Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему *Метод виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання*

Галузь знань *12 – Інформаційні технології*
Шифр і назва галузі знань

Спеціальність *122 – Комп'ютерні науки*
Шифр і назва спеціальності

Освітня програма *Комп'ютерні науки*
Назва освітньої програми

Виконав: *студент 2 курсу, група КНм-23-2* _____ *Дітмар СЛОБОДЯН*
Курс, група виконавця — Підпис — Ім'я, прізвище

Керівник: *к.т.н., доцент кафедри КН* _____ *Олександр ПАСІЧНИК*
Науковий ступінь, посада — Підпис — Ім'я, прізвище

Нормоконтроль: *к.т.н., доцент кафедри КН* _____ *Руслан БАГРІЙ*
Науковий ступінь, посада — Підпис — Ім'я, прізвище

До захисту допускаю:
Зав. кафедри КН, д.т.н., професор _____ *Олександр БАРМАК*
Підпис — Ім'я, прізвище

18 грудня 2024 р.

Хмельницький 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет *інформаційних технологій*
Кафедра *комп'ютерних наук*
Освітній ступінь *магістр*
Галузь знань *12 – Інформаційні технології*
Спеціальність *122 – Комп'ютерні науки*

ЗАТВЕРДЖУЮ
Завідувач кафедри *комп'ютерних наук*

_____
(підпис)
д.т.н., професор Олександр БАРМАК
«*02*» *вересня* *2024* року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

1. Тема кваліфікаційної роботи магістра: *«Метод виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання»*

2. Завдання видано студенту *Дітмару СЛОБОДЯНУ*
Ім'я, прізвище

3. Керівник роботи *доцент кафедри КН Олександр ПАСІЧНИК*
Ім'я, прізвище

4. Затверджені наказом університету від «*26* « *серпня* *2024* р. № *60* .

5. Дата видачі завдання студенту: «*02* « *вересня* *2024* р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Метою кваліфікаційної роботи магістра є підвищення точності виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання. Досягнення мети роботи передбачає виконання таких задач: провести аналіз методів, алгоритмів та підходів машинного навчання до виявлення аномалій в Active Directory корпоративних інформаційних систем; спроєктувати метод виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання; виконати програмну реалізацію методу виявлення аномалій в Active Directory; провести експериментальне тестування реалізованого методу за наборами даних

Реферат

Кваліфікаційна робота магістра присвячена розв’язанню задачі виявлення аномалій в Active Directory засобами машинного навчання.

Актуальність теми. Active Directory є ключовою службою каталогів у середовищі Windows, що відповідає за автентифікацію, авторизацію та керування обліковими записами в корпоративних мережах. Водночас сервери та бази даних у домені Windows залишаються основними об’єктами атак, адже компрометація Active Directory відкриває шлях до критичної інформації та привілейованого доступу. Наявні підходи на основі статичних сигнатур або правил часто не встигають за динамічно змінюваними методами зловмисників та проявами нових кіберзагроз. Тому проєктування нового методу, що дасть можливість автоматично відстежувати аномалії в Active Directory та відповідно виявляти загрози на ранніх етапах, є особливо актуальною для безпеки серверних середовищ та баз даних.

Об’єкт дослідження – процес виявлення аномалій в Active Directory для захисту серверів та баз даних.

Предмет дослідження – методи, алгоритми та підходи машинного навчання до виявлення аномалій в Active Directory.

Мета і задачі роботи – підвищення точності виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання.

Досягнення мети роботи передбачає виконання таких задач:

1. Провести аналіз методів, алгоритмів та підходів машинного навчання до виявлення аномалій в Active Directory корпоративних інформаційних систем.
2. Спроєктувати метод виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання.
3. Виконати програмну реалізацію методу виявлення аномалій в Active Directory.
4. Провести експериментальне тестування реалізованого методу за наборами даних.

Методи дослідження. Для розв'язання поставлених задач у роботі використано методи модульного проєктування для створення методу з кількома алгоритмами, методи машинного навчання для аналізу подій в Active Directory, методи тестування на реальних і синтетичних даних для перевірки точності та повноти виявлення аномалій, методи статистичного аналізу для оцінювання результативності методу, а також методи симуляції аномалій для створення контрольованих сценаріїв зламу.

Наукова новизна одержаних результатів. Спроєктовано метод виявлення аномалій в Active Directory для захисту серверів та баз даних, який відрізняється від наявних підходів модульною архітектурою із залученням кількох алгоритмів машинного навчання на різних етапах атаки та використанням інтеграційного оброблення результатів виявлення, що дало змогу підвищити точність виявлення складних багатоступеневих атак та знизити кількість хибних спрацьовувань завдяки гнучкому аналізу сукупності аномальних подій у середовищі Windows.

Апробація результатів кваліфікаційної роботи магістра та публікації. Основні наукові та практичні результати пройшли апробацію на науково-практичній конференції – XVI Всеукраїнська науково-практична конференція "Актуальні проблеми комп'ютерних наук АПКН-2024", м. Хмельницький, ХНУ, 15–16 листопада 2024 р. (Слободян Д. А., Радюк П. М., Цивадиць П. О. Метод виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання. *Актуальні проблеми комп'ютерних наук АПКН-2024* : матеріали XVI Всеукр. науково-практ. конф., м. Хмельницький, 15–16 листоп. 2024 р. Хмельницький, 2024. С. 463–466. URL: https://elar.khmnu.edu.ua/handle/123456789/17152).

Структура та обсяг роботи. Кваліфікаційна робота магістра складається із завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань з 40 найменувань та 3 додатків. Загальний обсяг кваліфікаційної роботи складає 106 сторінок, з поміж яких 87 сторінок основного тексту та 19 сторінок додатків. У роботі наведено 17 рисунків та 2 таблиць.

Ключові слова: Active Directory, безпека серверів, виявлення аномалій, машинне навчання, Kerberoasting, Pass-the-Hash.

Зміст

Перелік скорочень

Скорочення, термін, позначення	Пояснення
AD	Active Directory
ADDS	Active Directory Domain Services
API	Application Programming Interface
CN	Common Name
CPU	Central Processing Unit
DC	Domain Controller
DN	Distinguished Name
DNS	Domain Name System
GNN	Graph Neural Network
GPU	Graphics Processing Unit
GUI	Graphical User Interface
IDS	Intrusion Detection System
IP	Internet Protocol
JSON	JavaScript Object Notation
LDAP	Lightweight Directory Access Protocol
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Multilayer Perceptron
OS	Operating System
OU	Organizational Unit
PtH	Pass the Hash
RAM	Random Access Memory
RDP	Remote Desktop Protocol
RF	Request Frequency
RNN	Recurrent Neural Network

SAM	Security Accounts Manager
SIEM	Security Information and Event Management
SMB	Server Message Block
SPN	Service Principal Name
SQL	Structured Query Language
SSH	Secure Shell
SVM	Support Vector Machine
TGS	Ticket Granting Service
UPN	User Principal Name

Вступ

Актуальність теми. Active Directory є ключовою службою каталогів у середовищі Windows, що відповідає за автентифікацію, авторизацію та керування обліковими записами в корпоративних мережах. Водночас сервери та бази даних у домені Windows залишаються основними об’єктами атак, адже компрометація Active Directory відкриває шлях до критичної інформації та привілейованого доступу. Наявні підходи на основі статичних сигнатур або правил часто не встигають за динамічно змінюваними методами зловмисників та проявами нових кіберзагроз. Тому проєктування нового методу, що дасть можливість автоматично відстежувати аномалії в Active Directory та відповідно виявляти загрози на ранніх етапах, є особливо актуальною для безпеки серверних середовищ та баз даних.

Об’єкт дослідження – процес виявлення аномалій в Active Directory для захисту серверів та баз даних.

Предмет дослідження – методи, алгоритми та підходи машинного навчання до виявлення аномалій в Active Directory.

Мета і задачі роботи – підвищення точності виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання.

Досягнення мети роботи передбачає виконання таких задач:

1. Провести аналіз методів, алгоритмів та підходів машинного навчання до виявлення аномалій в Active Directory корпоративних інформаційних систем.
2. Спроєктувати метод виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання.
3. Виконати програмну реалізацію методу виявлення аномалій в Active Directory.
4. Провести експериментальне тестування реалізованого методу за наборами даних.

Методи дослідження. Для розв’язання поставлених задач у роботі використано методи модульного проєктування для створення методу з кількома алгоритмами, методи машинного навчання для аналізу подій в Active Directory,

методи тестування на реальних і синтетичних даних для перевірки точності та повноти виявлення аномалій, методи статистичного аналізу для оцінювання результативності методу, а також методи симуляції аномалій для створення контрольованих сценаріїв зламу.

Наукова новизна одержаних результатів. Спроєктовано метод виявлення аномалій в Active Directory для захисту серверів та баз даних, який відрізняється від наявних підходів модульною архітектурою із залученням кількох алгоритмів машинного навчання на різних етапах атаки та використанням інтеграційного оброблення результатів виявлення, що дало змогу підвищити точність виявлення складних багатоступеневих атак та знизити кількість хибних спрацьовувань завдяки гнучкому аналізу сукупності аномальних подій у середовищі Windows.

Апробація результатів кваліфікаційної роботи магістра та публікації. Основні наукові та практичні результати пройшли апробацію на науково-практичній конференції – XVI Всеукраїнська науково-практична конференція “Актуальні проблеми комп’ютерних наук АПКН-2024”, м. Хмельницький, ХНУ, 15–16 листопада 2024 р. (Слободян Д. А., Радюк П. М., Цивадиць П. О. Метод виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання. *Актуальні проблеми комп’ютерних наук АПКН-2024* : матеріали XVI Всеукр. науково-практ. конф., м. Хмельницький, 15–16 листоп. 2024 р. Хмельницький, 2024. С. 463–466. URL: https://elar.khmnu.edu.ua/handle/123456789/17152).

Структура та обсяг роботи. Кваліфікаційна робота магістра складається із завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань з 40 найменувань та 3 додатків. Загальний обсяг кваліфікаційної роботи складає 106 сторінок, з поміж яких 87 сторінок основного тексту та 19 сторінок додатків. У роботі наведено 17 рисунків та 2 таблиць.

РОЗДІЛ 1 Аналітичний огляд виникнення аномалій в Active Directory для захисту серверів та баз

1.1 Огляд основних компонентів інтелектуальної служби Active Directory каталогів операційної системи Windows

Доменне середовище Microsoft є однією з найбільш критичних систем у корпоративній мережі. Реалізація служб каталогів домену Windows, відома як Active Directory (AD). Це найбільш використовувана реалізація служб каталогів, яка використовується в екосистемі Windows [2].

Microsoft AD – це система, яка зберігає і керує інформацією про об'єкти у внутрішній мережі, таких як користувачі, групи, комп'ютери, принтери, додатки та файли. Кожен об'єкт має свої особливі характеристики, дозволи та зв'язки з іншими об'єктами. AD організовує всі ці дані в ієрархічній структурі, що полегшує доступ до них для користувачів, забезпечуючи зручний спосіб керування доступом та безпекою в організації.

Служба каталогів AD – це спеціалізоване рішення для служби каталогів в мережевих операційних системах Microsoft. Мережеві операційні системи – це термін, що описує середовище, в якому зберігаються різноманітні ресурси, такі як облікові записи користувачів, груп та комп'ютерів, у централізованому сховищі. Служба каталогів, яка забезпечує доступ до цього сховища, називається Active Directory Domain Services (ADDS) [3].

Існує багато інших систем, що мають схожі характеристики з каталогами, наприклад, система доменних імен (DNS) або електронні поштові системи. Проте існують стандарти, що визначають, як повинна бути реалізована і доступна справжня служба каталогів, зокрема протокол X.500, а також його еволюція – полегшений протокол доступу до каталогів (LDAP). AD побудований на протоколі LDAPv3, який є оновленою версією LDAP, впровадженого у 1997 році. Перша версія Microsoft AD була випущена разом з Windows 2000 і з того часу є частиною операційних систем Windows Server [4].

Інформація, яка зберігається в AD, формує вигляд ієрархічної структури, відомої як схема AD [5]. Однак насправді ці дані зберігаються в плоскій базі даних, яка складається з рядків та стовпців. Ця база даних AD зберігається у файлі NTDS.dit (де dit – це інформаційне дерево каталогу) [6]. Кожен запис у базі даних AD є об'єктом. Структура AD включає два основних типи об'єктів: контейнери та неконтейнери (також називаються листовими об'єктами). Логічна структура AD ґрунтується на концепції доменів, які називаються доменами Windows або доменами AD. Домен містить логічні компоненти, що забезпечують досягнення адміністративних цілей організації, і встановлює межі безпеки для об'єктів, що знаходяться в ньому [7].

Домен є базовою одиницею, яка об'єднує об'єкти мережі та відповідає за контроль доступу до ресурсів. Кілька доменів можуть бути об'єднані в дерево доменів, а кілька дерев можуть формувати ліс. Ліс є найбільшою логічною одиницею AD, що об'єднує кілька доменів і визначає межі безпеки. Кожен ліс має єдину схему та глобальний каталог, які забезпечують цілісність даних. Відокремлення логічної структури від фізичної топології мережі підвищує гнучкість керування та зменшує адміністративні витрати [8]. Зв'язки між доменами, деревами, лісами та підрозділами зображені на рисунку 1.1.

В межах домену AD існує ще одна ієрархічна структура меншого масштабу. Об'єкти з подібними вимогами до безпеки та адміністрування зазвичай групуються в контейнери, відомі як організаційні підрозділи (рисунок 1.1). Ці підрозділи використовуються для формування ієрархії об'єктів у домені.

У багатодоменній структурі AD для пошуку об'єктів використовується глобальний каталог [9]. Це спеціальний контролер домену, який містить інформацію про всі об'єкти в межах лісу, але лише з обмеженим набором атрибутів. Дані глобального каталогу зберігаються на серверах, призначених для цієї ролі, і використовуються для оброблення міждоменних запитів.

Контролери домену, що виконують ролі майстра операцій, відповідають за виконання специфічних завдань для підтримки узгодженості даних та запобігання конфліктам у базі даних AD. У ADDS визначено п'ять ролей майстра операцій.

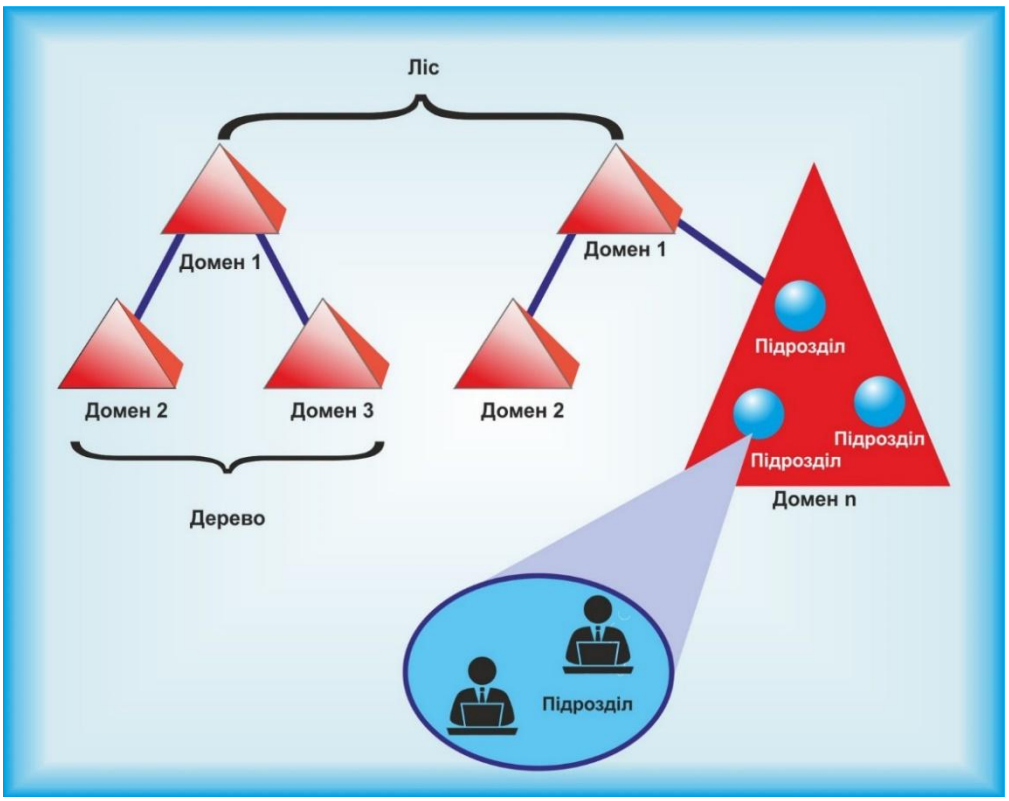


Рисунок 1.1 – Архітектура AD [1]

Функція виконання операцій єдиним головним комп'ютером забезпечує обробку запитів у випадках, коли реплікація за участю кількох головних комп'ютерів є неможливою або недопустимою. Для цього передбачено п'ять різних типів таких операцій [10].

Ролі, що виконуються на одному контролері домену в межах лісу [10]:

– майстер схеми;

– майстер доменних імен;

Ролі, що виконуються на одному контролері домену в межах домену:

– емулятор основного контролера домену (PDC);

– майстер інфраструктури;

– майстер відносного ідентифікатора (RID).

База даних ADDS використовує розширювану підсистему зберігання Microsoft Jet Blue, котра дає змогу для кожного контролера домена мати базу розміром до 16 терабайт і 1 мільярд об'єктів.

Базу ADDS можна поділити на три логічні сховища:

– схема – це шаблон, який визначає всі типи об’єктів, їх класи та атрибути, а також синтаксис атрибутів; всі дерева в межах одного лісу використовують спільну схему, що забезпечує єдину структуру;

– конфігурація – зберігає інформацію про структуру лісу та дерев AD, включаючи топологію реплікації між сайтами;

– домен – містить усі дані про об’єкти, створені в межах конкретного домену.

Кожен об’єкт в AD має своє розрізнювальне ім’я (DN). DN ідентифікує об’єкт на основі його повного шляху в AD. DN складається трьох типів атрибутів імен AD:

– domainComponent (DC) – це доменна частина;

– organisatinalUnitName (OU) – контейнер, в якому знаходиться об’єкт;

– commonName (CN) – позначає загальне ім’я.

У структурі, що зображено на рисунку 1.2, подано об’єкти певного університету.

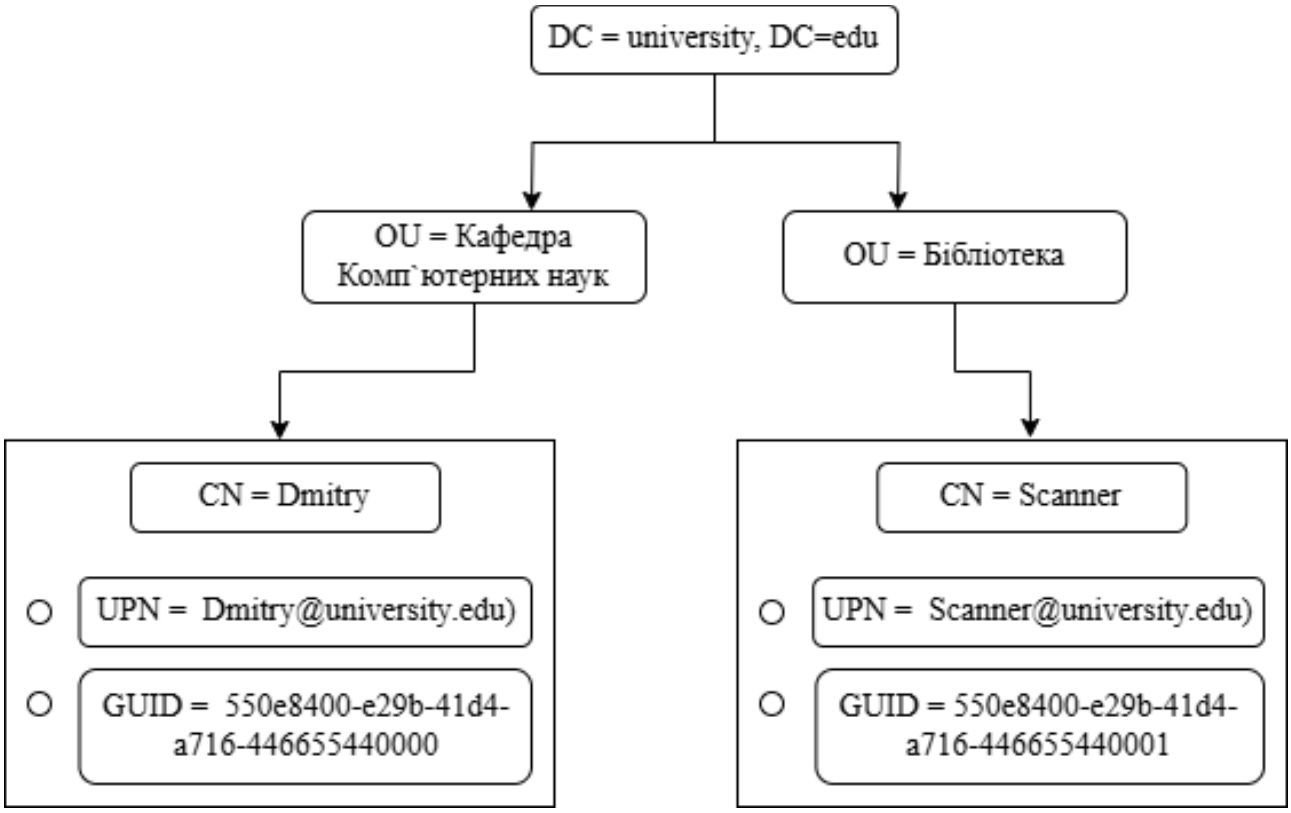


Рисунок 1.2 – Схема прикладу DN [10]

На верхньому рівні знаходиться DC – university.edu, потім OU, такі як «Бібліотека» та «Кафедра». В межах підрозділу «Кафедра» є об’єкт типу CN: Дмитро, а у підрозділі «Біліотека» – об’єкт «Сканер». Кожен об’єкт має власний DN, що може виглядати наприклад так: DN: CN = Dmitry, OU = Кафедра Комп’ютерних наук, DC=university, DC=edu.

У AD ім’я користувача (UPN) – це ім’я користувача та домен у форматі електронної адреси. У UPN ім’я користувача супроводжується роздільником «@», після якого йде домен активного каталогу [11]. Прикладом UPN є scanner@university.edu (рисунок 1.2).

Усі облікові записи користувачів AD повинні мати UPN. Неявне UPN генерується системою під час створення облікового запису, якщо адміністратор явно не створив UPN. Кожне UPN має бути унікальним у домені.

Ще один ідентифікатор для об’єктів є глобальний унікальний ідентифікатор, що є 128-бітним незмінним ідентифікатором, який використовується для пошуку та реплікації об’єктів в AD. Його значення зберігається у вигляді п’яти груп, що містять 8-4-4-4-12 шістнадцяткових цифр. (рис 1.2).

1.2 Процеси автентифікації в середовищі Windows

Операційні системи Windows передбачають обов’язкове використання дійсного облікового запису для здійснення входу в систему та отримання доступу до локальних чи мережевих ресурсів. Автентифікація визначається як процес підтвердження заявленої ідентичності користувача, тоді як авторизація полягає у встановленні прав доступу до певних ресурсів. AD виступає основною технологією для збереження даних про ідентифікацію в доменних середовищах, тож вона має прямий зв’язок із процесами автентифікації та авторизації.

Користувачі можуть проходити автентифікацію на комп’ютерах з операційною системою Windows через інтерактивний процес входу. Цей процес поділяється на три види:

– локальний вхід;

– вхід віддаленого доступу;

– під час входу в систему Windows користувачі можуть використовувати локальний або доменний обліковий запис.

Тип облікового запису визначає, як система підтверджує ідентифікацію користувача: через локальну базу даних безпеки комп'ютера (SAM) або через базу даних AD – NTDS.dit.

Локальний вхід забезпечує доступ до ресурсів самого комп'ютера та мережевих ресурсів, які цей комп'ютер надає. У свою чергу, доменний вхід дає змогу отримувати доступ як до локальних, так і до доменних ресурсів.

Для віддаленого доступу зазвичай використовуються стандартні мережеві порти [12]:

– 445 (SMB) – для протоколу Server Message Block, який забезпечує обмін файлами, доступ до принтерів і інші мережеві функції;

– 3389 (RDP) – для протоколу Remote Desktop Protocol, що надає віддалений доступ до робочого столу;

– 135 (RPC/DCOM) – для протоколу Remote Procedure Call, який використовується для зв'язку між клієнтом і сервером у мережі; цей порт забезпечує доступ до таких служб, як планувальники завдань, адміністрування та друк.

Протокол NT LAN Manager (NTLM) належить до сімейства протоколів автентифікації, вбудованих у Windows. Основним механізмом роботи NTLM є автентифікація на основі процесу запиту/відповіді [13]. Протокол NTLM застосовується для локальної автентифікації на системах, які не є частиною домену, або на пристроях, налаштованих як члени робочої групи.

Windows використовує пакет безпеки Negotiate для автоматичного вибору протоколу автентифікації. Якщо Kerberos доступний для обох систем, які беруть участь у процесі, він буде використовуватись як пріоритетний. У разі, якщо одна з систем не підтримує Kerberos, Negotiate автоматично переключається на NTLM [14].

На рисунку 1.3 зображено етапи автентифікації Kerberos, які відбуваються, коли користувач намагається підключитися до служби [15].

Kerberos дає змогу користувачам легко отримувати доступ до мережевих служб, просто повторно запитуючи сервісні квитки. Для цього клієнт повинен використати спеціальний ідентифікатор, який називаються ім’я принципала служби (SPN), який дає змогу точно визначити потрібну службу. SPN включає тип служби, ім’я хоста та номер порту, зберігаючись в AD. Для автентифікації кожна служба повинна мати хоча б один такий ідентифікатор, прив’язаний до її облікового запису. Прикладом SPN для серверу бази даних може виглядати так: SQLSvc/UnivDB01.campus.local:1433.

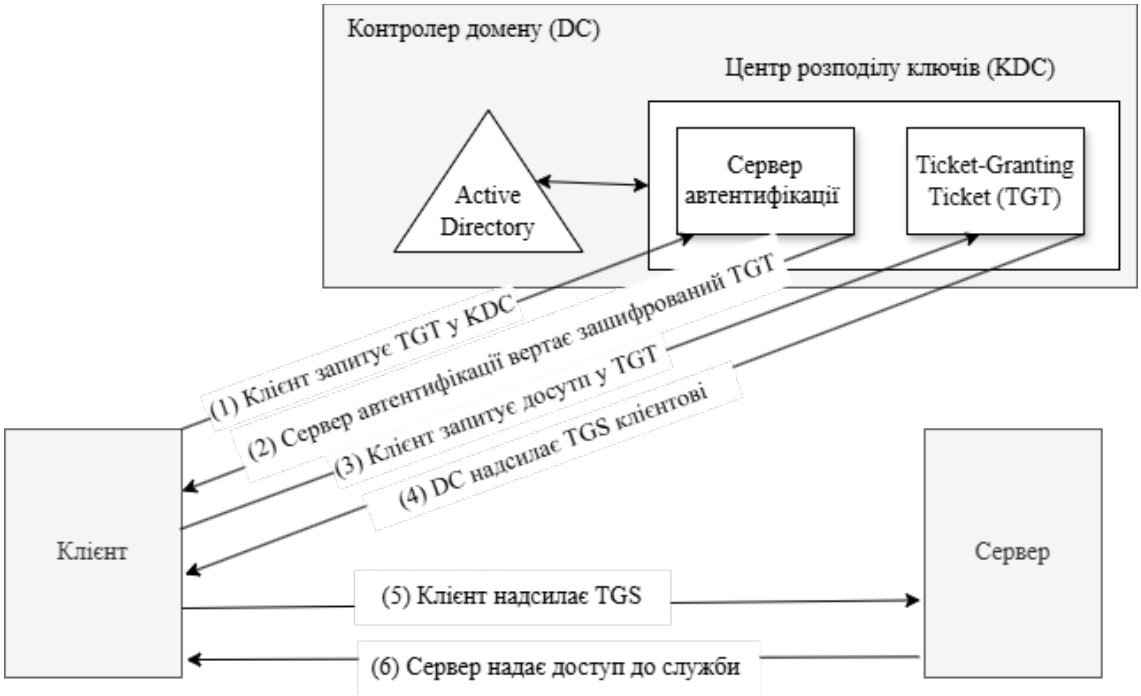


Рисунок 1.3 – Етапи автентифікації Kerberos [15]

Як було сказано раніше, база даних AD зберігається у контролері домену, а саме у файлі NTDS.dit, але у Windows є й інші місця, де можна знайти облікові дані. Основні сховища облікових даних у Windows:

– Cached Credentials;

– SAM;

– NTDS.dit.

Cached Credentials – містить кешовані облікові записи. Ці записи дають можливість користувачам входити до своїх облікових записів, навіть якщо немає

зв’язку з доменним контролером. У кеші зберігаються хешовані дані для автентифікації, що дає змогу забезпечувати доступ до системи в автономному режимі. Кількість таких кешованих записів контролюється політиками групи (GP).

SAM – є один із ключових баз даних, оскільки він зберігає локальні облікові записи користувачів і груп, їхні паролі та інші атрибути, визначені на конкретному комп’ютері. В DC SAM зберігає лише обліковий запис та пароль відновлення адміністратора, оскільки вся автентифікація доменних користувачів здійснюється через базу NTDS.dit, а SAM використовується лише для резервного доступу у разі збоїв [12].

База даних NTDS.git забезпечує автентифікації в доменних середовищах Windows, та є центральних сховищем AD. Цей файл містить інформацію про всі облікові записи домену, включно з користувачами, групами, комп’ютерами та службами. Також тут зберігаються хеші паролів для кожного облікового запису. Саме через цю базу проходить автентифікація користувачів і служб у доменному середовищі, що тісно пов’язано з протоколом Kerberos.

1.3 Служба каталогів Active Directory, як ціль кібератак

Служба AD є ключовою складовою інформаційної інфраструктури більшості організацій, забезпечуючи централізоване управління ідентифікацією, автентифікацією та авторизацією користувачів і ресурсів у корпоративних мережах. Завдяки своїй критичній ролі AD часто стає мішенню для кібератак, спрямованих на захоплення контролю над мережею, викрадення конфіденційних даних або забезпечення тривалого доступу до систем [16].

Зазвичай атака починається з компрометації одного пристрою в мережі [17]. Отримавши початковий доступ до облікового запису користувача або машини, зловмисник розпочинає нумерацію домену для збору інформації про структуру AD, облікові записи, групи та ресурси. На цьому етапі атакуючий може використовувати Brute Force атаки для підбору паролів слабко захищених облікових записів [18]. Недостатні політики блокування або використання простих паролів значно

полегшують виконання цієї атаки. Зібрана інформація дає змогу виявити вразливі місця, такі як слабкі облікові записи або сервісні облікові записи з надмірними привілеями. Це стає основою для подальшого розширення доступу до системи.

Отримавши початковий доступ, зловмисники прагнуть підвищити свої привілеї для доступу до чутливих даних або адміністрування системи. Основними цілями є:

– локальні облікові записи з адміністративними правами;

– сервісні облікові записи, які часто налаштовані з надмірними привілеями;

– облікові записи адміністраторів домену.

Одним із типових методів ескалації є Kerberoasting, який полягає у витягу квитків Kerberos для розшифрування паролів сервісних облікових записів.

Після отримання підвищених привілеїв зловмисники концентруються на закріпленні в системі, що дає змогу їм тривалий час залишатися непоміченими. Цей етап включає кілька ключових дій.

а) Бокове переміщення (Lateral Movement) [19]. Зловмисник пересувається мережею, використовуючи легітимні облікові дані, отримані раніше, для доступу до інших систем. Наприклад, через вразливості SMB або RDP зловмисник може досягти систем із вищим рівнем доступу.

б) Ескалація привілеїв та стійкість Suspicious Privilege Escalation (SPE) [20]. В межах атаки SPE зловмисник змінює налаштування групових політик, створює приховані облікові записи або додає себе до привілейованих груп. SSP Persistence є ще одним поширеним методом, що включає використання Security Support Provider для тривалого доступу до облікових даних користувачів навіть після зміни паролів.

Останній етап зосереджений на досягненні основної мети атаки. Залежно від мотивів зловмисників це може включати викрадення даних, підготовку до атак на відмову в обслуговуванні (DoS) або використання системи для власних цілей, наприклад, криптомайнінгу [21].

File Exfiltration via SMB є типовою атакою на цьому етапі [22]. Використовуючи SMB-протокол, зловмисники отримують доступ до конфіденційних файлів і баз даних. Завдяки маскуванню під легітимний трафік така діяльність часто

залишається непоміченою системами моніторингу. Послідовність цих кроків і їхні цілі ілюструються на рисунку 1.4.

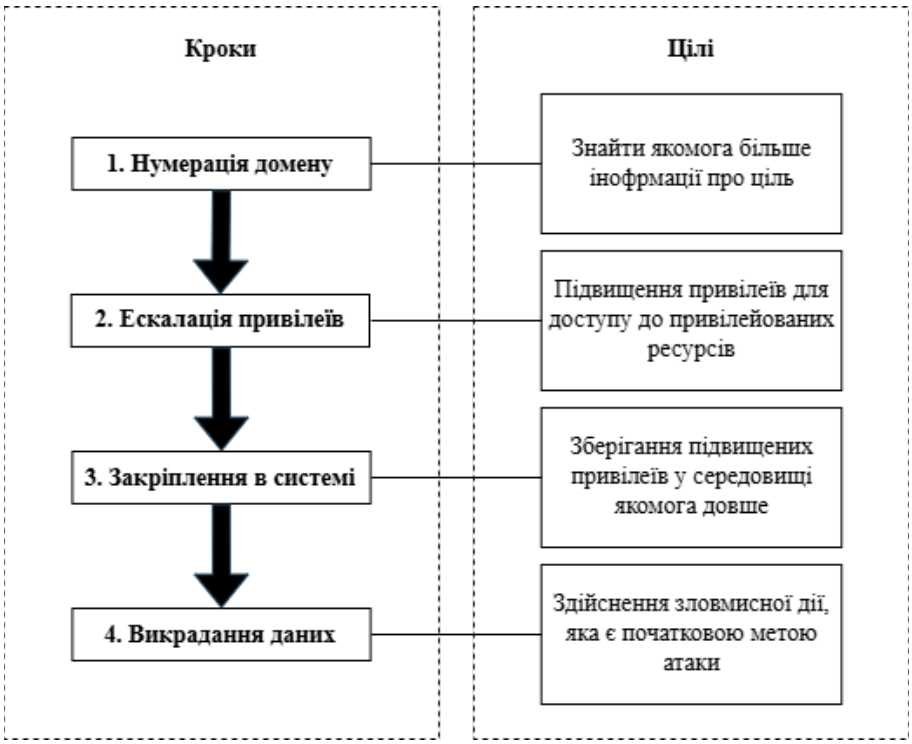


Рисунок 1.4 – Етапи атаки на AD [22]

Kerberoasting полягає у скануванні ідентифікатора SPN. Ці ідентифікатори використовуються для автентифікації клієнтів до відповідних служб через Kerberos. Зловмисник, який має доступ до дійсного облікового запису користувача, може перерахувати всі SPN, що зареєстровані в домені, за допомогою методу, який називається SPN Scanning. Цей процес дає змогу зловмиснику зібрати інформацію про всі служби, доступні в домені. Після чого зловмисник запитує TGS в DC, де частина цього квитка зашифрована хешем пароля облікового запису служби, до якої зловмисник хоче отримати доступ. Він може спробувати перебрати всі можливі варіанти паролів для зламу хешу, оскільки атака проводиться в офлайн режимі, без взаємодії з DC. Після успішного відновлення пароля зловмисник отримує доступ до облікового запису служби. У багатьох випадках такі облікові записи володіють значними привілеями у мережі, що відкриває можливості для бічного переміщення (lateral movement) або подальших дій із компрометації інфраструктури [23].

Атака Lateral Movement – це одна з ключових тактик кіберзловмисників, яка дає змогу їм поширювати доступ у внутрішній мережі організації після початкового проникнення. Метою цього процесу є отримання контролю над важливими ресурсами, такими як сервери баз даних, облікові записи з високими привілеями, або ж доступ до конфіденційних даних.

Яскравим прикладом складної атаки на AD є інцидент із українським оператором зв'язку «Київстар» у грудні 2023 року [24]. Російське хакерське угрупування «Солнцепек», за попередньою інформацією, отримало доступ до ключових вузлів IT-інфраструктури компанії. Основним методом стало використання облікового запису співробітника для проникнення у внутрішню мережу. Згодом зловмисники отримали доступ до AD, що дало змогу їм впливати на критичні сервіси, зокрема мобільний зв'язок і інтернет. Це призвело до перебоїв у роботі сервісів для 24 мільйонів користувачів [25].

Доступ до AD дав змогу зловмисникам маніпулювати доменними контролерами, системами резервного копіювання та хмарними ресурсами компанії. У рамках атаки знищено понад 10 тисяч комп'ютерів і 4 тисячі серверів. Такі дії стали можливими через недостатні заходи кіберзахисту та довготривале перебування хакерів у мережі з метою дослідження її структури та вразливостей.

1.4 Огляд наявних методів машинного навчання для виявлення аномалій в Active Directory

Наявні підходи до виявлення аномалій у середовищі AD можна поділити на дві категорії: методи, засновані на статичних правилах і сигнатурах, та методи, що використовують машинне навчання (ML) [26–28] для аналізу поведінки користувачів і систем [29].

Методи, засновані на сигнатурах, традиційно служать основою для виявлення аномалій у середовищі Windows. Вони покладаються на ідентифікацію специфічних ознак атак, таких як послідовності команд, що вказують на зловмисну активність, або поведінкові аномалії, які можна виявити за допомогою визначених правил [30].

Наприклад, у роботі [31] згадуються артефакти та сигнатури, які можна знайти в подіях безпеки Windows, для виявлення аномалій, пов’язаних із протоколом Kerberos, зокрема Kerberoasting. CERT-EU також опублікувала технічну документацію щодо захисту Golden Ticket через моніторинг підписів рядків у подіях та часу життя квитків Kerberos [32]. Проте, такі підходи мають обмеження, зокрема високу залежність від попереднього знання ознак атак і складність адаптації до нових загроз [33].

У відповідь на ці виклики у дослідженнях з’явилися підходи, що використовують методи ML для аналізу аномальної поведінки в AD. Наприклад, у роботі [34] запропоновано метод виявлення аномалій за допомогою ймовірнісних моделей, побудованих на основі ланцюгів Маркова, що дають можливість моделювати поведінку облікових записів для виявлення відхилень. Інший підхід, описаний у [35], ґрунтується на використанні алгоритму k-NN для виявлення аномальної активності у журналах автентифікації Windows. Ще один метод передбачає використання алгоритму One-Class Support Vector Machine (SVM) для аналізу подій Windows, зокрема тих, що стосуються адміністративних привілеїв, як-от події 4674 та 4688 [36]. Ці підходи дають можливість знизити залежність від сигнатур і створюють нові можливості для адаптації до раніше невідомих атак.

Окремі дослідження зосереджуються на аналізі атак, пов’язаних із бічним переміщенням у середовищі AD. У роботі [37] порівнюються різні класифікатори ML для виявлення таких атак, як Pass-the-Hash (PtH) [38], Pass-the-Ticket (PtT) [39] тощо. Інше дослідження [40] пропонує методи кластеризації для аналізу аномалій у середовищах Windows, які загалом показали результативність, хоча і з підвищеним рівнем хибнопозитивних результатів.

1.5 Висновок до розділу 1 та постановка задачі

У першому розділі детально проаналізовано механізми виникнення аномалій у середовищі AD та пояснено, чому AD є однією з найбільш вразливих і привабливих цілей для зловмисників. Основними перевагами аналізу є виявлення ключових проблемних точок у взаємодії облікових записів, серверів і сервісів Windows, а також

систематизація найпоширеніших тактик атак, таких як Kerberoasting, PtH, бічне lateral movement та підроблення квитків аутентифікації.

Встановлено, що традиційні захисні рішення, як от сигнатурні або евристичні методи, не здатні повноцінно відстежувати багатоступеневі та приховані загрози, які можуть імітувати звичну поведінку користувачів. Це робить їх малоефективними у випадках складних атак, які включають етапи ескалації привілеїв або переміщення в мережі. Аналіз підтвердив, що для вирішення цих проблем необхідно використовувати адаптивні алгоритми ML, які можуть виявляти приховані аномалії та адаптуватися до нових загроз.

Перевагою алгоритмів ML, які були використані в роботі, є їхня здатність моделювати складну поведінку зловмисників, наприклад, за допомогою GNN для аналізу графів взаємодії в мережі або LSTM для виявлення аномалій у часових послідовностях. Autoencoder показав високу точність у розпізнаванні відхилень у даних, характерних для підроблених квитків. Тому використання алгоритмів ML у виявленні аномалій в AD є перспективним напрямом, який, попри певні виклики, дає змогу суттєво підвищити рівень кіберзахисту корпоративних мереж.

Отже, метою цього дослідження є підвищення точності виявлення аномалій в AD засобами машинного навчання, що дасть змогу підвищити рівень кіберзахисту серверів та баз даних у корпоративних мережах Windows.

Для досягнення поставленої мети передбачено розв’язання таких завдань.

1. провести аналіз методів, алгоритмів та підходів машинного навчання до виявлення аномалій в AD корпоративних інформаційних систем.

2. Спроєктувати метод виявлення аномалій в AD для захисту серверів та баз даних засобами машинного навчання.

3. Виконати програмну реалізацію методу виявлення аномалій в AD.

4. Провести експериментальне тестування реалізованого методу за наборами даних.

РОЗДІЛ 2. Метод виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання

2.1. Проєктування методу виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання

У цьому розділі розглядаються основні аспекти задачі розробки методу виявлення аномалій у середовищі AD та вимоги до його реалізації. Середовище AD є складною структурою, яка використовується для управління ресурсами організацій. Через це воно часто стає ціллю атак, спрямованих на отримання доступу до даних, підвищення привілеїв або захоплення контролю над мережею. Етапи атак зазвичай включають такі дії, як доменне сканування, ескалація привілеїв, переміщення мережею та утримання доступу. Кожен з цих етапів вимагає специфічного підходу до аналізу, що зумовило модульний підхід до побудови системи.

Запропонований метод складається з трьох основних етапів, кожен з яких спеціалізується на виявленні аномалій на певному етапі атаки. Схема методу подана на рисунку 2.1.

Вхідні дані є такими.

1) Журнали подій Windows, зокрема Security Log (для LDAP), DNS Log (для DNS).

2) Записи, що містять:

– IP-адресу ініціатора запиту;

– ціль запиту (ім’я хоста або IP);

– час запиту (Timestamp);

– тип події («LDAP», «DNS» тощо).

Етап 1. Доменне сканування.

Крок 1.1. Збір даних про LDAP і DNS запити.

1) Система зчитує журнали подій Windows, фіксуючи всі звернення до служб LDAP та DNS.

2) Зібрані записи об’єднуються у формат, де для кожного запиту міститься:

– джерело (IP) і ціль (ім’я або IP хоста);

– час здійснення запиту;

– додаткові атрибути, як-от статус виконання, ім’я користувача тощо.

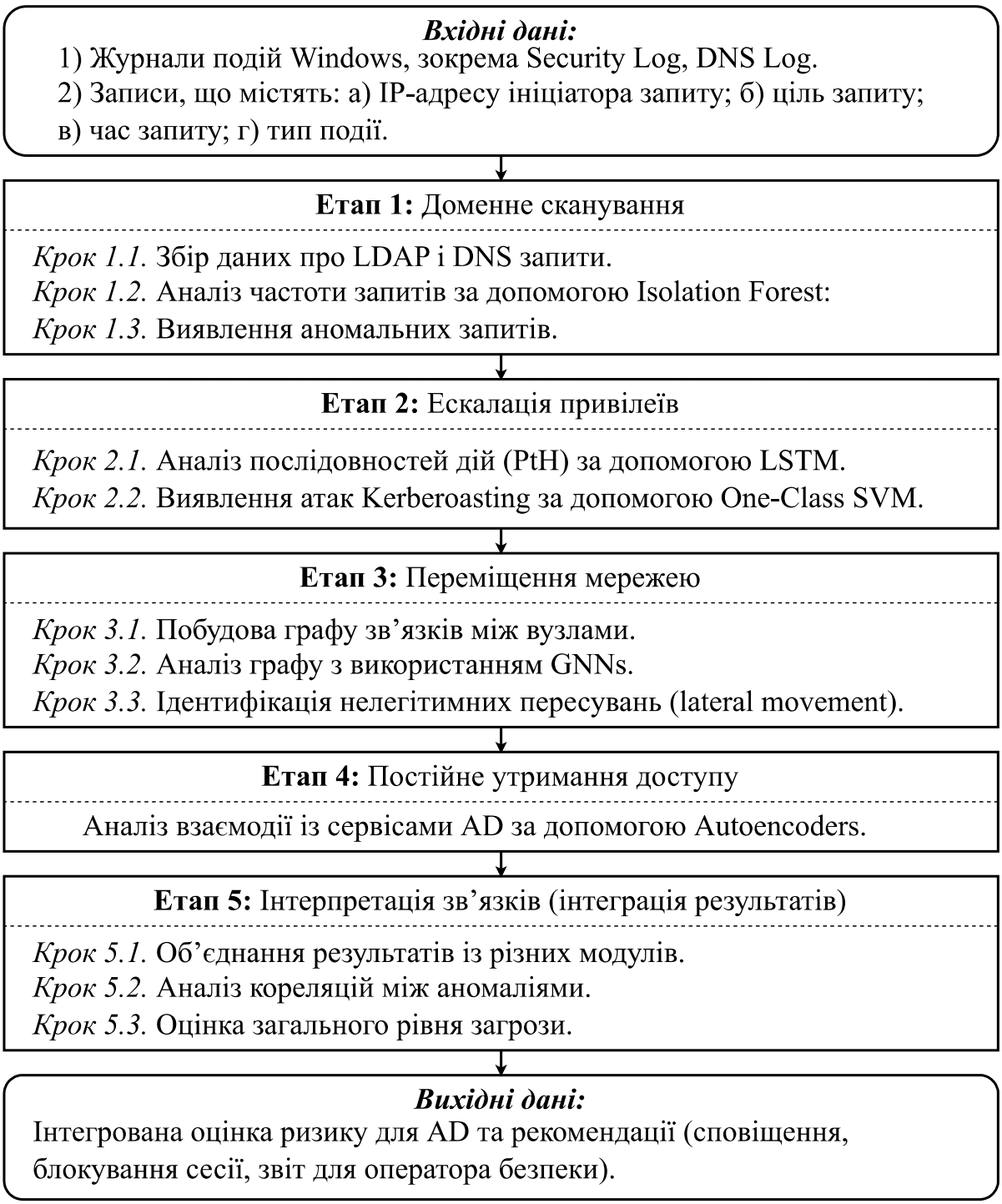


Рисунок 2.1 – Схема запропонованого методу виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання

Крок 1.2. Аналіз частоти запитів за допомогою Isolation Forest:

– на основі журналів формується вибірка показника Request Frequency – кількість запитів на одиницю часу;

– алгоритм Isolation Forest навчається на «звичайних» (типових) даних для виявлення відхилень від нормального рівня RF;

– результатом є модель, здатна присвоювати кожному новому запиту оцінку аномальності.

Крок 1.3. Виявлення аномальних запитів:

– модель Isolation Forest оцінює вхідні (нові) записи й визначає їх аномальність, беручи до уваги нетипову (завищену) частоту звернень;

– якщо відбувається аномально велика кількість LDAP або DNS-запитів із конкретної IP-адреси за короткий час, запити позначаються як підозрілі;

– вихідні дані етапу: перелік (або мітки) аномальних запитів, що потенційно свідчать про сканування домену.

Eman 2. Ескалація привілеїв.

Крок 2.1. Аналіз послідовностей дій (PtH) за допомогою Long Short-Term Memory (LSTM):

– визначаються часові ряди аутентифікацій для кожного користувача: послідовність логонів, їх інтервали та характеристики;

– модель LSTM навчається на «нормальних» патернах поведінки (звичайні послідовності в часі);

– під час аналізу нових послідовностей LSTM виявляє аномальні патерни, притаманні атакам PtH.

Крок 2.2. Виявлення аномалій Kerberoasting за допомогою One-Class SVM:

– система відокремлює події Kerberos (TGS) із журналів;

– модель One-Class SVM навчається на вибірці нормальних TGS-запитів (типова поведінка) і формує межу «нормального»;

– під час аналізу нових TGS-запитів модель виявляє ті, що мають аномальні ознаки (Brute Force-подібна активність), позначаючи їх як потенційні спроби Kerberoasting;

– вихідні дані етапу: а) мітки аномальних послідовностей (PtH), б) позначені підозрілі TGS-запити (Kerberoasting).

Етап 3. Переміщення мережею.

Крок 3.1. Побудова графу зв'язків між вузлами:

– кожен унікальний «об'єкт» (користувач, пристрій, сервіс) стає вузлом графа;

– події доступу чи автентифікації між вузлами формують ребра (user → server, user → service);

– дані структуруються так, щоб уможливити подальший аналіз.

Крок 3.2. Аналіз графу з використанням Graph Neural Networks (GNNs):

– модель GNN вивчає представлений граф, враховуючи ознаки вузлів (роль, IP, тип облікового запису) та ребер (тип доступу, час, частота);

– GNN дає змогу виявляти як локальні відхилення (несподівані з'єднання), так і глобальні зміни в топології (атипові маршрути, зміна патернів взаємодії).

Крок 3.3. Ідентифікація нелегітимних пересувань (lateral movement):

– якщо GNN виявляє ланцюги доступу, нетипові для конкретного користувача (або групи користувачів), ці зв'язки позначаються аномальними;

– вихідні дані етапу: список або показник аномальних переміщень у мережі (спроби доступу до незвичних вузлів, використання нетипових шляхів).

Етап 4. Постійне утримання доступу.

Крок 4.1. Аналіз взаємодії із сервісами AD за допомогою Autoencoders:

– модель Autoencoder навчається на «звичайних» шаблонах взаємодії із сервісами (типові квитки, нормальна тривалість сесій);

– за умови введення нових подій модель намагається їх «відтворити», і якщо різниця (помилка реконструкції) суттєва – вважається, що подія нетипова;

– до аномальних можуть належати випадки використання підроблених Golden Ticket чи Silver Ticket, коли механізми Kerberos працюють поза нормою;

– вихідні дані етапу: визначення потенційних сценаріїв постійного утримання доступу.

Етап 5. Інтерпретація зв'язків (інтеграція результатів).

Крок 5.1. Об'єднання результатів із різних модулів:

– “інтерпретатор зв’язків” отримує позначені аномалії від Isolation Forest (сканування), LSTM/One-Class SVM (ескалація привілеїв), GNN (lateral movement) та Autoencoder (утримання доступу);

– усі аномальні події складаються в узагальнену вибірку.

Крок 5.2. Аналіз кореляцій між аномаліями:

– аналізується хронологічна близькість (чи відбувалися кілька підозрілих подій за один проміжок часу);

– виявляються символічні збіги (один користувач, одна IP-адреса або спільний сервер);

– перевіряється, чи можуть аномалії бути складовими ланцюгової атаки (наприклад, спочатку сканування, потім ескалація привілеїв, далі lateral movement).

Крок 5.3. Оцінка загального рівня загрози:

– система підраховує сумарний ризик з урахуванням кількості виявлених аномалій, їх серйозності та взаємозв’язків;

– вихідні дані етапу: інтегрована оцінка ризику для AD та рекомендації (сповіщення, блокування сесії, звіт для оператора безпеки).

Взаємопов’язана послідовність виконання.

1) Збір логів (LDAP, DNS, Kerberos тощо) → передача даних на обробку Етапу 1 (Isolation Forest).

2) Одночасний або послідовний аналіз тих самих або додаткових логів:

– етап 2 (LSTM/One-Class SVM) фокусується на автентифікаційних послідовностях і Kerberoasting-активності;

– етап 3 (GNN) будує граф взаємодій для виявлення lateral movement;

– етап 4 (Autoencoder) виявляє спроби постійного утримання доступу.

3) Усі результати (позначені аномалії, їх оцінка) консолідуються в Етапі 5, де визначається чи є події елементами єдиної складної атаки або ж окремими спорадичними відхиленнями.

4) Фінально формується загальний висновок про рівень загрози та повідомлення для адміністраторів безпеки AD.

Завдяки такій п’ятиетапній (модульній) архітектурі запропонованого методу виявлення аномалій можливо виявляти аномалії на різних стадіях злому AD (починаючи зі сканування і закінчуючи довготривалою присутністю в мережі) та об’єднувати результати модулів у єдине уявлення про потенційні атаки. Це дає змогу збільшити точність та знизити кількість хибних спрацювань, а також масштабувати рішення, додаючи нові алгоритми ML чи додаткові джерела логів за потреби.

2.2. Побудова моделей машинного навчання для виявлення аномалій в Active Directory

2.2.1 Виявлення записів LDAP та DNS за допомогою Isolation Forest

Алгоритм Isolation Forest є одним з найефективніших методів для виявлення аномалій, зокрема в контексті великих обсягів даних, що генеруються в середовищі AD, таких як журнали подій. Основна ідея цього методу полягає у виявленні аномальних точок через ізоляцію, яка здійснюється за допомогою випадкових розподілів простору ознак. Такий підхід дає змогу швидко та точно ідентифікувати аномалії, не вимагаючи великих обчислювальних ресурсів, що є важливим аспектом при обробці великих наборів даних.

Основною особливістю алгоритму Isolation Forest є принцип ізоляції аномальних точок. Ідея цього механізму полягає в тому, що аномальні точки ізолюються швидше за нормальні. Причина цього полягає в тому, що аномальні точки зазвичай мають значення, які сильно відрізняються від більшості інших точок, що знаходяться в просторі ознак. Цей підхід дає змогу точно виділяти незвичайну активність, що відрізняється від типової поведінки.

Переваги використання Isolation Forest для доменного сканування є такими.

а) Результативність при великих обсягах даних: Завдяки своїй лінійній складності алгоритм може швидко створювати ізоляційні дерева для журналів із величезною кількість записів LDAP та DNS, які є основою автентифікаційних процесів всіх користувачів.

б) Автоматична ізоляція аномалій: Алгоритм ізолює аномальні точки швидше, ніж нормальні, що дає змогу легко ідентифікувати незвичайну активність у журналах подій. Де більшість даних є нормальними, а аномалії становлять малу частку.

в) Виявлення короткострокових аномалій: У випадку доменного сканування зловмисники можуть виконувати запити з високою частотою за короткий час. Isolation Forest дає змогу швидко визначати такі тимчасові піки аномальної активності, оскільки глибина ізоляції таких точок значно менша через їхню відмінність від нормального трафіку.

г) Гнучкість: Запити до LDAP та DNS можуть містити багато різних ознак. Isolation Forest дає змогу працювати з будь-якою кількістю ознак без необхідності попередньої підгонки до конкретного розподілу даних.

Щоб реалізувати зазначені переваги, необхідно зрозуміти основний механізм роботи Isolation Forest. Алгоритм ґрунтується на ідеї випадкових поділів простору даних для ізоляції точок. Саме цей підхід дає змогу точно виявляти аномалії у великих наборах даних.

Алгоритм побудови Isolation Forest полягає у створенні серії випадкових дерев. Кожне дерево будується шляхом випадкових поділів простору ознак, де кожен поділ намагається ізолювати певну точку даних. Для аномальних точок цей процес потребує меншої кількості поділів, оскільки вони відрізняються від інших точок у наборі даних. Отже, глибина ізоляції точки є мірою її аномальності: чим менше поділів потрібно для ізоляції точки, тим більш аномальною вона вважається.

Формально, глибина ізоляції для точки x_i визначається як $h(x_i)$, що є середнім значенням кількості поділів, необхідних для ізоляції цієї точки через кілька випадкових дерев:

$$h\left(x_i\right)=\frac{E\left(h\left(\left(x_i\right)\right)\right)}{c\left(n\right)}, \tag{2.1}$$

де $h(x_i)$ – глибина ізоляції для точки x_i, $E(h(x_i))$ – середнє значення глибини ізоляції по всіх деревах, $c(n)$ – нормалізаційна константа, що враховує розмір вибірки n.

Для виявлення аномалій використовується середнє значення $E[h(x)]$, яке порівнюється з певним порогом γ:

– якщо $E[h(x)] \leq \gamma$, точка x вважається нормальною;

– якщо $E[h(x)] > \gamma$, точка x позначається як аномальна

Поріг γ визначається через емпіричний аналіз або на основі заздалегідь заданого рівня аномальності в даних. Для великих наборів даних, де нормальні точки переважають, γ зазвичай вибирається так, щоб відсоток аномальних точок був невеликим – від 1% до 5%.

Точка x проходить через кілька дерев (Tree 1, Tree 2, ..., Tree T), побудованих випадковим чином, і для кожного дерева обчислюється її глибина ізоляції $h(x_i)$. Для кожної точки обчислюється середнє значення глибини ізоляції по всіх деревах, яке відображає її аномальність. Чим менше поділів потрібно для ізоляції точки, тим більш аномальною вона вважається.

Аномальні точки, завдяки своїм відмінностям, швидко ізолюються в дереві через меншу кількість поділів. У той час як нормальні точки мають більшу кількість поділів через їхню схожість з іншими точками. Тому аномалії зазвичай мають вищі значення $E(h(x_i))$.

У випадку доменного сканування алгоритм Isolation Forest буде використаний для виявлення активності, що відрізняється від типової поведінки в середовищі AD. Після навчання моделі на нормальних даних, алгоритм буде здатний ідентифікувати аномальні точки, що сигналізують про потенційну загрозу.

2.2.2 Виявлення Kerberoasting за допомогою One-Class SVM

Як було детально описано в п. 1.3, атака Kerberoasting полягає в отриманні сервісних квитків для слабких паролів сервісних акаунтів, з подальшим їх брутфорсуванням. Зловмисник використовує уразливості в механізмі Kerberos для запиту Service Tickets для акаунтів, що мають слабкі паролі, і згодом намагається зламати ці паролі, використовуючи брутфорс. Щоб точно виявити таку атаку, можна

застосувати метод One-Class SVM, який дає змогу виявляти аномалії в запитах до сервісів Kerberos, що є характерними для Kerberoasting.

One-Class SVM є спеціалізованою модифікацією класичного методу SVM, призначеною для роботи з однокласними даними, тобто даними, що відображають лише «нормальну» поведінку системи. Основна мета методу полягає в побудові гіперплощини, яка відокремлює нормальні дані від аномальних у багатовимірному просторі ознак.

Метод One-Class SVM виявляє аномалії, знаходячи межу (гіперплощину), яка максимально охоплює нормальні дані та мінімізує число хибно-позитивних виявлень. Для атаки Kerberoasting це означає, що модель тренується на нормальних запитах до сервісів Kerberos, а потім використовується для виявлення відхилень, характерних для аномальних запитів, що можуть свідчити про атаки.

Процес виявлення аномалій за допомогою One-Class SVM можна розподілити на кілька основних етапів.

1) Збір даних про «здорової» зони: На першому етапі збираються дані про типову поведінку користувачів у середовищі AD. Це мають бути запити до сервісу Kerberos, що не містять ознак аномалій. Ці дані використовуються для навчання моделі, яка окреслює межу нормальної поведінки.

2) Попередня обробка даних: Зібрані дані обробляються для відбору релевантних ознак. Цей етап також включає нормалізацію даних, щоб усі ознаки мали однаковий вплив на навчання моделі.

3) Навчання моделі на «здорових» даних: На основі попередньо оброблених даних модель One-Class SVM будує гіперплощину, яка максимально охоплює нормальні точки та мінімізує кількість аномальних точок. Це дає змогу створити межу, яка відображає «здорову зону».

4) Визначення аномальних запитів: Після навчання моделі надходять нові дані, а саме поточні запити до Kerberos, які необхідно перевірити на аномальність. Ці запити проходять обробку, після чого модель One-Class SVM класифікує їх на дві категорії:

Аномалія: Якщо запит виходить за межі визначеної гіперплощини.

Немає аномалії: Якщо запит відповідає типовій поведінці.

На рисунку 2.2 зображено графічне подання гіперплощини One-Class SVM.

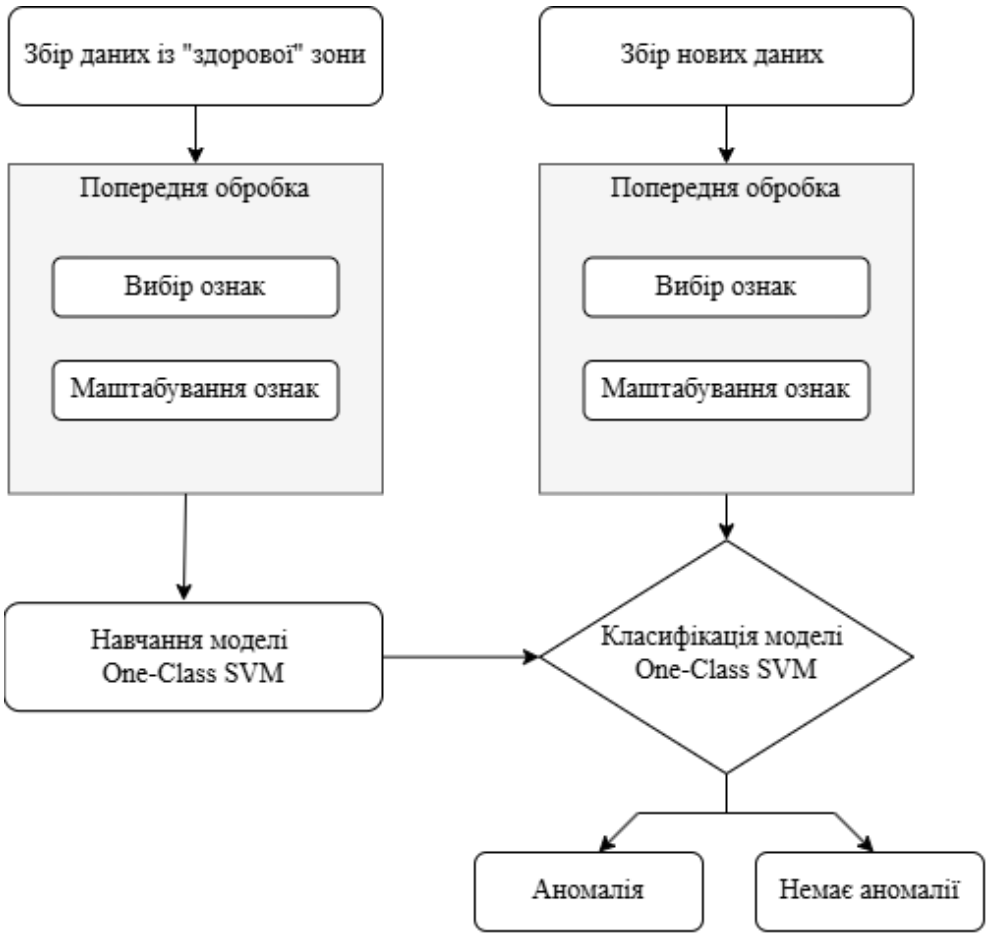


Рисунок 2.2 – Кроки процесу One-Class SVM для виявлення аномалії

Гіперплощина визначає межу між нормальними і аномальними точками (рисунок 2.3). Точки, що знаходяться всередині цієї межі, вважаються нормальними, тоді як ті, що поза межами, класифікуються як аномальні. Використання функції ядра дає змогу моделі працювати з більш складними, нелінійними даними, що є важливим для виявлення підозрілих запитів до Kerberos.

Побудова гіперплощини для моделі One-Class SVM, яка мінімізує кількість точок, що визначаються як аномальні, формалізується наступною оптимізаційною задачею:

$$\min \frac{1}{2} \| w \|^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \max\left(0, \rho - w^T \phi\left(x_i\right)\right) - \rho, \tag{2.2}$$

де w – вектор ваг, що визначає напрямок гіперплощини, $\phi(x_i)$ – функція ядра, що проєктує дані в простір більшої розмірності, ρ – параметр, що визначає кордон між нормальними і аномальними даними, v – параметр, який визначає частку допустимих аномалій у моделі, n – кількість точок у навчальній вибірці.

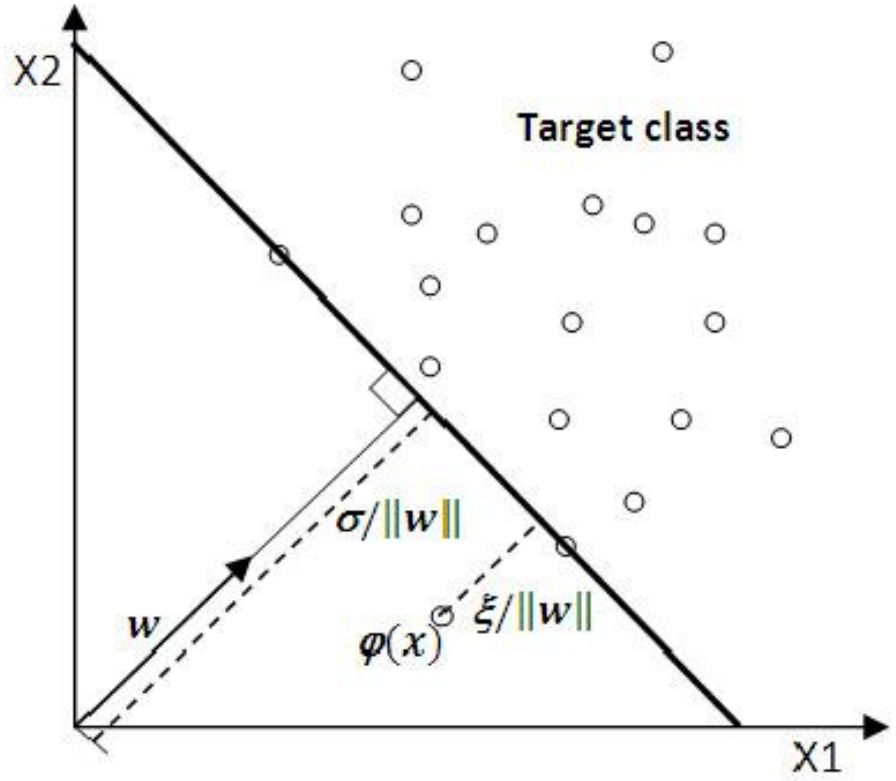


Рисунок 2.3 – Графік гіперплощини One-Class SVM

Функція ядра $\phi(x_i)$ дає змогу враховувати складні взаємозв'язки між ознаками, що є критично важливим для виявлення запитів до Kerberos із прихованими аномальними характеристиками. Це дає змогу One-Class SVM успішно ідентифікувати навіть ті аномалії, які можуть не проявлятися у лінійному просторі. Отже, модель є ефективним інструментом для виявлення аномалій типу Kerberoasting.

2.2.3 Виявлення Pass-the-Hash за допомогою Long Short-Term Memory

Метод LSTM є одним із підходів у глибинному навчанні, що ґрунтується на архітектурі глибокого навчання під назвою Recurrent Neural Network (RNN). Її

ключова особливість полягає у здатності точно працювати з послідовними даними та запам’ятовувати залежності на довгих часових відрізках. Це робить LSTM ідеальним інструментом для аналізу дій користувачів у середовищі AD, де поведінкові шаблони можуть змінюватися залежно від попередніх дій. Для виявлення аномалій PtH використання LSTM є доцільним, оскільки цей метод дає змогу виявити нетипові послідовності входів до системи, що є ключовим маркером такої атаки.

Попри звичайні RNNs, LSTM вирішує одну з основних проблем RNN – «зникнення градієнта». У класичних RNN під час тривалого навчання на довгих послідовностях значення градієнтів, необхідних для оновлення ваг, поступово зменшуються до майже нульових. Це ускладнює запам’ятовування важливих залежностей у віддалених у часі точках послідовності. LSTM, завдяки використанню спеціальної архітектури, яка включає комірки пам’яті та контрольні механізми «запам’ятовування» та «забування», дає змогу уникнути цієї проблеми. Отже, LSTM може точно навчатися на довгих послідовностях та зберігати релевантну інформацію для оброблення поточних входів.

Алгоритм LSTM був обраний для виявлення PtH через низку переваг.

а) Аналіз послідовності дій: Атака PtH передбачає використання крадених хешів паролів для автентифікації. Ці дії формують певні нетипові послідовності входів у систему. LSTM дає змогу аналізувати часові ряди запитів та виділяти аномальні послідовності.

б) Здатність працювати з довгими часовими залежностями: Атаки PtH можуть проявлятися у послідовностях запитів, які розтягнуті у часі. LSTM точно обробляє такі дані завдяки механізмам «запам’ятовування» та «забування», що забезпечують адаптацію до поведінкових шаблонів.

в) Гнучкість у налаштуванні: Модель LSTM дає змогу налаштовувати розмір пам’яті, кількість шарів та інші гіперпараметри, що забезпечує адаптацію до специфіки даних, таких як журнали аутентифікації у AD.

г) Підтримка роботи з великими наборами даних: LSTM точно працює з великими обсягами послідовних даних, що є важливим для оброблення великих журналів подій AD.

Виявлення PtH за допомогою LSTM ґрунтується на аналізі часових залежностей у запитах до системи. Завдяки своїй здатності розпізнавати складні послідовності дій, LSTM може навчитися нормальній поведінці користувачів у системі та виявляти аномальні патерни, характерні для атак PtH.

Розглянемо кожен з елементів архітектури LSTM і їх роль у виявленні аномалій, зокрема атак PtH.

1) Шар забування (Forget Gate): Цей шар вирішує, яку частину попередньої інформації з комірки пам’яті слід забути. В контексті атак PtH важливо видаляти з пам’яті запити, які не є частиною аномальних дій, наприклад, звичайні спроби доступу до системи. Це дає змогу моделі зосередитися лише на тих запитах, які можуть свідчити про спроби автентифікації з викраденими хешами паролів або інших підозрілих дій.

2) Шар введення (Input Gate): Шар введення контролює, яка нова інформація додається до пам’яті комірки. Для виявлення аномалій PtH цей шар оцінює нові запити і порівнює їх з попередніми даними. Якщо нові запити аномальні (наприклад, автентифікація з використанням викраденого хешу), ця інформація додається до пам’яті, що допомагає зберігати критичну інформацію для подальшого аналізу.

3) Шар комірки пам’яті (Memory Cell): Комірка пам’яті зберігає важливу інформацію для довготривалих залежностей. Вона допомагає зберігати дані про поведінку користувачів протягом часу, що важливо для виявлення аномалій PtH, які можуть проявлятися через зміни в поведінці користувачів у середовищі AD. Комірка пам’яті дає змогу зберігати контекст і виявляти послідовності дій, характерні для підозрілих атак.

4) Шар виходу (Output Gate): Шар виходу визначає, яку частину інформації з комірки пам’яті використовувати для поточного кроку. Для виявлення аномалій PtH цей шар аналізує зміни в звичайних шаблонах запитів користувачів. Якщо запити з незвичних IP-адрес або в незвичайний час відхиляються від нормальної поведінки, це сигналізує про потенційну атаку. В такому випадку модель генерує сигнал тривоги для подальшого розслідування.

2.2.4 Виявлення Lateral Movement за допомогою Graph Neural Networks

GNN є інноваційним підходом у глибинному навчанні, який спеціалізується на роботі з даними, що представлені у вигляді графів. У середовищі AD, де вузли відображають пристрої, користувачів або облікові записи, а ребра – взаємодії між ними, GNN демонструє високу результативність у виявленні аномалій у зв'язках. Атаки lateral movement характеризуються встановленням неочікуваних зв'язків між вузлами, що робить GNN ідеальним інструментом для ідентифікації таких аномалій.

Причини вибору GNN для аналізу Lateral movement є такими.

а) Інтеграція структури графу та атрибутів: Традиційні методи, як от Isolation Forest, працюють з структурованими даними та мають обмежену здатність враховувати залежності в графових структурах. Атаки lateral movement часто створюють аномальні зв'язки між вузлами, які зазвичай не взаємодіють. GNN дає змогу враховувати не лише атрибути вузлів, але й їх взаємодії через агрегацію інформації з сусідніх вузлів, що робить можливим більш точний аналіз, зважаючи на структуру графу та атрибути (наприклад, IP-адреси, часові мітки запитів). Це дає значну перевагу перед методами, які не враховують контекст взаємодій між вузлами.

б) Обробка складних і нелінійних залежностей: На відміну від RNN, які орієнтовані на послідовності даних, GNN здатні обробляти як локальні, так і глобальні залежності в мережі. Це важливо, оскільки атаки lateral movement часто проявляються у вигляді складних нелінійних взаємозв'язків. У GNN за допомогою агрегації інформації від сусідів (як видно в схемі) можна моделювати такі залежності, що дає змогу точно аналізувати навіть великі і заплутані графи з численними взаємодіями.

в) Гнучкість у роботі з графами: Мережі AD є динамічними, і зв'язки між вузлами змінюються з часом (нові сесії, атаки). GNN надає можливість оновлювати ваги та відображення вузлів відповідно до змін у графі, що дає змогу моделювати нові зловмисні активності. Це особливо важливо в контексті аналізу lateral movement, де зловмисники постійно змінюють свою тактику, і мережа повинна адаптуватися до нових ситуацій.

г) Адаптація до великомасштабних мереж: Методи, такі як CNN, працюють точно на зображеннях або з іншими типами структурованих даних, однак вони не масштабуються на великі графи. GNN, навпаки, оптимізовані для оброблення складних мережевих графів, де є тисячі вузлів і ребер. Це робить їх особливо підходящими для аналізу даних в масштабах великих доменів AD, де зв'язки і дані постійно зростають та змінюються.

GNNs працюють за ітеративним принципом, який складається з кількох основних етапів.

1. Графове подання даних: Мережеві дані перетворюються на граф, де вузли відображають об'єкти (користувачів або пристрої), а ребра – їхні взаємодії. Кожен вузол має набір атрибутів, що характеризують його (наприклад, IP-адреса, кількість запитів). Ребра також можуть бути атрибутовані, наприклад, часом встановлення з'єднання.

2. Агрегація сусідньої інформації: Для кожного вузла збирається інформація з його сусідів у графі. Цей процес дає змогу інтегрувати локальні залежності, використовуючи функції агрегації, такі як середнє значення, максимізація або зважені коефіцієнти на основі механізму уваги.

3. Оновлення вузлових атрибутів: Зібрана інформація використовується для оновлення атрибутів вузлів, що дає змогу враховувати як локальний контекст, так і глобальні залежності. Це оновлення виконується через параметризовані функції, наприклад, через Multilayer Perceptron (MLP).

4. Визначення аномальних зв'язків: Після кількох ітерацій агрегації та оновлення вузлових атрибутів модель класифікує зв'язки (ребра) як нормальні або аномальні. Аномальні зв'язки позначаються як ті, що не відповідають типовій поведінці мережі.

Граф G представлений у вигляді набору вузлів V та ребер E:

$$G = (V, E), \tag{2.3}$$

Кожен вузол $v_i \in V$ має початковий вектор ознак x_i, а кожне ребро $e_{ij} \in E$ може мати атрибути w_{ij}.

На кожному кроці t атрибути вузла $h_i^{(t)}$ оновлюються за допомогою інформації від його сусідів $N(i)$:

$$h_i^{(t+1)} = \sigma\left(W^{(t)} h_i^{(t)} + \sum_{j \in \mathcal{N}(i)} \alpha_{ij} W^{(t)} h_j^{(t)} \right), \tag{2.4}$$

де $W^{(t)}$ – матриця ваг на шарі *t*, $h_i^{(t)}$ – вектор ознак вузла на шарі *t*, $N(i)$ – набір сусід на вузлі v_i ,α_{ij} – коефіцієнт уваги між вузлами v_i та v_j , σ – активаційна функція.

Після кількох ітерацій агрегації вузли отримують оновлені атрибути. Для кожного ребра e_{ij}, його клас визначається за допомогою функції:

$$y_{ij} = f\left(h_i, h_j, w_{ij}\right), \tag{2.5}$$

де f – багатошаровий MLP, який на основі ознак вузлів і ребра класифікує його як нормальний або аномальний.

2.2.5 Виявлення Silver та Gorden Tickets за допомогою Autoencoder

Autoencoder є ефективним підходом для виявлення аномалій у складних системах, таких як AD, завдяки своїй здатності навчатися на нормальних даних і виявляти відмінності, які вказують на потенційні аномалії. У випадку атак Silver та Golden Tickets, при яких зловмисники створюють підроблені квитки аутентифікації, автоенкодер здатний виявляти аномалії на основі високої помилки відновлення між вхідними та відновленими даними. Нижче подамо основні переваги використання Autoencoder для виявлення Silver та Gorden Tickets.

1. Моделювання нормальної поведінки. Традиційні методи виявлення аномалій, такі як Isolation Forest або Local Outlier Factor, можуть виявляти аномалії в структурованих даних. Проте вони обмежені при роботі з багатовимірними та нелінійними залежностями. Autoencoder створює модель нормальної поведінки шляхом навчання на даних, що відображають типову взаємодію в середовищі AD. Коли відбувається атака, наприклад, з підробкою Silver чи Golden Ticket, модель не

здатна відновити ці дані з високою точністю, що дає змогу виявити аномалії через високий рівень помилки реконструкції.

2. Здатність обробляти складні атрибути. Autoencoder здатний працювати з багатовимірними даними, такими як атрибути квитків Kerberos, час дій, IP-адреси тощо, і комбінувати їх у компактні подання, що дає змогу моделі більш точно враховувати взаємозв'язки між різними характеристиками. Це особливо важливо для атак Silver та Golden Tickets, де зловмисники можуть змінювати багато аспектів взаємодії в системі.

3. Гнучкість у виявленні аномалій. У порівнянні з іншими методами, такими як RNN або LSTM, які орієнтовані на часові залежності, Autoencoder дає змогу створювати узагальнену модель нормальної поведінки, незалежно від конкретного часу. Це важливо для атак Silver та Golden Tickets, де зловмисники можуть діяти в будь-який час без створення чітких часових шаблонів.

4. Оптимізація для великих обсягів даних. Завдяки своїй архітектурі, Autoencoder здатний працювати з великими наборами даних, що дає змогу здійснювати ефективний аналіз журналів подій у середовищі AD, де дані можуть бути об'ємними і різноманітними. Autoencoder здатний зменшити розмірність вхідних даних до компактного подання, що знижує обчислювальні витрати при обробці великих обсягів даних.

Autoencoder складається з двох основних компонентів: енкодера і декодера. Перший компонент, енкодер, зменшує розмірність вхідних даних $X \in R^d$ до компактного латентного подання $h \in R^m$, при m < d. Формула для енкодера:

$$h = f(X, \theta_e) = \alpha \cdot X, \tag{2.6}$$

де $f(X, \theta_e)$ – функція енкодера, яка перетворює вхідний вектор X у латентне подання h, α – набір параметрів, що визначають шари енкодера, h – латентне подання розмірності m.

Декодер відновлює вихідні дані $\hat{X} \in R^d$ з латентного подання h:

$$\hat{X} = g\left(h, \theta_d\right) = \beta \cdot X, \tag{2.7}$$

де $g(h, \theta_d)$ – функція декодера, яка перетворює латентне подання h назад у вихідний вектор $\hat{X}$, β – набір параметрів декодера, що визначають його структуру.

Для оцінки якості реконструкції в Autoencoder використовується функція втрат, яка зазвичай є середньоквадратичною помилкою між вхідними даними X та відновленими даними $\hat{X}$:

$$L = \frac{1}{d}\sum_{i=1}^{d}\left(X_i - \hat{X}_i\right)^2, \tag{2.8}$$

де X_i – i-й елемент вхідного вектора X, $\hat{X}_i$ – i-й елемент відновленого вектора $\hat{X}$, d – кількість елементів у векторі X

Аномальні дані, які сильно відрізняються від нормальних, призведуть до високої функції втрат L. В такому випадку, реконструйовані дані матимуть велику помилку відновлення, що свідчить про їх аномальність. Цей підхід дає змогу виявляти атаки, такі як Silver та Golden Tickets, через аномальні шаблони, які неможливо точно відновити моделлю.

2.3. Заключний етап методу виявлення аномалій під назвою “Інтерпретатор зв’язків”

Інтерпретатор зв’язків є заключним етапом запропонованого методу виявлення аномалій, що дає можливість виявляти окремі аномалії та аналізувати взаємозв’язки між ними. Цей етап розширює функціональність системи, забезпечуючи виявлення складних атак, які можуть включати кілька етапів, що реалізуються через різні типи атак і аномалій, що спостерігаються в результатах з різних модулів.

Інтерпретатор зв’язків має здатність об’єднувати підозрілі активності, виявлені різними модулями, зокрема:

– Isolation Forest для виявлення аномалій у запитах до LDAP або DNS;

– LSTM для виявлення аномалій PtH;

– One-Class SVM для виявлення аномалій Kerberoasting;

– GNNs для виявлення переміщень мережею;

– Autoencoders для виявлення аномалій Golden Ticket та Silver Ticket.

Ідея цього етапу полягає в тому, що одна аномалія, виявлена в одному модулі, може бути не достатньою для серйозного висновку про загрозу. Однак коли кілька модулів одночасно виявляють підозрілі активності, це може свідчити про більш складну атаку, що охоплює кілька етапів або взаємопов'язаних атак.

Інтерпретатор зв'язків дає змогу обчислювати загальний рівень загрози, поєднуючи різні підозрілі активності та визначаючи, чи є між ними тісний зв'язок, що вказує на складну, багатоступеневу атаку, або ж це просто ізольовані аномалії.

Висновки до розділу 2

У другому розділі роботи запропоновано метод для виявлення аномалій у середовищі AD, який ґрунтується на використанні багатомодульної системи з різними алгоритмами ML. Запропонований підхід охоплює широкий спектр атак, включаючи сканування, ескалацію привілеїв, бічне переміщення та підробку квитків, завдяки спеціалізації кожного модуля на певному типі загроз. Особливістю спроєктованого методу є включення етапу "інтерпретатор зв'язків", який аналізує кореляції між результатами роботи окремих модулів, забезпечуючи виявлення складних багатостадійних аномалій.

Кожен модуль методу спеціалізується на окремому типі аномалій, що відповідають різним атакам. Так, для виявлення аномальної активності, пов'язаної зі скануванням, використовується Isolation Forest; для аналізу послідовностей дій користувачів під час атак на основі ескалації привілеїв застосовуються LSTM та One-Class SVM; GNN аналізують зв'язки між вузлами для ідентифікації нелегітимних переміщень мережею. Модулі на базі Autoencoder точно визначають підробку аутентифікаційних квитків, як от Golden Ticket чи Silver Ticket. У підсумку, інтеграція цих модулів на основі різних алгоритмів ML дає змогу комплексно виявляти небезпечні події в AD, підвищуючи надійність корпоративної інформаційної системи.

РОЗДІЛ 3 Програмна реалізація методу виявлення аномалій в Active Directory засобами машинного навчання та підготовка навчальних даних

3.1 Ознаки атак у середовищі Active Directory

Спроєктований метод спрямовано на виявлення чотирьох ключових векторів атак: доменне сканування, ескалація привілеїв, переміщення по мережі та постійне утримання доступу. Система ґрунтується на аналізі логів і запитів до AD, а також на дослідженні графових зв'язків між вузлами мережі. На рисунку 3.1 подано загальну схему основних завдань для виявлення потенційних аномалій в AD.

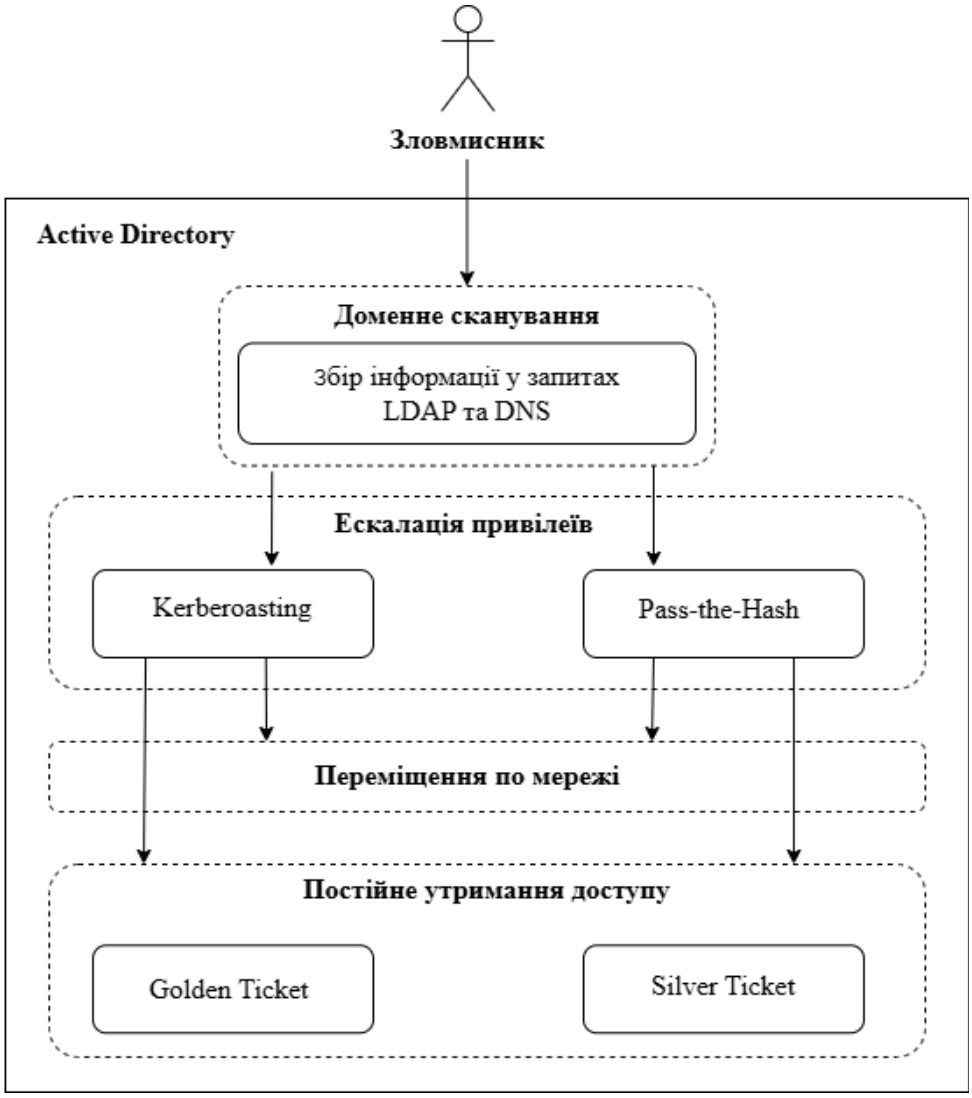


Рисунок 3.1 – Схема основних типів атак на AD, що розглядаються, відповідно до запропонованого методу

Відповідно до поданої схеми на рисунку 3.1, доцільно детально розглянути кожен із чотирьох векторів атак окремо.

1. Доменне сканування. Першим етапом є отримання повної інформації про середовище AD, включно зі структурою мережі, об’єктами каталогу, а також ключовими сервісами. Для цього зловмисники зазвичай використовують запити до LDAP та DNS, оскільки вони дають змогу отримати відомості про облікові записи, групи доступу, політики та розташування критично важливих ресурсів (наприклад, доменних контролерів).

2. Ескалація привілеїв. Після збирання докладної інформації про мережу зловмисники переходять до отримання доступу з вищим рівнем привілеїв. Типовими прикладами є техніки Kerberoasting та PtH. У разі Kerberoasting характерним є надмірна кількість запитів квитків TGS (Ticket Granting Service) з єдиного облікового запису, а також застосування слабких або застарілих алгоритмів шифрування (наприклад, RC4).

3. Переміщення по мережі (Lateral Movement). Отримавши розширені права, зловмисники починають нелегітимно розширювати свою присутність, досягаючи інших вузлів у мережі. Цей етап супроводжується появою нових, нетипових з’єднань та підвищенням інтенсивності взаємодій між вузлами, яка суперечить звичним операційним патернам. Ознаками такої активності також можуть бути доступ до ресурсів, на які користувач не мав прав раніше, або підключення до того самого ресурсу за різними обліковими записами протягом короткого часового інтервалу.

4. Постійне утримання доступу. На завершальному етапі зловмисник прагне забезпечити тривале збереження контролю над інфраструктурою AD за допомогою підроблених квитків аутентифікації, найчастіше Golden Ticket або Silver Ticket. Для Golden Ticket характерним є значно подовжений час дії квитка, який не відповідає типовим політикам, а також наявність розбіжностей у часових мітках автентифікації, що не знаходить відображення у стандартних журналах подій. Натомість Silver Ticket може бути виявлено за ознаками незвичних підключень до сервісів без проходження централізованої автентифікації через контролер домену, а також через використання параметрів, нетипових для звичайної взаємодії.

Отже, викривальні сліди кожного етапу атаки дають змогу побудувати всебічну картину дій зловмисника. Зібрані дані потребують попереднього оброблення та аналізу, що уможливлює генерацію синтетичних датасетів для навчання алгоритмів машинного навчання, орієнтованих на виявлення аномалій у середовищі AD.

3.2 Створення синтетичних даних на основі ознак аномалій

Вихідною проблемою є брак доступних реальних логів через вимоги конфіденційності й безпеки, а також недостатню кількість відкритих датасетів із повноцінними прикладами атак. Замість цього було ухвалено рішення генерувати синтетичні набори даних, які належно відображають ключові аспекти роботи AD та різноманітні вектори атак. Такий підхід уможливлює:

1. Формування достатньої кількості аномальних прикладів. Через низьку ймовірність реальних атак у загальному обсязі логів (зазвичай <5% від усіх записів) алгоритми машинного навчання можуть не мати достатнього матеріалу для адекватного розпізнавання аномалій.

2. Репрезентативність та відтворюваність. Синтетичні логи імітують реальні умови експлуатації AD, включно з часовими параметрами, типовими послідовностями запитів, розподілом ролей і прав користувачів, а також частотними характеристиками дій.

3. Технічні механізми генерації. Одним із базових принципів є структурування на основі реальних шаблонів подій, що передбачає використання знеособлених метаданих із реальних середовищ AD для формування профілів нормальної поведінки. Це охоплює спектр типових IP-адрес, часові закономірності активності, а також розподіл кількості запитів до LDAP і DNS упродовж робочого дня.

4. Гнучкість налаштування сценаріїв. Завдяки параметризації можливе варіювання розміру домену, зокрема кількості контролерів, облікових записів та структури OU, що дає змогу краще відтворювати реальні конфігурації AD.

5. Етап попереднього оброблення. Перед подальшим використанням синтетичні дані проходять фільтрацію, нормалізацію часових міток та атрибутів (User, IP-адреса, Service, Result Code тощо), а також позначаються мітками (між “норма” та “аномалія”). Отримані підготовлені набори застосовують для навчання обраних алгоритмів (Isolation Forest, One-Class SVM, LSTM, GNN, Autoencoder тощо).

Отже, створення синтетичних наборів даних на основі виявлених ознак аномалій дає змогу відтворити різноманітні моделі дій зловмисників і забезпечити належну кількість підозрілих прикладів для підготовки алгоритмів машинного навчання. У подальших підрозділах детально описано механізми генерації та специфіку формованих датасетів для кожного типу атаки.

3.2.1 Створення даних для виявлення аномалій у запитах LDAP та DNS

Вхідними даними для побудови моделі виявлення аномалій у запитах, наведених на рисунку 3.1, слугують журнали запитів LDAP і DNS, зібрані в середовищі AD.

ID	Protocol	User	Query_Type	Query_Target	Source_IP	Res	Timestamp	Anom_label
2	DNS	unknown	Add	Domain Admins	227.108.188.45	20	14:00:01	1
5	LDAP	unknown	Search	Domain Admins	98.131.37.171	4	14:00:05	1
12	LDAP	unknown	Search	DNS Records	168.218.183.215	89	14:00:08	1
7	LDAP	serv_acc	Search	DNS Records	52.83.250.111	89	07:14:00	0
35	DNS	user1	Add	Domain Admins	134.58.44.173	13	00:54:00	0

Рисунок 3.2 – Приклад синтетичних даних для LDAP та DNS запитів

Кожен запис містить інформацію про протокол (Protocol), користувача (User), тип запиту (Query_Type), цільовий об’єкт (Query_Target), IP-адресу джерела, кількість отриманих результатів (Res) і час запиту (Timestamp). Для підготовки даних до аналізу часові мітки нормалізовано до спільного формату, а IP-адреси верифіковано з урахуванням локальної підмережі та правилами знеособлення.

Додатково введено ідентифікатор сесії, який дає змогу відстежувати серію запитів від одного користувача протягом певного періоду.

У процесі маркування аномальних записів у журналах LDAP та DNS враховано кілька ключових критеріїв. По-перше, на підозрілу активність вказують запити, що відбуваються майже одночасно з інтервалом у кілька секунд, оскільки така поведінка нетипова для пересічного користувача в середовищі AD. По-друге, різка зміна IP-адрес у межах короткого часового проміжку може свідчити про навмисне розподілення трафіку з метою обходу систем виявлення. Крім того, додатковим чинником ризику є висока частота звернень до критично важливих об’єктів, зокрема до групи Domain Admins чи специфічних DNS-записів доменних контролерів, що зазвичай рідко запитуються звичайними обліковими записами.

3.2.2 Створення даних для виявлення аномалій Pass-the-Hash

У процесі формування даних для моделювання атаки PtH (рисунок 3.3) було зібрано журнали автентифікації, що реєструють детальний перебіг автентифікаційних запитів у середовищі Windows: кожен запис містить інформацію про користувача (User), IP-адресу джерела (Source_IP), цільовий хост (Target_Host), метод автентифікації (Auth_Method), наявність чи відсутність попередньої автентифікації (Pre_Auth), ознаку використання привілейованих облікових записів (Privil), а також часову позначку спроби входу (Time_Of_Attempt). Для кожного запису в таблиці додано поле Anom_label, що слугує індикатором підозрілої або звичайної поведінки.

ID	User	Source_IP	Target_Host	Auth_Method	Pre_Auth	Privil	Time_Of_Attempt	Anom_label
1	adm$svc	192.168.1.10	DC1.corp.local	NTLMv2	0	1	2024-06-01 12:01:00	1
2	unknown	192.168.5.15	DC1.corp.local	NTLMv2	0	0	2024-06-01 12:02:30	1
3	unknown	192.168.6.21	DC1.corp.local	NTLMv2	0	0	2024-06-01 12:04:00	1
4	b_up_srv	192.168.1.20	FileServer01	Kerberos	1	0	2024-06-01 12:05:00	0
5	empl1	192.168.3.18	Workstation03	Kerberos	1	0	2024-06-01 12:06:30	0

Рисунок 3.3. – Приклад даних для виявлення аномалійи PtH

Основна увага під час позначення аномалій приділяється кільком ключовим факторам. По-перше, підозрілим уважається надмірна частота спроб автентифікації, що надходять із різних IP-адрес за короткі часові проміжки. Цей патерн часто відображає скоординовану атаку з метою «проломити» систему автентифікації або замаскувати масивні перебори хешів під трафік від декількох вузлів. По-друге, критичним маркером є відсутність попередньої автентифікації (Pre_Auth=0) у поєднанні з використанням NTLMv2.

Отже, синтетичні дані для PtH уособлюють як стандартизовані поля, характерні для аудиту безпеки в Windows, так і спеціалізовані ознаки (Pre_Auth, Privil) для точнішого розмежування тривіальних і загрозливих дій. Завдяки цьому аналітики та алгоритми машинного навчання отримують засіб для розпізнавання прихованих патернів, що в реальних умовах часто залишаються поза увагою, якщо не виконувати цілеспрямований аналіз частоти входів, розподілу IP-адрес і типу протоколу автентифікації. –

3.2.3 Створення даних для виявлення аномалій Kerberoasting

Вхідні дані для моделювання атаки Kerberoasting становлять журнали запитів на отримання сервісних квитків TGS (TGS Request), які відображають роботу механізму Kerberos у середовищі AD (рисунок 3.4).

	ID	User	Source_IP	Timestamp	Encryp	Service_Name
0	21	main	192.168.5.15	2024-06-01 12:02:00	0x11	krbtgt
1	5	machine$	192.168.1.20	2024-06-01 12:03:00	0x12	Service01$
2	19	user_tech$	192.168.1.15	2024-06-01 12:01:30	0x12	FileServer01$
3	8	system$	192.168.2.18	2024-06-01 12:04:00	0x12	SQLService$

Рисунок 3.4 – Приклад даних для виявлення аномалійи Kerberoasting

Для навчання моделі One-Class SVM використано виняткова «нормальні» дані, які імітують штатну роботу системи без ознак втручання зловмисників. Такий підхід дало змогу алгоритму визначити «межі» типової поведінки й реагувати на будь-яке відхилення як на потенційну аномалію. Зокрема, моделюється, що чимало стандартних сервісів (Service_Name) звертаються по квитки TGS, використовуючи AES128 або AES256, і роблять це у часових проміжках, характерних для стандартних робочих сценаріїв.

Аномальні приклади, які відображають особливості Kerberoasting (наприклад, раптово збільшену кількість запитів для слабкішого алгоритму шифрування чи звернення до нетипових сервісних акаунтів), не включено до навчальної вибірки, проте застосовано під час тестування та валідації моделі. Це дало змогу оцінити, наскільки точно One-Class SVM виявляє невластиві патерни, що пов’язані зі зламом облікових записів через витяг хешів із квитків TGS.

Такий підхід із «розділенням» нормальних і аномальних даних дає змогу тренувати алгоритм на сценаріях штатного функціонування домену й тестувати здатність виявляти нетипові звернення, властиві атакам Kerberoasting.

3.2.4 Створення даних для виявлення аномалій Lateral Movement

У межах моделювання атаки Lateral Movement було сформовано синтетичні журнали, що імітують покрокову взаємодію між вузлами мережі з метою розширення доступу або досягнення критично важливих ресурсів. Кожен запис у такому журналі містить унікальний ідентифікатор (ID), ім’я користувача (User), IP-адресу джерела (Source_IP), IP-адресу цільової системи (Target_IP), часову мітку (Timestamp), порт для підключення (Port) та мітку аномальності (Anom_label). Такий набір атрибутів дає змогу не лише аналізувати окремі спроби входу, а й відстежувати цілісні ланцюжки переміщення від однієї системи до іншої, що є ключовим для виявлення складних векторів атак на кшталт Lateral Movement (рисунок 3.5).

ID	User	Source_IP	Target_IP	Timestamp	Port	Anom_label
11	jsmith	192.168.1.10	192.168.1.12	2024-06-01 12:01:00	445	0
23	jsmith	192.168.1.12	192.168.6.25	2024-06-01 12:02:30	3389	0
34	HR_JohnS	192.168.5.15	192.168.5.20	2024-06-01 12:03:00	5985	1
14	HR_JohnS	192.168.5.20	192.168.6.25	2024-06-01 12:04:30	135	1
58	HR_JohnS	192.168.5.40	192.168.5.45	2024-06-01 12:06:00	135	1

Рисунок 3.5 – Приклад даних для виявлення аномалійи Lateral Movement.

Технічна складова цього підходу полягає в тому, що кожна сесія фіксує протокол чи службу, яку використовують для підключення (наприклад, SMB через порт 445 або RDP через порт 3389). Завдяки цьому можна оцінювати, чи відповідає обраний механізм з’єднання типовим робочим сценаріям. Наприклад, SMB застосовують для доступу до спільних папок і ресурсів усередині домену, а RDP для віддалених підключень до робочих станцій. Відхилення від звичних портів чи часу доби можуть слугувати раннім сигналом потенційно небезпечної активності.

Аналіз ланцюжків переходів відбувається через зіставлення поточного Target_IP із наступним Source_IP для того самого користувача. Послідовність спроб, що логічно узгоджена (192.168.1.10 → 192.168.1.12 → 192.168.6.25), здебільшого відбиває нормальну роботу адміністративного облікового запису. З одного боку, користувач «jsmith» у типових ситуаціях саме так пересувається мережею, застосовуючи SMB для копіювання файлів і RDP для підключення до інших вузлів, роблячи це в межах стандартного робочого часу. Натомість користувач «HR_JohnS» може демонструвати підозрілу поведінку, якщо його переміщення мають розриви: наприклад, перехід 192.168.5.15 → 192.168.5.20 → 192.168.6.25 виглядає логічно, але раптовий стрибок 192.168.5.40 → 192.168.5.45 може не узгоджуватися з попередніми діями й відбуватися з використанням порту 135 (RPC/DCOM). Це свідчить про потенційне приховане бокове переміщення або використання вразливостей, що дає змогу несанкціоновано отримати доступ до бажаного сервера.

Завдяки тому, що синтетичні журнали охоплюють різноманітні сценарії (від звичайної щоденної роботи до послідовностей, характерних для складних бокових переміщень), алгоритми ML отримують достатньо ознак для диференціювання типових і нетипових шляхів доступу. У процесі налаштування та аналізу цих даних дослідники можуть застосовувати як класичні методи (наприклад, виявлення патернів за допомогою алгоритмів кластеризації), так і сучасні підходи на основі GNN, що здатні враховувати структуру взаємозв'язків між вузлами в контексті домену.

Такий високорівневий аналіз часто дає змогу викрити приховані взаємозалежності, які виявляються недоступними для традиційних засобів моніторингу та систем аналізу логів.

3.2.5 Створення даних для виявлення Silver Tickets та Golden Tickets

Розглянемо механізм формування синтетичних даних для виявлення аномалійи Silver Ticket, що передбачає використання зловмисниками підроблених сервісних квитків TGS без звернення до доменного контролера (DC). На рисунку 3.5 продемонстровано приклад журнала, де кожен запис містить такі атрибути: User, Source_IP, Target_Host, Service_Name, Req_To_DC (ознака звернення до DC), Timestamp та Anom_label.

З технічного погляду, відмінність між легітимною та аномальною поведінкою полягає у значенні поля Req_To_DC. Якщо Req_To_DC = 1, це свідчить про коректну автентифікацію, коли клієнт отримує сервісний квиток TGS у доменному контролері. Натомість Req_To_DC = 0 вказує на те, що жодного звернення до DC не відбулося, а сервіс прийняв підроблений квиток локально, не перевіряючи його справжність у центральній базі Kerberos. Здебільшого, ця «локальна автентифікація» реалізується шляхом використання власних ключів сервісу (зазвичай у файлі service account key) – так і виникає можливість створити Silver Ticket, дійсний лише для одного певного сервісу чи хоста.

На рисунку 3.6 наведено приклад, де користувач user1 виконує доступ до CIFS (FileServer01) та PRINT$ (PrintServer01), причому в полі Req_To_DC зафіксовано 0.

Це означає, що сервіс підтвердив автентифікацію без залучення DC, і в журналі контролера домену не залишається відповідних подій. У реальному середовищі зловмисник, маючи Silver Ticket, може звертатися до певного сервісу фактично необмежену кількість разів, доки діє підроблений квиток, не викликаючи підозрілих логів на рівні DC. У наведеному синтетичному сценарії користувач (user1) позначений як аномальний (Anom_label = 1), адже така поведінка свідчить про можливе використання підробленого квитка TGS.

ID	User	Source_IP	Target_Host	Service_Name	Req_to_DC	Timestamp	Anom_label
12	user1	192.168.1.15	FileServer01	CIFS	1	2024-06-01 12:01:00	1
21	user1	192.168.1.15	PrintServer01	PRINT$	1	2024-06-01 12:02:00	1
33	srv$	192.168.3.10	SharePoint01	HTTP	0	2024-06-01 12:03:00	0
14	srv$	192.168.3.20	DBServer01	MSSQL$Admin	0	2024-06-01 12:04:00	0

Рисунок 3.6 – Приклад даних для виявлення аномалійи Silver Tickets

Протилежна ситуація спостерігається для облікового запису srv$, чиї звернення до SharePoint01 (HTTP) та DBServer01 (MSSQL$Admin) мають Req_To_DC = 1. Це свідчить, що сервісний квиток отримано від DC у коректний спосіб, і запис про цю дію з’являється у журналах доменного контролера. Відповідно, атрибут Anom_label для цих рядків дорівнює 0, оскільки вони демонструють звичайну, легітимну взаємодію в середовищі AD.

Для підвищення правдоподібності синтетичних даних у процесі генерації враховувалися часові закономірності (наприклад, запити в межах робочого часу), типові Service_Name (CIFS, PRINT$, HTTP, MSSQL$Admin) і Target_Host (FileServer01, PrintServer01, SharePoint01, DBServer01), а також розподіл Source_IP відповідно до структурованих підмереж домену. Що стосується відображення подібних подій у реальному середовищі, Silver Ticket-атака є особливо небезпечною, адже дає змогу приховати сліди компрометації на рівні DC. Завдяки створеному синтетичному набору можна точно перевіряти алгоритми виявлення, орієнтовані на кореляційний аналіз логів, пошук закономірностей невідповідності схемі взаємодії

Kerberos та визначення нетипової тривалості квитків або відсутності запитів автентифікації на DC.

Далі розглянемо підхід до формування синтетичних даних для моделювання Golden Ticket, коли зловмисник створює підроблений TGT (Ticket Granting Ticket), що надає фактично необмежений доступ у домені. На рисунку 3.7 наведено приклад журнального запису, де кожен рядок містить такі поля: User, Source_IP, Target_Host, Service_Name, Ticket_Start, Ticket_End, SID, а також Anom_label. Ключовим чинником тут є аномально тривалий час дії квитка, який суттєво перевищує типові 8–10 годин, визначені політиками Kerberos.

ID	User	Source_IP	Target_Host	Serv_Name	Ticket_Start	Ticket_End	SID	Anom_label
1	admin$	192.168.1.10	DC1.corp.local	krbtgt	2024-06-01 12:00	2025-06-01 12:00	S-1-5-21-Admin	1
2	admin$	192.168.1.10	FileServer01	CIFS	2024-06-01 12:01	2025-06-01 12:01	S-1-5-21-Admin	1
3	srv$	192.168.2.20	DC1.corp.local	krbtgt	2024-06-01 12:02	2024-06-01 20:02	S-1-5-21-NormalUser	0
4	srv$	192.168.2.25	SharePoint01	HTTP	2024-06-01 12:03	2024-06-01 20:03	S-1-5-21-NormalUser	0

Рисунок 3.7 – Приклад даних для виявлення аномалійи Golden Tickets

З технічного погляду, відстеження значення Ticket_End дає змогу виявляти нетипові інтервали, наприклад, 365 днів чи кілька років. У справжньому середовищі AD такий показник одразу виходить за межі дозволених політик і свідчить про компрометацію облікового запису “krbtgt” або використання підроблених ключів Kerberos. Додатково слід перевіряти вміст SID, оскільки для Golden Ticket часто встановлюють ідентифікатор привілейованої групи (на кшталт S-1-5-21-Admin) навіть для користувачів, які не мають відповідних прав. Це дає зловмиснику змогу обійти стандартні механізми контролю доступу та автоматично отримувати розширені дозволи.

У наведених прикладах для облікового запису admin$ спостерігається неприродно тривалий час дії квитка – 365 днів замість кількох годин, що явно суперечить штатним налаштуванням AD. Крім того, SID S-1-5-21-Admin свідчить про привілеї, нетипові для машини або звичайного службового акаунта. Натомість запис srv$ демонструє стандартну тривалість квитка (8 годин) із нормальним SID, не

виявляючи ознак компрометації. Завдяки такому контрасту між «нормальними» та «аномальними» рядками алгоритми виявлення загроз можуть точно навчатися розпізнавати патерни, характерні для Golden Ticket.

3.3 Попереднє оброблення даних

Нижче наведено узагальнений опис процесу попереднього оброблення даних, який дає змогу підготувати набір ознак, необхідних для побудови ефективних моделей виявлення аномалій у середовищі AD. Як уже зазначалося у п. 3.1, нетипова поведінка нерідко проявляється в аномальній частоті чи послідовності дій. З огляду на це, у процесі оброблення логів формують такі додаткові ознаки:

1. Часовий інтервал між запитами (Time_Since_Last_Attempt). Ця ознака обчислюється як різниця (у секундах або мілісекундах) між часовими мітками двох послідовних запитів одного й того самого користувача. Вона дає змогу виявляти епізоди раптового зростання інтенсивності звернень, характерні, наприклад, для масового сканування LDAP або DNS. Подамо це у вигляді формули:

$$\text{Dif\_Time\_Of\_Attempt} = T_{\text{Attempt}_i} - T_{\text{Attempt}_{i\text{-}1}}, \tag{3.1}$$

де T_{Attempt_i} – час виконання *i*-го запиту.

Для першого запиту кожного користувача це значення встановлюється в 0, адже немає попередньої події для порівняння.

2. Кількість запитів за певний часовий інтервал (Attempts_Per_Window). Ця ознака визначає, скільки звернень до AD (LDAP, DNS, автентифікації тощо) виконується в межах вікна часу WWW. Вона є особливо важливою для виявлення PtH та інших атак, що передбачають швидке повторення автентифікаційних запитів. Формально подамо це так:

$$\text{Attempts\_Per\_Window} = \sum_{j=1}^{N} \delta\left(T_j \in [T_i - W, T_i]\right), \tag{3.2}$$

де T_i – час виконання поточного запиту, NNN – загальна кількість запитів у логах, а $\delta(\cdot)$ – індикаторна функція, яка дорівнює 1, якщо момент часу T_j потрапляє у відрізок $[T_i - W, T_i]$, інакше 0.

3. Ланцюжок переходів IP-адрес (Chain_ID). Задля виявлення Lateral Movement вводять ознаку, що відображає цілісну послідовність переходів користувача між вузлами. Якщо запис (рядок у логах) показує перехід від однієї IP-адреси до іншої, то системі необхідно перевірити, чи є він продовженням поточного ланцюга, чи ініціює новий.

а) Для першого запису кожного користувача Chain_ID встановлюють у 1.

б) Якщо $\text{Source\_IP}_i = \text{Target\_IP}_{i-1}$ і водночас $User_i = User_{i-1}$, то поточний запис продовжує той самий ланцюжок (Chain_ID лишається незмінним).

в) Якщо ж умова не виконується, створюється новий ланцюжок (Chain_ID збільшується на 1).

Така обробка дає змогу відстежувати «логічну» послідовність переходів і виявляти розриви та нестандартні переміщення, притаманні атакам бокового переміщення (Lateral Movement).

4. Аномально великий час життя квитка (Ticket_Lifetime). Для Golden Ticket важливим є час дії підробленого квитка, що часто набагато перевищує звичні політики AD (8–10 годин). Виходячи з полів Ticket_Start і Ticket_End, обчислюють:

$$\text{Ticket\_Lifetime} = \text{Ticket\_End} - \text{Ticket\_Start}, \tag{3.3}$$

де Ticket_Start і Ticket_End відповідно означають часи початку та закінчення дії квитка Kerberos. Якщо це значення значно перевищує максимально дозволені політикою домену (наприклад, може становити кілька діб чи навіть років), можна припустити компрометацію квитка (зокрема, з використанням Golden Ticket).

Застосування цих додаткових ознак дає змогу моделям виявлення аномалій у середовищі AD точніше визначати відхилення від типової поведінки, зокрема частоту звернень, непослідовні ланцюжки переміщень і нетипову тривалість дії Kerberos-квитків.

3.4 Реалізація та навчання модулів виявлення аномалій

На рисунку 3.8 наведено узагальнену схему, що ілюструє послідовність етапів від збору логів до формування інтегральної оцінки загрози.

Рисунок 3.8 – Схема застосування запропонованого методу для виявлення аномалій

3.4.1 Реалізація Isolation Forest для виявлення аномалій LDAP і DNS

Розглянемо поданий опис реалізації Isolation Forest для виявлення аномальних запитів LDAP та DNS. Цей метод ґрунтується на ідеї випадкових розподілів простору ознак, які швидше ізолюють спостереження з нетиповими значеннями. У середовищі AD запити до LDAP і DNS часто виступають основними механізмами взаємодії між клієнтами та контролерами домену, тому відхилення в частоті, вмісті чи часу

виконання таких запитів дають змогу виявляти різні вектори атак, як-от доменне сканування, збір даних про критичні об’єкти чи спроби обходу системи безпеки.

Рисунок 3.9 ілюструє загальний процес реалізації алгоритму Isolation Forest для виявлення аномалій у запитах LDAP і DNS.

У схемі на рисунку 3.9 показано послідовні етапи формування дерев, оцінювання глибини ізоляції для кожного запису та обчислення інтегрального показника аномальності. Завдяки випадковим розподілам у просторі ознак аномальні запити, що мають нетипові значення (наприклад, надмірну частоту звернень або звернення до критично важливих об’єктів), ізолюються значно швидше від нормальних точок, що і дає змогу точно виявляти потенційні атаки.

1. Підготовка даних та визначення ознак. Передусім формуються синтетичні логи, які відтворюють як типову активність (рівномірні запити упродовж робочого часу, переважно до стандартних об’єктів), так і аномальну (раптове зростання інтенсивності, звернення до рідкісних ресурсів чи повторні запити з різних IP-адрес). Для кожного запису (запиту) визначають низку ключових атрибутів:

– Protocol (LDAP або DNS);

– User (обліковий запис або службовий акаунт);

– Query_Type (тип запиту LDAP чи тип DNS-запису);

– Query_Target (цільовий об’єкт або ім’я ресурсу);

– Source_IP (IP-адреса джерела);

– Res (кількість отриманих результатів);

– Timestamp (час запиту).

Окрім базових полів, можуть бути додані Time_Since_Last_Attempt та Attempts_Per_Window (згідно з розділом 3.3) для фіксації частоти звернень.

2. Настроювання параметрів Isolation Forest. Алгоритм Isolation Forest має низку налаштовуваних параметрів, серед яких:

– n_estimators – кількість «випадкових дерев» (рекомендується 100–200);

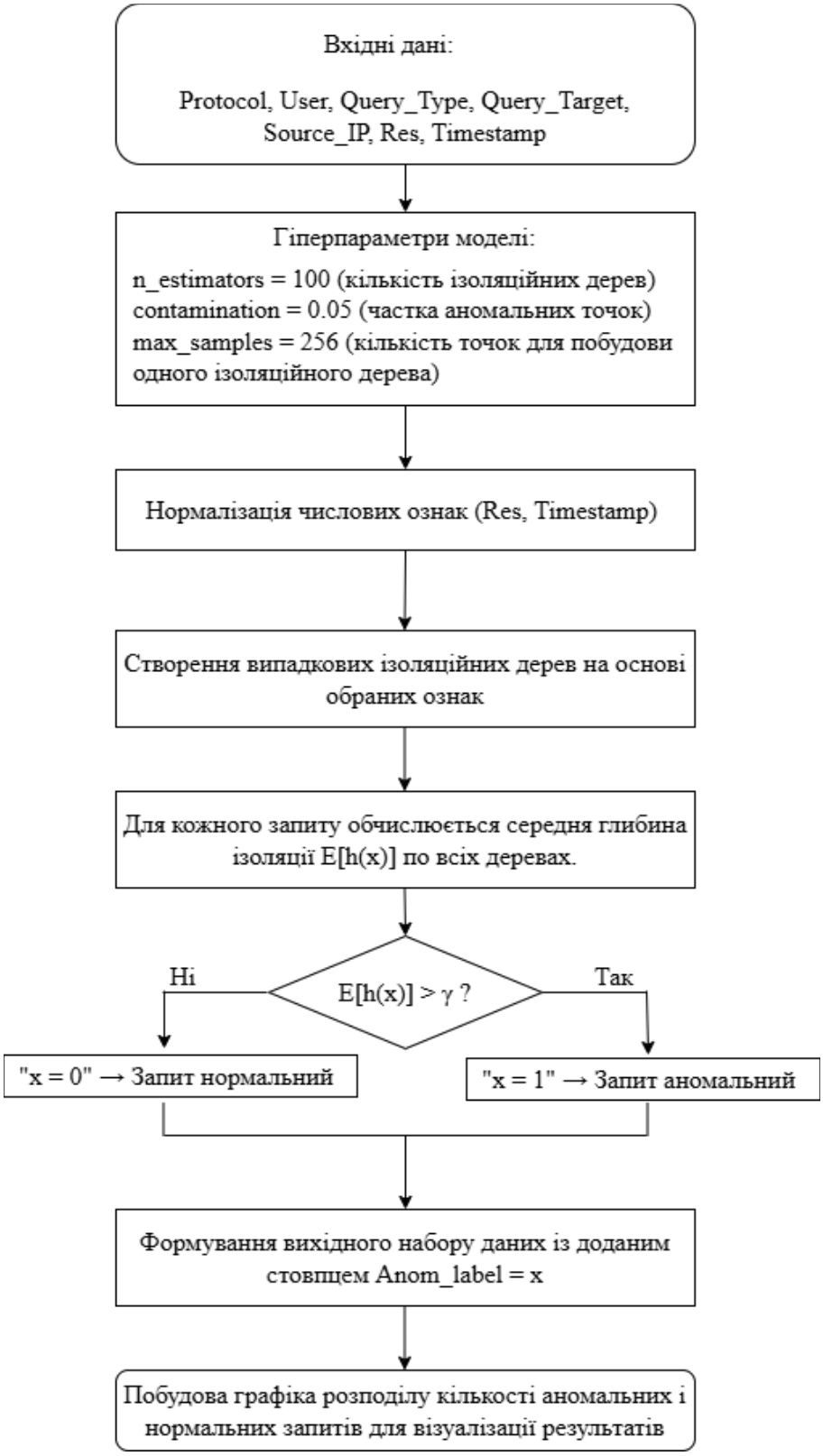


Рисунок 3.9 – Блок-схема алгоритму Isolation Forest для виявлення аномалій в запитах LDAP та DNS

– max_samples – кількість вибірок (записів) для побудови кожного дерева; може бути задана як відсоток від загального обсягу даних (наприклад, max_samples=256 чи 0.2 від усіх записів);

– max_features – кількість ознак, що випадково вибираються для поділу на кожному вузлі дерева;

– contamination – приблизна частка аномальних зразків у наборі; якщо оцінити, що близько 5% трафіку є підозрілим, встановлюють contamination=0.05.

Під час проведення експериментів параметри можуть оптимізуватися за допомогою перехресної перевірки або на основі внутрішніх метрик алгоритму (наприклад, середньої глибини ізоляції).

3. Побудова та навчання моделі. На основі відібраного набору ознак формується матриця розмірності N×M, де N – кількість записів, а M – кількість ознак. Модель Isolation Forest будується за таким принципом:

– формування дерев: на кожному дереві випадковим чином обирають підмножину ознак та відповідні пороги поділу, поки всі точки не ізолюються в листах;

– вимірювання «глибини» ізоляції: чим швидше запис відокремлюється, тим він «підозріліший»;

– агрегація результатів: усереднюють глибину ізоляції для кожного запису за всіма деревами й отримують інтегральний показник аномальності.

4. Оцінка результатів і виявлення аномалій. Після навчання кожному запису призначають «score» аномальності – що вищий score, то більша ймовірність, що запит є нетиповим. Встановлюється порогове значення (threshold), вище якого запис класифікується як аномальний. Типові приклади в середовищі AD:

– запити зі значно вищою частотою за короткий проміжок (може свідчити про сканування LDAP чи перебір DNS-записів);

– звернення до рідкісних чи критичних об’єктів (Domain Admins, DNS-записи контролера домену);

– запити з незвичних IP-адрес або облікових записів, які раніше не спостерігалися в логах.

5. Виявлені патерни аномалій і нормальна активність. Згідно з побудованою моделлю, за високими показниками аномальності зазвичай ховаються:

– множинні запити LDAP від «порожніх» або «невідомих» облікових записів;

– часті DNS-запити щодо одного і того ж ресурсу з різних IP-адрес (корельована активність для маскування);

– звернення до Domain Admins або високопривілейованих OU.

Натомість нормальні запити мають рівномірний розподіл за часом, звертаються до поширених служб і виконуються під відомими акаунтами з IP-адрес офісної мережі.

6. Візуалізація та інтерпретація результатів. Для зручності аналізу використовують методи проєкції високовимірних даних чи підсвічують найбільш «глибоко ізольовані» точки на графіку. У реальному середовищі AD відхилення, виявлені Isolation Forest, зазвичай передаються в систему SIEM чи на аналітику безпеки для додаткового розгляду. Важливо пам’ятати, що алгоритм можна налаштовувати під конкретну інфраструктуру, змінюючи поріг, частку очікуваних аномалій тощо.

Отже, Isolation Forest виявляється ефективним методом початкового автоматизованого виявлення аномальних запитів LDAP і DNS. Завдяки випадковій ізоляції нетипових точок та невисокій обчислювальній складності, метод добре масштабується навіть на великі журнали подій і підходить як перша лінія виявлення потенційного доменного сканування чи інших атак на ранніх стадіях.

3.4.2 Реалізація One-Class SVM для Kerberoasting

Нижче наведено основні критерії, за якими визначають аномальні запити на отримання квитків TGS у межах атаки Kerberoasting:

1. Надмірна частота запитів. Зловмисники можуть неодноразово звертатися по квитки за короткий проміжок часу, щоб потім спробувати зламати витягнуті хеші паролів. Отже, нетипово велика кількість запитів TGS з однієї IP-адреси чи одним обліковим записом свідчить про ймовірну спробу брутфорс-атаки.

2. Використання застарілих методів шифрування. Запити з типом шифрування 0x17 (RC4-HMAC) можуть вказувати на Kerberoasting, оскільки RC4-HMAC є більш вразливим до криптографічних атак. Натомість легітимна активність зазвичай ґрунтується на сучасних алгоритмах (0x11, 0x12), які застосовують AES-шифрування (AES128, AES256).

3. Орієнтованість на критичні служби. Аномальні запити нерідко спрямовані на високопривілейовані або рідко використовувані службові акаунти (MSSQL$Admin, SharePointAdmin тощо). Оскільки ці облікові записи мають розширені права або зберігають конфіденційну інформацію, їх компрометація є привабливою метою для зловмисників.

4. Відсутність символу «$» у назві облікового запису. Значна частина легітимних звернень надходить від машинних акаунтів (machine$, system$, srv$), які за замовчуванням мають символ «$» у імені. Якщо запит безпосередньо ініціюється користувачем без такої ознаки, це може свідчити про позастандартний сценарій (наприклад, вручну запущений скрипт чи утиліта для витягування хешів).

Модель, що навчена на «нормальних» даних (де переважає AES-шифрування і логічно розподілені в часі запити від машинних акаунтів), реагує на будь-який відхилений патерн як на аномалію. Отже, One-Class SVM дає змогу вчасно виявляти можливі спроби Kerberoasting, орієнтовані на застарілі алгоритми шифрування та облікові записи з підвищеними привілеями.

3.4.3 Використання LSTM для виявлення Pass-the-Hash

Нижче наведено опис підходу до використання LSTM для виявлення аномалійи PtH, коли зловмисник обходить необхідність знати фактичний пароль, використовуючи викрадений хеш для автентифікації:

1. Побудова архітектури моделі. Для виявлення PtH на основі часових рядів застосовують один або кілька LSTM-шарів, що дає змогу обробляти послідовності запитів у логах.

а) Кількість шарів і розмір пам’яті. У типовому сценарії використовують один або два LSTM-шари, що дає змогу зберігати та обробляти залежності на рівні часової послідовності. Кожен шар зазвичай містить від 64 до 256 нейронів (при невеликих наборах даних) або більше (при великих обсягах логів). Підвищення кількості шарів і розміру прихованого стану може покращити точність, але й збільшує ризик перенавчання та потребує більшої обчислювальної потужності.

б) Розмір вхідного шару. Кожен приклад (запис) може бути відтворений у вигляді вектора ознак (User, Source_IP, Target_Host, Auth_Method, Time_Since_Last_Attempt тощо). Для часових рядів зазвичай формують послідовність таких записів за кожного користувача, вказуючи, як змінюється його активність протягом певного періоду.

2. Вибір функцій активації та оптимізатора. Правильна конфігурація цих модулів безпосередньо впливає на результативність навчання та здатність моделі LSTM узагальнювати дані.

а) Функції активації. Унутрішні механізми LSTM покладаються на сигмоїдну (σ) та тангенсову (tanh) активації для керування процесами «запам’ятовування» та «забування» інформації у комірках. Для вихідного шару (наприклад, Dense) часто використовують сигмоїдну активацію, якщо завдання полягає в бінарній класифікації (аномалія/норма), або softmax, коли є кілька класів.

б) Оптимізатори. У більшості випадків використовують алгоритми Adam або RMSProp, оскільки вони динамічно коригують швидкість навчання під час градієнтного спуску. Для запобігання перенавчанню модель доповнюють відсіканням градієнтів (gradient clipping), а також регуляризацією (dropout або L2-регуляризація), що допомагає зберегти узагальнювальну здатність LSTM на великих наборах даних.

3. Аналіз часових послідовностей активності користувачів. На цьому етапі LSTM-модель опрацьовує часові ряди, що описують поведінку користувачів під час автентифікації.

а) Формування вхідних часових вікон. Журнали автентифікації перетворюють на послідовності, де кожен елемент відповідає окремому запису (спробі входу, використанню певного протоколу тощо). Такі послідовності формують із фіксованою

довжиною (наприклад, вікном у 10 чи 20 подій) або динамічно розмежовують за певними часовими інтервалами.

б) Виявлення PtH через нетипові патерни. Модель LSTM, яка “запам’ятовує” попередні стани, здатна виявляти різкі скачки у кількості входів, відсутність попередньої автентифікації чи зміну IP-адрес. Такі патерни часто супроводжують атаки PtH, коли зловмисник використовує викрадений хеш для масових входів із різних вузлів мережі.

Визначення тривожних ознак. Під час узагальнення послідовностей LSTM вихідна активація (або інший механізм, наприклад, attention) може підсвітити найбільш нетипові елементи. У разі виявлення значних відхилень у короткому часовому проміжку (низка швидких спроб підключень з різних IP, привілейовані облікові записи тощо) модель підвищує «рахунок підозрілості» (anomaly score).

3.4.4 Реалізація GNN для аналізу Lateral Movement

Нижче наведено опис підходу до реалізації GNN для аналізу Lateral Movement, коли зловмисник поширює свій доступ на інші вузли мережі.

1. Побудова графу взаємодій вузлів мережі AD. Спершу необхідно відтворити структуру мережі у формі графа за формулою (2.3).

Кожному вузлу присвоюють набір атрибутів (IP-адреса, роль у домені, службовий чи користувацький акаунт) і, за потреби, кожне ребро може також мати атрибути (тип протоколу, час останнього підключення). Це дає змогу моделювати різноманітні сценарії бокового переміщення, коли зловмисник переходить від одного вузла до іншого.

2. Налаштування GNN: кількість ітерацій агрегації, тип функцій активації.

а) Кількість ітерацій агрегації (depth). У більшості випадків використовують 2–3 ітерації агрегації, аби кожен вузол мав змогу отримати інформацію від найближчих сусідів та сусідів другого порядку. Надмірна глибина може призвести до «розмивання» контексту й погіршення результатів (проблема over-smoothing).

б) Тип функцій активації. Використано ReLU чи для нелінійного оброблення, щоб підвищити здатність мережі виокремлювати складні зв’язки в графі.

в) Інші параметри. Для запобігання перенавчанню зазвичай застосовують dropout на рівні вузлів чи ребер, а також L2-регуляризацію.

3. Навчання моделі на нормальних графах зв’язків і тестування на аномальних.

а) Підготовка навчальної вибірки. Формують граф (або набір графів), що відповідає типовій взаємодії в домені AD: адміністративні облікові записи, які регулярно підключаються через SMB/RDP до серверів, користувацькі машини, що взаємодіють із файловими ресурсами тощо. Ця структура вважається «нормальною».

б) Процес навчання. GNN навчається представляти вузли (node embedding) або ребра (edge embedding) у низьковимірному просторі Отже, щоб «схожі» за патернами взаємодії вузли/ребра розташовувалися близько одне до одного. Для суто аномального виявлення застосовують або one-class підхід (коли є тільки нормальні дані), або позначають рідкісні нетипові ребра (які існують лише в штучно змодельованих атаках) і навчають модель розпізнавати такі відхилення.

в) Тестування на аномальних зразках. На етапі перевірки до мережі додають «аномальні» з’єднання чи вузли, які імітують бокове переміщення (наприклад, несподівані RDP-сесії, підключення до критичних серверів без відповідної історії доступів). Модель перевіряє, наскільки ці ребра чи вузли відхиляються від нормального розташування в векторному просторі, й може класифікувати їх як підозрілі.

Отже, GNN дає змогу вийти за межі простого аналізу окремих логів і розглядати загальну структуру взаємодій у домені AD, виокремлюючи нетипові з’єднання або вузли, що свідчать про Lateral Movement. Завдяки механізмам агрегації й нелінійній обробці атрибутів вузлів і ребер, GNN здатна виявляти складні й багатокрокові шляхи, якими може рухатися зловмисник, набагато ефективніше порівняно з традиційними методами на основі окремих метрик чи сигнатур.

3.4.5 Використання Autoencoders для виявлення Golden та Silver Tickets

Нижче наведено підхід до використання автоенкодерів (Autoencoders) для виявлення підроблених квитків аутентифікації, характерних для атак Golden та Silver Tickets у середовищі AD.

1. Архітектура автоенкодера: кількість шарів енкодера та декодера. Автоенкодер складається з двох основних компонентів: енкодера (Encoder) і декодера (Decoder).

а) Енкодер приймає вхідний вектор ознак (наприклад, User, Service_Name, Ticket_Lifetime) і згортає його у більш компактне латентне представлення (h). Кількість шарів енкодера та їхня розмірність залежать від складності вхідних даних: зазвичай застосовують один-два шари з 64–128 нейронами (для невеликих датасетів) чи більше (якщо даних багато).

б) Декодер виконує протилежну дію: відновлює початковий вектор ознак з латентного простору. За кількістю та розміром шарів декодер часто дзеркально відтворює структуру енкодера, аби мати змогу точно реконструювати вхідні дані.

2. Функція втрат для аналізу відновлювальної помилки. Під час навчання модель намагається мінімізувати різницю між вхідним вектором і його відновленим варіантом ($X\hat{}$). Типовою функцією втрат виступає середньоквадратична помилка (MSE):

Для Golden чи Silver Tickets підроблені квитки часто суттєво відрізняються від «нормальних», тож відновлювальна помилка (reconstruction error) стає істотно більшою, сигналізуючи про потенційну аномалію.

3. Тестування моделі на підроблених квитках аутентифікації.

а) Навчання на «нормальних» даних. Як і у випадку з One-Class SVM, автоенкодер тренують переважно на прикладах, які відображають легітимну взаємодію: типовий час дії TGT (8–10 годин), коректні SID та наявність повної автентифікації через DC тощо. Модель «навчається» відтворювати ці структури з мінімальною помилкою.

б) Перевірка на «аномальних» прикладах. Для Golden Tickets (нетипово довгий час життя квитка, SID, що належить до привілейованих груп) або для Silver Tickets (брак автентифікації через DC, ознака Req_To_DC=0) під час відновлення автоенкодер отримує значно вищу помилку. Відповідно, ці зразки кваліфікують як потенційні аномалії.

Отже, автоенкодери дають змогу навчити модель на переважно «здоровому» середовищі та виявляти нетипові шаблони, властиві підробленим квиткам. Коли алгоритм стикається з нетиповою часовою тривалістю, SID чи відсутністю звернення до DC, відновлювальна помилка різко зростає, вказуючи на ймовірну компрометацію механізмів Kerberos.

3.5 Інтеграція запропонованих модулів в єдину систему

Нижче подано опис поєднання запропонованих модулів виявлення аномалій у середовищі AD, що має на меті забезпечити узгоджену взаємодію між ними.

1. Основні блоки реалізованого методу. До ансамблевої системи належать кілька логічних блоків, кожен із яких виконує специфічні задачі з виявлення та кореляції аномалій у середовищі AD.

а) Модулі виявлення. Кожен модуль спеціалізується на певному векторі загроз. Наприклад:

– Isolation Forest для виявлення аномалій у запитах LDAP/DNS.

– One-Class SVM або LSTM для аналізу атак ескалації привілеїв (Kerberoasting, PtH).

– GNN для виявлення бокового переміщення (lateral movement).

– Autoencoders для розпізнавання атак Golden та Silver Tickets.

Кожен із цих модулів отримує необхідні журнали подій (синтетичні чи реальні) й виводить базовий показник аномальності (score) або бінарну оцінку (anomal/normal).

б) Інтеграційний шар. Його мета – узгоджувати та агрегувати вихідні дані, отримані від усіх модулів. Цей шар виступає проміжним каналом, який ретельно

зіставляє формат записів, збирає метаінформацію про інциденти й узагальнює їх у єдиній «панелі» показників.

в) Інтерпретатор зв’язків. Цей блок об’єднує результати від кожного модуля й виконує більш високорівневу аналітику, зокрема часову та символьну кореляції подій. На основі зведених даних формується інтегральний «score» загрози, який дає змогу визначити, чи пов’язані аномалії з різних модулів у єдиний складний сценарій атаки.

2. Механізми взаємодії модулів та обміну даними. Для узгодженої роботи всіх складових системи реалізовано такі процеси.

а) Єдине сховище (Data Lake / Data Warehouse). Усі журнали подій (LDAP, DNS, Kerberos, мережеві взаємодії, записи автентифікації) збираються в уніфікованому сховищі, з якого кожен модуль виявлення витягує потрібні дані згідно зі своїм набором ознак. Це дає змогу централізовано керувати логами та спрощує інтеграцію нових джерел.

б) Форматування та узгодження атрибутів. Оскільки різні модулі можуть використовувати різні формати (векторні уявлення, символьні поля, часові ряди), на рівні інтеграційного шару відбувається узгодження ключових полів (User, Source_IP, Port тощо). Це уможливлює коректну кореляцію та обмін метаданими між алгоритмами.

в) Передавання проміжних результатів. Деякі виявлені аномальні об’єкти (наприклад, нестандартні ребра в графі GNN чи підозрілі IP-адреси в Isolation Forest) позначаються як «підвищеного ризику» й можуть слугувати додатковою вхідною ознакою для інших модулів (зокрема, для LSTM, який аналізує послідовності дій користувачів).

г) Кореляційний запис аномалій. Усі підозрілі події з різних модулів фіксуються в єдиному реєстрі аномалій. Кожному запису призначають унікальний ідентифікатор, час (Timestamp), обліковий запис (User) або ресурс (Host), на який спрямовано дії, а також первинний показник аномальності (score) від конкретного модуля. Інтерпретатор зв’язків оцінює, чи стосуються ці записи того самого етапу атаки, чи вказують на різні, але взаємопов’язані загрози.

ґ) Зворотний зв’язок та уточнення. Якщо фахівці з безпеки підтвердять або спростують певну аномалію, ці дані вносять до системи для «тренування» або «донавчання» модулів (active learning). Так забезпечується поступове вдосконалення якості виявлення, зокрема зменшується кількість хибнопозитивних спрацювань.

Нижче наведено опис реалізації заключного етапу “Інтерпретатор зв’язків”, який дає змогу інтегрувати результати з різних модулів виявлення та виявляти складні багатоетапні атаки.

1. Використання кореляційних методів для аналізу результатів модулів. Інтерпретатор отримує проміжні висновки (рівень аномальності, тип атаки, часовий інтервал) від кожного детектора (Isolation Forest, One-Class SVM, LSTM, GNN, Autoencoders) і виконує над ними кореляційний аналіз.

а) Часова кореляція: визначає, чи відбувалися підозрілі події одночасно або з коротким зсувом у часі.

б) Символьна кореляція: перевіряє спільні атрибути (User, IP-адреса, Resource) між різними аномаліями. Якщо, скажімо, два детектори сигналізують про підвищену загрозу для одного й того самого облікового запису, це може вказувати на початок скоординованої атаки.

в) Семантична кореляція: відстежує, чи пов’язані певні події за логікою мережного проникнення (доменне сканування → ескалація привілеїв → бокове переміщення).

2. Визначення критеріїв для комплексного оцінювання загрози. Щоб обчислювати інтегральний «рівень небезпеки» (threat score) або змінювати його динамічно, застосовують кілька підходів.

а) Вага результатів модулів. Кожному з детекторів призначають коефіцієнт важливості (наприклад, більш надійний або тренований на масштабніших даних модуль отримує вищу вагу).

б) Порогові значення. Якщо сукупний показник аномальності (sum чи max) перевищує певний поріг, систему переводять у режим підвищеної готовності (alert).

в) Штрафні бали. Події, що повторюються або корелюють із кількома різними векторами загроз (наприклад, одночасний сигнал від LSTM і Isolation Forest),

отримують збільшений «штрафний бал», завдяки чому інтегральний показник швидко зростає.

3. Побудова графу взаємозв'язків між аномаліями. Для візуалізації та більш комплексного аналізу інтерпретатор формує граф, у якому:

– вузли (nodes) відповідають конкретним аномальним подіям, виявленим різними модулями (наприклад, підозрілий запит Kerberoasting, нетипова RDP-сесія чи аномально довгий квиток).

– ребра (edges) відображають кореляційні зв'язки між подіями (спільний користувач, часовий збіг, той самий вузол-ціль тощо).

Такий підхід полегшує виявлення ланцюжків атаки (attack chain), коли одна подія переходить в іншу (сканування → ескалація → lateral movement → постійний доступ). Аналітики безпеки можуть візуально оцінити інтенсивність зв'язків (наприклад, товщину ліній між подіями), а також переглянути, які облікові записи чи IP-адреси мають найбільшу кількість підозрілих зв'язків.

У підсумку «Інтерпретатор зв'язків» виступає як логічний центр кореляції, що не лише інтегрує дані з різних детекторів, а й встановлює смислові зв'язки між аномаліями. Це дає змогу швидко розпізнавати складні, багатоступеневі вектори атак, які залишилися б непоміченими за умови аналізу результатів кожного модуля окремо.

Висновки до розділу 3

У третьому розділі роботи було узагальнено результати реалізації запропонованого методу виявлення аномалій у середовищі AD та детально оцінено результативність кожного модуля. Аналіз показав, що використання спеціалізованих алгоритмів ML дає змогу виявляти специфічні вектори атак з високою точністю. Зокрема, Isolation Forest ідентифікує нетипові запити до LDAP/DNS, характерні для доменного сканування, за рахунок ізоляції аномальних точок у просторі ознак. Модулі One-Class SVM і LSTM, зосереджені на ескалації привілеїв, демонструють високу результативність завдяки аналізу відхилень від типових зразків і послідовностей аутентифікації відповідно. GNN виявилися корисними для

моделювання мережевих взаємозв’язків та виявлення бічного переміщення, тоді як Autoencoders дають змогу точно діагностувати підробку квитків типу Golden і Silver Tickets завдяки аналізу відхилень у відновлених даних.

Описано підготовчі заходи до експериментального тестування програмної реалізації запропонованого методу в реальних умовах. Особливу увагу приділено налаштуванню параметрів кожного модуля, зокрема параметрів ізоляції, часових вікон та обсягу тренувальних даних, а також адаптації інтеграційного шару під специфіку цільової мережі. Очікується, що результати експериментального тестування, зокрема точність, повнота та швидкість реагування, підтвердять результативність і доцільність запропонованого методу виявлення аномалій у середовищі AD.

РОЗДІЛ 4 Дослідження та експериментальне тестування програмної реалізації за спроєктованим методом виявлення аномалій в Active Directory

У цьому розділі описується експериментальна перевірка запропонованого в розділі 2 методу виявлення аномалій в середовищі AD та проведення досліджень щодо результативності кожного з модулів, інтегрованих разом. Як зазначено в попередніх розділах, метод складається з таких ключових модулів.

а) Модуль “Доменне сканування” (Isolation Forest).

б) Модуль “Ескалація привілеїв” (LSTM, One-Class SVM).

в) Модуль “Переміщення мережею” (GNNs).

г) Модуль “Постійне утримання доступу” (Autoencoders).

ґ) Етап “Інтерпретатор зв’язків” для аналізу кореляцій.

Головна мета експерименту – визначити ступінь точності, повноти, кількість хибнопозитивних спрацьовувань та інші важливі метрики для кожного з модулів. Додатково оцінюється час оброблення подій, пов’язаний з ML, оскільки надмірні затримки можуть знизити практичну цінність методу.

4.1 Опис тестового середовища

Для проведення дослідження було розгорнуто тестове середовище, яке імітує типовий корпоративний домен Microsoft із декількома контролерами домену та сервером баз даних. Складові тестової мережі подано нижче.

1. Контролери домену (DC1, DC2):

– працюють під управлінням Windows Server 2019 із розгорнутими службами ADDS;

– зберігають журнал подій безпеки (Security Event Log), де фіксуються події входу в систему, запити Kerberos тощо.

2. Клієнтські машини (Workstations):

– кілька віртуальних робочих станцій з ОС Windows 10;

– на кожній машині симулювались дії реальних користувачів: вхід до домену, запити до файлових серверів, до сервісів баз даних.

3. Сервер баз даних (DBServer):

– також під керуванням Windows Server 2019;

– під’єднаний до домену, містить Microsoft SQL Server із даними, доступ до яких часто потребує Kerberos-квитків.

4. Програмна реалізація методу виявлення аномалій:

– містить модулі Isolation Forest, One-Class SVM, LSTM, GNN та Autoencoder;

– функціонує на окремому сервері (Linux Ubuntu 22.04), де реалізовано платформу Python 3.9, а також бібліотеки ML.NET та PyTorch;

– для зберігання й оброблення даних використовується СКБД PostgreSQL, а також кешування з допомогою Redis (для проміжного зберігання подій журналів Windows).

Топологія мережі передбачає, що в діапазоні 192.168.100. працюють клієнтські машини, в 192.168.101. – сервери (DC, DBServer), а в 192.168.102. – сервер з модулями виявлення аномалій. Взаємодія з журналами подій реалізована через Windows Event Forwarding (або за потреби через агента Winlogbeat), що відправляє дані до центрального вузла для подальшого аналізу.

4.2 Збір та підготовлення даних

Для навчання та тестування модулів з попередньо обраних алгоритмів ML використовувалися такі категорії даних:

– журнали безпеки Windows (події 4624, 4625, 4768, 4769, 4672, 4688 тощо), що відображають інформацію про успішні та невдалі спроби входу, автентифікацію Kerberos, запуск процесів тощо;

– журнали DNS (події 513, 514), які дають змогу фіксувати спроби доменного сканування та нетипово часті DNS-запити;

– синтетичні дані: штучно згенеровані події для моделювання сценаріїв атак Kerberoasting, PtH, Golden Ticket і бічного переміщення в мережі (lateral movement).

Оскільки різні журнали подій містять відмінні формати записів, для єдиного відображення використовується уніфікована схема, де кожному запису відповідають різні поля. Ця схема використовується для злиття й нормалізації даних із різних журналів подій (Windows Security Log, DNS Log, синтетичні дані тощо). Схему оформлено у вигляді таблиці 4.1.

Таблиця 4.1 – Уніфікована схема для аналізу подій у середовищі AD

Поле	Тип / Формат	Опис
Timestamp	DateTime (UTC або лок.)	Час реєстрації події в універсальному або локальному часовому поясі.
EventID	Ціле число	Унікальний ідентифікатор типу події (наприклад, 4624, 4625, 4768 тощо).
User	Рядок	Обліковий запис (UPN або SID), який ініціював або виконав дію.
SourceHost	Рядок	Ім’я хоста або IP-адреса джерела, з якого надійшов запит.
TargetHost	Рядок	Ім’я хоста або IP-адреса цільового вузла (наприклад, DC, сервер додатка, бази даних тощо).
EventType	Рядок	Тип події (наприклад, Logon, Kerberos TGS Request, LDAP query, DNS query тощо).
ServiceName (SPN)	Рядок	Ідентифікатор службового принципала (Service Principal Name) для подій Kerberos.
Status / Result	Рядок / код	Статус виконання операції (Success / Failure). Може включати розширений код помилки.
AdditionalData	Рядок (Json/Key=Value)	Будь-які додаткові дані (наприклад, номер порту, пакетна статистика, атрибути з DNS- або LDAP-запитів).

Відповідно до таблиці 4.1, маємо такі поля.

– Timestamp – Завдяки уніфікованій часовій мітці можна зіставляти події з різних джерел та встановлювати причинно-наслідкові зв’язки;

– EventID – дає змогу швидко зрозуміти, про який тип дії йдеться (автентифікація, скидання пароля, з’єднання через RDP тощо);

– User – позначення конкретного облікового запису домену або SID (при використанні внутрішнього формату Windows);

– SourceHost – відображає вузол (або IP), який ініціював дію – це може бути клієнтська машина, інший сервер тощо;

– TargetHost – показує, до якого вузла чи служби зверталась ця дія;

– EventType – уніфікований класифікатор дій, що дає змогу алгоритмам ML групувати події за контекстом;

– ServiceName (SPN) – потрібен для ідентифікації Kerberos-служб (зокрема, при аналізі можливих атак Kerberoasting);

– Status / Result – дає можливість відрізнити успішні події від невдалих, що особливо важливо при виявленні Brute Force- або PtH-активності;

– AdditionalData – гнучке поле, яке дає можливість зберегти специфічну інформацію (параметри з журналів, наприклад, “KerbTicketEncryptType=RC4” чи “LDAP Filter=(objectCategory=…)”).

Завдяки такому уніфікованому відображенню, як це подано в таблиці 4.1, всі журнали подій можуть бути зведені в єдиний формат, який зручно піддавати автоматизованому аналізу з допомогою ML-алгоритмів.

У процесі підготовки даних для ML важливим етапом була стандартизація ознак, яка гарантує, що всі ознаки мають однаковий вплив на модель, незалежно від їх початкового масштабу. Це особливо критично в нашому контексті виявлення аномалій в AD, де ознаки можуть включати різноманітні метрики, такі як частота запитів, часові інтервали подій, а також коди помилок. Для нормалізації частотних ознак, зокрема частоти запитів (FR) до AD, застосовувалась мінімаксна нормалізація, масштабуючи їх значення до діапазону від 0 до 1. Цей метод дає змогу уникнути ситуації, коли ознаки з більшими значеннями домінують в процесі навчання моделі. Формально, нормалізація частоти запитів (FR) обчислювалась за формулою:

$$\mathrm{FR}_{\mathrm{norm}} = \frac{\mathrm{FR} - \mathrm{FR}_{\mathrm{min}}}{\mathrm{FR}_{\mathrm{max}} - \mathrm{FR}_{\mathrm{min}}}, \tag{4.1}$$

де $\mathrm{FR}_{\mathrm{norm}}$ – нормалізоване значення частоти запитів, FR – початкове значення частоти запитів, $\mathrm{FR}_{\mathrm{min}}$ – мінімальне значення частоти запитів у навчальній вибірці, і $\mathrm{FR}_{\mathrm{max}}$ – максимальне значення частоти запитів у навчальній вибірці.

Окрім частотних ознак, до стандартизації також підлягали часові ознаки, такі як інтервали між подіями, для яких застосовувався метод стандартизації Z-score. Метод Z-score стандартизує значення Отже, що вони мають середнє значення 0 та стандартне відхилення 1, що є корисним для ознак, що мають розподіл навколо середнього значення та не мають чітких меж. Стандартизація часових ознак обчислювалась за формулою:

$$t_{\mathrm{stand}} = \frac{t - \mu_t}{\sigma_t}, \tag{4.2}$$

де t_{stand} – стандартизоване значення часу, t – початкове значення часової ознаки, μ_t – середнє значення часової ознаки у навчальній вибірці, та σ_t – стандартне відхилення часової ознаки у навчальній вибірці.

Після стандартизації за формулами (4.1)–(4.2), зібрані дані були розділені на дві вибірки: навчальну та тестову. Навчальна вибірка, що складала 80% даних, використовувалася для навчання моделей ML, включаючи Isolation Forest, LSTM, One-Class SVM, GNN та Autoencoder. Тестова вибірка, що становила 20% даних, використовувалась виключно для оцінювання точності навчених моделей і не брала участі в процесі їх навчання. Особливо, для алгоритму One-Class SVM частина експериментів проводилась з однокласним підходом, при якому на навчання подається лише нормальна активність, щоб перевірити результативність даного алгоритму за наявності лише позитивних прикладів.

У підмножині «навчальна» передбачені відмітки про нормальну активність та еталонні приклади аномальної активності (наприклад, модифіковані Kerberos-

квитки). Для One-Class SVM частина експериментів проводилась з однокласним підходом (де на навчання подається лише нормальна активність).

4.3 Методика проведення експерименту

4.3.1 Порядок тестування та сценарії потенційних атак

Для оцінювання точності методу було відтворено найбільш поширені загрози, що описані в розділі 1 та під які спроєктовано метод виявлення аномалій. Нижче формалізуємо ці загрози.

1. Доменне сканування:

– підвищена кількість LDAP та DNS-запитів за короткий проміжок часу (Scouting);

– виявлення у логах великої частки невдалих запитів до AD.

2. Ескалація привілеїв:

– випадки аномальних послідовностей аутентифікації (атака PtH);

– типові Brute Force запити Kerberos (Kerberoasting), коли короткочасна частота TGS-запитів значно перевищує норму.

3. Переміщення мережею (Lateral Movement): зміна патернів доступу до критичних вузлів: GNN мала знайти рідкісні або «неочікувані» шляхи між користувачами та ресурсами.

4. Постійне утримання доступу (Persistence): спроби застосування Golden Ticket або Silver Ticket для доступу до ресурсів, які раніше не належали цьому користувачу. Autoencoder має фіксувати значні відхилення від звичних шаблонів у квитках.

1. Первинний збір нормальних даних (2 тижні). Система працювала в «пасивному режимі» для побудови профілів нормальної активності.

2. Ін’єкція аномалій. По закінченні «навчання» було розгорнуто скрипти, які автоматизовано створювали сценарії атак (Kerberoasting, PtH тощо).

3. Оцінювання та фіксація результатів. Для кожного епізоду атаки оцінювали:

– чи була вона виявлена;

– час спрацювання методу;

– які модулі виявили аномалію;

– кількість хибнопозитивних сигналів.

4. Інтеграція результатів у “Інтерпретатор зв’язків”. Перевіряли, як кореляція між виявленими аномаліями підвищує загальну впевненість у детектуванні атаки.

4.3.3 Метрики оцінювання атак

У процесі оцінювання продуктивності розробленої системи виявлення аномалій в AD, було використано кілька ключових метрик, що дають змогу кількісно оцінити точність, повноту та надійність системи. За підсумками тестування обчислювались такі показники.

1. Accuracy (Точність) – вимірює загальну частку правильних рішень системи Поміж усіх спроб класифікації. Вона показує, наскільки часто модель робить правильні передбачення як для нормальних, так і для аномальних подій. Формула для обчислення точності виглядає так:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{4.3}$$

де TP (True Positives) – кількість правильно класифікованих аномалій, TN (True Negatives) – кількість правильно класифікованих нормальних подій, FP (False Positives) – кількість неправильно класифікованих нормальних подій як аномальних, FN (False Negatives) – кількість неправильно класифікованих аномальних подій як нормальних.

2. Precision (Влучність) та Recall (Повнота). Ці метрики дають більш детальну інформацію про результативність моделі, особливо у випадку незбалансованих даних, де кількість аномалій може бути значно меншою за кількість нормальних подій.

Precision показує, яка частка з усіх виявлених системою аномалій дійсно є аномаліями. Вона вимірює, наскільки точно модель ідентифікує аномалії.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{4.4}$$

Recall показує, яка частка з усіх існуючих аномалій була правильно виявлена системою. Вона вимірює, наскільки повно модель виявляє аномалії.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{4.5}$$

Високе значення влучності свідчить про те, що система робить мало хибнопозитивних спрацювань, а високе значення повноти свідчить про те, що система виявляє більшість аномалій.

3. False Positive Rate (FPR) показує, яка частка нормальних подій була хибно класифікована як аномальні.

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{4.6}$$

Метрики (4.3)–(4.6) є необхідними для повного оцінювання якості роботи розробленої системи виявлення аномалій, вони дають змогу оцінити її здатність точно ідентифікувати аномалії в середовищі AD, мінімізуючи водночас кількість помилкових спрацювань.

4.4 Результати експериментів

4.4.1 Оцінювання за статистичними метриками для різних модулів

В результаті етапу підготовлення даних було згенеровано 1000 вибірок для кожного модуля, усього 5000 вибірок. За проведеними обчислювальними експериментами розв’язано задачі виявлення аномалій в AD як задачі класифікації

для кожного типу атак окремо. Як наслідок, було отримано такі матриці невідповідностей для задачі виявлення аномалій для кожного модуля (рисунок 4.1).

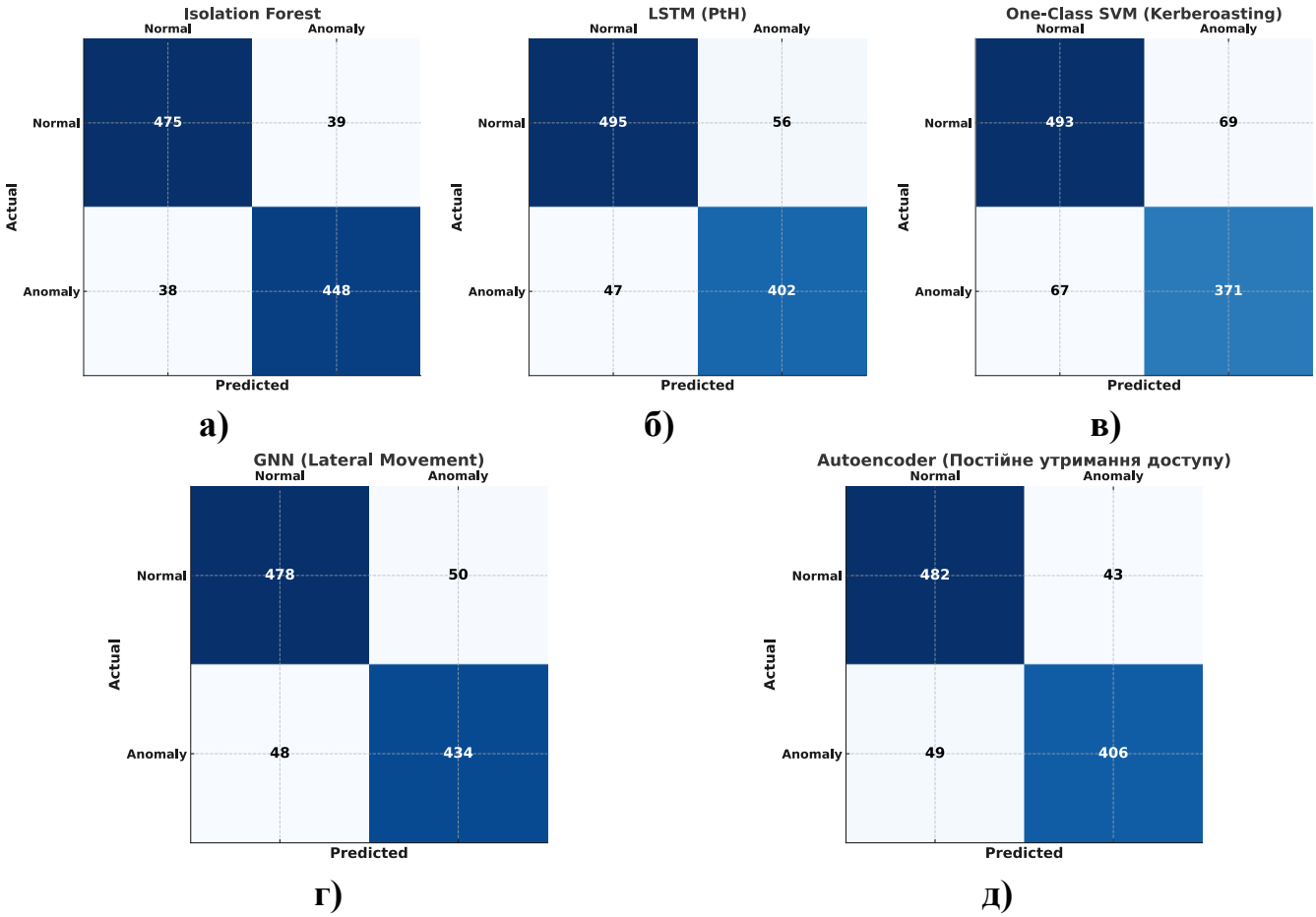


Рисунок 4.1 – Матриці невідповідностей для модулів виявлення аномалій: а) Isolation Forest (доменне сканування); б) LSTM (ескалація привілеїв PtH); в) One-Class SVM (ескалація привілеїв Kerberoasting); г) GNN (lateral movement); д) Autoencoder (постійне утримання доступу).

Матриці невідповідностей на рисунку 4.1 демонструють результати роботи різних модулів виявлення аномалій у завданні аналізу подій. Аналізуючи ці матриці, можна побачити, що всі модулі мають високі показники точності (accuracy), однак розподіл помилок між хибно позитивними (FP) і хибно негативними (FN) результатами варіюється залежно від модулів.

Зокрема, Isolation Forest, який використовується для доменного сканування, має одну з найвищих точностей (94.2%) і найнижчий рівень хибно позитивних результатів (FP = 39). Це свідчить про здатність модуля точно ідентифікувати

нормальні події, хоча невелика кількість хибно негативних результатів (FN = 38) вказує на те, що деякі аномалії залишаються нерозпізнаними.

LSTM, що налаштований для виявлення аномалій типу PtH, демонструє трохи нижчу точність (91.8%), а також значну кількість хибно позитивних (FP = 56) і хибно негативних (FN = 47) результатів. Це може свідчити про складність розпізнавання цього типу атак через близькість шаблонів нормальних і аномальних подій.

One-Class SVM, який спеціалізується на виявленні атак типу Kerberoasting, показує точність 88.5%, що є найнижчим Поміж усіх модулів. Значна кількість хибно негативних результатів (FN = 67) вказує на проблеми у виявленні аномальних подій. Модуль також має порівняно високий рівень хибно позитивних результатів (FP = 69), що може свідчити про необхідність подальшого налаштування моделі.

GNN, що орієнтований на аналіз Lateral Movement, демонструє одну з найкращих характеристик з точністю 92.7% і відносно збалансованим розподілом помилок. Невелика кількість хибно позитивних (FP = 50) і хибно негативних (FN = 48) результатів свідчить про здатність моделі точно справлятися із завданням.

Autoencoder, який використовується для виявлення постійного утримання доступу, має високу точність (93.3%) та демонструє добре збалансовані показники FP (43) і FN (49). Це свідчить про надійність модуля в класифікації подій, хоча деякі аномалії можуть залишатися нерозпізнаними. Нижче наведено таблицю 4.2, що демонструє результати для кожного модуля з урахуванням відповідного типу атаки.

Із таблиці 4.2 можна зробити такі висновки.

1. Isolation Forest показав найкращу точність (94.2%) при виявленні нетипового доменного сканування, демонструючи низький відсоток хибнопозитивних спрацьовувань (4.3%).

2. LSTM добре впорався зі сценаріями PtH (91.8%), але мав дещо вищий FPR (6.1%), ніж Isolation Forest.

3. One-Class SVM для атаки Kerberoasting виявився точним (Recall 82.1%, Precision 85.4%), хоча зіставлення з іншими підходами показує, що цей алгоритм має дещо вищий відсоток помилкових спрацьовувань.

Таблиця 4.2 – Порівняльна оцінка за метриками Accuracy, Precision, Recall та FPR (%) для кожного модуля

Модуль	Типовий сценарій атаки	Accuracy	Precision	Recall	FPR
Isolation Forest	Доменне сканування	94.2	90.5	92.1	4.3
LSTM (PtH)	Ескалація привілеїв	91.8	88.9	87.3	6.1
One-Class SVM	Ескалація привілеїв (Kerb.)	88.5	85.4	82.1	7.7
GNN	Lateral Movement	92.7	86.2	90.3	5.4
Autoencoder	Постійне утримання доступу	93.3	90.7	87.9	4.5

4. GNN підтвердив свою перевагу в завданнях, де важливим є аналіз графа зв’язків. Виявлення lateral movement (кілька послідовних підозрілих переходів між машинами) продемонструвало Recall 90.3%.

5. Autoencoder для визначення підроблених квитків (Golden/Silver Tickets) забезпечив Accuracy понад 93%, а FPR лишається досить низьким (4.5%).

Загалом, результати показують, що кожен модуль добре справляється зі своїм специфічним завданням, хоча є простір для оптимізації. Зокрема, модулям LSTM і One-Class SVM потрібно вдосконалити алгоритми для зменшення хибно позитивних і хибно негативних результатів. Використання додаткових даних або вдосконалення архітектури моделей може сприяти підвищенню їхньої продуктивності.

4.4.2 Аналіз етапу “Інтерпретатор зв’язків”

Для складніших сценаріїв, коли зловмисник послідовно запускає кілька типів атак (наприклад, спочатку Kerberoasting, далі PtH, а пізніше lateral movement), модуль “Інтерпретатор зв’язків” дав змогу підвищити загальну точність системи в середньому на 5–7%. Це досягається за рахунок кореляції результатів: коли декілька модулів вказують на аномалії в одному й тому ж часовому проміжку для однакових облікових записів, ймовірність складної атаки оцінюється як вища.

Середній час оброблення одного запису журналу (Event Log) склав:

– ~0.85 мс у модулі Isolation Forest;

– ~1.20 мс у LSTM/One-Class SVM (необхідність уніфікації подій та обчислення характеристик);

– ~1.40 мс у GNN (попереднє завантаження підграфів та зв’язків);

– ~0.95 мс у Autoencoder.

Такі показники вважаються прийнятними для великого доменного середовища, де надходить до кількох сотень тисяч подій на добу. Оптимізація інфраструктури (наприклад, розподілена обробка, паралельні черги) дає змогу обробляти й більші обсяги логів.

Паралельно для контролю тестувалося звичайне правило: “Якщо за 10 хвилин фіксується більше N невдалих автентифікацій, надсилається тривога”. Такі прості сигнатурні підходи продемонстрували:

– багато хибних спрацьовувань (FPR до 20%), оскільки деякі легітимні групи користувачів або системні служби можуть короткочасно створювати велику кількість невдалих спроб.

– меншу загальну точність (~70–75%), особливо в складніших атаках, які маскуються під звичайну поведінку.

Отже, розглянутий нами підхід із ML випереджає статичні/сигнатурні методи, особливо в складних комбінованих атаках.

Висновок до розділу 4

У четвертому розділі кваліфікаційної роботи проведено експериментальне тестування програмної реалізації запропонованого методу на реальних та штучних наборах даних, що дало можливість підтвердити її результативність у виявленні аномалій у середовищі AD. Запропоновані алгоритми, інтегровані у модульну архітектуру, продемонстрували високу точність і повноту виявлення аномалій, включаючи складні атаки, які можуть маскуватися під типову поведінку. Експериментальні результати підтвердили результативність кожного з модулів (Isolation Forest, One-Class SVM, LSTM, GNN, Autoencoder) у відповідних категоріях атак, а також доцільність використання “Інтерпретатор зв’язків”, який забезпечує інтеграцію та аналіз результатів.

Особливу увагу приділено здатності методу виявляти складні багатоступеневі аномалії, що відповідають комбінованим сценаріям Kerberoasting і lateral movement. Завдяки «інтерпретатору зв’язків» вдалося суттєво знизити кількість як хибнопозитивних, так і хибнонегативних спрацьовувань. Показники продуктивності підтверджують, що час оброблення подій є прийнятним навіть для великих корпоративних мереж, а за рахунок паралелізації модулів є можливість виконання в умовах високого навантаження, обробляючи мільйони записів щоденно.

Разом із тим виявлено певні виклики, зокрема високу ресурсоємність оброблення логів у пікові періоди. Для забезпечення стабільної роботи корпоративної системи у великих організаціях рекомендовано використовувати розподілені обчислювальні середовища або заздалегідь планувати ресурси для оброблення великих обсягів даних. Загалом результати експериментального тестування підтвердили результативність програмної реалізації запропонованого методу для виявлення як стандартних, так і «просунутих» аномалій в AD, що робить цей метод доцільним для впровадження з метою підвищення рівня кібербезпеки на підприємствах.

Загальні висновки

Кваліфікаційна робота магістра подає результати успішно розв’язаної задачі виявлення аномалій в AD засобами ML, що дає змогу підвищити рівень кіберзахисту серверів та баз даних у корпоративних мережах Windows.

Проведений огляд джерел підтвердив провідну роль AD як ключової ланки у забезпеченні централізованої автентифікації, авторизації та управління доступом, що робить AD однією з першочергових мішеней для атак різної складності. Виявлено, що звичайні механізми безпеки на основі простих евристик чи статичних сигнатур часто не охоплюють усіх можливих сценаріїв дій зловмисників, оскільки вони не здатні реагувати на багатокрокові та приховані атаки. Ця ситуація стає особливо небезпечною в умовах ескалації привілеїв і вмілого маскування зловмисника під легітимного користувача, коли типовий нагляд може пропустити критичні ознаки вторгнення.

Запропонований метод виявлення аномалій у другому розділі відзначається збалансованою сукупністю алгоритмів ML та концепцією модульного підходу, що підкреслює різнобічність і гнучкість рішення. Кожен алгоритм ML виконує окреме завдання: одні сконцентровані на пошуку нетипових частот запитів, інші – на виявленні аномальних патернів поведінки користувачів або незвичних послідовностей автентифікації. Така спеціалізація дала змогу підвищити точність виявлення складних, багатокрокових проникнень, сприяючи швидшому виявленню відхилень від норми.

Програмну реалізацію запропонованого методу можна розгорнути у більшості корпоративних середовищ, де вже використовуються різні рішення з моніторингу логів і подій безпеки. Однією з вагомих переваг є гнучка організація збирання даних, що дає змогу легко адаптувати механізми збору й аналізу під потреби конкретної компанії, враховуючи топологію мережі та масштаб домену.

Результати тестування запропонованого методу та його програмної реалізації підтвердили високу точність та надійність обраних алгоритмів і механізму їх кореляційного об’єднання. Під час експериментів було успішно виявлено більшість

відомих методик зламу, зокрема такі, як Kerberoasting, Pass-the-Hash та Golden Ticket, що свідчить про високу результативність запропонованого підходу в реальних умовах.

Під час виконання кваліфікаційної роботи була втілена ідея застосування модульного підходу з декількома алгоритмами ML, кожен із яких орієнтований на окремий аспект безпеки в AD та взаємодіє через інтеграційний «інтерпретатор зв’язків». Така архітектура дала змогу детально аналізувати широкий спектр загроз у середовищі Windows, від банального сканування портів до витончених атак із підробкою квитків Kerberos. Завдяки цьому забезпечується вищий рівень реагування на складні кібератаки, а відтак реалізований метод може бути рекомендований великим організаціям, які прагнуть підвищити загальну стійкість своєї серверної інфраструктури та баз даних.

Поміж виділених обмежень залишається потреба у великих масивах логів, а також значна вимогливість до обчислювальних потужностей, що ускладнює роботу зі швидкими потоками подій у розгалужених доменах. У майбутньому доцільно розгорнути масштабовану, розподілену архітектуру збору та оброблення даних, яка б могла більш оперативно реагувати на зміну обсягів логів у часі. Крім того, інтеграція підходів активного навчання дасть можливість динамічно оновлювати моделі в межах запропонованого методу з урахуванням новітніх форм загроз і суттєво знизити кількість помилкових спрацювань у тривалій перспективі, роблячи цей метод більш адаптивним та перспективним у галузі кібербезпеки.

Перелік посилань

1. Слободян Д. А., Радюк П. М., Цивадиць П. О. Метод виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання. *Актуальні проблеми комп'ютерних наук АПКН-2024* : матеріали XVI Всеукр. науково-практ. конф., м. Хмельницький, 15–16 листоп. 2024 р. Хмельницький, 2024. С. 463–466. URL: https://elar.khmnu.edu.ua/handle/123456789/17152

2. Active directory domain services overview. *Microsoft Learn: Build skills that open doors in your career*. URL: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview (date of access: 17.10.2024).

3. Schwichtenberg H. Active directory. *Windows PowerShell 5 und PowerShell 7*. München, 2020. P. 997–1065. URL: https://doi.org/10.3139/9783446460812.056 (date of access: 17.10.2024).

4. Mastering active directory : design, deploy, and protect active directory domain services for windows server 2022. 3rd ed. Birmingham, UK : Packt Publishing, Limited, 2023. 780 p.

5. Semantic alignment of ontologies meaningful categories with the generalization of descriptive structures / E. A. Manziuk. *Problems in programming*. 2022. Vol. 3, no. 4. P. 355–363. URL: https://doi.org/10.15407/pp2022.03–04.355

6. Metcalf S. How attackers dump active directory database credentials. *Active Directory Security*. URL: https://adsecurity.org/?p=2398 (date of access: 17.10.2024).

7. Active directory: core concepts, architecture and best practices. *Windows OS Hub*. URL: https://woshub.com/active-directory/ (date of access: 17.10.2024).

8. Create collections – configuration manager. *Microsoft Learn: Build skills that open doors in your career*. URL: https://learn.microsoft.com/en-us/mem/configmgr/core/clients/manage/collections/create-collections (date of access: 17.10.2024).

9. Global catalog – win32 apps. *Microsoft Learn: Build skills that open doors in your career*. URL: https://learn.microsoft.com/en-us/windows/win32/ad/global-catalog (date of access: 23.12.2024).

10. КМ: Лекція 22. Архітектура служби каталогів Active Directory. *Інформаційний портал Технічного фахового коледжу*. URL: https://e-tk.lntu.edu.ua/mod/page/view.php?id=3576 (дата звернення: 22.09.2024).

11. Wright G. What is the user principal name (UPN) in Active Directory? – TechTarget Definition. *WhatIs*. URL: https://www.techtarget.com/whatis/definition/User-Principal-Name-UPN#:~:text=In%20Microsoft%20Active%20Directory,%20a,UPN%20is%20[email%20protected]. (date of access: 17.10.2024).

12. Windows authentication overview. *Microsoft Learn: Build skills that open doors in your career*. URL: https://learn.microsoft.com/en-us/windows-server/security/windows-authentication/windows-authentication-overview (date of access: 17.10.2024).

13. NTLM overview. *Microsoft Learn: Build skills that open doors in your career*. URL: https://learn.microsoft.com/en-us/windows-server/security/kerberos/ntlm-overview (date of access: 17.10.2024).

14. Understanding HTTP Authentication – WCF. *Microsoft Learn: Build skills that open doors in your career*. URL: https://learn.microsoft.com/en-us/dotnet/framework/wcf/feature-details/understanding-http-authentication (date of access: 17.10.2024).

15. Mora J. NTLM vs Kerberos | Microsoft community hub. *Techcommunity.Microsoft.COM*. URL: https://techcommunity.microsoft.com/blog/askds/ntlm-vs-kerberos/4120658 (date of access: 17.10.2024).

16. Active directory attacks–steps, types, and signatures / B. I. Mokhtar et al. *Electronics*. 2022. Vol. 11, no. 16. P. 2629. URL: https://doi.org/10.3390/electronics11162629 (date of access: 17.10.2024).

17. Group policy overview for windows. *Microsoft Learn: Build skills that open doors in your career*. URL: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/group-policy/group-policy-overview (date of access: 17.10.2024).

18. Pass the hash attack defense | AD security 101. *Semperis*. URL: https://www.semperis.com/blog/how-to-defend-against-pass-the-hash-attack/ (date of access: 17.10.2024).

19. Smiliotopoulos C., Kambourakis G., Kolias C. Detecting lateral movement: a systematic survey. *Heliyon*. 2024. Vol. 10, no. 4. P. 26317. URL: https://doi.org/10.1016/j.heliyon.2024.e26317 (date of access: 23.12.2024).

20. Privilege escalation attack detection and mitigation in cloud using machine learning / M. Mehmood et al. *IEEE Access*. 2023. Vol. 11. P. 46561–46576. URL: https://doi.org/10.1109/access.2023.3273895 (date of access: 23.12.2024).

21. Net.exe – Win32 apps. *Microsoft Learn: Build skills that open doors in your career*. URL: https://learn.microsoft.com/en-us/windows/win32/winsock/net-exe-2 (date of access: 17.10.2024).

22. Cybersecurity – attack and defense strategies: infrastructure security with red team and blue team tactics. Birmingham, UK : Packt Publishing, 2018. 384 p.

23. Modern_ActiveDirectory: modern active directory. *GitHub*. URL: https://github.com/dakhama-mehdi/Modern_ActiveDirectory (date of access: 17.10.2024).

24. Свобода Р. «Україна прокинеться – і все працюватиме». Детально про «Київстар», атаку, Фрідмана та «російський слід». *Радіо Свобода*. URL: https://www.radiosvoboda.org/a/ukrayina-prokynetsya-i-vse-pratsyuvatyme-kyyivstar-ataka-fridman-rosiyskyy-slid-/32729514.html (дата звернення: 22.09.2024).

25. Кучерявець М. Атака на «Київстар». Що каже СБУ про причетність російських хакерів. *РБК-Украина*. URL: https://www.rbc.ua/rus/news/ataka-kiyivstar-shcho-kazhe-sbu-prichetnist-1702462775.html (дата звернення: 22.09.2024).

26. Radiuk P., Pavlova O., Hrypynska N. An ensemble machine learning approach for Twitter sentiment analysis. *The 6th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2022).* Volume I: Main Conference : CEUR-Workshop Proceedings. Vol. 3171. (Gliwice, Poland, 12–13 May 2022). CEUR-WS.org, Aachen, 2022. P. 387–397. URL: http://CEUR-WS.org/Vol–3171/paper32.pdf (date of access: 18.10.2024).

27. Intelligent data analysis using artificial neural networks for decision making in the education domain / P. M. Radiuk et al. *Herald of Khmelnytskyi National University. Technical sciences.* 2021. Vol. 303, no. 6. P. 111–114. URL: https://www.doi.org/10.31891/2307–5732–2021–303–6–111–114 (date of access: 18.10.2024).

28. Information system for public places and institutions visualization with opportunities of inclusive access and optimal routing / O. Pavlova et al. *Computer systems and information technologies*. 2022. Vol. 1, no. 6. P. 62–68. URL: https://doi.org/10.31891/CSIT–2022–1–8 (date of access: 19.10.2024).

29. Ebad S. A. Lessons learned from offline assessment of security-critical systems: the case of Microsoft's active directory. *International journal of system assurance engineering and management*. 2021. URL: https://doi.org/10.1007/s13198-021-01236-2 (date of access: 17.10.2024).

30. Metcalf S. Finding passwords in SYSVOL & exploiting group policy preferences. *Active Directory Security*. URL: https://adsecurity.org/?p=2288 (date of access: 17.10.2024).

31. Metcalf S. How attackers dump active directory database credentials. *Active Directory Security*. URL: https://adsecurity.org/?p=2398 (date of access: 17.10.2024).

32. Qatinah S. H., Al-Baltah I. A. Kerberos protocol: security attacks and solution. *2024 1st International Conference on Emerging Technologies for Dependable Internet of Things (ICETI)* : Proceedings, Sana'a, Yemen, 25–26 November 2024. New York, NY, USA, 2024. P. 1–7. URL: https://doi.org/10.1109/iceti63946.2024.10777133 (date of access: 17.10.2024).

33. Kotlaba L., Buchovecká S., Lórencz R. Active directory kerberoasting attack: detection using machine learning techniques. *7th International Conference on Information Systems Security and Privacy* : Proceedings, Online, 11–13 February 2021. Setúbal, Portugal, 2021. P. 376–383. URL: https://doi.org/10.5220/0010202803760383 (date of access: 17.10.2024).

34. Enhancing critical infrastructure security: unsupervised learning approaches for anomaly detection / A. Pinto et al. *International Journal of Computational Intelligence Systems*. 2024. Vol. 17, no. 1. P. 236. URL: https://doi.org/10.1007/s44196-024-00644-z (date of access: 17.10.2024).

35. Senturk Z., Irmak E. Persistence techniques in Microsoft active directory: detection and mitigation strategies. *2024 12th International Symposium on Digital Forensics and Security (ISDFS)* : Proceedings, San Antonio, TX, USA, 29–30 April 2024. New York, NY, USA, 2024. URL: https://doi.org/10.1109/isdfs60797.2024.10527234 (date of access: 17.10.2024).

36. M G., Sethuraman S. C. A comprehensive survey on deep learning based malware detection techniques. *Computer Science Review*. 2023. Vol. 47. P. 100529. URL: https://doi.org/10.1016/j.cosrev.2022.100529 (date of access: 17.10.2024).

37. Mailewa A., Rozendaal K. A novel method for moving laterally and discovering malicious lateral movements in windows operating systems: a case study. *Advances in Technology*. 2022. Vol. 2, no. 3. P. 291–321. URL: https://doi.org/10.31357/ait.v2i3.5584 (date of access: 17.10.2024).

38. Vivian Felicite J. G., Ayala-Rivera V., Portillo-Dominguez A. O. Detecting Pass-the-Hash attack in a Microsoft active directory environment using an open-source approach. *2024 12th International Conference in Software Engineering Research and Innovation (CONISOFT)* : Proceedings, Puerto Escondido, Mexico, 28 October – 1 November 2024. New York, NY, USA, 2024. P. 175–184. URL: https://doi.org/10.1109/conisoft63288.2024.00031 (date of access: 23.12.2024).

39. Chacko A. A., Edwin B., Thanka M. R. Detecting the lateral movement in cyberattack at the early stage using machine learning techniques. *Lecture Notes in Electrical Engineering*. Singapore, 2022. P. 581–588. URL: https://doi.org/10.1007/978-981-19-2177-3_54 (date of access: 23.12.2024).

40. Automated Detection of Ransomware in Windows Active Directory Domain Services Using Log Analysis and Machine Learning / B. Keyogeg et al. *Authorea*. 2024. P. 1–10. URL: https://doi.org/10.22541/au.172779663.36925703/v1 (date of access: 17.10.2024).

ДОДАТКИ

Додаток А

Копії наукових публікацій

Актуальні проблеми комп'ютерних наук

УДК 004.4

Слободян Д.А., Радюк П.М., Цивадиць П.О.

Хмельницький національний університет

МЕТОД ВИЯВЛЕННЯ АНОМАЛІЙ В ACTIVE DIRECTORY ДЛЯ ЗАХИСТУ СЕРВЕРІВ ТА БАЗ ДАНИХ ЗАСОБАМИ МАШИННОГО НАВЧАННЯ

У даній роботі запропоновано метод виявлення аномалій у середовищі Active Directory (AD) з використанням алгоритмів машинного навчання. Основна увага приділена вивченню класифікації та виявленню відхилень у поведінці користувачів, що можуть сигналізувати про загрозу безпеці. Запропоновані алгоритми, зокрема One-Class SVM та Local Outlier Factor (LOF), забезпечують виявлення аномальної поведінки в реальному часі, що дозволяє знизити кількість хибнопозитивних спрацьовувань.

This paper proposes a method of detecting anomalies in the Active Directory (AD) environment using machine learning algorithms. The main focus is on studying the classification and identifying deviations in user behavior that may signal a security threat. The proposed algorithms, in particular One-Class SVM and Local Outlier Factor (LOF), provide detection of anomalous behavior in real time, which allows to reduce the number of false positives.

Захист середовища Active Directory (AD) є ключовим аспектом сучасної корпоративної безпеки, оскільки AD керує ідентифікацією та доступом до критичних ресурсів в організації. Однією з найчастіших та найбільш небезпечних загроз є атака Kerberoasting, яка дозволяє викрадати дані без необхідності підвищених привілеїв. Статичні методи виявлення таких атак часто виявляються недостатніми, оскільки не здатні ефективно розпізнавати змінну поведінку зловмисників у реальних мережах [1]. Це створює потребу у використанні гнучких підходів на основі машинного навчання, які здатні адаптуватися до різноманітних сценаріїв аномальної активності та забезпечувати надійніший захист серверів і баз даних.

Попередні дослідження у сфері виявлення атак на середовище AD здебільшого ґрунтуються на статичних правилах або сигнатурних методах для визначення шкідливих дій [2]. Такі методи залежать від заздалегідь визначених умов та порогових значень, які можуть бути неефективними в умовах реальних мереж. Статичні правила часто спричиняють велику кількість хибнопозитивних спрацювань, оскільки вони не враховують відмінності в поведінці користувачів та систем у різних середовищах. Крім того, ці методи можуть бути вразливими до обходу, оскільки зловмисники можуть легко адаптувати свої дії, змінюючи шаблони або зменшуючи частоту запитів, щоб залишатися непоміченими.

Машинне навчання поступово інтегрується як засіб для виявлення аномалій, проте більшість попередніх досліджень обмежується базовими моделями,

які не завжди здатні ефективно розрізняти нормальні та шкідливі дії в AD, особливо під час атак на облікові дані, таких як Kerberoasting [3]. Існує необхідність у розробці методів, які не тільки підвищують точність виявлення, але й знижують кількість помилкових спрацювань, адаптуючись до специфічного середовища організації.

Мною було впроваджено підхід з використанням алгоритмів машинного навчання, зокрема One-Class SVM та LOF, для виявлення атак на облікові дані в Active Directory. Запропонована методика відходить від статичних правил і пропонує адаптивний підхід, який здатен виявляти приховані загрози на основі аналізу аномалій у поведінкових даних AD. Це дозволяє підвищити рівень захисту серверів та баз даних, одночасно зменшуючи навантаження на адміністраторів завдяки скороченню кількості хибнопозитивних сповіщень.

Основною метою дослідження є розробка методу виявлення аномалій у AD для підвищення рівня захисту серверів та баз даних, використовуючи засоби машинного навчання, зокрема алгоритми One-Class SVM та LOF. Завдання дослідження включає адаптацію обраних моделей до особливостей середовища AD, оптимізацію їх продуктивності та зменшення кількості хибнопозитивних спрацювань.

Для виявлення аномальної активності в AD алгоритм One-Class SVM виконує навчання на основі даних, які представляють нормальну поведінку, та виявляє аномалії шляхом визначення віддалених відхилень. **LOF**, навпаки, оцінює локальну щільність даних, що дозволяє виявляти точки з меншою щільністю порівняно з їхніми сусідами. Ключовими ознаками для виявлення аномалій були обрані кількість запитів на окремі послуги, тип облікового запису та IP-адреса джерела (таблиця 1).

Для тестування методів виявлення атак я створив синтетичні дані, які містять ці ознаки, що дозволило натренувати моделі та оцінити їхню ефективність на тестових даних. Це забезпечило більш контрольоване середовище для аналізу продуктивності моделей і дозволило оптимізувати налаштування гіперпараметрів під специфічні умови середовища AD.

Таблиця 1 – Характеристики використаних ознак для виявлення аномалій

Тип ознаки	**Опис**
Кількість запитів	Частота запитів на квитки від окремих користувачів або IP-адрес
Тип облікового запису	Класифікація облікових записів за типом (особистий, неособистий, системний)
IP-адреса	IP-адреса, з якої здійснено запит на квиток, з поділом на сегменти мережі
Тип запитуваної послуги	Характеризує тип доступу (додаток , база даних, тощо)

Результати тестування наведені на рис. 1, рис 2, рис 3.

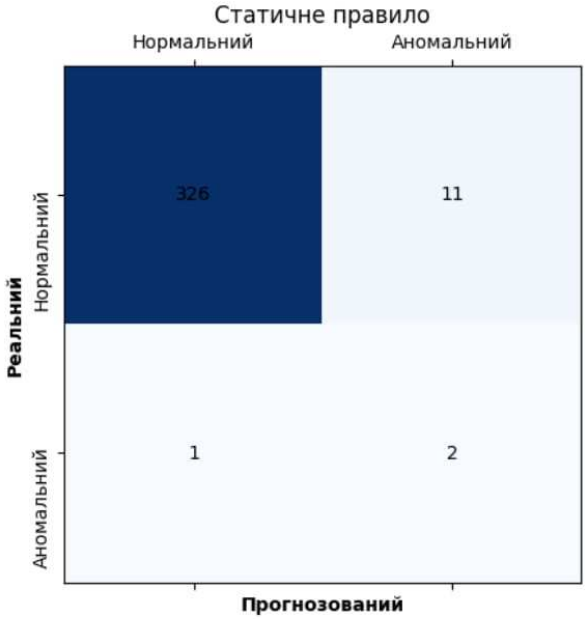


Рисунок 1 – Результати тестування моделі статичним правилом

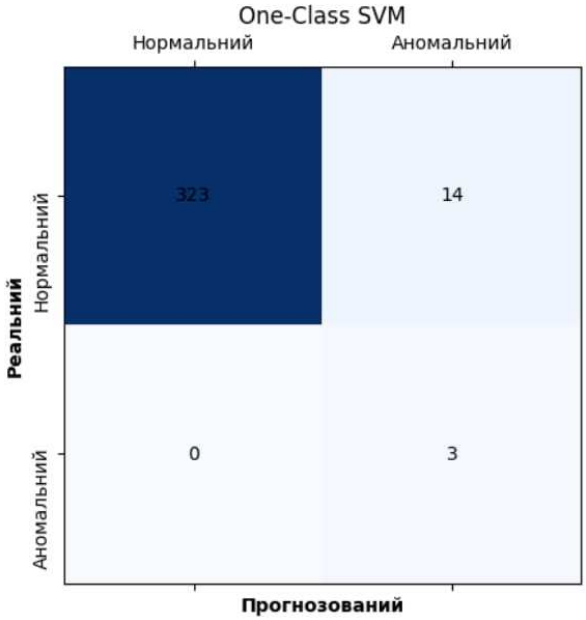


Рисунок 2 – Результати тестування моделі One-Class SVM

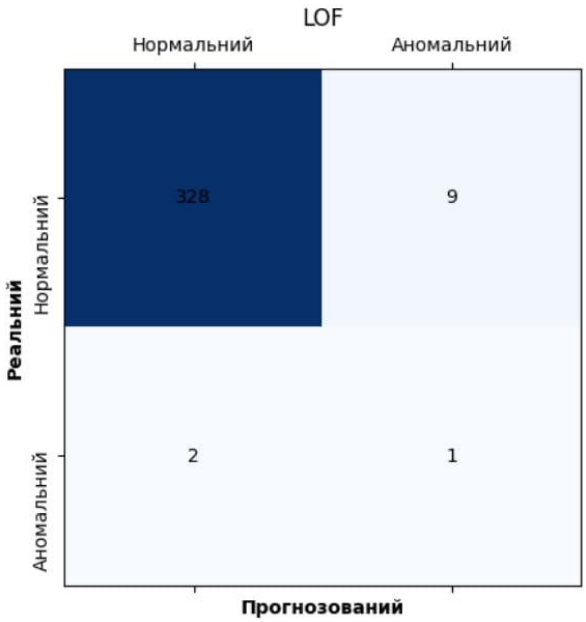


Рисунок 3 – Результати тестування моделі LOF

Результати експериментів показують, що методи машинного навчання зменшують кількість хибнопозитивних спрацювань порівняно зі статичним правилом (рис. 1), яке є менш гнучким у динамічних середовищах. **One-Class SVM** (рис. 2) виявився найбільш ефективним серед розглянутих алгоритмів, оскільки забезпечив вищу точність при мінімальній кількості хибнонегативних спрацювань, дозволяючи точніше ідентифікувати спроби атаки типу Kerberoasting.

Отже, запропонований метод виявлення аномалій у AD забезпечує вищий рівень безпеки завдяки зниженню хибних сповіщень та підвищенню точності виявлення Kerberoasting-атак. Подальші дослідження можуть бути спрямовані на розширення моделей ML, щоб охопити нові типи атак у середовищі AD та оптимізувати алгоритми для роботи в реальному часі.

Перелік посилань

1. Kotlaba, L., Buchovecká, S., & Lórencz, R., Active Directory Kerberoasting Attack: Detection using Machine Learning Techniques, 2021, pp. 376–383.
2. Chandola, V., Banerjee, A., Kumar, V., Anomaly Detection: A Survey, ACM Computing Surveys, Vol. 41, No. 3, 2009.
3. Bulatov, A., Machine Learning for Anomaly Detection in Active Directory, International Journal of Cybersecurity, Vol. 8, No. 2, 2021.

Додаток Б

Лістинг програмного коду

Файл рішення AnomalyDetectionSolution.sln:

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 2022
VisualStudioVersion = 17.0.31903.59
MinimumVisualStudioVersion = 10.0.40219.1

Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "ADAnomalyDetection.Core", "ADAnomalyDetection.Core\ADAnomalyDetection.Core.csproj", "{AAAABBBB-CCCC-DDDD-EEEE-FFFFFFFFFFFF}"
EndProject

Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "ADAnomalyDetection.ML", "ADAnomalyDetection.ML\ADAnomalyDetection.ML.csproj", "{11112222-3333-4444-5555-666677778888}"
EndProject

Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "ADAnomalyDetection.App", "ADAnomalyDetection.App\ADAnomalyDetection.App.csproj", "{9999AAAA-BBBB-CCCC-DDDD-EEEEFFFF0000}"
EndProject

Global
    GlobalSection(SolutionConfigurationPlatforms) = preSolution
        Debug|Any CPU = Debug|Any CPU
        Release|Any CPU = Release|Any CPU
    EndGlobalSection
    GlobalSection(ProjectConfigurationPlatforms) = postSolution
        {AAAABBBB-CCCC-DDDD-EEEE-FFFFFFFFFFFF}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
        {AAAABBBB-CCCC-DDDD-EEEE-FFFFFFFFFFFF}.Debug|Any CPU.Build.0 = Debug|Any CPU
        {AAAABBBB-CCCC-DDDD-EEEE-FFFFFFFFFFFF}.Release|Any CPU.ActiveCfg = Release|Any CPU
        {AAAABBBB-CCCC-DDDD-EEEE-FFFFFFFFFFFF}.Release|Any CPU.Build.0 = Release|Any CPU

        {11112222-3333-4444-5555-666677778888}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
        {11112222-3333-4444-5555-666677778888}.Debug|Any CPU.Build.0 = Debug|Any CPU
        {11112222-3333-4444-5555-666677778888}.Release|Any CPU.ActiveCfg = Release|Any CPU
        {11112222-3333-4444-5555-666677778888}.Release|Any CPU.Build.0 = Release|Any CPU

        {9999AAAA-BBBB-CCCC-DDDD-EEEEFFFF0000}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
        {9999AAAA-BBBB-CCCC-DDDD-EEEEFFFF0000}.Debug|Any CPU.Build.0 = Debug|Any CPU
        {9999AAAA-BBBB-CCCC-DDDD-EEEEFFFF0000}.Release|Any CPU.ActiveCfg = Release|Any CPU
        {9999AAAA-BBBB-CCCC-DDDD-EEEEFFFF0000}.Release|Any CPU.Build.0 = Release|Any CPU
    EndGlobalSection
EndGlobal
```

Файл ADAnomalyDetection.Core.csproj:

```
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
```

```
    <TargetFramework>net6.0</TargetFramework>
    <RootNamespace>ADAnomalyDetection.Core</RootNamespace>
    <AssemblyName>ADAnomalyDetection.Core</AssemblyName>
  </PropertyGroup>

  <ItemGroup>
    <!-- Якщо потрібно: <PackageReference Include="System.DirectoryServices" Version="..."/> -->
  </ItemGroup>

</Project>
```

Клас AdEventRecord.cs:

```
using System;

namespace ADAnomalyDetection.Core.Entities
{
    /// <summary>
    /// Базова модель події (лог-запис) з Active Directory або Windows EventLog.
    /// </summary>
    public class AdEventRecord
    {
        public DateTime Timestamp { get; set; }
        public int EventId { get; set; }
        public string User { get; set; } // UPN або SAM-ім'я
        public string SourceHost { get; set; }
        public string TargetHost { get; set; }
        public string EventType { get; set; }
        public string ServiceName { get; set; } // Kerberos SPN
        public string Status { get; set; }
        public string AdditionalData { get; set; }
    }
}
```

Інтерфейс IDataProvider.cs:

```
using System.Collections.Generic;
using ADAnomalyDetection.Core.Entities;

namespace ADAnomalyDetection.Core.Interfaces
{
    public interface IDataProvider
    {
        /// <summary>
        /// Метод для отримання списку логів з певного джерела (EventLog, бази даних тощо).
        /// </summary>
        IEnumerable<AdEventRecord> GetAdEventRecords();
    }
}
```

Клас WindowsEventLogProvider.cs:

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using ADAnomalyDetection.Core.Entities;
using ADAnomalyDetection.Core.Interfaces;

namespace ADAnomalyDetection.Core.DataProviders
{
    public class WindowsEventLogProvider : IDataProvider
    {
        private readonly string _logName;

        public WindowsEventLogProvider(string logName = "Security")
        {
            _logName = logName;
        }

        public IEnumerable<AdEventRecord> GetAdEventRecords()
        {
            // Приклад зчитування записів із логу Windows:
            var eventLog = new EventLog(_logName);
            foreach (EventLogEntry entry in eventLog.Entries)
            {
                // Спрощене перетворення
                yield return new AdEventRecord
                {
                    Timestamp = entry.TimeGenerated,
                    EventId = entry.EventID,
                    User = entry.UserName,
                    SourceHost = entry.MachineName,
                    TargetHost = "Unknown",
                    EventType = entry.Source,
                    ServiceName = null,
                    Status = entry.EntryType.ToString(),
                    AdditionalData = entry.Message
                };
            }
        }
    }
}
```

Клас IsolationForestModule.cs, що реалізований з бібліотекою ML.NET:

```
using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.ML;
using Microsoft.ML.Data;
using ADAnomalyDetection.Core.Entities;

namespace ADAnomalyDetection.ML.Algorithms
{
```

```
public class IsolationForestModule
{
    private MLContext _mlContext;
    private ITransformer _model;

    public IsolationForestModule()
    {
        _mlContext = new MLContext(seed: 0);
    }

    public void Train(IEnumerable<AdEventRecord> trainingData)
    {
        // Перетворюємо записи у вектор числових ознак.
        var data = trainingData.Select(r => new AnomalyData
        {
            // Приклад: обираємо довільні 2-3 фічі: EventId, HourOfDay, LengthOfMessage...
            EventId = r.EventId,
            HourOfDay = r.Timestamp.Hour,
            MessageLength = r.AdditionalData?.Length ?? 0
        }).ToList();

        var trainDataView = _mlContext.Data.LoadFromEnumerable(data);

        var pipeline = _mlContext.Transforms.Concatenate("Features", nameof(AnomalyData.EventId),
                                                         nameof(AnomalyData.HourOfDay),
                                                         nameof(AnomalyData.MessageLength))
            // У ML.NET немає готового Isolation Forest, тому беремо аналог – Randomized PCA.
            .Append(_mlContext.AnomalyDetection.Trainers.RandomizedPca(
                        featureColumnName: "Features",
                        rank: 2,
                        oversampling: 2))
            ;

        _model = pipeline.Fit(trainDataView);
    }

    /// <summary>
    /// Виконує оцінювання і виявляє аномальні записи
    /// </summary>
    public IEnumerable<(AdEventRecord Record, bool IsAnomaly, float Score)>
Predict(IEnumerable<AdEventRecord> data)
    {
        var dataForPred = data.Select(r => new AnomalyData
        {
            EventId = r.EventId,
            HourOfDay = r.Timestamp.Hour,
            MessageLength = r.AdditionalData?.Length ?? 0
        }).ToList();

        var dataView = _mlContext.Data.LoadFromEnumerable(dataForPred);
        var transformed = _model.Transform(dataView);
```

```
                var predictions = _mlContext.Data.CreateEnumerable<AnomalyPrediction>(transformed,
reuseRowObject: false).ToList();

                // З'єднуємо з оригінальними записами
                return data.Zip(predictions, (rec, pred) =>
                    (rec, pred.Prediction[0] == 1, pred.Score[0]));
            }


            // Допоміжні класи
            private class AnomalyData
            {
                public float EventId { get; set; }
                public float HourOfDay { get; set; }
                public float MessageLength { get; set; }
            }

            private class AnomalyPrediction
            {
                [VectorType(1)]
                public float[] Prediction { get; set; }

                [VectorType(1)]
                public float[] Score { get; set; }
            }
        }
    }
```

Клас OneClassSvmModule.cs:

```
using System.Collections.Generic;
using System.Linq;
using ADAnomalyDetection.Core.Entities;

namespace ADAnomalyDetection.ML.Algorithms
{
    public class OneClassSvmModule
    {
        public void Train(IEnumerable<AdEventRecord> normalData)
        {
            // Сюди можна додати логіку навчання "One-Class SVM".
            // Реально – або власна реалізація, або виклик зовнішньої DLL.
        }

        public IEnumerable<(AdEventRecord, bool)> Predict(IEnumerable<AdEventRecord> data)
        {
            foreach (var rec in data)
            {
                bool isAnomaly = (rec.EventId == 4769) || (rec.EventType?.Contains("TGS") ?? false);
                yield return (rec, isAnomaly);
            }
        }
```

```
    }
}
```

Класи, що реалізують моделі нейронних мереж (LstmModule.cs, GnnModule.cs та AutoencoderModule.cs):

```
namespace ADAnomalyDetection.ML.Algorithms
{
    public class LstmModule
    {
        public void TrainSequentialModel(/*...*/)
        {
            // 1. Ініціалізувати TorchSharp, налаштувати LSTM-шари.
            // 2. Зробити forward/backward pass по батчах логів.
        }

        public bool IsAnomalousSequence(/*певна послідовність*/)
        {
            // Повернути true/false для сигналу аномалії.
            return false;
        }
    }

    public class GnnModule
    {
        public void BuildGraph(/*вузли, ребра*/)
        {
            // Побудова графа AD: вузли = User, Computer, ServiceAccount; ребра = доступи.
        }

        public void TrainGnn(/*...*/)
        {
            // Forward/backward pass GNN через TorchSharp / DGL for .NET (якщо існує).
        }

        public bool DetectLateralMovement(/*...*/)
        {
            // Логіка виявлення, якщо зміни в структурі графа незвичні.
            return false;
        }
    }

    public class AutoencoderModule
    {
        public void TrainAutoencoder(/*...*/)
        {
            // Налаштування шарів енкодера/декодера у TorchSharp.
        }

        public float ReconstructionError(/*дані*/)
        {
```

```
            // Обчислити помилку реконструкції.
            return 0.0f;
        }

        public bool IsAnomaly(float errorThreshold, float currentError)
        {
            return currentError > errorThreshold;
        }
    }
}
```

Файл ADAnomalyDetection.App.csproj:

```
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <TargetFramework>net6.0</TargetFramework>
    <RootNamespace>ADAnomalyDetection.App</RootNamespace>
    <AssemblyName>ADAnomalyDetection.App</AssemblyName>
    <OutputType>Exe</OutputType>
  </PropertyGroup>

  <ItemGroup>
    <ProjectReference Include="..\ADAnomalyDetection.Core\ADAnomalyDetection.Core.csproj" />
    <ProjectReference Include="..\ADAnomalyDetection.ML\ADAnomalyDetection.ML.csproj" />
  </ItemGroup>

</Project>
```

Клас Program.cs:

```
using System;
using System.Linq;
using ADAnomalyDetection.Core.DataProviders;
using ADAnomalyDetection.ML.Algorithms;
using ADAnomalyDetection.Core.Entities;

namespace ADAnomalyDetection.App
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("=== AD Anomaly Detection Demo ===");

            // 1. Завантаження даних
            var provider = new WindowsEventLogProvider("Security");
            var allRecords = provider.GetAdEventRecords().ToList();

            // При бажанні можна поділити на тренувальні (нормальні) і тестові дані
            var trainRecords = allRecords.Where(r => r.Status == "Success").Take(500).ToList();
            var testRecords = allRecords.Skip(500).ToList();
```

```
            // 2. IsolationForest-подібний модуль
            var isolationForest = new IsolationForestModule();
            isolationForest.Train(trainRecords);

            var ifPredictions = isolationForest.Predict(testRecords);
            Console.WriteLine("\n--- IsolationForest Results ---");
            foreach (var (rec, isAnomaly, score) in ifPredictions.Take(5))
            {
                Console.WriteLine($"EvtID={rec.EventId}, IsAnomaly={isAnomaly}, Score={score}");
            }

            // 3. One-Class SVM-подібний модуль
            var oneClassSvm = new OneClassSvmModule();
            oneClassSvm.Train(trainRecords);
            var svmResults = oneClassSvm.Predict(testRecords);

            Console.WriteLine("\n--- OneClassSVM Results ---");
            foreach (var (rec, isAnomaly) in svmResults.Take(5))
            {
                Console.WriteLine($"EvtID={rec.EventId}, IsAnomaly={isAnomaly}");
            }

            // 4. LSTM
            var lstmModule = new LstmModule();
            lstmModule.TrainSequentialModel(/*...*/);
            // Наприклад, перевірити якусь послідовність логів
            bool seqIsAnomaly = lstmModule.IsAnomalousSequence(/*...*/);
            Console.WriteLine($"\nLSTM sequence anomaly = {seqIsAnomaly}");

            // 5. GNN
            var gnnModule = new GnnModule();
            gnnModule.BuildGraph(/*побудова з testRecords*/);
            gnnModule.TrainGnn(/*...*/);
            bool lateralMovement = gnnModule.DetectLateralMovement(/*...*/);
            Console.WriteLine($"GNN lateral movement detected = {lateralMovement}");

            // 6. Autoencoder
            var aeModule = new AutoencoderModule();
            aeModule.TrainAutoencoder(/*...*/);
            float error = aeModule.ReconstructionError(/*...*/);
            bool aeAnomaly = aeModule.IsAnomaly(0.5f, error);
            Console.WriteLine($"Autoencoder anomaly = {aeAnomaly} (Error={error})");

            Console.WriteLine("\n=== Demo Complete ===");
            Console.ReadLine();
        }
    }
}
```

Додаток В

Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

МЕТОД ВИЯВЛЕННЯ АНОМАЛІЙ В ACTIVE DIRECTORY ДЛЯ ЗАХИСТУ СЕРВЕРІВ ТА БАЗ ДАНИХ ЗАСОБАМИ МАШИННОГО НАВЧАННЯ

Виконав:
студент 2 курсу, групи КНм-23-2
Слободян Дітмар Андрійович

Керівник:
к.т.н., доцент кафедри КН
Пасічник Олександр Анатолійович

Актуальність

- Active Directory є ключовою службою каталогів у середовищі Windows, що відповідає за автентифікацію, авторизацію та керування обліковими записами в корпоративних мережах.
- Водночас сервери та бази даних у домені Windows залишаються основними об’єктами атак, адже компрометація Active Directory відкриває шлях до критичної інформації та привілейованого доступу.
- Наявні підходи на основі статичних сигнатур або правил часто не встигають за динамічно змінюваними методами зловмисників та проявами нових кіберзагроз.
- Тому проєктування нового методу, що дасть можливість автоматично відстежувати аномалії в Active Directory та виявляти загрози на ранніх етапах, є особливо актуальною для безпеки серверних середовищ та баз даних.

Мета і задачі роботи

Метою кваліфікаційної роботи магістра є підвищення точності виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання.

Досягнення мети роботи передбачає виконання таких задач:

1. Провести аналіз методів та підходів до виявлення кібератак на інформаційні системи на підприємствах.

2. Спроєктувати метод виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання.

3. Реалізувати метод виявлення аномалій в Active Directory у вигляді модуля програмного забезпечення.

4. Провести експериментальне тестування реалізованого модуля за наборами даних.

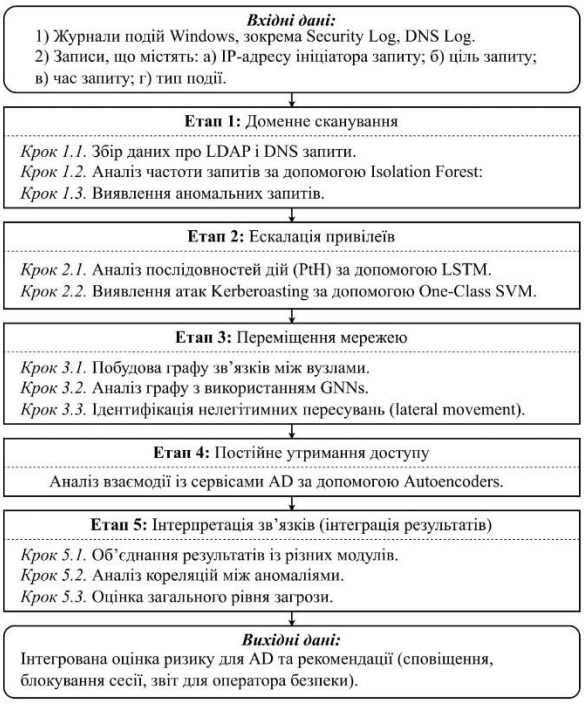
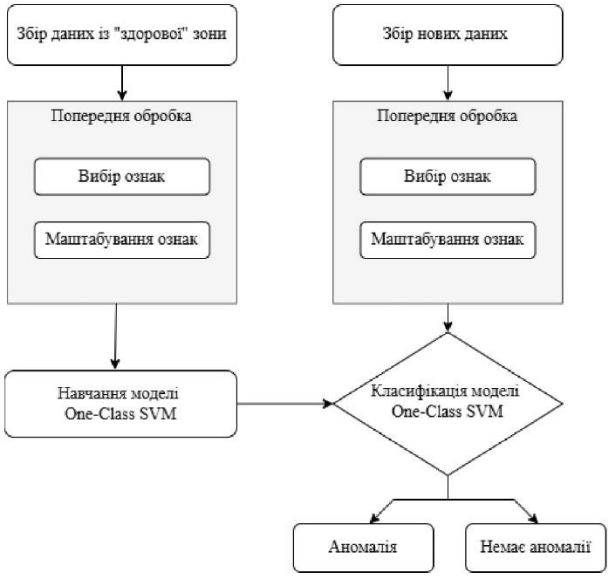


Схема запропонованого методу виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання

Кроки процесу One-Class SVM для виявлення аномалії



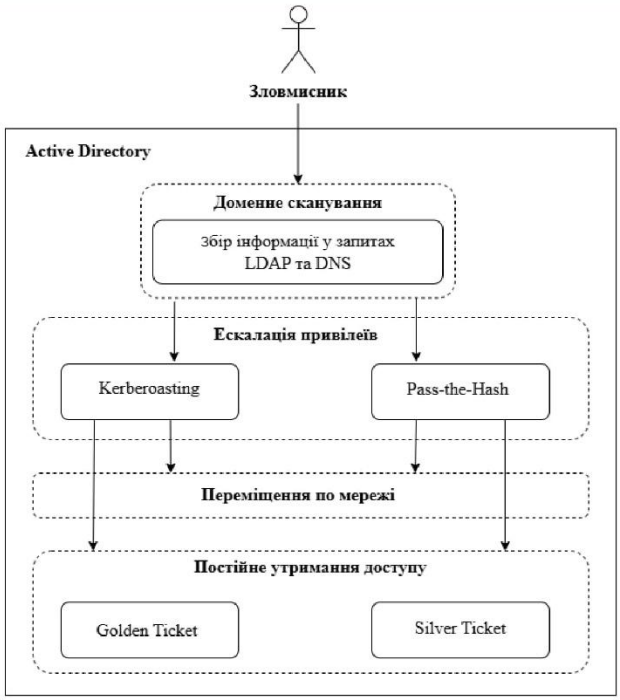


Схема основних типів атак на AD, що розглядаються, відповідно до запропонованого методу

Приклад даних для виявлення атак

ID	Protocol	User	Query_Type	Query_Target	Source_IP	Res	Timestamp	Anom_label
2	DNS	unknown	Add	Domain Admins	227.108.188.45	20	14:00:01	1
5	LDAP	unknown	Search	Domain Admins	98.131.37.171	4	14:00:05	1
12	LDAP	unknown	Search	DNS Records	168.218.183.215	89	14:00:08	1
7	LDAP	serv_acc	Search	DNS Records	52.83.250.111	89	07:14:00	0
35	DNS	user1	Add	Domain Admins	134.58.44.173	13	00:54:00	0

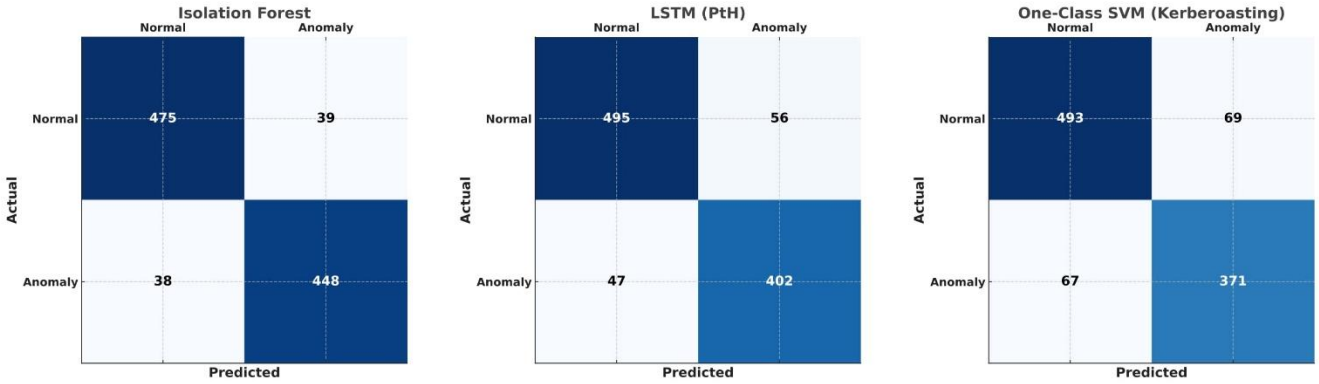
	ID	User	Source_IP	Timestamp	Encryp	Service_Name
0	21	main	192.168.5.15	2024-06-01 12:02:00	0x11	krbtgt
1	5	machine$	192.168.1.20	2024-06-01 12:03:00	0x12	Service01$
2	19	user_tech$	192.168.1.15	2024-06-01 12:01:30	0x12	FileServer01$
3	8	system$	192.168.2.18	2024-06-01 12:04:00	0x12	SQLService$

ID	User	Source_IP	Target_Host	Auth_Method	Pre_Auth	Privil	Time_Of_Attempt	Anom_label
1	adm$svc	192.168.1.10	DC1.corp.local	NTLMv2	0	1	2024-06-01 12:01:00	1
2	unknown	192.168.5.15	DC1.corp.local	NTLMv2	0	0	2024-06-01 12:02:30	1
3	unknown	192.168.6.21	DC1.corp.local	NTLMv2	0	0	2024-06-01 12:04:00	1
4	b_up_srv	192.168.1.20	FileServer01	Kerberos	1	0	2024-06-01 12:05:00	0
5	empl1	192.168.3.18	Workstation03	Kerberos	1	0	2024-06-01 12:06:30	0

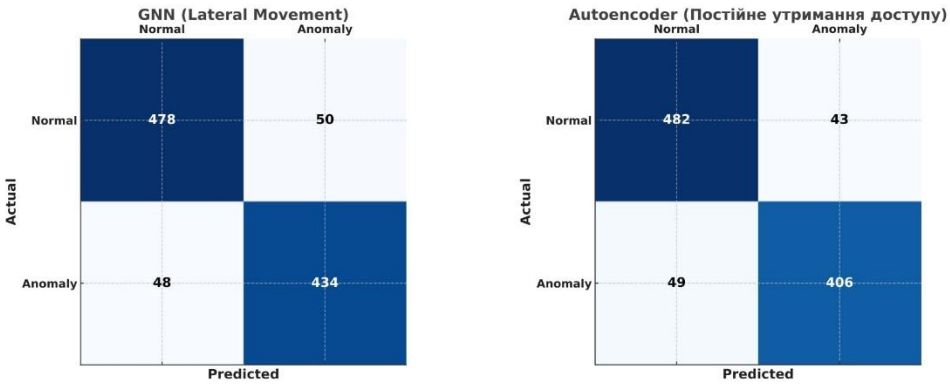
ID	User	Source_IP	Target_IP	Timestamp	Port	Anom_label
11	jsmith	192.168.1.10	192.168.1.12	2024-06-01 12:01:00	445	0
23	jsmith	192.168.1.12	192.168.6.25	2024-06-01 12:02:30	3389	0
34	HR_JohnS	192.168.5.15	192.168.5.20	2024-06-01 12:03:00	5985	1
14	HR_JohnS	192.168.5.20	192.168.6.25	2024-06-01 12:04:30	135	1
58	HR_JohnS	192.168.5.40	192.168.5.45	2024-06-01 12:06:00	135	1

Схема застосування запропонованого методу для виявлення аномалій

Результати задачі виявлення аномалій (1)



Результати задачі виявлення аномалій (2)



Результати задачі виявлення аномалій (3)

Модуль	Типовий сценарій атаки	Accuracy	Precision	Recall	FPR
Isolation Forest	Доменне сканування	94.2	90.5	92.1	4.3
LSTM (PtH)	Ескалація привілеїв	91.8	88.9	87.3	6.1
One-Class SVM	Ескалація привілеїв (Kerb.)	88.5	85.4	82.1	7.7
GNN	Lateral Movement	92.7	86.2	90.3	5.4
Autoencoder	Постійне утримання доступу	93.3	90.7	87.9	4.5

Висновки

1. Кваліфікаційна робота магістра подає результати успішно розв'язаної задачі виявлення аномалій в Active Directory засобами машинного навчання, що дає змогу підвищити рівень кіберзахисту серверів та баз даних у корпоративних мережах Windows.

2. Запропонований метод ґрунтується на модульному підході, де кожен алгоритм машинного навчання (Isolation Forest, LSTM, One-Class SVM, GNN, Autoencoder) спеціалізується на певному типі атак, як от Kerberoasting, Pass-the-Hash або lateral movement.

3. Проведене тестування підтвердило надійність та точність обраних алгоритмів. Запропонований підхід успішно виявляє типові та «просунуті» методи зламу, забезпечуючи високу точність навіть у складних сценаріях, однак має обмеження, що пов'язані з великим навантаженням на обчислювальні ресурси та потребою у значних обсягах логів.

4. Реалізована система адаптована для розгортання в корпоративних середовищах, завдяки гнучкому підходу до збору й оброблення даних.

5. У майбутньому пропонується розвивати систему через впровадження розподіленої архітектури для масштабованої обробки логів, а також інтеграції методів активного навчання.

Thu Dec 26 09:09:41 EET 2024, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 11%**

	Документ		Сумарний збіг по Базі Даних	
ID: 163247 Назва: КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА на тему Метод виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання Додано в БД: 2024-12-26 Автора: Дітмар СЛОБОДЯН Керівники: Олександр ПАСІЧНИК Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	116360	1682	2200 (2%)	35 (2%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Дітмар СЛОБОДЯН

Співавтор:

Назва: Метод виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання

Науковий керівник: Олександр ПАСІЧНИК, к.т.н., доцент

Підрозділ: Кафедра комп'ютерних наук

Коефіцієнт подібності 1:1.9%

Коефіцієнт подібності 2:0.9%

Мікропробіли: 0

Заміна букв: 9

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2024-12-26 09:07:40.0

Після аналізу Звіту подібності констатую наступне:

☑ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

Дата 26.12.2024 експерт Пересвєтов Р.С.

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: *Метод виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання*

Автор: *студент групи КНм-23-2 Слободяна Дітмара Андрійовича*

Спеціальність: *122 Комп'ютерні науки*

Освітня програма: *освітньо-професійна*

Науковий керівник: *к.т.н., доцент кафедри КН Пасічник Олександр Анатолійович*

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	***відповідає***
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Обсяг запозичень, виявлений системами перевірки на збіги, ідентичність чи схожість, становить:

– за системою Anti-Plagiarism: 1.0%: виявлені текстові збіги не є плагіатом, оскільки запозичення містять загальновідомі терміни та скорочення;

– за системою StrikePlagiarism: для КП 1 – 1.9%, для КП 2 – 0.9%: виявлені запозичення відносяться до загальновідомих фраз, термінів та словосполучень та знаходяться у розділі, що присвячений огляду існуючих методів та алгоритмів; знайдені збіги не описують авторських розробок і не стосуються безпосередньо результатів дослідження.

Отже, запозичення є допустимими та відносяться до описаних вище і адресуються до першоджерел, що, з урахуванням наведених обґрунтувань, свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ *Олександр ПАСІЧНИК*

Гарант ОП _____ *Руслан БАГРІЙ*

Завідувач кафедри КН _____ *Олександр БАРМАК*

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук

ВІДГУК НАУКОВОГО КЕРІВНИКА

на кваліфікаційну роботу магістра

студента *гр. КНм-23-2 Слободяна Дітмара Андрійовича*
за темою *Метод виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання*

1. Актуальність теми

Кваліфікаційна робота магістра на тему «Метод виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання» є вкрай актуальною щодо сучасних викликів інформаційної безпеки. Active Directory є основною службою управління ідентифікацією та доступом у корпоративних мережах, через що вона часто стає мішенню складних багаторівневих кібератак. Традиційні підходи, що ґрунтуються на сигнатурах або евристичних правилах, не можуть точно та надійно виявляти нові кібератаки. Натомість застосування машинного навчання дає змогу не лише автоматизувати процес моніторингу, але й значно підвищити точність та швидкість виявлення аномальної активності. Отже, тема роботи відповідає актуальним запитам компаній на рішення з кіберзахисту, особливо в умовах постійного збільшення складності та кількості атак.

2. Відповідність роботи предметній області 122 Комп'ютерні науки та загальним вимогам наукових робіт

Робота повністю відповідає спеціальності 122 Комп'ютерні науки, адже в ній реалізовано добре відомі підходи до інтелектуального аналізу даних та машинного навчання, та досліджено їхнє практичне використання для захисту інформаційних систем. Структура, зміст та оформлення роботи відповідають загальним вимогам до наукових праць.

3. Професійні та особистісні якості магістранта

Під час виконання роботи магістрант Слободян Дітмар продемонстрував високий рівень професійної компетентності, відповідальності та самостійності. Він проявив аналітичні здібності та вміння працювати з великими обсягами наукової літератури, зокрема з теми інформаційної безпеки та машинного навчання. Його наполегливість та орієнтованість на результат відзначилися позитивно на якості роботи.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Робота виконана магістрантом повністю самостійно. Слободян Дітмар самостійно проводив аналіз наукових джерел, обирав відповідні методи, реалізовував запропоновані алгоритми та тестував їх у програмному середовищі. Усі ключові результати, включно із запропонованим методом виявлення аномалій, є особистим внеском магістранта.

5. Наукова новизна та оригінальність запропонованих підходів

Наукова новизна роботи полягає у створенні модульної архітектури для виявлення аномалій в Active Directory із застосуванням кількох методів машинного

навчання. Запропоновані методи включають Isolation Forest для аналізу запитів до LDAP та DNS, метод опорних векторів для виявлення атак типу Kerberoasting, LSTM для ідентифікації Pass-the-Hash, графічні нейронні мережі для аналізу бічного переміщення та автоенкодери для виявлення атак Golden та Silver Tickets. Унікальність підходу полягає в інтеграційному компоненті для аналізу кореляції між аномаліями, що підвищує точність виявлення складних багатоступеневих атак.

6. Ступінь оволодіння методами дослідження

Магістрант продемонстрував глибокі знання та вміння у використанні сучасних технологій машинного навчання. Крім того, він ефективно застосовував ці інструменти для проєктування, програмної реалізації та тестування власного методу. Здатність інтегрувати різні алгоритми в єдину систему, оптимізувати їх та аналізувати отримані результати свідчить про достатній рівень володіння методами дослідження.

7. Повнота та якість розкриття теми роботи

Тема роботи розкрита повно. Виконано аналіз сучасних викликів у кібербезпеці, наведено огляд підходів до виявлення аномалій, реалізовано новий метод і перевірено його результативність на практичних сценаріях. Тому робота повністю відповідає вимогам спеціальності 122 Комп'ютерні науки.

8. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Матеріал викладено логічно та послідовно, із детальними поясненнями кожного етапу дослідження. Аргументація є переконливою, а стиль викладу відповідає науковому підходу. Літературна грамотність магістранта забезпечує доступність та зрозумілість наведених результатів, що позитивно впливає на загальне враження від роботи.

9. Можливість практичного застосування кваліфікаційної роботи, окремих її частин

Запропонований метод може бути використаний для підвищення рівня інформаційної безпеки в корпоративних мережах. Окремі модулі мають перспективи впровадження у системи моніторингу та аналізу активності в Active Directory.

10. Висновок про можливість допуску кваліфікаційної роботи до захисту, на яку оцінку заслуговує робота

З огляду на задовільний рівень виконання, отримані результати та відповідність усім вимогам до кваліфікаційних робіт магістра, вважаю, що кваліфікаційна робота Слободяна Дітмара може бути допущена до захисту.

Рекомендована оцінка – «задовільно».

Керівник _____ *к.т.н., доцент кафедри КН Олександр ПАСІЧНИК*

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук

ВІДГУК ОПОНЕНТА

на кваліфікаційну роботу магістра

студента *гр. КНм-23-2 Слободяна Дітмара Андрійовича*
за темою: *Метод виявлення аномалій в Active Directory для захисту серверів та баз даних засобами машинного навчання*

1. Актуальність обраної теми

Розробка методу виявлення аномалій в Active Directory для захисту серверів та баз даних є важливим та актуальним завданням, враховуючи зростання складності та кількості кібератак у корпоративних мережах. Active Directory відіграє ключову роль в автентифікації, авторизації та управлінні обліковими записами, через що його компрометація може призвести до серйозних наслідків. Використання машинного навчання для автоматизованого моніторингу та аналізу активності в Active Directory є передовим підходом, що дозволяє адаптуватися до нових загроз та підвищити точність виявлення атак. Таким чином, тема роботи повністю відповідає сучасним запитам у сфері інформаційної безпеки.

2. Відповідність роботи предметній області 122 Комп'ютерні науки та загальним вимовам до наукових робіт

Робота Слободяна Дітмара відповідає спеціальності 122 "Комп'ютерні науки", адже вона охоплює як теоретичні, так і прикладні аспекти дослідження. Магістрант використав сучасні методи аналізу даних і реалізував їх у контексті інформаційної безпеки. Усі етапи дослідження, від огляду літератури до експериментального тестування, виконані відповідно до вимог до кваліфікаційних робіт. Робота демонструє задовільний рівень підготовки автора у галузі 12 Інформаційні технології.

3. Повнота розкриття мети та завдань дослідження

Мету та завдання роботи розкрито належно. Автор провів детальний аналіз наявних методів виявлення аномалій в Active Directory, чітко сформулював структуру та логіку нового підходу, реалізував запропоновані алгоритми та перевірив їх результативність на практиці. Результати експериментального тестування підтвердили досягнення поставлених завдань, хоча більш глибокий аналіз обмежень методу міг би покращити розуміння його застосовності.

4. Наявність наукової новизни

Наукова новизна роботи полягає у створенні модульного підходу до виявлення аномалій в Active Directory із залученням таких методів машинного навчання, як Isolation Forest, One-Class SVM, Graph Neural Networks та Autoencoders. Особливо важливим є інтеграційний компонент, який дозволяє проводити кореляційний аналіз аномалій. Такий підхід значно підвищує точність виявлення складних атак, зокрема Pass-the-Hash та Kerberoasting. Втім, детальніше висвітлення порівняння з іншими сучасними методами могло би покращити наукову обґрунтованість запропонованого рішення.

5. Зміст кожного розділу роботи

Робота структурована чітко та логічно. Перший розділ містить аналіз літератури з питань кібербезпеки Active Directory, визначає основні загрози та їхній вплив на корпоративні системи. Другий розділ присвячено проєктуванню методу, зокрема опису вибраних алгоритмів машинного навчання. Третій розділ висвітлює програмну реалізацію та інтеграцію модулів. Четвертий розділ містить експериментальне тестування, аналіз результатів та порівняння з традиційними підходами. Насамкінець, наведено загальні висновки до всієї роботи.

6. Ступінь розкриття теми роботи

Тема роботи розкрита достатньо повно. Автор продемонстрував хороший рівень розуміння предметної області, обґрунтував необхідність створення нового методу та детально описав його архітектуру. Втім, робота могла б виграти від більш детального аналізу обмежень запропонованого підходу, зокрема потенційних проблем із масштабованістю та залежністю результатів від параметрів алгоритмів.

7. Якість оформлення кваліфікаційної роботи

Кваліфікаційна робота оформлена відповідно до стандартів. Всі основні частини, такі як перелік скорочень, вступ, огляд літератури, методологія, результати тестування, висновки та список літератури, подані належним чином.

8. Недоліки оформлення кваліфікаційної роботи

Графічний матеріал роботи іноді потребує уточнення: відсутність підписів осей або недостатньо чітка візуалізація результатів експериментів знижують інформативність поданих графіків. Також бракує таблиці з порівнянням продуктивності методу з іншими сучасними підходами.

9. Недоліки кваліфікаційної роботи

У роботі недостатньо розглянуто питання про можливі помилкові спрацьовування методу та їхній вплив на ефективність. Не висвітлено в достатній мірі, як метод працюватиме у випадку обмеженого обсягу даних для навчання. Також робота могла б виграти від додаткового порівняння запропонованого методу з комерційними рішеннями у сфері кібербезпеки. Разом з тим, вказані недоліки не заперечують доволі хороший рівень виконаної кваліфікаційно роботи та значущість її результатів.

10. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота

Кваліфікаційна робота магістранта Слободяна Дітмара є завершеним дослідженням із чіткою структурою, науковою новизною та практичною цінністю. Враховуючи задовільний рівень виконання та забезпечення всіх вимог до кваліфікаційної роботи магістра, а також позитивні результати тестування, я вважаю, що кваліфікаційна робота може бути допущена до захисту.

Рекомендована оцінка – «задовільно».

Опонент (прізвище, ім'я, по батькові, посада, місце роботи)

Говорущенко Т.О., декан ФІТ КНУ

«18» 12 2024 р.

(підпис)