Міністерство освіти і науки України

«Київський національний університет імені Тараса Шевченка»

Факультет інформаційних технологій

Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
завідувач кафедри кібербезпеки
та захисту інформації
_____ Н.В. Лукова-Чуйко
« » червня 2021р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

дипломної роботи
бакалавра
(назва освітнього рівня)

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

спеціальність 125 Кібербезпека
(код і назва спеціальності)

освітня програма Кібербезпека
(назва освітньої програми)

на тему: «Розробка елементів системи захисту інформації від загроз компрометації облікових даних»

Виконавець: студентка IV курсу, групи КБ-41

_____ **Моклякова Катерина Павлівна**
(підпис) (прізвище ім'я по-батькові)

	Прізвище, ініціали	**Підпис**
Керівник	Ігніска В.І.	

Нормоконтроль	Даков С. Ю.	

Київ 2021

Міністерство освіти і науки України

«Київський національний університет імені Тараса Шевченка»

Факультет інформаційних технологій

Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
завідувач кафедри кібербезпеки
та захисту інформації
_____ Н.В. Лукова-Чуйко
«10» жовтня 2020 р.

ЗАВДАННЯ
на виконання дипломної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

освітньої програми Кібербезпека
(назва освітньої програми)

Студентці **КБ-41** **Мокляковій Катерині Павлівні**
(група) (прізвище ім'я по-батькові)

Тема дипломної роботи Розробка елементів системи захисту інформації від загроз компрометації облікових даних

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Тема дипломної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №2 від 08.10.2020 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Статистичні дані про типи загроз компрометації облікових даних, концепція мультифакторної автентифікації

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Необхідно ознайомитися з методами компрометації та підходами до захисту облікових даних; розглянути підходи до побудови мультифакторної автентифікації; обрати програмний застосунок і розробити архітектуру для системи мультифакторної автентифікації у середовищі з локальною мережею

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практична цінність Запропоновані в роботі рішення можуть бути інтегровані в існуючі системи захисту інформації, як елементи систем автентифікації.

5. ДАТА ВИДАЧІ ЗАВДАННЯ

Дата видачі завдання:12 жовтня 2020 року

Завдання видав		В.І. Ігніска
	(підпис)	(ініціали, прізвище)
Завдання прийняла до виконання		К.П. Моклякова
	(підпис)	(ініціали, прізвище)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
1	Уточнення постановки задачі	25.01.2021 – 01.02.2021	*виконано*
2	Аналіз літератури	01.01.2021 – 15.02.2021	*виконано*
3	Визначення умов генерації факторів автентифікації	15.02.2021 – 25.02.2021	*виконано*
4	Дослідження ймовірності компрометації системи мультифакторної автентифікації	25.02.2021 – 10.03.2021	*виконано*
5	Розробка архітектури мультифакторної автентифікації	01.03.2021 – 25.03.2021	*виконано*
6	Інтеграція доменних сервісів з елементами системи захисту інформації	25.03.2021 – 28.04.2021	*виконано*
7	Інтеграція токенів автентифікації з елементами системи захисту інформації	28.04.2021 – 20.05.2021	*виконано*
8	Оформлення пояснювальної записки	20.05.2021 – 08.06.2021	*виконано*
9	Підготовка до захисту дипломної роботи	09.06.2021 – 21.06.2021	*виконано*

Завдання видала		В.І. Ігніска
	(підпис)	(ініціали, прізвище)
Завдання прийняла до виконання		К.П. Моклякова
	(підпис)	(ініціали, прізвище)

Термін подання дипломної роботи до ЕК 08 червня 2021 року

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Розробка елементів системи захисту інформації від загроз компрометації облікових даних» складається зі вступу, основної частини, що містить 3 розділи, висновків і списку літератури та джерел. Загальний обсяг роботи – 60 сторінок. Робота містить 22 рисунків, 1 таблицю. Список використаних джерел включає 56 джерел.

Об’єкт дослідження – процес захисту облікових даних від компрометації.

Мета роботи – Розробка елементів системи захисту інформації від загроз компрометації облікових даних.

Предмет дослідження – методи системи захисту інформації від загроз компрометації облікових даних.

Метод дослідження – аналіз захисту від загроз компрометації облікових даних.

Практичне значення роботи: запропоновані в роботі рішення можуть бути інтегровані в існуючі системи захисту інформації, як елементи систем автентифікації.

Ключові слова: захист облікових даних, автентифікація, злам акаунту (account break), мультифакторна автентифікація, засіб захисту, захист інформації.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

AD	–	Active Directory
API	–	Application Programming Interface
Bastion	–	Wallix Bastion
IPS	–	Intrusion Prevention System
IT	–	Information Technology
MFA	–	Multifactor authentication
MCA	–	Multi-channel authentication
MDA	–	Multi-device authentication
PAM	–	Privileged Access Manager
SSL	–	Secure Sockets Layer
SaaS	–	Software as a service
TLS	–	Transport Layer Security
VPN	–	Virtual Private Network
ІКС	–	Інформаційно-комунікаційні технології
ПЗ	–	Програмне забезпечення
ЦОД	–	Центр обробки даних

ЗМІСТ

ВСТУП

Відповідно до Закону України “Про захист інформації в інформаційно-телекомунікаційних системах” [1], порядком доступу до інформації в системі є умови отримання користувачем можливості обробляти інформацію в системі та правила обробки цієї інформації. Таким чином, доступ до інформації повинен відбуватися лише авторизованими користувачами, чиї права на перегляд чи редагування будь-яких відомостей були підтверджені з використанням технічних засобів. Однак, загроза несанкціонованого доступу залишається, за даними фонду OWASP - другою найбільшою загрозою веб-додатків у 2021 році є порушення автентифікації [2]. Реалізація функцій автентифікації та управління сеансами, часто виконується неправильно, це дозволяє зловмисникам компрометувати паролі, ключі або токени сеансів, а також використовувати інші недоліки для отримання несанкціонованого доступу до систем. Незважаючи на те, що зловмисникам зазвичай необхідно скористатися методами соціальної інженерії аби викрасти облікові дані, наслідки від реалізації такої загрози несуть руйнівних характер для організації будь-якої сфери діяльності, а особливо для державних установ.

Тому, потреба в дослідженні можливостей покращення сучасних методів захисту інформації спрямованих на збереження облікових даних користувачів інформаційних систем від компрометації, а також розробка елементів системи захисту інформації від даного типу загроз є актуальною в сьогоденні.

РОЗДІЛ 1
ПОНЯТТЯ ОБЛІКОВИХ ДАНИХ, ДОСЛІДЖЕННЯ МЕТОДІВ КОМПРОМЕТАЦІЇ АКАУНТІВ ТА ЇХ ЗАХИСТУ

1.1 Поняття облікових даних та процесу автентифікації

Під обліковим записом, або акаунтом користувача можна розуміти сукупність налаштувань та дозволів, специфічних для нього, яка визначає які дії користувач може чи не може виконувати в межах інформаційної системи, додатку або іншого ресурсу [3]. Ці налаштування також використовуються для захисту даних користувача від доступу інших неавторизованих осіб.

Акаунти користувача, залежно від наданих йому прав можна умовно розділити на 3 групи [4]:

• Адміністратор (також root, superuser) - має повний контроль над інформаційною системою, сервісом чи ресурсом, встановлює чи змінює програмне забезпечення, керує налаштуваннями системи та безпеки. Має повний доступ до конфігурації системи.

• Звичайний користувач - користувач, якому потрібно запускати програми, використовувати функції системи чи сервісу для досягнення своїх цілей. Однак, адміністративні функції для такого типу користувачів обмежені.

• Гість - може використовувати лише програмне забезпечення, або сервіси, згідно встановлених адміністратором дозволів, і не може змінювати системні налаштування. Цей тип акаунтів може повторювати права звичайних користувачів, однак, зазвичай призначений для тимчасового користування.

Отже, модель загроз для різних облікових записів буде відрізнятися як і необхідність додаткових заходів безпеки. Захист акаунтів адміністраторів, які також називають “з підвищенними привілеями” є критично важливим, адже у разі його компрометації скомпрометованою є система та сегмент мережі, де вона розташована. Звичайно, до облікових записів користувачів чи гостьових акаунтів

також повинні бути застосовані заходи безпеки, та пріоритет все одно спрямований на захист облікових даних з підвщенними привілеями. Крім того, організаціям, при автентифікації користувачів на системах, які містять чутливу інформацію, необхідно забезпечувати вищий рівень впевненості (Assurance Level) [5], що заснований на надійності процесу автентифікації та впевненості в тому, що особа насправді є тією, за кого вона себе видає.

Обліковими даними є набір відомостей, які використовуються абонентом для отримання доступу до свого акаунту. Отримання доступу відбувається в декілька етапів:

• Ідентифікація - це процедура, надання суб'єктом ідентифікатора, попередньо зареєстрованого системою.

• Автентифікація - це процес перевірки справжності ідентифікації користувача.

• Авторизація - це функція визначення прав / привілеїв доступу до ресурсів.

Таким чином, облікові дані для авторизації включають унікальний ідентифікатор (наприклад, логін, поштова адреса, номер телефону і т.д.), та один або декілька ключів - факторів автентифікації (пароль, цифровий сертифікат, біометричні дані та ін.).

Розуміння автентифікації користувачів є надзвичайно важливим, оскільки це ключовий етап у процесі захисту конфіденційної інформації від несанкціонованого доступу. Цифрова автентифікація встановлює, що суб'єкт, який намагається отримати доступ до цифрової послуги, контролює один або кілька дійсних автентифікаторів, пов'язаних із цифровою ідентичністю цього суб'єкта [6].

В процесі автентифікації можна визначити 3 суб'єкта [7]. Заявник (Climant) - індивідуум, який повинен довести свою правомірність доступу до ресурсу, довіреніа сторона (Relying party) - постачальник послуг та ресурсів; Верифікатор (Verifier) - система, яка перевіряє легітимність заявників. Коли заявник успішно демонструє володіння та контроль одного або декількох аутентифікаторів верифікатору за

допомогою протоколу автентифікації, верифікатор може підтвердити, що заявник є дійсним абонентом (рис. 1.1).

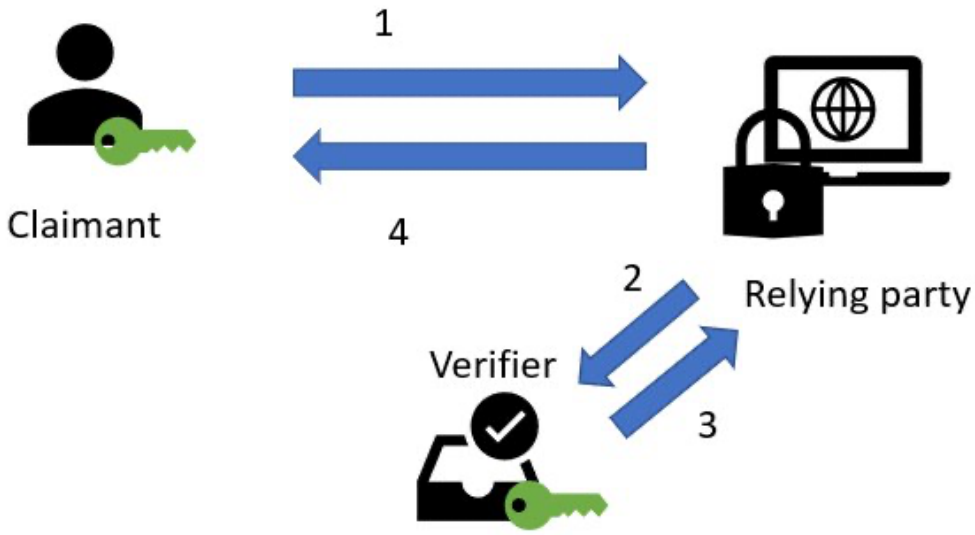

Рисунок 1.1 Процес автентифікації

Важливим моментом є те, що автентифікація не визначає авторизацію заявника або його права доступу. Довірені сторони можуть використовувати автентифіковану ідентичність або атрибути абонента з іншими факторами для прийняття рішень щодо авторизації.

Фактори автентифікації можна розділити на декілька класів, деякі з них (1-3) є широко використовуваними, інші ж застосовуються рідше (4-5):

1. Базуються на знанні: "те, що користувач знає" і стосується методу перевірки ідентичності користувача шляхом зіставлення однієї або декількох "секретних" відомостей, що надаються фізичною особою (заявником), із джерелами інформації, пов'язаними із заявником (наприклад, пароль, пін-код, кодова фраза та інші).

2. Засновані на власності: "те, що ти тримаєш". Цей клас автентифікації базується на токенах, якими володіє користувач. Принцип автентифікації спирається не на пам’ять користувача, а скоріше на здатність користувача довести право власності на токен (фізичні токени, програмні токени, цифровий сертифікат, смарт-карти і т.д.).

3. Базуються на біометричних характеристиках: стосується використання фізіологічних та поведінкових біометричних даних для автентифікації користувачів (відбиток пальця, відбиток сітківки ока, розпізнавання обличчя і т. д.) [8].

4. Залежно від місцезнаходження: незалежне визначення місцезнаходження абонента органом локалізації може гарантувати, що користувач насправді знаходиться в заявленому місці і передати цю інформацію верифікатору (місцезнаходження за даними GPS , Wi-fi, стільникових мереж і т.д.)[9].

5. Залежно від часу: автентифікація на основі часу, коли користувач намагається скористатися ресурсом чи сервісом (будні/вихідні дні, робочі чи неробочі години тощо).

Найбільш поширеним і звичним для користувачів є процес автентифікації за допомогою ідентифікатора (логіна) та пароля (секретної строки). Це приклад однофакторної автентифікації, яка вважається вразливою через велику кількість атак та несанкціонованого доступу до облікових записів, автентифікація яких пов’язана з одним фактором.

Багатофакторна автентифікація використовує принаймні два фактори для перевірки користувача. Крім того, вони повинні походити з різних класів автентифікації. Отже, якщо первинний доступ надається після перевірки пароля, використання ще одного пароля не буде відігравати роль другого фактора.

1.2 Методи компрометації акаунтів

Автентифікація, яка зазвичай базується на імені користувача та паролі, забезпечує зловмисникам простий спосіб отримати доступ до системи [10]. Оскільки паролі - це лише дані, зловмисники мають безліч різних прийомів, які вони можуть використовувати для викрадення пароля без присутності фізично, зокрема:

• Атаки грубої сили

Атака грубої сили - це вид атаки, коли зловмисник використовує підхід спроб і помилок, намагаючись вгадати дійсні облікові дані користувача [11]. Зазвичай ці атаки автоматизовані для перебору списків імен користувачів та паролей.

Автоматизація цього процесу, особливо за допомогою спеціальних інструментів, потенційно дозволяє зловмиснику здійснювати величезну кількість спроб входу з високою швидкістю. Атака грубої силли не завжди складається з випадкових імен користувачів та паролів. Використовуючи базову логіку або загальнодоступні джерела, зловмисники можуть налаштувати атаки грубої сили, щоб зробити адаптовані словники перебору. Це, в свою чергу, значно підвищує ефективність атак. Наприклад, ідентифікатором користувача може виступати його електронна поштова адреса. Як відомо корпоративні адреси будуються за деяким шаблоном, поширеною методикою є: ім'я.прізвище@назва організації.домен, та інші варіації використання ім'я та прізвища. Так як електронна адреса майже завжди відкрито поширюється співробітником, для зловмисника буде легко взначити ідентифікатор користувача.

Сервіси, які покладаються на авторизацію на основі пароля як єдинного методу автентифікації користувачів, можуть бути вразливими, якщо не застосовують достатній захист від атак перебору. До особливостей побудови систем, які породжують вразливості перебору облікових даних можна віднести: статус-коди відповіді HTTP-протоколу, повідомлення про помилки, час відповіді сервісу. Так, наприклад, у випадку повідомлень про помилки, вони можуть повідомляти користувачю, що й ідентифікатор і пароль були введені неправильно, або лише пароль, що вже дає зловмиснику інформацію про правильність підбору логіна.

Атака грубої сили спрямована на статичний, рідко змінюваний пароль, теж може мати успіх. Для захисту, розробник системи може обмежети кількість спроб авторизації для користувача. Однак, такий підхід теж може дати перевагу зловмиснику: повідомлення про блокування, зазвичай тимчасове, однозначно підтверджує правильність ідентифікатора. Таким чином, віднайшовши декілька акаунтів, зловмисник може перебирати пароль для кожного з облікових записів по черзі, уникаючи їх блокування. Іншим підходом, що завтосовується здебільшого веб-додатками, може бути обмеження пропускної здатності користувача, в такому випадку ір-адреса з якої відбувається перебір облікових даних буде автоматично

заблокована. Незважаючи на це, такі інструменти як tor-браузер [12], можуть змінити справжню адресу зловмисника і допомогти йому обійти заходи безпеки.

• Соціальна інженерія

Соціальна інженерія - це термін, що використовується для широкого кола зловмисних дій, які здійснюються завдяки взаємодії людей [13]. Соціальна інженерія використовує психологічні маніпуляції, щоб обдурити користувачів для скоєння порушень політики безпеки або видачі конфіденційної інформації. До технік соціальної інженерії видносяться наступні атаки:

- Фішинг: електронна пошта, що зазвичай має за мету мету - використання підроблених веб-сайтів або форм для викрадення облікових даних користувачів та іншої конфіденційної інформації;

- Претекстинг або сфабрикований сценарій: зловмисник зазвичай вимагає певні відомості від жертви, щоб підтвердити особу, в цей час дані викрадаються і використовуються для викрадення ідентичності користувача;

- Бейтінг (з англ. – наживка) [14]: вид атак, подібний до фішингу, але пропонує користувачу якийсь продукт: фільм, комп'ютерну гру та ін. В обмін на його облікові дані;

- quid pro quo: атаки, які обіцяють вигоду в обмін на інформацію.

Звичайно, список усіх атак, що використовують методи соціальної інженерії не може бути повністю охопленим, зважаючи також на розвиток напряму.

• Фішинг

Так як різновид соціальної інженерії фішинг набув найбільшого поширення та ефективності серед атак, що використовують людський фактор, слід розглянути його детальніше. Актуальність проблеми також підтверджують новини про фішингові атаки і на держустанови України [15].

Електронна пошта є і надалі буде найбільшим вектором загроз для приватних осіб та організацій. Неймовірного поширення фішингові електронні листи набули за часів пандемії [16]. Фішингові кампанії можуть бути персоналізовані й направлені на співробітників окремої організації, або загальними та масово поширеними.

Незалежно від змісту повідомлення, такий лист скоріше за все містить шкідливе посилання, або вкладення. Головною метою фішингово листа є змусити користувача відкрити шкідливе посилання чи файл та викрасти його конфіденційні дані. Одним з прикладів компрометації облікового запису користувача за допомогою фішингу є посилання на фейкову сторінку, що виглядає знайомою користувачеві. Таким чином, жертва авторизується ніби на правомірному ресурсі, який насправді керується зловмисником. Це означає, що будь-які дані введені користувачем на зловмисному сервісі, будуть перехоплені та використані нападником.

Так як людський фактор несе найбільшу вразливість, заходи захисту від фішингових атак можуть лише зменшити ймовірність реалізації загрози. Захист від фішингу включає: навчання та тренінги для усіх співробітників організації, технічні засоби блокування переходу на недовірені домени, захист кінцевих пристроїв від запуску програм прихованих разом з вкладеннями електронних повідомлень тощо.

Використання мультифакторної автентифікації може бути неефективним у разі фішингової атаки та недосконалості функції авторизації ресурсу. Наприклад, веб-ресурс може використовувати авторизаційний токен аби залишати користувача “в системі”. Таким чином, якщо зловмисник зможе перехопити дані авторизаційного токена, то отримає доступ до облівого запису, незалежно від використання мультифакторної автентифікації [17].

• Кейлоггінг

Досить багато систем автентифікації включають введення пароля-тексту як основний спосіб автентифікації в системі. Крім інших загроз, зловмисник може зчитати дані, які вводить користувач під час процесу автентифікації, щоб отримати важливі деталі, необхідні для доступу [18]. Для досягнення цього може використовуватися кейлоггер.

Кейлоггер - це інструмент, який намагається зафіксувати всі натискання клавіш пристрою, на який він встановлений, за умови, що цільовий пристрій має фізичний спосіб введення текстових даних. Кейлоггери в основному використовуються без відома користувача. Незважаючи на це, кейлоггери

правомірно використовуються для нагляду за комп'ютерами компанії на предмет зловживання [19].

Кейлоггери можна поділити на два великі класи: апаратні та програмні [20]. Апаратні кейлоггери - це кейлоггери, які працюють на окремому апаратному пристрої. Зазвичай апаратні пристрої кейлоггера можуть бути приховані або введені в цільову машину таким чином, що їх важко або практично неможливо виявити. Апаратні кейлоггери зазвичай маскуються як периферійні пристрої, що мають спільні інтерфейси, такі як USB або PS / 2, що відповідають інтерфейсам пристрою введення [21]; [22]. Програмні кейлоггери - це програми або скрипти, що існують і виконуються в операційній системі цільової машини. Програмні кейлоггери, на відміну від апаратних, використовують ресурси та пам'ять головної ОС. Кейлоггінг на мобільних пристроях - це відносно нова концепція. Проте, є кілька досліджень та декілька реальних прикладів кейлоггінгу на мобільних пристроях [23]; [24].

Основою захисту від апаратних кейлоггерів є обережність користувачів: регулярна перевірка обладнання на наявність будь-яких модифікацій, моніторинг осіб які мають фізичний доступ до інформаційної системи, а також уникнення використання загальнодоступних та фізично незахищених комп’ютерів. Захист від програмних кейлогерів включає як технічні заходи: використання програм захисту кінцевих пристроїв, так і організаційні: використання лише надійних, перевірених та широко прийнятих джерел для завантаження будь-яких файлів. Звичайно, впровадження захисту від описаного типу загроз компрометації облікових даних є важливим, та варто розуміти, що ці заходи все одно не забезпечують стовідсоткового захисту.

Одним з найбільш перспективних підходом до боротьби від компрометації акаунта за допомогою кейлоггера є використання багатофакторної автентифікації. При її використанні зловмисник може перехопити лише ті фактори автентифікації, які користувач вводить на зараженій інформаційній системі.

• Зберігання пароля відкриим текстом

Вразливості пов’язані зі зберіганням паролей у відкритом вигляді виникли досить давно. Поштовхом до привернення уваги до проблеми була розробка

Бенджаміном Делпі програмного застосунку Mimicatz [25]. За допомогою утиліти можна “витягти” пароль користувача інформаційної системи, зокрема з ОС Windows 7, який на рівні відладки зберігається у відкритому вигляді. Звичайно, з оновленням ця проблема була вирішена на рівні операційної системи. Однак, зберігання та використання паролей відкритим текстом досі є поширеним явищем в різного роду застосунках. Якщо взяти до уваги популярний нині напрямок безперервної розробки (з англ. DevOps [26]), то при налаштуванні деяких інструментів пароль повинен бути або переданий адміністратору застосунку фізично, або за допомогою автоматизованого скрипта, що породжує ризики його викрадення та зловживання.

Поруч зі зберіганням паролей у відкритому вигляді також існує проблема використання паролей за замовчування. Інформаційні технології та системи можуть мати встановлений статичний і однаковий для кожного, хто її використовує, пароль. При конфігурації більшість застосунків вимагають невідкладну зміну пароля за замовченням, однак це не означає, що користувач не встановить той самий пароль.

Таким чином, аби захистити облікові дані від компрометації шляхом експлуатації вразливості пароля у відкритому вигляді, варто використовувати гешування паролей, назначати унікальні паролі з комбінації літер, цифр та спеціальних символів до будь-якого роду облікових записів; також, використання менеджерів паролей спрощує генерацію і зберігання облікових даних користувача. Крім того, у разі недоліків системи, пов’язаних зі зберіганням облікових даних захистити обліковий запис може використання мультифакторної автентифікації.

• Скидання облікових даних

Системи, що дозволяють правомірному користувачеві відновити або змінити забуті облікові дані, аби отримати доступ до ресурса, також несуть вразлисість несанкціонованих дій інших осіб стосовно відновлення доступу до олікового запису.

Системи, які покладаються на "секретні питання", такі як "назва першої школи" або "день народження", є тривіальними. Системи, які надсилають нагадування на резервну електронну адресу або номер телефону, можуть вийти з ладу, якщо користувач змінить адресу або номер, дозволяючи при цьому зареєструвати адресу з тим самим ідентифікатором комусь іншому.

Детально розглянувши загрози компрометації акаунтів та дослідження стосовно їх захисту [27], можемо визначити, що найбільш ефективним методом захисту від несанкціонованого доступу шляхом викрадення облікових даних є використання мультифакторної автентифікації. Однак, підхід до впровадження автентифікації з використанням декількох факторів має свої обмеження і проблеми.

1.3 Мультифакторна автентифікація

Багатофакторна автентифікація (MFA) була вперше запропонована в 1986 році компанією RSA [28]. Однак широкого поширення вона набула останнім часом: фактором прискорення став перехід на віддалену роботу. Пандемія змусила підприємства по всьому світу швидко переконфігурувати свою діяльність, щоб дати можливість працівникам виконувати роботу віддалено, однак це підвищує ризик несанкціонованого доступу до ресурсів організації. Таким чином, криза COVID-19 пришвидшила інтеграцію посиленої автентифікації. В результаті 2021 рік може стати роком, коли багатофакторна автентифікація стане нормою [29]. Компанії розглядають можливості інтеграції другого фактора в архітектуру автентифікації для захисту своїх бізнес-активів. Звичайні користувачі також можуть підвищити захист особистих облікових записів від атак викрадення облікових даних.

Токени автентифікації під виглядом статичних паролей зберігаються провайдером облікових записів у вигляді хешу. Токени, що базуються на власності можуть бути програмними чи апаратними. Стандартний апаратний токен - це невеликий пристрій, як правило, у загальному форм-факторі кредитної картки або підвіски. Найпростіші апаратні токени виглядають ідентично USB-накопичувачу і містять невелику кількість сховища, що зберігає сертифікат або унікальний ідентифікатор, такі токени часто називають ключами [30].

Одноразові паролі (OTP), що генеруються автономним апаратним токеном, можна вважати класичним методом багатофакторної автентифікації, алгоритми обрахування одноразового пароля для апаратних та програмних токенів однакові. Стосовно технічної складової, OTP стандартизований за IETF та компаніями, що займаються верифікацією. Багато рішень OTP є секретними та/або приватними, однак деякі, такі як OATH (зазвичай HMAC-SHA1), є відкритими та широко використовуються та підтримуються між провайдерами OTP. OATH використовує алгоритм OTP, заснований на подіях (також може підтримувати час), який зазвичай використовує секретну послідовність символів спільно з «насінням», відому лише сторонам автентифікації: користувацьке програмне забезпечення, пристрій та сервер

OTP. та номер запиту, а іноді й інші дані, такі як унікальне насіння клієнта тощо. Усі дані запускаються за допомогою алгоритму (зазвичай HMAC-SHA1) для генерації OTP. Синхронізаційні алгоритми генерування OTP бувають наступних типів [31]:

- тип виклику-відповіді OTP: Генератор системи OTP передає секретну фразу користувача, разом із «насінням», отриманим від сервера як частину завдання, через кілька ітерацій захищеної хеш-функції створюється одноразовий пароль.

- синхронізований за часом тип OTP: синхронізовані за часом одноразові паролі, в середині токена знаходиться точний годинник, який синхронізовано з годинником на сервері автентифікації.

- тип OTP, синхронізований з подіями: кожен раз, коли користувач запитує одноразовий пароль на основі події, він збільшує внутрішнє значення лічильника на одиницю. На сервері кожен раз, коли відбувається успішна автентифікація, сервер також збільшує своє значення лічильника на одиницю.

- тип OTP синхронізованого за часом-подією: як у токенах, що базуються на подіях, так і в тих, що базуються на часі, сервер може автоматично виправляти проблеми синхронізації в певних межах. Для токенів, заснованих на подіях, сервер завжди знає нижню межу поточного значення лічильника (тобто значення лічильника, використаного в попередній спробі автентифікації), але не верхню межу. Отже, якщо одноразовий пароль невпізнаний, сервер може спробувати кілька значень лічильника, що перевищують очікуване значення лічильника, щоб перевірити, чи збігаються будь-які з них.

Одноразовий пароль на базі часу (TOTP) - це комп'ютерний алгоритм, який генерує одноразовий пароль, який використовує поточний час як джерело унікальності. Розширення алгоритму одноразового пароля на базі HMAC передбачає, що HMAC - це процес автентифікації повідомлень за допомогою криптографічних хеш-функцій та спільного секретного ключа між сервером та клієнтом.

У прикладі апаратного пристрою, він виконує функцію контексту близькості, що підтверджує доступ користувача до фізичного пристрою. Апаратні

автентифікатори можуть бути двох типів: відключені токени, окремі пристрої, які не мають прямого підключення до клієнтської системи (користувачі повинні вводити ОТР вручну за допомогою клавіатури); та підключені токени, які передають генеровані ТР, чи збережені сертифікати, клієнту через фізичне з'єднання, як правило, універсальну послідовну шину - USB (type-A, type-B, type-C), NFC-чіп, Bluetooth та інші [32].

Використання апаратних токенів передбачає наступне: організація повинна надати користувачам інструкції щодо належного захисту автентифікатора від крадіжки або втрати. Також повинен бути підготований механізм відкликання або призупинення дії автентифікатора одразу після повідомлення абонента про підозру у втраті або крадіжці автентифікатора.

Програмні токени зберігаються на електронних пристроях загального призначення, таких як настільний комп'ютер, ноутбук, КПК і т.д. У випадку з мобільним телефоном програмний токен часто є мобільним додатком. Програмні токени підтримують ОТР, що передається мобільною мережею – SMS (такий підхід вважається небезпечним, через недосконалість мобільних мереж); ОТР, що генерується відповідно алгоритму і є однаковим на сервері автентифікації та в додатку користувача (MobileOTP); push-нотифікації; голосові дзвінки; біометричні дані, за наявності функцій зчитування відбитку пальців, сітківки ока, обличчя (FaceID) та інше.

Створення безпечного середовища завжди корелює із зручністю користувачів. Таким чином, деякі методи захисту від несанкціонованого доступу не використовуються не через їх недосконалість, а через складність для користувачів. Тому, безпека – це компроміс між бажаннями користувачів та захистом інформації. Зважаючи на цей факт, існують обмеження використання програмних токенів, зокрема пов’язаних з мобільними пристроями – різні сервіси підтримують різні додатки (програмні токени), що вимагає від користувача самостійного встановлення підтримки та оновлення деякої кількості додатків. Окрім того, програмні токени для мобільних пристроїв встановлюються на персональні пристрої користувачів, що також не є кращою практикою. Не кожний співробітник компанії має мобільний

пристрій, здатний підтримувати додаток з програмним токеном. З іншого боку, використання мобільних повідомлень чи дзвінків збільшує вартість для організації, так як кожне повідомлення має визначену ціну.

Універсальні автентифікатори такі як Google Authenticator підтримуються значною кількістю додатків та ресурсів, однак виникають питання бекапу токенів, блокування додатку статичним кодом чи біометрією та ін. Залежно від конкретних випадків кращим вибором може бути використання як апаратних, так і програмних токенів. В межах дипломної роботи буде використано програмний токен, що зберігається на мобільному пристрої і генерує одноразові паролі на основі часу. Тому одним із завдань є інтеграція токену з системою автентифікації таким чином, щоб секретний ключ генерації був доступний лише користувачеві і тільки один раз.

1.4 Автентифікація у корпоративному середовищі

При конфігурації надійної автентифікації на початку виникає питання – як реалізувати мультифакторну автентифікацію?

Згідно статистики, більше 80% корпорацій використовують операційну систему Windows для побудови своєї IT інфраструктури [33]. Сервіси Windows - Microsoft Active Directory Domain Services, дозволяють конфігурування і управління директоріями. Директорія – це абстракція, яка показує де в мережі знаходиться той чи інший об’єкт (організація, індивідуум, ресурс). Стандартизований протокол LDAP (Lightweight Directory Access Protocol) спрощує організацію та пошук об’єктів в каталогах. Таким чином, системні адміністратори мають змогу централізовано додавати чи видаляти користувачів, надавати їм права доступу та привілеї для роботи з системою. Зберігання криптографічних ключів у захищеному центральному місці робить процес автентифікації масштабованим та ремонтопридатним. Доменні служби Active Directory - це рекомендована та стандартна технологія для зберігання інформації про особу (включаючи криптографічні ключі, які є обліковими даними користувача). Active Directory вимагає реалізацію NTLM і Kerberos за замовчуванням. Автентифікація Windows

варіюється від простого входу в систему, який ідентифікує користувачів за фактором знання - наприклад, пароля, до більш потужних механізмів безпеки, які використовують щось, що є у користувача - наприклад, токени, сертифікати відкритих ключів та біометричні дані.

Операційна система Windows реалізує набір протоколів автентифікації за замовчуванням, включаючи Kerberos, NTLM, протокол захисту транспортного рівня /рівень захищених сокетів (TLS/SSL) та Digest, як частину розширюваної архітектури. Крім того, деякі протоколи об'єднуються в пакети автентифікації, такі як Negotiate та Credential Security Support Provider [34]. Ці протоколи та пакети забезпечують автентифікацію користувачів, комп'ютерів та служб. Процес автентифікації, у свою чергу, дозволяє авторизованим користувачам та службам отримувати безпечний доступ до ресурсів.

Автентифікація на інформаційній системі під управлінням Windows локально відбувається наступним чином: як тільки користувач ввів облікові дані, вони передаються підсистемі локальної безпеки (LSA), яка одразу генерує хеш пароля. Хешування – одностороннє криптографічне перетворення, що унеможливлює відновлення вихідної послідовності. Таким чином, у відкритому вигляді пароль ніде не зберігається, а єдиним, хто його знає є користувач.

Далі, захисна служба LSA звертається до диспетчера облікових записів безпеки (SAM) і повідомляє йому ім'я користувача. Диспетчер звертається до бази SAM і знаходить хеш пароля вказаного користувача, згенерований при створенні облікового запису (або в процесі використання пароля). Наступним кроком, LSA порівнює хеш введеного пароля і хеш з бази SAM, якщо вони співпадають, автентифікація вважається успішною, а хеш введеного пароля заноситься до сховища служби LSA і знаходиться там до завершення сеансу користувача. Якщо користувач хоче ввійти в домен, автнетифікація відбувається по іншому. Механізм автентифікації Windows AD вимагає використання протоколу Kerberos, однак, якщо одна зі сторін не може використати цей протокол, то можуть бути використані: NT LAN Manager (NTLM), NTLMv2, а також LAN Manager – який не є безпечним на даний момент.

Таким чином, автентифікація на системах під управлінням ОС Windows – має захищенні сценарії, а також проста у використанні. Системи, що входять до домену – групи ресурсів, повинні використовувати протокол Kerberos, або, у разі його недоступності, NTLNv2. Та незважаючи на це, згідно проведених досліджень автентифікація заснована на одному факторі (незалежно від того, який фактор використовується) не є захищенною. Отже, монополізація корпоративної IT інфраструктури продуктами Microsoft є виправданою. Однак, актуальні на даний час протоколи та механізми автентифікації Active Directory не підтримують мультифакторної автентифікації. Тому використовуються різні підходи імплементації посиленої автентифікації.

Незважаючи на менше поширення в корпоративному секторі, все ж варто розглянути процес автентифікації на Linux чи Unix системах. Традиційно Linux та інші Unix-подібні системи просто автентифікували користувачів відповідно запису у файлі /etc/passwd. Кожен користувач мав доступ лише до читання файлу паролів, і зашифровані паролі були доступні кожному, хто мав доступ до системи. Ця проста конструкція зробила файли паролів вразливими до "словникових атак", коли зловмисник шифрував загальні слова та порівнював результат з тими, що були у файлі паролів. Якщо збіг було знайдено, зловмисник знав пароль. Як протидія, Linux та інші Unix-подібні системи перейшли із стандартного файлу пароля на “тіньовий” файл паролів, куди паролі перемістилис із традиційного файлу /etc/passwd в інший (зазвичай /etc/shadow). Оскільки файл /etc/passwd повинен бути доступним для читання будь-яким користувачем системи, переміщення чутливих хеш-кодів паролів із доступного для читання файлу обмежило доступність хешів лише для кореневого користувача. Багато експертів вважають, що наявність єдиного механізму автентифікації для кожної служби в системі (логіни терміналів, локальні входи, входи в мережу тощо) занадто негнучка [35]. Як правило, кожна служба потребує власного коду автентифікації або повинна використовувати єдиний доступний механізм. Модулі аутентифікації, що підключаються (PAM) [36] дозволяють додавати різні модулі для автентифікації нових служб та для додавання нових механізмів автентифікації старих служб. PAM також може бути використаний для

увімкнення автентифікації тіньових файлів для програм, які не підтримують його. Тобто, використання додаткових PAM модулів може забезпечити надійну автентифікацію, хоча і потребує налаштування. Отже, автентифікація на Linux/Unix системах також не підтримує мультифакторної автентифікації без встановлення додактового програмного забезпечення чи використанні проксі-системи.

Окрім локального доступу до системи, також існує необхідність віддалених підключень. Зазвичай, системи Windows використовують протокол віддаленого доступу до робочого столу (RDP), при цьому атентифікація може відбуватися на рівні мережі NLA, або подібно звичайному процесу входу на систему (в цьому випадку, на відміну від попереднього, спочатку відбувається підключення, а потім авторизація). Unix системи використовують протокол захищених сокетів (SSH), аби створити захищену віддалену сесію. Для автентифікації за протоколом SSH, можна обрати власне ключ шифрування, або пароль.

Зважаючи, на проведені дослідження, існує потреба в дослідженні та пошуці шляхів вдосконаленні процесу автентифікації на інформаційних системах для підвищення безпеки і зменшення ризику несанкціонованого доступу.

1.5 Підходи до впровадження мультифакторної автентифікації

Попередні дослідження показали, що практичне впровадження мультифакторної автнентифікації у IT інфраструктурі, потребує використання елементів, які могли б доповнити існуючий механізм підтвердження ідентичності клієнта, або повноцінних систем надійної автентифікації, що цілком заміняють існуючі технології.

Схема побудови надійної автентифікацію, що базується на ідентифікаторі користувача, першому факторі автентифікації, який заснований на знанні – пароль; а також другому факторі, заснованому на власності: програмний чи апаратний токен є найбільш поширеною та оптимальною на даний час. Архітектура такого рішення потребує наступні елементи: механізм автентифікації на системі, або ресурсі,

провайдер облікових даних (ідентифікатора та першого фактору) та провайдер другого фактору, токен автентифікації (другий фактор).

Одним із підходів захисту доступу до інформаційних систем є встановлення додаткового програмного забезпечення – агентних рішень. На перший погляд, така реалізація є зручною і надійною, тому що операційна система не авторизує користувача до того як він пройде автентифікацію за двома чи більше факторами. Агентне рішення може бути ефективним у разі використання кінцевих пристроїв у вигляді корпоративних ноутбуків чи персональних комп'ютерів, однак, вже на цьому етапі необхідно чітко запланувати процес інсталяції та оновлення агентів. Програмне забезпечення, що використовується для мультифакторної автентифікації може бути як частиною VPN-клієнта, так і окремою програмою.

Хоча агентні рішення з надійної автентифікації можуть цілком задовільнити безпеку кінцевих пристроїв звичайних користувачів, виникають питання захисту тих пристроїв, як не підтримують встановлення додаткового програмного забезпечення. Наприклад, мережеве обладнання, що не має веб- інтерфейсу та не дозволяє встановити ПЗ третіх сторін, клас рішень Інтернету речей – що не мають обчислювальної потужності та пам'яті, щоб підтримувати коректну роботу програми з надійної автентифікації та інші, адміністрування яких залишається необхідним.

Аби задовільнити потреби користувачів, зокрема адміністраторів, існує підхід використання централізованої системи автентифікації, яка буде авторизувати користувачів для доступу до ресурсів. Таким чином, можна обрати проксі-рішення, яке буде виконувати підключення до необхідного обладнання по протоолах віддаленого доступу (нарпиклад, RDP SSH і т.д.), разом з тим підтримуючи мультифакторну автентифікацію. Для того, щоб користувачі могли використовувати свої доменні облікові записи також необхідна підтримка інтеграції з LDAP/AD доменом. Проксі-рішення може вирішити проблему надійної автентифікації без необхідності встановлення додаткового програмного забезпечення. Таким чином, одним із завдань дипломної роботи є вибір проксі-рішення, для авторизації

користувачів на системах без необхідності встановлення додаткового програмного забезпечення.

Окрім питання клієнтської сторони побудови надійної автентифікації, виникає також необхідність у провайдері облікових даних та другого фактору. Так як, ні OpenLDAP [37] на системах Unix, ні Active Directory на Windows системах не підтримують використання, а отже і управління та зберігання другого фактору автентифікації, існує потреба у використанні окремого рішення, метою якого є управління токенами автентифікації чи заміна каталогу користувачів повноцінною (такою, що підтримує керування обліковими даними з двома чи більше факторами) системою автентифікації. Однак, у разі використання повноцінного рішення автентифікації виникає ризик її компрометації, що несе за собою негативні наслідки. Так, наприклад, якщо інформаційна система, яка керує автентифікацією була скомпрометована, то і перший і другий фактори не захистять ресурси від несанкціонованого доступу. У разі використання розподіленої моделі, коли перший фактор керується однією інформаційною системою, а другий іншою – при компрометації однієї з них, ресурси залишаються захищеними, так як фактор автентифікації на другій системі залишається дійсним і захищає обліковий запис від несанкціонованого доступу. Беручи до уваги поширення інформаційних систем під управлінням ОС Windows, одним із завдань побудови мультифакторної автентифікації залишається інтеграція чи синхронізація доменних користувачів з проксі-рішенням та провайдером другого фактору.

Провайдер другого фактору - це веб-сервіс, що має деякий інтерфейс, та підтримує обробку HTTP-запитів. Це може бути сервіс, що пересилає повідомлення по СМС або електронною поштою, або сервіс, що генерує коди другого фактора автентифікації, або сервіс, який взаємодіє з користувачем через власне мобільний додаток, і так далі. Важливою складовою є те, що до провайдера можна звертатися за допомогою HTTP-запитів. Таким чином, системи-провайдери другого фактору, які підтримують використання як програмних так і апаратних токенів можуть розміщуватися в локальній інфраструктурі організації, або в хмарному середовищі. Обидва підходи мають свої переваги і недоліки (табл. 1.1)

Таблиця 1.1

Переваги і недоліки розміщення провайдера другого фактора

Розміщення провайдера другого фактора	Переваги	Недоліки
В хмарному середовищі	- Гнучкість ресурсів - Доступність з глобальної мережі - Дешевше, порівняно з локальною інфраструктурою - Відновлення після аварій (Disaster recovery) - Відповідність стандартам на рівні хмарного провайдера	- Необхідність агента на провайдері облікових записів, що буде час від часу синхронізувати акаунти користувачів - Відсутність тотального контролю - Складний процес міграції [38] - Необхідне інтернет-підключення - Безпека та конфіденційність може бути проблемою - Не можна використовувати в державних установах та банкових ІТС [1]; [39] - Завжди платне рішення
Локально	- Доступ до глобальної мережі не обов'язковий - Повний контроль над системою - Більш безпечний ніж хмарне середовище - Підходить до використання в державному секторі - Є безкоштовні та платні рішення	- Необхідність технічної підтримки - Значне інвестування в інфраструктуру - Складніше забезпечити відновлення після аварій та постійний бекап

Проаналізувавши дані, можна визначити, що побудова надійної автентифікації в локальному середовищі – актуальне питання, зокрема, для державного сектору та банкових систем. Так, відповідно закону України [1] – система обробки інформації чи її елементи не можуть розташовуватися на територіях, де влада України не здійснює своїх повноважень. Беручи до уваги те, що провайдери автентифікації зазвичай надають послуги, які знаходться в публічній

хмарі, що, в свою чергу, означає неможливість визначити реальне фізичне розташування обчисювальної системи, де зберігається інформація, використання провайдера другого фактора, який розміщується в хмарному середовищі неможливо для державних установ. Крім того, згідно вимог Національного Банку України [39], використання хмарних сховищ дозволяється лише за умови розташування на території України, що також унеможливлює користування послугами хмарних провайдерів другого фактору.

Отже, окремим завданням дипломного проектування дослідження провайдера другого фактора, що буде розміщуватися в локальній інфраструктурі без необхідності доступу до глобальної мережі та його інтеграція з іншими елементами системи захисту інформації.

1.6 Взаємодія компонентів архітектури надійної автентифікації

Окрім вибору і побудови архітектури системи надійної автентифікації, питання взаємодії компонентів є важливим і також має свої обмеження. Так як автентифікація для користувачів каталогу відбувається централізовано і окремі системи можуть не мати даних автентифікації для авторизації користувача, необхідне використання мережевих протоколів, здатних безпечно передати дані автентифікації.

Розглянемо процес автентифікації за протоколом NTLMv2, це друга версія протоколу NTLM, що є більш покращеною версією свого попередника [40].

- Клієнт виконує запит до сервера, передаючи при цьому ім’я користувача і домена, у відповідь сервер надсилає випадкове число – запит сервера.

- Клієнт генерує випадкове число, куди додається мітка часу. Запит сервера та запит клієнта об’єднуються і від цієї послідовності обчислюється HMAC-MD5 хеш. Після чого, від даного значення повторно береться HMAC-MD5 хеш [32 - 41], ключем котрого є NT-хеш пароля користувача. Отриманий результат має назву NTLMv2-відповіддю і разом із запитом клієнта надсилається серверу.

- Сервер, отримавши NTLMv2-відповідь і запит клієнта об’єднує останній та запит сервера і так само обчислює HMAC-MD5 хеш, після чого передає його та NTLMv2-відповідь контролеру домена. У разі співпадіння контролер повертає серверу відповідь про успішну автентифікацію.

Основним недоліком протоколу NTLM є те, що він не передбачає взаємну автентифікацію клієнта і сервера, це багато в чому обумовлено тим, що протокол спочатку розроблявся для невеликих мереж, де всі вузли вважаються легітимними.

Kerberos - мережевий протокол аутентифікації, який пропонує механізм взаємної автентифікації клієнта і сервера перед встановленням зв'язку між ними. Основу інфраструктури Kerberos складає Центр розподілення ключів (KDC) який є довіренbv центром adтентифікації для всіх учасників мережі. Область Kerberos – це простір імен для яких даний KDC є довіреним, як правило ця область обмежена простором імен домену DNS, в Active Directory область Kerberos збігається з доменом AD. Область Kerberos записується у вигляді відповідного йому доменного імені DNS, але в верхньому регістрі. Обліковим записом Kerberos є будь-який учасник відносин безпеки: обліковий запис користувача, комп'ютер, мережева служба і т.д.

Центр поширення ключів містить довготривалі ключі для всіх облікових записів, в більшості практичних реалізацій Kerberos довготривалі ключі формуються на основі пароля і є так званим "секретом для двох". При цьому довгострокові ключі ні за яких обставин не передаються по мережі і розташовуються в захищених сховищах (KDC), або зберігаються тільки на час сеансу.

У структурі Active Directory центр поширення ключів розташовується на контролері домену, але кожна з сутностей є самостійною і виконує свої функції. Так Kerberos відповідає тільки за автентифікацію клієнтів, тобто засвідчує, що хтось є саме тим, за кого себе видає. Авторизацією, тобто контролем прав доступу, займається контролер домену, в свою чергу дозволяючи або обмежуючи доступ клієнта до того чи іншого ресурсу. Kerberos працює наступним чином:

- бажаючи пройти перевірку справжності в мережі, клієнт передає KDC відкритим текстом своє ім'я, ім'я домену та мітку часу, зашифровані довготривалим

ключем клієнта. Мітка часу в даному випадку виступає в ролі автентифікатора - певної послідовності даних, за допомогою якої вузли можуть підтвердити свою автентичність.

- KDC використовує довготривалий ключ користувача і розшифровує мітку часу, яку порівнює з власним поточним часом.

- Якщо мітка часу дійсна, KDC видає клієнтові сеансовий ключ і квиток (тікет) на отримання квитка (TGT), який містить копію сеансового ключа і відомості про клієнта,

- TGT шифрується за допомогою довгострокового ключа KDC. Сеансовий ключ шифрується за допомогою довгострокового ключа клієнта, а отримана від клієнта мітка часу повертається назад, зашифрована вже сеансовим ключем. TGT є дійсним в обмежений проміжок часу.

- Клієнт розшифровує сеансовий ключ і мітку часу [42]. Клієнт приймає TGT і використовує його при роботі.

Доменні сервіси використовують Lightweight Directory Access Protocol (LDAP протокол) – «легку» версію протоколу доступу до директорій - передбачає меншу кількість коду порівняно з попередником. Протокол прикладного рівня для доступу до служби каталогів X.500., що дозволяє проводити операції автентифікації, пошуку і порівняння, а також додавання, зміни або видалення записів. Одним з механізмів розмежування доступу користувачів та безпеки LDAP є простий рівень автентифікації і безпеки (SASL) – фреймворк для надання аутентифікації і захисту даних в протоколах на основі з'єднань. SASL має три режими роботи:

- Без автентифікації: анонімний доступ;

- Проста автентифікація: користувач надає ідентифікатор у вигляді Distinguished name та пароль (відкритим текстом);

- Рівень простої автентифікації та безпеки: клієнт і сервер «домовляються» про механізм безпеки (Kerberos, TLS та ін.)

Механізмом безпеки для LDAP є LDAPS (LDAP з використанням SSL). Протокол LDAP спроектований так, щоб надсилати інформацію у відкритому вигляді. Таким чином, використання LDAP «над» SSL надає послуги шифрування.

LDAPS звичайно використовується в середовищі Microsoft, базою даних Active Directory, однак є сенс фільтрації трафіку для певності використання лише захищеної версії протоколу LDAP. Незважаючи на простоту використання LDAP/LDAPS він не реалізує підтримки другого фактора автентифікації.

Іншим протоколом автентифікації є OAuth 2 - це система авторизації, яка дозволяє програмам отримувати обмежений доступ до облікових записів користувачів у службі HTTP, таких як Facebook, GitHub та DigitalOcean [43].

Основні етапи процесу авторизації за протоколом OAuth2:

- Додаток запитує авторизацію для доступу до ресурсів служби у користувача.

- Якщо користувач підверджується, програма отримує дозвіл на авторизацію.

- Додаток запитує токен доступу від сервера авторизації (API). Це робиться шляхом представлення його ідентичності та дозволу на отримання дозволу.

- Якщо ідентифікація програми автентифікована і надання авторизації є дійсним, API видає токен доступу до програми. Авторизація завершена.

- Додаток запитує ресурс у API та представляє токен доступу для автентифікації.

- Якщо токен є дійсним, API подає ресурс додатку.

Мова розмітки тверджень безпеки (SAML) [44] - це формат відкритих стандартних даних на основі XML для обміну даними автентифікації та авторизації між сторонами: постачальником ідентифікаційних даних та постачальником послуг.

Типові етапів процесу автентифікації за протоколом SAML:

- Користувач отримує доступ до віддаленого додатку за допомогою посилання.

- Додаток ідентифікує походження користувача (за субдоменом, IP-адресою користувача тощо), перенаправляє користувача назад до постачальника ідентифікаційних даних, запитуючи автентифікацію.

- Користувач або має існуючий активний сеанс браузера у постачальника ідентифікатора, або встановлює його, увійшовши до акаунту.

- Постачальник ідентифікаційних даних формує відповідь на автентифікацію у вигляді XML-документа, що містить ідентифікатор користувача. Цей документ

підписується за допомогою сертифіката X.509, а потім надсилається постачальнику послуг.

- Постачальник послуг (який уже знає постачальника ідентифікаційних даних і має відбиток його сертифіката) отримує відповідь автентифікації та перевіряє її за допомогою відбитка сертифіката.

- Особа користувача встановлюється, і користувачеві надається доступ до програми.

Служба віддаленої автентифікації (RADIUS) - це мережевий протокол, який забезпечує централізоване управління автентифікацією, авторизацією та обліком користувачів, які підключаються та використовують мережеві послуги.

- Автентифікація RADIUS починається, коли користувач запитує доступ до мережевого ресурсу через сервер віддаленого доступу (RAS). Користувач вводить ім’я користувача та пароль, які RADIUS сервер зашифровує перед надсиланням для процесу автентифікації.

- Сервер RADIUS перевіряє точність інформації, шляхом порівняння наданої користувачем інформації з локально збереженою базою даних або посиланням на зовнішні джерела, такі як сервери Active Directory.

- Сервер відповідає, прийнявши, кинувши виклик або відхиливши запит користувача. У разі виклику RADIUS-сервер запитує у користувача додаткову інформацію для підтвердження його ідентифікатора користувача.

Розглянуті протоколи: LDAP, Kerberos, OAuth2, SAML та RADIUS є корисними для різних цілей авторизації та автентифікації та часто використовуються разом із системою єдиного входу. Згідно завдань дипломної роботи, розробка елементів надійної автентифікації буде базуватися на каталозі AD, що використовує LDAP, проксі-рішенні, та провайдері другого фактору, який підтримує RADIUS-протокол.

1.7 Завдання дипломної роботи

Завданнями дипломної роботи є:

- Аналіз літератури;
- Визначення умов генерації факторів автентифікації;
- Дослідження ймовірності компрометації системи мультифакторної автентифікації;
- Розробка архітектури мультифакторної автентифікації;
- Інтеграція доменних сервісів з елементами системи захисту інформації;
- Інтеграція токенів автентифікації з елементами системи захисту інформації.

Висновки за розділом 1

У першому розділі було проведено аналіз джерел, що описують загрози компрометації облікових даних, а саме – атаки грубої сили, кейлогери, соціальна інженерія, скидання облікових даних та інші, і визначено, що найкращим захистом від компрометації облікових даних є використання мультифакторної автентифікації.

Крім того, було проведено дослідження підходів та обмежень до побудови надійної автентифікації, що може захистити інформаційні системи від несанкціонованого доступу шляхом компрометації акаунтів.

Таким чином, за результатами першого розділу було визначено завдання дипломної роботи і виконано аналіз літератури.

РОЗДІЛ 2
ОСОБЛИВОСТІ МУЛЬТИФАКТОРНОЇ АВТЕНТИФІКАЦІЇ ВІДНОСНО КАНАЛУ ПЕРЕДАЧІ ТА ПРИСТРОЮ

2.1 Мультиканальна автентифікація

Незважаючи на те, що мультифакторна автентифікація несумнівно є останнім рубежом захисту облікового запису – навіть якщо деякі облікові дані були викрадені, використання декількох факторів не дозволяє зловминсику атворизуватися на системі, MFA все ж має свої обмеження та проблеми безпеки. Ні кожен метод другого фактора, ні архітектура авторизації не є ідеально захищеними, тому дослідження продвинутої конфігурації системи мультифакторної автентифікації є актуальними.

Багатоканальна автентифікація (MCA) - це тип MFA, який передбачає використання принаймні двох різних каналів передачі. Слід враховувати, що токени можуть бути такими, що передаються мережею (веб чи не-веб), і такими, що не передаються – «витягуваними».

Поряд з використанням декількох каналів, можна взяти за приклад деякі підприємства, які розширили процес автентифікації на багаторівневу автентифікацію [45]. Вона працює наступним чином: для використання деяких функцій, сервіс вимагає додаткові кроки автентифікації. Наприклад, деякі банки дозволяють користувачам авторизуватися до свого банківського рахунку, використовуючи єдиний пароль. Однак, для оплати рахунків за комунальні послуги або переказу коштів, користувач повинен надати інший пароль або парольну фразу для авторизації транзакції. Можна простежити, що багаторівнева модель використовує одноканальну автентифікацію з використанням двох паролів, що не є надійним на сьогодні.

Кожен канал передачі має свої проблеми безпеки. Якщо канал, який може бути скомпрометований, використовується як основний канал, система повинна

використовувати багатоканальну функціональність, для усунення слабких сторін каналу. Якщо канал вважається надійним, авторизація може бути здійснена за допомогою того самого каналу. Інакше кажучи, мультифакторна автентифікація може бути здійснена за допомогою одного каналу передачі, якщо він вважається надійним. Різні канали автентифікації мають різний ризик бути скомпрометованими [46]. Можливі канали для отримання фактора автентифікації можуть включати push-сповіщення, програмний токен, SMS / E-mail OTP тощо (рис. 2.1).

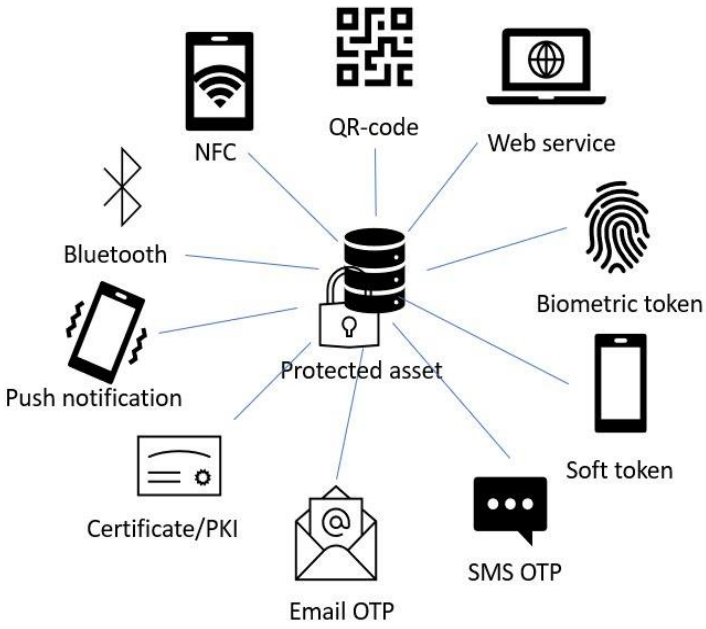

Рисунок 2.1 Різні канали для отримання автентифікатора

Ми також можемо розділити автентифікатори на проштовхувані до користувачів – наприклад, користувач приймає push-повідомлення, або такими, що надаються - коли користувач надає свій цифровий сертифікат. Однак ризик клонованих або викрадених "витягнутих" токенів залишається, так само як і атаки на канал передачі.

У разі використання мобільної мережі - SMS OTP, що доступний по всьому світу. Використовуючи SS7 або Over-The-Air Man-In-The-Middle перехоплення SMS, OTP, отримані з перехопленого SMS, можуть бути використані зловмисником для отримання несанкціонованого доступу до облікових записів користувачів.

Зловмисник може використовувати перехоплений OTP для відновлення паролів / PIN-кодів для облікових записів або в поєднанні з атакою Неструктурованих додаткових службових даних, що перемикає номер телефону, пов'язаний з обліковим записом [46]. Отримавши доступ до SS7 мережі та номер телефону жертви, зловмисник може підслухати розмову, визначати місцезнаходження людини, перехоплювати повідомлення, щоб отримати доступ до послуг мобільного банкінгу і т.д. [47].

Іншим прикладом може бути технологія Bluetooth. Bluetooth вперше був представлений в 1998 році, і з тих пір набув поширення та був включений майже в усі пристрої, які можуть зберігати особисті дані, а також у багато інших, які цього не роблять (наприклад, пристрої IoT). Радіостанції Bluetooth можна знайти в ноутбуках, настільних комп'ютерах, стільникових телефонах, телевізорах, клавіатурах, "розумних" тостерах і навіть скіммерах кредитних карток. Якщо для передачі даних потрібне з'єднання з бездротовою мережею короткого діапазону, можливо використовувати технологію Bluetooth. І з точки зору безпеки, ця технологія є відносно безпечною, оскільки більшість людей вважає, що вона має невеликий діапазон дії. Однак, насправді діапазон дії залежить від ряду факторів, включаючи радіоантену в пристрої. Щоб скомпрометувати канал передачі - Bluetooth, зловмисник повинен знаходитися в радіусі дії пристрою та радіостанції Bluetooth, щоб мати можливість використовувати його. Хакери можуть отримати доступ до пристрою за допомогою декількох методів, наприклад:

- Bluebugging. Bluebugging - це тип атаки Bluetooth, за допомогою якої хакери можуть отримати доступ до пристрою та підслуховувати телефонні дзвінки, підключатися до Інтернету, надсилати та отримувати текстові повідомлення та електронні листи та навіть здійснювати дзвінки. Зазвичай ця вразливість присутня в старих моделях телефоних пристроїв.

- Блюджек. Це найбільш поширений тип атаки Bluetooth і досить нешкідливий, оскільки хакер може надсилати спам у вигляді текстових повідомлень на зламаний пристрій. Bluejacking не дає хакерам доступ до смартфона або даних на ньому.

- Блюзнарфінг. Хакери можуть здійснювати атаку bluesnarfing на пристрої, коли вони перебувають у радіусі близько 90 метрів. Це одна з найнебезпечніших атак Bluetooth, оскільки, навіть якщо ваш пристрій перебуває в режимі, який неможливо виявити, хакери можуть атакувати його та отримати доступ до всієї особистої інформації у вашому пристрої. Використання цієї вразливості дає зловмисникам можливість копіювати всю інформацію на пристрої жертви, включаючи фотографії та відео, номер телефону, список контактів, електронні листи та паролі. Однак, режим невидимки ускладнює зловмисникам з'ясування моделі та назви вашого пристрою [48].

Окрім наведених атак, можна згадати вразливості в IoT пристроях. З точки зору безпеки пристроїв IoT та використання їхніх вразливостей, Bluetooth - це сфера, яку часто ігнорують. З точки зору безпеки, Bluetooth як правило, сприймають за нешкідливу технологію через її досить короткий діапазон. З'єднання Bluetooth захищено шифруванням, але хакери продовжують знаходити та використовувати вразливі місця. Як тільки хакер отримує доступ до пристрою з підтримкою Bluetooth, він потенційно може отримати доступ до даних на цьому пристрої, а також доступ до інших пристроїв в мережі.

Зв'язок на невеликих відстаня (NFC) - це набір стандартів для мобільних пристроїв, призначених для налагодження радіозв'язку між ними шляхом дотику або переміщення на невеликій відстані. Стандарт NFC регулює радіотехнологію, яка дозволяє двом пристроям взаємодіяти, коли вони знаходяться в безпосередній близькості, як правило, не більше декількох сантиметрів, що забезпечує безпечний обмін інформацією. Хоча діапазон зв'язку NFC обмежений кількома сантиметрами, стандарт не гарантує безпечний зв'язок і деякі типи атак все ж зстосовуються до даної технології. Чинний стандарт ISO фактично не описує контрзаходів проти методів атак NFC; наприклад, технологію можна атакувати за допомогою однієї з класичних наступальних схем – атаки людина посередині. Окрім цього, не пропонується захист від прослуховування, що робить обмін даними вразливим до модифікації даних. [49]

Тому жоден з каналів не є достатньо безпечним для здійснення багатофакторної автентифікації самостійно і повинен використовуватися разом з іншими для захисту доступу до системи. Це зменшує ризик зламу облікового запису, оскільки зловмисникам потрібно скомпрометувати всі залучені канали.

2.2 Питання безпеки токена на мобільному пристрої

Як було зазначено раніше, програмні токени можуть керуватися мобільними пристроями. Через їх широке поширення та можливості, виникає проблема зберігання і передачі токенів доступу до веб-ресурсів централізовано з одного пристрою.

У випадку з мобільним телефоном токен на основі програмного забезпечення часто є просто мобільним додатком. Це означає, що при отриманні доступу до ресурсу з мобільного телефону всі фактори, незалежно від каналу передачі, утримуються одним пристроєм. У випадку, якщо смартфон був скомпрометований чи викрадений, ефективність МСА знижується до нуля (рис.2.2).

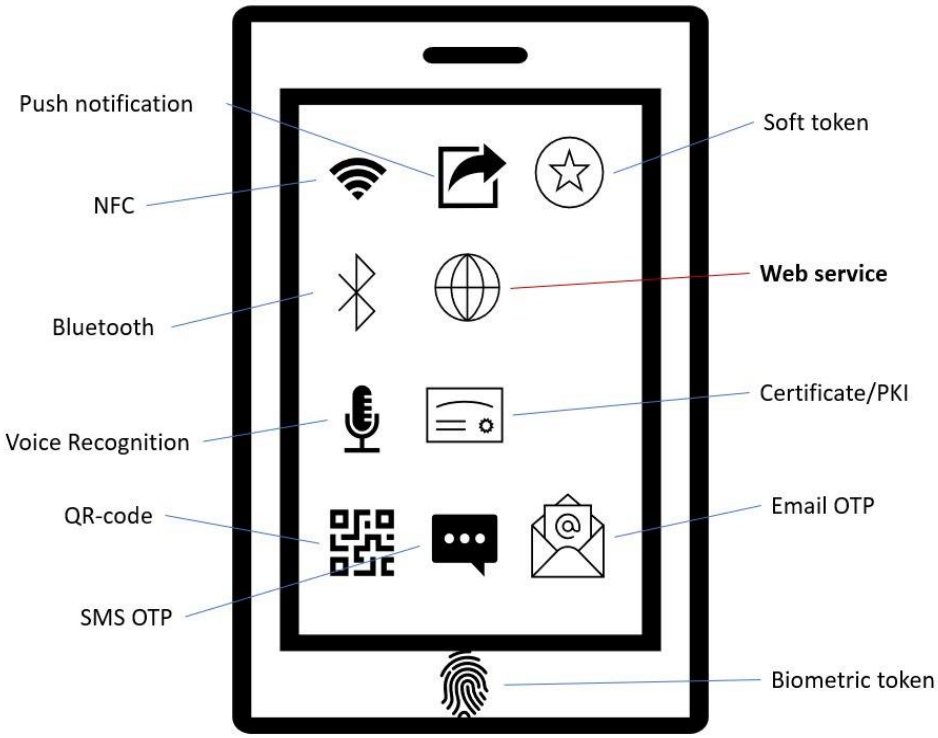

Рисунок 2.2. Токени концентрованії в одному пристрої

Якщо мобільний пристрій було вкрадено, він все ще може бути захищений паролем або PIN-кодом. Підходом до захисту пристроїв, що зберігають дані, необхідні для автентифікації, також може бути використання окремих паролей для мобільних додатків, що зберігають токени. Однак це просто гальмує хакерів, але не зупиняє їх. Більше того, відповідно статистики [47], 51% людей використовують

однакові паролі як для робочих, так і для особистих акаунтів. З іншого боку, мобільні пристрої можуть бути заражені шкідливим програмним забезпеченням, і це відкриває поверхню для атак, навіть якщо реалізована багатофакторна автентифікація.

2.3 Автентифікація на декількох пристроях

Коли для автентифікації заявника використовується багато пристроїв, це приклад автентифікації на декількох пристроях. Наявність декількох пристроїв означає, що ми можемо посилити механізм контролю доступу, використовуючи кілька каналів, за допомогою яких можна перевірити особистість. Деякі широко розповсюджені програми, наприклад, розроблені організацією Google можуть точно ідентифікувати пристрій, який користувач використовує для доступу до послуг. Існує декілька способів ідентифікації пристрою: ідентифікатор пристрою, міжнародна ідентифікація мобільного обладнання (IDEI), обліковий запис користувача тощо. Як результат, ми можемо підвищити надійність, якщо MFA вимагає надання користувачем другого фактора через пристрій, окремий від того, щоб був використаний для автентифікації першого фактора.

Наприклад, коли користувач автентифікується на веб-ресурсі з персонального комп'ютера, можна використовувати апаратний або програмний токен. Відповідно авторизація з мобільного телефону повинна використовувати програмне забезпечення або будь-який доступний токен, що зберігається на іншому пристрої. Отже, зловмисник повинен одночасно зламати кожен задіяний пристрій, щоб скомпрометувати систему мультифакторної автентифікації.

Окремим аспектом є використання мультифакторної автентифікації за котнекстом. Зважаючи на це, сервіс буде довіряти інформаційній системі, або ір-адресі з якої відбулася успішна автентифікація і не буде запитувати другий фактор при повторному використанні цієї системи чи адреси. З одного боку, цей підхід полегшує процес автентифікації для користувача, однак, вразливість того, що

зловмисник скористається цією особливістю процесу автентифікації також несе загозу несанкціонованого доступу.

2.4 Переваги використання схеми мультиканальної та автентифікації на декількох пристроях

Переваги впровадження архітектури MCA та MDA відповідно до ймовірності компрометації можуть мати великий вплив [48]. Якщо ми використовуємо один пристрій - d і багато каналів - c з ймовірністю компрометації p_d, p_c. Імовірність компрометації схеми автентифікації буде дорівнювати (2.1)

$$P = p_d + p_{c1}p_{c2} \dots p_{cn} \tag{2.1}$$

де n - кількість використовуваних каналів. Така сама ситуація трапляється, якщо ми використовуємо багато пристроїв з одним каналом.

Таким чином, у випадку присутності MCA та MDA в одній схемі, ймовірність зменшується, оскільки зловмиснику тепер необхідно скомпрометувати або всі канали, або всі пристрої, або деякі канали, і деякі пристрої. Наприклад, схема з 2 пристроями та 2 каналами матиме ймовірність компрометації, рівну (2.2)

$$P = p_{d1}p_{d2} + p_{c1}p_{c2} + p_{d1}p_{c2} + p_{c1}p_{d2} \tag{2.2}$$

Отже, важливо враховувати канал передачі, а також розглядати залучення різних пристроїв для посилення надійності другого фактора та схеми автентифікації в цілому. Завдяки багатофакторній автентифікації з використанням декількох пристроїв та різних каналів передачі фактора автентифікації, користувачі можуть посилити захист пріоритетних облікових записів, навіть якщо була порушена конфіденційність інформаційної системи. Однак, повинен бути впроваджений точний спосіб ідентифікації пристрою, що не буде впливати на анонімність, конфіденційність та безпеку користувача і послуги.

Висновки за розділом 2

Отже, було розглянуто безпеку факторів автентифікації, та визначено, як різні канали передачі та пристрої, що зберігають токени користувачів можуть підвищити безпеку системи мультифакторної автентифікації у разі несанкціонованого доступу до інформаційних систем, що місять токен, або компрометації незахищених каналів передачі.

Таким чином, в другому розділі було визначено умови генерації факторів автентифікації та проведено дослідження ймовірності компрометації системи мультифакторної автентифікації.

РОЗДІЛ 3
ПОБУДОВА НАДІЙНОЇ АВТЕНТИФІКАЦІЇ

3.1 Загальний опис архітектури

Для практичної побудови надійної, двох факторної автентифікації користувачів, що мають привілейований доступ на системах було обрано наступні інструменти:

- Microsoft Active Directory (MS AD) – служба каталогів корпорації Microsoft. За замовчуванням облікові дані Windows перевіряються в базі даних Менеджера облікових записів безпеки (SAM) на локальному комп'ютері або в Active Directory на приєднаному до домену комп'ютері за допомогою служби Winlogon [49]. Облікові дані збираються шляхом введення їх користувачем в інтерфейсі для входу, або програмно через програмний прикладний інтерфейс (API) для подання до цілі автентифікації. Так як розгляд доменних сервісів не є метою роботи, для простоти функції контролера домена будемо розглядати в якості провайдера облікових записів.

- Wallix Bastion (Bastion) [50] – проксі-рішення з управління привілейованим доступом. Управління привілейованим доступом відноситься до систем, які надійно керують обліковими записами користувачів, які мають підвищені дозволи на критичних корпоративних ресурсах. Це можуть бути адміністратори, пристрої, програми та інші типи користувачів. Привілейовані облікові записи користувачів є цілями кіберзлочинців, тому використання двох факторної автентифікації є надзвичайно важливим завданням. Таким чином, Wallix Bastion обраний в якості системи, на якій відбувається авторизація.

- LinOTP [51] - це гнучка платформа з відкритим кодом, корпоративного рівня для надійної автентифікації. Особливістю вибору саме цієї платформи є функціональність та можливість використання в локальній інфраструктурі, без необхідності синхронізації користувачів з хмарними сховищами. LinOTP виступає в

якості провайдера другого фактора, саме цей застосунок перевіряє правильність наданого другого фактора відповідно до облікових даних користувача.

- FreeRADIUS [52] - RADIUS сервер з відкритим вихідним кодом. Це альтернатива іншим комерційним RADIUS серверам, оскільки він модульний і функціональний. В ході реалізації архітекттури надійної автентифікації обраний RADIUS сервер виступає в ролі коннектора між LinOTP та Wallix Bastion.
- Google Authenticator [53] – мобільний додаток для двухетапної автнетифікації за допомогою заснованого на часі одноразового пароля (TOTP) та HMAC алгоритму одноразово пароля (HOTP), розроблений корпорацією Google.

Отже, схема рішення матиме наступний вигляд (рис. 3.1)

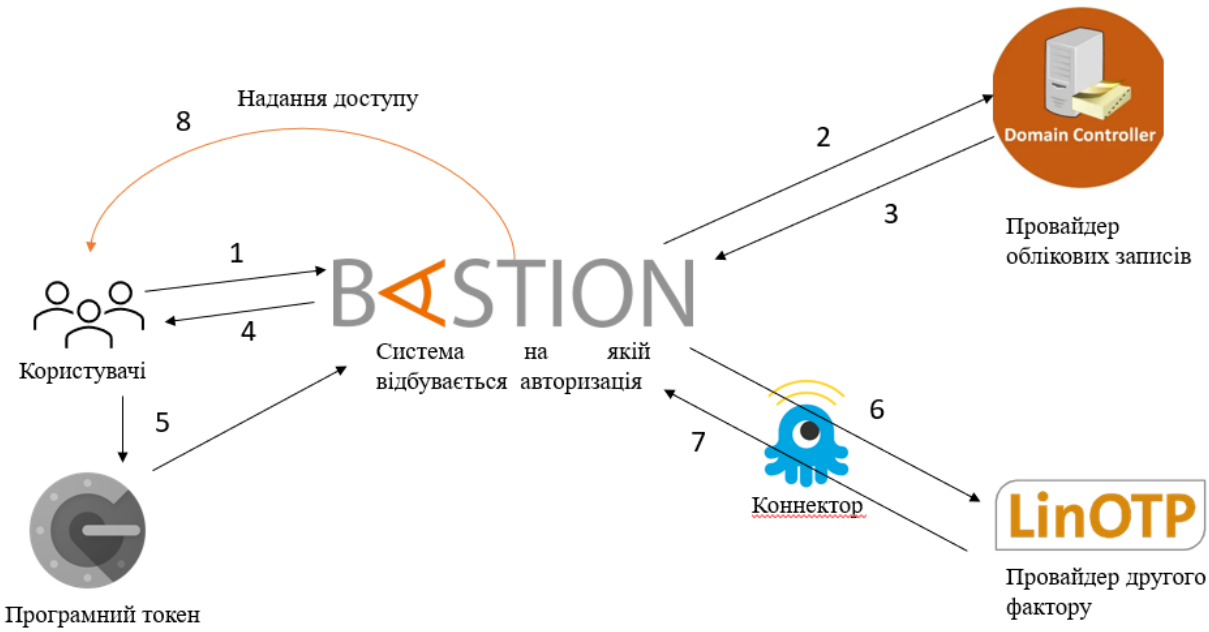

Рисунок 3.1 Архітектура надійної автентифікації

Архітектура рішення відповідно до мережевих протоколів, що використовуються (рис. 3.2).

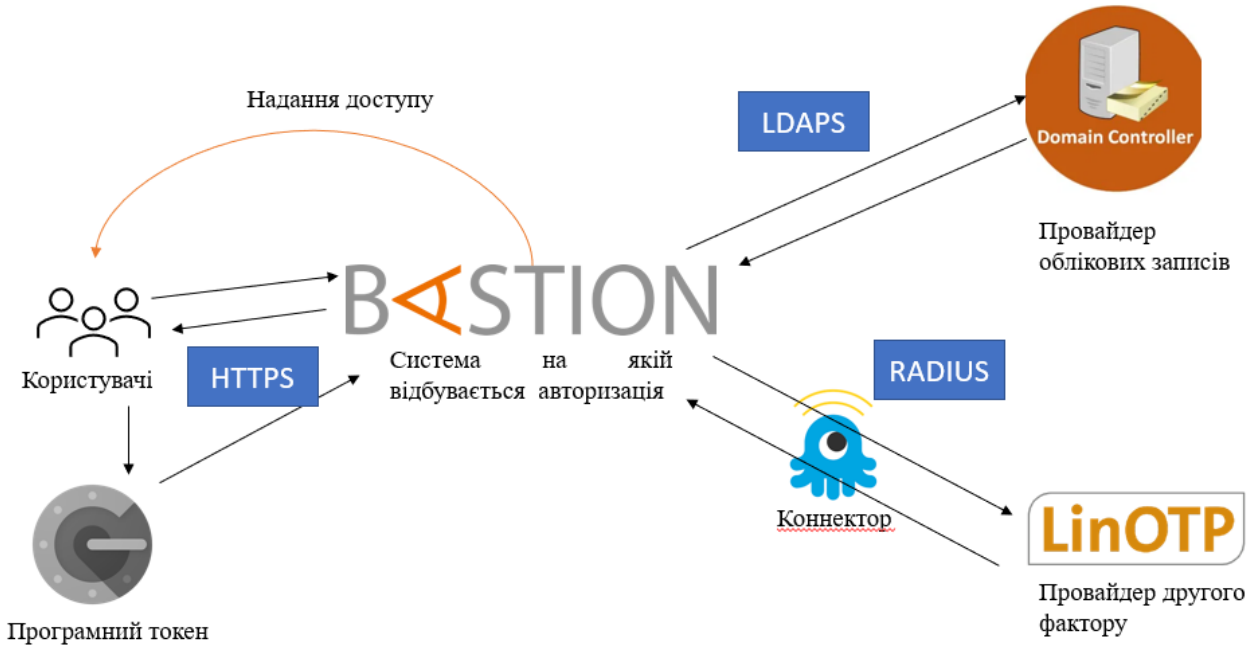

Рисунок 3.2 Архітектура - мережеві протоколи

Відповідно до рисунку 3.1, процес двох факторної автентифікації відбувається покроково наступним чином:

1. Користувач заявляє про свій намір авторизуватися на системі надаючи ідентифікатор та перший фактор у вигляді статичного пароля.

2. Система Wallix Bastion робить запит до провайдера облікових даних: чи існує користувач з таким ідентифікатором та чи правильний пароль було вказано.

3. Microsoft Active Diretory – надсилає відповідь про успішну чи неуспішну автентифікацію заявника.

4. Wallix Bastion вимагає в користувача надати другий фактор.

5. Користувач за допомогою Google Authenticator надає одноразовий пароль на основі часу.

6. Wallix Bastion робить запит до RADIUS сервера з даними автентифікатора, які передаються до провайдера LinOTP.

7. LinOTP вираховує поточний токен користувача, звіряє з наданим і надсилає відповідь про успішну чи неуспішну автентифікацію.

8. Якщо первинна та вторинна автентифікації пройшли успішно Wallix Bastion авторизує користувача та надає йому доступ до системи.

Інші процеси, що виконуються на етапі підготовки рішення включають: створення користувачів в MS AD; інтеграцію MS AD та Bastion; імпорт та синхронізацію користувачів MS AD до LinOTP; інтеграцію LinOTP та Bastion за допомогою FreeRADIUS; Реєстрацію користувачами токенів в Google Authenticator на порталі самообслуговування LinOTP.

Мережеві протоколи, що використовуються, відповідно до рис. 3.2 наступні:

- Зв'язок між користувачами та системою Wallix Bastion – HTTPS;
- Wallix Bastion та Microsoft Active Directory – LDAPS;
- Wallix Bastion LinOTP – RADIUS (з використанням FreeRADIUS у вигляді коннектора);

Таким чином, було розроблено архітектуру рішення відповідно до інфраструктури та мережевого з'єднання, наступним кроком є його практична реалізація.

3.2 Реалізація розробленої архітектури.

Для того, аби реалізувати розроблену архітектуру треба встановити рішення та провести їх інтеграцію, а також дозволити користувачам реєструвати токени. Так як, метою реалізації є саме налаштування другого етапу автентифікації (другого фактору), початковою точкою будемо вважати [54]:

- Microsoft Windows Server 2019 зі встановленим сервісом Active Directory Domain Services, та попередньо створеними користувачами.
- Wallix Bastion, інтегрований з MS AD.
- Debian 10 – Операційна система на якій буде встановлений LinOTP.
- Користувачі з програмним застосунком Google Authenticator.

1. Інсталляція LinOTP:

На операційній системі Debian 10 необхідно додати репозиторій LinOTP, ця дія виконується записом в файл:

```
echo ‘deb http://www.linotp.org/apt/debian buster linotp linotp-deps’ >
/etc/apt/sources.list.d/linotp.list
```

Для перевірки справжності пакету можна імпортувати ключ:

```
apt-get install dirmngr
apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 913DFF12F86258E5
```

Дані користувачів повинні зберігатися в базі даних, для цього потрібно встановити базу даних – Mariadb [55]:

```
apt-get update
apt-get install mariadb-server
mysql_secure_installation
```

Встановлюємо LinOTP:

```
apt-get install linotp
```

2. Інтеграція з Active Directory.

Після коректного встановлення служби, веб-панель управління застосунку LinOTP повинна бути доступною за посиланням - https://<ip-адреса-linotp>/manage

Інтеграція з каталогами Active Directory необхідна для можливості LinOTP використовувати облікові дані користувачів аби вони могли реєструвати токени, які «прив’язуються» до їхнього облікового запису. З головного меню необхідно перейти до LinotpConfig, а саме UserIdResolvers та створити новий UserIdResolver, обравши тип LDAP (рис.3.3). Таким чином, буде створена сутність, що може читати ідентифікатори користувачів та присвоювати токени окремим акаунтам.

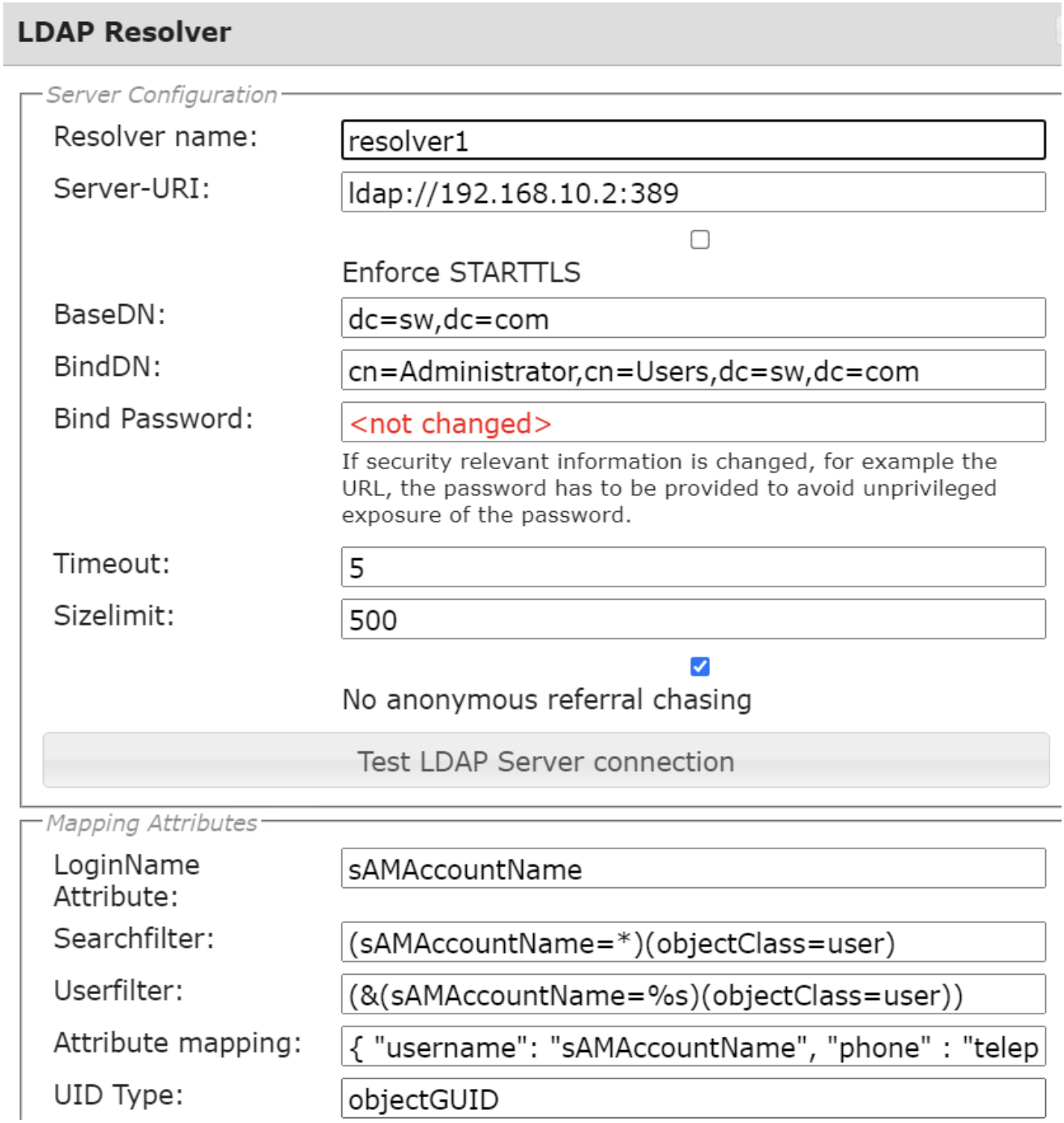

Рисунок 3.3 UserIdResolver

На рисунку 3.3 зображені поля мають наступні значення:

- Server-URI - адреси серверів LDAP або LDAPS.

- Base DN - дерево, яке містить необхідні дані.

- Bind DN – характерне ім'я, що використовується для підключення до сервера LDAP. Це має бути додатковий обліковий запис, створений для LinOTP, без інших дозволів.

- Bind password - пароль облікового запису запиту.

- Timeout - містить одне або два значення у секундах формату 5.0; 7.5 Перше значення - це час очікування для підключення та входу на сервер LDAP. Друге значення - це час очікування відповіді від сервера LDAP після встановлення сеансу перед розірванням з'єднання.

- Sizelimit - скільки користувачів запитується з LDAP.

- No anonymous referral cache - під час налаштування LDAPIdResolver для підключення до Active Directory і для BaseDN встановлені компоненти домену верхнього рівня.

- Атрибут loginname- це атрибут, що містить ім’я, яке користувачі використовуватимуть для входу.

- Фільтр пошуку - використовується для фільтрації об'єктів користувача, знайдених у налаштованій базовій DN

- Фільтр користувачів - у свою чергу буде використаний для вирішення проблем користувачів. Наприклад, якщо користувач увійде в систему, він надасть ім'я для входу. Ім'я для входу буде замінено на "% s" у фільтрі користувача, щоб перетворити ім'я входу на відмінне ім'я об'єкта користувача в LDAP.

- Зіставлення атрибутів – фунція отримає більше інформації про користувача з сервера LDAP і зіставить їх для виправлення внутрішніх ідентифікаторів LinOTP. Так, наприклад, телефон атрибута LinOTP можна використовувати для токена SMS, і тут можна вказати, який атрибут LDAP слід зіставити для цієї мети.

- Тип UID - це ідентифікатор, що однозначно визначає запис користувача. Так, наприклад, якщо ім’я користувача змінено, користувач може ввійти з новим іменем та раніше призначеними ідентифікаторами. Цей атрибут повинен бути objectGUID для активних каталогів та entryUUID для OpenLDAP.

Тепер необхідно додати Realm: для цього треба задати йому ім'я, і вказати LDAP Resolver, який був тільки створений. UserIdResolver-и організовані в «області», щоб LinOTP міг їх використовувати. До одного Realm-у може бути віднесено багато UserIdResolver-ів, така побудова реалізує концепцію multitenancy – управління декількома організаціями, без необхідності встановлення рішення на

інфаструктурі кожної з них. На цьому етапі можемо перевірити синхронізованих користувачів у вікні User View (рис. 3.4).

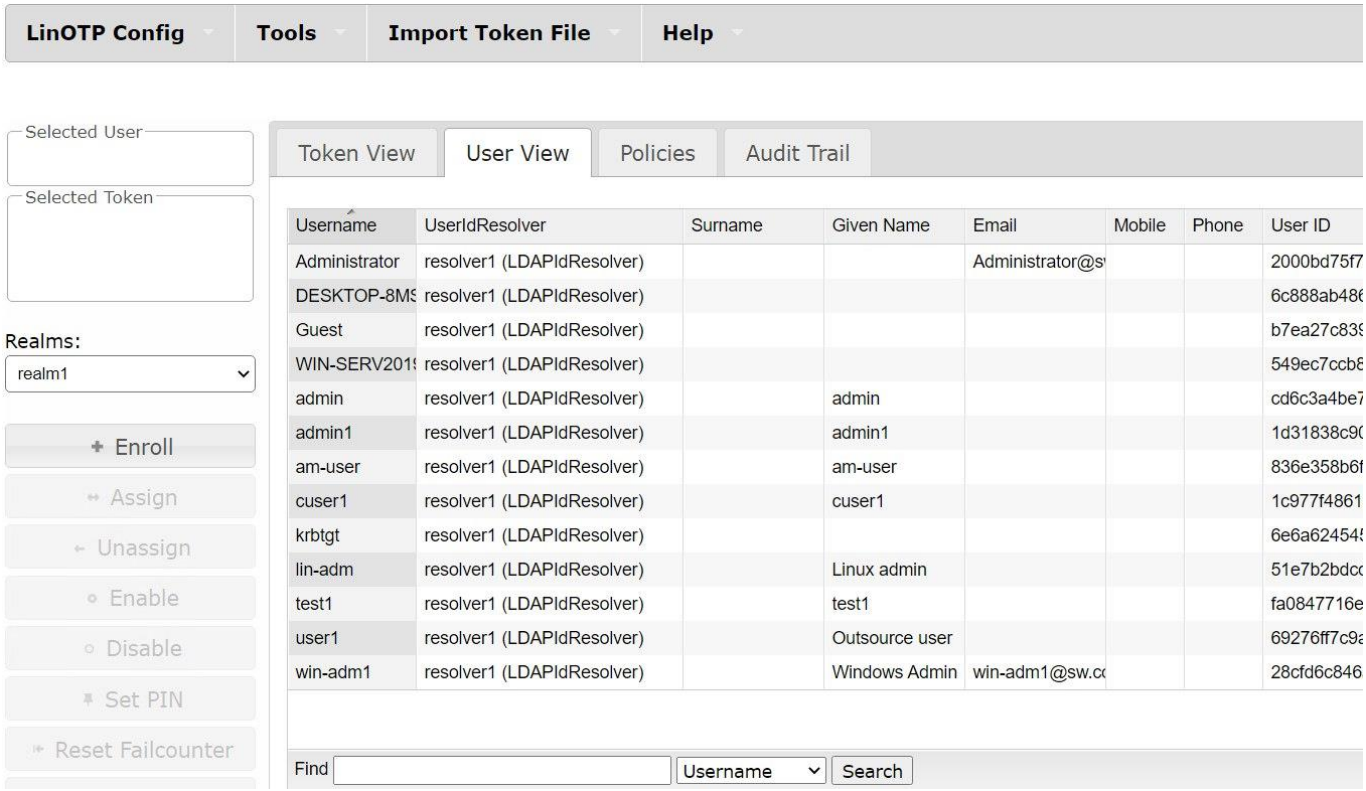

Рисунок 3.4 Вікно User View - коректна інтеграція з Active Directory

3. Інтеграція LinOTP та FreeRADIUS.

Клієнт RADIUS - це пристрій, який підключається до сервера LinOTP з метою аутентифікації за допомогою протоколу RADIUS. Цей протокол реалізується (або може бути реалізований) у широкому діапазоні процедур входу, наприклад SSL VPN, брандмауера або постачальника облікових даних LinOTP.

Пристрій включає попередньо налаштований сервер freeRADIUS. З міркувань безпеки клієнт RADIUS за замовчуванням не приймається. Отже, потрібно попередньо налаштувати, кому дозволено зв’язуватися з сервером LinOTP через RADIUS і який секрет слід використовувати для захисту з’єднання клієнт-сервер RADIUS.

Для початку, треба встановити програмний застосунок FreeRADIUS і додаткові пакети для коректної інтеграції рішень:

apt-get install linotp-adminclient-cli python-ldap freeradius python-passlib python-bcrypt git libio-all-lwp-perl libconfig-file-perl libtry-tiny-perl

Створюємо символічні посилання для FreeRADIUS:

```
ln -s /etc/freeradius/3.0/sites-available /etc/freeradius/sites-available
ln -s /etc/freeradius/3.0/sites-enabled /etc/freeradius/sites-enabled
ln -s /etc/freeradius/3.0/clients.conf /etc/freeradius/clients.conf
ln -s /etc/freeradius/3.0/users /etc/freeradius/users
```

Встановлюємо модуль автентифікації linotp-auth-freeradius-perl:

```
git clone https://github.com/LinOTP/linotp-auth-freeradius-perl.git
cd linotp-auth-freeradius-perl/
cp radius_linotp.pm /usr/share/linotp/radius_linotp.pm
```

Далі необхідно відредагувати конфігураційний файл наступним чином (рис.3.5):

```
nano /etc/freeradius/sites-enabled/linotp
```

```
server linotp {
listen {
  ipaddr = *
  port = 1812
  type = auth
}
listen {
  ipaddr = *
  port = 1813
  type = acct
}
authorize {
  preprocess
  update {
    &control:Auth-Type := Perl
  }
}
authenticate {
  Auth-Type Perl {
    perl
  }
}
accounting {
  unix
  }
}
```

Рисунок 3.5 Конфігурація RASIUS-сервера

В директорії sites-enabled необхідно залишити лише файл з назвою linotp.

Наступним кроком треба додати RADIUS-клієнтів, згідно архітектури - клієнтом є система Wallix Bastion (рис. 3.6):

nano /etc/freeradius/clients.conf

```
client Wallix {
  ipaddr = 192.168.10.10
  secret = your_secret
}
```

Рисунок 3.6 Конфігурація RADIUS-клієнта

Використовуємо конектор perl в якості бази користувачів (рис. 3.7):

nano /etc/freeradius/users

```
DEFAULT Auth-type := perl
```

nano /etc/freeradius/3.0/mods-available/perl

```
perl {
  filename = /usr/share/linotp/radius_linotp.pm
  func_authenticate = authenticate
  func_authorize = authorize
 }
```

Рисунок 3.7 Конфігурація perl-конектора

Наступним кроком є створення конекторів для перевірки облікових даних, що надійшли до RADIUS-сервера (рис.3.8):

nano /etc/linotp2/rlm_perl.ini

```
URL=https://< ip-адреса-linotp >/validate/simplecheck
REALM=realm1
RESCONF=resolver1
Debug=True
SSL_CHECK=False
```

Рисунок 3.8 Конфігурація конектора перевірки облікових даних

4. Реєстрація токенів користувачів.

На цьому етапі необхідно організувати сервіс, де користувачі зможуть самостійно реєструвати токени. Цей функціонал реалізований в LinOTP як selfservice портал [56].

Переходимо до LinOTPConfig > policies і створюємо нову політику. Згідно архітектури рішення користувачі будуть використовувати TOTP, що реєструється в Google Authenticator. Таким чином, параметри політики матимуть вигляд (рис. 3.9).

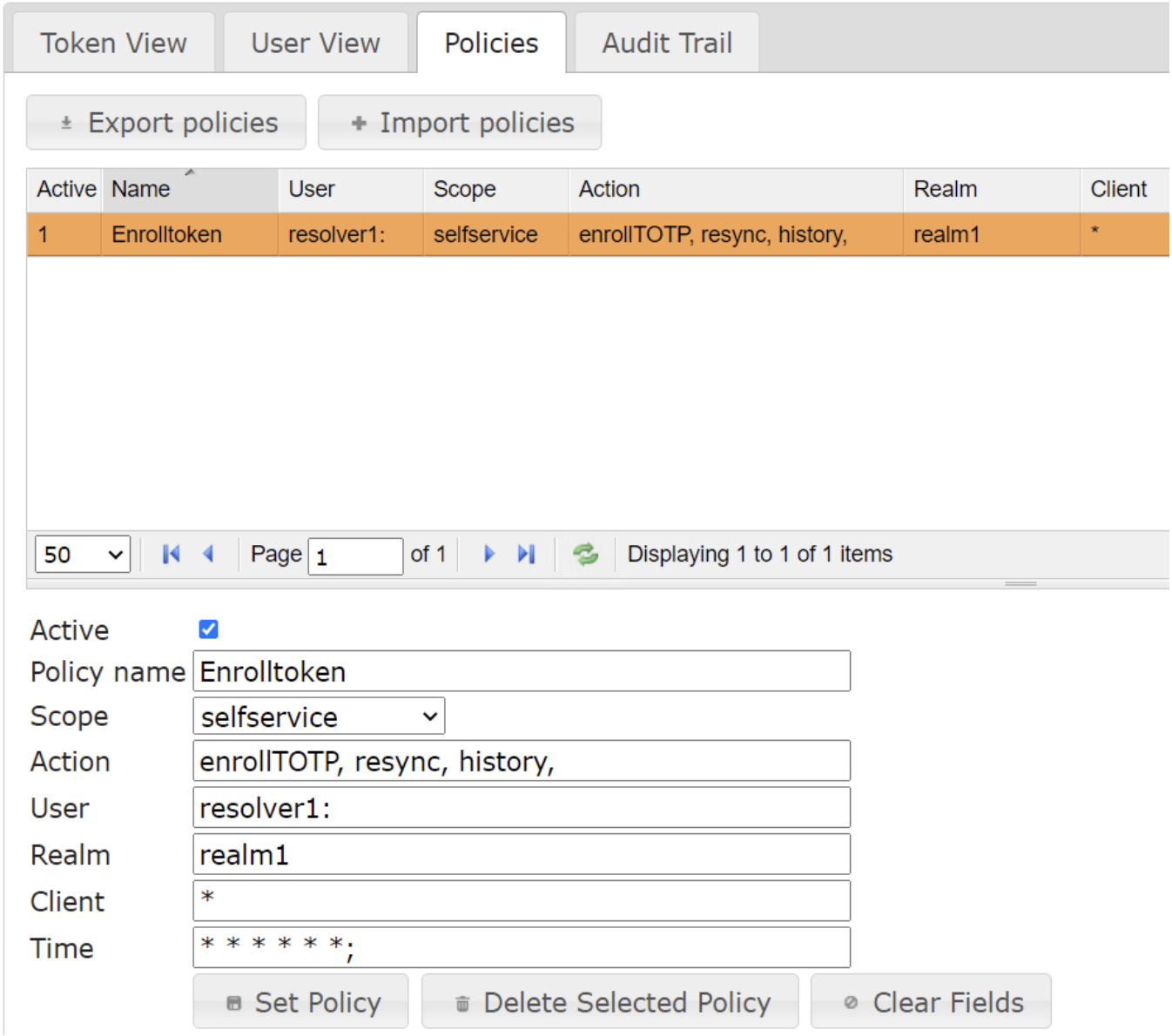

Рисунок 3.9 Політика сервісу реєстрації токенів

Якщо необхідно визначити політики для користувачів, які отримують доступ до порталу самообслуговування, згідно рисунку 3.9 потрібно ввестиопції конфігурації:

- Назва політики: “назва політики”.

- Сфера застосування: “самообслуговування”.

- Користувач: “*, ім’я користувача, регулярний вираз”, щоб детально показати користувачів до яких застосовується політика.

- Сфера (Realm): "*, ім'я realm-у" -потрібно вставити ім’я області в поле realm-у. Тоді ця політика працюватиме для всіх користувачів у цій області, які входять на портал самообслуговування. Також можна поставити * у поле сфери, таким чином, політика буде чинною для всіх сфер.

- Клієнт: “FQDN, IP-Addr, мережа” – є можливість додати клієнтів у цій політиці, щоб визначити іншу поведінку на Порталі самообслуговування залежно від того, звідки користувач буде входити на Портал самообслуговування. Дана концепція є прикладом контексної автентифікації.

- Час: якщо це поле порожнє означає будь-який час.

Таким чином, посилання для користувачів має вигляд:

https://< ip-адреса-linotp > або https://< ip-адреса-linotp > /selfservice/login (рис.3.10). Це означає, що користувач може отримати доступ до порталу самообслеговування використовуючи свої доменні облікові дані.

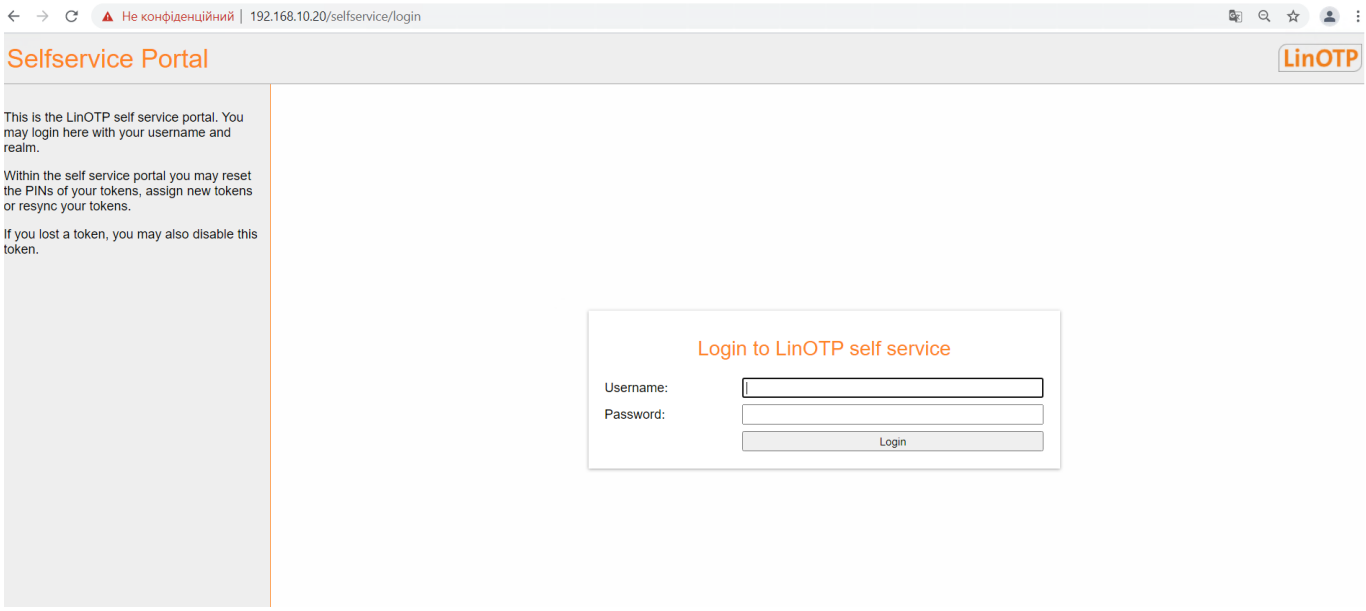

Рисунок 3.10 Selfservice портал LinOTP

5. Wallix Bastion Інтеграція

Останнім етапом налаштування надійної автентифікації буде інтеграція рішень системою Wallix. Щоб Wallix Bastion запитував у користувачів другий

фактор, потрібно додати зовнішню автентифікацію. У вікні Configuration -> External authentication додаємо нову аутентифікацію типу RADIUS. Secret вказуємо той, що був використаний раніше на етапі конфігурації freeRADIUS. (рис. 3.11)

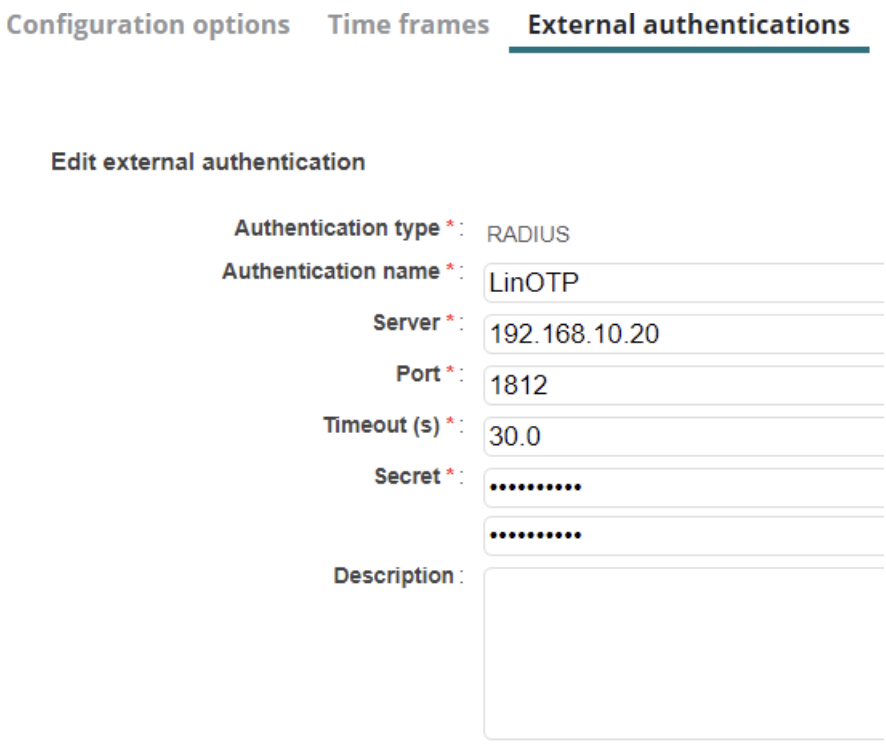

Рисунок 3.11 Зовнішня автентифікація на Wallix Bastion

На сторінці LDAP / AD domains потрібно обрати домен, який раніше був синхронізований з LinOTP. У полі Secondary authentication необхідно додати linotp – зовнішню RADIUS-автентифікацію, яку було створено на попередньому кроці і застосувати конфігурацію. Після цього домен буде виглядати так: (рис.3.12 – 3.13)

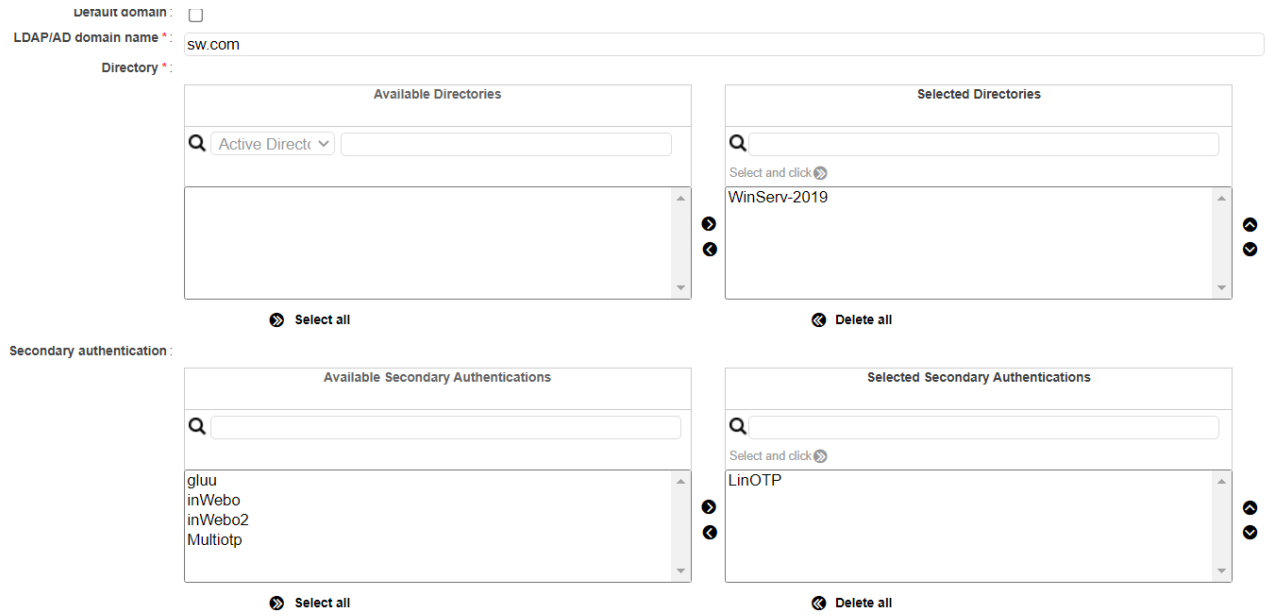

Рисунок 3.12 Конфігурація первинної та вторинної автентифікації

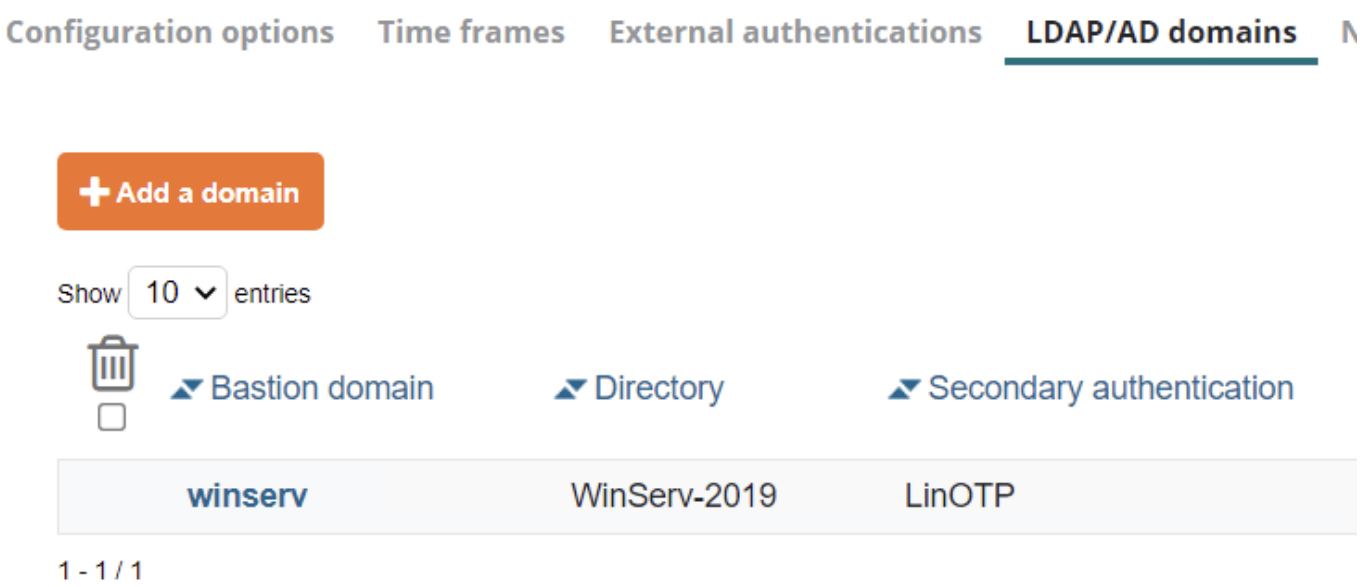

Рисунок 3.13 Конфігурація LDAP/AD domains

З цього моменту можливо автентифікуватися і пройти авторизацію у Wallix Bastion за допомогою двохфакторної автентифікації, де другий фактор перевіряє LinOTP (рис. 3.14 – 3.18).

Рисунок 3.14 Запит первинної автентифікації

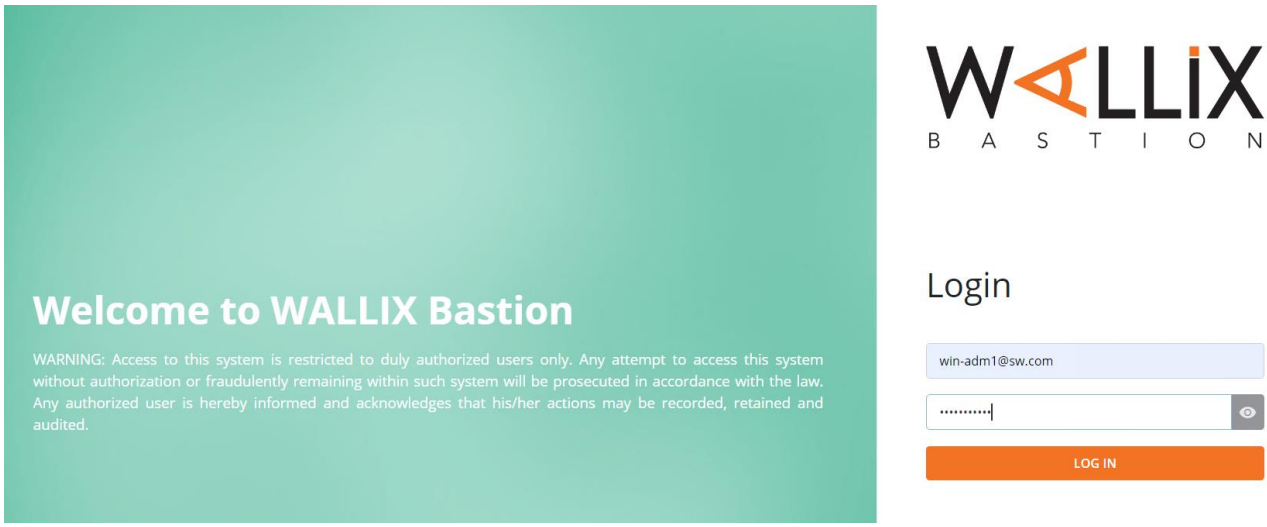

Рисунок 3.15 Ідентифікатор та перший фактор користувача

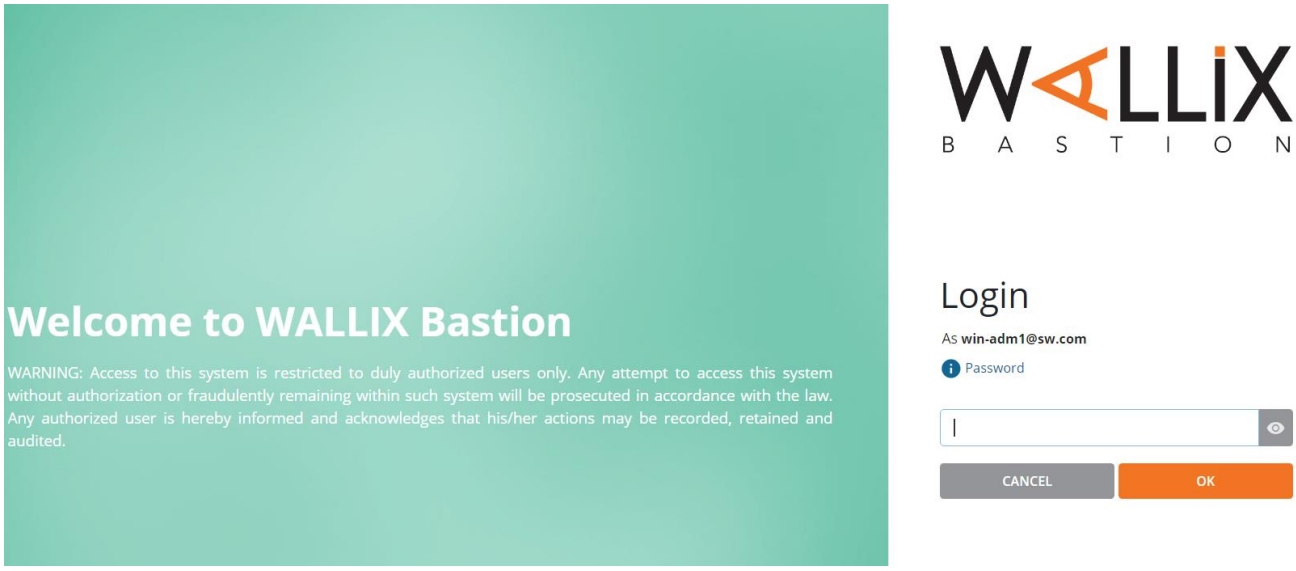

Рисунок 3.16 Запит другого фактору користувача

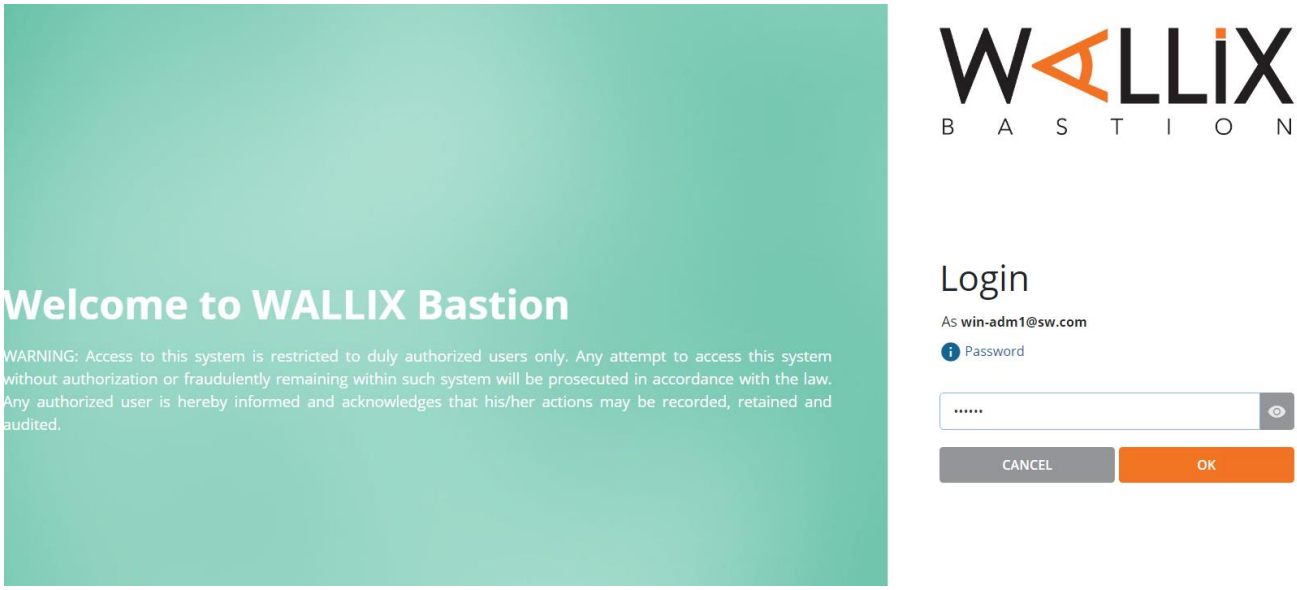

Рисунок 3.17 Вторинна автентифікація

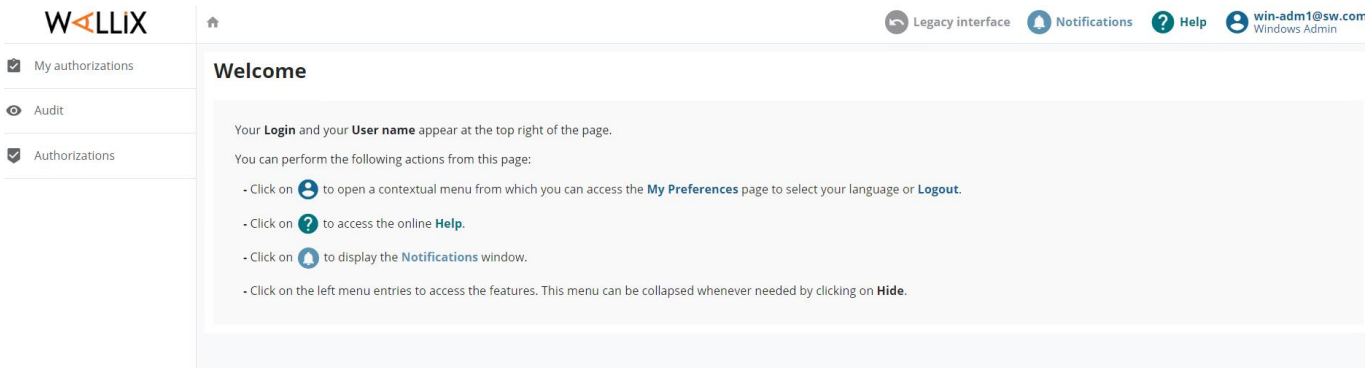

Рисунок 3.18 Успішна автентифікація та авторизація користувача на Wallix Bastion

Для того, щоб на програмному рівні переконатися у коректності роботи сервісу, можна перевірити витяг з журналу подій FreeRADIUS (рис. 3.19)

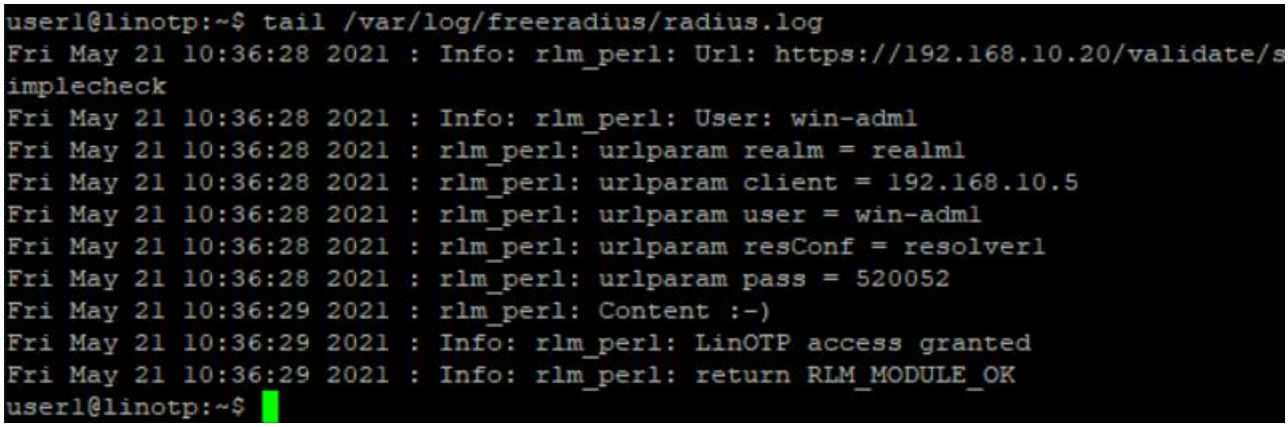

Рисунок 3.19 Запис логу про успішну автентифікацію

Таким чином, взаємодія систем побудована коректно, і виконує завдання, які були визначені в попередніх розділах.

Висновки за розділом 3

Таким чином, в третьому розділі було розроблено архітектуру надійної автентифікації, яка дозволяє користувачам, зокрема адміністраторам, централізовано, використовуючи свої доменні облікові дані та одноразовий пароль, отримувати доступ до різного характеру інформаційних систем (включаючи сервери, персональні комп'ютери, мережеве обладнання, веб-додатки, спеціалізовані програми та ін.).

Крім того, використання обраних систем, особливо провайдера другого фактора – LinOTP робить дану технологію придатною до використання в державних установах та банкових системах, де обробляється конфіденційна інформація.

В ході третього розділу було виконано наступні завдання дипломної роботи:

- Розроблено архітектуру мультифакторної автентифікації;
- Інтегровано доменні сервіси та елементи системи захисту інформації;

- Інтегровано токени автентифікації з елементами системи захисту інформації.

ВИСНОВКИ

У ході дипломної роботи було досліджено – процес захисту облікових даних від компрометації і визначено основні загрози компрометації облікових даних. Для зменшення ризиків була запропонована архітектура надійної автентифікації, що використовує два фактори для перевірки ідентичності користувача. Пропонується використання проксі-рішення, яке нівелює проблему встановлення та оновлення додаткового програмного забезпечення на інформаційних системах до яких відбувається доступ. Також запропоноване рішення містить провайдера другого фактора, що не потребує доступу до глобальної мережі і може використовуватися локально.

В першому розділі дипломної роботи було проаналізовано літературу, визначено основні загрози компрометації та підходи до захисту облікових даних. Проведено дослідження побудови системи мультифакторної автентифікації у корпоративному середовищі.

В другому розділі було виконано завдання генерації факторів автентифікації та проведено дослідження ймовірності компрометації системи мультифакторної автентифікації.

В третьому розділі було розроблено архітектуру мультифакторної автентифікації, інтегровано доменні сервіси та елементи системи захисту інформації, інтегровано токени автентифікації з елементами системи захисту інформації.

Таким чином, були виконані всі завдання дипломної роботи і в результаті розроблено архітектуру, що складається з елементів системи захисту інформації для захисту від загроз компрометації облікових даних в корпоративних мережах.

Отже, мета роботи, а саме розробка елементів системи захисту інформації від загроз компрометації облікових даних була досягнена.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Закон України “Про захист інформації в інформаційно-телекомунікаційних системах” [Електронний ресурс] //Верховна Рада України - 2020. - Режим доступу до ресурсу: https://zakon.rada.gov.ua/laws/show/80/94-вр#Text

2. Top 10 Web Application Security Risks [Електронний ресурс] //OWASP – 2021. – Режим доступу до ресурсу: https://owasp.org/www-project-top-ten/

3. Microsoft Windows 7 [Електронний ресурс] //Microsoft – 2021. – Режим доступу до ресурсу: http://www2.westsussex.gov.uk/LearningandDevelopment/IT%20Learning%20Guides/Microsoft%20Windows%207/14%20User%20accounts.pdf

4. A Guide to User Accounts [Електронний ресурс] //Digital Skills Academy - 2018. - Режим доступу до ресурсу: http://cre8te.co.uk/wp-content/uploads/2018/07/User-Accounts-FINAL.pdf

5. Evaluation Assurance Level [Електронний ресурс] //Wikipedia – 2021. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Evaluation_Assurance_Level

6. Digital Identity Guidelines [Електронний ресурс] //NIST – 2017. – Режим доступу до ресурсу: https://pages.nist.gov/800-63-3/sp800-63-3.html

7. N. E. Hastings and D. F. Dodson, "Quantifying assurance of knowledge based authentication," in ECIW 2004: The 3rd European Conference on Information Warfare and Security, 2004

8. K. Renaud, "Quantifying the quality of web authentication mechanisms: a usability perspective," Journal of Web Engineering, vol. 3, pp. 95-123, 2004.

9. Using Geographical Location as an Authentication Factor to Enhance mCommerce Applications on Smartphones // Torben Kuseler & Ihsan Alshahib Lami - 2012. - Режим доступу до ресурсу: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.736.6702&rep=rep1&type=pdf

10. NISTIR 7983, Report: Authentication Diary Study, surveys user behaviors for coping with the friction and burden imposed by managing their portfolios user IDs and

password [Електронний ресурс] //NIST - 2014.- Режим доступу до ресурсу: https://csrc.nist.gov/publications/detail/nistir/7983/final.

11. Vulnerabilities in password-based login [Електронний ресурс] //Port Swigger - 2021. - Режим доступу до ресурсу: https://portswigger.net/web-security/authentication/password-based

12. Forensic Analysis of Tor Browser: A Case Study for Privacy and Anonymity on the Web [Електронний ресурс] //Abid Khan Jadoon, Waseem Iqbal, Muhammad Faisal Amjad, Hammad Afzal,Yawar Abbas Bangash - 2019.- Режим доступу до ресурсу: https://www.researchgate.net/publication/332004753_Forensic_Analysis_of_Tor_Browser_A_Case_Study_for_Privacy_and_Anonymity_on_the_Web

13. Social Engineering Attacks to Watch Out For [Електронний ресурс] //DAVID BISSON - 2019.- Режим доступу до ресурсу: https://www.tripwire.com/state-of-security/security-awareness/5-social-engineering-attacks-to-watch-out-for/

14. Baiting [Електронний ресурс] //keepnet labs – 2020. - Режим доступу до ресурсу: https://www.keepnetlabs.com/baiting/

15. Масштабна фішингова атака на державні установи України [Електронний ресурс] //CERT-UA - 2021. - Режим доступу до ресурсу: https://cert.gov.ua/article/10011

16. Understanding and dealing with phishing during the COVID-19 pandemic [Електронний ресурс] //ENISA - 2020. - Режим доступу до ресурсу: https://www.enisa.europa.eu/news/enisa-news/understanding-and-dealing-with-phishing-during-the-covid-19-pandemic

17. New Exploit Hacks LinkedIn 2-Factor Authentication [Електронний ресурс] //Kevin Mitnick - 2018. - Режим доступу до ресурсу: https://www.youtube.com/watch?v=xaOX8DS-Cto

18. DIFFERENT TYPES OF KEYLOGGERS. Mitigation and risk relevancy in modern society [Електронний ресурс] //Toni Blåfield – 2020. – Режим доступу до ресурсу: https://trepo.tuni.fi/bitstream/handle/10024/122479/BlåfieldToni.pdf?sequence=2

19. Sbai H., Goldsmith M., Meftali S., Happa J., A Survey of Keylogger and Screen- logger Attacks in the Banking Sector and Countermeasures to Them, Castiglione A., Pop F., Ficco M., Palmieri F., Cyberspace Safety and Security, CSS, 2018, p. 1 (cited 14.11.2019)

20. What's the Difference Between Hardware Keylogger and Keylogger Software [Електронний ресурс] //EaseMon - 2020. - Режим доступу до ресурсу: https://www.easemon.com/whats-the-differences-between- hardware-keylogger-and-keylogger-software.html

21. Detecting Hardware Keyloggers [Електронний ресурс] //Mihailowitsch, F., DeepSec - 2010. - Режим доступу до ресурсу: https://deepsec.net/docs/Slides/2010/DeepSec_2010_Detecting_Hard-ware_Keylogger.pdf

22. Was That Always There? A Hardware Keylogger Threat [Електронний ресурс] //Ipswich Jablokow - 2017. - Режим доступу до ресурсу: https://blog.ipswitch.com/was-that-always-there-a-hardware- keylogger-threat

23. Keylogger Attacks on Banking Apps Increase [Електронний ресурс] //Promon, security news - 2018. - Режим доступу до ресурсу: https://promon.co/security-news/keylogger-banking-apps/

24. 'Invisible Man' malware runs keylogger on your banking apps [Електронний ресурс] //Thomson, I. - 2017. - Режим доступу до ресурсу: https://www.theregis-ter.co.uk/2017/08/02/banking_android_malware_in_uk/

25. Mimikatz [Програмний застосунок] //Benjamin DELPY - 2011. - Режим доступу до ресурсу: https://github.com/gentilkiwi/mimikatz/wiki

26. What is DevOps? [Електронний ресурс] //CyberArk –2019. - Режим доступу до ресурсу: https://www.cyberark.com/what-is/devops-security/#.

27. New research: How effective is basic account hygiene at preventing hijacking [Електронний ресурс] //Kurt Thomas and Angelika Moscicki – 2019. - Режим доступу до ресурсу: https://security.googleblog.com/2019/05/new-research-how-effective-is-basic.html

28. Case Study #2: Offering Two-Factor Authentication [Електронний ресурс] //K. Bankston, R.Schulman – 2018. - Режим доступу до ресурсу: https://www.newamerica.org/in-depth/getting-internet-companies-do-right-thing/case-study-2-offering-two-factor-authentication/.

29. Cybersecurity in the Age of the COVID-19 Remote Worker and Beyond [Електронний ресурс] //Brian G. Cesaratto – 2020. - Режим доступу до ресурсу: https://www.workforcebulletin.com/2020/05/06/cybersecurity-in-the-age-of-the-covid-19-remote-worker-and-beyond/

30. The Basics of Information Security (Second Edition) // Jason Andress - 2014.

31. A Study on Efficient OTP Generation using Stream Cipher with Random Digit [Електронний ресурс] //Young Sil Lee, HyoTaek Lim, HoonJae Lee - 2010. - Режим доступу до ресурсу: https://www.researchgate.net/publication/224128295_A_study_on_efficient_OTP_generation_using_stream_cipher_with_random_digit -2010.

32. YubiKey 5 Series [Електронний ресурс] //Yubico - 2021. - Режим доступу до ресурсу: https://www.yubico.com/products/yubikey-5-overview/

33. Market Share Statistics for Internet Technologies [Електронний ресурс] //NetMarket Share - 2017. - Режим доступу до ресурсу: https://netmarketshare.com/operating-system-market-share

34. Windows Authentication Concepts [Електронний ресурс] //Microsoft – 2016. – Режим доступу до ресурсу: https://docs.microsoft.com/en-us/windows-server/security/windows-authentication/windows-authentication-concepts

35. Hardening Linux authentication and user identity 2004 [Електронний ресурс] //Linux.com – 2014. - https://www.linux.com/news/hardening-linux-authentication-and-user-identity/#.

36. Pluggable Authentication Modules [Електронний ресурс] //Wikipedia. – 2014. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/Pluggable_Authentication_Modules

37. OpenLDAP [Програмний застосунок] // OpenLDAP. - Режим доступу: https://www.openldap.org/

38. Pros and Cons of Cloud Storage [Електронний ресурс] // Secure Storage Services Ltd – 2020. - Режим доступу до ресурсу: https://www.securestorageservices.co.uk/article/11/pros-and-cons-of-cloud-storage

39. Роз'яснення щодо питань використання хмарних технологій банками України [Електронний ресурс] //Національний Банк України – 2017. - Режим доступу до ресурсу: https://gigacloud.ua/uploads/0/416-nbu.pdf

40. Аутентификация в системах Windows. Часть 1 [Електронний ресурс] //Уваров А.С. – 2016. – Режим доступу до ресурсу: NTLMhttps://interface31.ru/tech_it/2015/03/autentifikaciya-v-sistemah-windows-chast-1-ntlm.html

41. HMACMD5 Class [Електронний ресурс] //Microsoft. – 2021. - Режим доступу до ресурсу: https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography.hmacmd5?view=net-5.0

42. Аутентификация в системах Windows. Часть 2 - Kerberos [Електронний ресурс] //Уваров А.С. – 2016. – Режим доступу до ресурсу: https://interface31.ru/tech_it/2016/07/autentifikaciya-v-sistemah-windows-2-kerberos.html

43. Authentication Protocols: LDAP vs Kerberos vs OAuth2 vs SAML vs RADIUS [Електронний ресурс] //Bernhard Mehl. – 2018. - Режим доступу до ресурсу: https://www.getkisi.com/blog/authentication-protocols-overview

44. Единый вход с использованием SAML [Електронний ресурс] //Ispring. – 2021. - Режим доступу до ресурсу: https://www.ispring.ru/articles/ediniy-vhod-s-ispolzovaniem-saml

45. "Password Management", T. G. o. t. H. K. S. A. Region, Ed., ed, - 2008.

46. Technical report on SS7 vulnerabilities and mitigation measures for digital financial services transactions [Електронний ресурс] //ITU – 2020. - Режим доступу до ресурсу: https://figi.itu.int/wp-content/uploads/2021/04/Technical-report-on-the-SS7-vulnerabilities-and-their-impact-on-DFS-transactions_f-1-1.pdf

47. Stealthy SS7 Attacks // Sergey Puzankov – 2017. Positive Technologies

48. Types of Bluetooth Attacks in 2020 [Електронний ресурс] //Signils – 2020. - Режим доступу до ресурсу: https://www.signils.com/types-of-bluetooth-attacks-in-2020/

49. Near Field Communication (NFC) Technology, Vulnerabilities and Principal Attack Schema [Електронний ресурс] //Pierluigi Paganini – 2013. - Режим доступу до ресурсу: https://resources.infosecinstitute.com/topic/near-field-communication-nfc-technology-vulnerabilities-and-principal-attack-schema/

50. A support architecture for multichannel, multifactor authentication. [Електронний ресурс] //Karen Renaud, Richard Cooper, Mohamed Al Fairuz – 2014. - Режим доступу до ресурсу: https://www.researchgate.net/publication/242266843

51. IntelliTrust Authentication Service [Електронний ресурс] //Entrust – Режим доступу до ресурсу: https://www.entrust.com/digital-security/identity-and-access-management

52. MFA as protection against account compromise [Електронний ресурс] //Kateryna Mokliakova, Vira Ignisca – 2021. - Режим доступу до ресурсу: http://kbzi.knu.ua/2021/04/16/15-16

53. Wallix Bastion [Програмний застосунок] //Wallix. - Режим доступу: https://www.wallix.com/

54. LinOTP [Програмний застосунок] //LinOTP. – Режим доступу: http://linotp.org/about.html

55. FreeRADIUS [Програмний застосунок] //FreeRADIUS. - Режим доступу: https://freeradius.org/

56. Google Authenticator [Програмний застосунок/Мобільний додаток] //Google Inc. – Режим доступу: https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2&hl=ru&gl=US

57. LinOTP — инсталляция и использование [Електронний ресурс] //Kateryna Mokliakova. – 2021. - Режим доступу до ресурсу: https://habr.com/ru/post/556808/

58. Credentials Processes in Windows Authentication [Електронний ресурс] //Microsoft. – 2016. - Режим доступу до ресурсу: https://docs.microsoft.com/en-

us/windows-server/security/windows-authentication/credentials-processes-in-windows-authentication

59. MariaDB [Програмний застосунок] //MariaDB Foundation. - Режим доступу: https://mariadb.org/

60. Selfservice Portal [Електронний ресурс] //LinOTP. – 2021. - Режим доступу до ресурсу:http://linotp.org/doc/latest/part-management/selfservice/index.html