name          DistinguishedName
----          -----------------
ms-Mcs-AdmPwd CN=ms-Mcs-AdmPwd,CN=Schema,CN=Configuration,DC=corpmarkeiev,DC=net


PS C:\Users\Administrator> Get-ADObject -LDAPFilter "(name=ms-Mcs-AdmPwdExpirationTime)" -SearchBase "CN=Schema,CN=Confi
guration,DC=corpmarkeiev,DC=net" |
>>   Select-Object Name,DistinguishedName

Name                       DistinguishedName
----                       -----------------
ms-Mcs-AdmPwdExpirationTime CN=ms-Mcs-AdmPwdExpirationTime,CN=Schema,CN=Configuration,DC=corpmarkeiev,DC=net
```

Рис. 3.3.3 Перевірка цілісності схеми AD та атрибутів LAPS

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> Import-Module AdmPwd.PS
>>
>> Find-AdmPwdExtendedRights -OrgUnit "OU=HR_Department,DC=corpmarkeiev,DC=net"
>> Find-AdmPwdExtendedRights -OrgUnit "OU=Developers_Department,DC=corpmarkeiev,DC=net"
>> Find-AdmPwdExtendedRights -OrgUnit "OU=IT_Department,DC=corpmarkeiev,DC=net"

ObjectDN                                      ExtendedRightHolders
--------                                      --------------------
OU=HR_Department,DC=corpmarkeiev,DC=net       {NT AUTHORITY\SYSTEM, CORPMARKEIEV\Domain Admins}
OU=Developers_Department,DC=corpmarkeiev,D... {NT AUTHORITY\SYSTEM, CORPMARKEIEV\Domain Admins}
OU=IT_Department,DC=corpmarkeiev,DC=net       {NT AUTHORITY\SYSTEM, CORPMARKEIEV\Domain Admins}


PS C:\Users\Administrator> Import-Module GroupPolicy
PS C:\Users\Administrator> Get-GPO -Name "GPO_LAPS_Workstations" |
>>   Select-Object DisplayName,Id

DisplayName           Id
-----------           --
GPO_LAPS_Workstations f9ce0270-c191-4849-b111-1368e6c79794
```

Рис. 3.3.4 Перевірка цілісності схеми AD та атрибутів LAPS

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> Get-GPInheritance -Target "OU=HR_Department,DC=corpmarkeiev,DC=net"
>> Get-GPInheritance -Target "OU=Developers_Department,DC=corpmarkeiev,DC=net"
>> Get-GPInheritance -Target "OU=IT_Department,DC=corpmarkeiev,DC=net"


Name                   : hr_department
ContainerType          : OU
Path                   : ou=hr_department,dc=corpmarkeiev,dc=net
GpoInheritanceBlocked  : No
GpoLinks               : {GPO_Restrictions_Managers, GPO_LAPS_Workstations}
InheritedGpoLinks      : {GPO_Restrictions_Managers, GPO_LAPS_Workstations, Default Domain Policy}

Name                   : developers_department
ContainerType          : OU
Path                   : ou=developers_department,dc=corpmarkeiev,dc=net
GpoInheritanceBlocked  : No
GpoLinks               : {GPO_Restrictions_Developers, GPO_LAPS_Workstations}
InheritedGpoLinks      : {GPO_Restrictions_Developers, GPO_LAPS_Workstations, Default Domain Policy}

Name                   : it_department
ContainerType          : OU
Path                   : ou=it_department,dc=corpmarkeiev,dc=net
GpoInheritanceBlocked  : No
GpoLinks               : {GPO_Restrictions_HelpDesk, GPO_LAPS_Workstations}
InheritedGpoLinks      : {GPO_Restrictions_HelpDesk, GPO_LAPS_Workstations, Default Domain Policy}
```

Рис. 3.3.5 Перевірка прав на організаційні одиниці та GPO LAPS

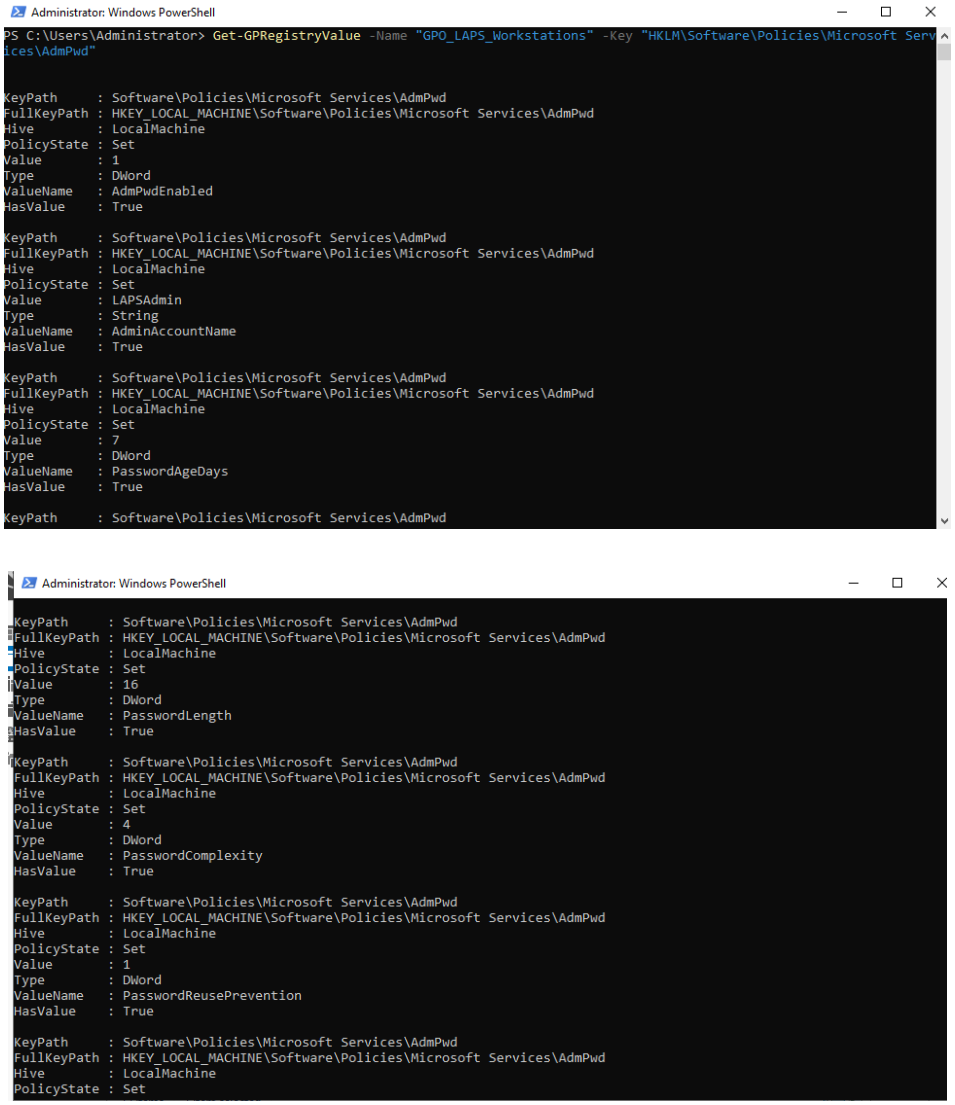


Рис. 3.3.6 Перегляд налаштувань політики LAPS у реєстрі

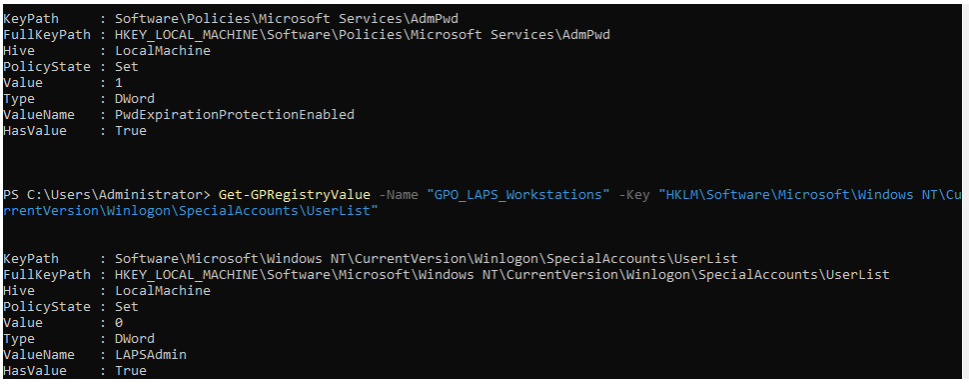


Рис. 3.3.7 Перегляд налаштувань політики LAPS у реєстрі

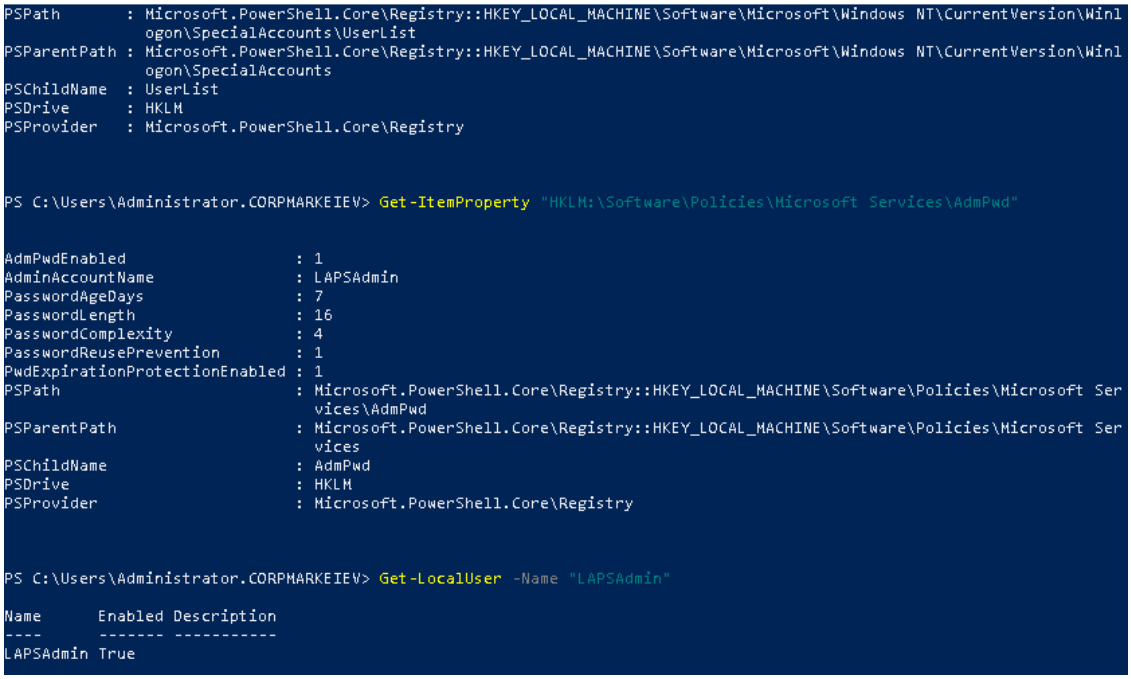


Рис. 3.3.8 Перевірка застосування політик на WS-01

Плейбук audit_logging.yml відповідає за централізоване увімкнення та уніфікацію журналювання безпекових подій у всьому домені corpmarkeiev.net. Його мета – забезпечити однаковий рівень видимості подій безпеки як на контролерах домену, так і на робочих станціях, щоб подальший аналіз інцидентів і спроб атак ґрунтувався на повних та стандартизованих журналах. Усі налаштування виконуються уніфіковано через Ansible, тому їх можна швидко відтворити, модифікувати або розгорнути в іншому середовищі, не вносячи змін вручну через графічні консолі.

У рамках плейбуку створюються дві окремі групові політики. Перша GPO_Audit_DomainControllers – прив’язується до контейнера Domain Controllers і націлена на контролер домену SRV-MRKV. Для цієї політики вмикаються розширені категорії аудиту Logon/Logoff та Directory Service Changes, що дозволяє фіксувати усі події входу й виходу користувачів, а також зміни в об’єктах каталогу Active Directory, включно з операціями над обліковими записами та групами. Додатково задаються параметри розміру та політики перезапису системних журналів, щоб уникнути швидкого перезаповнення й втрати важливих записів. Друга політика – GPO_Audit_Workstations – лінкується до організаційних підрозділів з робочими станціями й концентрується на категорії Logon/Logoff, забезпечуючи

реєстрацію успішних і невдалих входів користувачів на клієнтських машинах. Після створення й лінкування політик плейбук примусово оновлює групові політики на доменному контролері та клієнтах і виводить актуальний стан локальної audit-конфігурації, що дозволяє наочно продемонструвати ефект від автоматизованого налаштування. У підсумку audit_logging.yml формує єдиний стандарт аудиту подій безпеки для всіх вузлів домену й готує базу для подальшого моніторингу, аналізу інцидентів та навчальних пен-тестів у межах дипломного проєкту.

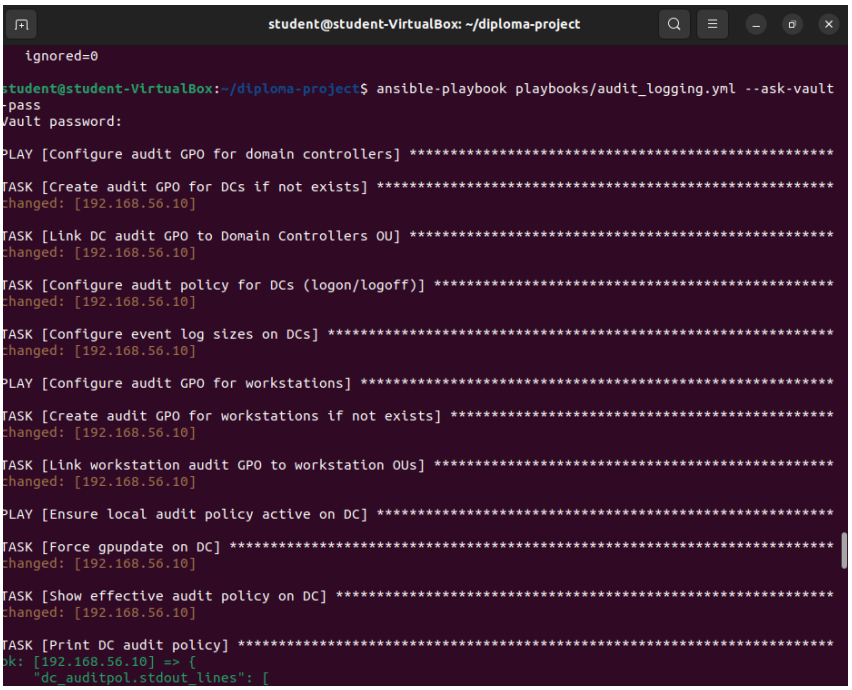


Рис. 3.3.9 Запуск плейбуку audit_logging.yml

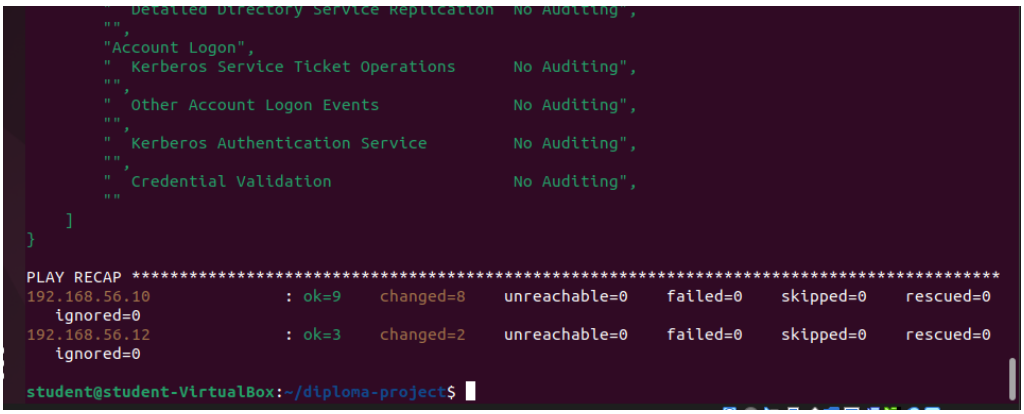


Рис. 3.3.10 Успішне завершення роботи плейбуку audit_logging.yml

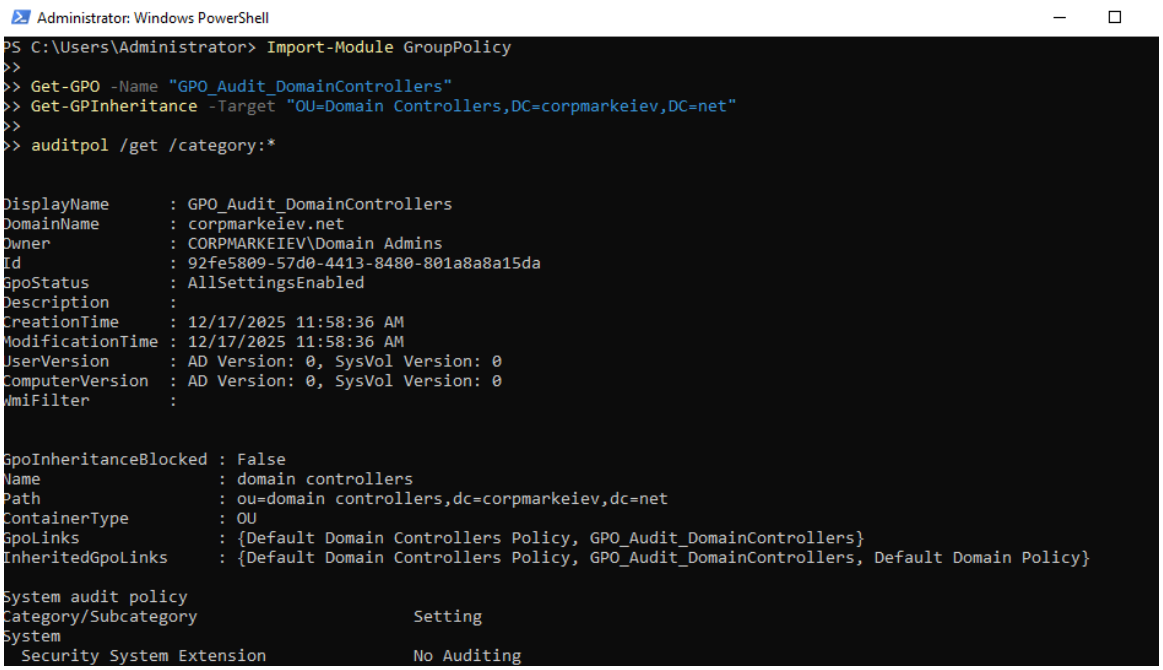


Рис. 3.3.11 Перевірка створеної політики на контролері домену

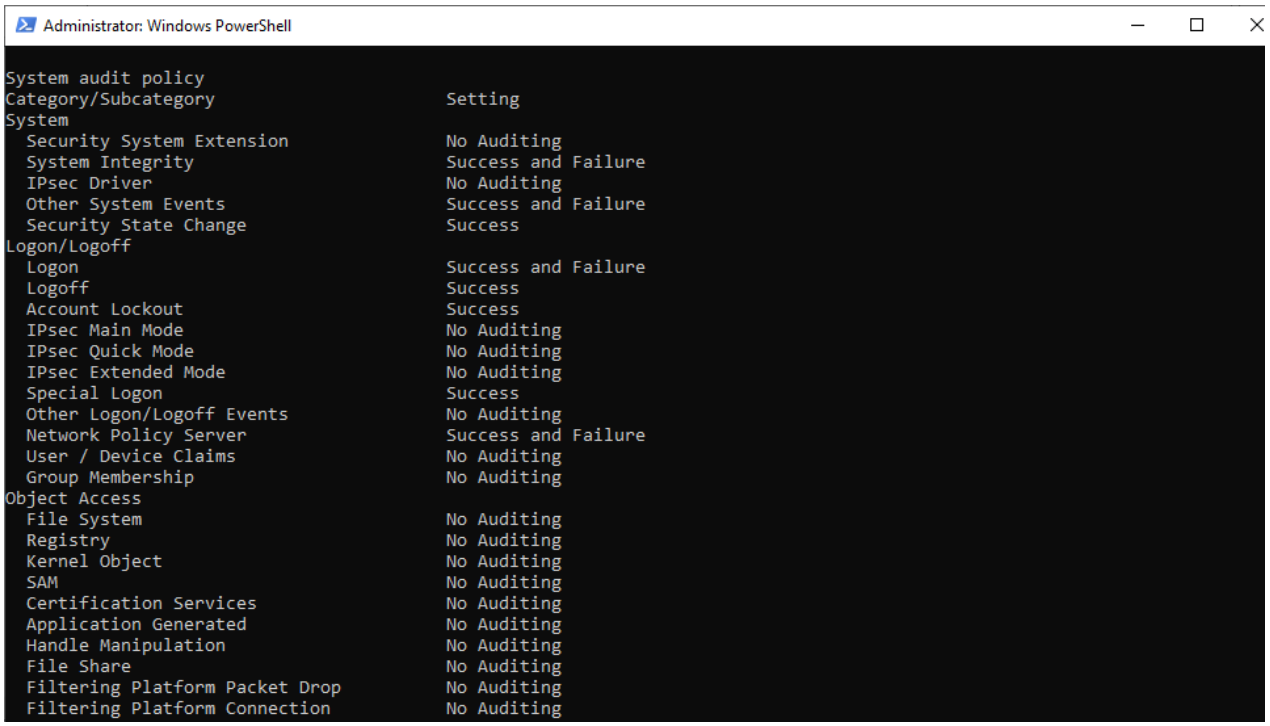


Рис. 3.3.12 Перевірка створеної політики на контролері домену

```
Policy Change
  Audit Policy Change                          Success
  Authentication Policy Change                 Success
  Authorization Policy Change                  No Auditing
  MPSSVC Rule-Level Policy Change              No Auditing
  Filtering Platform Policy Change             No Auditing
  Other Policy Change Events                   No Auditing
Account Management
  Computer Account Management                  Success
  Security Group Management                    Success
  Distribution Group Management                No Auditing
  Application Group Management                 No Auditing
  Other Account Management Events              No Auditing
  User Account Management                      Success
DS Access
  Directory Service Access                     Success
  Directory Service Changes                    No Auditing
  Directory Service Replication                No Auditing
  Detailed Directory Service Replication       No Auditing
Account Logon
  Kerberos Service Ticket Operations           Success
  Other Account Logon Events                   No Auditing
  Kerberos Authentication Service              Success
  Credential Validation                        Success
```

Рис. 3.3.13 Параметри створеної політики на контролері домену

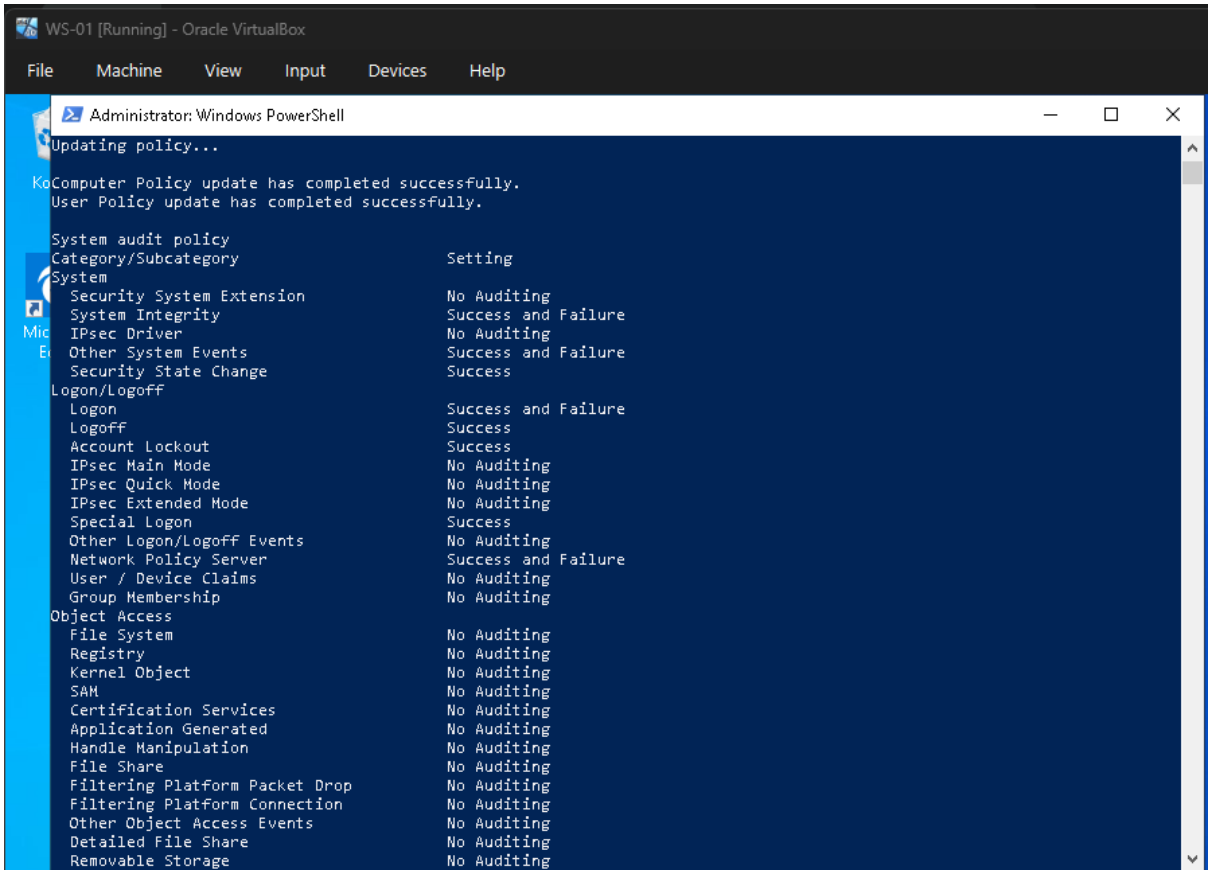


Рис. 3.3.14 Параметри створеної політики на робочій станції

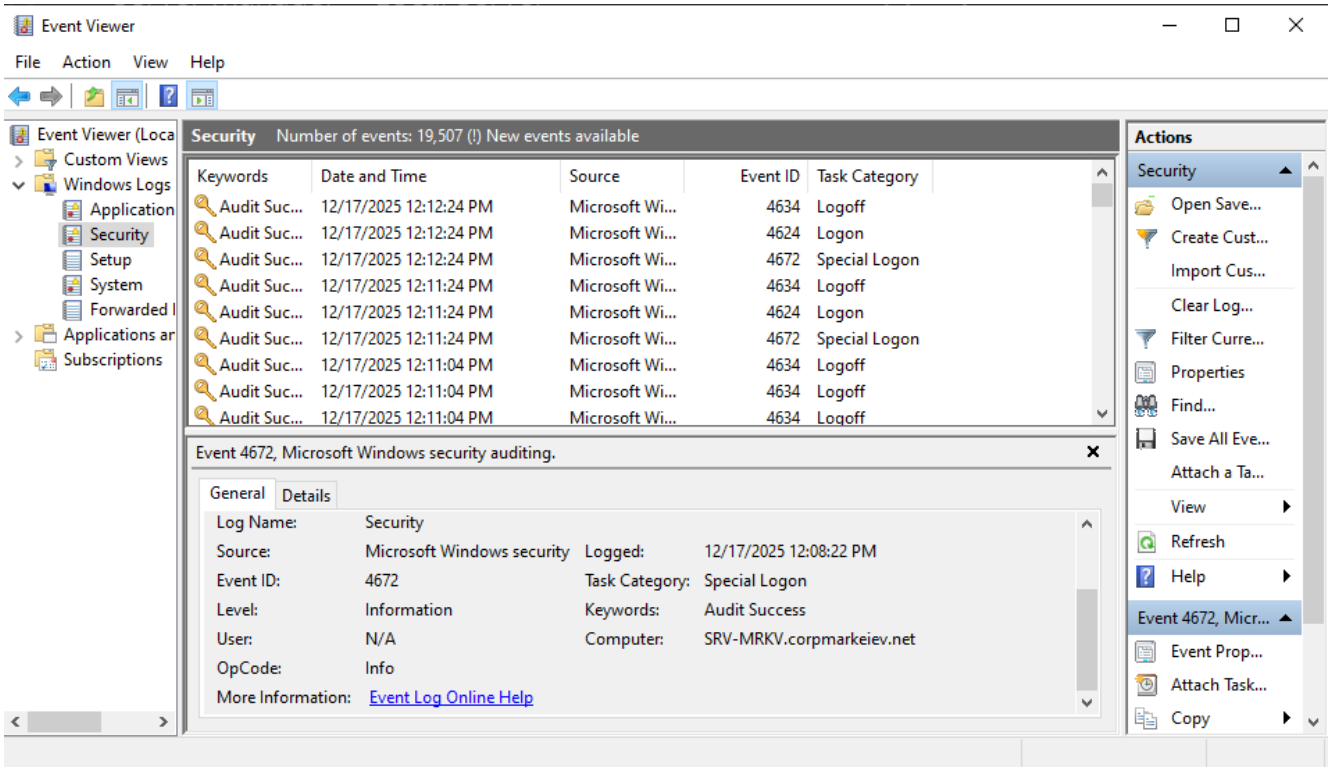


Рис. 3.3.15 Наявні події входу\виходу в Event Listener-Security

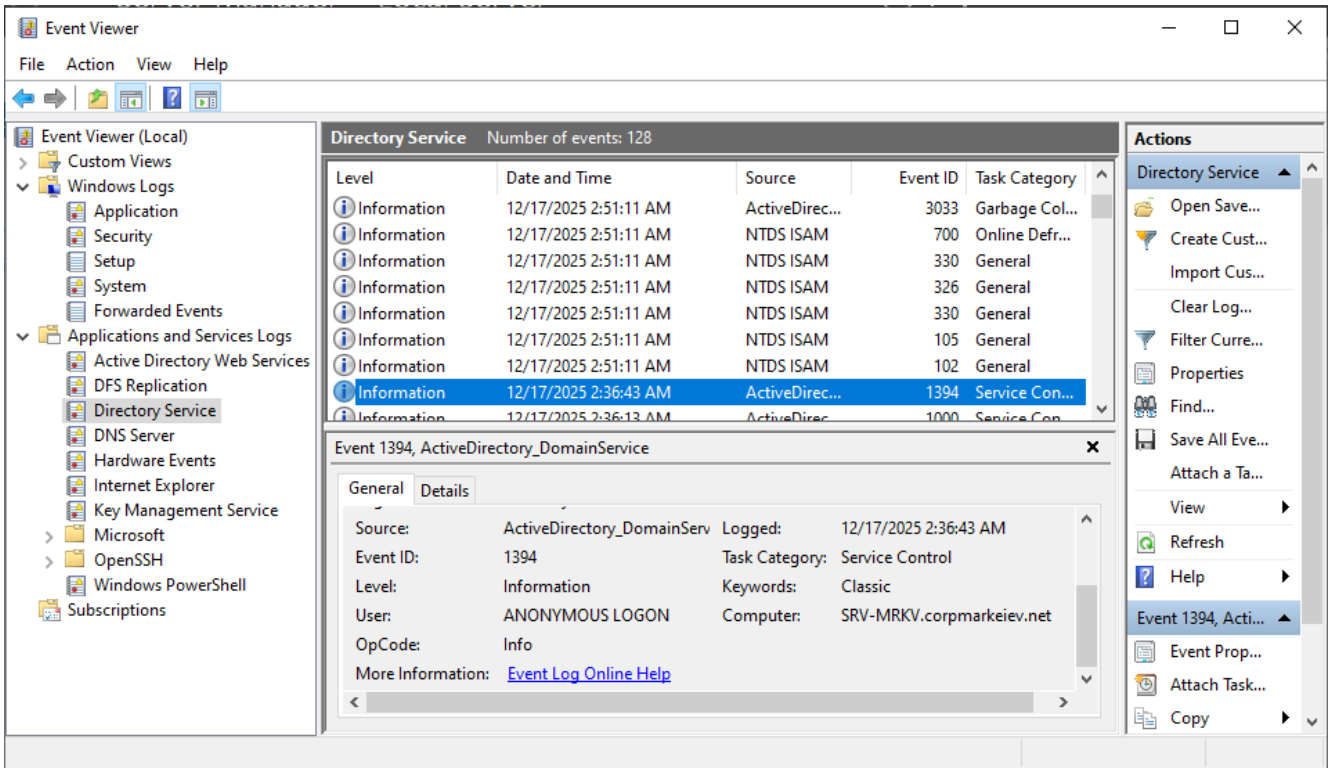


Рис. 3.3.16 Наявні події змін в Event Listener-Security Service

На цьому етапі було реалізовано автоматизоване резервне копіювання контролера домену за допомогою Ansible та вбудованого інструменту Windows Server

Backup. Основна мета – забезпечити регулярне створення резервної копії стану системи на окремий диск, відокремлений від системного тому, щоб підвищити надійність відновлення служби каталогів Active Directory у разі збоїв.

Першим кроком плейбук встановлює на сервері роль Windows Server Backup. Для цього використовується задача, яка додає функцію Windows-Server-Backup разом з необхідними інструментами керування, що дозволяє виконувати резервне копіювання як через графічний інтерфейс, так і через командний рядок wbadmin. Такий підхід гарантує, що перед запуском резервного копіювання всі потрібні компоненти вже присутні на контролері домену.

Спеціально для зберігання резервних копій був створений окремий віртуальний диск з літерою E:, позначений як «For Backup». Плейбук використовує цей диск як ціль -backupTarget:E: для операції System State backup, що відповідає рекомендаціям не зберігати резервні копії на системному томі C:. Така ізоляція зменшує ризик одночасної втрати як робочої системи, так і резервів.

Далі плейбук виконує разове резервне копіювання стану системи за допомогою команди wbadmin start systemstatebackup -backupTarget:E: -quiet. Ця команда створює резервну копію критичних компонентів контролера домену: бази даних Active Directory, SYSVOL, системного реєстру та файлів завантаження, що необхідні для повного відновлення ролі домен-контролера. Результат виконання команди реєструється та виводиться у звіті Ansible, що дозволяє задокументувати успішне завершення операції.

Щоб резервне копіювання виконувалося регулярно без ручного втручання, плейбук створює в Планувальнику завдань Windows окреме завдання з назвою, наприклад, Ansible_SystemState_Backup. Це завдання налаштоване на щоденний запуск у заданий час і виконує ту саму команду wbadmin start systemstatebackup -backupTarget:E: -quiet від імені облікового запису SYSTEM з найвищим рівнем привілеїв. Таким чином, резервне копіювання стану системи стає повністю автоматизованим і не залежить від дій адміністратора.

Після виконання основних дій плейбук додатково запускає команду wbadmin get versions, щоб вивести список доступних резервних копій на диску E:. Це дозволяє підтвердити наявність щонайменше однієї актуальної версії System State backup і надати докази успішного налаштування в супровідній документації. У поєднанні з перевіркою наявності завдання у Планувальнику та вмісту каталогу E:\WindowsImageBackup це демонструє, що етап резервного копіювання домен-контролера реалізовано коректно та відповідає вимогам проекту.

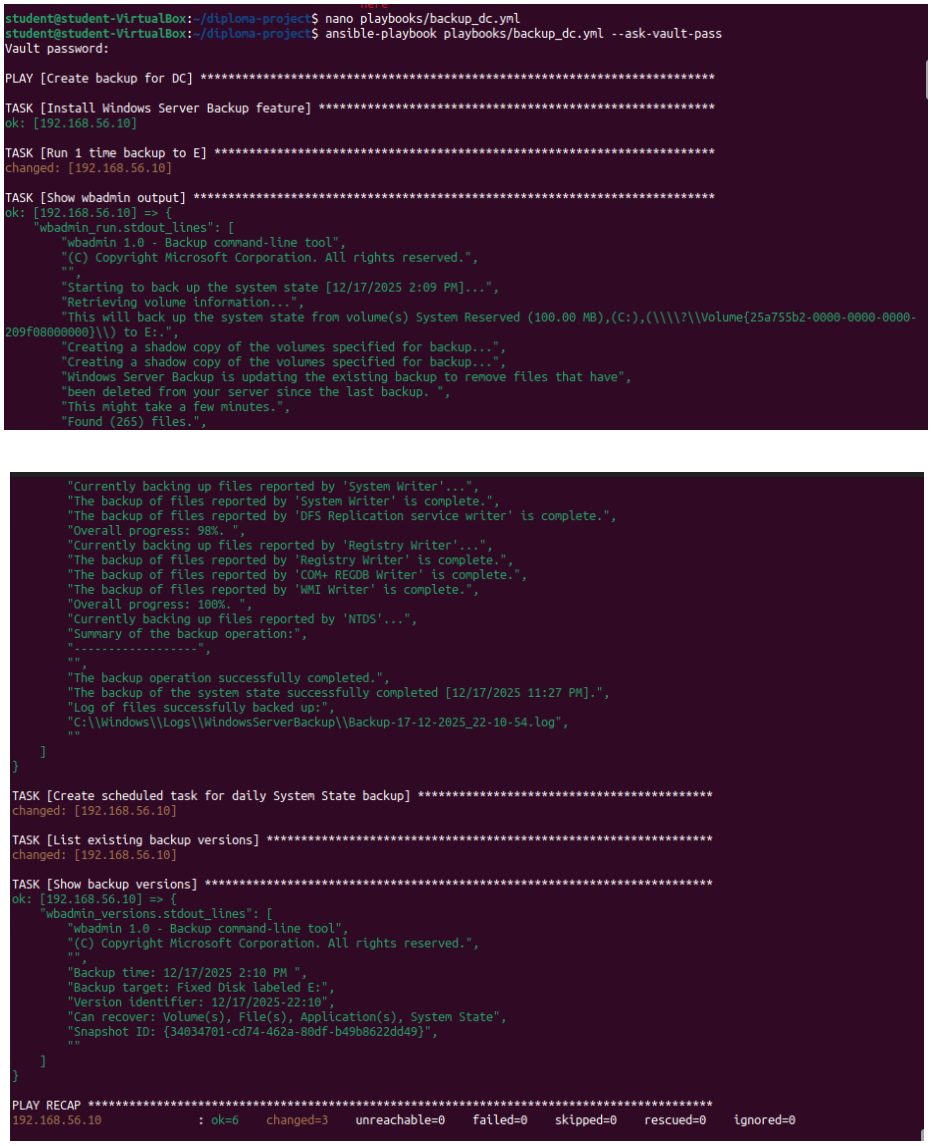


Рис. 3.3.17 Виконання плейбуку backup_dc.yml

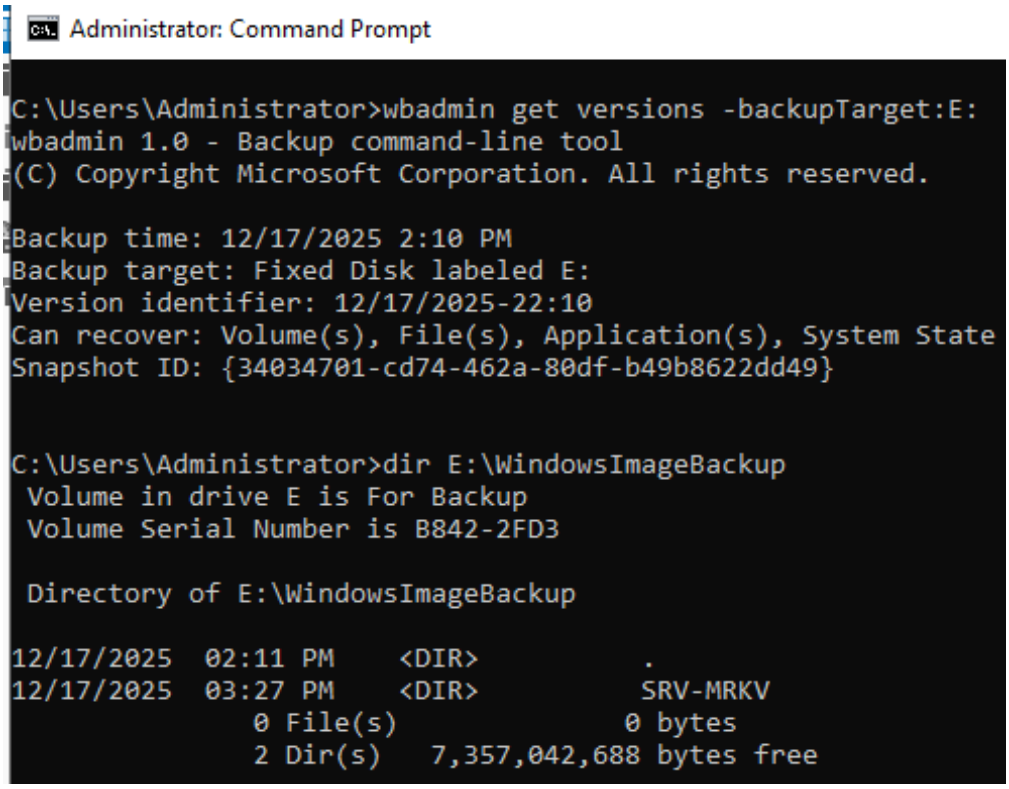


Рис. 3.3.18 Перевірка створеної резервної копії

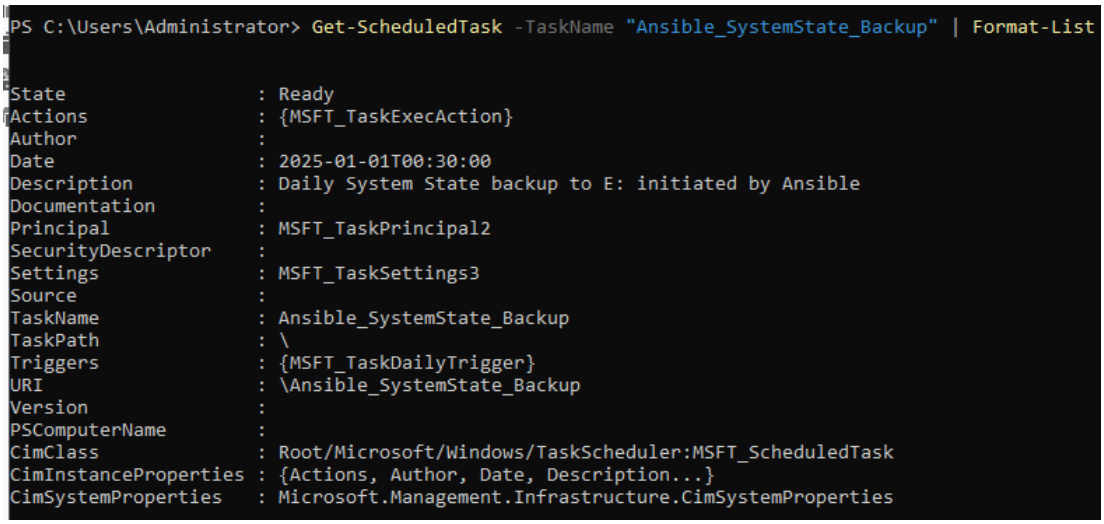


Рис. 3.3.19 Перевірка запланованого завдання через командний рядок PS

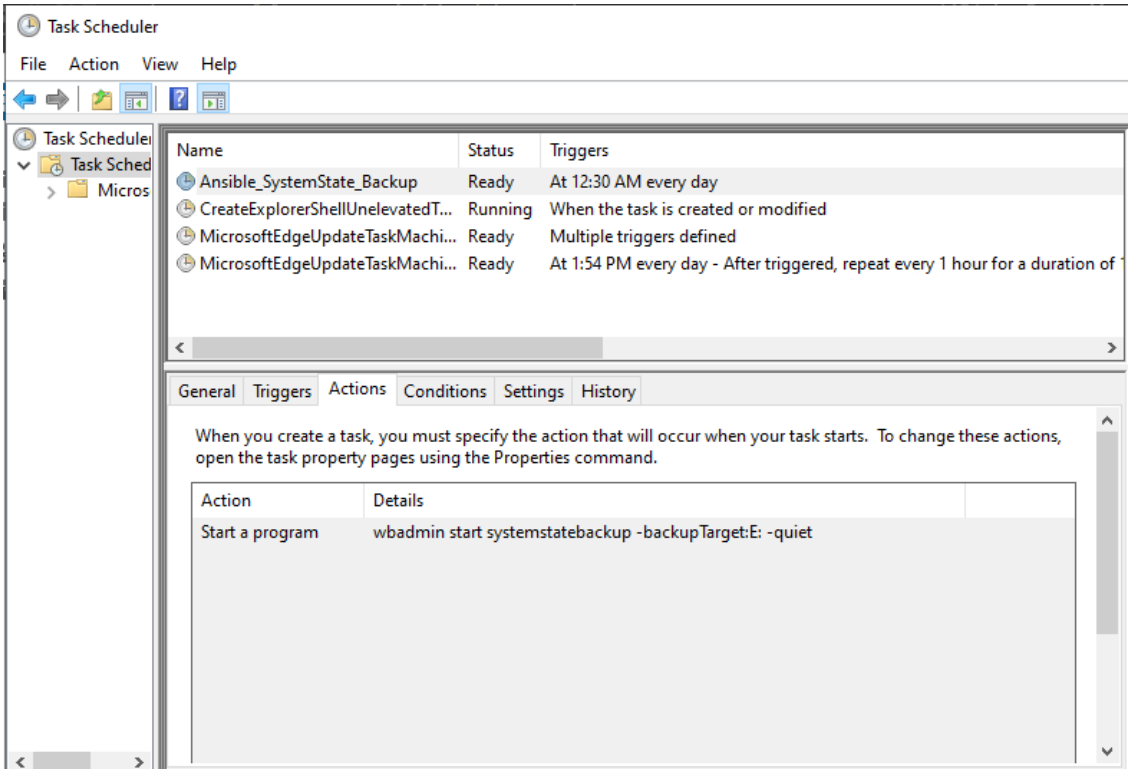


Рис. 3.3.20 Перевірка запланованого завдання в планувальнику

3.4 Тестування на проникнення Pass-the-Hash

Експериментальний етап передбачає симуляцію атаки Pass-the-Hash проти робочої станції WS-01 з використанням Ansible як оркестратора та спеціалізованих інструментів безпеки. Для цього на контролері Ansible створюється окремий файл змін attack.yml, що містить опис об’єкта атаки: IP-адресу станції, ім’я цільового облікового запису та його NTLM-хеш. Файл розміщується в директорії inventory/group_vars/windows/attack.yml та шифрується за допомогою механізму Ansible Vault, що запобігає зберіганню чутливих даних у відкритому вигляді.

Підготовка WS-01 до симуляції включала примусове відкриття мережевої служби SMB. Для цього на цільовій станції було ввімкнено правила брандмауера Windows, пов’язані зі спільним доступом до файлів і принтерів: зокрема, правило FPS-SMB-In-TCP для вхідного TCP-порту 445, а також набір правил WMI для забезпечення роботи віддаленого WMI-каналу, який використовує Impacket-інструмент wmiexec.py. Додатково за необхідності були створені правила для RPC/DCOM з метою уникнення таймаутів з’єднання, після чого доступність хоста та сервісу

SMB перевірялася за допомогою nmap -p 445 -Pn -n 192.168.56.12 з боку Ansible-ноди.

На стороні Windows-станції для отримання NTLM-хешу локального адміністративного акаунта LAPSAdmin використовувався інструмент Mimikatz. Після входу на WS-01 з правами адміністратора було отримано вивід із поточними обліковими даними сесій. У секції, що відповідає користувачу LAPSAdmin, зчитувався рядок формату NTLM : <hash>, який відповідає NT-частині NTLM-хешу; цей хеш було об'єднано з «порожнім» LM-хешем у форматі aad3b435b51404eeaad3b435b51404ee:<NT_hash> та перенесено у файл attack.yml як значення змінної attack_ntlm_hash.

Реалізація самої атаки була оформлена в окремому плейбуку attack_sim.yml. Плейбук працює на хості localhost, підтягує змінні з attack.yml та виконує три основні кроки: по-перше, створює локальну директорію logs і запускає Nmap-скан порту 445 на WS-01, зберігаючи результат у файл; по-друге, аналізує вивід Nmap і, якщо 445 недоступний, коректно завершує сценарій із повідомленням про недоцільність атаки; по-третє, у разі доступності SMB виконує команду wmiexec.py -hashes <LM:NT> LAPSAdmin@<192.168.56.12> "whoami", реєструє код повернення та вивід, записує їх у окремий лог-файл і на основі цих даних формує вердикт. Якщо команда завершується з нульовим кодом і повертає коректну відповідь whoami з WS-01, плейбук фіксує стан FAIL з повідомленням, що атака пройшла успішно та LAPS не запобіг використанню викраденого NTLM-хешу на цій станції.

У ході виконання експерименту саме такий сценарій і був зафіксований: після відкриття необхідних портів та налаштування WMI виклик wmiexec.py з NTLM-хешем LAPSAdmin зміг успішно автентифікуватися на WS-01 і виконати команду whoami, що у плейбуці інтерпретується як «FAIL: Атака пройшла успішно. Result: LAPS не працює або налаштований некоректно». З безпекової точки зору це демонструє важливе обмеження LAPS: за умови повної компрометації ок-

ремої станції засіб не перешкоджає повторному використанню цих облікових даних на тому ж самому хості. LAPS знижує ризик латерального переміщення між різними вузлами, але не захищає від атак, коли зловмисник уже повністю контролює конкретну машину, що є важливим висновком для інтерпретації результатів експериментального етапу.

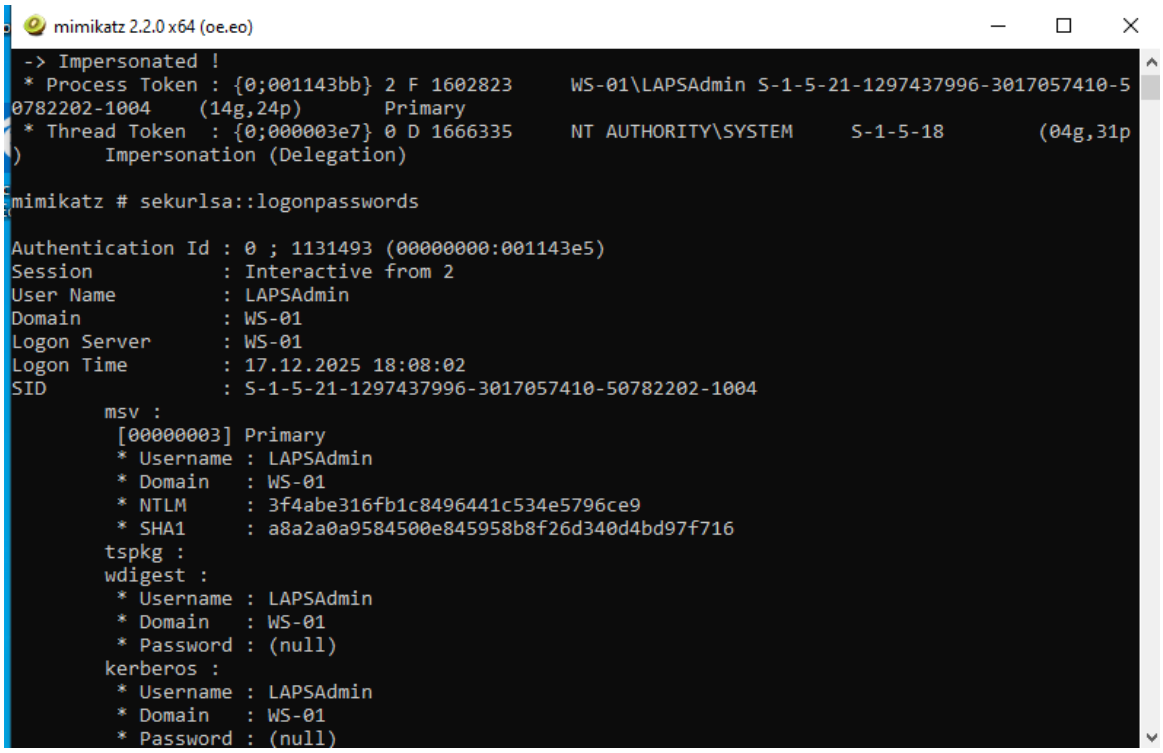


Рис. 3.4.1 Отримання NTLM хешу LAPSAdmin за допомогою Mimikatz

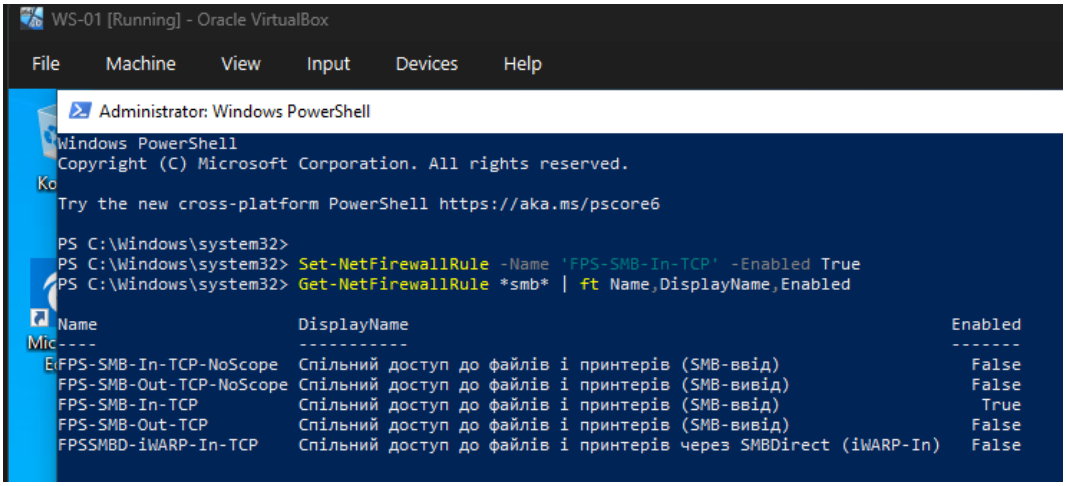


Рис. 3.4.2 Відкриття мережевої служби SMB для вхідного порту 445 TCP

```
PS C:\Windows\system32> Get-NetFirewallRule | Where-Object {
>>     $_.DisplayName -like '*WMI*' -or $_.DisplayName -like '*Windows Management*'
>> } | Select Name,DisplayName,Enabled,Direction

Name                        DisplayName                                  Enabled Direction
----                        -----------                                  ------- ---------
WMI-WINMGMT-In-TCP-NoScope  Інструментарій керування Windows (WMI-ввід)    False   Inbound
WMI-WINMGMT-Out-TCP-NoScope Інструментарій керування Windows (WMI-вивід)   False  Outbound
WMI-WINMGMT-In-TCP          Інструментарій керування Windows (WMI-ввід)    False   Inbound
WMI-WINMGMT-Out-TCP         Інструментарій керування Windows (WMI-вивід)   False  Outbound


PS C:\Windows\system32> Set-NetFirewallRule -Name 'WMI-WINMGMT-In-TCP' -Enabled True
>> Set-NetFirewallRule -Name 'WMI-WINMGMT-In-TCP-NoScope' -Enabled True
PS C:\Windows\system32> Get-NetFirewallRule 'WMI-WINMGMT-In-TCP','WMI-WINMGMT-In-TCP-NoScope' |
>>   Select Name,DisplayName,Enabled,Direction

Name                        DisplayName                                  Enabled Direction
----                        -----------                                  ------- ---------
WMI-WINMGMT-In-TCP-NoScope Інструментарій керування Windows (WMI-ввід)    True   Inbound
WMI-WINMGMT-In-TCP         Інструментарій керування Windows (WMI-ввід)    True   Inbound
```

Рис. 3.4.3 Відкриття WMI каналів для портів 135

```
student@student-VirtualBox:~/diploma-project$ nmap -p 445 -Pn -n 192.168.56.12
Starting Nmap 7.80 ( https://nmap.org ) at 2025-12-18 04:41 EET
Nmap scan report for 192.168.56.12
Host is up.

PORT    STATE    SERVICE
445/tcp filtered microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 2.09 seconds
student@student-VirtualBox:~/diploma-project$ ansible-playbook playbooks/attack_sim.yml --ask-vault-pass
Vault password:

PLAY [Simulate Pass-the-Hash attack on WS-01] **********************************************************************

TASK [Ensure logs directory exists] ********************************************************************************
ok: [localhost]

TASK [Scan SMB port 445 on WS-01] **********************************************************************************
ok: [localhost]

TASK [Save Nmap output to file] ************************************************************************************
changed: [localhost]

TASK [Check if port 445 is open] ***********************************************************************************
ok: [localhost]

TASK [Stop if SMB is not accessible] *******************************************************************************
fatal: [localhost]: FAILED! => {"changed": false, "msg": "Ціль 192.168.56.12 недоступна або порт 445 закритий. Симуляція не виконує
ться."}

PLAY RECAP *********************************************************************************************************
localhost                  : ok=4    changed=1    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0

student@student-VirtualBox:~/diploma-project$
```

Рис. 3.4.4 Безуспішна атака при закритій SMB службі

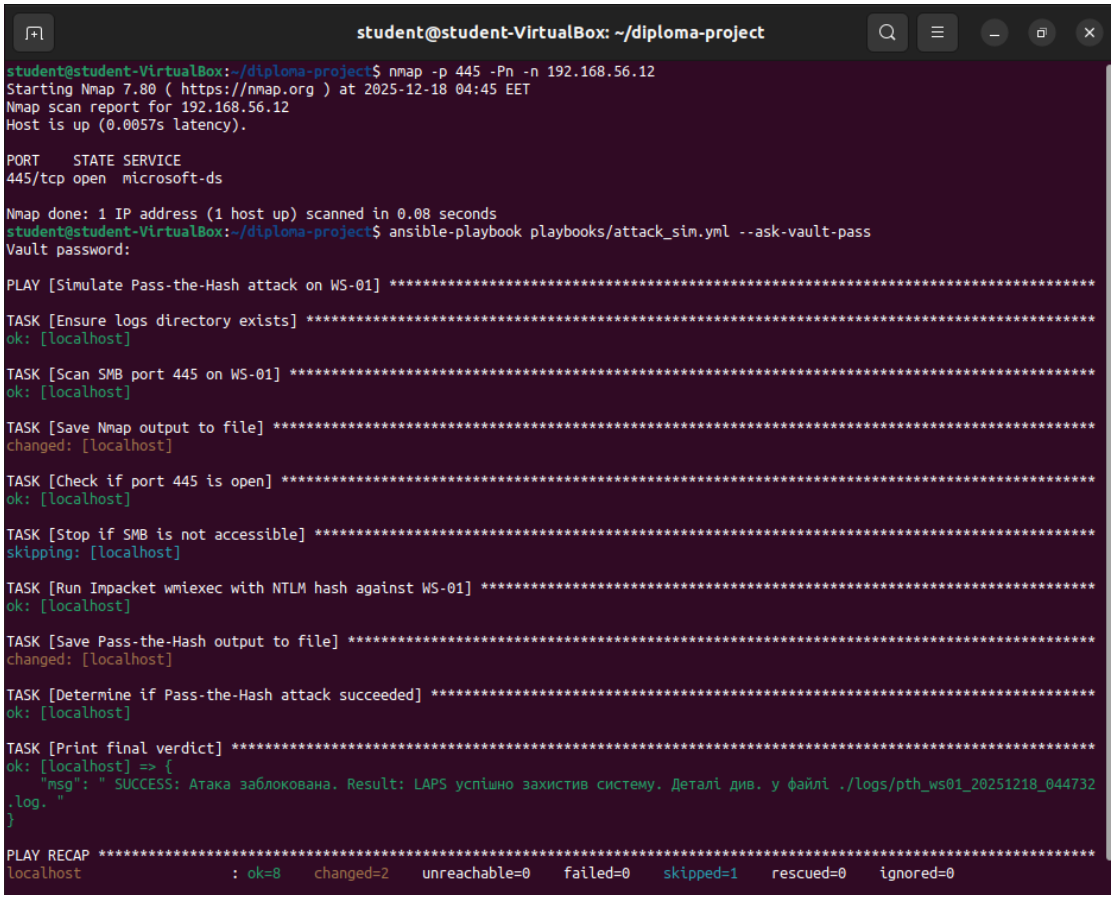


Рис. 3.4.5 Безуспішна атака при відкритих портах TCP 445 але закритих WMI

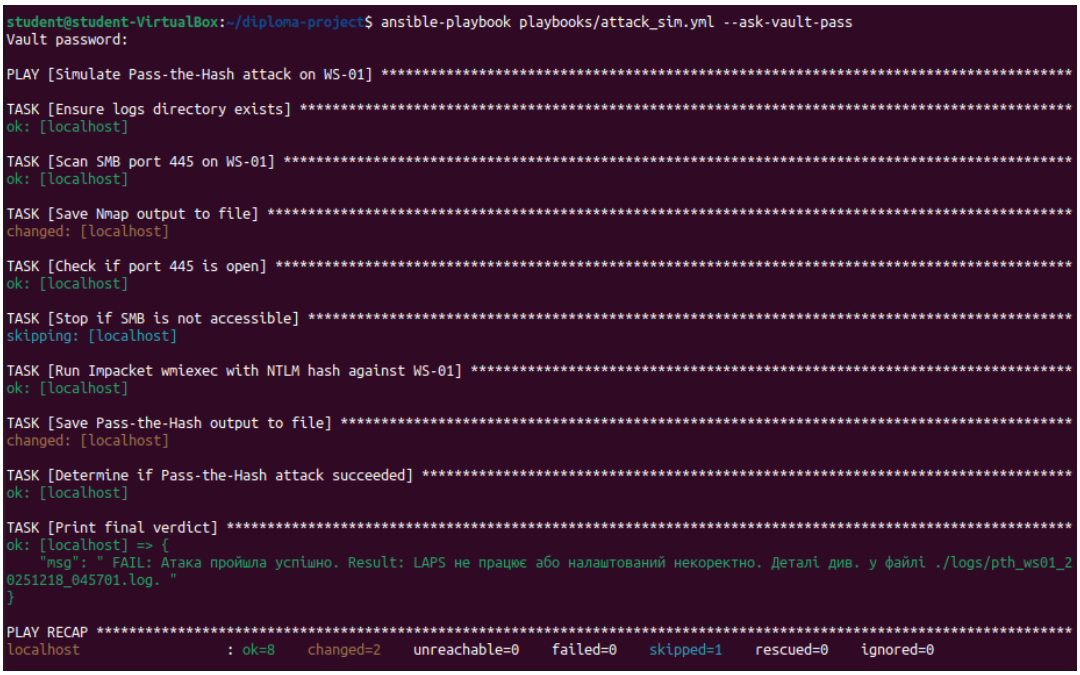


Рис. 3.4.6 Успішна атака при повністю відкритих портах TCP та SMB



Рис. 3.4.7 Вміст лог файлу результатів атаки

Висновки до третього розділу

У третьому розділі здійснено програмну реалізацію та тестування розробленої системи автоматизації в ізольованому середовищі VirtualBox, що об'єднало контролер домену Windows Server 2022, клієнтську станцію Windows 10 та керуючий вузол Ansible. В ході роботи було налаштовано захищений канал WinRM та створено набір плейбуків для автоматичного розгортання інфраструктури, формування структури Active Directory і гнучкого управління груповими політиками. Окрім цього, успішно впроваджено підсистему безпеки з рішенням LAPS, налаштовано розширений аудит і регулярне резервне копіювання, а ефективність захисних механізмів підтверджено шляхом симуляції атаки Pass-the-Hash.

ВИСНОВКИ

При виконанні випускної кваліфікаційної роботи було проведено аналіз методів захисту та автоматизації корпоративних мереж, було розглянуто сучасні вектори атак на Microsoft Active Directory та існуючі інструменти оркестрації. Було вирішено, що найбільш ефективним засобом досягнення мети є використання платформи Ansible у поєднанні з нативними можливостями PowerShell, що дозволяє реалізувати підхід «Інфраструктура як код».

Було спроектовано архітектуру тестового середовища на базі віртуалізації, яка забезпечує ізоляцію та безпеку проведення експериментів. Було враховано необхідність захищеної комунікації між керуючим вузлом та серверами, для чого було обрано та налаштовано протокол WinRM із шифруванням SSL, що гарантує конфіденційність передачі даних керування в мережі.

На основі мети кваліфікаційної роботи було поставлено технічне завдання, котре включало в себе розробку модульної системи для розгортання, адміністрування та захисту домену. Було досліджено механізми взаємодії Ansible з Windows-системами та визначено необхідні колекції модулів для реалізації поставлених завдань.

Далі було розгорнуто програмний комплекс, за допомогою якого виконано автоматичне налаштування контролера домену, включаючи інсталяцію доменних служб та конфігурацію DNS. Після цього було автоматизовано створення ієрархічної структури організаційних одиниць, груп безпеки та облікових записів користувачів, між якими було чітко розмежовано права доступу відповідно до рольової моделі.

За допомогою розроблених сценаріїв та маніпуляцій з реєстром, було налагоджено обмеження для різних груп користувачів до критичних системних функцій, таких як: доступ до панелі керування, редактора реєстру, інструментів командного рядка та підключення USB-накопичувачів. Окремо було впроваджено систему LAPS для захисту паролів локальних адміністраторів та налаштовано автоматичне резервне копіювання стану системи.

В результаті роботи було проведено успішне тестування системи на стійкість до атак та створено дієві автоматизовані рішення, які дозволяють значно пришвидшити розгортання захищеної інфраструктури Microsoft Active Directory та мінімізувати ризики людських помилок.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. MS AD: [Електронний ресурс] / CyberArk. – Режим доступу: https://www.cyberark.com/what-is/active-directory/ – Загол. з екрану.
2. What Is Active Directory and How Does It Work? [Електронний ресурс] / Lepide. – Режим доступу: https://www.lepide.com/blog/what-is-active-directory-and-how-does-it-work/ – Загол. з екрану.
3. Mitigating Pass-the-Hash (PtH) Attacks: [Електронний ресурс] / Microsoft. – Режим доступу: https://www.microsoft.com/en-us/security/blog/2014/06/23/mitigating-pass-the-hash-pth-attacks-and-other-credential-theft-techniques/ – Загол. з екрану.
4. NTLM EXPLAINED: [Електронний ресурс] / Crowdstrike. – Режим доступу: https://www.crowdstrike.com/cybersecurity-101/ntlm-windows-new-technology-lan-manager/ – Загол. з екрану.
5. Kerberos Authentication: [Електронний ресурс] / Fortinet. – Режим доступу: https://www.fortinet.com/resources/cyberglossary/kerberos-authentication – Загол. з екрану.
6. Group Policy Objects: [Електронний ресурс] / Microsoft. – Режим доступу: https://learn.microsoft.com/en-us/previous-versions/windows/desktop/policy/group-policy-objects – Загол. з екрану.
7. AD DS: Fine-Grained Password Policies: [Електронний ресурс] / Microsoft. – Режим доступу: https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc770394(v=ws.10 – Загол. з екрану.
8. CHEF: [Електронний ресурс] / Intellipaat. – Режим доступу: https://intellipaat.com/blog/what-is-chef/ – Загол. з екрану.
9. Ansible: [Електронний ресурс] / Ansible. – Режим доступу: https://www.ansible.com/how-ansible-works/ – Загол. з екрану.
10. Ansible for Windows: [Електронний ресурс] / Ansible Documentation. – Режим доступу: https://docs.ansible.com/ansible/latest/os_guide/windows_usage.html – Загол. з екрану.

11. MDT: [Електронний ресурс] / SmartDeploy. – Режим доступу: https://www.smartdeploy.com/blog/microsoft-deployment-toolkit-pros-cons-of-a-free-tool/ – Загол. з екрану.
12. DSC: [Електронний ресурс] / Microsoft. – Режим доступу: https://learn.microsoft.com/en-us/powershell/dsc/overview?view=dsc-2.0 – Загол. з екрану.
13. Powershell: [Електронний ресурс] / Microsoft. – Режим доступу: https://learn.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7.4 – Загол. з екрану.
14. Oracle VM VirtualBox User Manual: [Електронний ресурс] / Oracle. – Режим доступу: https://www.virtualbox.org/manual/UserManual.html – Загол. з екрану.
15. Installation and Configuration for Windows Remote Management (WinRM): [Електронний ресурс] / Microsoft. – Режим доступу: https://learn.microsoft.com/en-us/windows/win32/winrm/installation-and-configuration-for-windows-remote-management – Загол. з екрану.
16. New-ADOrganizationalUnit: [Електронний ресурс] / Microsoft. – Режим доступу: https://learn.microsoft.com/en-us/powershell/module/activedirectory/new-adorganizationalunit?view=windowsserver2022-ps – Загол. з екрану.
17. GPO and Powershell: [Електронний ресурс] / CloudBrothers. – Режим доступу: https://cloudbrothers.info/en/manage-group-policies-powershell/ – Загол. з екрану.
18. Group Policy Examples: Most Useful GPOs for Security: [Електронний ресурс] / ActiveDirectoryPRO. – Режим доступу: https://activedirectorypro.com/group-policy-examples-most-useful-gpos-for-security/ – Загол. з екрану.
19. Local Administrator Password Solution (LAPS): [Електронний ресурс] / Microsoft. – Режим доступу: https://learn.microsoft.com/en-us/windows-server/identity/laps/laps-overview – Загол. з екрану.
20. What is LDAP? [Електронний ресурс] / Okta. – Режим доступу: https://www.okta.com/identity-101/what-is-ldap/ – Загол. з екрану.

21. Network Address Translation (NAT): [Електронний ресурс] / GeeksforGeeks. – Режим доступу: https://www.geeksforgeeks.org/network-address-translation-nat/ – Загол. з екрану.
22. Setting up Active Directory via PowerShell: [Електронний ресурс] / Microsoft. – Режим доступу: https://www.microsoft.com/en-gb/industry/blog/technetuk/2016/06/08/setting-up-active-directory-via-powershell/ – Загол. з екрану.
23. Mimikatz: [Електронний ресурс] / GitHub. – Режим доступу: https://github.com/gentilkiwi/mimikatz – Загол. з екрану.

ДОДАТКИ

Додаток А

Лістинг конфігураційних файлів та плейбуків

Лістинг файлу конфігурації ansible.cfg

```
[defaults]
inventory = ./inventory/hosts.yml
host_key_checking = False
deprecation_warnings = False
```

Лістинг файлу інвентаризації inventory/hosts.yml

```
all:
  children:
    windows:
      children:
        dc:
          hosts:
            dc01:
              ansible_host: 192.168.56.10
        clients:
          hosts:
            ws01:
              ansible_host: 192.168.56.12
      vars:
        ansible_connection: winrm
        ansible_port: 5986
        ansible_winrm_transport: basic
        ansible_winrm_server_cert_validation: ignore
        ansible_user: Administrator
        ansible_winrm_read_timeout_sec: 120
        ansible_winrm_operation_timeout_sec: 60
```

Лістинг плейбука розгортання контролера домену playbooks/promote_dc.yml

```
- name: Deploy Domain Controller
```

```
hosts: [dc]
gather_facts: yes
vars:
  domain_name: "corpmarkeiev.net"
  netbios_name: "CORPMARKEIEV"
  recovery_password: "{{ ansible_password }}"
tasks:
  - name: Change Hostname to SRV-MRKV
    ansible.windows.win_hostname:
      name: SRV-MRKV
    register: hostname_res
  - name: Reboot if hostname changed
    ansible.windows.win_reboot:
      when: hostname_res.reboot_required
  - name: Install Active Directory Domain Services
    ansible.windows.win_feature:
      name: AD-Domain-Services
      state: present
      include_management_tools: yes
    register: feature_install
  - name: Create new Domain Forest
    microsoft.ad.domain:
      dns_domain_name: "{{ domain_name }}"
      safe_mode_password: "{{ recovery_password }}"
      domain_netbios_name: "{{ netbios_name }}"
      install_dns: yes
      domain_mode: Win2012R2
      forest_mode: Win2012R2
    register: domain_install
  - name: Reboot after Domain Promotion
    ansible.windows.win_reboot:
      msg: "Server is rebooting to finish AD installation"
      reboot_timeout: 900
    when: domain_install.reboot_required
```

Лістинг плейбука створення структури AD playbooks/ad_structure.yml

```
- name: Create AD Structure (OU, Group and Users)
  hosts: dc
  gather_facts: no
  vars:
    domain_name: "corpmarkeiev.net"
    base_dn: "DC=corpmarkeiev,DC=net"
    default_user_pass: "{{ ansible_password }}"
  tasks:
    - name: Create OUs
      microsoft.ad.ou:
        name: "{{ item }}"
        path: "{{ base_dn }}"
        state: present
        protect_from_deletion: no
      loop:
        - "IT_Department"
        - "HR_Department"
        - "Developers_Department"
    - name: Create Groups
      microsoft.ad.group:
        name: "{{ item.name }}"
        path: "OU={{ item.ou }},{{ base_dn }}"
        scope: global
        category: security
        state: present
      loop:
        - { name: "Role_Developers", ou: "Developers_Department" }
        - { name: "Role_HelpDesk", ou: "IT_Department" }
        - { name: "Role_Managers", ou: "HR_Department" }
    - name: Create Users
      microsoft.ad.user:
        identity: "{{ item.username }}"
        name: "{{ item.fullname }}"
        firstname: "{{ item.firstname }}"
```

```
    surname: "{{ item.surname }}"
    upn: "{{ item.username }}@{{ domain_name }}"
    password: "{{ default_user_pass }}"
    path: "OU={{ item.ou }},{{ base_dn }}"
    state: present
    enabled: yes
    password_expired: no
    groups:
      add:
        - "{{ item.group }}"
  loop:
    - firstname: "Oleh"
      surname: "Skrypka"
      fullname: "Oleh Skrypka"
      username: "oskrypka"
      ou: "Developers_Department"
      group: "Role_Developers"
    - firstname: "Maria"
      surname: "Koval"
      fullname: "Maria Koval"
      username: "mkoval"
      ou: "HR_Department"
      group: "Role_Managers"
    - firstname: "Petro"
      surname: "Bondar"
      fullname: "Petro Bondar"
      username: "pbondar"
      ou: "IT_Department"
      group: "Role_HelpDesk"
```

Лістинг плейбука приєднання клієнтів playbooks/domain_join.yml

```
- name: Join Client Workstations to Domain
  hosts: clients
  gather_facts: yes
  vars:
```

```
    domain_name: "corpmarkeiev.net"
    dns_server: "192.168.56.10"
    domain_admin_user: "Administrator@{{ domain_name }}"
    domain_admin_pass: "{{ ansible_password }}"

  tasks:
    - name: Set DNS Server to Domain Controller IP
      ansible.windows.win_dns_client:
        adapter_names: "*"
        ipv4_addresses:
          - "{{ dns_server }}"
    - name: Join Computer to Domain
      microsoft.ad.membership:
        dns_domain_name: "{{ domain_name }}"
        domain_admin_user: "{{ domain_admin_user }}"
        domain_admin_password: "{{ domain_admin_pass }}"
        state: domain
      register: domain_join_res
    - name: Reboot if needed
      ansible.windows.win_reboot:
        msg: "Rebooting to apply Domain Membership"
      when: domain_join_res.reboot_required
```

Лістинг плейбука налаштування GPO playbooks/gpo_setup.yml

```
- name: Setup GPO Restrictions based on Access Matrix
  hosts: dc
  gather_facts: no
  vars:
    domain_base: "DC=corpmarkeiev,DC=net"
    gpo_list:
      - name: "GPO_Restrictions_Managers"
        target_ous:
          - "OU=HR_Department,{{ domain_base }}"
        settings:
```

```
- key: "HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer"
  value_name: "NoControlPanel"
  type: "DWord"
  value: 1
- key: "HKCU\\Software\\Policies\\Microsoft\\Windows\\System"
  value_name: "DisableCMD"
  type: "DWord"
  value: 2
- key: "HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer"
  value_name: "DisallowRun"
  type: "DWord"
  value: 1
- key: "HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer\\DisallowRun"
  value_name: "1"
  type: "String"
  value: "powershell.exe"
- key: "HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer\\DisallowRun"
  value_name: "2"
  type: "String"
  value: "powershell_ise.exe"
- key: "HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer\\DisallowRun"
  value_name: "3"
  type: "String"
  value: "pwsh.exe"
- key: "HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\System"
  value_name: "DisableRegistryTools"
  type: "DWord"
  value: 1
- key: "HKLM\\Software\\Policies\\Microsoft\\Windows\\Installer"
  value_name: "DisableMSI"
  type: "DWord"
  value: 2
- key: "HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer"
  value_name: "NoRun"
  type: "DWord"
  value: 1
```

```
      - key: "HKLM\\Software\\Policies\\Microsoft\\Windows\\RemovableStorageDevices"
        value_name: "Deny_All"
        type: "DWord"
        value: 1
  - name: "GPO_Restrictions_Developers"
    target_ous:
      - "OU=Developers_Department,{{ domain_base }}"
    settings:
      - key: "HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer"
        value_name: "NoControlPanel"
        type: "DWord"
        value: 1
      - key: "HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\System"
        value_name: "DisableRegistryTools"
        type: "DWord"
        value: 1
      - key: "HKLM\\Software\\Policies\\Microsoft\\Windows\\Installer"
        value_name: "DisableMSI"
        type: "DWord"
        value: 2
      - key: "HKLM\\Software\\Policies\\Microsoft\\Windows\\RemovableStorageDevices"
        value_name: "Deny_All"
        type: "DWord"
        value: 1
  - name: "GPO_Restrictions_HelpDesk"
    target_ous:
      - "OU=IT_Department,{{ domain_base }}"
    settings:
      - key: "HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer"
        value_name: "NoRun"
        type: "DWord"
        value: 1
tasks:
  - name: Configure GPOs using PowerShell
```

```
ansible.windows.win_powershell:
  script: |
    $ErrorActionPreference = "Stop"
    if (-not (Get-Module -ListAvailable -Name GroupPolicy)) {
        Write-Error "Module GroupPolicy not found."
        exit 1
    }
    Import-Module GroupPolicy
    $JsonData = '{{ gpo_list | to_json }}'
    $GpoList = $JsonData | ConvertFrom-Json
    foreach ($Policy in $GpoList) {
        $GpoName = $Policy.name
        Write-Output "`n--- Processing GPO: $GpoName ---"
        $Gpo = Get-GPO -Name $GpoName -ErrorAction SilentlyContinue
        if (-not $Gpo) {
            New-GPO -Name $GpoName -Comment "Managed by Ansible" | Out-Null
            Write-Output "CREATED: GPO object '$GpoName'"
        } else {
            Write-Output "EXISTS: GPO object '$GpoName'"
        }
        foreach ($TargetOu in $Policy.target_ous) {
            if ([adsi]::Exists("LDAP://$TargetOu")) {
                $Inheritance = Get-GPInheritance -Target $TargetOu -ErrorAction SilentlyContinue
                $LinkExists = $false
                if ($Inheritance -and $Inheritance.GpoLinks) {
                    foreach ($Link in $Inheritance.GpoLinks) {
                        if ($Link.DisplayName -eq $GpoName) { $LinkExists = $true }
                    }
                }
                if (-not $LinkExists) {
                    New-GPLink -Name $GpoName -Target $TargetOu -LinkEnabled Yes | Out-Null
                    Write-Output "LINKED: To '$TargetOu'"
                } else {
                    Write-Output "EXISTS: Link on '$TargetOu'"
                }
            } else {
```

```
            Write-Warning "OU not found: $TargetOu. Skipping link."
          }
        }
        foreach ($Set in $Policy.settings) {
          try {
            Set-GPRegistryValue -Name $GpoName `
                    -Key $Set.key `
                    -ValueName $Set.value_name `
                    -Type $Set.type `
                    -Value $Set.value `
                    -ErrorAction Stop | Out-Null
            Write-Output "SET REGISTRY: $($Set.key)\$($Set.value_name) = $($Set.value)"
          } catch {
            Write-Error "FAILED setting registry key: $($Set.value_name). Error: $_"
          }
        }
      }
    register: gpo_output
  - name: Show PowerShell Execution Log
    debug:
      var: gpo_output.output
```

Лістинг плейбука впровадження LAPS playbooks/win_laps.yml

```
- name: Install LAPS MSI on all Windows hosts
  hosts: windows
  gather_facts: no
  tasks:
    - name: Ensure temp directory for MSI exists
      ansible.windows.win_file:
        path: C:\Temp
        state: directory
    - name: Copy LAPS MSI to Windows hosts
      ansible.windows.win_copy:
        src: ../files/LAPS.x64.msi
```

```
      dest: C:\Temp\LAPS.x64.msi
    - name: Install LAPS MSI (CSE only) on clients
      ansible.windows.win_package:
        path: C:\Temp\LAPS.x64.msi
        state: present
        arguments: '/quiet /norestart'
        log_path: C:\Temp\LAPS_install.log
      when: "'clients' in group_names"
    - name: Ensure local LAPSAdmin account exists on clients
      ansible.windows.win_user:
        name: "LAPSAdmin"
        password: "TmpP@ssw0rd!123"
        groups:
          - Administrators
        update_password: on_create
        password_never_expires: yes
        state: present
      when: "'clients' in group_names"
- name: Install LAPS MSI with management tools on domain controller
  hosts: dc
  gather_facts: no
  tasks:
    - name: Ensure temp directory for MSI exists on DC
      ansible.windows.win_file:
        path: C:\Temp
        state: directory
    - name: Copy LAPS MSI to DC
      ansible.windows.win_copy:
        src: ../files/LAPS.x64.msi
        dest: C:\Temp\LAPS.x64.msi
    - name: Install LAPS MSI with management tools on DC
      ansible.windows.win_package:
        path: C:\Temp\LAPS.x64.msi
        state: present
        arguments: '/quiet /norestart
ADDLOCAL=CSE,Management,Management.UI,Management.PS,Management.ADMX'
```

```
      log_path: C:\Temp\LAPS_DC_install.log
- name: Configure LAPS on domain controller
  hosts: dc
  gather_facts: no
  vars:
    laps_gpo_name: "GPO_LAPS_Workstations"
    laps_admin_account: "LAPSAdmin"
    laps_password_age_days: 7
    laps_password_length: 16
    laps_target_ous:
      - "OU=HR_Department,DC=corpmarkeiev,DC=net"
      - "OU=Developers_Department,DC=corpmarkeiev,DC=net"
      - "OU=IT_Department,DC=corpmarkeiev,DC=net"
    laps_restriction_gpos:
      - "GPO_Restrictions_Developers"
      - "GPO_Restrictions_HelpDesk"
      - "GPO_Restrictions_Managers"
  tasks:
    - name: Temporarily allow MSI installs in restriction GPOs
      ansible.windows.win_powershell:
        script: |
          Import-Module GroupPolicy -ErrorAction Stop
          $gpoNames = @(
          {{ laps_restriction_gpos | map('regex_replace','^(.*)$','"\1"') | join(",") }}
          )
          foreach ($gpoName in $gpoNames) {
              $gpo = Get-GPO -Name $gpoName -ErrorAction SilentlyContinue
              if ($gpo) {
                  $key = "HKLM\Software\Policies\Microsoft\Windows\Installer"
                  Set-GPRegistryValue -Name $gpoName -Key $key -ValueName "DisableMSI" -Type DWord -
Value 0
              }
          }
    - name: Import AdmPwd.PS module and extend AD schema
```

```
  ansible.windows.win_powershell:
    script: |
      Import-Module AdmPwd.PS -ErrorAction Stop
      $attr = Get-ADObject -LDAPFilter "(name=ms-Mcs-AdmPwd)" -ErrorAction SilentlyContinue
      if (-not $attr) {
          Update-AdmPwdADSchema
      }
- name: Set self-permissions for computers in target OUs
  ansible.windows.win_powershell:
    script: |
      Import-Module AdmPwd.PS -ErrorAction Stop
      $ous = @(
      {{ laps_target_ous | map('regex_replace','^(.*)$','"\1"') | join(",") }}
      )
      foreach ($ou in $ous) {
          Set-AdmPwdComputerSelfPermission -OrgUnit $ou
      }
- name: Allow Domain Admins to read LAPS passwords in target OUs
  ansible.windows.win_powershell:
    script: |
      Import-Module AdmPwd.PS -ErrorAction Stop
      $ous = @(
      {{ laps_target_ous | map('regex_replace','^(.*)$','"\1"') | join(",") }}
      )
      foreach ($ou in $ous) {
          Set-AdmPwdReadPasswordPermission -OrgUnit $ou -AllowedPrincipals "Domain Admins"
      }
- name: Create LAPS GPO if not exists
  ansible.windows.win_powershell:
    script: |
      Import-Module GroupPolicy -ErrorAction Stop
      $name = "{{ laps_gpo_name }}"
      $gpo = Get-GPO -Name $name -ErrorAction SilentlyContinue
      if (-not $gpo) {
          New-GPO -Name $name
      }
```

```
  - name: Configure LAPS settings in GPO
    ansible.windows.win_powershell:
      script: |
        Import-Module GroupPolicy -ErrorAction Stop
        $name = "{{ laps_gpo_name }}"
        $gpo = Get-GPO -Name $name -ErrorAction Stop
        $key = "HKLM\Software\Policies\Microsoft Services\AdmPwd"
        Set-GPRegistryValue -Name $name -Key $key -ValueName "AdmPwdEnabled" -Type DWord -Value 1
        Set-GPRegistryValue -Name $name -Key $key -ValueName "AdminAccountName" -Type String -Value
"{{ laps_admin_account }}"
        Set-GPRegistryValue -Name $name -Key $key -ValueName "PasswordAgeDays" -Type DWord -Value
{{ laps_password_age_days }}
        Set-GPRegistryValue -Name $name -Key $key -ValueName "PasswordLength" -Type DWord -Value
{{ laps_password_length }}
        Set-GPRegistryValue -Name $name -Key $key -ValueName "PasswordComplexity" -Type DWord -
Value 4
        Set-GPRegistryValue -Name $name -Key $key -ValueName "PasswordReusePrevention" -Type DWord -
Value 1
        Set-GPRegistryValue -Name $name -Key $key -ValueName "PwdExpirationProtectionEnabled" -Type
DWord -Value 1
        $hideKey = "HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Winlogon\SpecialAccounts\UserList"
        Set-GPRegistryValue -Name $name -Key $hideKey -ValueName "{{ laps_admin_account }}" -Type
DWord -Value 0


  - name: Link LAPS GPO to workstation OUs
    ansible.windows.win_powershell:
      script: |
        Import-Module GroupPolicy -ErrorAction Stop
        $name = "{{ laps_gpo_name }}"
        $ous = @(
        {{ laps_target_ous | map('regex_replace','^(.*)$','"\1"') | join(",") }}
        )
        foreach ($ou in $ous) {
          $inheritance = Get-GPInheritance -Target $ou
          $existing = $inheritance.GpoLinks | Where-Object { $_.DisplayName -eq $name }
          if (-not $existing) {
```

```
            New-GPLink -Name $name -Target $ou -LinkEnabled Yes
        }
    }

- name: Restore MSI restriction in previous GPOs
  ansible.windows.win_powershell:
    script: |
      Import-Module GroupPolicy -ErrorAction Stop
      $gpoNames = @(
      {{ laps_restriction_gpos | map('regex_replace','^(.*)$','"\1"') | join(",") }}
      )
      foreach ($gpoName in $gpoNames) {
        $gpo = Get-GPO -Name $gpoName -ErrorAction SilentlyContinue
        if ($gpo) {
          $key = "HKLM\Software\Policies\Microsoft\Windows\Installer"
          Set-GPRegistryValue -Name $gpoName -Key $key -ValueName "DisableMSI" -Type DWord -
Value 2
        }
      }
- name: Show basic LAPS status for WS-01
  ansible.windows.win_powershell:
    script: |
      Import-Module ActiveDirectory -ErrorAction Stop
      $comp = Get-ADComputer "WS-01" -Properties "ms-Mcs-AdmPwd","ms-Mcs-
AdmPwdExpirationTime"
      $comp | Select-Object Name,"ms-Mcs-AdmPwd","ms-Mcs-AdmPwdExpirationTime"
  register: laps_status
- name: Print LAPS status debug
  ansible.builtin.debug:
    var: laps_status.stdout_lines
```

Лістинг плейбука резервного копіювання playbooks/backup_dc.yml

```
---
- name: Create backup for DC
  hosts: dc
  gather_facts: no
```

```
  vars:
    backup_target_volume: "E:"
    backup_task_name: "Ansible_SystemState_Backup"
    backup_task_time: "00:30"
  tasks:
    - name: Install Windows Server Backup feature
      ansible.windows.win_feature:
        name: Windows-Server-Backup
        state: present
        include_management_tools: yes
    - name: Run 1 time backup to E
      ansible.windows.win_command:
        cmd: wbadmin start systemstatebackup -backupTarget:{{ backup_target_volume }} -quiet
      register: wbadmin_run
      changed_when: "'The backup operation successfully completed' in wbadmin_run.stdout or 'The backup
operation successfully completed' in wbadmin_run.stderr"
    - name: Show wbadmin output
      ansible.builtin.debug:
        var: wbadmin_run.stdout_lines
    - name: Create scheduled task for daily System State backup
      community.windows.win_scheduled_task:
        name: "{{ backup_task_name }}"
        description: "Daily System State backup to {{ backup_target_volume }} initiated by Ansible"
        actions:
          - path: wbadmin
            arguments: "start systemstatebackup -backupTarget:{{ backup_target_volume }} -quiet"
        triggers:
          - type: daily
            start_boundary: "2025-01-01T{{ backup_task_time }}:00"
        username: "SYSTEM"
        run_level: highest
        state: present
        enabled: yes
    - name: List existing backup versions
```

```
    ansible.windows.win_command:
      cmd: wbadmin get versions
    register: wbadmin_versions
  - name: Show backup versions
    ansible.builtin.debug:
      var: wbadmin_versions.stdout_lines
```

Лістинг плейбука налаштування аудиту playbooks/audit_logging.yml

```
- name: Configure audit GPO for domain controllers
  hosts: dc
  gather_facts: no
  vars:
    dc_audit_gpo_name: "GPO_Audit_DomainControllers"
    dc_ou_dn: "OU=Domain Controllers,DC=corpmarkeiev,DC=net"
  tasks:
    - name: Create audit GPO for DCs if not exists
      ansible.windows.win_powershell:
        script: |
          Import-Module GroupPolicy -ErrorAction Stop
          $name = "{{ dc_audit_gpo_name }}"
          $gpo = Get-GPO -Name $name -ErrorAction SilentlyContinue
          if (-not $gpo) {
              New-GPO -Name $name
          }
    - name: Link DC audit GPO to Domain Controllers OU
      ansible.windows.win_powershell:
        script: |
          Import-Module GroupPolicy -ErrorAction Stop
          $name = "{{ dc_audit_gpo_name }}"
          $ou = "{{ dc_ou_dn }}"
          $inheritance = Get-GPInheritance -Target $ou
          $existing = $inheritance.GpoLinks | Where-Object { $_.DisplayName -eq $name }
          if (-not $existing) {
              New-GPLink -Name $name -Target $ou -LinkEnabled Yes
          }
    - name: Configure audit policy for DCs (logon/logoff)
```

```
    ansible.windows.win_powershell:
      script: |
        $name = "{{ dc_audit_gpo_name }}"
        $categories = @(
          "Logon/Logoff",
          "DS Access"
        )
        foreach ($cat in $categories) {
            auditpol /set /policycategory:"$cat" /success:enable /failure:enable
        }
  - name: Configure event log sizes on DCs
    ansible.windows.win_powershell:
      script: |
        wevtutil sl Security /ms:209715200
        wevtutil sl System /ms:52428800
        wevtutil sl Application /ms:52428800
- name: Configure audit GPO for workstations
  hosts: dc
  gather_facts: no
  vars:
    ws_audit_gpo_name: "GPO_Audit_Workstations"
    ws_target_ous:
      - "OU=HR_Department,DC=corpmarkeiev,DC=net"
      - "OU=Developers_Department,DC=corpmarkeiev,DC=net"
      - "OU=IT_Department,DC=corpmarkeiev,DC=net"

  tasks:
    - name: Create audit GPO for workstations if not exists
      ansible.windows.win_powershell:
        script: |
          Import-Module GroupPolicy -ErrorAction Stop
          $name = "{{ ws_audit_gpo_name }}"
          $gpo = Get-GPO -Name $name -ErrorAction SilentlyContinue
          if (-not $gpo) {
```

```
            New-GPO -Name $name
        }
    - name: Link workstation audit GPO to workstation OUs
      ansible.windows.win_powershell:
        script: |
          Import-Module GroupPolicy -ErrorAction Stop
          $name = "{{ ws_audit_gpo_name }}"
          $ous = @(
          {{ ws_target_ous | map('regex_replace','^(.*)$','"\1"') | join(",") }}
          )
          foreach ($ou in $ous) {
            $inheritance = Get-GPInheritance -Target $ou
            $existing = $inheritance.GpoLinks | Where-Object { $_.DisplayName -eq $name }
            if (-not $existing) {
              New-GPLink -Name $name -Target $ou -LinkEnabled Yes
            }
          }
- name: Ensure local audit policy active on DC
  hosts: dc
  gather_facts: no
  tasks:
    - name: Force gpupdate on DC
      ansible.windows.win_command:
        cmd: gpupdate /force
    - name: Show effective audit policy on DC
      ansible.windows.win_command:
        cmd: auditpol /get /category:*
      register: dc_auditpol
    - name: Print DC audit policy
      ansible.builtin.debug:
        var: dc_auditpol.stdout_lines
- name: Ensure local audit policy active on workstations
  hosts: clients
  gather_facts: no
  tasks:
    - name: Force gpupdate on workstation
```

```
    ansible.windows.win_command:
      cmd: gpupdate /force
  - name: Show effective audit policy on workstation
    ansible.windows.win_command:
      cmd: auditpol /get /category:*
    register: ws_auditpol
  - name: Print workstation audit policy
    ansible.builtin.debug:
      var: ws_auditpol.stdout_lines
```

Лістинг плейбука симуляції атаки playbooks/attack_sim.yml

```
- name: Simulate Pass-the-Hash attack on WS-01
  hosts: localhost
  gather_facts: no
  vars_files:
    - ../inventory/group_vars/windows/attack.yml

  vars:
    nmap_log: "./logs/nmap_ws01_{{ lookup('pipe','date +%Y%m%d_%H%M%S') }}.log"
    pth_log: "./logs/pth_ws01_{{ lookup('pipe','date +%Y%m%d_%H%M%S') }}.log"
  tasks:
    - name: Ensure logs directory exists
      ansible.builtin.file:
        path: "./logs"
        state: directory
        mode: "0755"

    - name: Scan SMB port 445 on WS-01
      ansible.builtin.command:
        cmd: "nmap -p 445 -Pn -n {{ attack_target_ip }}"
      register: nmap_scan
      changed_when: false

    - name: Save Nmap output to file
      ansible.builtin.copy:
```

```
    dest: "{{ nmap_log }}"
    content: "{{ nmap_scan.stdout }}\n{{ nmap_scan.stderr }}"
  delegate_to: localhost

- name: Check if port 445 is open
  ansible.builtin.set_fact:
    smb_open: "{{ '445/tcp open' in nmap_scan.stdout }}"

- name: Stop if SMB is not accessible
  ansible.builtin.fail:
    msg: "Ціль {{ attack_target_ip }} недоступна або порт 445 закритий. Симуляція не виконується."
  when: not smb_open
- name: Run Impacket wmiexec with NTLM hash against WS-01
  ansible.builtin.command:
    cmd: >
      wmiexec.py
      -hashes {{ attack_ntlm_hash }}
      {{ attack_username }}@{{ attack_target_ip }}
      "whoami"
  register: pth_result
  changed_when: false
  failed_when: false
- name: Save Pass-the-Hash output to file
  ansible.builtin.copy:
    dest: "{{ pth_log }}"
    content: |
      COMMAND:
      wmiexec.py -hashes {{ attack_ntlm_hash }} {{ attack_username }}@{{ attack_target_ip }} "whoami"

      RETURN CODE:
      {{ pth_result.rc }}

      STDOUT:
      {{ pth_result.stdout }}
      STDERR:
      {{ pth_result.stderr }}
```

```
  delegate_to: localhost
- name: Determine if Pass-the-Hash attack succeeded
  ansible.builtin.set_fact:
    pth_success: >-
      {{ (pth_result.rc == 0)
        and ('Access is denied' not in pth_result.stdout)
        and ('STATUS_LOGON_FAILURE' not in pth_result.stderr)
        and ('Logon failure' not in pth_result.stdout) }}
- name: Print final verdict
  ansible.builtin.debug:
    msg: >-
      {% if pth_success %}
      FAIL: Атака пройшла успішно. Result: LAPS не працює або налаштований некоректно.
      Деталі див. у файлі {{ pth_log }}.
      {% else %}
      SUCCESS: Атака заблокована. Result: LAPS успішно захистив систему.
      Деталі див. у файлі {{ pth_log }}.
      {% endif %}
```