Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В.о. завідувача кафедри кібербезпеки та захисту інформації
_____Іван ПАРХОМЕНКО
«____» травня 2025 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань 12 *Інформаційні технології*
(шифр і назва галузі знань)

спеціальність *125 Кібербезпека та захист інформації*
(код і назва спеціальності)

освітній ступень *магістр*

освітньо-наукова програма *Кібербезпека*
(назва освітньо-професійної програми)

на тему: «Модель проактивного виявлення та запобігання кіберзагроз на основі машинного навчання»

Виконавець: студент ІІ курсу, групи КБм-22

_____ **Іван НЕЧИПОРЕНКО**
(підпис) (ім'я прізвище)

	Ім'я, ПРІЗВИЩЕ	**Підпис**
Науковий керівник	Лариса МИРУТЕНКО	
Нормоконтроль	Юрій БАБЕНКО	

Київ 2025

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедри
кібербезпеки та захисту
інформації
_____Іван ПАРХОМЕНКО
«25» жовтня 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності *125 Кібербезпека та захист інформації*
(код і назва спеціальності)

освітній ступень *магістр*

Здобувача **КБм-22** (група) **Нечипоренка Івана Петровича** (прізвище ім'я по батькові)

Тема кваліфікаційної роботи Модель проактивного виявлення та запобігання кіберзагроз на основі машинного навчання

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Рішення засідання кафедри кібербезпеки та захисту інфрмації факультету інформаційних технологій протокол № 4 від 24.10.2024 р.

2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень	Процес виявлення аномальної активності в середовищі Active Directory
Предмет досліджень	Методи логування, аналізу та машинного навчання для виявлення аномалій
Мета	Розробка та апробація поведінкової моделі виявлення атак у середовищі Active Directory
Вихідні дані для проведення роботи	Джерела подій на доменних контролерах, а саме Kerberos, Security, інфраструктура лабораторного стенду GOAD v3, стек Elasticsearch для збору й аналізу подій та інструменти симуляції атак.

3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна Покращення існуючих методів виявлення аномальної активності в середовищі Active Directory

Практична цінність Налаштована модель виявлення атак на середовище Active Directory

4. ЕТАПИ ВИКОНАННЯ РОБОТИ

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
1	Уточнення постановки задачі	25.10.2024 – 12.11.2024	*виконано*
2	Аналіз літературних джерел щодо виявлення атак в Active Directory	13.11.2024 – 12.02.2025	*виконано*
3	Аналіз методологій виявлення компрометації аутентифікації та аномалій	13.02.2025 – 21.02.2025	*виконано*
4	Долідження теоретичних аспектів оцінки захищеності аутентифікації у середовищі AD	22.02.2025 – 04.03.2025	*виконано*
5	Вибір методології та алгоритмів машинного навчання	05.03.2025 – 10.03.2025	*виконано*
6	Підготовка даних та конфігурація лабораторного середовища (GOAD v3, Elastic Stack)	11.03.2025 – 19.03.2025	*виконано*
7	Моделювання атак, навчання моделі, аналіз результатів адекватності та точності	20.03.2025 – 17.04.2025	*виконано*
8	Оформлення пояснювальної записки згідно з методичними рекомендаціями	18.04.2025 – 15.05.2025	*виконано*
9	Подача пакету документів на розгляд Екзаменаційної комісії	15.05.2025 – 19.05.2025	*виконано*

Завдання видала _____ Лариса МИРУТЕНКО
(підпис) (ініціали, прізвище)

Завдання прийняв до виконання _____ Іван НЕЧИПОРЕНКО
(підпис) (ініціали, прізвище)

Дата видачі завдання: 25.10.2024 р.
Термін подання кваліфікаційної роботи до ЕК 19.05.2025 р.

УДК 004.056.5

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Модель проактивного виявлення та запобігання кіберзагроз на основі машинного навчання» містить 105 сторінок (без додатків), 51 рисунок та 2 таблиці. Використано 52 літературних джерел.

Об’єкт дослідження — процес виявлення та протидії кіберзагрозам у корпоративних середовищах з використанням служби Active Directory.

Предмет дослідження — методи логування, аналізу та машинного навчання для виявлення аномалій.

Мета роботи — створення моделі виявлення та запобігання складним кібератакам у середовищі Active Directory з використанням методів машинного навчання, реалізованих у стеку Elastic Stack.

Методи дослідження — аналіз технічної документації, поведінкове моделювання, застосування емуляції атак (Kerberoasting, Golden Ticket), побудова моделі поведінкового аналізу.

У роботі досліджено особливості виявлення атак у доменному середовищі Active Directory, які не потребують доставки шкідливого ПЗ на цільові хости. На основі цього реалізовано централізований збір подій, розгорнуто лабораторне середовище GOAD v3, здійснено симуляцію складних атак та побудовано модель виявлення аномальної активності за допомогою модуля Anomaly Detection Elastic Stack.

Наукова новизна: вперше реалізовано модель виявлення атак на основі подій автентифікації, які здійснюються в рамках легітимних протоколів Active Directory, без застосування інструментів взлому на хостах. Запропоновано сценарії симуляцій атак та побудовано відповідну модель виявлення на основі рідкісних відхилень у поведінці користувачів.

Актуальність теми: у сучасних умовах корпоративні мережі стають основною ціллю для складних атак, зокрема через компрометацію механізмів автентифікації в Active Directory. Традиційні антивірусні та EDR-рішення не здатні виявити атаки, що виконуються з використанням штатних протоколів та інструментів. Це вимагає побудови поведінкових моделей, здатних виявити аномалії без залучення сигнатур або агентів. Представлена в роботі модель дозволяє суттєво підвищити рівень захищеності корпоративної інфраструктури.

Ключові слова: Active Directory, Elastic Stack, Kerberos, Kerberoasting, Golden Ticket, поведінковий аналіз, машинне навчання, виявлення аномалій, кіберзагрози, SIEM.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

AD	–	Active Directory – служба каталогів Microsoft для управління обліковими записами та ресурсами у доменах
DC	–	Domain Controller – доменний контролер
LDAP	–	Lightweight Directory Access Protocol – протокол доступу до каталогу
RPC	–	Remote Procedure Call – віддалений виклик процедур
DNS	–	Domain Name System – система доменних імен
SMB	–	Server Message Block – протокол доступу до файлів та принтерів
Kerberos	–	Протокол автентифікації на основі квитків (tickets)
NTLM	–	NT LAN Manager – застарілий протокол автентифікації
NetNTLM	–	Модифікація NTLM, що використовується в мережевому середовищі
KDC	–	Key Distribution Center – центр розподілу ключів у Kerberos
SID	–	Security Identifier – унікальний ідентифікатор об’єкта безпеки
SPN	–	Service Principal Name – ім’я службового облікового запису
TGT	–	Ticket Granting Ticket – квиток для отримання службових квитків (Kerberos)
TGS	–	Ticket Granting Service – служба видачі службових квитків (Kerberos)
GPO	–	Group Policy Object – групова політика Windows
EDR	–	Endpoint Detection and Response – рішення для виявлення атак на кінцевих точках
AV	–	Antivirus – антивірусне програмне забезпечення
SIEM	–	Security Information and Event Management – система управління інформацією та подіями безпеки

MITRE	–	Матриця тактик і технік атак, каталогізована MITRE
IOC	–	Indicators of Compromise – індикатори компрометації
TTPs	–	Tactics, Techniques and Procedures – тактики, техніки і процедури атак
FP	–	False Positive – хибно позитивний результат
FN	–	False Negative – хибно негативний результат
C2	–	Command and Control – канал керування зловмисником
GOAD	–	Game of Active Directory – лабораторне середовище для тестування безпеки AD
GOADv3	–	Версія 3 лабораторного середовища GOAD
EFK	–	Elasticsearch, Fleet, Kibana – стек логування та візуалізації
ElasticML	–	Elastic Machine Learning – модуль машинного навчання в Elasticsearch
Winlogbeat	–	Агент збору логів з Windows для Elastic Stack
API	–	Application Programming Interface – програмний інтерфейс додатків
JSON	–	JavaScript Object Notation – формат структури даних
CSV	–	Comma-Separated Values – текстовий формат з розділенням комами
UI	–	User Interface – інтерфейс користувача
GUI	–	Graphical User Interface – графічний інтерфейс користувача
CLI	–	Command Line Interface – інтерфейс командного рядка
RDP	–	Remote Desktop Protocol – протокол віддаленого доступу
WMI	–	Windows Management Instrumentation – інструментарій для керування Windows
Sysmon	–	System Monitor – утиліта моніторингу подій на рівні ядра
ELK	–	Elasticsearch, Logstash, Kibana – стек для обробки подій та логів
ML	–	Machine Learning – машинне навчання
AI	–	Artificial Intelligence – штучний інтелект

ADCS	–	Active Directory Certificate Services – служба сертифікації Microsoft
AS-REP	–	Відповідь на запит автентифікації без попередньої автентифікації (вразливість Kerberos)
ESC1-ESC8	–	Класи вразливостей в ADCS, описані SpecterOps
Anomaly Detection	–	Механізм виявлення аномалій у Elastic Stack
BloodHound	–	Інструмент для аналізу взаємозв'язків у AD
impacket	–	Набір бібліотек Python для роботи з мережевими протоколами Microsoft
.ccache	–	Формат квитків Kerberos для повторного використання

ЗМІСТ

ВСТУП

Інформаційні системи стали основою сучасного цифрового світу. Кожна організація — від малих підприємств до великих міжнародних корпорацій — активно використовує інформаційно-комунікаційні технології для підтримки бізнес-процесів, управління персоналом, зберігання та обробки даних.

У цьому контексті Active Directory (AD) виступає ключовою складовою корпоративної IT-інфраструктури, забезпечуючи централізоване управління обліковими записами, доступами та політиками безпеки. За даними статистики, понад 95% організацій світу використовують Active Directory, що робить його пріоритетною ціллю для кібератак.

Сучасні загрози, пов'язані з Active Directory, стають дедалі складнішими. Зловмисники дедалі частіше використовують тактики, що не потребують доставки традиційного шкідливого програмного забезпечення (malware) до цільової системи. Техніки Living off the Land (LotL) дозволяють використовувати вбудовані системні утиліти для прихованого переміщення мережею, ескалації привілеїв та контролю над середовищем. Це суттєво ускладнює виявлення атак за допомогою класичних методів, заснованих на сигнатурах або відомих індикаторах компрометації (IoC).

У зв’язку з цим виникає необхідність у впровадженні проактивних методів захисту, орієнтованих на виявлення аномалій у поведінці користувачів і сервісів. Саме такою є концепція поведінкового аналізу та виявлення відхилень, що набуває особливого значення у середовищі Active Directory.

Сучасні платформи, такі як Elastic Stack, дозволяють реалізувати подібний підхід. Інтеграція компонентів для збору логів (Winlogbeat, Sysmon, Audit Policy), обробки даних (Logstash, Elasticsearch) та візуалізації (Kibana) створює потужну основу для побудови системи виявлення загроз. Використання модуля Machine Learning в Elasticsearch дає змогу створювати моделі нормальної активності та автоматично виявляти відхилення, що

потенційно вказують на кіберзагрозу. Це дозволяє економити людські ресурси для аналітики подій безпеки, а також автоматизовано та швидко виявляти аномалії на фоні всіх родій, що передаються моделі для аналітики.

РОЗДІЛ 1
ACTIVE DIRECTORY ЯК ОБ'ЄКТ КІБЕРЗАГРОЗ

1.1 Значення Active Directory в корпоративних мережах

Active Directory (AD) — це служба каталогів від компанії Microsoft, призначена для централізованого керування Windows-мережами у корпоративному середовищі. Вона зберігає інформацію про користувачів, комп’ютери, групи та інші мережеві об’єкти і контролює доступ до них, відіграючи ключову роль у безпечному адмініструванні IT-інфраструктури організації. AD є основним компонентом серверної ОС Windows і широко використовується в компаніях різного масштабу як центральне сховище облікових записів і політик, що спрощує управління мережею.

Active Directory є ключовим елементом більшості корпоративних інформаційних систем. За різними оцінками, понад 95% компаній зі списку Fortune 500 використовують AD як базову службу для управління обліковими записами користувачів, автентифікації та авторизації в корпоративних мережах. Його популярність обумовлена гнучкістю, масштабованістю та широкими можливостями централізованого управління доступами до IT-ресурсів.

Active Directory слугує централізованим репозиторієм для усіх облікових записів і ресурсів мережі. Завдяки AD адміністратори можуть в одному місці створювати, змінювати та видаляти облікові записи користувачів, встановлювати їм паролі та призначати права доступу до ресурсів. Дана служба каталогів дозволяє об’єднувати користувачів у групи безпеки, що значно спрощує призначення дозволів – доступ до ресурсів надається групам, а не налаштовується окремо для кожного користувача. Для кінцевих користувачів централізована автентифікація визначає наявність

єдиного облікового запису для виконання всіх завдань, а саме увійшовши в домен один раз, співробітник автоматично отримує доступ до всіх дозволених йому сервісів та даних без повторного введення пароля (функція *Single Sign-On*). Таким чином, AD виконує роль єдиної точки управління обліковими записами і ресурсами, що підвищує ефективність адміністрування та загальний рівень контролю над мережевими ресурсами.

Домен в Active Directory – це логічна група об’єктів (користувачів, комп’ютерів тощо), що зберігаються в одній базі даних каталогу. Кожен домен обслуговується одним або кількома контролерами домену (DC), які зберігають копію цієї бази даних та відповідають за автентифікацію й авторизацію запитів на доступ. Контролер домену фактично є сервером Windows Server із встановленою роллю AD DS (служби доменів Active Directory) і відповідними пакетами програмного забезпечення для реалізації функціоналу відповідних ролей, який виконує перевірку облікових даних користувачів і комп’ютерів та видає їм дозволи згідно з налаштованими політиками.

У разі якщо в домені діє декілька контролерів, між ними реалізована постійна реплікація даних. Зміни, внесені до каталогової бази на одному DC (наприклад, створення чи видалення облікового запису, зміну пароля тощо), автоматично копіюються на інші контролери. Така розподілена модель забезпечує відмовостійкість і високу доступність – навіть якщо один із контролерів вийде з ладу, інший зможе продовжити обслуговувати користувачів. Також варто відзначити гарний рівень оптимізації даного функціоналу, оскільки система починала розвиватися в період коли не було потужних обчислювальних систем і по мірі розвитку системи в ній закладено гарну оптимізацію. Це проявляється в різних варіантах налаштувань реплікації домен контролерів, а також різних варіантах застосування групових політик для з використоанням мережевих файлових систем . Вони продовжують підтримуватя і в нових версіях.

1.2 Історія розвитку Active Directory

Ідея централізованого каталогу для управління обліковими записами бере свій початок з протоколу LDAP, розробленого ще в 1970-х роках для спрощення доступу до ресурсів та об'єктів мережі.

Microsoft вивела цю концепцію на новий рівень, представивши Active Directory разом із релізом Windows 2000 Server. Нова система забезпечила інтеграцію таких ключових протоколів і сервісів, як:

- Kerberos — для безпечної автентифікації на базі квитків;
- DNS — для пошуку і маршрутизації запитів до об'єктів каталогу;
- LDAP — для організації структури каталогу та доступу до неї;
- MS-RPC — для віддаленого виклику процедур і взаємодії між компонентами, наприклад принтерами.

Одночасно із цим у середовищі Windows продовжував використовуватися NTLM, для зберігання облікових даних, а саме паролів у хешованому вигляді що існував з часів Windows NT. Його мережевий варіант — NetNTLM — став стандартом для автентифікації у випадках, коли Kerberos був недоступним (наприклад, під час доступу до спільних папок або взаємодії зі старими системами). Хоча NET-NTLM вважається застарілим, він досі залишається підтримуваним у Active Directory через необхідність реалізації зворотної сумісності.

З роками функціональність Active Directory значно розширилася. З'явилися нові можливості та сервіси, зокрема:

- Forest Trusts — для побудови довірених відносин між доменами;
- Group Policy Objects (GPO) — для централізованого управління налаштуваннями комп'ютерів і користувачів;
- Federation Services (AD FS) — для реалізації єдиного входу (SSO) і інтеграції з веб-додатками;
- Azure AD Connect — для поєднання локальних і хмарних середовищ.

Microsoft, зі свого боку, не припиняє інвестувати в розвиток Active Directory: навіть у нових версіях Windows Server додано поліпшення безпеки, продуктивності й масштабованості AD, що демонструє довгострокову відданість підтримці цієї технології. Сьогодні Active Directory адаптувалася до сучасних потреб бізнесу – вона часто розгорнута в гібридних середовищах разом із Azure AD (в складі хмарного сервісу Microsoft Entra), підтримує багатофакторну автентифікацію та принципи Zero Trust, і залишається опорою для керування доступом у корпоративних мережах.

Таким чином, Active Directory перетворилось на багатофункціональну систему ідентифікації та доступу, яка поєднує в собі як сучасні, так і застарілі механізми автентифікації. Це робить її не лише потужним інструментом для організацій, але й привабливою ціллю для зловмисників, що використовують слабкі місця в реалізації старих протоколів, таких як NetNTLMv1/v2, SMB v1/v2.

1.3 Ключові компоненти та сервіси середовища AD

Active Directory — це не лише база даних, у якій зберігаються облікові записи користувачів. Це комплексна, розподілена система, що забезпечує централізоване управління усіма аспектами корпоративного середовища: ідентифікацією, автентифікацією, доступом до ресурсів і реалізацією політик безпеки.

Основні компоненти та сервіси Active Directory:

1. Domain Services (AD DS) Ядро Active Directory. Надає механізми для зберігання даних про об'єкти (користувачів, комп'ютери, групи), забезпечує їхню уніфікацію та організацію у ієрархічному вигляді. Забезпечує автентифікацію і авторизацію, відповідає за реплікацію змін між доменними контролерами.

2. Group Policy (GPO) Інструмент централізованого управління політиками в домені. Дозволяє адміністраторам встановлювати правила та

обмеження для користувачів і комп'ютерів, автоматизувати налаштування систем та підвищувати безпеку.

3. Domain Name System (DNS) Невід'ємна частина AD. Використовується для пошуку об'єктів у каталозі. Завдяки тісній інтеграції з AD, DNS забезпечує маршрутизацію запитів на сервіси каталогу та правильну роботу доменної інфраструктури, забезпечуючи швидкий пошук корпоративних ресурсів.

4. NetNTLM Застарілий, але досі широко використовуваний протокол автентифікації в AD-середовищах. Використовується в ситуаціях, коли Kerberos недоступний.

5. Kerberos Основний протокол автентифікації в AD. Забезпечує безпечний спосіб підтвердження ідентичності користувачів і сервісів без передачі паролів у відкритому вигляді. В основі лежить модель квитків (tickets), що дозволяє ефективно управляти доступом. Також тісно інтегрується з AD CS.

6. Lightweight Directory Services (AD LDS) Спрощена версія служби каталогів для сценаріїв, де не потрібні домени. Дозволяє організаціям розгортати службу каталогів без створення доменної інфраструктури.

7. Certificate Services (AD CS) Реалізує інфраструктуру відкритих ключів (PKI) в домені. Дозволяє створювати, видавати, створювати шаблони генерації сертифікатів, здійснювати логування видачі сертифікатів та керувати цифровими сертифікатами для забезпечення захищених комунікацій та автентифікації.

8. Federation Services (AD FS) Дозволяє встановлювати довіру між AD та іншими системами і організаціями. Забезпечує єдиний вхід (SSO) для користувачів в гібридних середовищах або для доступу до сторонніх веб-додатків.

Перелік основних компонентів їх опис та основне призначення , представлені в таблиці 1.1.

Таблиця 1.1

Перелік компонентів середовища Active Directory

Компонент	**Опис**	**Основне призначення**
AD DS(Active Directory Domain Services)	Зберігання об'єктів каталогу та автентифікація користувачів.	Ідентифікація, автентифікація, авторизація
GPO (Group Policy Object)	Централізоване управління конфігурацією користувачів і комп'ютерів.	Налаштування системи та політик безпеки
DNS (Domain Name System)	Пошук об'єктів каталогу та забезпечення коректної маршрутизації.	Розпізнавання імен, маршрутизація
NetNTLM	Протокол автентифікації, що використовується у разі відсутності Kerberos.	Автентифікація (локально та по мережі), сумісність
Kerberos	Протокол автентифікації на базі квитків.	Захищена автентифікація
AD LDS (Lightweight Directory Services)	Полегшена служба каталогів для спеціалізованих сценаріїв.	Легка служба каталогів
AD CS (Certificate Services)	Інфраструктура для створення і управління сертифікатами.	Підтримка PKI, автентифікація, шифрування
AD FS (Federation Services)	Федерація з іншими системами, забезпечення SSO.	Єдиний вхід (SSO), інтеграція

Усі перелічені компоненти і сервіси Active Directory взаємодіють між собою для забезпечення надійного та централізованого управління ідентифікацією, автентифікацією та авторизацією в корпоративній мережі.

1.4 Доменний контролер — серце інфраструктури Active Directory

Доменний контролер (Domain Controller, DC) — це центральний компонент архітектури Active Directory, який забезпечує ключові функції автентифікації, авторизації та зберігання об'єктів каталогу. Саме DC обробляє усі запити від користувачів і комп'ютерів для доступу до ресурсів домену. У разі відмови доменних контролерів нормальна робота мережі стає неможливою, оскільки більшість сервісів і доступу до них залежить від відповідей AD.

Доменний контролер виконує такі основні функції:

- Автентифікація користувачів та пристроїв. Використовуючи протоколи Kerberos або NTLM, DC перевіряє облікові дані та видає квитки доступу (Kerberos ticket-granting tickets — TGT).
- Авторизація запитів на доступ. На основі наданих квитків і політик безпеки DC визначає, які ресурси доступні для певного облікового запису.
- Реплікація каталогу. Усі доменні контролери в домені синхронізують між собою зміни для забезпечення консистентності об'єктів AD.
- Застосування групових політик. DC розсилає GPO для реалізації правил безпеки і конфігурацій на робочих станціях і серверах, що реалізовує можливість централізовано керувати користувачами та комп’ютерами..
- Обслуговування DNS-запитів. Доменні контролери, як правило, також виконують роль DNS-серверів, що забезпечують належну маршрутизацію в межах домену.

1.4.1 Поверхня атаки доменного контролера

Для виконання своїх завдань DC відкриває значну кількість мережевих портів. Саме наявність відкритих сервісів робить контролер пріоритетною ціллю для атакуючих. Навіть звичайний мережевий користувач може ініціювати з’єднання з доменними сервісами на відкритих портах. Нижче наведено використання мережевого сканера nmap для отримання інформації про активні сервіси на сервері DC:

```
sudo nmap -sC -sV -p- -T4 192.168.122.165
```

sudo — використання привілеїв адміністратора необхідне для повного доступу до низькорівневих функцій сканування, а саме встановлення TCP з’єднань.

nmap — утиліта для сканування мереж.

-sC — використання вбудованих скриптів (NSE, Nmap Scripting Engine) для збору додаткової інформації про сервіси, такі як версії, банери, підтримувані протоколи. Це дозволяє одразу побачити специфіку деяких сервісів (наприклад, чи доступний сервіс WinRM).

-sV — визначення версій сервісів, що працюють на знайдених портах. Дає змогу ідентифікувати не лише відкриті порти, а й конкретні програми та їх версії.

-p- — повне сканування всіх 65535 TCP портів. Це важливо для виявлення нетипових та динамічних портів, що використовуються, наприклад, для RPC або сторонніх сервісів, розгортання яких на DC несе пряму загрозу.

-T4 — встановлення високої швидкості сканування (aggressive timing policy), що дозволяє пришвидшити процес при прийнятному рівні навантаження на мережу.

192.168.10.10 — IP-адреса цільового доменного контролера в лабораторному середовищі.

Результат сканування наведено на рисунку 1.1

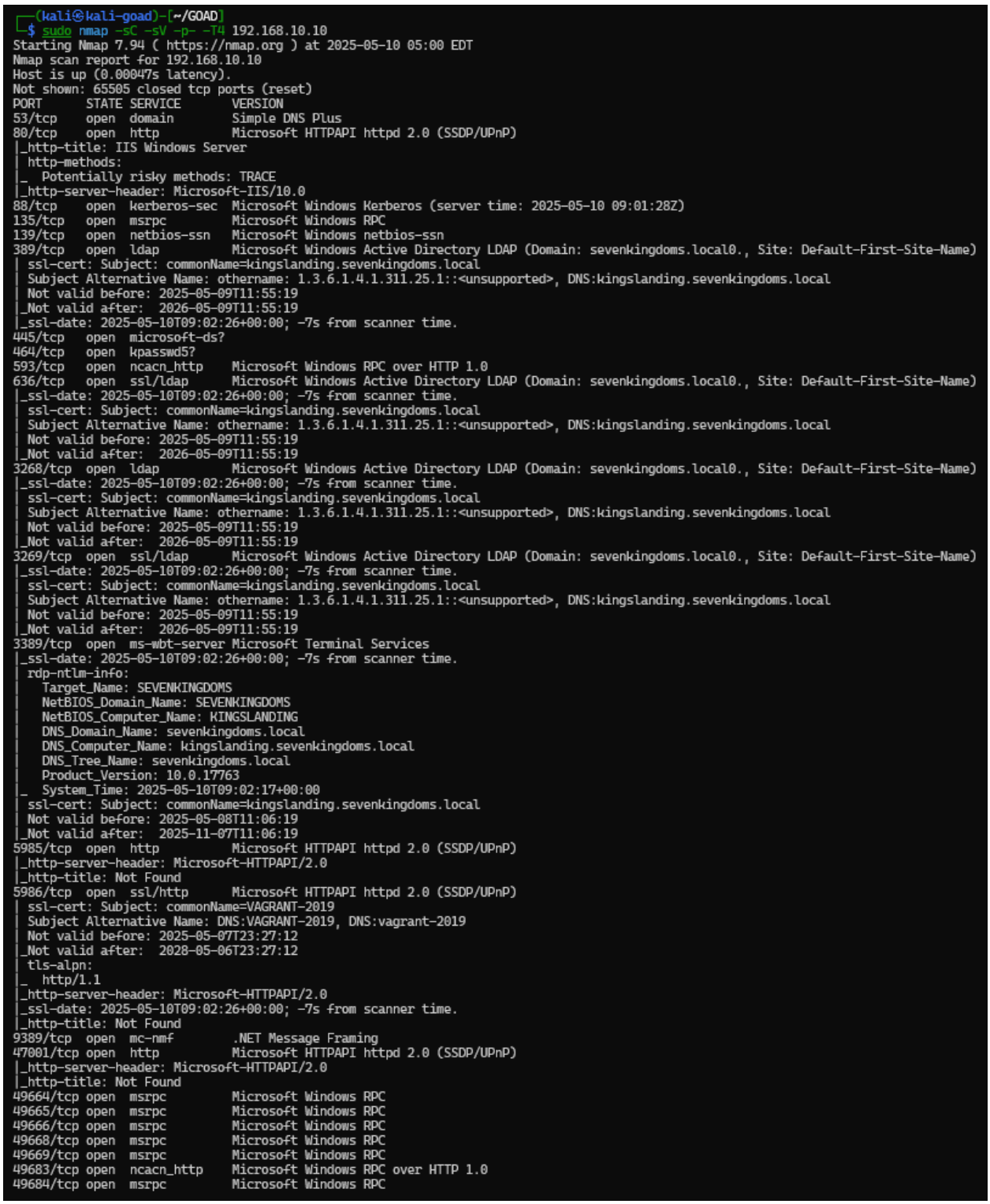


Рисунок 1.1 — Результати сканування доменного контролера за допомогою Nmap

Як показало сканування доменного контролера, для забезпечення функціонування Active Directory необхідна велика кількість відкритих портів, через які здійснюється взаємодія між компонентами системи та клієнтами.

Серед них — критично важливі сервіси, такі як Kerberos, LDAP, Global Catalog, RPC, SMB, а також застарілі або вразливі до атак протоколи, зокрема NetBIOS.

Таблиця 1.2

Відкриті порти на Доменному контролері

Порт	**Протокол**	**Сервіс**	**Опис**
53/tcp	DNS	Simple DNS Plus	Розпізнавання імен у домені
88/tcp	Kerberos	Microsoft Windows Kerberos	Автентифікація користувачів та сервісів
135/tcp	RPC	Microsoft Windows RPC	Віддалене управління і виклики процедур
139/tcp	NetBIOS-SSN	Microsoft NetBIOS	Старий протокол для обміну файлами
389/tcp	LDAP	Microsoft Active Directory LDAP	Доступ до об'єктів каталогу
445/tcp	SMB	Microsoft Directory Services	Доступ до файлів та спільних ресурсів
464/tcp	Kerberos	Microsoft Windows kpasswd5	Зміна паролів користувачів Kerberos
593/tcp	RPC over HTTP	Microsoft Windows RPC	Віддалене адміністрування

продовження таблиці 1.2

636/tcp	LDAPS	TCP Wrapped	Захищений доступ до LDAP
3268/tcp	Global Catalog	Microsoft Active Directory LDAP	Каталог усіх об'єктів лісу
3269/tcp	Global Catalog LDAPS	TCP Wrapped	Захищений доступ до глобального каталогу
3389/tcp	RDP	Microsoft Terminal Services	Дистанційний робочий стіл (RDP)
5985/tcp	HTTP	Microsoft HTTPAPI	Windows Remote Management (WinRM)
9389/tcp	.NET Message Framing	AD Web Services	Web-служби Active Directory
49666/tcp	RPC	Microsoft Windows RPC	Динамічний RPC порт

Наявність такої кількості відкритих сервісів обумовлює широкі можливості низькопривілейованих користувачів для взаємодії з доменним контролером. Зловмисники можуть використовувати ці протоколи як для проведення розвідки (Enumeration), так і для більш небезпечних цілей — атаки на автентифікацію (Pass-the-Hash, Kerberoasting, AS-REP Roasting), захоплення контролю над доменом (DC Sync, DC Shadow), отримання облікових даних користувачів (хешів паролів), оскільки кожен DC містить в собі базу даних NTDS.dit, в якій вони зберігаються а також для організації прихованих каналів та закріплення у мережі.

Саме тому доменний контролер розглядається як центральний і водночас найбільш привабливий об'єкт для атакуючих. Його компрометація

дозволяє отримати повний контроль над усіма об'єктами в домені, що робить захист та моніторинг активності DC надзвичайно важливою задачею в рамках побудови систем виявлення та запобігання кіберзагрозам.

1.5 MITRE ATT&CK і класифікація атак в Active Directory

Завдяки своїй відкритості та складності середовище Active Directory стало основою для розробки численних технік атак, що дозволяють зловмисникам отримувати доступ до критично важливих ресурсів та закріплюватися в мережі, зберігаючи постійну присутність в мережі.

Для систематизації подібних технік і тактик міжнародна спільнота безпеки використовує базу знань MITRE ATT&CK, яка відображає реальні методи та прийоми, що застосовуються під час кібератак.

У контексті Active Directory особливої актуальності набувають наступні тактики та техніки MITRE ATT&CK, що наведені на рисунку 1.2

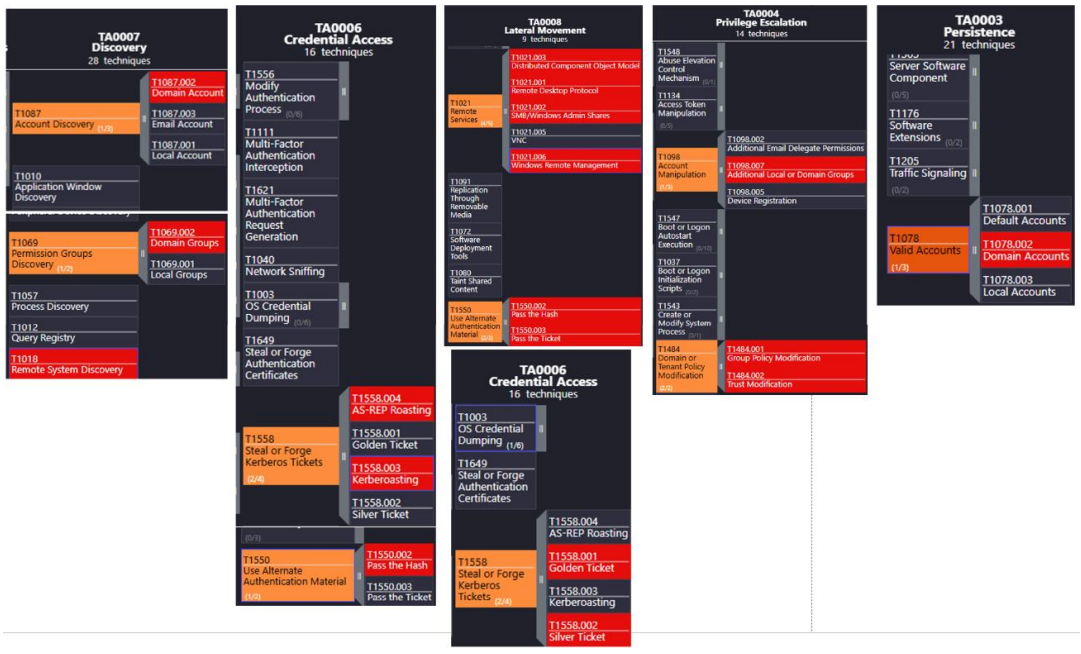


Рисунок 1.2 — Карта технік MITRE ATT&CK, релевантних для атак на Active Directory

В наступних підрозділах буде наведено огляд тактик, технік та інструментів, що використовуються зловмисниками для атак на компоненти середовища Active Directory.

1.5.1 Discovery (Виявлення)

Ціль — здійснення збіру даних про структуру домену, а саме облікові записи, парольні політики, мережеві ресурси.

- T1069.002 — Permission Groups Discovery: Domain Groups Дозволяє ідентифікувати критично важливі групи, такі як Domain Admins та Enterprise Administrators .
- T1087.002 — Account Discovery: Domain Account Отримання переліку облікових записів для подальших атак на автентифікацію.

Протоколи:

- LDAP (389/636 TCP) — Отримання об'єктів каталогу (SharpHound, BloodHound).
- SMB (445 TCP) — Визначення доступних хостів через доступ до спільних ресурсів.
- Kerberos (88 TCP) — Отримання інформації про SPN.

Структуру домену можливо досліджувати як без облікових даних, так і від імені користувача, що знаходиться в домені.

Команди, які можна використати без облікових даних:

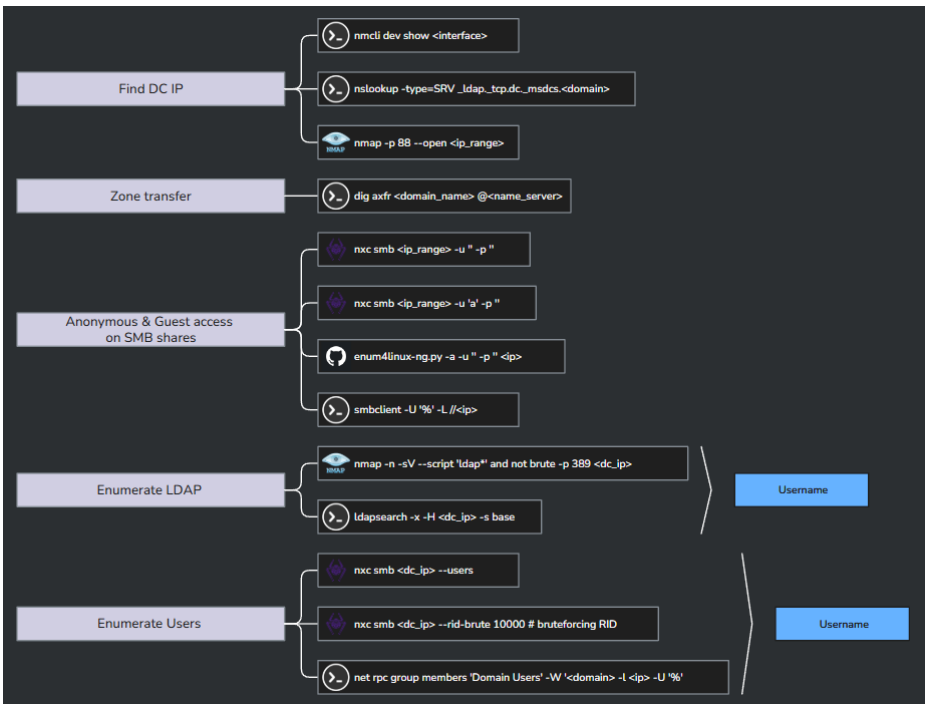


Рисунок 1.3 — Список технік і команд для дослідження домену без облікових даних

Команди для дослідження домену з обліковими даними доменного користувача(Паролі в відкритому вигляді, NT хеш, Kerberos квиток):

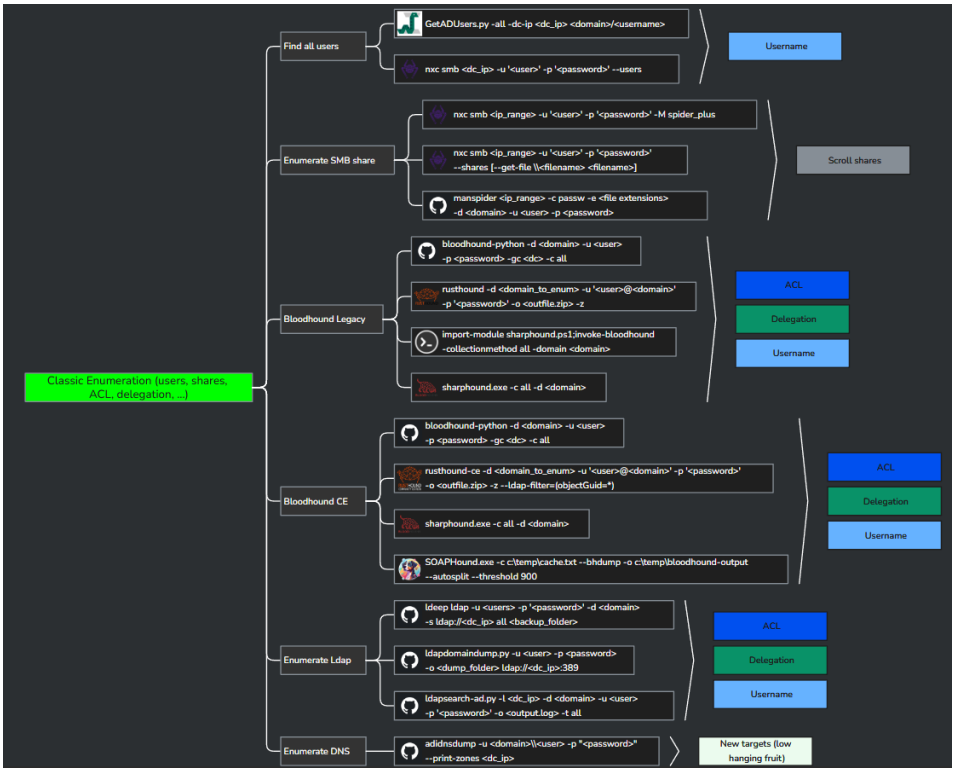


Рисунок 1.4 — Список технік і команд для дослідження домену з обліковими даними доменного користувача

1.5.2 Credential Access (Отримання облікових даних)

Ціль — отримання паролів, хешів або квитків для доступу до системи.

- T1558.003 — Kerberoasting Отримання Kerberos TGS квитків та офлайн-злам паролів службових облікових записів. Протокол: Kerberos (88 TCP).
- T1558.004 — AS-REP Roasting Витяг AS-REP відповідей для облікових записів без вимоги pre-auth. Протокол: Kerberos (88 TCP).
- T1003.006 — DCSync Використання прав на реплікацію для отримання NTLM/AES хешів. Протокол: LDAP (389/636 TCP), RPC (135 TCP, 49152–65535 TCP).
- T1550.002 — Pass the Hash Використання NTLM-хешу для автентифікації замість паролю в чистому вигляді.
- Протокол: SMB (445 TCP).

Інструменти: Mimikatz, Rubeus, Impacket, CrackMapExec, nxc, GetUserSPNs.py.

Існують можливості отримати облікові дані користувачів як без використання низькопривілейованого доменного облікового запису, так і з ним. Повний перелік інструментів та їх параметрів наведено на рисунках 1.5 та 1.6 відповідно.

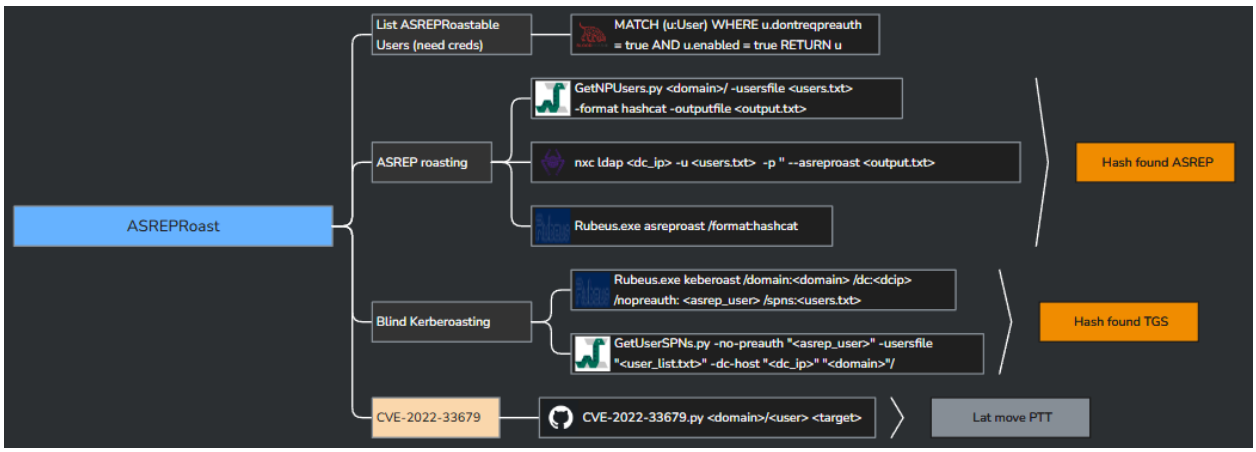


Рисунок 1.5 —Перелік команд для пошуку облікових даних користувачів без використання акаунту валідного облікового запису

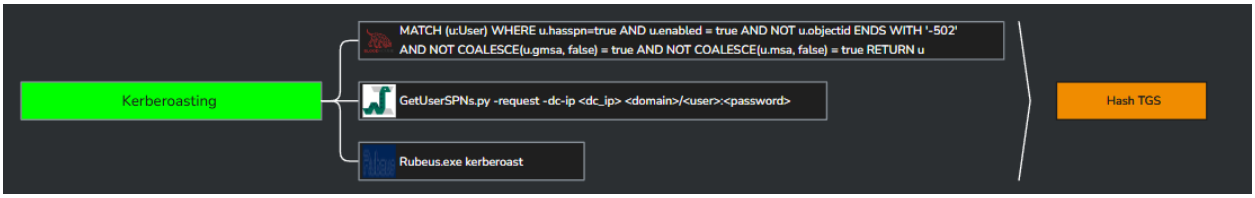


Рисунок 1.6 — Перелік команд для отримання облікових даних користувачів з валідного облікового запису

1.5.3 Lateral Movement (Переміщення мережею)

Ціль — розширення доступу після компрометації облікового запису для подальшої компрометації ресурсів мережі або підвищення привілеїв.

- T1021.001 — Remote Desktop Protocol (RDP) Використання RDP для підключення до систем і виконання дій від імені скомпрометованого облікового запису. Протокол: RDP (3389 TCP).
- T1021.002 — SMB/Windows Admin Shares Використання адміністративних спільних папок (ADMIN$, C$, IPC$) для копіювання файлів, модифікації існуючих з виуористанням іменованих каналів та виконання команд віддалено. Протокол: SMB (445 TCP).
- T1021.003 — Distributed Component Object Model (DCOM) Використання DCOM для віддаленого запуску об'єктів COM, що призводить до віддаленого виконання коду. Протокол: RPC/DCOM (135 TCP, 49152–65535 TCP).
- T1021.006 — Windows Remote Management (WinRM) Використання протоколу віддаленого адміністрування WinRM для виконання команд через оболонку PowerShell. Протоколи: HTTP (5985 TCP), HTTPS (5986 TCP).
- T1075 — Pass the Hash Використання NTLM-хешів для автентифікації та просування мережею. Протоколи: SMB (445 TCP), WinRM (5985/5986 TCP), RPC.

- T1550.001 — Pass the Ticket Використання квитка Kerberos TGT для доступу до ресурсів у мережі без необхідності автентифікації за паролем. Протокол: Kerberos (88 TCP).

Інструменти: PsExec, CrackMapExec, SMBExec, Impacket (psexec.py, wmiexec.py), Evil-WinRM, SharpCOM, PowerShell Remoting, RDP.

Повний перелік інструментів, необхідних привілеїв та їх параметрів наведено на рисунках 1.7 та 1.8.

Рисунок 1.7 Перелік команд для горизонтального переміщення за допомогою паролів в чистому вигляді

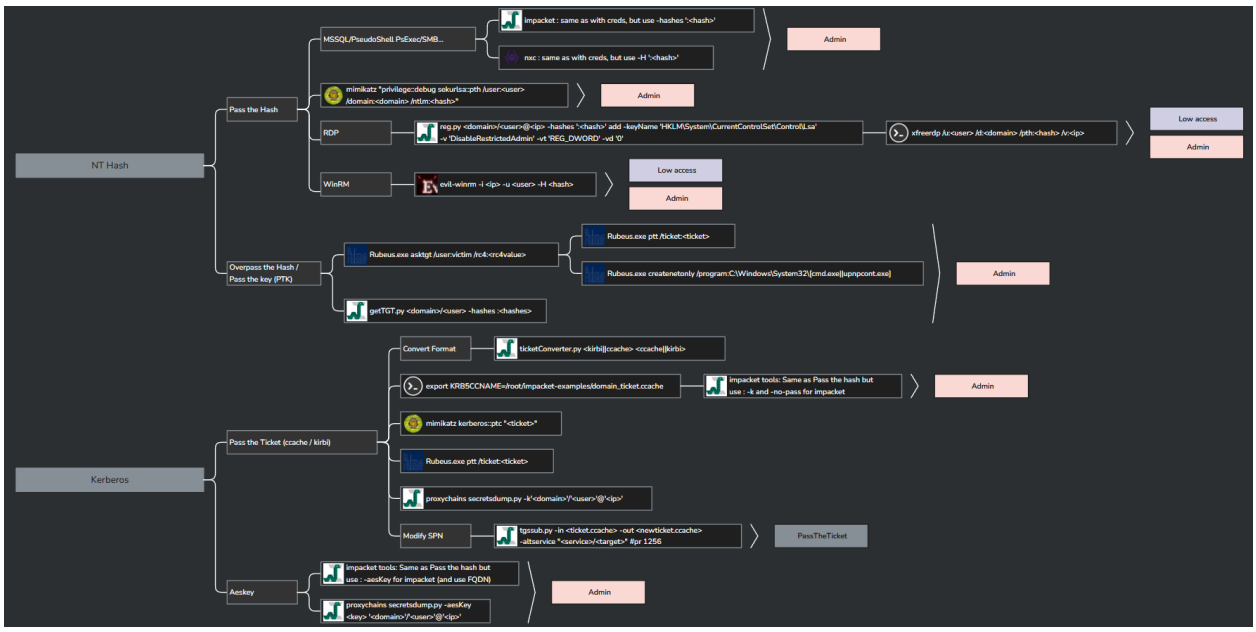


Рисунок 1.8 Перелік інструментів для горизонтального переміщення за допомогою валідного NT хешу або Kerberos квитка

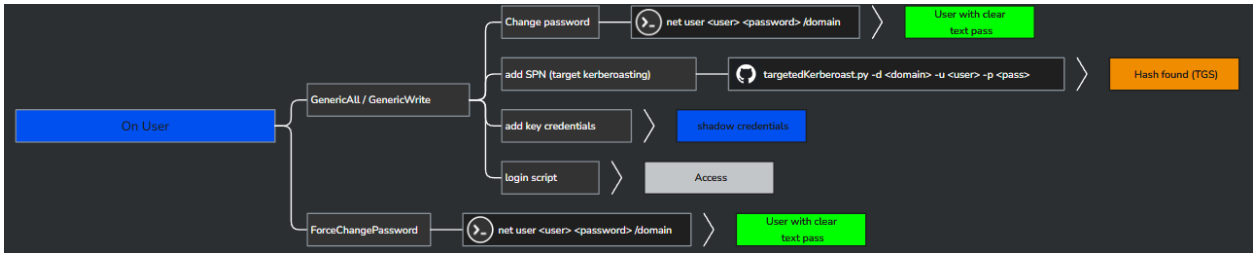
1.5.4 Privilege Escalation (Підвищення привілеїв)

Ціль — отримання повного контролю над середовищем AD.

- T1068 — Exploitation for Privilege Escalation Використання вразливостей.
- T1484.001 — Group Policy Modification Маніпуляція GPO для підвищення привілеїв. Протокол: SMB (445 TCP) — для доставки GPO.
- T1098 — Account Manipulation Створення або зміна облікових записів. Протокол: LDAP (389/636 TCP).

Інструменти: Mimikatz, PowerSploit, BloodHound.

Повний перелік інструментів, необхідних привілеїів та їх параметрів наведено на рисунку 1.9.



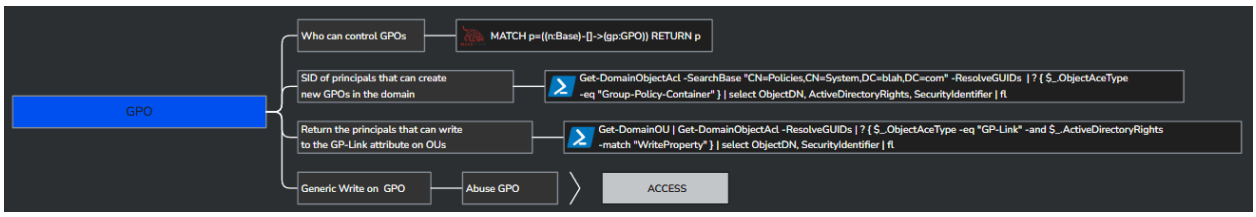


Рисунок 1.9 Список команд для підвищення привілеїв за допомогою маніпуляції акаунтами та груповими політиками

1.5.5 Persistence (Закріплення в системі)

Ціль — збереження доступу в скомпрометованому середовищі.

- T1484.002 — DCShadow Фальсифікація контролреа домену. Протокол: RPC (135 TCP, 49152–65535 TCP).
- T1078 — Valid Accounts Використання легітимних облікових записів.

Інструменти: Mimikatz, Rubeus, Impacket.

1.5.6 Спеціалізовані техніки атак на Kerberos

Окрім стандартних технік, існують спеціалізовані методи атак на Kerberos, які дозволяють зловмисникам отримувати або підробляти квитки для доступу до ресурсів:

- T1558.001 — Golden Ticket Зловмисники, маючи хеш пароля облікового запису KRBTGT, можуть створювати підроблені квитки TGT, що дозволяє їм отримувати доступ до будь-яких ресурсів у домені. Протокол: Kerberos (88 TCP). Сервіси: будь-які AD ресурси.
- T1558.002 — Silver Ticket Використання хешу пароля сервісного облікового запису для створення підроблених квитків TGS, що дозволяє отримувати доступ до сервісів середовища AD без взаємодії з KDC. Протоколи: Kerberos (88 TCP), SMB (445 TCP), HTTP, MSSQL.

- Diamond Ticket Зловмисники отримують легітимний TGT, розшифровують його, модифікують PAC (Privilege Attribute Certificate), перераховують підписи та знову шифрують квиток. Це дозволяє уникнути деяких механізмів виявлення. Протокол: Kerberos (88 TCP).
- Sapphire Ticket Використання техніки S4U2self+U2U для отримання PAC іншого користувача з високими привілеями, який потім вживлюється в легітимний квиток, що дозволяє зловмиснику діяти від імені цього користувача. Протокол: Kerberos (88 TCP).

Інструменти: Mimikatz, Rubeus, Impacket (ticketer.py) Повний перелік інструментів, необхідних привілеїв та їх параметрів наведено на рисунку1.10.

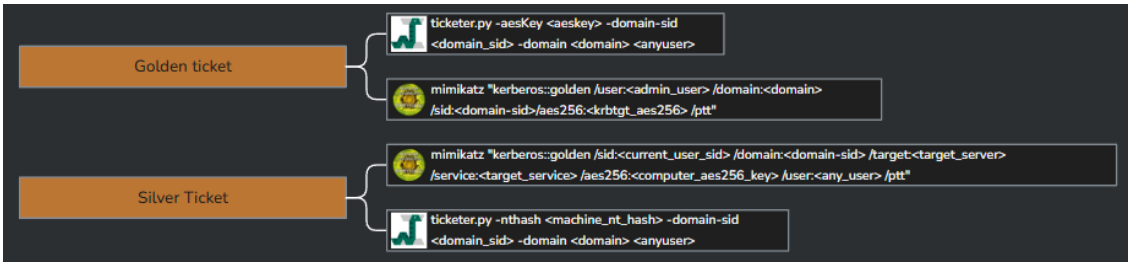


Рисунок 1.10 Список команд для атак на Kerberos

MITRE ATT&CK дозволяє структурувати та класифікувати численні методи атак на Active Directory, охоплюючи усі етапи — від первинного доступу до закріплення в системі. Розуміння цих технік є критично важливим для побудови ефективної системи виявлення аномальної поведінки. Саме виявлення дій, що виходять за межі нормального профілю поведінки користувачів і сервісів AD, є завданням поведінкового аналізу та машинного навчання, які будуть розглянуті у подальших розділах цієї роботи.

1.6 Виклики виявлення атак на основі легітимних дій та протоколів Active Directory

Active Directory побудовано на відкритих і стандартизованих протоколах, таких як Kerberos, LDAP, SMB та RPC. Ці протоколи відіграють

ключову роль у щоденній роботі доменної інфраструктури: забезпечують автентифікацію користувачів, взаємодію між доменними контролерами, доступ до спільних ресурсів та централізоване управління. Саме тому їхнє активне використання в мережі є нормальним і очікуваним явищем.

Однак ця особливість робить Active Directory надзвичайно вразливою до атак, які не виходять за рамки звичайного функціоналу системи. Зловмисники, маючи навіть мінімальний доступ до домену, можуть:

- здійснювати розвідку (enumeration) за допомогою LDAP або SMB;
- запитувати квитки Kerberos для проведення Kerberoasting та інших атак;
- використовувати наявні облікові записи і хеші для lateral movement через SMB та RPC;
- підробляти облікові дані або створювати квитки за допомогою Golden/Silver/Diamond/Sapphire Ticket технік без взаємодії зі сторонніми сервісами.

Усі ці дії виконуються через стандартні та необхідні для функціонування Active Directory протоколи і сервіси, а отже — виглядають легітимно. У багатьох випадках подібна активність є невіддільною частиною повсякденних бізнес-процесів (наприклад, реплікація контролерів домену чи оновлення групових політик). Саме це значно ускладнює роботу класичних систем виявлення загроз (IDS/IPS, SIEM), які в основному орієнтуються на сигнатури або відомі ознаки компрометації.

Додаткову складність становить і той факт, що кожне корпоративне середовище є унікальним. Структура домену, облікові записи, сервіси та првила взаємодії змінюються з часом. Постачальник рішень безпеки не може заздалегідь передбачити всі особливості конкретного оточення, що робить універсальні засоби моніторингу малоефективними.

Таким чином, виявлення атак у середовищі Active Directory вимагає:

- глибокого розуміння поведінкових патернів користувачів і сервісів у кожній конкретній організації;

- динамічного налаштування моніторингу під зміни в інфраструктурі;
- використання методів поведінкового аналізу та машинного навчання для визначення аномалій, що виходять за межі звичайної активності.

Саме побудова моделі поведінки та виявлення відхилень є найперспективнішим підходом до виявлення складних атак на Active Directory, що здійснюються без використання шкідливого ПЗ та із застосуванням легітимних протоколів.

Висновки за розділом 1

Active Directory є критично важливою складовою корпоративних мереж, яка забезпечує централізоване управління обліковими записами, автентифікацією та авторизацією доступу до ресурсів. Водночас його відкритість і складність роблять цю систему привабливою ціллю для атак. Було розглянуто історію розвитку Active Directory, ключові компоненти та сервіси, включно з Kerberos, LDAP, SMB, RPC та іншими протоколами, що забезпечують функціонування доменної інфраструктури.

Проведений аналіз показав, що доменний контролер виступає центральною ланкою, компрометація якої дозволяє зловмиснику отримати практично необмежений доступ до ресурсів організації. Відкрита поверхня атак (відкриті порти і сервіси) дозволяє ініціювати як розвідувальні дії, так і повноцінні атаки без необхідності доставки шкідливого програмного забезпечення.

Розглянута класифікація атак на основі MITRE ATT&CK дозволила систематизувати сучасні техніки, які активно використовуються проти Active Directory. Особлива увага приділена атакам, що експлуатують легітимні функціональні можливості самої системи — таким як Kerberoasting, DCSync, Golden/Silver/Diamond/Sapphire Ticket. Такі техніки дозволяють обійти

традиційні засоби виявлення загроз, оскільки вони ґрунтуються на нормальній взаємодії з протоколами та сервісами AD.

Виявлення подібних атак ускладнюється тим, що поведінка зловмисника практично не відрізняється від дій адміністратора або штатних сервісів. До того ж, універсальні рішення безпеки часто не враховують специфіку конкретної організаційної інфраструктури. У зв'язку з цим стає очевидною потреба у впровадженні поведінкового аналізу, здатного виявляти аномалії на основі відхилень від нормальних шаблонів взаємодії в Active Directory.

Саме побудова моделей поведінки та постійне оновлення знань про активність у мережі дозволить підвищити ефективність виявлення складних атак, що здійснюються без доставки шкідливого ПЗ та з використанням вбудованого функціоналу Windows і Active Directory.

РОЗДІЛ 2
ВИЯВЛЕННЯ АНОМАЛЬНОЇ АКТИВНОСТІ ТА ПОБУДОВА МОДЕЛІ ЗАГРОЗ В ACTIVE DIRECTORY

2.1 Особливості виявлення атак у середовищі Active Directory

Active Directory є критично важливою складовою корпоративної IT-інфраструктури. Його сервіси та протоколи забезпечують централізовану автентифікацію, авторизацію, управління обліковими записами та політиками безпеки. У той же час ці протоколи та сервіси активно використовуються зловмисниками для здійснення атак. Це створює низку викликів для їх виявлення традиційними засобами інформаційної безпеки.

Однією з ключових особливостей атак на Active Directory є використання легітимних протоколів і механізмів взаємодії. Зловмисники не потребують доставки та запуску шкідливого програмного забезпечення — натомість вони активно застосовують штатний функціонал AD для досягнення цілей атаки. Серед найбільш використовуваних протоколів варто виділити:

- Kerberos (порт 88/TCP) — автентифікація користувачів і сервісів.
- LDAP (порт 389/TCP, LDAPS — 636/TCP) — запити до каталогу облікових записів і об'єктів домену.
- SMB (порт 445/TCP) — доступ до спільних ресурсів та файлових систем.
- RPC (порт 135/TCP і динамічні порти) — віддалене адміністрування та реплікація каталогу.

Активність на цих портах і через ці протоколи є нормальною та очікуваною у будь-якому домені, що робить класичні рішення для виявлення атак (IDS/IPS, SIEM, антивіруси) малоефективними. Більшість таких рішень базуються на сигнатурах або IOC (Indicators of Compromise) і орієнтовані на

виявлення відомих шкідливих дій, тоді як атаки в середовищі AD зазвичай не мають характерних ознак компрометації.

Ще одним суттєвим обмеженням є складність створення універсальних правил виявлення. Кожне середовище Active Directory є унікальним. Топологія домену, кількість облікових записів, сервіси та політики відрізняються навіть у схожих організаціях. Універсальні політики та правила можуть бути або надто загальними, що призведе до пропуску атак, або ж занадто жорсткими, що викликатиме численні хибні спрацювання.

Також важливо враховувати, що зловмисники мають доступ до сигнатур як статичних так і поведінкових, через це вони мають змогу адаптовувати свої інструменти, щоб ухилятися від систем захисту тестуючи їх в лабораторному середовищі.

Важливою проблемою є й постійна зміна середовища. Доменна інфраструктура динамічна — створюються та видаляються облікові записи, змінюються ролі, запускаються нові сервіси. Це ускладнює підтримку актуальності правил виявлення на основі статичних підходів.

З огляду на вищезазначене, виявлення атак у середовищі Active Directory потребує використання методів, що дозволяють:

- аналізувати поведінку облікових записів, систем і сервісів у динаміці;
- враховувати контекст активності (хто, де, коли та як виконує ті чи інші дії);
- визначати відхилення від норми, що потенційно вказують на зловмисні дії.

У таких умовах найбільш перспективним підходом є побудова моделі нормальної поведінки користувачів і систем та подальше виявлення аномалій, що виходять за межі цієї моделі. Саме поведінковий аналіз дозволяє виявити приховані атаки, які здійснюються через легітимні протоколи та дії.

Подальші підрозділи цієї роботи будуть присвячені детальному розгляду джерел даних для аналізу, методам побудови моделей нормальної

активності, визначенню аномальних патернів та застосуванню машинного навчання для підвищення ефективності виявлення атак у Active Directory.

2.2 Розширене логування на контролерах домену

Контролери домену (Domain Controllers, DC) є центральними точками автентифікації в середовищі Active Directory (AD). Усі ключові події автентифікації, авторизації та запити до каталогу проходять через них. Тому розширене логування на DC є критично важливим для виявлення та аналізу атак, особливо тих, що використовують легітимні протоколи та сервіси.

Проте, за замовчуванням, детальне логування таких важливих компонентів, як LDAP, Kerberos, а також подій доступу до об'єктів каталогу в Active Directory, або не здійснюється зовсім, або є обмеженим. Це суттєво ускладнює виявлення ознак компрометації, локалізацію інцидентів та подальше розслідування, тобто навіть при бажанні буде важко знайти навіть статичні маркери компрометації, оскільки вони просто не створені у вигляді логів.

У такому вигляді фіксується лише мінімальний набір подій, необхідний для базового аудиту, що недостатньо для виявлення прихованих і складних атак, особливо таких, що базуються на легітимних діях у межах доменної інфраструктури.

У зв'язку з цим, адміністратори безпеки мають забезпечити включення розширених параметрів аудиту для отримання максимально повного журналу подій. Це дозволить не тільки виявляти спроби атак на ранніх стадіях, але й забезпечить збереження критично важливих слідів для подальшого аналізу.

У цьому підрозділі буде розглянуто, які типи подій необхідно активувати, як це зробити, а також які атаки можна виявити за допомогою зібраних журналів.

2.2.1 Логування LDAP-запитів

LDAP (Lightweight Directory Access Protocol) є основним протоколом для доступу до об'єктів каталогу в AD. Зловмисники можуть використовувати LDAP для збору інформації про облікові записи, групи, політики та інші об'єкти. Тому моніторинг LDAP-запитів є важливим для виявлення підозрілої активності.

Для увімкнення логування LDAP-запитів:

1. Відкрийте редактор реєстру (regedit).
2. Перейдіть до гілки:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Diagnostics

3. Створіть або змініть параметр 15 Field Engineering типу DWORD та встановіть значення 5. Зміни відображено на рисунку 2.1.

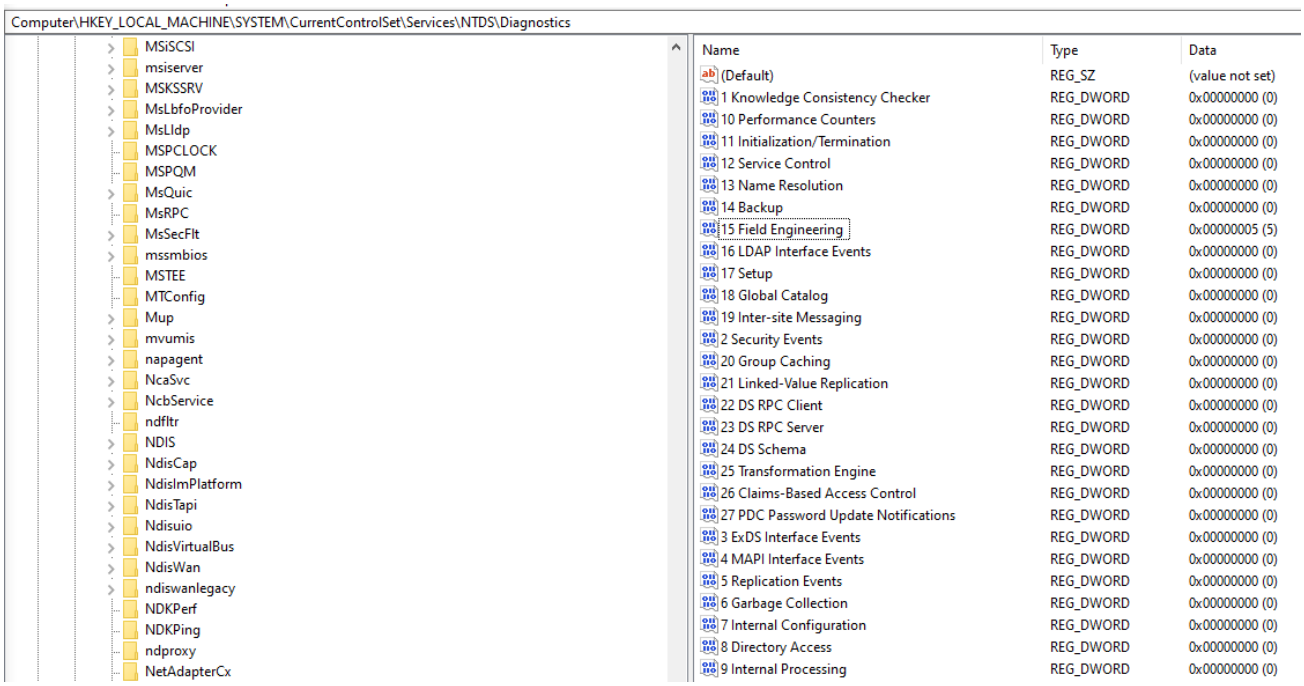


Рисунок 2.1 — модифікація реєстру для розширення логування LDAP

Це дозволить логувати LDAP-запити. Після цього в журналі Directory Service з'являться події:

- 1644 — деталі LDAP-запиту, включаючи ім'я користувача, IP-адресу клієнта, базу пошуку та фільтр.
- 2889 — невдалі або незахищені LDAP-з'єднання, включаючи IP-адресу клієнта.

Ці події можна використовувати для виявлення підозрілих або несанкціонованих запитів до каталогу. Наявність логів у журналі подій відображено на рисунку 2.2

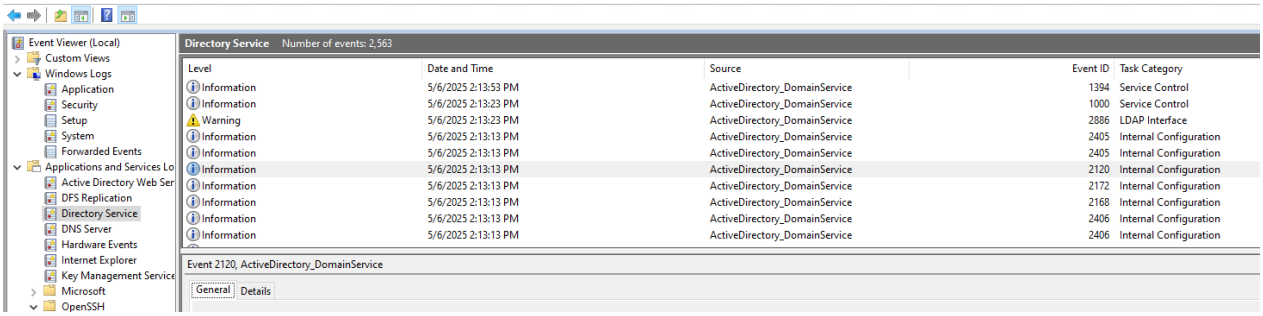


Рисунок 2.2 —відображення логів в журналі подій Directory Service

Відповідні техніки MITRE ATT&CK:

- T1615 — Виявлення політик групи (Group Policy Discovery)
- T1003.006 — DCSync
- T1558.003 — Kerberoasting

Інструменти, які можуть бути виявлені:

- BloodHound
- Mimikatz
- Impacket

2.2.2 Логування подій автентифікації Kerberos

Kerberos є основним протоколом автентифікації в AD. Зловмисники можуть експлуатувати його для отримання квитків доступу або підробки автентифікаційних даних.

Для увімкнення логування подій Kerberos:

1. Відкрийте редактор групових політик (Group Policy Management Editor).
2. Перейдіть до:

Computer Configuration > Policies > Windows Settings > Security Settings > Advanced Audit Policy Configuration > Audit Policies > Account Logon

3. Увімкніть політики:
   - Audit Kerberos Authentication Service
   - Audit Kerberos Service Ticket Operations

Зміни відображено на рисунку 2.3

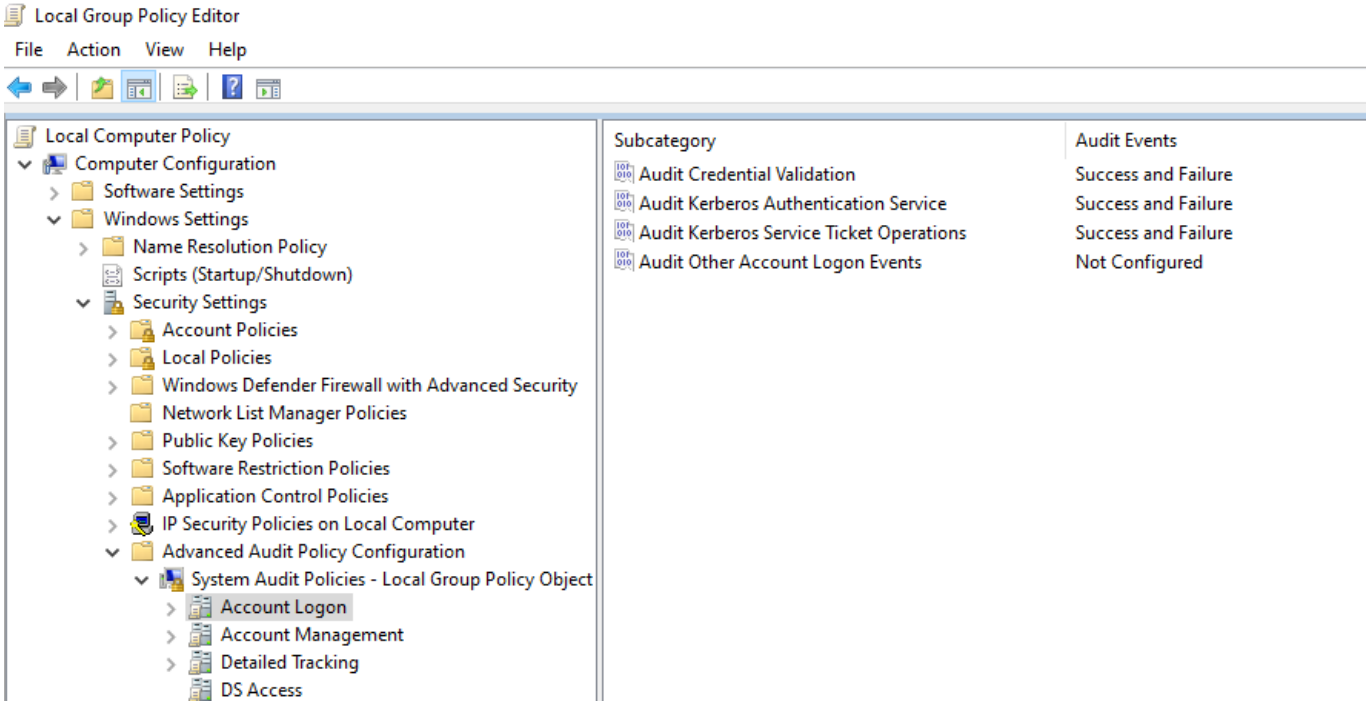


Рисунок 2.3 — модифікації локальних групових політик для аудиту подій Kerberos

Після цього в журналі Security з'являться події:

- **4768** — запит TGT (Ticket Granting Ticket)
- **4769** — запит TGS (Ticket Granting Service)
- **4771** — невдала автентифікація Kerberos
- **4776** — автентифікація за допомогою NTLM

Наявність логів у журналі подій відображено на рисунку 2.4

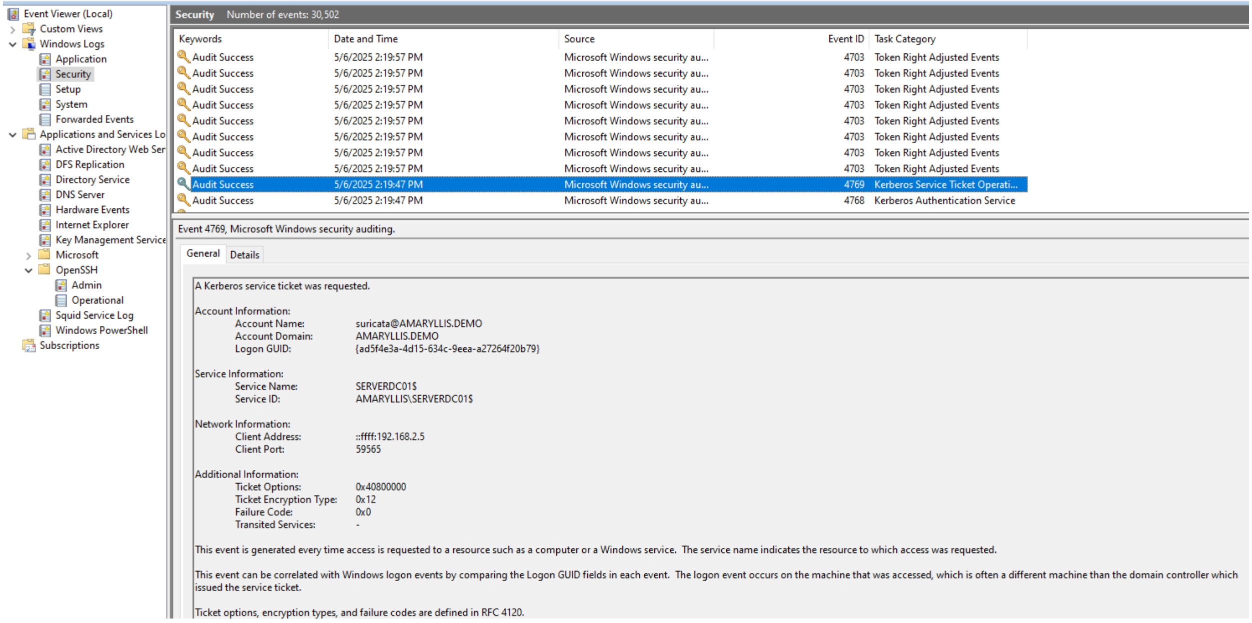


Рисунок 2.4 —відображення логів в журналі подій Kerberos

Ці події допомагають виявляти атаки, такі як Kerberoasting, AS-REP Roasting та використання підроблених квитків.

Відповідні техніки MITRE ATT&CK:

- T1558.003 — Kerberoasting
- T1558.004 — AS-REP Roasting
- T1550.003 — Pass the Ticket
- T1558.001 — Golden Ticket
- T1558.002 — Silver Ticket
- Diamond Ticket
- Sapphire Ticket

Інструменти, які можуть бути виявлені:

- Rubeus
- Impacket
- Mimikatz

2.2.3 Логування доступу до об'єктів каталогу

Зловмисники можуть змінювати об'єкти каталогу для закріплення в системі або підвищення привілеїв. Моніторинг доступу до об'єктів дозволяє виявляти такі дії.

Для увімкнення логування доступу до об'єктів:

1. Відкрийте редактор групових політик (Group Policy Management Editor).

2. Перейдіть до:

Computer Configuration > Policies > Windows Settings > Security Settings > Advanced Audit Policy Configuration > Audit Policies > DS Access

Зміни відображено на рисунку 2.5.

3. Увімкніть політику:

- Audit Directory Service Access

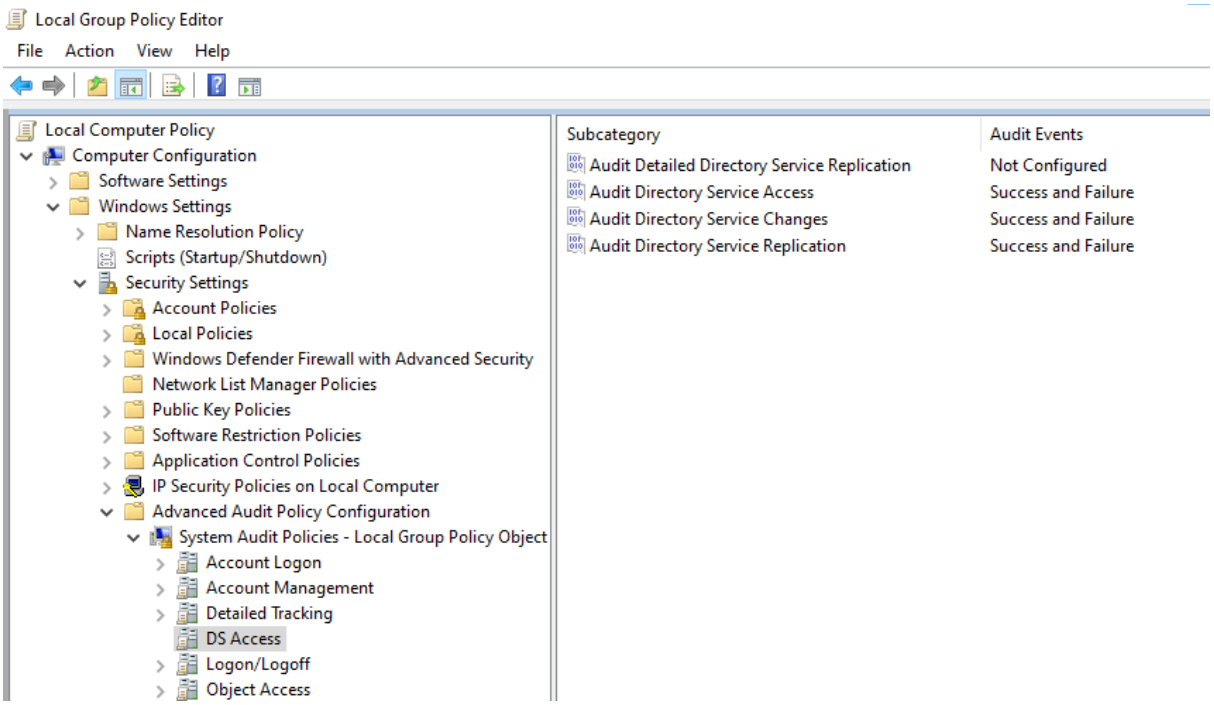


Рисунок 2.5 — модифікації локальних групових політик для аудиту подій Active directory service Access

Після цього в журналі Security з'являться події:

- **4662** — успішний або невдалий доступ до об'єкта каталогу

Ці події дозволяють відстежувати зміни в облікових записах, групах та інших об'єктах каталогу.

Відповідні техніки MITRE ATT&CK:

- T1098 — Маніпуляція обліковими записами
- T1484.002 — DCShadow

Інструменти, які можуть бути виявлені:

- Mimikatz
- DCShadow

2.3 Побудова рішення для централізованого збору та обробки подій на базі Elastic Stack

Розширене логування на контролерах домену забезпечує необхідний обсяг даних для аналізу активності в середовищі Active Directory. Однак наявність журналів подій на окремих серверах не гарантує їх ефективного використання для виявлення атак. Гіпотетична ситуація, при якій було здійснено комплексну атаку з повною компрометацією доменного середовища, що дає можливіть зловмиснику зачистити маркери своєї присутності в мережі шляхом видалення журналів подій не залишає шансів не тільки на з’язування першопричин компрометацій, а й вилучення присутності зловмисника з перметру корпоративної мережі. Для цього потрібне централізоване рішення, яке дозволяє збирати, зберігати, обробляти та аналізувати великі обсяги даних в безпечному місці. Оптимальним вибором для таких задач є платформа **Elastic Stack**.

2.3.1 Компоненти рішення

Elastic Stack (раніше відомий як EFK Stack) — це комплекс програмних рішень, до складу якого входять:

- Elastic Agent — універсальний агент для збору та відправки подій із серверів та робочих станцій. Замінює Winlogbeat і підтримує інтеграцію з Fleet для централізованого управління.

- Fleet + Kibana — модуль централізованого управління агентами. Дозволяє розгортати політики збору даних, контролювати стан агентів і швидко налаштовувати інтеграції через веб-інтерфейс.
- Elasticsearch — високо масштабована база для зберігання, обробки та аналізу подій.
- Kibana — веб-інтерфейс для візуалізації, пошуку, створення дашбордів і алертів.

Ця архітектура дозволяє створити єдину точку збору подій з усієї інфраструктури для подальшого аналізу та виявлення аномалій. Схема рішення з ключовими компонентами та повним шляхом логу від джерела до зберігання та обробки в ElasticSearch відображено на рисунку 2.6

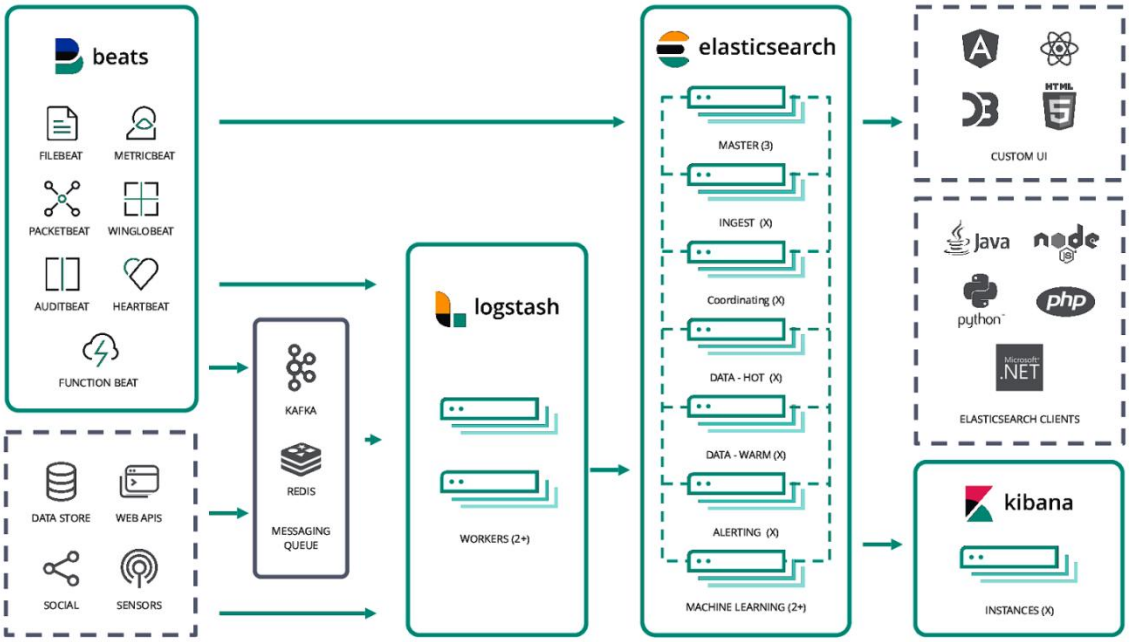


Рисунок 2.6 — схема компонентів рішення обробки подій на базі Elastic Stack

2.3.2 Централізований збір подій

Elastic Agent відіграє ключову роль у реалізації централізованого підходу до збору подій з усіх контролерів домену Active Directory. Його

можливості забезпечують ефективний моніторинг безпекових подій, гнучку конфігурацію, мінімізацію зайвого трафіку та надійність передачі даних.

Основні технічні аспекти включають:

- Інтеграція Windows Logs. У Fleet налаштовуються інтеграції, які охоплюють усі ключові журнали: Security, Directory Services, DNS, Kerberos, LDAP.
- Оптимізація обсягу даних. Агент дозволяє фільтрувати події за ідентифікаторами (Event ID), зберігаючи лише ті, що мають значення для безпеки (наприклад, 4624, 4625, 4768, 4769, 4662). Така фільтрація значно зменшує навантаження на мережу, об’єм індексації в Elasticsearch і кількість збережених даних, не втрачаючи при цьому критичної інформації, що також має гарний економічний аспект, оскільки вартість активу, що захищається не повинна перевищувати вартість засобів захисту.
- Захищена передача даних. Використання TLS шифрування забезпечує конфіденційність і цілісність даних під час їх транспортування в Elasticsearch. Усі з'єднання агента з сервером керування (Fleet Server) та кінцевим сховищем даних за замовчуванням встановлюються з використанням сертифікатів X.509, що запобігає перехопленню або підробці даних у мережі. Також доступна автентифікація агентів для гарантування, що лише авторизовані агенти можуть надсилати події.
- Додаткові модулі. Elastic Agent — це **модульний агент**, який підтримує встановлення декількох типів інтеграцій одночасно. Тому, він підтримує збір не лише логів, а й мережевих подій, системної інформації, а також інтеграцію з іншими системами (наприклад, EDR-рішеннями).

2.3.3 Попередня обробка та нормалізація даних

Після доставки подій з агентів, відбувається багатоступенева обробка, яка забезпечує структуризацію, нормалізацію, збагачення і підготовку даних

до подальшого аналізу. Це — критичний етап, що впливає на якість аналітики, швидкість пошуку та можливості кореляції.

Перелік стадій обробки:

- Ingest Pipeline. Автоматично перетворює поля, додає мітки та уніфікує структуру записів. Цей процес виконується **в режимі реального часу** під час надходження кожного документа й гарантує, що дані зберігаються вже у готовому до аналізу вигляді.
- Політика збереження даних. Дані організовуються у відповідних індексах, що дозволяє зберігати їх у різних режимах (гаряче, холодне, архівне сховище) в залежності від актуальності. Керування цими рівнями відбувається через ILM (Index Lifecycle Management) — політики, які автоматично переміщують індекси між рівнями зберігання на основі часу, обсягу чи індивідуальних умов. Це дозволяє масштабувати систему логування без втрати контролю над витратами.
- Виділення атрибутів. Користувачі, комп'ютери, IP-адреси, порти — усе це виноситься в окремі поля для полегшення аналітики. Нормалізація через схему Elastic Common Schema (ECS) дозволяє всі події, незалежно від джерела, подавати в уніфікованій формі. Це забезпечує можливість виконувати складні аналітичні запити або створювати кореляційні правила без прив’язки до конкретного формату логів.
- Теги безпеки та MITRE ATT&CK. Події автоматично маркуються відповідно до тактик і технік MITRE ATT&CK для подальшої побудови дашбордів і кореляційних правил.

2.3.4 Візуалізація та початковий аналіз

Kibana — це інтерактивна веб-консоль, яка забезпечує повноцінну взаємодію з даними в Elasticsearch та управління сервером Fleet. У контексті кібербезпеки та моніторингу домену Active Directory, Kibana виконує функції аналітичного інтерфейсу, платформи візуалізації, а також системи керування

правилами виявлення загроз. Завдяки цьому аналітики інформаційної безпеки можуть швидко оцінювати ситуацію, розслідувати інциденти та створювати власні сценарії виявлення.

Завдяки Kibana вдається ефективно працювати з отриманими даними:

- Дашборди для моніторингу. Відображають активність у домені в режимі реального часу (успішні та неуспішні входи, запити Kerberos, зміни об'єктів AD). Всі ці елементи оновлюються в реальному часі або з налаштованою періодичністю. Адміністратор або аналітик може **на одному екрані** відстежувати загальний стан безпеки домену, включаючи аномалії в активності користувачів чи хостів.
- Інтерактивний аналіз. Kibana надає інструменти **глибинного аналізу подій** (Data Discovery, Lens, TSVB), які дозволяють швидко звузити область дослідження за допомогою фільтрів, пошуку та агрегацій(наприклад, фільтрація за користувачами, комп'ютерами, IP-адресами), що допомагає оперативно виявляти підозрілу активність. Завдяки повній підтримці **Elastic** Common Schema (ECS)**,** події з різних джерел представлені у стандартизованому вигляді, що спрощує побудову запитів і порівняння даних між індексами.
- Алерти на основі правил. Kibana містить потужний двигун створення оповіщень (Detection Rules)**,** який дозволяє налаштовувати спрацювання на основі подій або результатів ML-аналізу. Аналітики можуть створювати як прості умови, так і складні кореляційні сценарії для налаштування сповіщень при ознаках атак, таких як: Brute Force, Kerberoasting, DCSync. Оповіщення можна налаштувати з автоматичним надсиланням повідомлень через Slack, Email, Webhook або навіть створенням нових записів у Jira чи іншій системі.

Kibana є не просто засобом візуалізації, а центральною платформою для моніторингу, розслідувань та виявлення загроз у доменах Active Directory.

Вона дозволяє:

- бачити ключову активність у реальному часі,

- ефективно звужувати фокус аналізу до конкретного користувача чи машини,
- оперативно отримувати сповіщення про підозрілу або злочинну активність.

У поєднанні з Elastic Security та MITRE ATT&CK, Kibana формує повноцінне робоче середовище для сучасного аналітика SOC.

2.3.5 Створення правил і інтеграція з MITRE ATT&CK

В Elastic Security (модуль безпеки Kibana) реалізовано програму для виявлення загроз (Detection Engine), який дозволяє створювати правила для автоматичного пошуку ознак атак в даних. Правило виявлення визначає умови, за яких на основі подій у Elasticsearch генерується оповіщення (alert).

Спочатку обирається тип правила – наприклад, звичайний запит, кореляція подій, порогове значення, співставлення індикаторів або правило машинного навчання. Від типу залежить формат умов: це може бути запит KQL/Lucene для фільтрації подій, мова EQL/ES|QL для послідовностей подій, налаштування порога кількості подій, результати аномалій ML тощо. Далі користувач задає, які індекси або джерела даних скануватиме правило, та формулюєте умови спрацювання – наприклад, KQL-фільтр по значеннях полів чи EQL-сценарій послідовності подій. Правила можна створювати на основі збережених пошукових запитів Kibana або історії розслідувань (Timeline), що спрощує побудову складних умов.

Однією з переваг Elastic Stack є можливість створення правил виявлення, які корелюють події з техніками MITRE ATT&CK:

- Підвищена кількість TGS запитів T1558.003 Kerberoasting.
- Аномальна DCSync активність (4662 + події реплікації) → T1003.006 DCSync.
- Масова реєстрація облікових записів або зміни в GPO → T1098 Account Manipulation, T1484.001 Group Policy Modification.

Ці правила дозволяють знизити навантаження на аналітиків і автоматизувати виявлення стандартних атак. Приклад таких запитів та описів наведено на рисунку 2.7.

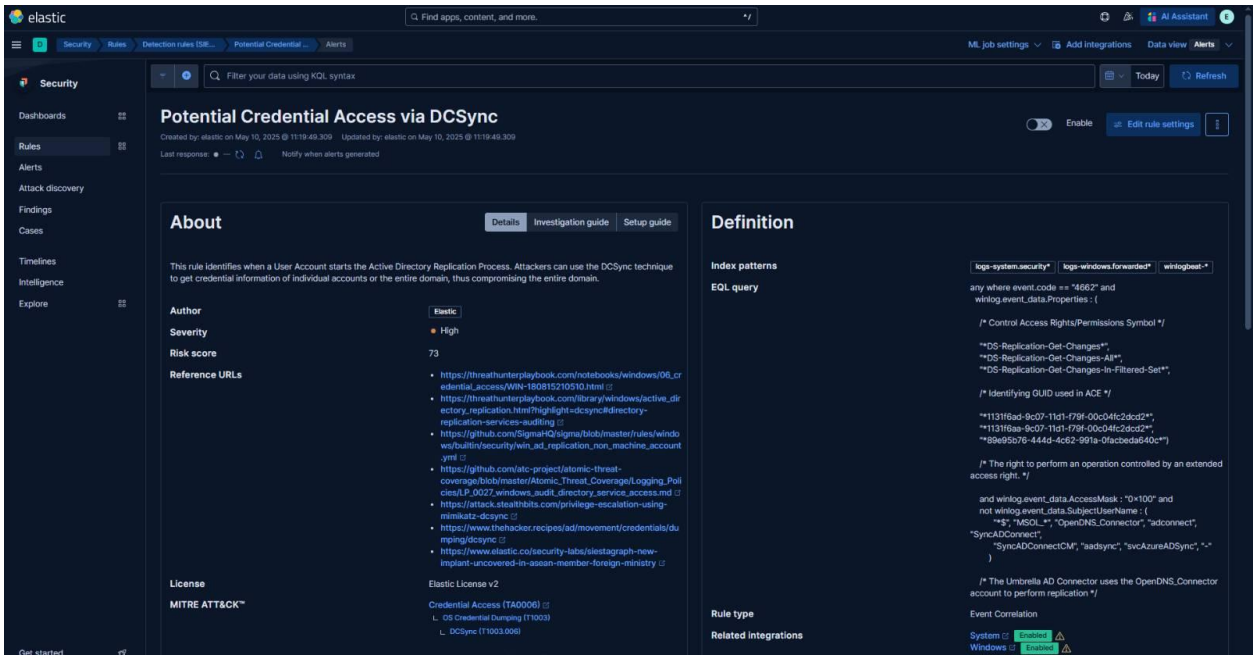


Рисунок 2.7 Вигляд правила для детектування аномальної DCSync активності

2.3.6 Роль централізованого збору в контексті поведінкового аналізу

Централізований збір і зберігання подій дозволяє формувати цілісне уявлення про активність користувачів, систем та сервісів у масштабі всієї інфраструктури, що критично важливо для виявлення складних та прихованих атак, автоматизованого аналізу відхилень та адаптації захисту під специфіку організації. А також, у межах Elastic Stack є необхідною передумовою для впровадження ефективного поведінкового аналізу (User and Entity Behavior Analytics, UEBA):

- Повнота даних. У фрагментованому підході дані знаходяться на окремих серверах, файлах і пристроях, що робить цілісний аналіз майже неможливим або вкрай трудомістким. Натомість централізований репозиторій логів агрегує дані з різних джерел і зберігає їх в одному місці. Це означає, що аналітики можуть легко виконувати пошук та кореляцію по всій

інфраструктурі одразу, отримуючи розуміння повної історії подій без потреби постійного переміщення між хостами для збору релевантної інформації.

- Зниження витрат людських ресурсів**.** Автоматичне збирання логів з усіх систем до спільного сховища прибирає потребу в ручних, рутинних діях, пов’язаних із отриманням даних. У традиційних підходах адміністратори витрачають багато часу на збирання файлів логів з різних серверів або написання скриптів для їх агрегації, що також схильне до помилок. Зазвичай такий підхід не дає змоги виявити атаку на ранніх стадіях для її швидкої нейтралізації. Elastic Stack значною мірою автоматизує цей процес**,** тож команди можуть зосередитися на аналізі, а не на підготовці даних. До того ж, одна централізована платформа легше підтримується і масштабується (не треба оновлювати кілька різних систем логування).
- Єдине місце для впровадження безпекових політик і контролів**.** При централізованому зборі легше впроваджувати універсальні політики зберігання даних, контролю доступу до логів та моніторингу відповідності. Усі події можна однаково захистити, резервувати, шифрувати тощо, застосувавши налаштування на рівні Elastic Stack. У фрагментованому підході кожне сховище логів могло б вимагати окремих налаштувань і залишати прогалини в безпеці. Таким чином, централізація не тільки сприяє ефективності аналізу, а й підвищує загальну керованість та безпечність інфраструктури журналювання.
- Стандартизація та узгодженість даних. При використанні різних інструментів для збору логів часто відсутня єдина схема або формат даних, що ускладнює зіставлення інформації. Централізований підхід (особливо з ECS) гарантує нормалізацію і стандартизацію логів під час збору. Всі події приводяться до спільного вигляду, тож аналітики працюють з узгодженими полями і поняттями. Це значно полегшує виявлення патернів поведінки по всій інфраструктурі – наприклад, визначити, що певна IP-адреса фігурує і в мережевих логах, і в системних, або що користувач здійснює дії на кількох серверах.

- Швидке виявлення і реагування. Час на виявлення та розслідування інциденту скорочується. Аналітик може майже миттєво зібрати потрібні факти з різних джерел через запити в Elasticsearch або перегляд в Kibana, тоді як у розподіленому середовищі потрібно було б опитувати команди різних систем або збирати файли логів вручну. Консолідований аналіз логів у режимі реального часу дозволяє оперативно помітити підозрілу активність або атаку і так само швидко відреагувати на неї. Це особливо важливо для запобігання розповсюдження атак. Наприклад, централізований моніторинг одразу покаже, що один і той самий користувач намагається увійти на десять різних серверів (можлива горизонтальна атака), і дасть можливість негайно блокувати такого користувача.
- Виявлення складних атак. На відміну від простих сигнатурних подій, повільні або розподілені атаки, такі як lateral movement, persistence або privilege escalation, не завжди проявляються як очевидні інциденти. Натомість вони складаються з серії дій, розосереджених у часі, просторі та між різними системами. Завдяки централізації логів можливо відслідковувати, як зловмисник переміщується від однієї машини до іншої, використовуючи вкрадені облікові дані, створення нового користувача або додавання себе до адміністративної групи.
- Навчання моделей машинного навчання. Моделі поведінкового аналізу потребують великої кількості якісних, репрезентативних та узгоджених даних, які можна отримати лише з централізованого джерела. Elastic Machine Learning дозволяє запускати такі моделі у реальному часі, а оцінка аномалії використовується для генерації оповіщень, ранжування ризиків або запуску подальших правил кореляції. Без централізації даних неможливо охопити повний контекст і побудувати надійні поведінкові моделі — вони будуть фрагментарними і неточними.
- Адаптація під середовище. Кожна організація має свої унікальні особливості: структура домену, години роботи, характер бізнес-процесів, географія, рівень автоматизації тощо. Централізована система дозволяє гнучко

налаштовувати правила та алгоритми виявлення відповідно до специфіки конкретної організації. Elastic дозволяє створювати кастомні правила, аналітичні дашборди, машини навчання та пайплайни обробки — і все це працює ефективно лише за умови повного охоплення подій через централізований збір.

Висновки за розділом 2

У другому розділі було детально проаналізовано особливості виявлення аномальної активності у середовищі Active Directory, що ґрунтується на використанні легітимних протоколів і служб (Kerberos, LDAP, RPC, SMB).

Було встановлено, що більшість сучасних атак на AD не передбачають доставку шкідливого програмного забезпечення, а натомість використовують вбудовані механізми самого середовища Active Directory. Це ускладнює їх виявлення за допомогою традиційних засобів захисту (SIEM, IDS/IPS), які орієнтуються на сигнатури або відомі IOC.

Ключовою умовою для ефективного виявлення таких загроз є розширене логування на доменних контролерах. У розділі було розглянуто налаштування та призначення подій, пов’язаних з LDAP-запитами, автентифікацією Kerberos та доступом до об’єктів каталогу. Визначено, які техніки MITRE ATT&CK можуть бути ідентифіковані за допомогою цих подій, а також які інструменти атак (наприклад, BloodHound, Mimikatz, Rubeus) можуть бути виявлені на основі зібраної інформації.

Окрему увагу приділено побудові рішення для централізованого збору та обробки подій на базі платформи Elastic Stack. Було охарактеризовано роль Elastic Agent у зборі подій, принципи нормалізації та маркування даних, створення правил виявлення, а також можливості інтеграції з MITRE ATT&CK. Обґрунтовано, що централізоване логування є необхідною умовою для подальшого поведінкового аналізу та побудови моделей машинного навчання, здатних виявляти приховані кіберзагрози в реальному часі.

РОЗДІЛ 3
РОЗГОРТАННЯ ЛАБОРАТОРНОГО СЕРЕДОВИЩА

3.1 Вибір платформи віртуалізації Proxmox та середовища GOAD

В рамках дослідження загроз інформаційній безпеці в середовищі Active Directory важливою складовою є наявність контрольованого та реалістичного середовища для моделювання атак, тестування засобів захисту та навчання.

Лабораторний стенд GOAD (Game of Active Directory) є одним із найефективніших інструментів для досягнення цих цілей, оскільки забезпечує змодельовану корпоративну інфраструктуру з кількома доменами, контролерами, робочими станціями та типовими конфігураціями політик.

Цей розділ присвячений опису процесу розгортання та налаштування GOAD у віртуалізованому середовищі Proxmox. Наведено архітектуру тестової мережі, розглянуто основні компоненти середовища, вимоги до ресурсів, порядок інсталяції, а також обґрунтовано вибір GOAD як платформи для практичного дослідження атак на Active Directory.

Лабораторне середовище відіграє ключову роль у забезпеченні достовірності результатів моделювання атак та перевірки ефективності засобів виявлення — зокрема, на основі можливостей машинного навчання в Elasticsearch. Створення такого середовища є необхідним етапом для подальшого аналізу поведінкових патернів, побудови моделей загроз і тестування інструментів проактивного виявлення кіберінцидентів.

Для практичної реалізації моделі проактивного виявлення кіберзагроз було обрано лабораторне середовище Game of Active Directory (GOAD) версії 3 та платформу віртуалізації Proxmox Virtual Environment (VE). GOAD є спеціалізованою навчальною Active Directory-лабораторією, розробленою для проведення тестувань на проникнення та відпрацювання типових атак на

інфраструктуру Windows-домена. У цій лабораторії розгорнуті декілька вразливих доменів Windows з заздалегідь заданими взаємними відносинами довіри, що дозволяє відтворити різні сценарії атак (ескалація привілеїв, переміщення по мережі тощо) в контрольованих умовах. Також дана лабораторія можи бути використана для навчання аналітиків організації, виявлення слабких місуь детектування та полем для впровадження та тестування нових технологій без нанесення шкоди існуючим процесам.

В якості платформи для розгортання GOAD обрано Proxmox VE – повнофункціональне середовище серверної віртуалізації. Proxmox підтримує гіпервізор KVM для створення повноцінних віртуальних машин та LXC-контейнерів, забезпечуючи централізоване керування через веб-інтерфейс.

Перевагами Proxmox є відсутність ліцензійних витрат, гнучкість у налаштуванні мережі (підтримка бриджів, VLAN, програмних маршрутизаторів) та наявність API для автоматизації розгортання та налаштування віртуальних машин.

3.2 Автоматизація розгортання засобами Packer, Terraform та Ansible

Щоб розгорнути складне середовище GOAD узгоджено та з мінімальними ручними налаштуваннями, використано підхід «інфраструктура як код» (Infrastructure as Code). Зокрема, застосовано зв’язку інструментів Packer, Terraform та Ansible для автоматизації підготовки шаблонів віртуальних машин, їхнього розгортання та конфігурації.

Packer. Packer використовується на етапі підготовки базових образів операційних систем. З його допомогою з офіційних ISO-образів Windows Server 2019 та Windows Server 2016 були автоматично створені шаблони VM у Proxmox. Packer-сценарій виконує безконтрольну (unattended) інсталяцію ОС, налаштовує початкові параметри (встановлює потрібні драйвери virtio, параметри мережі, облікові записи). Отримані шаблони слугують основою для

подальшого розгортання багатьох однотипних вузлів (наприклад, декількох контролерів домену) без повторної інсталяції вручну, що прискорює процес і уніфікує конфігурацію систем.

Terraform. На наступному етапі засобами Terraform описано інфраструктуру лабораторії і виконано її розгортання на Proxmox VE.

Застосовано офіційний провайдер Terraform для Proxmox, який через API створює потрібні віртуальні машини із заздалегідь підготовлених Packer-шаблонів. У конфігурації Terraform визначено параметри кожної VM (назва, диск, кількість vCPU, обсяг RAM, мережеві інтерфейси та VLAN). Terraform-план включає створення всіх компонентів середовища GOAD. Після ініціалізації (terraform init) і застосування конфігурації (terraform apply) Terraform автоматично розгортає задану кількість віртуальних машин на Proxmox та під’єднує їх до відповідних віртуальних мереж. Структуру конфігураційного файлу terraform для розгортання windows хостів наведено на рисунку 3.1

```
}root@GOAD-provisoning:~/GOAD/workspace/9bfe31-goad-proxmox/provider# cat windows.tf
variable "vm_config" {
  type = map(object({
    name               = string
    desc               = string
    cores              = number
    memory             = number
    clone              = string
    dns                = string
    ip                 = string
    gateway            = string
  }))

  default = {
    "dc01" = {
  name               = "DC01"
  desc               = "DC01 - windows server 2019 - 192.168.10.10"
  cores              = 2
  memory             = 3096
  clone              = "WinServer2019_x64"
  dns                = "192.168.10.1"
  ip                 = "192.168.10.10/24"
  gateway            = "192.168.10.1"
}
"dc02" = {
  name               = "DC02"
  desc               = "DC02 - windows server 2019 - 192.168.10.11"
  cores              = 2
  memory             = 3096
  clone              = "WinServer2019_x64"
  dns                = "192.168.10.1"
  ip                 = "192.168.10.11/24"
  gateway            = "192.168.10.1"
}
"dc03" = {
  name               = "DC03"
  desc               = "DC03 - windows server 2016 - 192.168.10.12"
  cores              = 2
  memory             = 3096
  clone              = "WinServer2016_x64"
  dns                = "192.168.10.1"
  ip                 = "192.168.10.12/24"
  gateway            = "192.168.10.1"
}
"srv02" = {
  name               = "SRV02"
  desc               = "SRV02 - windows server 2019 - 192.168.10.22"
  cores              = 2
  memory             = 6240
  clone              = "WinServer2019_x64"
  dns                = "192.168.10.1"
  ip                 = "192.168.10.22/24"
  gateway            = "192.168.10.1"
}
"srv03" = {
  name               = "SRV03"
  desc               = "SRV03 - windows server 2016 - 192.168.10.23"
  cores              = 2
  memory             = 5120
  clone              = "WinServer2016_x64"
```

Рисунок 3.1 – конфігураційний файл terraform для розгортання VM лабораторного середовища

Ansible. Завершальним кроком є конфігурація програмного середовища кожної розгорнутої машини за допомогою Ansible. Проект GOAD містить набір Ansible-плейбуків, що відповідають за розгортання ролей Active Directory, налаштування доменів та відносин довіри, створення облікових записів користувачів, встановлення необхідного ПЗ (наприклад, роль DHCP, IIS, SQL Server на окремих вузлах) та інше. Ansible виконує ці плейбуки на щойно створених VM. Linux-хости налаштовуються через протокол SSH, Windows-хости – через протокол WinRM. Наприклад, одним з плейбуків розгортається контролер домену DC01 на Windows Server 2019 і створюється новий домен, іншим – приєднується додатковий контролер до існуючого домену або встановлюється взаємна довіра між двома доменами різних лісів.

Таким чином, після завершення виконання Ansible плейбуків маємо повністю сконфігуроване програмне середовище лабораторії згідно зі сценарієм GOAD.

Загальний процес автоматизованого розгортання здійснюється в наступній послідовності: на provisioning host клоновано репозиторій GOAD та виконано сценарій goad.sh, що послідовно запускає Packer, Terraform і Ansible з потрібними параметрами. Такий підхід значно знижує ймовірність винекнення помилок та забезпечує повторюваність результату – всю лабораторію можна розгорнути через запуск одного скрипта або легко перевстановити за необхідності.

3.3 Мережева інфраструктура: роль pfSense, VLAN, NAT та портфорвардинг

Для ізоляції лабораторної мережі від основної мережі та гнучкого управління мережевим доступом використано віртуальний мережевий екран

pfSense. pfSense є дистрибутивом на базі FreeBSD, що надає функціонал маршрутизатора та міжмережевого екрану. У середовищі Proxmox pfSense розгорнуто як окрема віртуальна машина, підключена з одного боку до зовнішньої мережі (через інтерфейс WAN, міст до фізичного інтерфейсу Proxmox), а з іншого – до внутрішніх віртуальних мереж лабораторії (LAN-інтерфейси).

Всередині pfSense налаштовано дві віртуальні локальні мережі (VLAN) для сегментації трафіку різних доменів GOAD. Зокрема, виділено VLAN10 (192.168.10.0/24) для основного лабораторного середовища та VLAN20 (192.168.20.0/24) як допоміжна мережа.

Інтерфейси pfSense отримали адреси 192.168.10.1 та 192.168.20.1 відповідно і виступають шлюзами за замовчуванням для вузлів цих підмереж.

Через веб-інтерфейс pfSense увімкнено DHCP-сервери на обох VLAN для автоматичного призначення IP адрес віртуальним машинами лабораторії (за потреби IP окремих критичних серверів закріплено статично, оскільки доменні контролери або сервери на яких розгорнути бази даних не можуть мати змінних IP адрес). Налаштування привил LAN наведено на рисунку 3.2.

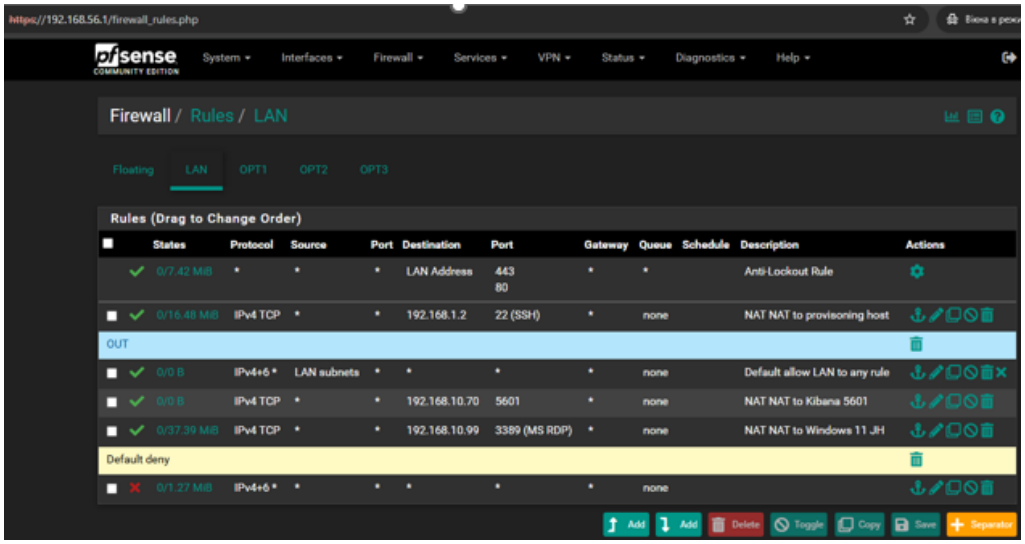


Рисунок 3.2 – Налаштування LAN в PfSense

На pfSense також реалізовано політики NAT та port forwarding, щоб забезпечити вибірковий доступ назовні та ззовні. Вихідний трафік з

лабораторії до Інтернету відбувається через (NAT) – це необхідно для оновлення ОС, завантаження необхідного ПЗ під час автоматичного розгортання (наприклад, скачування пакетів через Ansible) та емуляції доступу до зовнішніх ресурсів (наприклад, імітація користувача, що виходить у Інтернет). Вхідний трафік з зовнішньої мережі до лабораторії за замовчуванням заблокований для безпеки, відкрито лише окремі порти для віддаленого доступу, налаштовано перенаправлення порту 3389/TCP (RDP) з адреси pfSense (WAN) на внутрішню адресу jump-хоста Windows 11. Це дозволяє підключатися до jump-host з зовнішньої мережі для адміністрування або імітації атаки, не піддаючи ризику інші вузли.

Аналогічно, при потребі може бути відкрито доступ до веб-інтерфейсу Kibana (порт 5601) на сервері моніторингу або SSH (22) на Kali Linux. Налаштування NAT наведено на рисунку 3.3

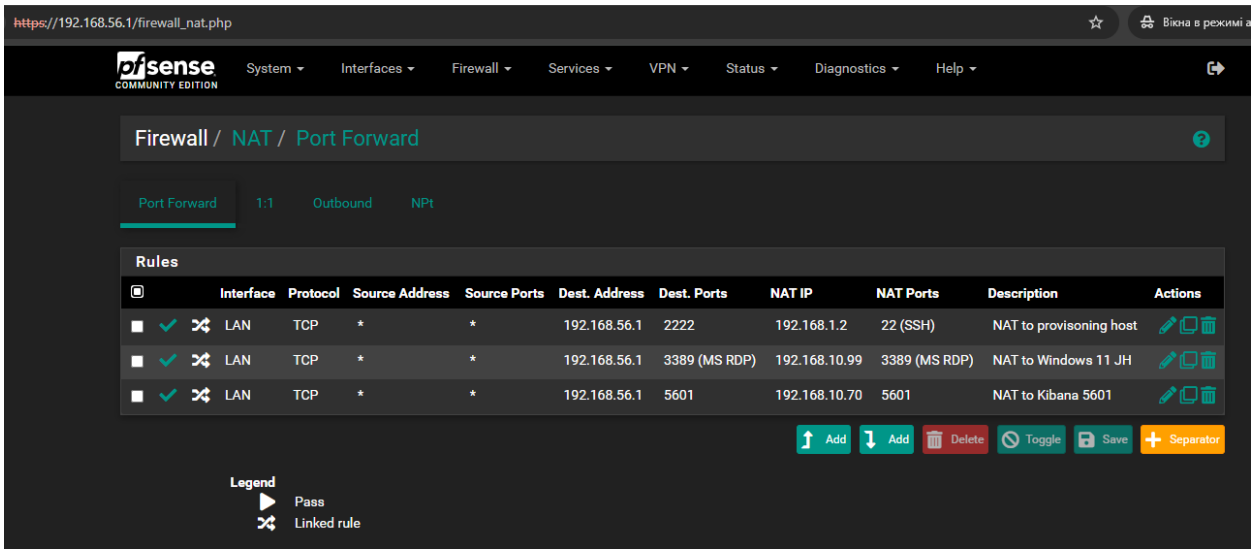


Рисунок 3.3 – Налаштування NAT в PfSense

Таким чином, pfSense виступає центральним комутаційним вузлом. Він ізолює середовище GOAD від корпоративної мережі, забезпечуючи контрольований доступ лише до потрібних сервісів, і маршрутизує трафік між VLAN10 та VLAN20 всередині лабораторії.

3.4 Склад віртуальних машин лабораторії

Розгорнуте середовище GOAD v3 складається з декількох груп вузлів, кожен з яких виконує визначену роль. Інфраструктура налічує п’ять основних Windows-серверів, що формують два Active Directory-ліси (три домени), а також ряд допоміжних вузлів для управління, доступу та моніторингу.

Перелік компонентів лабораторії:

- Provisioning Host**.** Хост підготовки розгортання – це окремий вузол (Linux-контейнер з ОС Ubuntu 22.04), з якого здійснюється автоматизована інсталяція всієї лабораторії. На ньому встановлено Packer, Terraform, Ansible та всі суміжні утиліти. Provisioning host підключено до обох VLAN лабораторії VLAN10 (основного сегмента) для можливості виконання Ansible-скриптів на всі внутрішні машини. Цьому контейнеру виділено 2 vCPU та 2048 МБ RAM.
- **PfSense.** Віртуальний маршрутизатор pfSense, описаний вище, забезпечує мережеву інфраструктуру. Він розгорнутий як VM з 2 vCPU, 4,096 МБ RAM і двома мережевими інтерфейсами (WAN та trunk для VLAN10/20). PfSense працює під управлінням FreeBSD і виконує функції DHCP-сервера, маршрутизатора та ME для сегментів лабораторії.
- Доменні сервери GOAD. Основу середовища складають п’ять Windows-серверів, розгорнутих сценарієм GOAD. Три з них виконують роль контролерів домену (DC), ще два – роль серверів-членів домену з встановленими сервісами для експлуатації (SQL Server, веб-сервер IIS, файлові сервіси WebDav, Центр сертифікації тощо). Відповідно до документації GOAD, розгорнуто один ліс SevenKingdoms із двома доменами (sevenkingdoms.local та дочірнім north.sevenkingdoms.local) та окремий ліс Essos (essos.local). Контролери доменів DC01 (kingslanding) – Windows Server 2019 у домені sevenkingdoms.local, DC02 (winterfell) – Windows Server 2019 у домені north.sevenkingdoms.local, DC03 **(meereen)** – Windows Server 2016 у домені essos.local. Два додаткові сервери: SRV02 (castelblack) на Windows Server 2019, приєднаний до домену north.sevenkingdoms.local, та **SRV03**

(braavos) на Windows Server 2016 у домені essos.local. На цих серверах Ansible розгортає рольове ПЗ (наприклад, SRV02 виконує роль файлового сервера та MSSQL для сценаріїв атак, SRV03 – сервер баз даних в іншому лісі). Кожна з цих VM отримала 4 vCPU та 8 ГБ оперативної пам’яті (контролерам домену виділено більше ресурсів для коректної роботи служб AD). Всі вони приєднані до VLAN10.

- Jump-host (Windows 11). Для доступу до внутрішньої мережі розгорнуто окрему клієнтську систему Windows 11, умовно названу jump-host. Цей вузол виконує роль допоміжної машини, через яку дослідник чи потенційний зловмисник може потрапити в ізольовану лабораторію. Jump-host підключено до VLAN10. На ньому встановлено інструменти адміністрування Active Directory та інше ПЗ, потрібне для аналізу загроз (наприклад, Sysinternals, RSAT). Через відповідне правило port forwarding (RDP) на pfSense до jump-host можна підключитися ззовні засобами віддаленого робочого столу, після чого вже з нього – отримати доступ до інших внутрішніх. Йому виділено 4 vCPU та 8 ГБ RAM.
- Kali Linux (Attack Box). Для моделювання дій зловмисника всередині мережі лабораторії додано віртуальну машину з дистрибутивом Kali Linux. Kali підключено до VLAN10; таким чином, вона знаходиться всередині середовища і може безпосередньо взаємодіяти з усіма вузлами першого сегменту та (через pfSense) з другим сегментом. На Kali встановлено типові інструменти тестування на проникнення (Nmap, Responder, Mimikatz, BloodHound тощо). Даний вузол може грати роль “машини нападника” у сценаріях, коли припускається, що зловмисник вже отримав початковий доступ до внутрішньої мережі (наприклад, шляхом компрометації jump-host чи іншого вузла) і продовжує розвиток атаки зсередини. Для Kali виділено 4 vCPU та 8 ГБ оперативної пам’яті.
- EFK Stack (Debian). Окремим компонентом інфраструктури є сервер моніторингу на базі ОС Debian, на якому розгорнуто стек EFK (Elasticsearch + Filebeat + Kibana). Цей вузол призначений для збору та

централізованого аналізу журналів подій зі всіх систем лабораторії. Зокрема, в рамках дипломної роботи планується збір Windows-журналів (Security, Sysmon тощо) з контролерів домену та інших машин, їхнє накопичення в Elasticsearch та візуалізація через Kibana для подальшого дослідження. Kibana на сервері EFK це веб-інтерфейс для побудови дашбордів і пошуку аномалій у зібраних журналах. Сервер EFK підключено до VLAN10 (IP 192.168.10.70) – цього достатньо, щоб отримувати логи з обох доменних сегментів. Ресурси для EFK виділено з певним запасом, зважаючи на вимогливість Elasticsearch: 4 vCPU, 8192 МБ RAM, 80 ГБ диску. Статус відповідних служб складових стеку через команду systemctl status відображено на рисунку 3.4

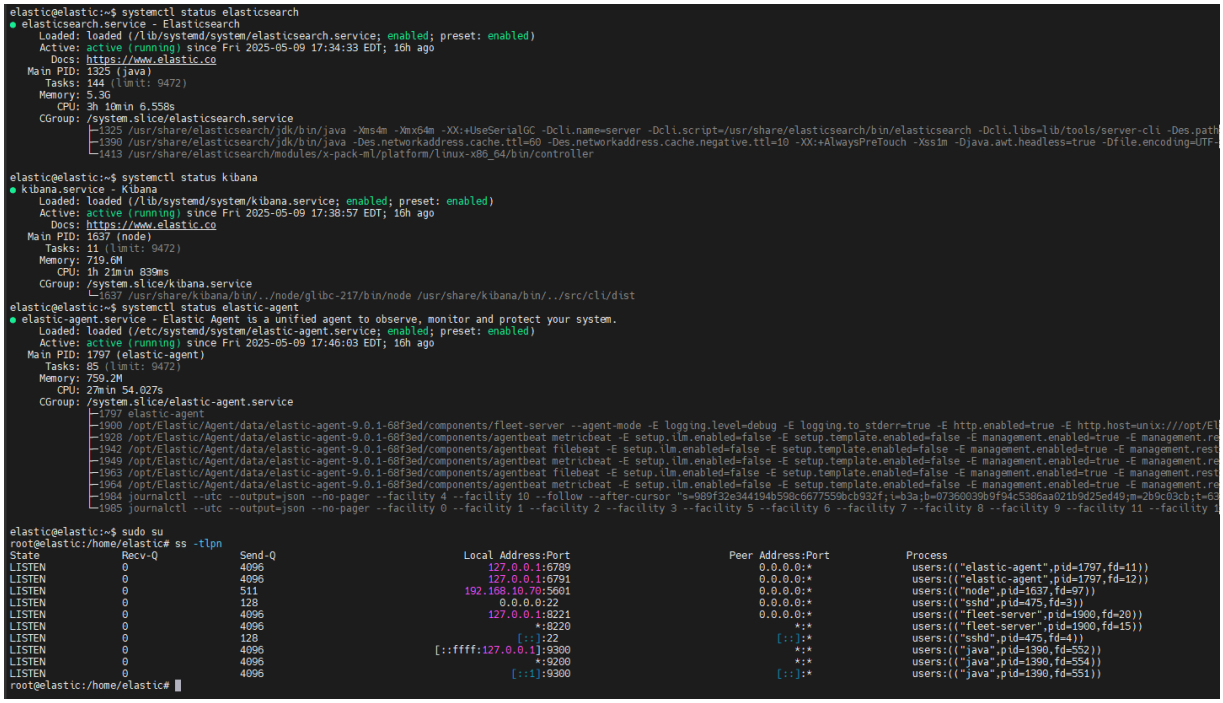


Рисунок 3.4 – Статус служб компонентів стеку EFK

Таким чином, загальна структура створеної лабораторної мережі включає 10 віртуальних вузлів: 1 provisioning-контейнер, 1 маршрутизатор pfSense, 5 Windows-серверів доменів (GOAD), 1 Windows 11 клієнт, 1 Kali Linux та 1 сервер моніторингу на Debian. На рисунку 3.5 показано основні компоненти середовища та їх підключення.

Resource Pool: GOAD

Summary | Members | Permissions

Add | Remove

Type	Description	Disk usage...	Memory us...	CPU usage	Uptime	Host CPU usage
lxc	103 (GOAD-provisoning)	6.3 %	2.4 %	0.0% of 2 CPUs	1 day 17:12:52	0.0% of 1 CPU
qemu	102 (GOAD-pfsense)	0.0 %	85.4 %	7.4% of 2 CPUs	1 day 03:18:34	14.8% of 1 CPU
qemu	104 (WinServer2016x64-cloudinit-qcow2)				-	
qemu	105 (WinServer2019x64-cloudinit-qcow2)				-	
qemu	106 (GOAD-SRV02)	0.0 %	83.2 %	11.0% of 4 CPUs	1 day 01:01:54	43.9% of 1 CPU
qemu	107 (GOAD-DC01)	0.0 %	79.6 %	9.7% of 4 CPUs	1 day 01:01:56	39.0% of 1 CPU
qemu	108 (GOAD-DC02)	0.0 %	84.5 %	13.6% of 4 CPUs	1 day 01:12:44	54.4% of 1 CPU
qemu	109 (GOAD-DC03)	0.0 %	87.9 %	5.3% of 4 CPUs	1 day 01:01:51	21.1% of 1 CPU
qemu	111 (GOAD-SRV03)	0.0 %	81.8 %	4.5% of 4 CPUs	1 day 01:01:49	18.1% of 1 CPU
qemu	115 (GOAD-Windows-JH)	0.0 %	91.2 %	5.8% of 4 CPUs	14:46:58	23.0% of 1 CPU
qemu	122 (GOAD-Kali)	0.0 %	12.4 %	0.2% of 4 CPUs	17:33:29	0.7% of 1 CPU
qemu	127 (Debian-12-80G-Template)				-	
qemu	128 (GOAD-ELK)	0.0 %	93.5 %	15.0% of 4 CPUs	16:56:53	59.8% of 1 CPU

Рисунок 3.5 – віртуальні машини лабораторного середовища

3.5 Розгортання середовища та перевірка конфігурації

Після підготовки усіх компонентів (Proxmox, мережі pfSense, шаблони VM) здійснено безпосередній запуск розгортання лабораторії GOAD v3. На provisioning-хості виконано скрипт goad.sh з параметрами, що вказують на розгортання повної версії лабораторії у провайдера Proxmox (-p proxmox). Даний сценарій спочатку перевірив наявність усіх необхідних залежностей та прав доступу, після чого автоматично розпочав інсталяцію (goad.sh -t install).

В процесі роботи скрипта консоль відображала послідовно етапи запуску Packer для підготовки відсутніх шаблонів (як правило, цей крок пропускається, якщо шаблони вже створено раніше), ініціалізація Terraform та застосування плану (створення VM у Proxmox), а далі багаторазове виконання різних Ansible-плейбуків. Процес виконання наведено на рисунку 3.6.

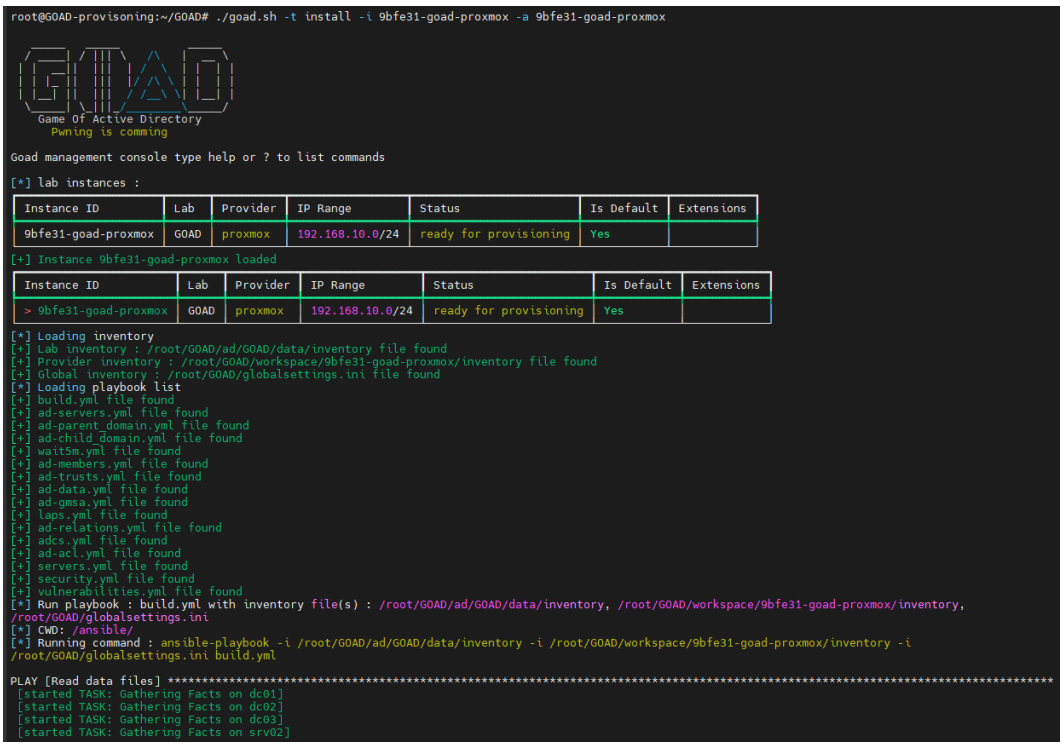


Рисунок 3.6 – Запуск інсталяційного скрипта

Скрипт виводить покрокове виконання кожного кроку розгортання лабораторного середовища, наприклад на рисунку 3.7 наведено виконання ansible плейбуків, що створюють вразливі налаштування середовища Active Directory.

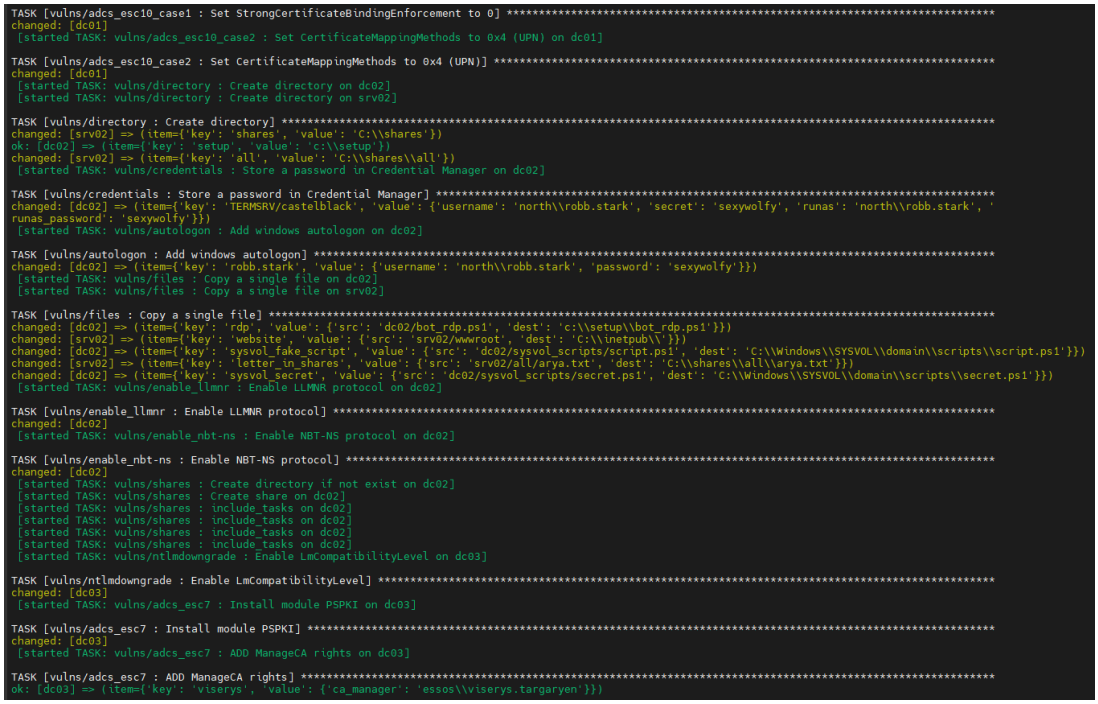


Рисунок 3.7 – виконання ansible плейбуків

По завершенню виконання скрипта виводиться підтвердження, що всі компоненти лабораторного середовища доступні, а також, що всі плейбуки були виконанні. Вивід наведено на рисунку 3.8

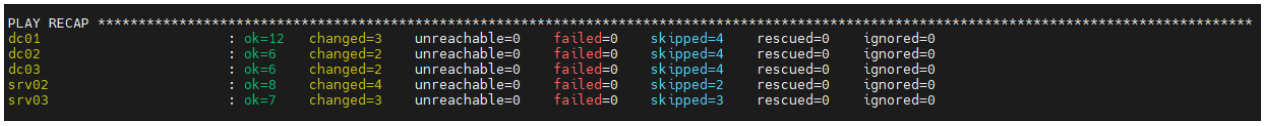


Рисунок 3.8 – вивід підтвердження вдалої інсталяції лабораторного середовища.

3.6 Архітектура Active Directory лабораторного середовища GOAD

Лабораторний стенд Game of Active Directory (GOAD) v3 розгорнуто у віртуальному середовищі Proxmox, що моделює інфраструктуру корпоративного Active Directory з трьома доменами у двох лісах. Така конфігурація відтворює складну мережу з головним доменом і піддоменом у лісі Seven Kingdoms та окремим довіреним лісом Essos. На рисунку 3.9 подано оглядову схему цієї інфраструктури, де кожен трикутник представляє домен з контролерами домену (DC) та сервером, а лінії відображають відносини довіри між лісами та зв’язки (наприклад, зв’язок MSSQL між серверами)

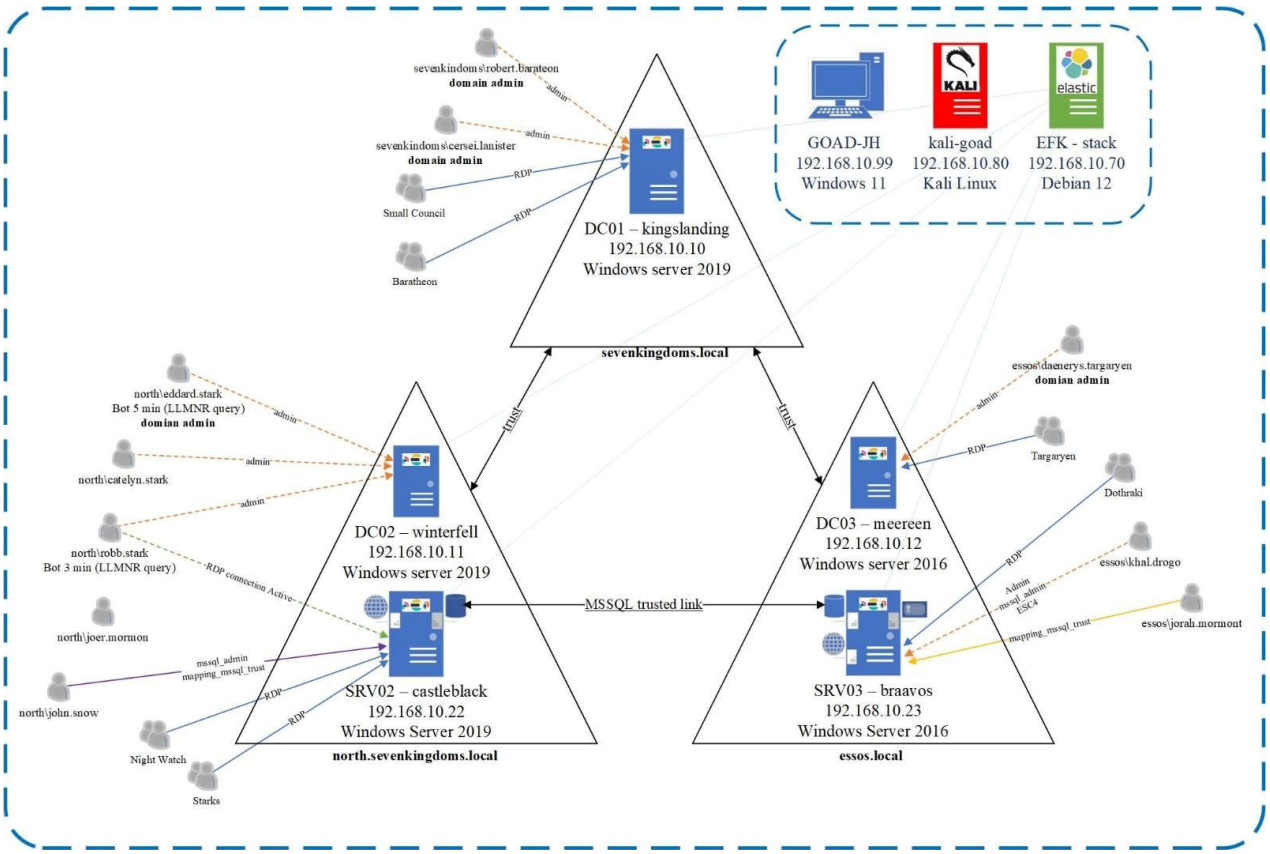


Рисунок 3.9 – Структурна схема лабораторного середовища.

3.6.1 Структура доменів і лісів

Game of Active Directory (GOAD) v3 побудовано у вигляді двох взаємопов'язаних лісів Active Directory, що загалом містять три домени. Перший ліс – SevenKingdoms – складається з кореневого домену sevenkingdoms.local та дочірнього домену north.sevenkingdoms.local. Другий ліс – Essos – представлений окремим доменом essos.local. Між лісами налаштовано довірчі відносини (Forest Trust), що дозволяє користувачам одного лісу бути членами груп іншого.

Склад і ролі серверів. GOAD v3 розгорнуто на п'яти віртуальних машинах, Кожен домен має щонайменше один контролер домену (DC), а також окремі сервери-члени домену з встановленими вразливими службами для відпрацювання атак. Всі сервери працюють під керуванням Windows Server 2016 або 2019; на більшості з них за замовчуванням увімкнено штатний

антивірус Windows Defender, за винятком одного спеціально вразливого сервера, де його вимкнено.

Домени sevenkingdoms.local та north.sevenkingdoms.local розгорнуті в межах єдиного лісу SevenKingdoms. Контролер kingslanding обслуговує кореневий домен sevenkingdoms, а winterfell – дочірній домен north. Окремий ліс Essos складається з домену essos.local, що обслуговується контролером meereen. Додатково в кожному домені присутній один сервер-член: castelblack у домені North та braavos в домені Essos. Ці членські сервери навмисно містять вразливі сервіси (веб-сервер, SQL Server, файлові ресурси тощо) та менш захищені конфігурації (наприклад, castelblack не має активного антивірусу) для спрощення виконання атак.

Адміністративні облікові записи. Для кожного домену визначено окремих доменних адміністраторів, що відображають тематичні персонажі. У кореневому домені sevenkingdoms.local права Domain Admins мають користувачі robert.baratheon та cersei.lannister. Дочірнім доменом north.sevenkingdoms.local керують адміністратори eddard.stark, catelyn.stark та robb.stark. В домені essos.local привілеї адміністратора домену належать daenerys.targaryen. Окрім того, на кожному окремому сервері-члені визначено свій локальний адміністратор: так, jeor.mormont є адміністратором сервера castelblack, а khal.drogo – адміністратор braavos. Вказані облікові записи використовуються для моделювання компрометації привілейованих користувачів під час виконання атак. Наприклад, один з доменних адміністраторів (robert.baratheon) входить до групи *Protected Users*, що ускладнює крадіжку його квитків Kerberos, тоді як інші адміністратори вразливіші. Таким чином, середовище містить різні типи адміністративних цілей – як класичних Domain Admins, так і локальних адміністраторів серверів та спеціальні групи, – що дає змогу відпрацювати сценарії ескалації привілеїв на різних рівнях.

3.6.2 Конфігурація служб (MSSQL, SMB, ADCS тощо).

У лабораторії GOAD v3 навмисно розгорнуто ряд служб та налаштувань Active Directory, відомих наявністю вразливостей або особливостей, корисних для атак. Нижче розглянуто ключові з них:

- Microsoft SQL Server (MSSQL): На серверах castelblack (North) і braavos (Essos) встановлено MSSQL Server, причому на обох реалізовані *довірені зв'язки* (linked servers) між базами даних різних доменів. На castelblack базою керує користувач jon.snow (SQL-адміністратор, привілегія sysadmin), а на braavos – khal.drogo. Налаштування SQL Server дозволяє (EXECUTE AS) під облікові записи інших користувачів: так, в домені North користувач samwel.tarly може виконувати команду EXECUTE AS LOGIN і отримувати права sa на SQL-сервері castelblack, а arya.stark – використовувати EXECUTE AS USER до рівня dbo в окремій базі. Аналогічно, в Essos налаштовано імперсонацію jorah.mormont на braavos для отримання ролі sa. Головне – між SQL-серверами встановлено взаємні *trusted link*: SQL-сервер castelblack довіряє braavos і навпаки. Це означає, що маючи доступ до одного SQL-сервера, атакувальник може перейти на інший. Практична реалізація – зіставлена з тематикою GOT – показує, що облікові дані jon.snow на castelblack дозволяють через linked server виконувати команди на braavos від імені sa (через USE LINK і xp_cmdshell). Отже, конфігурація MSSQL у GOAD v3 дає змогу відпрацювати атаки на довірені зв'язки SQL (спосіб латерального руху між доменами через бази даних).
- Файлові служби (SMB): В обох доменах є файлові ресурси (SMB-шари) на серверах-членах. castelblack і braavos мають налаштовані SMB папки, доступні для зберігання даних (у тому числі потенційно чутливих). З міркувань безпеки у реальних середовищах SMB-аутентифікацію (signing) увімкнено, але в GOAD вона вимкнена для моделювання атак relay. Зокрема, SMB signing відключено, що дозволяє реалізувати NTLM-релей атакуючим сервером. Крім того, деякі SMB-ресурси доступні анонімно або містять слабко

захищені дані. Так, в SYSVOL (мережевий шар контролерів домену) зберігаються скрипти запуску від імені системи, і в одному з них у відкритому вигляді або в зашифрованому вигляді розміщено паролі адміністраторів (наприклад, пароль jeor.mormont виявляється в скрипті на SYSVOL). Деякі користувачі теж використовують слабкі паролі, які можна виявити шляхом перегляду загальнодоступних файлів: наприклад, hodor має пароль, що збігається з іменем (user=password), а секрет samwell.tarly прописаний у тексті опису облікового запису в LDAP. Така конфігурація SMB дозволяє відпрацювати атаки типу Pass-the-Password (виявлення пароля в мережевих ресурсах), Password Spraying по слабких паролях та NTLM-relay на служби (оскільки відсутня аутентифікація , можливо провести реле NTLM на HTTP або LDAP).

- Active Directory Certificate Services (ADCS): В середовищі розгорнуто службу сертифікації Active Directory для емуляції атак на основі цифрових сертифікатів. Зокрема, налаштовано Enterprise CA з Web Enrollment (веб-інтерфейсом видачі сертифікатів). Наявність ADCS підтверджується доступністю веб-сторінки Enrollment за адресою http://192.168.10.23/certsrv (у лабораторії – на сервері з IP 192.168.10.23). Середовище сконфігуровано з типовими *Certificate Templates*, включно з шаблоном DomainController, який за замовчуванням дозволяє видачу сертифікатів контролера домену. У GOAD v3 реалізовано кілька відомих вразливостей й неправильних налаштувань ADCS, класифікованих як ESC (Enterprise Security Certification) 1, 2, 3, 4, 6, 8 тощо. Наприклад, увімкнено Enrollement Web Service і залишено без належного контролю шаблон DomainController – це відкриває шлях для атаки ESC8. Суть ESC8: зловмисник може змусити контролер домену аутентифікуватися на веб-службі реєстрації сертифікатів (через PetitPotam або інший метод coercion) і передати його NTLM-хеш для реллею. У лабораторії це демонструється: викликається *PetitPotam* на meereen.essos.local (DC Essos) для неналежної автентифікації на CA-сервері, а утиліта Impacket ntlmrelayx перенаправляє цю автентифікацію на веб-інтерфейс сертифікації та отримує сертифікат на ім’я

контролера. Отримавши такий сертифікат (що еквівалентно компрометації ключа контролера домену), атакуючий може автентифікуватися як контролер домену і отримати привілеї доменного адміністратора. Окрім ESC8, у середовищі можна випробувати інші атаки на ADCS: ESC1–ESC6 (небезпечно сконфігуровані шаблони сертифікатів, що дозволяють звичайним користувачам отримати привілеї), а також техніку Shadow Credentials (додавання альтернативного імені для облікового запису з наступною видачею сертифіката на чужий обліковий запис) – наприклад, *khal.drogo* має права GenericAll на обліковий запис *viserys.targaryen*, що дозволяє додати до нього сторонній сертифікат. Таким чином, наявність ADCS у GOAD v3 дає змогу моделювати цілий спектр атак на основі зловживання службою сертифікації (від NTLM-релейів і викрадення квитків Kerberos до непомітного отримання привілеїв через сертифікати).

- Інші налаштування безпеки: В середовищі також реалізовано додаткові елементи, що створюють умови для типових атак. Наприклад, на сервері castelblack розгорнуто веб-сервер IIS, який дозволяє завантаження ASP-файлів і працює під обліковим записом NT AUTHORITY\Network Service. Це означає, що на цей сервер можна непомітно завантажити web-shell (ASP-скрипт) і виконувати його з правами мережевої служби, що моделює компрометацію через уразливий веб-додаток. У домені SevenKingdoms присутні облікові записи, що навмисно сконфігуровані для Kerberos Delegation: зокрема, один з користувачів (sansa.stark) має надану опцію *Unconstrained Delegation*, а інший – *Constrained Delegation* до певних сервісів. Це дозволяє експериментувати з атаками через делегування аутентифікації (наприклад, викрадення квитків у незахищеного до делегування користувача). Також впроваджено LAPS (Local Administrator Password Solution) на деяких хостах: для певних учасників середовища (наприклад, *jorah.mormont*) передбачено можливість читання пароля локального адміністратора через атрибут LAPS, що імітує недостатньо захищений доступ до паролів локальних адміністраторів. Всі ці додаткові елементи розширюють різноманітність

можливих технік атаки, що можна відпрацювати в лабораторії. Повний перелік арпзлтвостей Active directory для досліджень наведено на рисунку 3.10

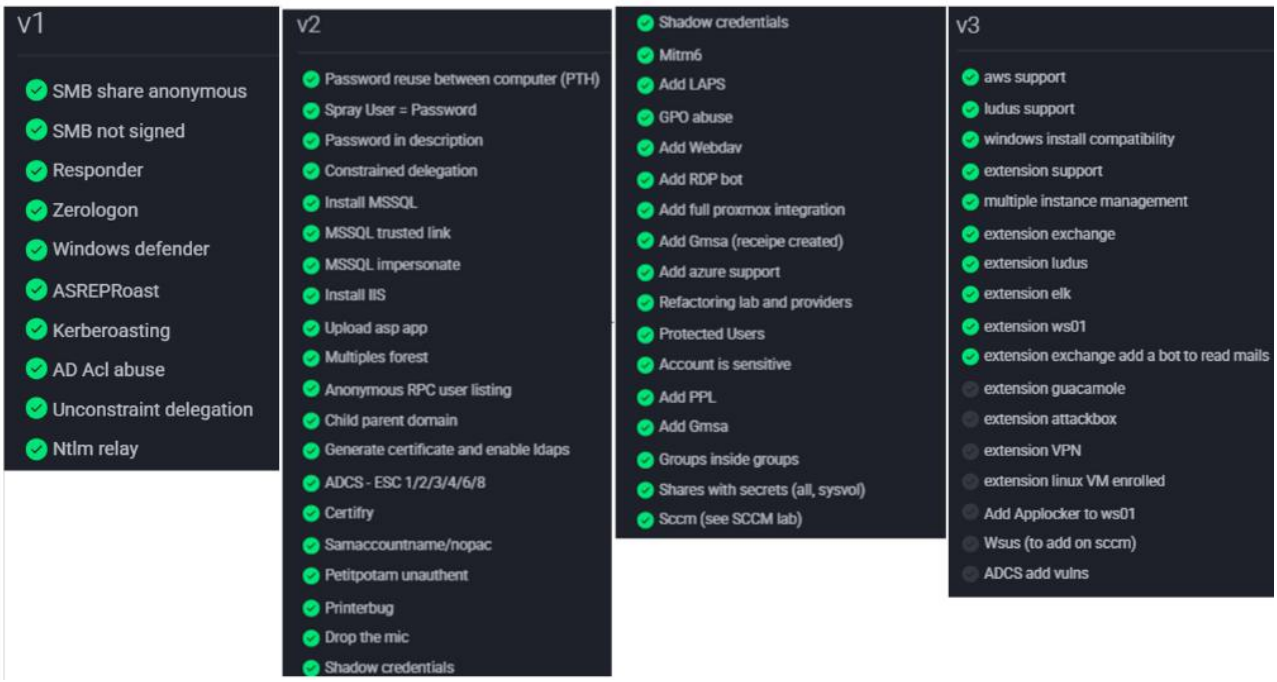


Рисунок 3.10 – Повний перелік атак на середовище GOAD.

Висновки за розділом 3

У третьому розділі було здійснено розгортання лабораторного середовища на базі платформи GOAD v3 в системі віртуалізації Proxmox. Це середовище моделює складну архітектуру корпоративної мережі Active Directory, що включає два ліси, три домени та п’ять серверів Windows з різними ролями та рівнями привілеїв. Створена інфраструктура є функціонально повноцінною та навмисно вразливою, що дозволяє досліджувати практичні аспекти атак і захисту.

Було реалізовано повну автоматизацію розгортання середовища за допомогою інструментів Packer, Terraform та Ansible. Налаштовано маршрутизатор і міжмережевий екран pfSense, створено ізольовані VLAN-сегменти, а також проведено NAT та port forwarding для зовнішнього доступу до критичних компонентів лабораторії. Застосування автоматизованого

скрипта goad.sh забезпечило швидке, повторюване та контрольоване розгортання лабораторії.

У підрозділах було детально описано архітектуру AD-оточення лабораторії GOAD, зокрема структуру доменів і лісів, ролі серверів, взаємозв’язки між компонентами та спеціально змодельовані вразливості (наприклад, ASREP Roasting, Kerberoasting, NTLM Relay, unconstrained delegation, MSSQL trust link, ADCS уразливості тощо). Це середовище створює унікальну можливість для вивчення як атак, так і побудови відповідних моделей виявлення в умовах, максимально наближених до реальних корпоративних мереж.

РОЗДІЛ 4
ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ МОНІТОРИНГУ АТАК

4.1 Загальний принцип роботи модуля Anomaly Detection

Модуль виявлення аномалій в Elasticsearch призначений для автоматичного аналізу часових рядів даних і знаходження нетипових патернів поведінки.

Він побудований на безконтрольному навчанні. Система спочатку формує модель нормальної поведінки за історичними даними (базові рівні), а далі оцінює нові спостереження відносно цієї моделі. Дані з індексів Elasticsearch надходять на вхід ML-завданню (Job), яке обчислює аномалії й присвоює їм «оціночний бал» (anomaly score).

Результати подаються в інтерфейс Kibana: фактичні значення метрик, межі очікуваного діапазону та позначені на графіках виявлені аномалії. Наприклад, у Kibana (Anomaly Explorer) можна побачити фактичні значення метрики, межі очікуваного діапазону та виявлені аномалії на одних і тих же графіках. За побудову моделі відповідає алгоритм Random Cut Forest – ліс випадкових розрізів, який ефективно ідентифікує викиди у складних даних. Модель видає для кожного інтервалу значення критерію аномальності (0–100), де великі значення вказують на суттєві відхилення від норми.

У контексті кібербезпеки модуль Anomaly Detection застосовують для виявлення нетипової поведінки в мережевих та системних логах, що може свідчити про зловмисну активність. Наприклад, різке зростання швидкості генерації логів чи мережевого трафіку може означати DDoS-атаку, а раптове зниження активності – збій або перевантаження сервісу. Аналогічно, аномалія в послідовності подій (наприклад, висока частота невдалих спроб логіну чи появи незвичних помилок) спрацьовуватиме як відхилення від моделі і

сигналізуватиме про потенційну атаку. Такий підхід дозволяє швидко виявляти невідомі загрози: алгоритми не потребують заздалегідь визначених сигнатур, а адаптуються до нових сценаріїв поведінки системи.

Інтерфейс Kibana надає візуальні інструменти (Anomaly Explorer, Swimlanes тощо) для аналізу цих аномалій, показуючи реальні та очікувані значення метрик і допомагаючи аналітикам швидко ідентифікувати підозрілі події.

Модуль Anomaly Detection інтегрований у стек Elastic (через X-Pack Machine Learning) і функціонує як окрема служба в межах кластера. Ключові технічні компоненти:

- Datafeed: процес, що витягує та передає до ML-завдання дані з заданих індексів Elasticsearch. Дані мають мати часову мітку (timestamp) й опціонально фільтруватися чи агрегуватися за полями.
- Job та детектори (Detectors): кожне ML-завдання (Job) складається з одного або кількох детекторів, які описують, які агрегації (кількість подій, сума, середнє тощо) слід обчислювати у фіксованих інтервалах (bucket span). Детектори аналізують часові ряди й виявляють у них аномальні відхилення. Можна також вказати influencers – поля, щоб система автоматично визначала атрибути (наприклад, IP-адреси, імена користувачів), що найсильніше «впливають» на аномалію.
- Алгоритм Random Cut Forest: основний метод ML в Elastic ML для виявлення аномалій. Це ансамбль випадкових дерев (ліс), що швидко ідентифікує викиди (аномалії) у багатовимірних даних. Алгоритм оцінює кожну точку даних за тим, наскільки вона відрізняється від «нормальних» патернів.
- При надходженні даних з Beats (auditbeat, packetbeat, winlogbeat тощо) або Elastic Agent можна активувати стандартні ML-аналізи. Наприклад, є попередньо налаштовані Job, що шукають аномальні обсяги трафіку на хості, рідкісні помилки в логах та інші нетипові патерни.

Вищеописані модулі та налаштування модуля машинного навчання в графічному інтерфейсі Kibana зображені на рисунку 4.1.

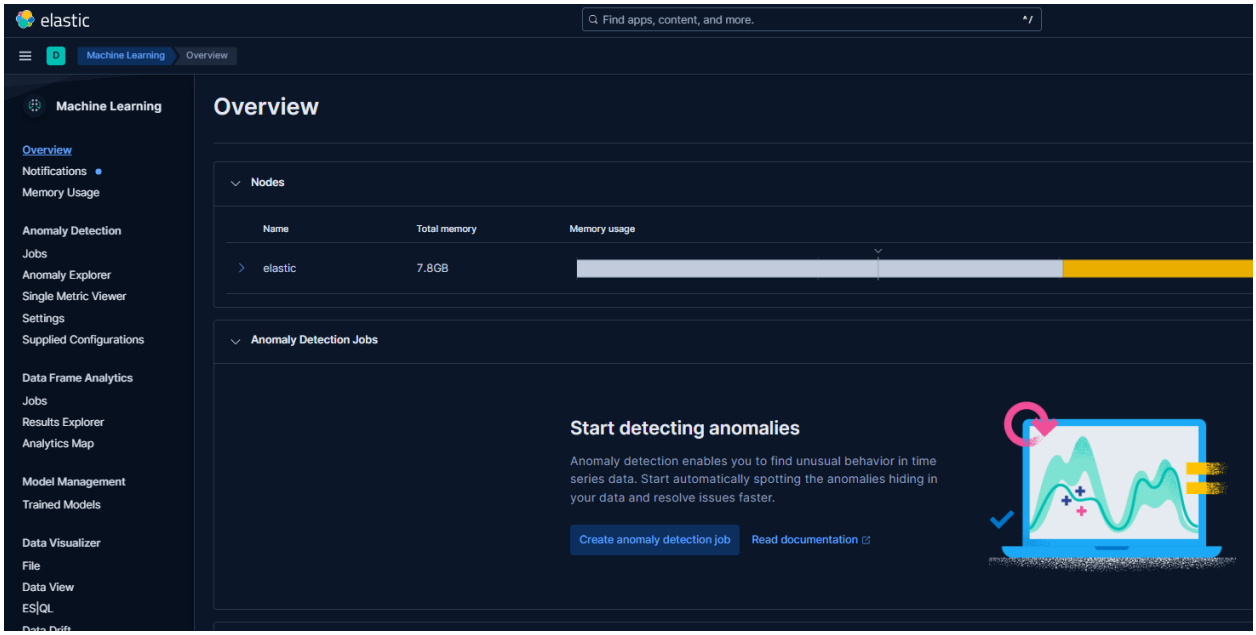


Рисунок 4.1 – Налаштування модулів машинного навчання в Elastic search

Переваги і недоліки

Переваги:

- Адаптивність та реактивність: ML-система самонавчається на нових даних і може швидко реагувати на невідомі сценарії атак без ручного оновлення правил.
- Масштабованість: алгоритми ефективно працюють із великими обсягами багатовимірних даних (наприклад, гігабайти логів із сотень джерел), тоді як традиційні порогові системи можуть не впоратися з таким навантаженням.
- Покращена точність: ML-моделі враховують широкий набір ознак, що допомагає зменшити частоту хибних спрацьовувань у порівнянні з жорстко прописаними правилами.

Недоліки:

- Хибні спрацювання: навіть за оптимальних налаштувань алгоритм може генерувати false positives (нормальні події, помічені як аномальні) або

false negatives (пропущені аномалії). Балансування між цими двома типами помилок вимагає ретельного тюнінгу і може бути нетривіальним.

- Чутливість до даних: надлишковий шум чи нерелевантні дані в логах можуть призвести до помилкових виявлень. Потрібні попередня обробка та нормалізація даних (фільтрація, обробка пропусків).
- Ресурсомісткість: навчання моделей на великих даних та підтримка їх роботи у реальному часі потребує значних обчислювальних ресурсів (CPU, пам’ять). У невеликих кластерах це може бути складніше, ніж у систем з фіксованими порогами.
- Інтерпретованість: результати ML-аналізу виражаються статистично, тому «чорний ящик» моделі може бути складнішим для розуміння порівняно з очевидними правилами, що ускладнює пояснення причин аномалій.

4.2 Емуляція атак

Для проведення атак була використана недоменна Linux-машина з мережевим доступом до контролерів домену GOAD, але без входження в сам домен. Всі інструменти для здійснення атак розташовані на цій машині і не доставляються на жодний доменний хост, тобто атака повністю виконується зовнішньо. Такий підхід дозволяє імітувати реальні сценарії атаки з мережі (віддалено) без шкоди для цільової системи.

4.2.1 Емуляція атаки Kerberoasting у середовищі Active Directory

Kerberoasting – це атака на протокол Kerberos в AD, спрямована на отримання хешів паролів сервісних облікових записів. Зловмисник, маючи доменні облікові дані звичайного користувача, запитує *Ticket Granting Service* (TGS) для будь-яких облікових записів із призначеним SPN (Service Principal Name). Отримані TGS-квитки шифруються ключами, похідними від паролів

сервісних облікових записів, що дозволяє здійснювати їх офлайн-розшифрування і відновлювати паролі. Мета моделювання такої атаки полягає в демонстрації можливості викрадення важливих облікових даних без розгортання шкідливого ПЗ на контролерах домену чи робочих станціях. Оскільки Kerberoasting використовує легітимні механізми Kerberos і зовнішні інструменти (у даному випадку Kali Linux з пакетом Impacket), він майже непомітний для традиційних EDR/антивірусних систем. Таким чином перевіряється здатність системи захисту виявити аномальну активність на основі поведінкових ознак, а не сигнатур чи наявності шкідливих процесів.

Для проведення атаки необхідно виконати наступні передумови:

1) Для правильного перетворення доменних імен в IP адреси і навпаки потрібно записати їх у файл /etc/hosts. Для цього попередньо доцільно провести сканування мережі з використанням утиліти netexec. Вигляд сканування мережі за допомогою утиліти netexec представлений на рисунку 4.2.

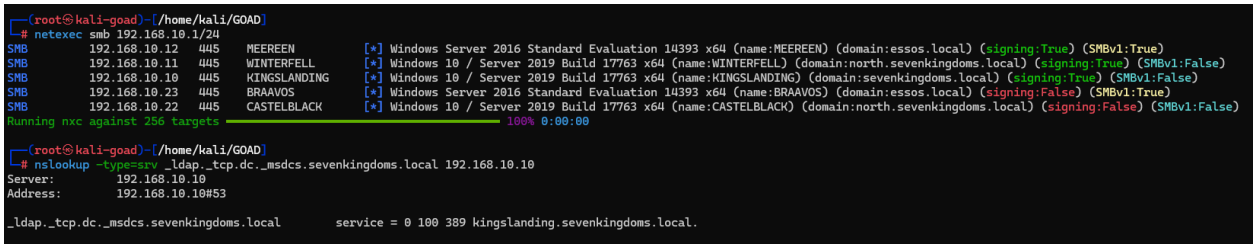


Рисунок 4.2 – Вивід просканованої мережі з використанням утиліти netexec

Після чого додати отримані відповідності ip та доменних імен у файл /etc/hosts. Вивід вмісту файлу наведений на рисунку 4.3.

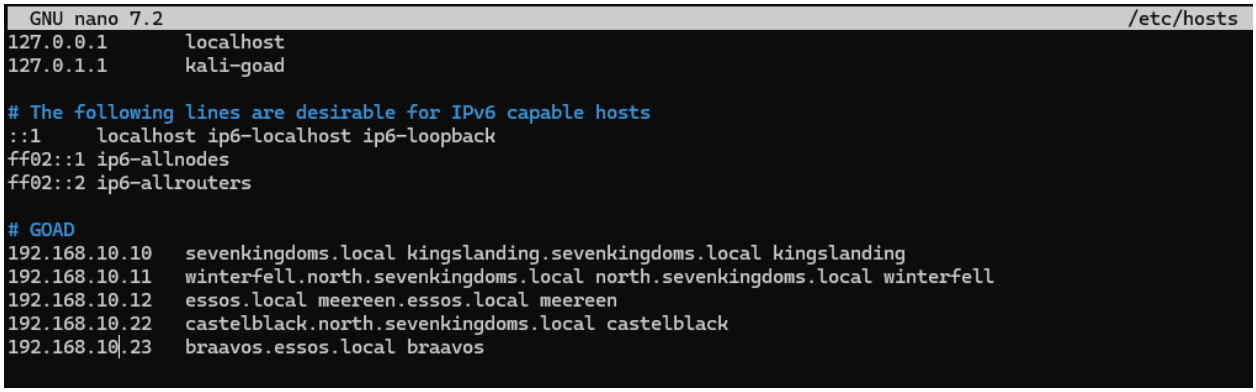


Рисунок 4.3 – Вивід вмісту /etc/hosts

Для коректної роботи протоколу Kerberos з linux системи потрібно встановити наступні параметри служби krb5 в конфігураційному файлі /etc/krb5.conf. Вивід вмісту файлу наведений на рисунку 4.4.

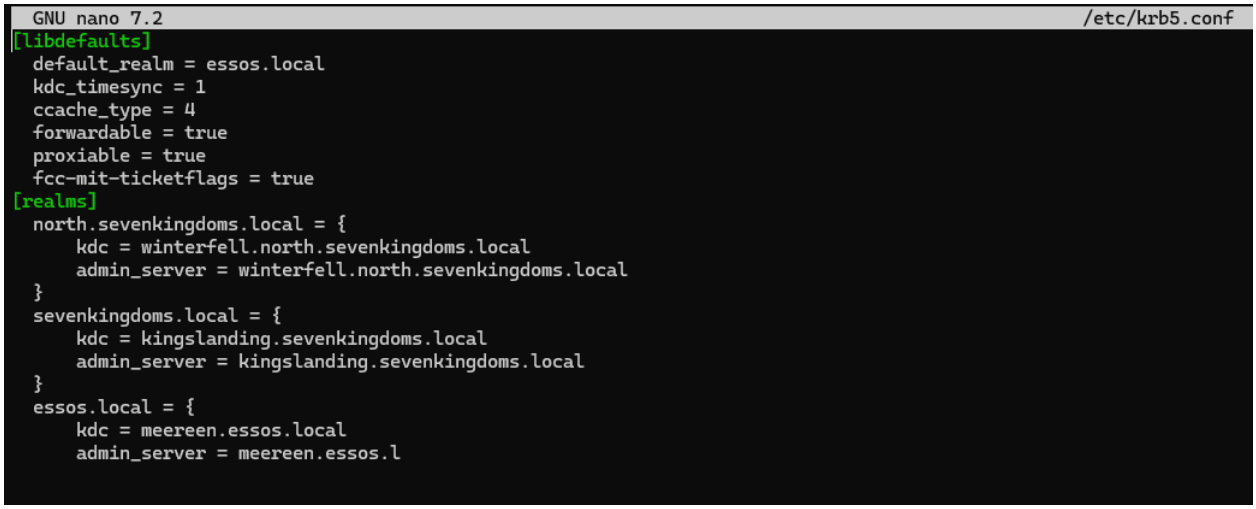


Рисунок 4.4 – Вивід вмісту /etc/krb5.conf

Атакуючий використовував Kali Linux як віддалений вузол з валідними доменними обліковими даними (у прикладі – користувач *brandon.stark* домену *north.sevenkingdoms.local* із паролем *iseedeadpeople*). Для отримання TGS квитків сервісних облікових записів застосовано утиліту Impacket – GetUserSPNs.py. За допомогою цієї команди здійснюється пошук усіх облікових записів із налаштованими SPN і запит відповідних TGS.

Детальніший алгоритм дій:

- Підключення до контролера домену: Impacket використовує передані облікові дані для встановлення Kerberos-сесії з контролером домену (IP 192.168.10.11). При цьому жодних інструментів не інсталюється на самого контролера чи членів домену.
- Виконання сканування SPN: Без ключа -request скрипт просто виводить список «roastable» облікових записів (з ненульовим SPN). З опцією -request відразу запитуються TGS-квитки для знайдених облікових записів, зберігаючи їх у вихідний файл.

- Збереження результатів: Отримані зашифровані квитки записуються в файл kerberoasting.hashes. Кожен рядок цього файлу містить хеш TGS-квитка у форматі Hashcat ($krb5tgs$...), що одразу готовий до зламу офлайн. Наприклад, для RC4-HMAC-криптографії (тип 23) хеш починається з $krb5tgs$23$*, а для AES-256 (тип 18) – з $krb5tgs$18$*.
- Подальше розшифрування: Хеші з файлу kerberoasting.hashes можна передати в брутфорс-інструменти (Hashcat, John the Ripper тощо) для відновлення паролів сервісних облікових записів.

У цьому прикладі застосовано таку команду:

```
impacket-GetUserSPNs -request -dc-ip 192.168.10.11 north.sevenkingdoms.local/brandon.stark:iseedeadpeople -outputfile kerberoasting.hashes
```

Вивід команди наведений на рисунку 4.5.

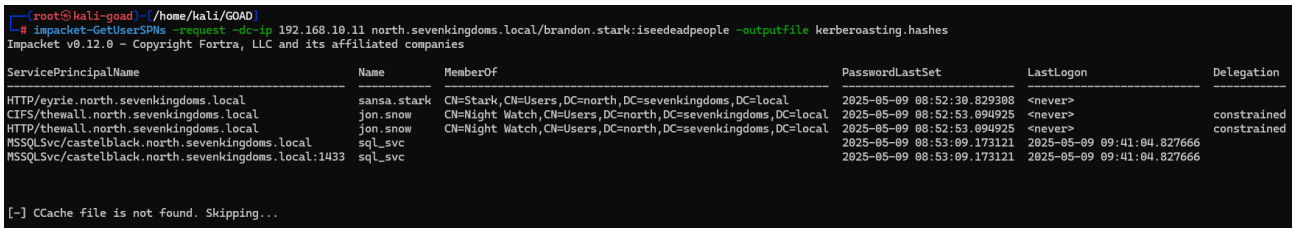


Рисунок 4.5 – Вивід команди impacket-GetUserSPNs

Її параметри мають такі значення:

- -request – вказує скрипту негайно отримати crackable хеш (тобто TGS-квиток) для кожного знайденого SPN. Без цього ключа утиліта лише перераховує вразливі облікові записи (зазвичай при цьому нічого не підвантажується).
- -dc-ip 192.168.10.11 – IP-адреса контролера домену, до якого скрипт відправляє запити Kerberos. Якщо цей параметр опущено, Impacket використовує DNS-ім’я контролера за доменом (що може бути неможливим у ізольованих середовищах).
- north.sevenkingdoms.local/brandon.stark:iseedeadpeople – домен, ім’я користувача та його пароль. У Impacket формат запису облікових даних

підтримує domain/username:password (або domain\\username:password). В нашому випадку це домен north.sevenkingdoms.local, користувач brandon.stark з паролем iseedeadpeople.

- -outputfile kerberoasting.hashes – назва файлу для збереження отриманих хешів. Без цього ключа всі хеші виводилися б у консоль. Вказане ім’я файлу kerberoasting.hashes використовується далі для офлайн-аналізу.

В результаті виконання команда формує файл kerberoasting.hashes, куди записуються знайдені зашифровані квитки TGS (їхні хеші). Кожен рядок файлу містить ім’я сервісу, ім’я облікового запису та сам хеш у спеціальному форматі. Наприклад, отримані рядки можуть виглядати як $krb5tgs$23$*ServicePrincipalName*...$*encrypted_data*, що відповідає RC4-шифруванню. Далі ці хеші піддаються брутфорсу через Hashcat (режим 13100) або інші інструменти, аби відновити паролі сервісних акаунтів. Отримані хеші наведені на рисунку 4.6.

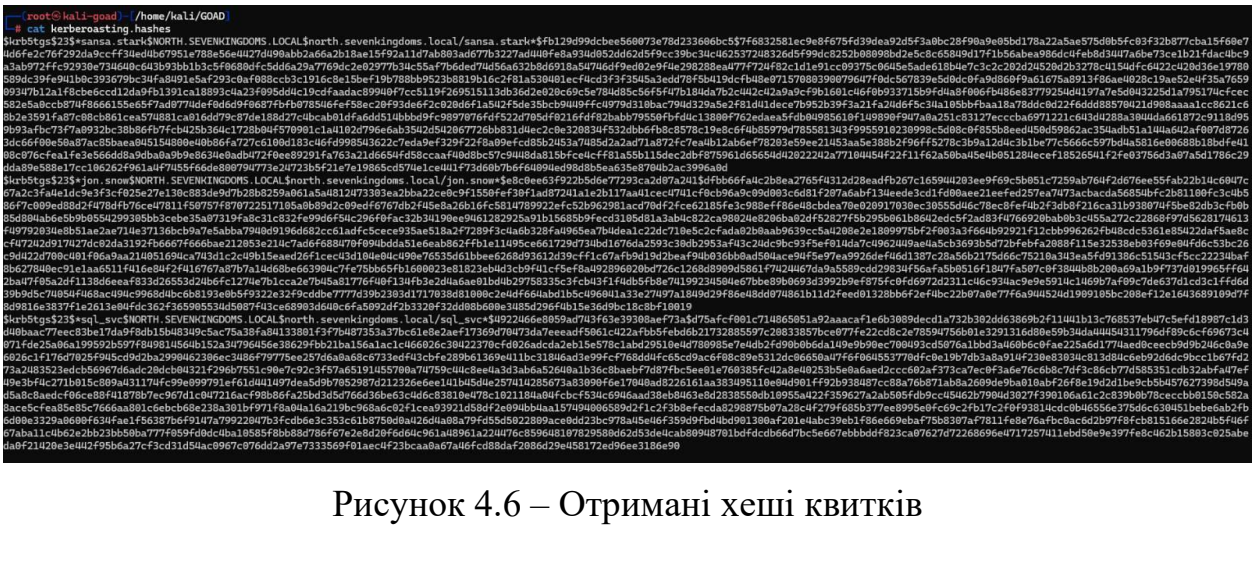


Рисунок 4.6 – Отримані хеші квитків

На стороні контролера домену кожен запит TGS квитка реєструється у системному журналі безпеки подією з кодом **4769** (успішно отримано квиток TGS). У типовому сценарії після вхідного входу користувач запитує квитки доступу до декількох сервісів (зазвичай до 10–20 TGS на день). Однак при Kerberoasting один обліковий запис може за короткий час послідовно запитувати багато різних сервісних квитків. Саме така аномально висока частота подій 4769 від одного користувача є ключовим індикатором атаки.

При проведені атаки було помічено аномальне отримання квитків одночасно для всіх сервісів від одного користувача із зниженим рівнем шифрування квитка(RC4-HMAC), що є індикатором компрометації користувача та виконання Kerberoasting атаки. Вивід подій із SIEM системи представлений на рисунку 4.7.

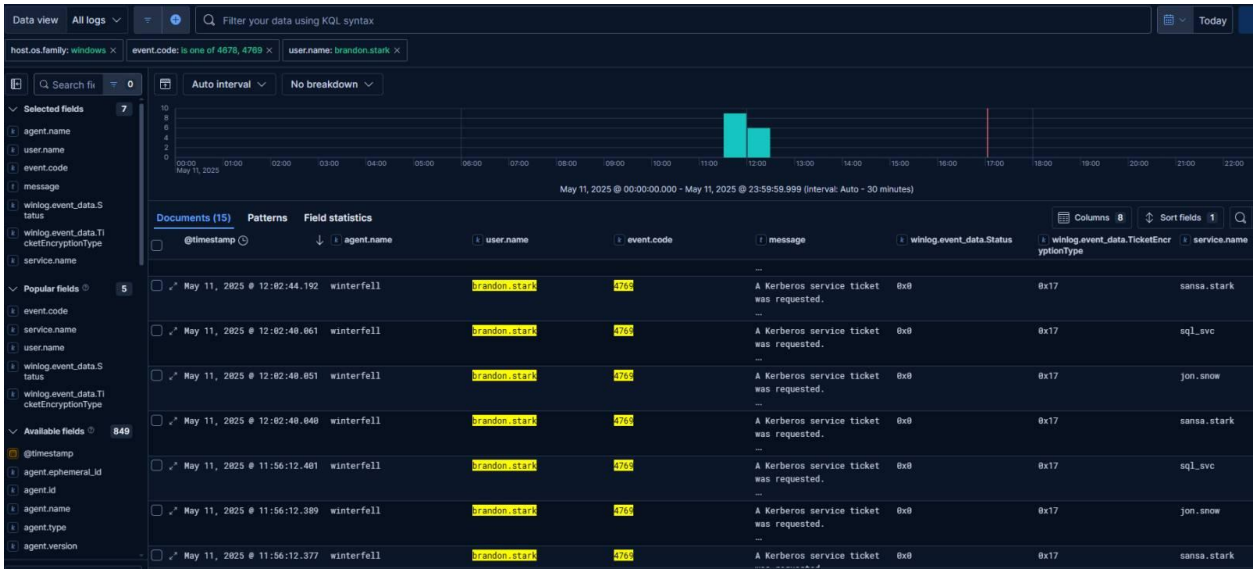


Рисунок 4.7 – Вивід подій отримання TGS квитків від користувача brandon.stark для різних сервісів

Подія 4769 містить деталі запиту: ім’я запитуючого облікового запису, ім’я сервісу (SPN), обліковий запис сервісу (Service ID), а також IP-адресу та порт клієнта. Наприклад, у журналі видно щось на кшталт “Service Name: *HTTP/*sqlservice; Service ID: *DOMAIN*\sqlservice; Client Address: *192.168.10.80*”. Важливо, що інструменти атаки Kerberoasting часто запитують квитки з використанням застарілого шифрування RC4-HMAC (код 0x17). У звичайних запитах Kerberos використовуються AES (0x11 або 0x12), тоді як RC4 (0x17) використовують, аби полегшити офлайн-розшифровку отриманих хешів. Тому поява багатьох подій 4769 із шифруванням 0x17 або декілька запитів від одного й того ж нелегендарного облікового запису (без суфікса $) за короткий час може свідчити про атаку. Вигляд квитка із такими особливостями представлений на рисунку 4.8

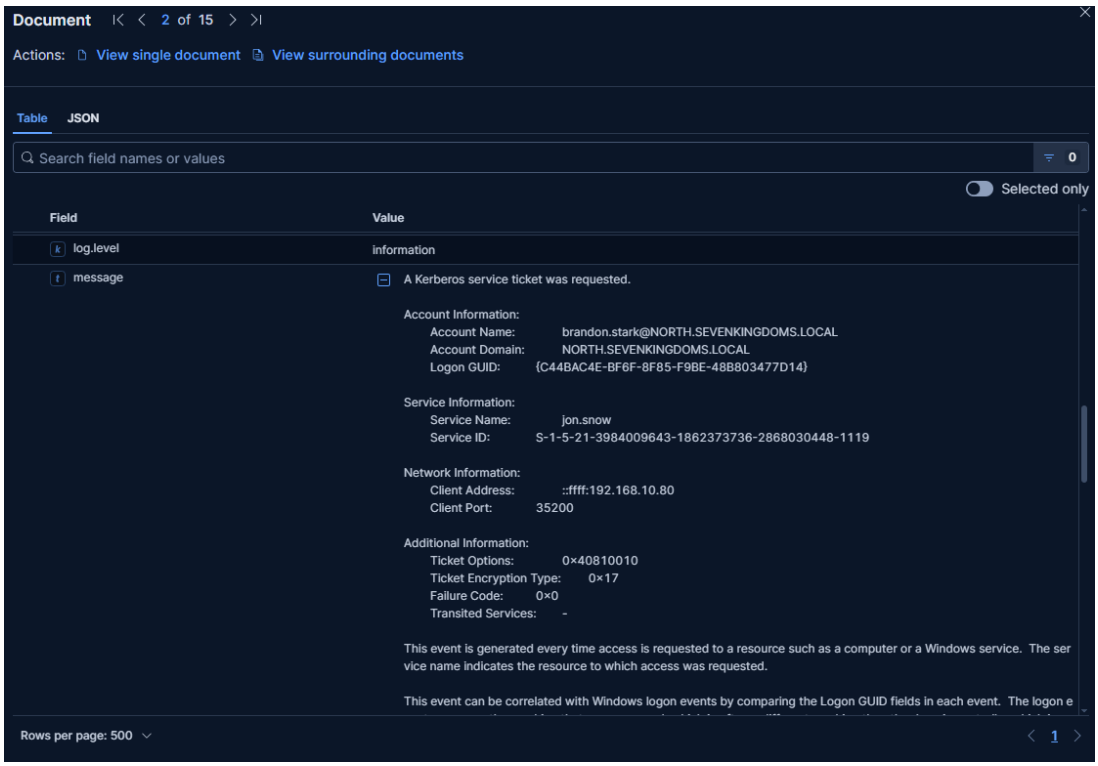


Рисунок 4.8 – Вигляд TGS квитка при проведенні Kerberoasting атаки

Оскільки Kerberoasting не потребує встановлення жодного зловмисного ПЗ і виконується від імені законного користувача, традиційні EDR/антивірусні системи його просто не відстежують. Такі рішення здебільшого аналізують підозрілі процеси або сигнатури, але не звертають уваги на стандартні запити Kerberos. Унаслідок цього активність Kerberoasting “проходить повз вушка”: на контролерах домену немає шкідливих подій, лише записи 4769, які технічно є легітимними.

Водночас ефективність моделі атаки Kerberoasting підкреслює важливість поведінкового аналізу і машинного навчання у захисті AD. Аналіз журналів може виділити аномальні патерни – наприклад, різке зростання кількості запитів TGS від одного користувача чи використання небезпечних алгоритмів шифрування.

Системи SIEM/XDR із ML-модулями можуть навчити звичайному «базовому» рівню запитів Kerberos і виявити незвичні відхилення. Зокрема, пошук акаунтів з непропорційно великою кількістю подій 4769 або із запитами підстарілого RC4-шифрування служить хорошим тригером. Загалом

проведена емуляція показала, що Kerberoasting як чисто мережевий сценарій є дуже стійким до традиційних захисних заходів і актуальним для відпрацювання логічних правил детектування на основі поведінкового моніторингу.

4.2.2 Емуляція атаки Golden Ticket у середовищі Active Directory

Атака Golden Ticket проводилася віддалено з Kali через легітимні протоколи (LDAP/DCE-RPC для збору даних і Kerberos для аутентифікації).

Основні кроки атаки були такими:

1. Отримання NTLM-хешу krbtgt. За допомогою утиліти impacket-secretsdump з параметром -just-dc-user була здійснена DCSync-операція для користувача krbtgt. Використано команду:

impacket-secretsdump -just-dc-user north/krbtgt north.sevenkingdoms.local/eddard.stark:'FightP3aceAndHonor!'@192.168.10.11

Параметри: north.sevenkingdoms.local/eddard.stark:'FightP3aceAndHonor!'@192.168.10.11 – облікові дані адміністратора домену (DC) та IP контролера. Опція -just-dc-user north/krbtgt вказує отримати дані тільки для користувача krbtgt з директорії домену. Результатом команди є вивід NTLM-хешу облікового запису krbtgt (наприклад, рядок виду krbtgt:502:...:13354bc6e1b48fff8d66a2090e909b27). Саме цей хеш потрібен для подальшого кроку – створення підробленого квитка.

Хід атаки зображений на рисунку 4.9

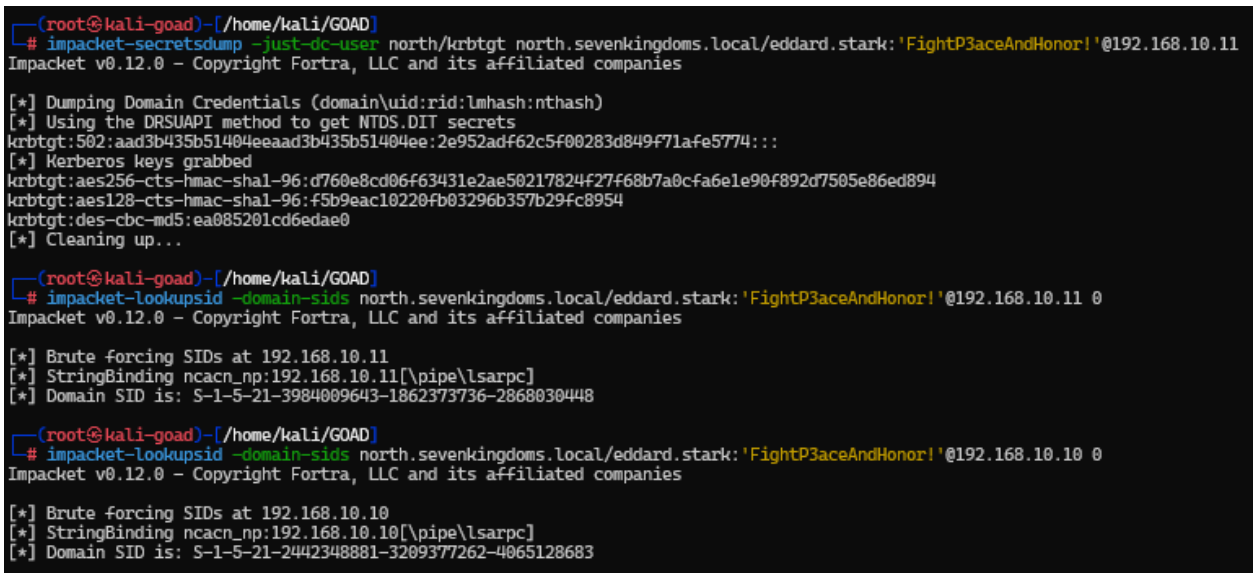


Рисунок 4.9 – отримання NTLM-хешу krbtgt та SID доменів

2. Формування Golden Ticket. Використано утиліту impacket-ticketer для генерації підробленого TGT. Команда має вигляд:

impacket-ticketer -nthash <krbtgt_hash> -domain-sid <Domain_SID> \ -domain north.sevenkingdoms.local -extra-sid <DomainAdmins_SID> goldenuser

Параметри: -nthash – NTLM-хеш krbtgt з попереднього кроку, -domain-sid – SID цільового домену (отриманий через lookupsid.py), -domain – назва домену, -extra-sid – SID групи (або груп) додаткових привілеїв (в прикладі – SID групи Enterprise Admins або Domain Admins, щоб надати підвищені права), а в кінці – ім’я фейкового користувача (goldenuser), від імені якого створюється квиток. В результаті цієї команди створюється файл типу goldenuser.ccache, що містить підроблений квиток Kerberos TGT з вказаними атрибутами (наприклад, з доданим SID адміністраторів через -extra-sid).

Експортування квитка через KRB5CCNAME. Щоб інструменти Impacket використовували згенерований квиток, встановлено змінну середовища KRB5CCNAME на шлях до .ccache файлу:

export KRB5CCNAME=/home/kali/GOAD/goldenuser.ccache

Згідно з документацією, змінна KRB5CCNAME вказує шлях до файлу кешу квитків Kerberos – завдяки цьому інструменти Impacket з ключем -k будуть завантажувати квиток безпеки з цього файлу.

Хід атаки продемонстрований на рисунку 4.10.

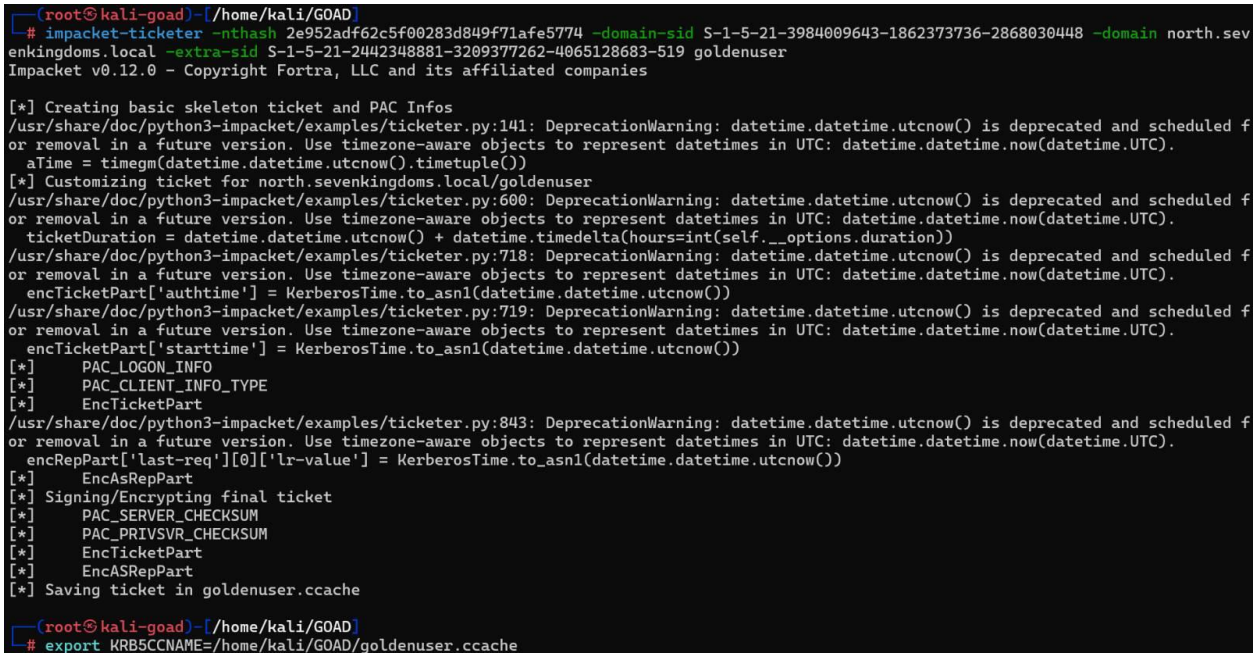


Рисунок 4.10 – формування Golden Ticket та експорт квитка через KRB5CCNAME

3. Використання штучно створеного квитка для доступу. Виконано команду impacket-secretsdump з параметрами -k та -no-pass, яка використовує Kerberos-квиток замість пароля:

impacket-secretsdump -k -no-pass -just-dc-ntlm \ north.sevenkingdoms.local/goldenuser@kingslanding.sevenkingdoms.local

Тут -k означає аутентифікацію за допомогою Kerberos-квитка (KRB5CCNAME), -no-pass – ігнорування пароля, а -just-dc-ntlm – запит реплікації NTLM-хешів з контролера домену (аналог DCSync). Параметр north.sevenkingdoms.local/goldenuser@kingslanding.sevenkingdoms.local означає, що за допомогою квитка користувача goldenuser (імпонована особа) проводиться запит до домену kingslanding.sevenkingdoms.local (в прикладі – батьківський домен лісу). Як результат – зливаються NTLM-хеші користувачів цільового домену без попередньої аутентифікації з допомогою пароля. Таким чином, після генерації Golden Ticket атакуючий отримує повний доступ до ресурсів та облікових записів AD.

Використання квитка з ціллю отримання всіх NTLM хешів в домені представлене на рисунку 4.11

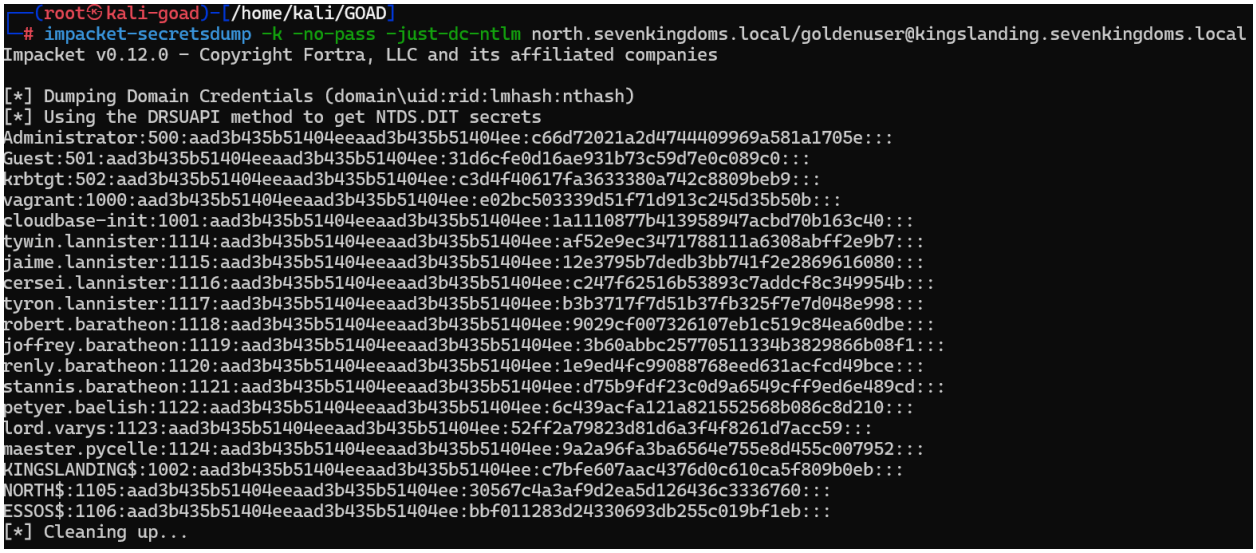


Рисунок 4.11 – використання отриманого квитка для дампу NTLM хешів з домену.

Емульовані команди та параметри детально відповідають знанням з літератури про Golden Ticket: наприклад, Impacket-скрипт ticketer.py дійсно записує квиток у .ccache файл, а команда secretsdump -k -no-pass дозволяє використовувати такий квиток для отримання інформації з контролера домену. Отримання NTLM-хешу krbtgt через DCSync (команда secretsdump -just-dc-user) також описане в інших джерелах як стандартний крок атаки T1558.001.

Атака Golden Ticket практично невловима звичайними антивірусами чи EDR, бо відбувається «легітимними» інструментами та протоколами. Зловмисник не залишає на машинах домену жодних нових процесів або файлів – він просто використовує легальну службу Kerberos і LDAP/AD функції (реплікація). Внаслідок цього стандартні сигнатури безпеки нічого не фіксують: наприклад, AV не відрізняє завантаження правдивого TGT від згенерованого, а EDR не виявляє підозрілої діяльності, бо немає незвичного виконання коду на хості. Завдяки повторному використанню “достовірного” квитка атакуючий може безперешкодно здійснювати свою активність,

ухиляючись від традиційних механізмиі захисту та аутентифікації (наприклад, підрахунку невдалих логінів).

Разом з тим, при відповідному моніторингу подій безпеки можна відслідкувати нетипову поведінку, пов'язану з фальсифікацією квитка. Зокрема, слід шукати невідповідності в Kerberos-автентифікації: ймовірне поєднання подій 4769 та 4768 може бути аномальним. Звичайно подія 4768 (запит TGT) має передувати 4769 (запит сервісного квитка), але при Golden Ticket можливе створення події 4769 без фіксації попередньо 4768. Іншими словами, «неможливо запросити сервісний квиток без TGT», тому відсутність запису про видану TGT та наявність 4769 вказують на підробку. Додаткові індикатори – повторне використання квитків або невідповідність SID.

Наприклад, у новіших версіях Windows у структурі Kerberos PAC з'являються додаткові поля (PAC_REQUESTOR), і якщо SID «видавника» квитка не збігається з очікуваним, генерується помилка (SID-mismatch). Така невідповідність може бути зафіксована в логах системи безпеки як сигнал про аномальну авторизацію.

Також корисно звертати увагу на нетипові властивості квитків: надто довгий час життя (у стандарті 10 годин, а підроблений може бути від 10 років) або рідкісні методи шифрування (наприклад, RC4 замість AES). Незвичними будуть запити високопривілейованих облікових записів ($) до сервісів, до яких вони рідко звертаються. У контексті Windows Security Log слід збирати події: 4769 (запит TGS – містить інформацію про ім'я цільового облікового запису, тип шифрування тощо), 4624 (логон – тут є SID та IP-адреса джерела) та 4627 (перевірка членства у групі – SID-значення груп). Наприклад, у випадку Golden Ticket можливі записи про те, що обліковий запис $ (сервіс) запитує вищий квиток, або що групи у квитку зловмисника не співпадають із очікуваними.

Описані кроки повністю відповідають технікам **MITRE ATT&CK**: створення та використання «золотого квитка» позначено як T1558.001 (Steal or Forge Kerberos Tickets: Golden Ticket), а подальше застосування отриманого

квитка (Pass-the-Ticket) – як T1550.003 (Pass the Ticket: Kerberos). Зокрема, MITRE вказує, що зловмисник, володіючи NTLM-хешем облікового запису KRBTGT, може генерувати облікові квитки з усіма правами в домені. Таким чином, емульована атака демонструє практичну реалізацію цих MITRE ATT&CK технік на прикладі домену GOAD.

4.2.3 Виявлення аномальних аутентифікацій через Kerberos у Active Directory

Атака здійснювалася з хосту Kali Linux за допомогою утиліти bloodhound-python з використанням облікового запису jon.snow, що є нетиповим для цієї індивідуальної обліковки. Аномалія полягала в тому, що з невідомого раніше хоста (Kali) на контролері домену з’явилися події видачі квитків Kerberos. Зокрема, у журналі безпеки були записані події типів 4768 і 4769 (TGT і TGS). Ці події реєструються на контролері домену щоразу, коли Key Distribution Center видає відповідно квиток авторизації (TGT) або сервісний квиток (TGS). Поява таких подій з IP-адреси Kali для користувача jon.snow спостерігалася вперше і не відповідала історичному профілю входів цього користувача. Вигляд запуску команди представлений на рисунку 4.12.

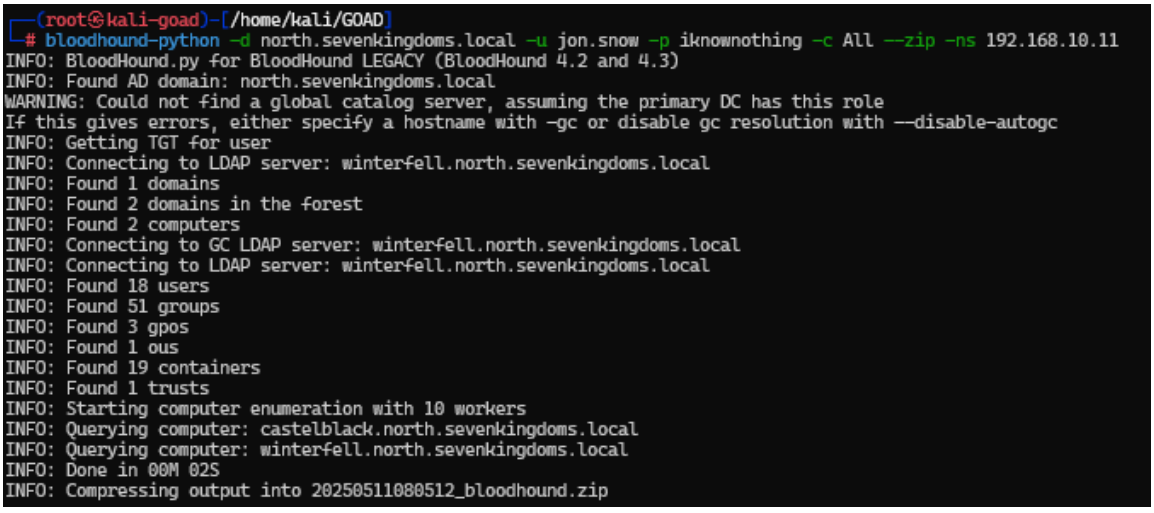


Рисунок 4.12 – запуск команди bloodhound-python для дослідження домену

Періодично атаки на сервісні облікові записи Kerberos (Kerberoasting) супроводжуються масовими запитами TGS. Дослідження показують, що велика кількість подій 4769 від одного облікового запису за короткий проміжок часу може свідчити про атаку. У нашому випадку множинні запити квитків від jon.snow з нового хоста (Kali) явно виходять за межі нормальної поведінки облікового запису.

Варто підкреслити, що традиційні сигнатурні засоби захисту (антивіруси/EDR) в даному випадку не спрацювали. Атака не передбачала встановлення шкідливого ПЗ на внутрішні хости – використовувалися звичайні Kerberos-запити, що виглядали легітимними. Як наслідок, система виявлення на основі сигнатур не зареєструвала загрози, адже не було відомого шкідливого індикатора. Натомість поведінковий аналіз ефективно виявляє відхилення від норми. Elastic ML веде базу звичайних дій кожного користувача (типові IP, час входів тощо) та спрацьовує на незвичайних подіях.

Наприклад, анональні патерни входу – такі як нетиповий час або місце входу користувача – вважаються підозрілими. У нашому сценарії факт вхідної сесії з незвичної IP-адреси було розпізнано як аномалію завдяки аналізу поведінки.

Вигляд отриманих результатів від запуску bloodhound-python у вигляді граф-схеми зображений на рисунку 4.13

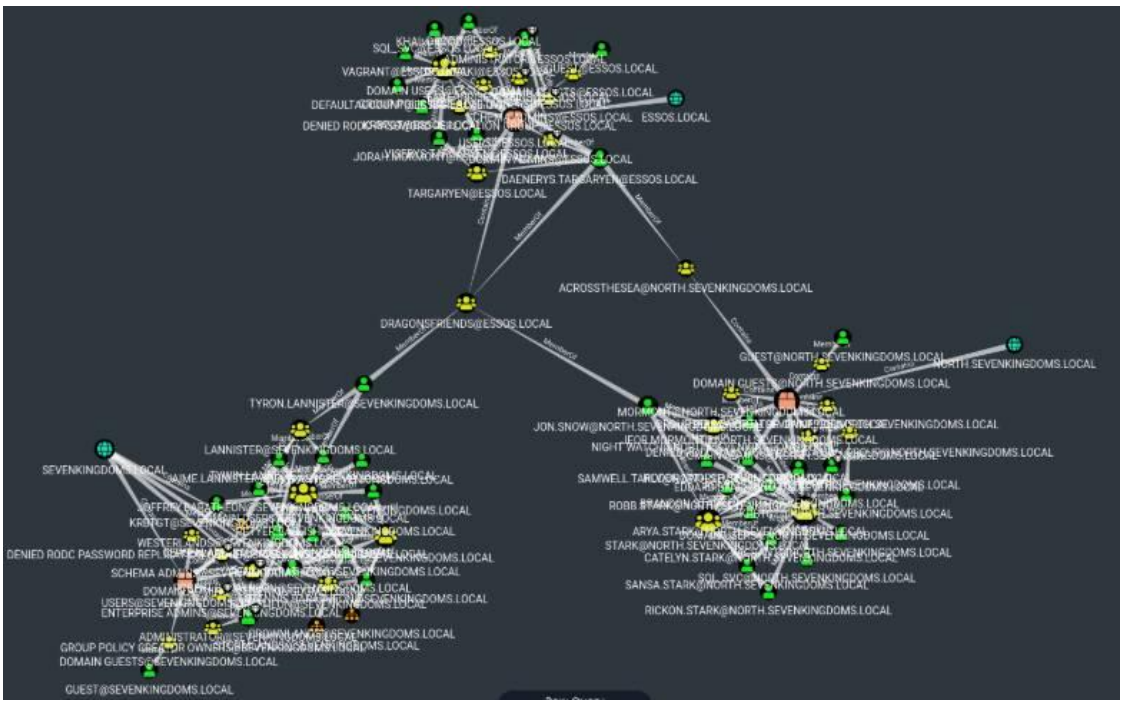


Рисунок 4.13 – вивід граф-схеми доменних залежностей зібраних за допомогою bloodhound

4.3 Виявлення атак за допомогою Elastic Anomaly Detector

Модуль машинного навчання Elastic Stack дозволяє виконувати виявлення аномальної активності у даних журналів, що надходять у реальному часі. Anomaly Detection Job, орієнтований на виявлення чотирьох критичних типів атак у середовищі Active Directory, що здійснюються через протокол Kerberos: Kerberoasting, AS-REP Roasting, аномальні аутентифікації з використанням протоколу Kerberos, а також використання Golden Ticket.

4.3.1 Мета та логіка побудови моделі

Побудована модель аналізує події типу 4769 (TGS-REQ), які фіксують запити квитків служби Kerberos. За логікою протоколу, нормальні TGS-запити мають надходити від користувачів, що попередньо отримали квиток TGT. Однак при атаках типу Golden Ticket або масовому витоку паролів ці запити можуть ініціюватися аномальною кількістю разів або з нехарактерних хостів.

Основною гіпотезою при формуванні моделі була наступна: аномальна активність Kerberos проявляється як поява нових або нетипових користувачів у логах подій 4769 з успішною автентифікацією (Status: 0x0), особливо у короткий проміжок часу.

4.3.2 Структура та конфігурація Job

Налаштування Anomaly Detection Job здійснювалося через інтерфейс Kibana та включає наступні параметри:

Тип: anomaly_detector, зображено на рисунку 4.14

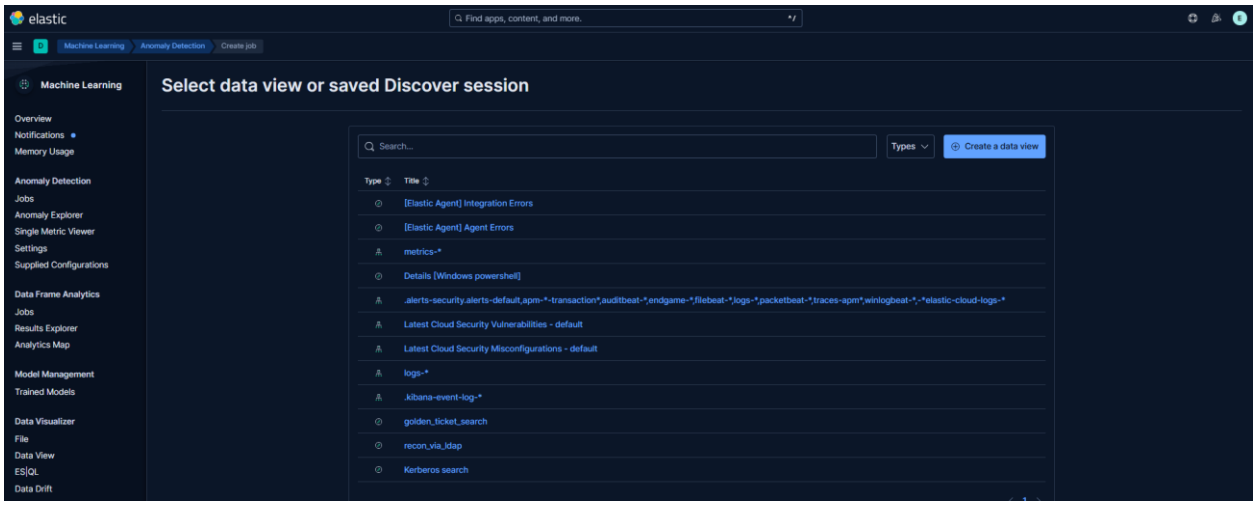


Рисунок 4.14 – налаштування anomaly detection

Функція: rare, зображено на рисунках 4.15 – 4.16

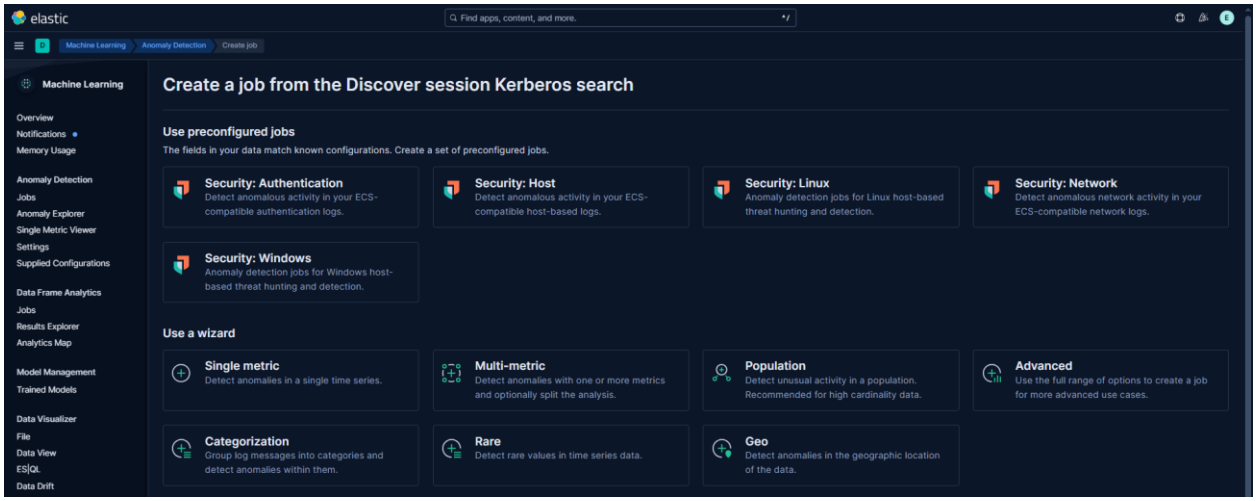


Рисунок 4.15 – обрання функці rare

Рисунок 4.16 – вивід даних, що потрапляють під фільтр подій аутентифікації Kerberos

Наступним фактором є задання впливових факторі. Їз повинно бути не більше трьох з додаванням основного поля для пошуку. Для цілей дослідження було використано поля user.name та winlog.event_data.TicketEncryptionType. Модель орієнтується на рідкісні значення поля user.name, що дозволяє виявляти користувачів, які вперше з’являються в системі або активність яких є нетиповою для заданого відрізку часу. Налаштування зображено на рисунках 4.17 - 4.18.

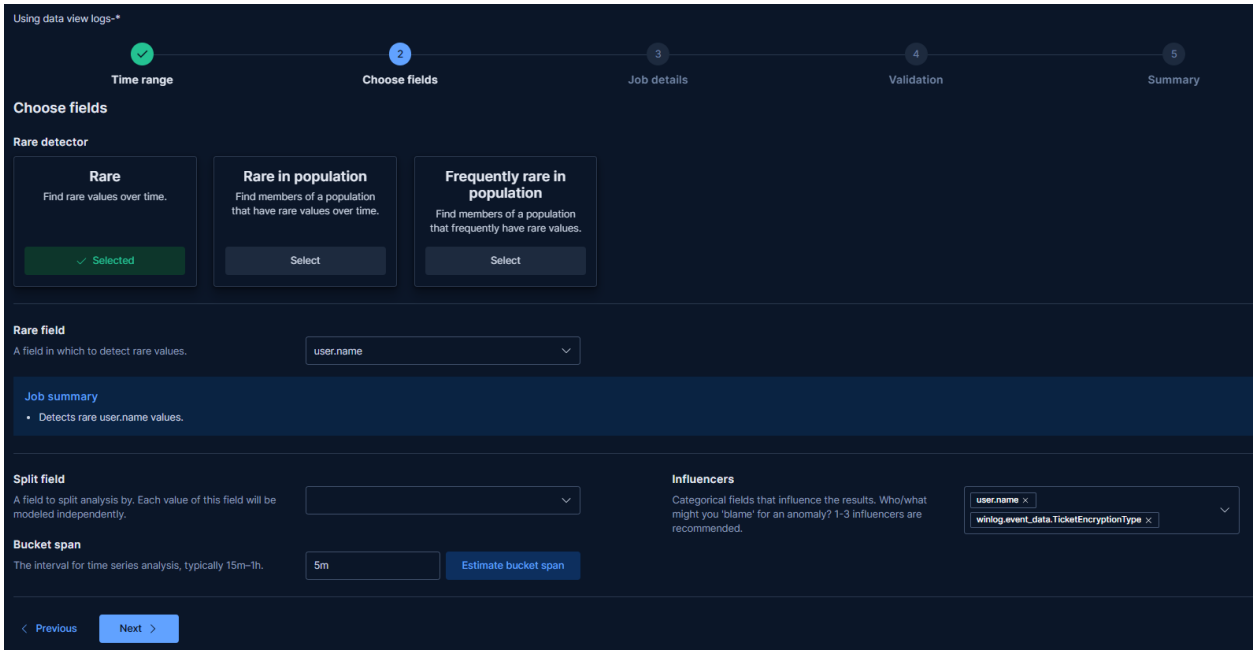


Рисунок 4.17 – налаштування впливових факторів моделі

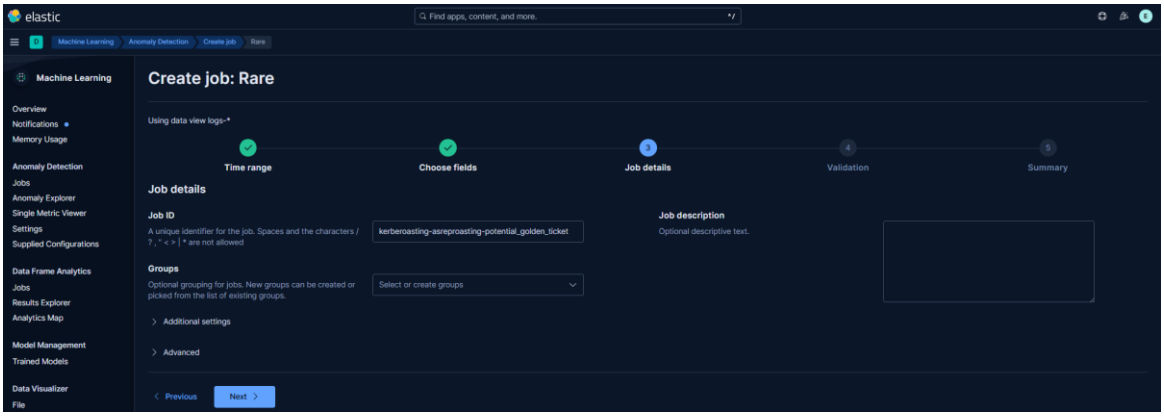


Рисунок 4.18 – задання Job ID

Перевірка налаштувань моделі перед запуском наведено на рисунку 4.19

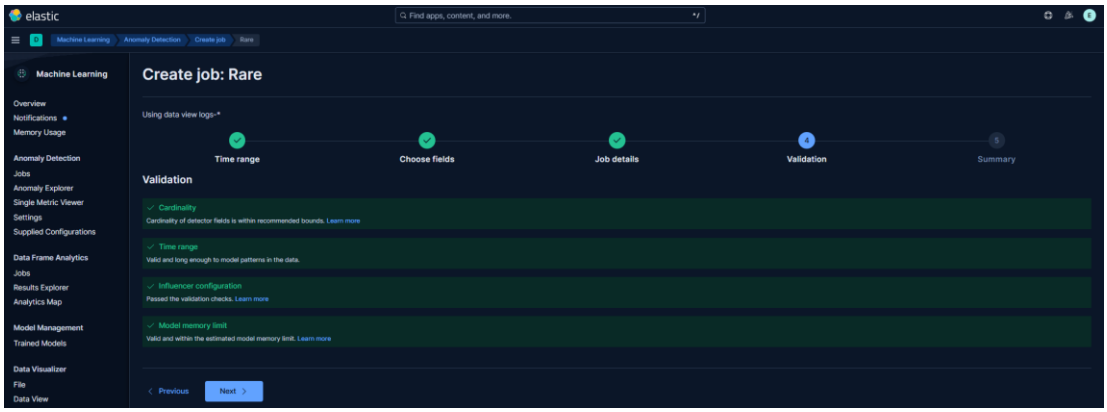


Рисунок 4.19 – підтвердження відповідності моделі вимогам налаштування детектора

Задачу створено та застосовано рисунок 4.20

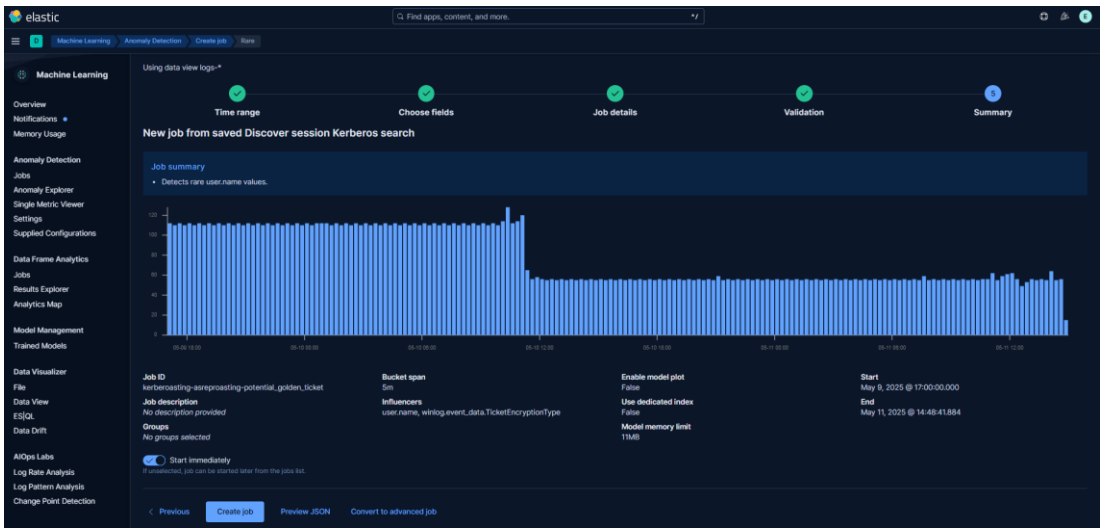


Рисунок 4.20 – Запуск моделі для проактивного виявлення загроз

Після виконання було отримено наступні параметри продуктивності та статистики:

- Кількість оброблених записів: 14 652
- Обсяг оброблених полів: 29 304
- Середній час обробки одного bucket: ~0.88 мс
- Максимальний час bucket: 28 мс
- Кількість bucket’ів: 567

Навантаження на систему було мінімальним, модель працювала у реальному часі, без затримок та з високою точністю побудови поведінкових патернів.

4.3.3 Виявлення аномалій

Протягом періоду моніторингу було зафіксовано кілька інцидентів, які не відповідали типовим патернам і саме ці інциденти були згенеровані:

- Поява нового користувача goldenuser, який не здійснював попередньо запитів до KDC, але був авторизований через події 4769.
- Активація великої кількості TGS-запитів з одного джерела за короткий проміжок часу.

- Поява раніше невідомих SID у полі extra-sid, що може свідчити про створення фальшивого квитка Golden Ticket.

Такі події були виявлені як аномальні завдяки функції rare by user.name, що дозволяє виявляти активність користувачів, які вперше з’являються у зазначеному часовому bucket. Відповідні параметри зображено на рисунках 4.21, 4.22.

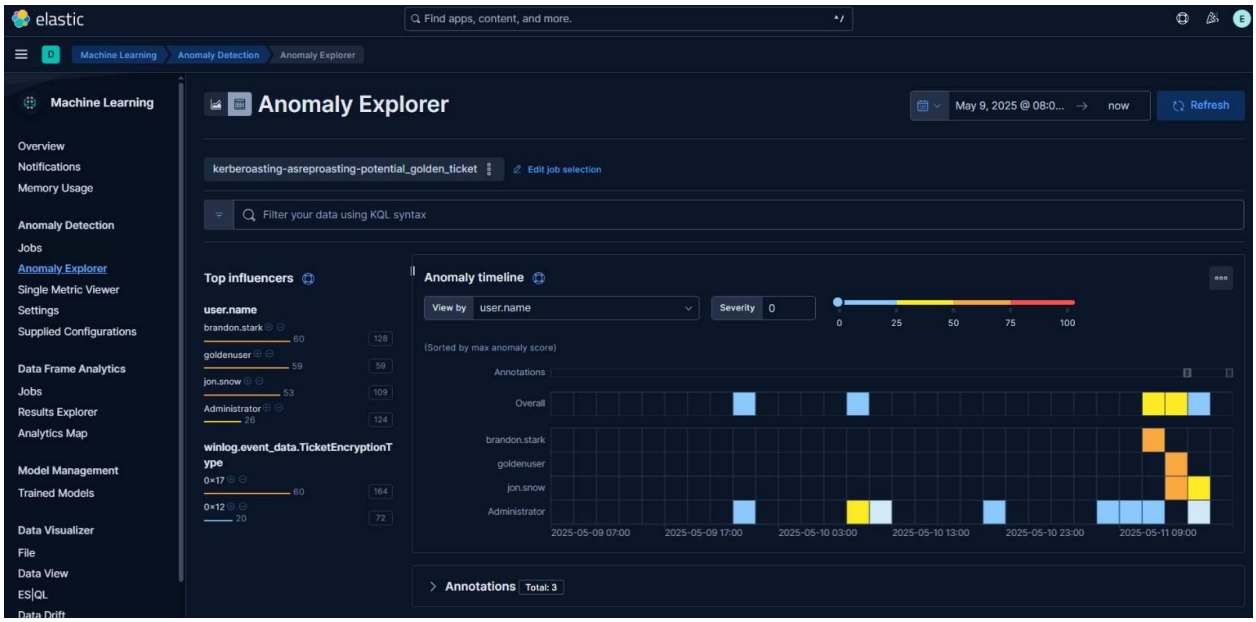


Рисунок 4.21 – виявлення аномалій в Anomaly Explorer

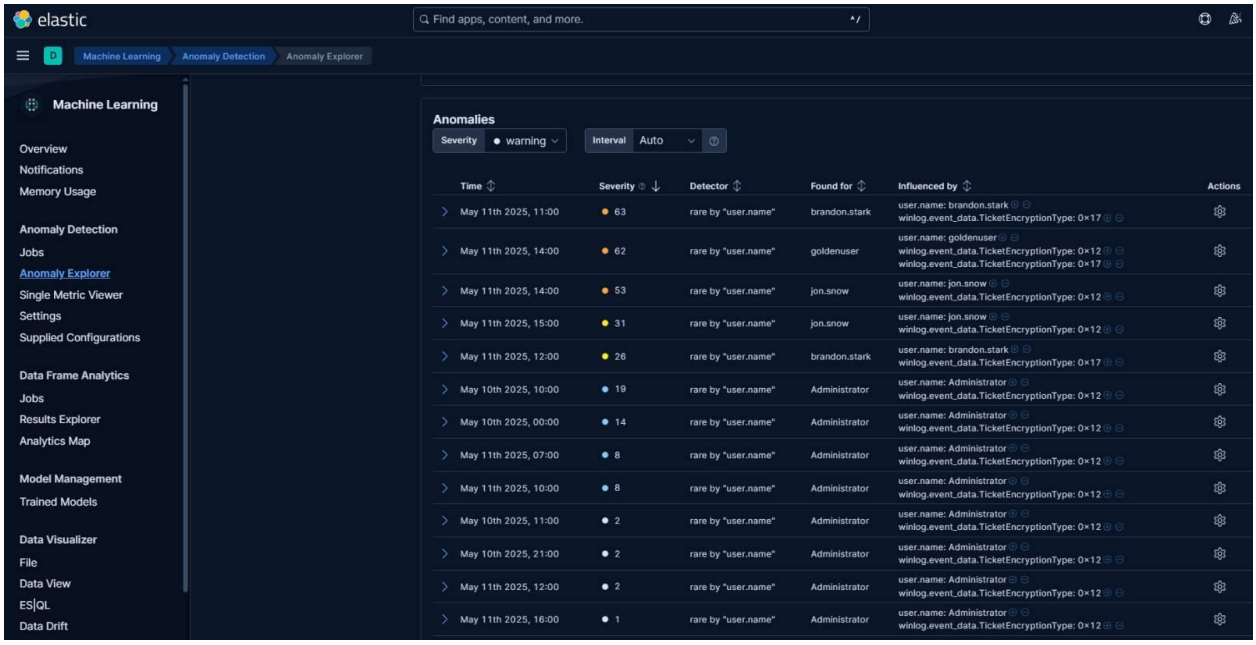


Рисунок 4.22 – підтвердження виявлення атак

Якщо ми розглянемо детальніше спрацювання, то параметр Probability що відображає ступінь впевненості системи у тому, що подія є аномальною або належить до певного класу значення 0.00116 означає, що ймовірність того, що подія нормальна — лише 0.116%. Відповідно, подія майже з 99.884% імовірністю вважається аномальною. Також варто звернути увагу, що при здійсненні цієї атаки використовується 2 різних типи шифруванн, а саме 0x12 та 0x17, це перехід також є аномальним явищем, було було вдало виявлено та подано як маркер аномальності. І в цьому контексті фактично отримано механізм виявлення найскладнішої атаки, оскільки вона передбачає наявності у зловмисника хешу паролю облікового запису krbtgt. Детектування атаки Golden Ticket наведено на рисунку 4.23.

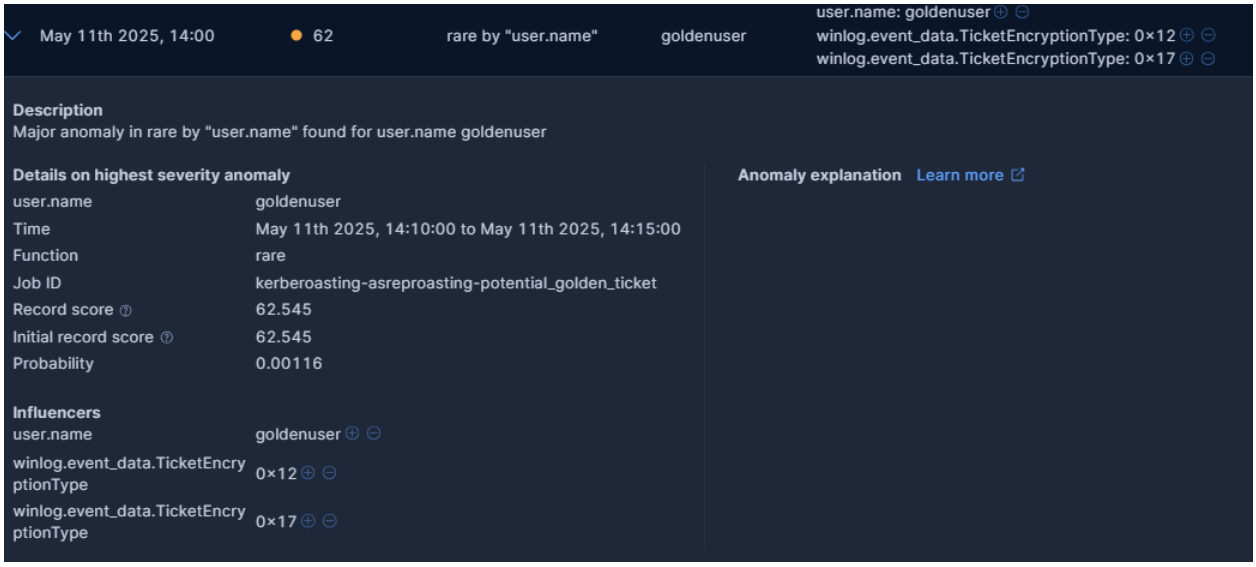


Рисунок 4.23 – виявлення атаки Golden Ticket

Виявлення просто аномальних аутентифікацій з використанням протоколу Kerberos наведено на рисунку 4.24.

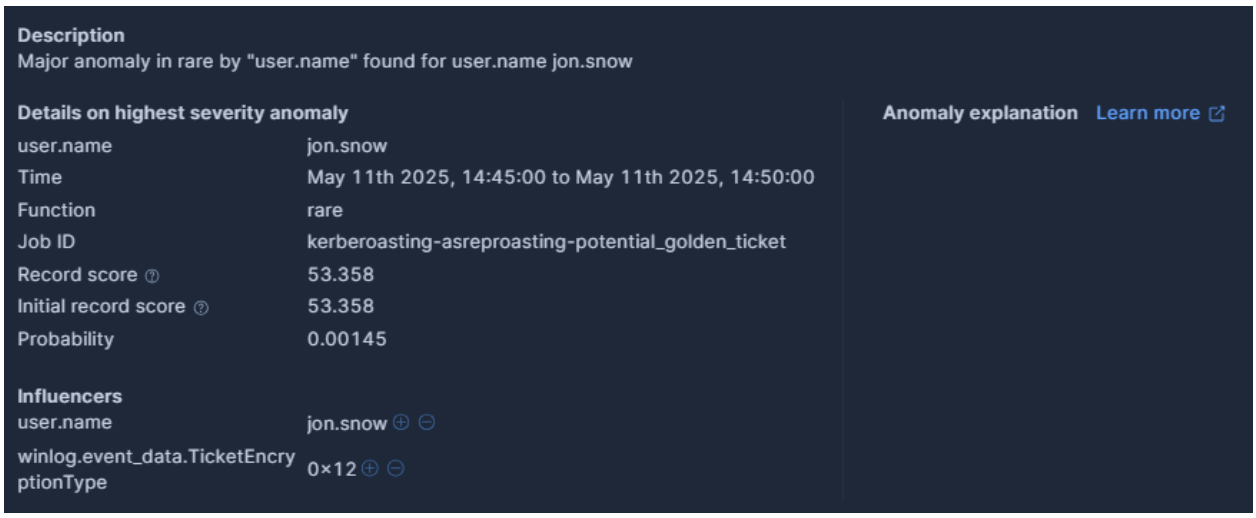


Рисунок 4.24 –виявлення аутентифікації користувача jon.snow при запуску Bloodhound-python

Також окермо варто відзначити детектування атаки Kerberoasting, оскільки вона на відміну від інших детектує аномально велику кількість запитів в рамках одного моітору, тобто підтверджується виявлення аномалій всіх типів. Підтвердження виявлення наведено на рисунку 4.25

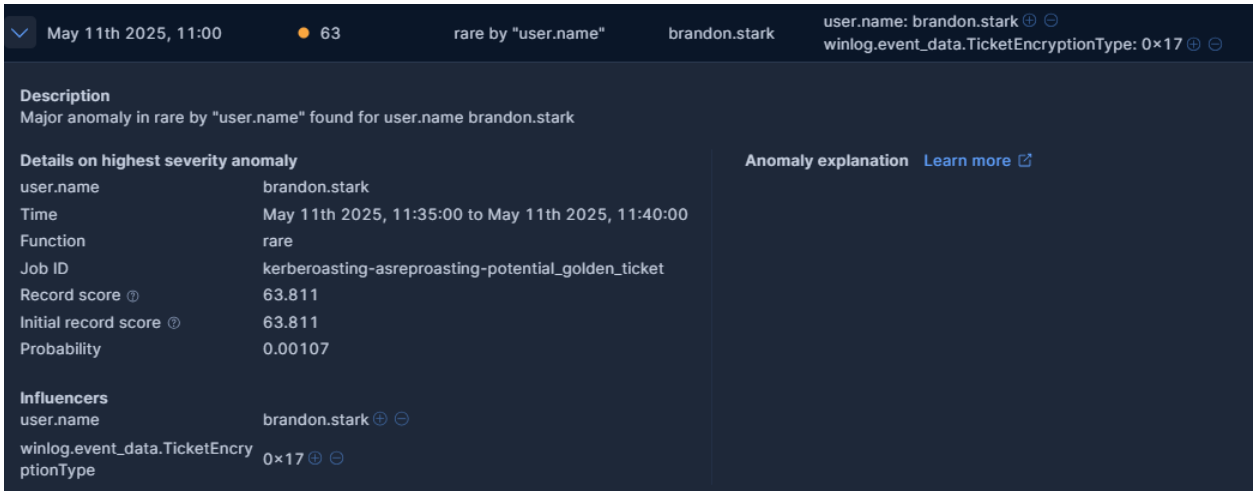


Рисунок 4.25 –виявлення аутентифікації користувача brandon.stark при атаці Kerberoasting

Конфігураційний файл моделі наведено в додатку А.

Висновки за розділом 4

У четвертому розділі було зосереджено увагу на застосуванні модулів машинного навчання Elastic Stack для виявлення атак у середовищі Active Directory. Основною метою стало тестування поведінкового аналізу у устворено лабораторному середовищі, побудованому на базі GOAD v3.

Для проведення тестування виявлення було проведено емуляцію типових атак на основі протоколу Kerberos, зокрема:

- Kerberoasting — шляхом запиту TGS-квитків для сервісних облікових записів з SPN-атрибутом;
- Golden Ticket — через експорт NTLM-хешу облікового запису krbtgt і подальше створення підроблених TGT-квитків;
- Аномальні аутентифікації Kerberos — моделювання підключень до ресурів AD, а саме LDAP сервера з використанням протоколу Kerberos.

Ключовою особливістю методики стало те, що всі атаки виконувались ззовні домену, без доставки шкідливого ПЗ на внутрішні хости. Це дозволило моделювати реалістичний сценарій обходу антивірусів, EDR та інших хостових засобів захисту.

Паралельно було створено та налаштовано Elastic ML Anomaly Detection Job, що виявляє відхилення у поведінці автентифікації на основі подій 4769 (TGS-запити). Модель побудована з використанням функції rare по полю user.name, що дозволяє виявляти нестандартну активність від рідкісних користувачів. Зафіксовані аномалії супроводжувались низькими значеннями ймовірності (наприклад, probability: 0.00116), що свідчить про високу достовірність результатів моделі.

Також виявлено, що Elastic Stack ефективно агрегує, обробляє та корелює події на основі логів Windows Security, і дозволяє реалізувати механізм поведінкового аналізу без необхідності створення класичних сигнатур або IOC.

Таким чином, у межах розділу:

- було реалізовано повний цикл атак на Active Directory;

- продемонстровано реальну здатність Elastic ML Job виявляти аномалії без попереднього визначення правил;
- підтверджено доцільність побудови моделей машинного навчання для детектування прихованих кіберзагроз у середовищі, де використовуються легітимні інструменти й протоколи.

Отримані результати демонструють ефективність поєднання методів поведінкового аналізу та машинного навчання для проактивного моніторингу та виявлення загроз, що актуально для сучасних корпоративних IT-інфраструктур.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи «Модель проактивного виявлення та запобігання кіберзагроз на основі машинного навчання» було реалізовано повний цикл побудови системи поведінкового аналізу, орієнтованої на виявлення складних атак у середовищі Active Directory, що здійснюються без доставки шкідливого програмного забезпечення нп хости, що знаходяться під управлінням Active Directory.

У першому розділі проведено комплексний аналіз Active Directory як ключового компонента сучасної корпоративної IT-інфраструктури. Розглянуто його архітектуру, основні сервіси, історію розвитку та вимоги в контексті кібербезпеки. Особливу увагу приділено тактикам і технікам зловмисників, класифікованих у базі знань MITRE ATT&CK, що використовують легітимні функціонування AD та вбудованих утиліт системи.

У другому розділі описано особливості виявлення аномальної активності в середовищі AD. Визначено джерела подій, необхідні для поведінкового аналізу: Kerberos, LDAP, доступ до об’єктів каталогу. Налаштовано розширене логування та реалізовано систему централізованого збору та обробки даних на базі Elastic Stack, включно з попередньою обробкою, нормалізацією, візуалізацією та інтеграцією з MITRE ATT&CK. Продемонстровано значення централізації логів для точного аналізу поведінки.

У третьому розділі розгорнуто лабораторне середовище на базі GOAD v3 у віртуалізаційній системі Proxmox. За допомогою інструментів автоматизації (Ansible, Packer, Terraform) реалізовано повноцінну AD-інфраструктуру з двома лісами, декількома доменами та спеціально змодельованими вразливостями. Це середовище дозволило моделювати реальні сценарії атак і протестувати механізми виявлення.

У четвертому розділі здійснено емуляцію атак, що охоплюють Kerberoasting, Golden Ticket та аномальні запити до Kerberos. Усі атаки виконано з недоменної машини Kali Linux, що виключає можливість їх виявлення класичними засобами EDR/AV. На основі зібраних подій побудовано модель машинного навчання в Elastic Stack з використанням модуля Anomaly Detection. Налаштовано Job, який виявляє рідкісні патерни поведінки, зокрема незвичні аутентифікації, що є маркерами компрометації.

Таким чином, у процесі дослідження було:

- Проаналізовано сучасні підходи до виявлення кіберзагроз у середовищі Active Directory.
- Досліджено техніки атак, які не потребують доставки шкідливого ПЗ.
- Побудовано лабораторне середовище, яке моделює реальні мережеві умови.
- Реалізовано централізовану систему збору, аналізу та візуалізації подій.
- Розроблено та протестовано поведінкову модель виявлення атак на основі Elastic ML.

Результати підтвердили, що використання поведінкового аналізу та машинного навчання у поєднанні з гнучкими можливостями Elastic Stack дозволяє ефективно виявляти приховані загрози в AD, які залишаються невидимими для класичних сигнатурних рішень. Створена модель є адаптивною, масштабованою та придатною до впровадження в реальному корпоративному середовищі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Microsoft. Active Directory Domain Services Overview [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/en-us/windows-server

2. MITRE ATT&CK® Navigator [Електронний ресурс]. – Режим доступу: https://attack.mitre.org

3. Event ID 4769: A Kerberos service ticket was requested [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4769

4. Elastic Security documentation [Електронний ресурс]. – Режим доступу: https://www.elastic.co/guide/en/security

5. Proxmox Virtual Environment [Електронний ресурс]. – Режим доступу: https://www.proxmox.com

6. GOAD – AD Lab [Електронний ресурс]. – Режим доступу: https://orange-cyberdefense.github.io/GOAD

7. BloodHound GitHub repository [Електронний ресурс]. – Режим доступу: https://github.com/BloodHoundAD/BloodHound

8. Impacket by SecureAuth Corporation [Електронний ресурс]. – Режим доступу: https://github.com/fortra/impacket

9. Ansible Documentation [Електронний ресурс]. – Режим доступу: https://docs.ansible.com

10. Sysmon – System Monitor [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon

11. Windows Security Auditing Events [Електронний ресурс]. – Режим доступу: https://www.ultimatewindowssecurity.com

12. Kerberos Protocol Extensions [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-kile

13. Microsoft Defender for Identity [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/en-us/defender-for-identity

14. What is EDR? Endpoint Detection and Response [Електронний ресурс]. – Режим доступу: https://www.paloaltonetworks.com

15. Elastic Machine Learning Documentation [Електронний ресурс]. – Режим доступу: https://www.elastic.co/guide/en/machine-learning

16. OWASP. Authentication Cheat Sheet [Електронний ресурс]. – Режим доступу: https://cheatsheetseries.owasp.org

17. Windows Security Monitoring [Електронний ресурс]. – Режим доступу: https://www.sans.org/white-papers/36872

18. What is LDAP and how does it work? [Електронний ресурс]. – Режим доступу: https://www.varonis.com/blog/ldap

19. Wazuh SIEM Documentation [Електронний ресурс]. – Режим доступу: https://documentation.wazuh.com

20. Goseva K., Mitev G. Advanced methods for detecting credential attacks in AD environments // Journal of Cybersecurity, 2021. – №2. – С. 34–41.

22. Mimikatz Documentation [Електронний ресурс]. – Режим доступу: https://github.com/gentilkiwi/mimikatz

23. Packer by HashiCorp [Електронний ресурс]. – Режим доступу: https://www.packer.io

24. Terraform Registry [Електронний ресурс]. – Режим доступу: https://registry.terraform.io

25. Elastic Search Logstash Pipelines [Електронний ресурс]. – Режим доступу: https://www.elastic.co/guide/en/logstash

26. Syslog Protocol Specification [Електронний ресурс]. – Режим доступу: https://tools.ietf.org/html/rfc5424

28. Threat hunting with Windows Event Logs [Електронний ресурс]. – Режим доступу: https://www.splunk.com

29. Detect Kerberos abuse using anomaly detection // Elastic Blog [Електронний ресурс]. – Режим доступу: https://www.elastic.co/blog

30. Elastic SIEM Use Cases [Електронний ресурс]. – Режим доступу: https://www.elastic.co/security

31. Mayfly’s GOAD setup guide [Електронний ресурс]. – Режим доступу: https://mayfly277.github.io/categories/goad

33. Rubeus GitHub project [Електронний ресурс]. – Режим доступу: https://github.com/GhostPack/Rubeus

35. Certipy for AD CS abuse detection [Електронний ресурс]. – Режим доступу: https://github.com/ly4k/Certipy

36. Elastic Security – Detection Rules [Електронний ресурс]. – Режим доступу: https://github.com/elastic/detection-rules

37. HAFNIUM and Exchange Server Exploitation – Microsoft Threat Intel [Електронний ресурс]. – Режим доступу: https://www.microsoft.com/security/blog

38. AD Attack Paths – CyberArk [Електронний ресурс]. – Режим доступу: https://www.cyberark.com/resources/threat-research

40. Tactics, Techniques, and Procedures (TTPs) [Електронний ресурс]. – Режим доступу: https://www.cisa.gov/resources-tools

ДОДАТКИ

ДОДАТОК А
КОНФІГУРАЦІЙНИЙ ФАЙЛ МОДЕЛІ ВИЯВЛЕННЯ АТАК НА KERBEROS


```
"job_id": "kerberoasting-asreproasting-potential_golden_ticket",
 "job_type": "anomaly_detector",
 "job_version": "12.0.0",
 "create_time": 1746964376547,
 "model_snapshot_id": "1746964388",
 "datafeed_config": {
  "datafeed_id": "datafeed-kerberoasting-asreproasting-
potential_golden_ticket",
  "job_id": "kerberoasting-asreproasting-potential_golden_ticket",
  "authorization": {
   "roles": [
    "superuser"
   ]
  },
  "query_delay": "103416ms",
  "chunking_config": {
   "mode": "auto"
  },
  "indices_options": {
   "ignore_unavailable": false,
   "expand_wildcards": [
    "open"
```

```
  ],
  "allow_no_indices": true,
  "ignore_throttled": true
},
"query": {
  "bool": {
    "must_not": [
      {
        "bool": {
          "should": [
            {
              "wildcard": {
                "user.name": {
                  "value": "*$"
                }
              }
            }
          ],
          "minimum_should_match": 1
        }
      }
    ],
    "filter": [
      {
        "match_phrase": {
          "event.code": "4769"
        }
      },
      {
        "match_phrase": {
```

```
          "winlog.event_data.Status": "0x0"
        }
      }
    ]
  }
},
"indices": [
  "logs-*"
],
"scroll_size": 1000,
"delayed_data_check_config": {
  "enabled": true
},
"state": "started",
"node": {
  "id": "p9MTrzrDRK2baQtNQ-lSwQ",
  "name": "elastic",
  "ephemeral_id": "TcPFa0EcTqG0wiraCBK2mg",
  "transport_address": "127.0.0.1:9300",
  "attributes": {
    "ml.config_version": "12.0.0",
    "ml.max_jvm_size": "4294967296",
    "ml.allocated_processors_double": "4.0",
    "ml.allocated_processors": "4",
    "ml.machine_memory": "8326430720"
  }
},
"assignment_explanation": "",
"timing_stats": {
  "job_id": "kerberoasting-asreproasting-potential_golden_ticket",
```

```
    "search_count": 114,
    "bucket_count": 565,
    "total_search_time_ms": 5259,
    "average_search_time_per_bucket_ms": 9.307964601769912,
    "exponential_average_search_time_per_hour_ms": 684.1328932163927
   },
   "running_state": {
    "real_time_configured": true,
    "real_time_running": true,
    "search_interval": {
     "start_ms": 1746969300577,
     "end_ms": 1746969450000
    }
   }
  },
  "description": "",
  "analysis_config": {
   "bucket_span": "5m",
   "detectors": [
    {
     "detector_description": "rare by \"user.name\"",
     "function": "rare",
     "by_field_name": "user.name",
     "detector_index": 0
    }
   ],
   "influencers": [
    "user.name",
    "winlog.event_data.TicketEncryptionType"
   ],
```

```
  "model_prune_window": "30d"
},
"analysis_limits": {
  "model_memory_limit": "11mb",
  "categorization_examples_limit": 4
},
"data_description": {
  "time_field": "@timestamp",
  "time_format": "epoch_ms"
},
"model_plot_config": {
  "enabled": false,
  "annotations_enabled": false
},
"model_snapshot_retention_days": 10,
"daily_model_snapshot_retention_after_days": 1,
"results_index_name": "shared",
"allow_lazy_open": false,
"data_counts": {
  "job_id": "kerberoasting-asreproasting-potential_golden_ticket",
  "processed_record_count": 14652,
  "processed_field_count": 29304,
  "input_bytes": 1496156,
  "input_field_count": 29304,
  "invalid_date_count": 0,
  "missing_field_count": 0,
  "out_of_order_timestamp_count": 0,
  "empty_bucket_count": 0,
  "sparse_bucket_count": 0,
  "bucket_count": 567,
```

```
  "earliest_record_timestamp": 1746799200743,
  "latest_record_timestamp": 1746969422137,
  "last_data_time": 1746969553552,
  "input_record_count": 14652,
  "log_time": 1746969553552,
  "latest_bucket_timestamp": 1746969300000
},
"model_size_stats": {
  "job_id": "kerberoasting-asreproasting-potential_golden_ticket",
  "result_type": "model_size_stats",
  "model_bytes": 30848,
  "peak_model_bytes": 43368,
  "model_bytes_exceeded": 0,
  "model_bytes_memory_limit": 11534336,
  "total_by_field_count": 8,
  "total_over_field_count": 0,
  "total_partition_field_count": 2,
  "bucket_allocation_failures_count": 0,
  "memory_status": "ok",
  "assignment_memory_basis": "current_model_bytes",
  "output_memory_allocator_bytes": 0,
  "categorized_doc_count": 0,
  "total_category_count": 0,
  "frequent_category_count": 0,
  "rare_category_count": 0,
  "dead_category_count": 0,
  "failed_category_count": 0,
  "categorization_status": "ok",
  "log_time": 1746969553566,
  "timestamp": 1746969300000
```

```
},
"forecasts_stats": {
  "total": 0,
  "forecasted_jobs": 0
},
"state": "opened",
"node": {
  "id": "p9MTrzrDRK2baQtNQ-lSwQ",
  "name": "elastic",
  "ephemeral_id": "TcPFa0EcTqG0wiraCBK2mg",
  "transport_address": "127.0.0.1:9300",
  "attributes": {
    "transform.config_version": "10.0.0",
    "xpack.installed": "true",
    "ml.config_version": "12.0.0",
    "ml.max_jvm_size": "4294967296",
    "ml.allocated_processors_double": "4.0",
    "ml.allocated_processors": "4",
    "ml.machine_memory": "8326430720"
  }
},
"assignment_explanation": "",
"open_time": "5252s",
"timing_stats": {
  "job_id": "kerberoasting-asreproasting-potential_golden_ticket",
  "bucket_count": 567,
  "total_bucket_processing_time_ms": 499.99999999999983,
  "minimum_bucket_processing_time_ms": 0,
  "maximum_bucket_processing_time_ms": 28,
  "average_bucket_processing_time_ms": 0.8818342151675482,
```

"exponential_average_bucket_processing_time_ms":
1.0674358923712404,
"exponential_average_bucket_processing_time_per_hour_ms":
16.480330069176233
}
}

ДОДАТОК Б
СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Тези-наукових конференцій

Нечипоренко І. П., Мирутенко Л. В. МОДЕЛЬ ПРОАКТИВНОГО ВИЯВЛЕННЯ ТА ЗАПОБІГАННЯ КІБЕРЗАГРОЗ НА ОСНОВІ МАШИННОГО НАВЧАННЯ. МЕТОДИ, ЗАСОБИ ТА ЗАХОДИ МЕНЕДЖМЕНТУ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ, ПРОБЛЕМИ КІБЕРБЕЗПЕКИ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМ. *PCSICS*, м. Київ, Україна. 2025. С. 117–121.