ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА ІНФОРМАЦІЙНОЇ ТА КІБЕРНЕТИЧНОЇ БЕЗПЕКИ

Пояснювальна записка

до магістерської роботи
на тему:

«ТЕХНОЛОГІЇ ЗАХИСТУ СЕРВІСІВ ДЛЯ МИТТЄВОГО ОБМІНУ ПОВІДОМЛЕННЯМИ TELEGRAM ТА WHATSAPP ВІД БОТІВ ТА СПАМУ»

Виконав: студент 6 курсу, групи БСДМ-61
Спеціальності 125 Кібербезпека
Освітньо-професійної програми
«Інформаційна та кібернетична безпека»
(шифр і назва спеціальності)
Костирєв Л.В
(прізвище та ініціали)
Керівник Довженко Н.М.
(прізвище та ініціали)
Рецензент _____
(прізвище та ініціали)
Нормоконтролер Чумак Н.С.

КИЇВ – 2020

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Інститут ННІЗІ

Кафедра Інформаційної та кібенетичної безпеки

Ступінь вищої освіти Магістр

Спеціальність 125 Кібербезпека

Освітньо-професійна програма Інформаційна та кібернетична безпека

ЗАТВЕРДЖУЮ
Завідувач кафедри ІКБ
_____ Г.І. Гайдур
“___” _____ 2020 року

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

Костирєву Леоніду Віталійовичу
(прізвище, ім’я, по батькові)

1. Тема магістерської роботи: «Технології захисту сервісів для миттєвого обміну повідомленнями Telegram та WhatsApp від ботів та спаму»

керівник магістерської роботи Довженко Надія Михайлівна, к.т.н., доцент
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «13» жовтня 2020 року №230

2. Строк подання студентом магістерської роботи 15.12.2020 року

3. Вихідні дані до магістерської роботи

1) Сервіси для миттєвого обміну повідомленнями

2) Політики безпеки

3) Технічні системи захисту

4) Наукова література. Міжнародні стандарти.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз особливостей функціонування сервісів для миттєвого обміну повідомленнями Telegram та WhatsApp.

2. Методи та засоби забезпечення безпеки в сервісах для миттєвого обміну повідомленнями.

3. технологій захисту сервісів для миттєвого обміну повідомленнями.

4. Розробка системи захисту інформації в соціальних мережах

5. Перелік графічного матеріалу

1. Обєкт, предмет та мета дослідження
2.
3.
4.
5.
6.
7.
8.
9. Висновки.

6. Дата видачі завдання_____07.10.2020р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Примітка
1.	Аналіз науково-технічної літератури	09.10.2020р.	виконано
2.	Аналіз особливостей функціонування сервісів для миттєвого обміну повідомленнями Telegram та WhatsApp.	14.10.2020р.	виконано
3.	Дослідження проблем забезпечення безпеки в сервісах для миттєвого обміну повідомленнями.	25.10.2020р.	виконано
4.	Дослідження технологій захисту сервісів для миттєвого обміну повідомленнями.	15.11.2020р.	виконано
5.	Розробка прототипу	21.11.2020р.	виконано
6.	Реферат, вступ, висновки.	08.11.2020р.	виконано
7.	Підготовка презентації до захисту.	14.12.2020р.	виконано

Студент _____ Сергієнко М.А.
(підпис) (прізвище та ініціали)

Керівник магістерської роботи _____ Довженко Н.М.
(підпис) (прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

ПОДАННЯ
ГОЛОВІ ДЕРЖАВНОЇ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ МАГІСТЕРСЬКОЇ РОБОТИ

Направляється студент **Костирєв Л.В.** до захисту магістерської роботи
(прізвище та ініціали)
спеціальності 125 Кібербезпека
(шифр і назва спеціальності)
на тему: "Дослідження шляхів та розробка рекомендацій щодо захисту мереж іот від кібератак»
Магістерська робота і рецензія додаються.
Директор інституту _____ В. А.Савченко
(підпис) (прізвище та ініціали)

Довідка про успішність

Костирєв Л.В. за період навчання в інституті
(прізвище та ініціали студента)
ННІЗІ з 2018 року до 2020 року повністю виконав навчальний план за напрямом підготовки, спеціальністю з таким розподілом оцінок за:
національною шкалою: відмінно ____%, добре ____%, задовільно ____%;
шкалою ECTS: A ____%; B ____%; C ____%; D ____%; E ____%.
Секретар інституту, факультету (відділення) _____ _____
(підпис) (прізвище та ініціали)

Висновок керівника магістерської роботи

Студент **Костирєв Л.В.** обрав тему роботи, метою якої було дослідити шляхи та розробити рекомендації щодо захисту мереж IoT від кібератак. Перелік використаних джерел свідчить про вміння дипломником розбиратись в наукових питаннях та застосовувати їх при дослідженнях. Під час виконання магістерської роботи **Костирєв Л.В**. показав відмінну теоретичну та практичну підготовку, вміння самостійно вирішувати питання і робити висновки.
Роботу виконував сумлінно, акуратно та вчасно за планом.
Все це дозволяє оцінити виконану магістерську роботу студента **Костирєв Л.В**. на оцінку **«відмінно»** та присвоїти кваліфікацію 2149.2 професіонал з організації інформаційної безпеки, викладач вищих навчальних закладів.
Керівник магістерської роботи _____ Довженко Н.М.
(підпис)
"____" _____ 2020 року

Висновок кафедри про магістерську роботу

Магістерська робота розглянута. Студент **Костирєв Л.В.**
(прізвище та ініціали)
допускається до захисту даної магістерської роботи в Державній екзаменаційній комісії.
Завідувач кафедри Інформаційної та кібернетичної безпеки
(назва)
_____ Гайдур Г.І.
(підпис) (прізвище та ініціали)
"____" _____ 2020 рік

РЕЦЕНЗІЯ
на магістерську роботу

студента **Костирєва Леоніда Віталійовича**

на тему: **«Технології захисту сервісів для миттєвого обміну повідомленнями Telegram та WhatsApp від ботів та спаму»**

Актуальність: Одним із найважливіших напрямків розвитку ринку телекомунікацій України є формування та розвиток національного сегмента інформаційного суспільства і входження нашої країни до світової інформаційної спільноти. Робота присвячена одному з найперспективніших напрямків розвитку телекомунікацій – безпеці мережам Інтрнету речей. Технології системи з самого початку орієнтовані на полегшення та автоматизування великої кількості процессів у різних галузях. Сьогодні існує кілька підходів до побудови безпеки данному типу мереж. Тому тема магістерської роботи є актуальною і своєчасною.

Позитивні сторони:

1. Обґрунтовано необхідність розробки рекомендацій щодо захисту мереж Інтрнету речей.
2. Розроблено рекомендації для підвищення ефективності захисту від кібератак.

Недоліки:

1. При аналізі існуючих засобів забеспечення безпеки доцільно було зробити порівняльну таблицю.
2. Із роботи не зрозуміло яким чином впроваджувати рекомендації в реальні мережі.

Вищезгадані зауваження не впливають на загальну позитивну оцінку бакалаврської роботи.

Висновок: робота заслуговує оцінку "відмінно", а студент – Косенко Владислав Віталійович гідний присвоєння кваліфікації професіонал з організації інформаційної безпеки, викладач вищих навчальних закладів.

Якість роботи	
Виконано на замовлення підприємства	
Виконано за тематикою НДР	
Виконано з макетом	
Виконано з застосуванням ЕОМ та МПТ	√
Має практичну цінність	√
Проект-частина комплексної теми	

Підпис рецензента

_____

_____

(П.І.Б.)

_____

(посада, науковий ступінь, вчене звання)

РЕФЕРАТ

Текстова частина магістерської роботи: 88 сторінок, 15 рисунків, 8 таблиць, 27 джерел.

Об'єкт дослідження – сервіси для миттєвого обміну повідомленнями Telegram та WhatsApp.

Предмет дослідження – засоби та технології захисту користувачів Telegram та WhatsApp від ботів та спаму.

Мета роботи – полягає в проектуванні та дослідженні методів забезпечення захисту сервісів для миттєвого обміну повідомленнями Telegram та WhatsApp.

Методи дослідження – теорія складних систем, теорія масового обслуговування, теорія інформації, практичне тестування програмного забезпечення.

В роботі проаналізовано…

Проведено глибинний аналіз….

Досліджено принципи…

Виокремлено основні…

Зазначено….

Проаналізовано…..

Досліджено……..

Галузь використання –.

ЧАТ-БОТ, МЕСЕНДЖЕРИ, СЕРВІСИ МИТТЄВОГО ОБМІНУ ПОВІДОМЛЕННЬ, БОТ, МЕРЕЖА, СИСТЕМА, КІБЕРАТАКА, АТАКА, ХМАРА, ПЕРЕДАЧА ДАННИХ.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

АСУ – автоматизована система управління

Меседжер – сервіс для миттєвого обміну повідомленнями

БД – база даних

ЕОМ – електронна обчислювальна машина

ІБ – інформаційна база

ІДД – індексне поле, в якому допускається дублювання значень

ІЗ – інформаційне забезпечення

ІС – інформаційна система

ІНД - індексне поле, в якому не допускається дублювання значень

КТЗ – комплекс технічних засобів

ЛОМ - локальна обчислювальна мережа

ОС – операційна система

ПЕОМ – персональна електронна обчислювальна машина

ПК – персональний комп’ютер

СУБД - система керування базами даних

ПГК – Приватна група користувачів

ПЗ – програмне забезпечення

ВГК – відкрита група користувачів

Mb – мега байт

IP-адреса – Цифрова адреса мережевого обладнання

Порт – Цифрова додаткова адреса до ip-адресу, використовується службами ОС для роботи в мережі.

Сокет – відношення ip-адреси до порта, наприклад 192.168.0.1:80

СППР – система підтримки прийняття рішень

ОПР– особа котра приймає рішення

Спамер – особа котра розповсюджує спам

ВСТУП

Актуальність дослідження. Проблема спаму у сервісах для миттєвого обміну повідомленнями Telegram та WhatsApp набула особливу гостроту в останнє десятиліття, у зв'язку з вкрай неефективним функціонуванням державної системи забезпечення безпеки користувачів таких сервісів та необізнаності самих користувачів в умовах розвитку диджиталізації, та диспропорцією, що збільшується, між ростом числа користувачів та ростом кількості користувачів, що не достатньо обізнані з правилами безпечного використання меседжерів. Сучасні європейські дослідження показують, приблизно 90% усіх кіберзагроз припадають на приватний сектор. Тобто, це не є цільові атаки на великі компанії, або окремих користувачів Той, рядовий користувач Telegram, або WhatsApp наражається на вдесятеро більший ризик стати жертвою кібератаки ніж підприємство. Це логічне твердження також засновано на тому, що в корпоративному секторі користувач більш обізнаний ніж у приватному. Джерелом більшості таких кібератак є спам.

Багато кібератак розраховані саме на неуважність та необізнаність користувачів. Наприклад Homograph Attack(фішінг), де URL фішінгового сайту має мінімальні, а іноді і неспостерігаємі відрізності від сайту оригіналу (наприклад англійську літеру «о» замінили на кириличну літеру «о»). Користувач відкриває фішинговий сайт, бачить знайомий(зкопійований з оригіналу) дизайн сайту, вводить туди свій логін та пароль, або данні кредитної карти і ними заволодіває зловмисник.

У зв’язку з середнім щорічним збільшенням кількості викрадених данних банківськіх карт в Україні на 4-5 %, відповідальність за безпеку персональних данних лягає на плечі користувача.

Об’єкт дослідження – сервіси для миттєвого обміну повідомленнями Telegram та WhatsApp.

Предмет дослідження – засоби та технології захисту користувачів Telegram та WhatsApp від ботів та спаму.

Мета роботи – полягає в проектуванні та дослідженні методів забезпечення захисту сервісів для миттєвого обміну повідомленнями Telegram та WhatsApp.

Відповідно до мети наукового дослідження були поставлені та розв'язані наступні завдання:

- досліджена характеристика об'єкта автоматизації;
- досліджені джерела спаму;
- досліджені налаштування внутрішні налаштування сервісів що
- досліжені API меседжерів;
- описано характеристики програмної реалізації анти-спам боту для відкритих та закритих груп користувачів;
- досліджено проектування користувацького інтерфейсу для роботи з анти-спам боту;
- розроблені рішення за видами забезпечень (програмне, технічне,організаційне) ;
- досліджено економічну ефективність проекту.

Результати, досягнуті в процесі роботи дозволяють автоматизувати процеси підтримки прийняття рішень в управлінні безпекою групп користувачів.

Методи дослідження – теорія складних систем, теорія масового обслуговування, теорія інформації, практичне тестування програмного забезпечення.

Практичне значення одержаних результатів полягає в використані для практичного проектування та реалізації системи Інтернет-комерції.

Апробація результатів:

1 АНАЛІЗ ОСОБЛИВОСТЕЙ ФУНКЦІОНУВАННЯ СЕРВІСІВ ДЛЯ МИТТЄВОГО ОБМІНУ ПОВІДОМЛЕННЯМИ TELEGRAM ТА WHATSAPP

1.1 Характеристика об’єкта дослідження – спам в сервісах миттєвого обміну повідомленнями Telegram та WhatsApp

WhatsApp - популярна безкоштовна система миттєвого обміну текстовими повідомленнями для мобільних і інших платформ з підтримкою голосового і відеозв'язку. Дозволяє пересилати текстові повідомлення, зображення, відео, аудіо, електронні документи та навіть програмні установки через Інтернет.
WhatsApp Messenger доступний для iPhone, BlackBerry, Windows Phone, Android і Symbian .

У лютому 2016 року WhatsApp досяг 1 мільярда активних користувачів на місяць, і був найбільш часто використовуваних месенджером на сьогоднішній день.

Telegram — Кросплатформенна програмне забезпечення, яке дозволяє обмінюватися текстовими, аудіо та відео повідомленнями та файлами, а також безкоштовно телефонувати іншим користувачам програми.

MTProto(Рисунок 1.1) - криптографічний протокол, який використовується в системі обміну повідомленнями Telegram для шифрування листування користувачів.

Аналіз протокола та окремих елементів його реалізації в мессенджере Telegram проводився за матеріалами з відкритих джерел.

AES - симетричний 256-бітний алгоритм, прийнятий урядом США як стандарт.

RSA - криптографічний алгоритм, в основі якого лежить обчислювальна складність завдання факторизації великих цілих чисел.

Метод Діффі-Хеллмана - дозволяє отримати двом і більш співрозмовникам секретний ключ по незахищеному від прослуховування, однак захищеному від підміни каналу зв'язку.

SHA-1, MD5 - хеш-алгоритми, використовувані в багатьох криптографічних протоколах і додатках для безпечного хешірованія.В відміну від протоколу Double Ratchet, який застосовується WhatsApp і вже встиг отримати схвалення відомих фахівців в області захисту інформації, розробники MTProto не поспішають надавати свій продукт для незалежного аудиту. З одного боку, це робить алгоритм потенційно вразливим для атак, з іншого - на сьогоднішній день не зафіксовано жодного успішного дії, що призвів до розшифровки сообщеній.

Творці месенджера заявляють про гарантії безпеки у відношенні передачі зашифрованих даних. Для підтвердження своїх слів Павло Дуров періодично організовує конкурси, в яких учасникам пропонується розшифрувати листування двох співрозмовників. Призовий форд становить 200 тис. Доларів, однак до теперішнього часу жоден хакер не зміг прочитати зашифровані повідомлення. Справедливості заради слід зазначити, що багато експертів досить скептично ставляться до подібних конкурсів, вважаючи їх скоріше продуктами піару, ніж реальним доказом захищеності системи.

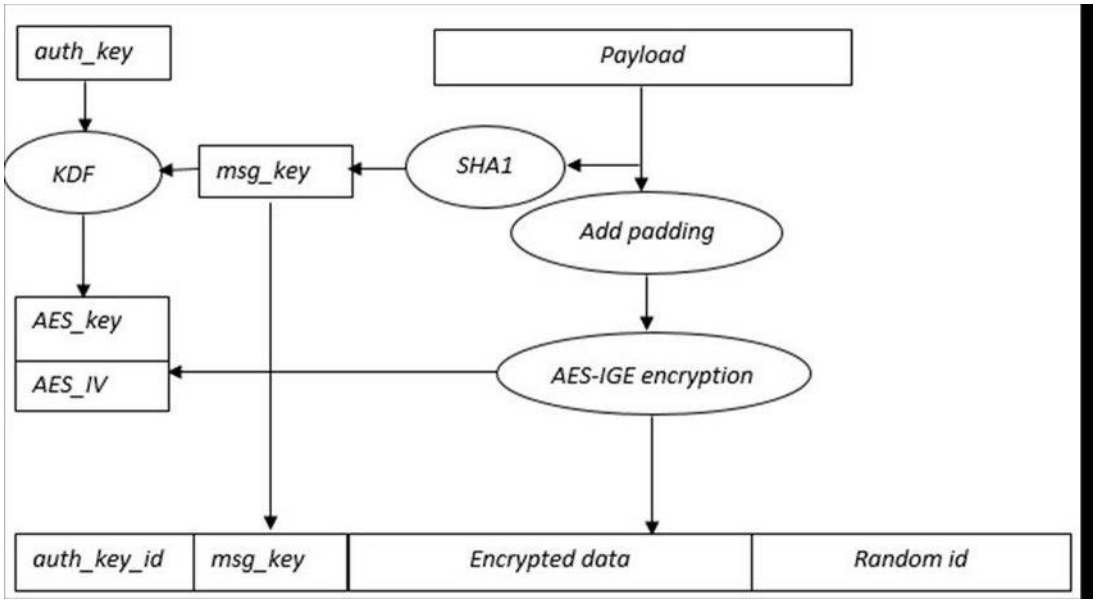


Рис.1.1. Процесс шифрування в секретних чатах телеграм

В основі протоколу лежить оригінальна комбінація симетричного алгоритму шифрування AES (в режимі IGE), протокол Діффі-Хеллмана для обміну 2048-бітними RSA-ключами між двома пристроями та ряд хеш-функцій. Протокол допускає використання шифрування end-to-end з опціональною звіркою ключів.

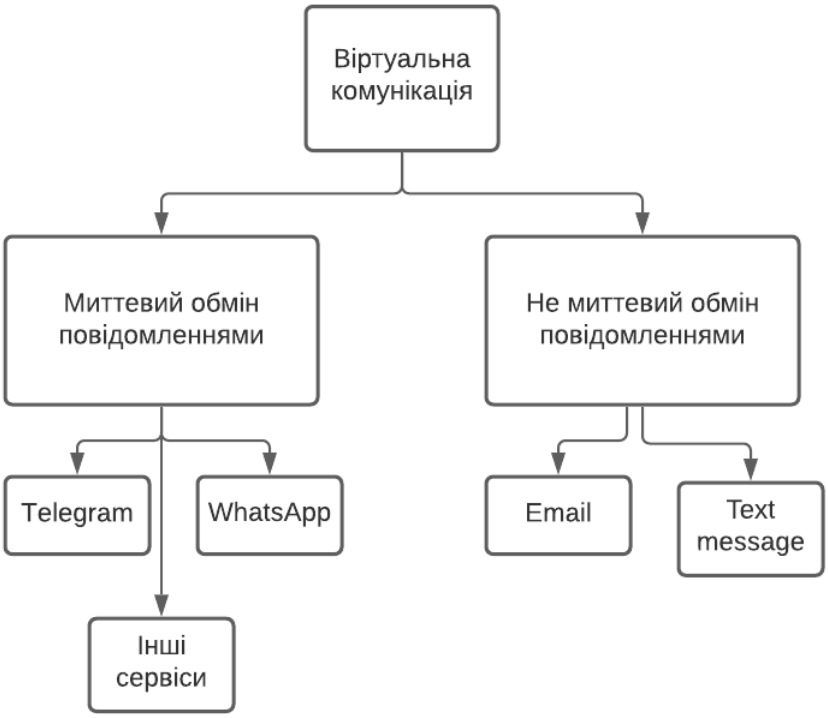


Рисунок 1.2

Як видно з Рисунка 1.2, віртуальне спілкування можна розділити на дві різні типи концепцій обміну повідомленнями. З технічної точки зору спеціальні програми обміну миттєвими повідомленнями - це комп’ютерні програми, які використовують WebSocket, або інші протоколи зв'язку в режимі реального часу, що забезпечують надійність і швидкість повторногодля передачі тексту в режимі реального часу.

Просте розрізнення, яке можна було легко помітити під час роботи з сервівсів обміну миттєвими повідомленнями - це можливість спостерігати за діями іншої людиниспілкування. Наприклад, у сучасних програмах обміну миттєвими повідомленнями можна подивитись, як друга людина друкує в той же час, тоді як це поки неможливо використати в електронних поштових клієнтах або

під час надсилання текстових повідомлень через мережу телефонного оператора.Чат-боти, які є новими на сцені сучасних комунікаційних програм, повинні бути класифіковані. Оскільки чат-боти за своєю суттю використовують ті ж протоколи зв'язку, що не миттеві програми обміну повідомленнями, вони повинні бути класифіковані в розділі Миттєві повідомлення.Ілюстрація високого рівня архітектури віртуального спілкування може бути більш складною і брати до уваги спілкування, засноване на людині, як текстове спілкування з другог та комп’ютерне спілкування, як текстовий спілкування за допомогою чатуботів. Однак цього рівня абстракції та деталей архітектури достатньо для розуміння основ чат-ботів та як вони вписуються в загальну картину.

Спам — масове розсилання кореспонденції рекламного чи іншого характеру людям, які не висловили бажання її одержувати. Передусім термін «спам» стосується рекламних електронних листів. В нашому випадку це повідомлення що містять не бажану інформацію, рекламу, фішінгові-сторінки, Drive-by Download сторінки.

Фішинг - вид інтернет-шахрайства, метою якого є отримання доступу до конфіденційних даних користувачів - логінів і паролів. Це досягається шляхом проведення масових розсилок електронних листів від імені популярних брендів, а також особистих повідомлень всередині різних сервісів, наприклад, від імені банків або всередині соціальних мереж. У листі часто міститься пряме посилання на сайт, зовні відрізнити від справжнього, або на сайт з перенаправленням. Після того як користувач потрапляє на підроблену сторінку, шахраї намагаються різними психологічними прийомами спонукати користувача ввести на підробленої сторінці свої логін і пароль, які він використовує для доступу до певного сайту, що дозволяє шахраям отримати доступ до акаунтів і банківських рахунків. По іншому явище фішингу можна описати як один з різновидів соціальної інженерії, заснована на незнанні користувачами основ мережевої безпеки: зокрема, багато хто не знає простого факту: сервіси не розсилають листів з проханнями повідомити свої облікові дані, пароль та інше.

Для захисту від фішингу виробники основних інтернет-браузерів домовилися про застосування однакових способів інформування користувачів про те, що вони відкрили підозрілий сайт, який може належати шахраям. Нові версії браузерів вже мають таку можливість, яка відповідно іменується «антифішинг».

Фішингові посилання - це посилання на шахрайські інтернет-ресурси, найчастіше на копії сайтів відомих організацій, банків, інтернет-магазинів, соціальної мережі і т.д. Перейшовши по такому посиланню ви потрапляєте на ресурс, де вас різними прийомами намагаються змусити ввести свій логін і пароль.

Зазвичай такі посилання вам надсилають поштою або в особистому повідомленні, наприклад, в соціальних мережах. Це або прямі посилання, перейшовши за якими ви потрапляєте на сайт, який майже нічим не відрізняється від справжнього, або посилання з перенаправленням (переадресацією), перейшовши за якими ви перенаправляє на інші сайти і в кінцевому підсумку потрапляєте на ресурс шахраїв. Потрапивши на такий сайт, ви можете відразу і не зрозуміти, що перебуваєте в пастці, а шахраї тим часом намагаються з вас вивудити потрібну їм інформацію у вигляді логіна і пароля.

Існує кілька видів фішингових посилань:

Пряма - посилання веде на ту ж сторінку, що і її адресу.

З перенаправленням - посилання веде вас спочатку на один сайт, потім перенаправляє на інший, потім на третій і т.д.

Прихована - посилання зовні виглядає правильно, але в реальності веде на фішингову сторінку.

Фішинговий сайт - це сайт, який повністю або частково скопійований з оригінального, але таким не є. Метою таких сайтів є розкрадання логіна і пароля, який ви використовуєте на оригінальному сайті. Користувач переходить за фішинговою посиланням і бачить перед собою звичайний портал, яким постійно користується. Нічого не підозрюючи вводить свої логін і пароль, які відразу ж стають відомі зловмисникам.

Такі сайти зовні повністю копіюють оригінальні, але якщо придивитися до адресою в адресному рядку, то ви побачите в ньому помилку.

Існує багато різних технічних способів захисту від фішингу, але «слабка ланка» в цьому ланцюзі - сам користувач, його цікавість і неуважність дозволяють шахраям ефективно застосовувати методи соціальної інженерії.

- Потрібно бути уважним до того де і що ви вводите.
- Не переходьте по дивним і не перевіреним вами посиланнями.
- Обов'язково звертайте увагу на адресу сторінки в адресному рядку сторінки.

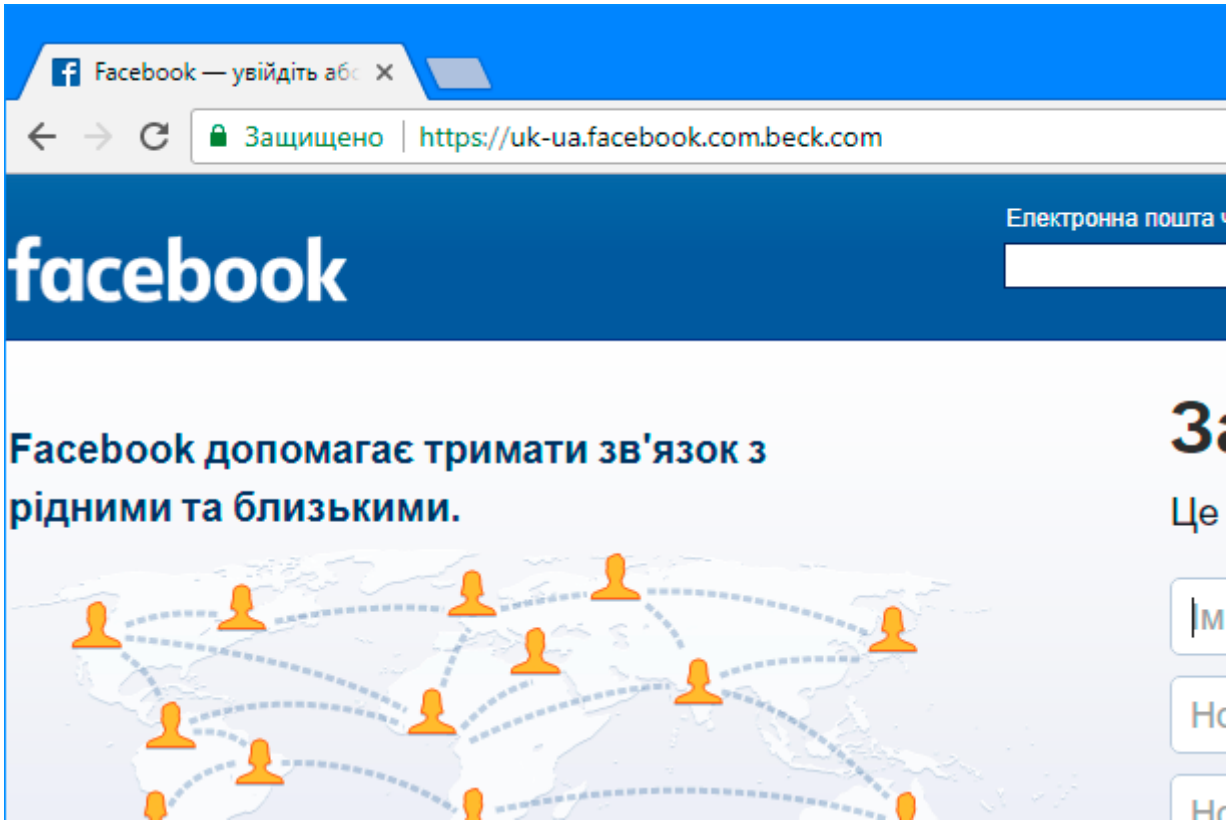


Рисунок 1.3 Фішінгове посилання

Якщо ви випадково ввели свій логін і пароль на ресурсі зловмисників і тільки потім помітили, що це обман, то відразу ж зайдіть на справжній сайт і змініть ті дані, які ви вводили - пароль, по можливості логін, секретні питання для відновлення даних. Також, обов'язково повідомте про те, що сталося в службу підтримки ресурсу.

Зазвичай портали роблять розсилку про спробу злому акаунтів і про те, що користувачі можуть отримати лист з шахрайською посиланням.

З цього випливає що джерелами спаму у меседжерах є приватні, публічні групові чати, а також особисті чати.

Спам з'являється за наступнии сценарієм:

- учасник групи додає спам-бота в групу
- учасник додає спамера в групу
- користувач власноруч запускає спам-бота
- спамер пише вам особисто

Спам направлений на користувача

Майже всі користувачі меседжерів, колись в своєму житті зіштовхувались зі спамом.

1.2. Поняття користувачів, груп, ботів та каналів та їх класифікація

Чати в меседжерах класифікують: за кількістю людей; за типом; за ступенем приватності.

Види чатів за кількості людей: Індивідуальні чати та Групові чати.

Види чатів за типом:

- Звичайний чат – вид чату, де два користувачі спілкуються між собою.

Може бути поділеним, ще на два типи:

- Контакт – доданий вами до списку контактів користувач сервісу.
- Не контакт – не доданий вами до списку контактів користувач сервісу, що додав вас у свій список контактів
- Канал – вид чату, де один користувач є автором, а всі інші читачами.

З концептуальної точки зору канали дають читачам, з одного боку, можливість відчути себе на одному рівні з автором (публікації каналів виглядають так само, як і обмін особистими повідомленнями), а з іншого - дозволяють користувачам споживати контент в зручній системі координат в форматі окремого діалогу (відштовхуючись від хронології публікації матеріалів).

- Група користувачів – це окремий вид чатів, в якому можливе спілкування відразу великої кількості учасників. Вони бувають публічні

(запрошення за посиланням) та приватні(запрошувати можуть лише члени групи, або виключно адмінстратор). Створювати групи можуть всі користувачі. Для кращого управління групою творець може призначати адміністраторів і визначати їх функціонал.

- Чат бот - спеціальна програма, що виконує автоматично, або за заданим розкладом будь-які дії через інтерфейси, призначені для людей. Таких ботів можна використовувати в якості спамерів. Бот виглядає як звичайний чат, однак спілкування відбувається не з людиною, а з програмою, яка може прийняти замовлення на виклик машини, якщо це - бот для онлайн замовлень, або надіслати свіжі статті, якщо це - новинний бот. Не так давно чат-боти здобули велику популярність, перевтілившись з розваги в більш серйозну річ, так як вони, восновному, стали використовуватися для вирішення серйозних бізнес-завдань. В епоху інформаційних технологій - це нормальне явище, а тим більше – мережі Інтернет, адже суспільство давно перейшло на «нове» і ділове, і не ділове спілкування. По-перше, чат-боти - це «платформи» для вирішення бізнес-завдань. По-друге, чат-бот - це програма, яка підтримує діалог з користувачем, вибираючи відповіді з бази даних: ви запитуєте, де пообідатиі тут же отримуєте миттєвий відповідь. Крім того, чат-боти виконують безліч корисних функцій у виконанні рутинних операцій, пошуку інформації, об'єднання даних, роботі з клієнтами. Чат-бот як віртуальний співрозмовник має базу знань, яка являє собою набори можливих питань користувача і відповідних їм відповідей. Найбільш поширеними варіантами дляотримання потрібної відповіді є ключові слова, збіг фрази, збіг контексту. Завжди існують якісь прості і легкі повиконання справи, на які не хочеться витрачати час. Тут на допомогу завждиможуть прийти чат-боти.

Види чатів за ступенем приватності: секретні чати та звичайні чати.

Секретні чати (Secret Chats) - окремий вид чатів, які використовують наскрізне шифрування. Історія повідомлень в таких чатах не зберігається в хмарі, а тільки на пристроях відправника і одержувача. З Секретних чатів Ви не зможете пересилати повідомлення іншим користувачам, а лише цитувати їх безпосердньо всередині цього чату.

У Секретних чатах є можливість встановити таймер для видалення повідомлень або файлів через певний час після їх прочитання або відкриття. Коли встановлений час мине, повідомлення віддалиться і для відправника, і для одержувача.

Наскрізне шифрування (також кінцеве шифрування; англ. End-to-end encryption) - спосіб передачі даних, в якому тільки користувачі, які беруть участь в спілкуванні, мають доступ до повідомлень. Таким чином, використання наскрізного шифрування не дозволяє отримати доступ до криптографічних ключів з боку третіх осіб.

Для обміну ключами можуть бути застосовані симетричний і асиметричний алгоритми. Наскрізне шифрування передбачає, що ключі шифрування відомі тільки спілкується між собою сторонам. Для реалізації даної умови може бути використана схема з попередніми поділом секрету або, наприклад, протокол Діффі-Хелмана, який використовується в месенджерах WhatsApp і Telegram

Наскрізне шифрування гарантує, що доступ до вихідного тексту повідомлення є тільки у відправника і одержувача. Це означає, що призначена для користувача інформація стає недоступною навіть серверів, які передають дані.

Шифрування і дешифрування відбувається на кінцевих пристроях користувачів. Крім того, дані залишаються зашифрованими, поки не будуть доставлені до місця призначення. Тому часто наскрізне шифрування також називають «нульовий доступ» або «шифрування на стороні клієнта». Однак, слід розрізняти кінцеве шифрування при передачі даних і шифрування на стороні клієнта при зберіганні даних.

Спам бот - Програми, які завантажили інтернет-канал потоком непотрібної інформації як правило, рекламного характеру.

Творці месенджера мають великий досвід в роботі з соціальними мережами. Саме тому відразу були прийняті жорсткі заходи по боротьбі з небажаними повідомленнями у вигляді спеціальної кнопки «спам». Виглядає це так. Якщо вам пише першим користувач не зі списку контактів, ви маєте право поскаржитися на його повідомлення, натиснувши кнопку «спам». В результаті користувач

блокується і вже не може вам писати, а на нього надходить скарга. Скарги накопичуються і призводять до обмеження аккаунта на вихідні повідомлення, тимчасово або назавжди. Начебто ідеальна схема, на перший погляд. Причому з висловлювань розробників, всі скарги розглядаються, а неправомірні відхиляються.

В меседжерах що розглядаються в роботі слід зазначити, що секретний чат відрзняється від звичайного, лише наявністю наскрізного шифрування, що є лише додатковую мірую безпеки, але ніяк не захищає користувачів від зловмисників, які наприклад викрали аккаунт вашого знайомого.

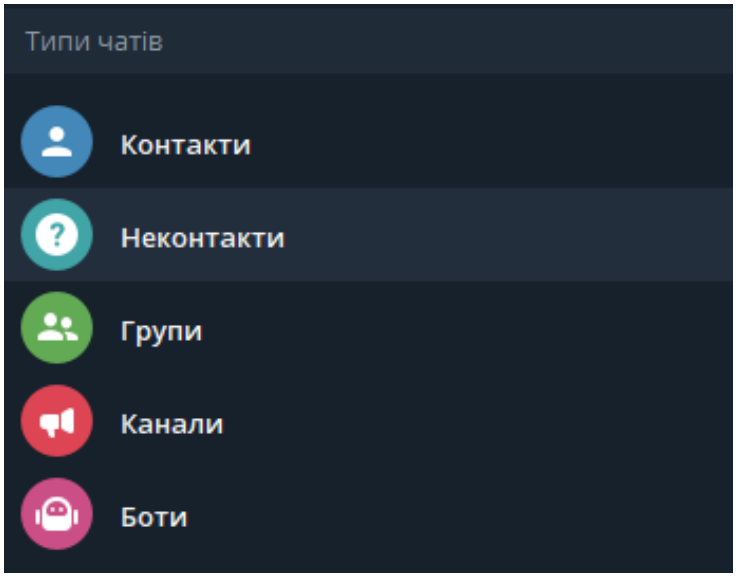


Рисунок 1.4. Типи чатів

1.3 Характеристика програмної реалізації анти-спам боту

Парсер - це ПО, що виділяє певні частини інформації з масиву даних. Алгоритм роботи парсера може відрізнятися в різних реалізаціях, але основний принцип залишається незмінним. У розробці анти-спам боту, як і в багатьох областях, так чи інакше пов'язаних з інтернетом і інформацією, найчастіше доводиться обробляти величезні обсяги даних. Йдеться про такі масштаби, які виділити вручну не просто складно - неможливо: іноді буває потрібно зібрати тисячі, а то й мільйони записів. У таких випадках для збору інформації зазвичай задіюють спеціальне програмне забезпечення, яке називається парсером. Що це за

програма, залежить від області: іноді парсер пишеться самостійно, але в дрібних компаніях частіше застосовують готові рішення.

1. Програма сканує дані, що надходять на вхід, будь то текст, веб-сторінка або інший набір інформації, і виокремлює з них деякі елементи.

2. Що саме буде виділяти парсер з масиву даних - залежить від конкретного завдання. Зазвичай програми можна налаштовувати таким чином, щоб отримувати потрібні результати.

3. Правила пошуку найчастіше задаються регулярними виразами - рядками, складеними за певними правилами і дають програмі пояснення, що і як шукати.

4. На основі зібраної інформації формується звіт або таблиця, в якій відображені всі отримані результати.

Етапи роботи парсера умовно можна розділити на три процеса: сканування масиву інформації, виділення з нього потрібних даних в залежності від заданого правила, складання звіту про знайдені елементах.

Саме в роботі буде використовуватись модуль argparse

Модуль argparse дозволяє легко писати зручні інтерфейси командного рядка. Програма визначає, які аргументи вона вимагає, і argparse зрозуміє, як проаналізувати їх із sys.argv. Модуль argparse також автоматично генерує повідомлення про довідку та використання та видає помилки, коли користувачі надають програмі недійсні аргументи.

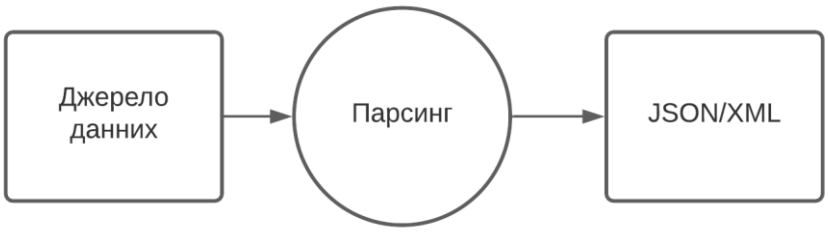


Рисунок 1.5 Схема роботи парсера

Парсери , як я вже згадував вище, існує для автоматизації збору інформації. У мессенджере збирають посилання на профілі, групи та канали. Цілі бувають різними: організація розсилок, спам, інформування клієнтів, інвайтинг

(запрошення учасників за логіном, без попереднього попередження. За такі дії блокують аккаунт).

Розсилки теж не приєднуються: будь-хто може поставити на них позначку "Спам" та модератори Telegram зреагують і заблокють аккаунт, що массово росповсюджує однакові повідомлення багатьом адресатам.

- тимчасове блокування, без можливості відправити повідомлення;
- постійне блокування можливостей входу в поточний акаунт.

Розвиток Телеграм багато в чому визначається наявністю великого числа ботів - невеликих сервісних програм-роботів. Їх може створити кожен користувач, знайомий з програмуванням на середньому рівні. Telegram API Bot - це програмний інтерфейс, що дозволяє програмувати власного бота.

В Бот Телеграм API всі елементи управління є об'єктами, які представлені в JSON, тобто у вигляді рядка, заданої за певними правилами. Це дозволяє проводити обмін даними по мережі максимально швидко і найменш ресурсно затратно, так як передається не програмний код, а набір пар «ключ: значення» в текстовому вигляді. У таблиці наведено всі типи API. Велика частина об'єктів призначена для створення команд бота. Ключі дадуть більш розширене уявлення про можливості об'єкта.

API включає в себе об'єкти і команди, призначені для установки поведінки бота Telegram. Використовуючи інтерфейс, ви можете створювати власні програмні коди, які при запуску в Telegram починають працювати як боти.

Всі методи (а їх досить багато) діляться на групи:

1. Отримання оновлень і інформації.
2. Робота в чаті.
3. Відправка різних елементів.
4. Робота зі стікерами.
5. Оновлення повідомлень.
6. Режим inline.
7. Платіжний функціонал.
8. Для ігор.

Bot API представляє собою HTTP-інтерфейс для роботи з ботами вTelegram. Кожен бот - це спеціальний аккаунт, створений для автоматичного оброблення та відправлення повідомлень. існує двапротилежних за логікою способу отримання оновлень від бота:

- Long pulling - додаток автоматично опитує сервера Telegram на наявність будь-яких оновлень для бота. За замовчуванням 100мс;
- Webhook - сервера Telegram самі сповіщають додаток насервері як тільки з'являться будь-які оновлення.

Принцип роботи взаємодії чат-бота і користувача зображений на рисунку 1.2 в вигляді високорівневої схеми.

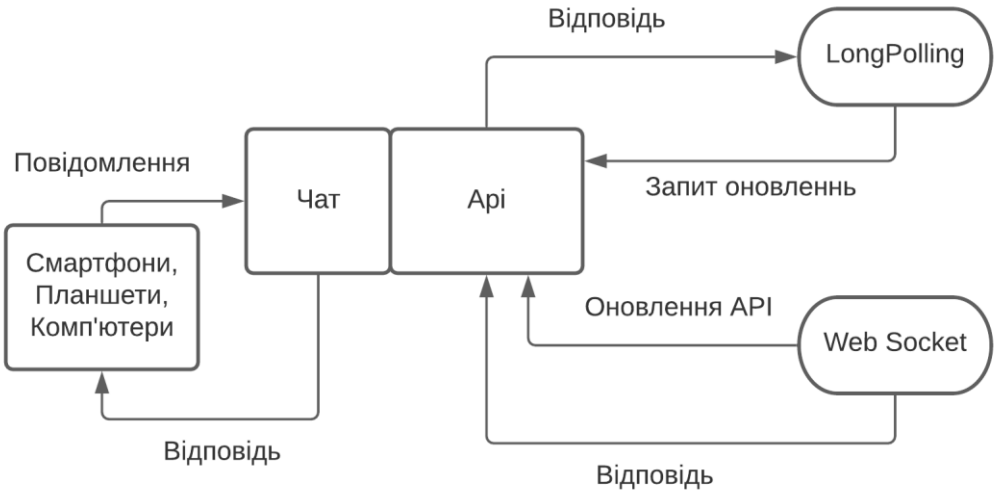


Рисунок 1.6. Принцип роботи чат-бота на платформі Telegram

Telegram API підтримується безліччю мов програмування. Це дає можливість вибору творцеві.

Можна використовувати Node.js Telegram Bot API. Тут необхідне знання не тільки мови, а й уміння поводитися з цим фреймворком, що перетворив клієнтський мову в повноцінний серверний інтерфейс.

Одним з найпопулярніших для написання ботів з використанням Telegram Bot API є PHP. Ця мова спочатку був призначений для створення серверних web-

додатків. Він відрізняється простотою, логічністю і специализированностью саме для web-середовища.

Часто використовується Telegram Bot API в Python. Ця мова відрізняється мінімалізмом і досить простий у вивченні. Він дуже популярний через свою продуктивності.

Класикою є застосування Telegram Bot API в C ++. Мова не можна назвати простим, але він є базою, на якій були створені всі інші перераховані вище ЯП. Відповідно в ньому не закладено певна спеціалізація. Інструменти дозволяють створювати будь-які додатки.

Тобто використовувати з Telegram API можна використати мови програмування: Python, JavaScript (Node.js Telegram API), PHP, C++

Крім об'єктів API має набір методів, які дозволяють відправляти повідомлення, файл, фото стікери, редагувати і багато іншого. Всі ці команди можна знайти в описі API на офіційному сайті.

Для створення в Telegram існує спеціальний сервіс @Botfather. В ньому є набір команд, за допомогою яких створюється новий бот. Назва бота обов'язково закінчується на «bot». Після того як буде закріплений токен (ідентифікатор), новий бот створений. Авторизація здійснюється через токен. Щоб запустити програму в Телеграм, потрібно запустить преопределену команду «/start». Також для кожного робота зарезервовані команди «/settings» і «/help».

Всі запити мають вигляд: https://api.telegram.org/bot<token>/КОМАНДА

Всього існує 4 способи подачі запиту:

1. Запит в URL
2. application / x-www-form-urlencoded
3. application / json (не підходить для завантаження файлів)
4. multipart / form-data (для завантаження файлів)
5. Доступні як GET, так і POST запити.

Вхідні оновлення будуть зберігатися на сервері до тих пір, поки їх необроблять, але не довше 24 годин. Незалежно від способу отримання оновлень, у відповідь відправляється об'єкт Update, серіалізовані в JSON.

Всі запити до Telegram Bot API повинні здійснюватися через HTTPS внаступному вигляді: https://api.telegram.org/bot<token>/НАЗВА_МЕТОДА.

Найпростіший спосіб спробувати команди API - адресний рядок в браузері. Результатом буде JSON рядок. Приклад Telegram API

Таблиця 1.1

Об’єкти Telegram API

Назва	Опис	Ключі
User	Користувач сервісу Telegram	Id, first_name, last_name, username
Chat	Чат	Id, type, title, username, first_name last_name, all_members_are_administrators
Message	Повідомлення	message_id, from date, chat, forward_from, forward_date, reply_to_message, text, entities, audio, document, photo, sticker, video, voice, caption, contact, location, venue, new_chat_member, left_chat_member, new_chat_title, new_chat_photo, delete_chat_photo, group_chat_created, supergroup_chat_created, channel_chat_created migrate_to_chat_id, migrate_from_chat_id pinned_message
MessageEntity	Окрема частина повідолення, посилання, хештег	Type, length, url, offset
PhotoSize	Зображення заданого розміру. Превью до файлу або стікера	file_id, width, height, file_size

Продовження Таблиці 1.1

Audio	Аудіозапис	file_id, duration, performer, title, mime_type file_size
Document	Будь який файл, що не є зображенням, стікером, аудіо, або відеозаписом.	file_id, thumb, file_name, mime_type, file_size
Sticker	Стікер – наліпка, аналог смайлів	file_id, width, height, thumb, file_size
Video	Відеозапис	file_id, width, height, duration, thumb, mime_type, file_size
Voice	Голосове повідомлення	file_id, duration, mime_type, file_size
Contact	Контакт	phone_number, first_name, last_name, user_id
Location	Точка на карті	Longitude, latitude
Venue	Обьєкт на карті	Location, title, address, foursquare_id
UserProfilePhotos	Фото профіля користувача	total_count, photos
File	Готовий до завантаження файл	file_id, file_size, file_path
ReplyKeyboardMarkup	Клавіатура з можливістю відповіді	Keyboard, resize_keyboard, one_time_keyboard Selective,
KeyboardButton	Кнопка на клавіатурі для відповіді	Text, request_contact, request_location

Продовження Таблиці 1.1

ReplyKeyboardHide	Заміняє стандартну клавіатуру бота, на стандартну клавіатуру Telegram	hide_keyboard, selective
InlineKeyboardButton	Кнопка на вбудованій клавіатурі.	Text, url, callback_data, switch_inline_query, switch_inline_query_current_chat, callback_game
CallbackQuery	Запит для зворотнього зв'язку	Id, from, message, inline_message_id, data
ForceReply	Ємулює дії користувача, вибір користувача, та кнопку «Відповіти»	force_reply, selective
ResponseParameters	Повідомляє про статус виконная запиту	migrate_to_chat_id, retry_after
InlineKeyboardMarkup	Вбудована клавіатура під повідомленням	inline_keyboard

Обмін повідомленнями відбувається у вигляді запитів. У наступній таблиці наведено приклади деяких з них.

Таблиця 1.2

Запити Telegram API

Метод	Дія
getMe	Дозволяє отримати інформацію про користувача
sendMessage	Відправляє повідомлення
sendPhoto	Відправляє фото
sendAudio	Відправляє аудіозапис
sendDocument	Відправляє документ
sendVideo	Відпраляє відеозапис
sendContact	Відпраляє контакт
getUpdates	Отримує оновлення з чату

1.4 Мова програмування Python

Python це мова програмування загального призначення, націлений впершу чергу на підвищення продуктивності самого програміста,ніж коду, який він пише. Говорячи простою мовою, на Python можнанаписати практично що завгодно (веб та десктоп додатки, ігри,скрипти по автоматизації, комплексні системи розрахунку, системиуправління життєзабезпеченням і багато іншого) без відчутних проблем.Більш того, поріг входження низький, а код багато в чому лаконічний і зрозумілий навіть тому, хто ніколи на ньому не писав. За рахунок простоти коду,подальший супровід програм, написаних на Python, стає легше і приємніше в порівнянні з Java або C ++. А з точки зору бізнесу це показує за собою скорочення витрат і збільшення продуктивностіпраці співробітників. Безсумнівним достоїнством є те, що інтерпретатор Python реалізований практично на всіх платформах і операційних системах. Першим таким мовою був C.

Важлива риса - розширюваність мови, цього надаєтьсявелике значення і, як пише сам автор Гвідо ван Россум, мова була задумана саме як розширювана. Це

означає, що є можливість вдосконалення мови усіма зацікавленими програмістами. Інтерпретатор написаний на C і вихідний код доступний для будь-яких маніпуляцій. У разі необхідності, можна вставити його в свою програму і використовувати як вбудовану оболонку. Або ж, написавши на Свої доповнення до Python і скомпілювавши програму, отримати "Розширений" інтерпретатор з новими можливостями. Наступна перевага - наявність великого числа підключаються до програм модулів, що забезпечують різні додаткові можливості. Такі модулі пишуться на C і на самому Python і можуть бути розроблені усіма досить кваліфікованими програмістами. В приклад можна навести такі модулі:

Numerical Python - розширені математичні можливості,такі як маніпуляції з цілими векторами і матрицями;

Tkinter - побудова додатків з використанням графічногопризначеного для користувача інтерфейсу (GUI).

OpenGL - використання великої бібліотеки графічногомоделювання дво- і тривимірних об'єктів Open Graphics Library фірми Silicon Graphics Inc. Даний стандарт підтримується, в тому числі, в такихпоширених операційних системах як Microsoft Windows 95 OSR 2,98 і Windows NT 4.0. Єдиним недоліком, поміченим автором, є порівняно невисока швидкість виконання Python-програми, що обумовлено її інтерпретуванням. Однак, це з лишком окупається достоїнствами мови при написанні програм не дуже критичних до швидкості виконання. З використаних модулів можна виділити: TeleBot який більш детально буде описаний нижче.

Цей модуль є оболонкою над запитами до Telegram Bot API, використовується для спрощення та мінімізації написаного коду.

Всі типи визначені в types.py. Всі вони повністю відповідаютьвизначення типів API Telegram, за винятком from поля Message, яке перейменовано в from_user (оскільки from це зарезервованийтокен Python). Таким чином, до таких атрибутів як message_id можна звертатися безпосередньо, наприклад: message.message_id. Варто звернути увагу, що атрибут message.chat може належати, як певногокористувачеві, так і до групового чату.

Об'єкт Message також має content_type атрибут, який визначає тип повідомлення. Атрибут content_type може бути одним знаступних рядків: text, audio, document, photo, sticker, video і т. д. Можна використовувати декілька типів в одній функції. Всі методи API розташовані в класі TeleBot. Вони перейменовані, щоб слідувати загальним угодами про імена Python. наприклад: sendMessage перейменований send_message, editMessageTextвedit_message_text. Оброблювач повідомлень - це функція, прикрашена декоратор мпримірника TeleBot. Обробники повідомлень складаються з одного абодекількох фільтрів. Кожен фільтр повертає True або False дляпевного повідомлення, і якщо повертається True, то обробник отримуєдозвіл на обробку повідомлення.

Висновки до розділу 1

Проаналізовані джерела спаму, його види, та шляхи росповсюдження.

Досліджено Telegram API та способи написання чатботів, та описані мови програмування на яких можлива реалізація таких ботів

Було виокремлено як саме бот може аналізувати текст, за допомогою парсера.

Описано об’єкти ключі, та запити до Telegram API, а також способи їх взаємодії з повідомленнями.

Підкреслено, як саме спам доходить до кінцевого пункту відправки, а також його різновиди.

В якості мови для написання сервера був обраний Python

2 ДОСЛІДЖЕННЯ ПРОБЛЕМ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ В СЕРВІСАХ ДЛЯ МИТТЄВОГО ОБМІНУ ПОВІДОМЛЕННЯМИ

Ідентифікація користувача Telegram та WhatsApp пов’язана з номером телефону користувача. Користувач вводить номер телефону до якого привязаний аккаунт. Потім сервер WhatsApp, або Telegram надсилає SMS із кодом підтвердження, користувач подає код із отриманого текстового повідомлення та WhatsApp створює, або входить в свій обліковий запис. Якщо користувач вже має авторизацію на пристрої цей код він може отримати в спеціальний офіційний чат. Також в цей чат приходять інші системні повідомленя, та списки оновлень від сервісів, приклад таких повідомлень наведено на рисунку 2.1.

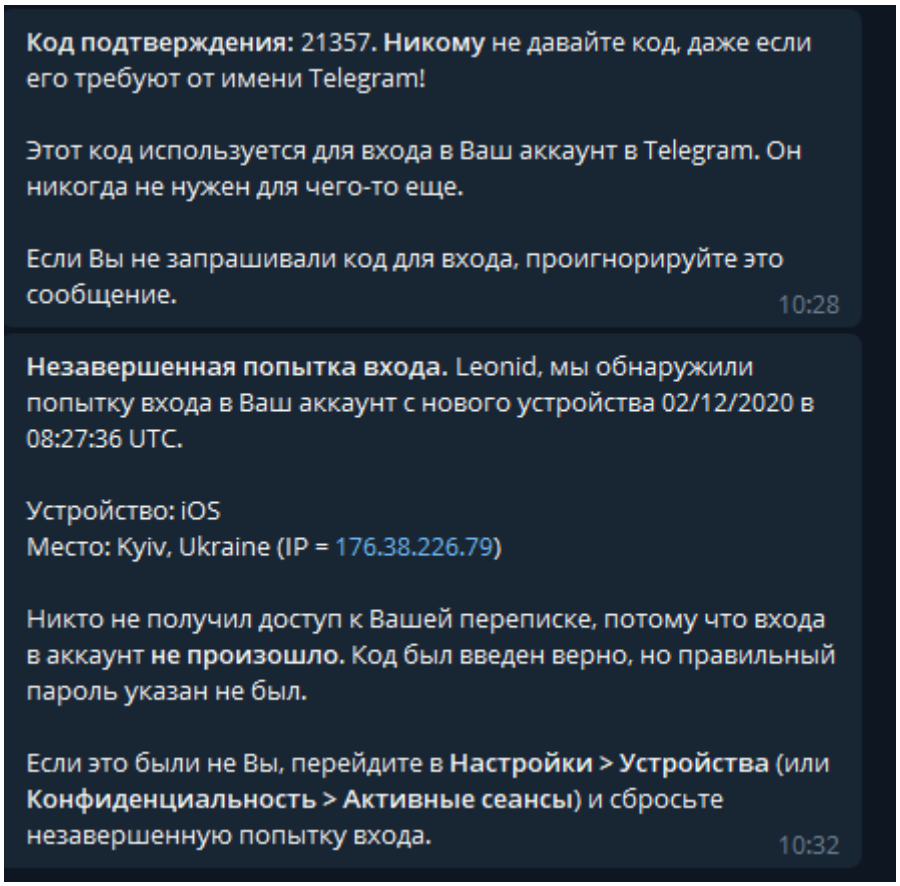


Рис.2.1. Приклад системних повідомлень

З вище написаного виходить, що спосіб взлому вашогу аккаунти тільки 1, це SMS-спуфінг.

Уразливість полягає в способі авторизації користувача. Для даної процедури використовується реальний номер телефону, на який відправляється СМС-код для підтвердження входу в аккаунт. В основі подібного методу передачі даних лежить технологія SS7 (Signaling System # 7), яка розроблялася 40 років тому і має слабкі параметрами безпеки за сучасними мірками. Теоретично зловмисники можуть перехопити СМС з кодом і зламати акаунт. А оскільки в звичайному режимі Telegram зберігає всі повідомлення на своїх серверах, хакери можуть отримати доступ до всієї листуванні конкретного користувача.

SMS-спуфінг - це технологія, яка використовує службу коротких повідомлень (SMS), доступну для більшості мобільних телефонів та персональних цифрових помічників, щоб встановити, від кого надходить повідомлення, замінивши номер мобільного телефону, що походить (Sender ID), на буквено-цифровий текст. Підробка має як законне використання (встановлення назви компанії, від якої надсилається повідомлення, встановлення власного номера мобільного телефону або назви продукту), так і нелегітимне використання (наприклад, видавання себе за іншу особу, компанію, продукт). Це також може надсилати "таємничі" повідомлення, схожі на те, що вони надходять із законних номерів або контактів.

Найновіші у своїй захищеності, хто користується популярною сьогодні послугою - двофакторна авторизація за допомогою SMS. Такі послуги пропонуються банками, відомими мессенджерами соціальними мережами, поштовими системами. Та великою кількістю інших інтернет-ресурсів, вимагаючих реєстрацію. Не просто так цей метод вважається самим безпечним., Ваш мобільний телефон завжди при вас, як гаманець та інші особисті речі, для яких ви пристально стежте. Тому тому він є підтверджувальним фактором аутентифікації (застосовується лише після введення основного пароля). Але такі SMS можна перехватити за допомогою знання алгоритму SS7.

Протокол SS7, також відомий як Сигналізаційна система № 7, відноситься до мереж передачі даних і до ряду технічних протоколів або правил, які регулюють обмін данними за ним. Він був розроблений у 1970-х роках для

відслідування та підключення визовів у різних мережах операторів зв'язку, а тепер він зазвичай використовується для розсилки біллінгових сотових зв'язків та відправлення текстових повідомлень у доповнення до маршрутизації мобільних та стаціонарних викликів між операторами та регіональними комунікаційними центрами.

Також в розглянутих сервісів є можливість додати двофактурну автинфікацию, тобто не тільки за рахунок тимчасового коду, а і за допомогою паролю. Що є додатковою мірую безпеки.

Зловмисник підключається до сигнальної мережі SS7 і відправляє службову команду Send Routing Info для SM (SRI4SM) в мережевому каналі, вказуючи номер телефону, атакуваного абонента за вашими параметрами. Домашня абонентська сеть надсилає у відповідь наступну технічну інформацію: IMSI (International Mobile Subscriber Identity) та адресу MSC, за котрою в даний час надає послуги підписчика.

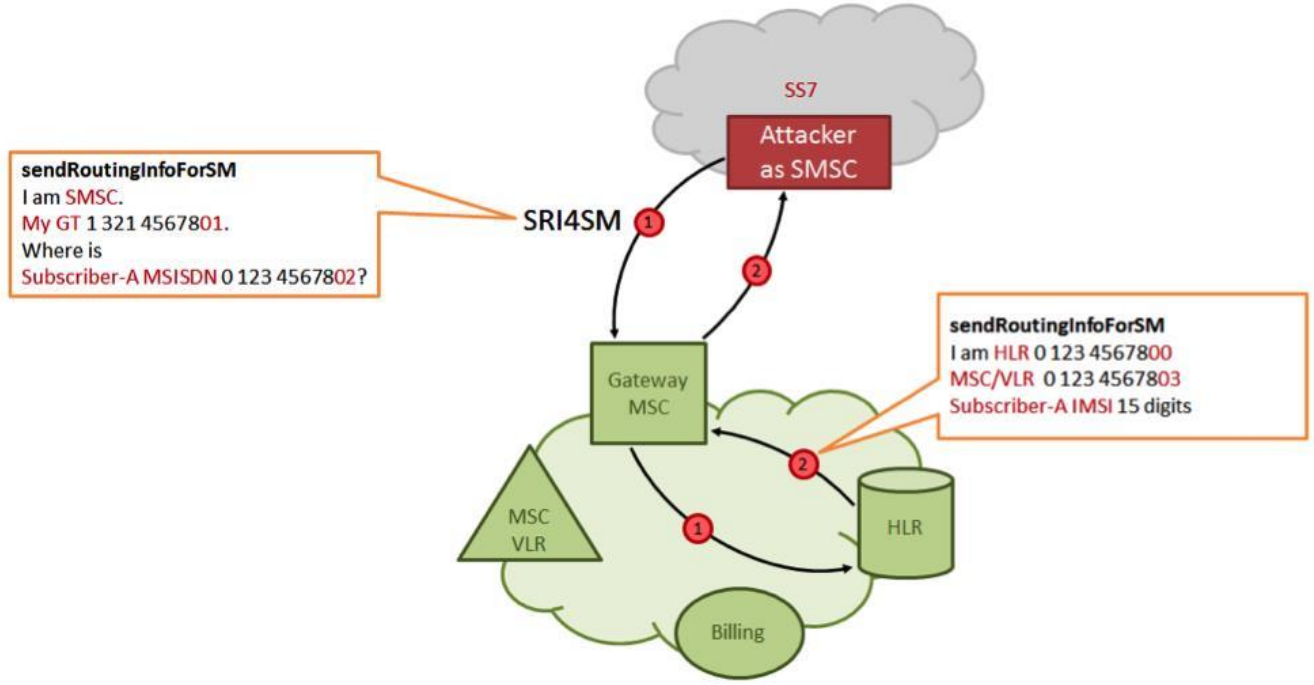


Рис.2.2

Після цього зловмисник змінює адресу білінгової системи в профілі передплатника на адресу своєї власної псевдобіллінговой системи (наприклад, повідомляє, що абонент прилетів на відпочинок і в роумінгу зареєструвався на

новій білінгової системи). Як відомо, ніяку перевірку така процедура не проходить. Далі атакуючий вводить оновлений профіль в базу даних VLR через повідомлення «Insert Subscriber Data» (ISD).

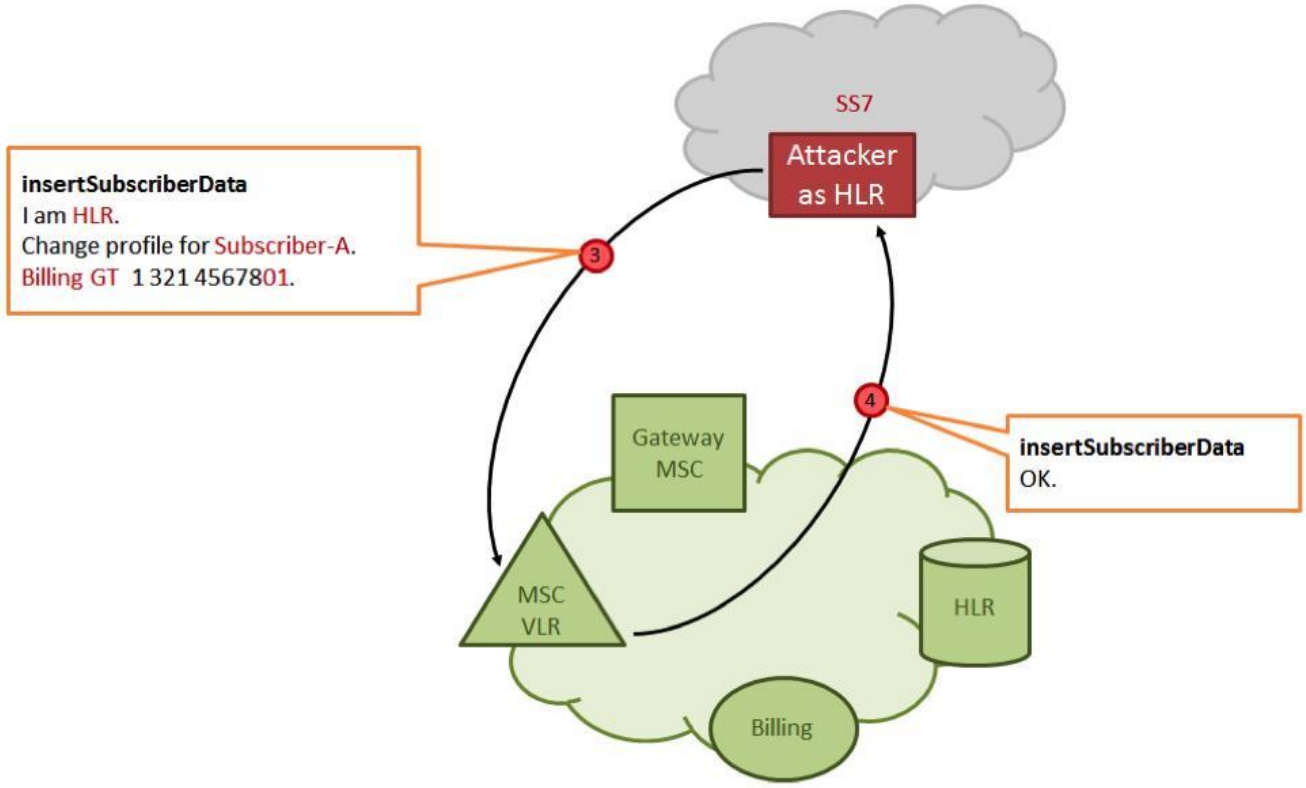


Рис.2.3

Коли атакується абонент здійснює вихідний дзвінок, його комутатор звертається до системи зловмисника замість фактичної білінгової системи. Система зловмисника відправляє комутатора команду, що дозволяє перенаправити виклик третій стороні, контрольованої зловмисником. Приклад на рисунку 2.4

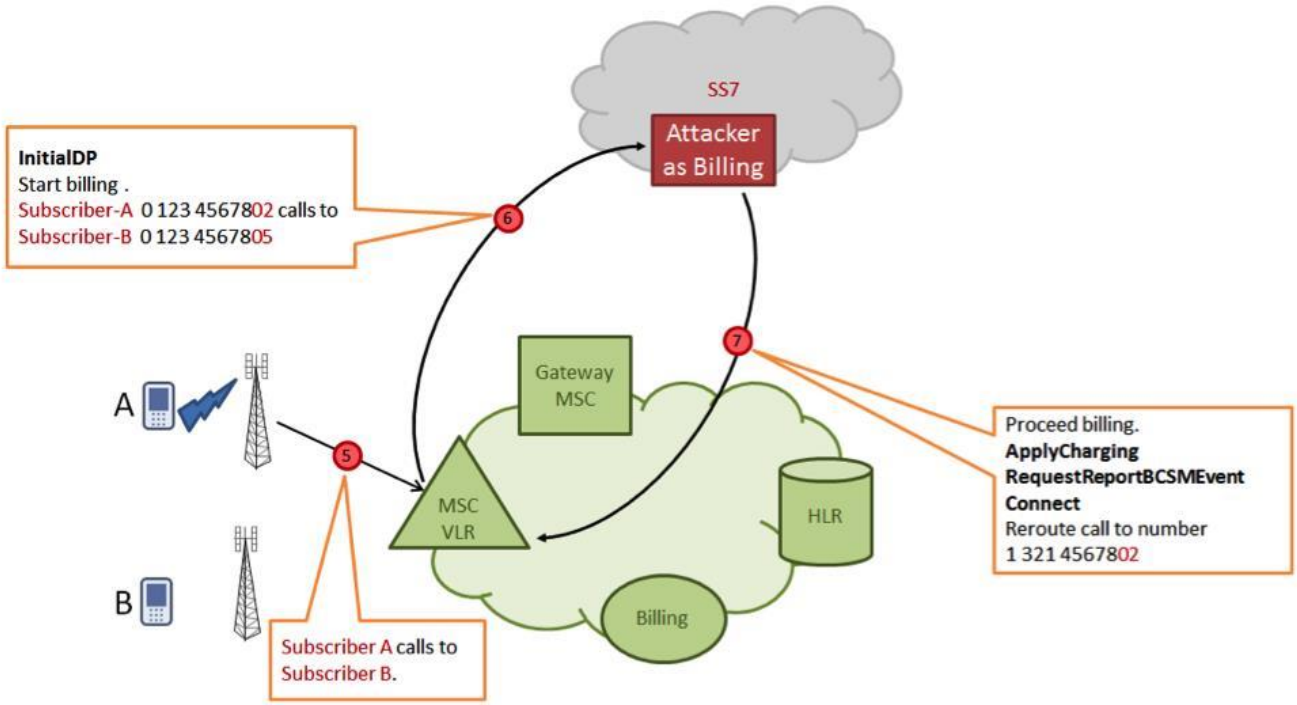


Рис.2.4

У сторонньому місці встановлюється конференц-зв'язок з трьома передплатниками, дві з них є реальними (абонент а і викликається b), а третій вводиться зловмисником незаконно і здатний прослуховувати і записувати розмову. Відповідним чином отримуємо і SMS атакується. Маючи доступ до псевдобіллінговой системі, на яку вже зареєструвався наш абонент, можна отримати будь-яку інформацію, яка приходить або йде з його телефону. Приклад на рисунку 2.5

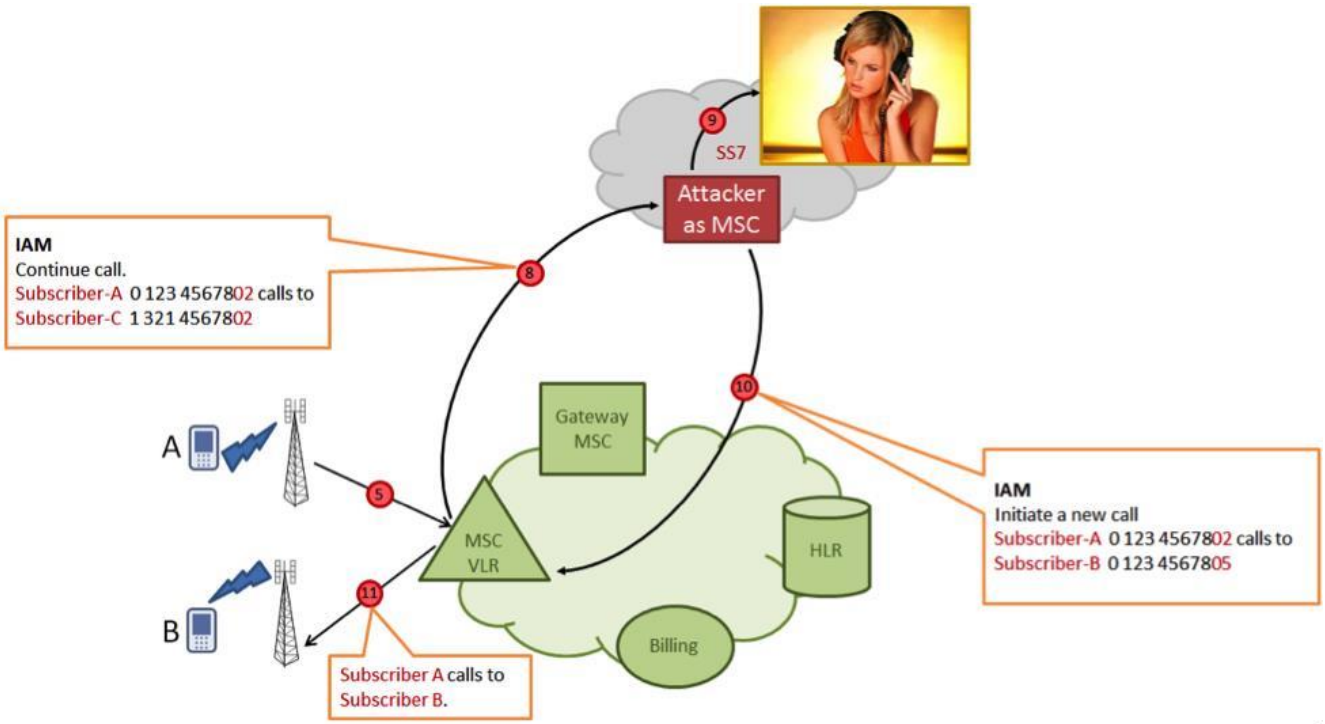


Рис.2.5

На завершення хотілося б сказати пару слів про те, як оператори стільникового зв'язку дивляться на безпеку SS7. По-перше, вони вважають (і небезпідставно), що неможливо отримати абсолютно нелегальний доступ до мережі SS7. По-друге, між усіма операторами є угоди на пропуск трафіку, і навіть якщо раптом з боку якогось оператора будуть помічені зловмисні дії, то не важко буде провести блокування за адресою джерела. По-третє, не так багато відомо випадків зловживання засобами SS7.

Всі три твердження абсолютно справедливі. Але отримати підключення до SS7 можливо, наприклад, під прикриттям провайдера VAS-послуг. Швидше за все, це не вийде зробити в Україні, проте є країни з більш лояльним законодавством у сфері телекомунікацій. Перш ніж припинити шкідливі дії, їх треба помітити. Вкрай мале число операторів проводить постійний моніторинг мереж SS7 на предмет вторгнень і атак. І навіть якщо атака через мережу SS7 буде помічена і припинена, ніхто не зможе сказати, коли вона почалася і як довго тривала.

Розглянему іншу вразливість сервісів, атака, що використовує уяву «обовьязкову» зміну ключів у WhatsApp. У чатах WhatsApp у випадку, коли

отримує клієнт не отримує зашифровані повідомлення тривалий час знаходиться в статусі «не в мережі» або вимкнув пристрій, генерується новий ключ.

Повідомлення пересилається на сервер WhatsApp, де буде видаватися отримувачу коли він знову зьявиться в мережі. Тобто зьявляється посередник з ще одним ключовим шифруванням, до якого може бути отримати доступ до розробників. За затвердженням , адміністрація мессенджера WhatsApp таким чином може запропонувати отримати повідомлення іншим особам.

Проблему вирішує спілкування в секретних чатах. В цьому випадку прочитати переписку можна тільки за допомогою реальної крадіжки телефону, так як повідомлення не зберігаються на сервері, а передаються виключно між двома пристроями.

Висновки до розділу 2

Проаналізовано проблеми забезпечення безпеки в сервісах для миттєвого обміну повідомленнями, а конкретно спосіб перехвату SMS повідомленнь SS7

Досліджено гіпотетичний спосіб, яким можна через вразливості протоколів SS7, можна заволодіти аккаунтом в Telegram та WhatsApp

Виокремлено, що спосіб працює лише з аккунтами де відсутня двофакторна аунтифікація.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ….

3.1 СЕРЕДОВИЩЕ РОЗРОБКИ VISUAL STUDIO CODE

Visual studio code - це легкий, але потужний редактор вихідного коду,який сумісний з windows, macos і linux.ide підтримує велику кількість популярних мовпрограмування. Підтримувані розширення і середовища виконанняпредставлені нижче:

- javascript;
- typescript;
- node.js;
- c ++;
- c #;
- java;
- python;
- php;
- go;
- .net;
- unity.

Для зручності розробки програмних продуктів VS CODE включає всебе вбудований відладчик, інструменти для роботи з GIT-репозиторіями, підсвічування синтаксису, засоби для рефакторинга I INTELLISENCE (технологіяавтодоповнення, яка пропонує команду за першими літерами).IDE VS CODE була анонсована 29 квітня 2015 року компанієюmicrosoft в рамках конференції BLINK.

18 листопада 2015 року редактор буввипущений в реліз під ліцензії Массачусетського Технологічного Інституту.VISUAL STUDIO CODE заснований на Electron - фреймворк , що дозволяє використовуючи node.js розробляти настільні додатки, що працюють на ядрі BLINK. Незважаючи на те, що редактор заснований на Electron , він невикористовує редактор Atom .

Замість нього був реалізований веб-редактор Monaco,розроблений для VISUAL STUDIO ONLINE.

3.2 HTTP-ЗАПИТИ І REST API.

Rest - це стиль архітектури програмного забезпечення для побудовиро зподілених масштабованих веб-сервісів, які використовують http запити Протокол передачі гіпертексту (http), один з протоколів стеку tcp / ip (transmission control protocol / internet protocol), був спочаткурозроблений для публікації і отримання html (hyper text markuplanguage) сторінок і тепер використовується для розподіленихінформаційних систем.

Http використовується у всесвітній павутині дляпередачі даних і є одним з найбільш широко застосовуванихприкладних протоколів.http визначає протокол типу запит / відповідь. Коли клієнт,наприклад веб браузер, посилає повідомлення запиту на сервер, http протокол визначає типи повідомлень, які використовуються клієнтом для запитувеб сторінки, а також типи повідомлень, що застосовуються сервером для відповіді.трьома распространѐннимі типами повідомлень є get, post і put.на малюнку 2.1 наведено приклад get-запиту.малюнок 2.1 - http протокол з використанням get post і put використовуються, щоб посилати повідомлення, якізавантажують дані на веб сервер. Наприклад, коли користувач вводитьдані в форму, вбудовану в веб сторінку, post включає дані вповідомлення, що посилається на сервер.put завантажує ресурси або контент на веб сервер.

будучи надзвичайно гнучким, http не є безпечним протоколом.

HTTPS не є окремим протоколом. Це звичайний HTTP, що працює через шифровані транспортні механізми SSL і TLS. Він забезпечує захист від атак, заснованих на прослуховуванні мережевого з'єднання - від сніфферскіх атак і атак типу man-in-the-middle, за умови, що будуть використовуватися шифрувальні засоби і сертифікат сервера перевірений і йому довіряють.

За замовчуванням HTTPS URL використовує 443 TCP-порт (для незахищеного HTTP - 80). Щоб підготувати веб-сервер для обробки https-з'єднань, адміністратор повинен отримати і встановити в систему сертифікат відкритого і закритого ключа для цього веб-сервера. У TLS використовується як асиметрична схема шифрування (для вироблення загального секретного ключа), так і симетрична (для обміну даними, зашифрованими загальним ключем). Сертифікат відкритого ключа підтверджує приналежність даного відкритого ключа власнику сайту. Сертифікат відкритого ключа та сам відкритий ключ надсилаються клієнту при встановленні з'єднання; закритий ключ використовується для розшифровки повідомлень від клієнта.

Існує можливість створити такий сертифікат, не звертаючись до центру сертифікації. Підписуються такі сертифікати цим же сертифікатом і називаються самоподпісанного (self-signed). Без перевірки сертифіката якимось іншим способом (наприклад, дзвінок власнику і перевірка контрольної суми сертифіката) таке використання HTTPS схильне атаці посередника.

Ця система також може використовуватися для аутентифікації клієнта, щоб забезпечити доступ до сервера тільки авторизованим користувачам. Для цього адміністратор зазвичай створює сертифікати для кожного користувача і завантажує їх в браузер кожного користувача. Також будуть прийматися всі сертифікати, підписані організаціями, яким довіряє сервер. Такий сертифікат зазвичай містить ім'я та адресу електронної пошти авторизованого користувача, які перевіряються при кожному з'єднанні, щоб перевірити особу користувача без введення пароля.

У HTTPS для шифрування використовується довжина ключа 40, 56, 128 або 256 біт. Деякі старі версії браузерів використовують довжину ключа 40 біт (приклад тому - IE версій до 4.0), що пов'язано з експортними обмеженнями в США. Довжина ключа 40 біт не є надійною. Багато сучасних сайти вимагають використання нових версій браузерів, що

підтримують шифрування з довжиною ключа 128 біт, з метою забезпечити достатній рівень безпеки. Шифрування з довжиною ключа 128 біт значно ускладнює підбір паролів і доступ до особистої інформації.

Традиційно на одному IP-адресу може працювати тільки один HTTPS-сайт. Для роботи декількох HTTPS-сайтів з різними сертифікатами буде застосований ефект розширення TLS під назвою Server Name Indication (SNI).

Повідомлення post завантажують інформацію на сервер у виглядізвичайного тексту, який може бути перехоплений і прочитаний. Аналогічно,відповіді сервера, як правило, html сторінки також не зашифровані.для безпечної комунікації через інтернет використовується безпечний http протокол (https), щоб отримувати доступ або публікувати інформацію на веб сервері. Https може використовувати аутентифікацію і шифрування, щоб убезпечити дані, коли вони переміщаються між клієнтом і сервером. Https визначає додаткові правила для проходження даних між прикладним і транспортним рівнями.

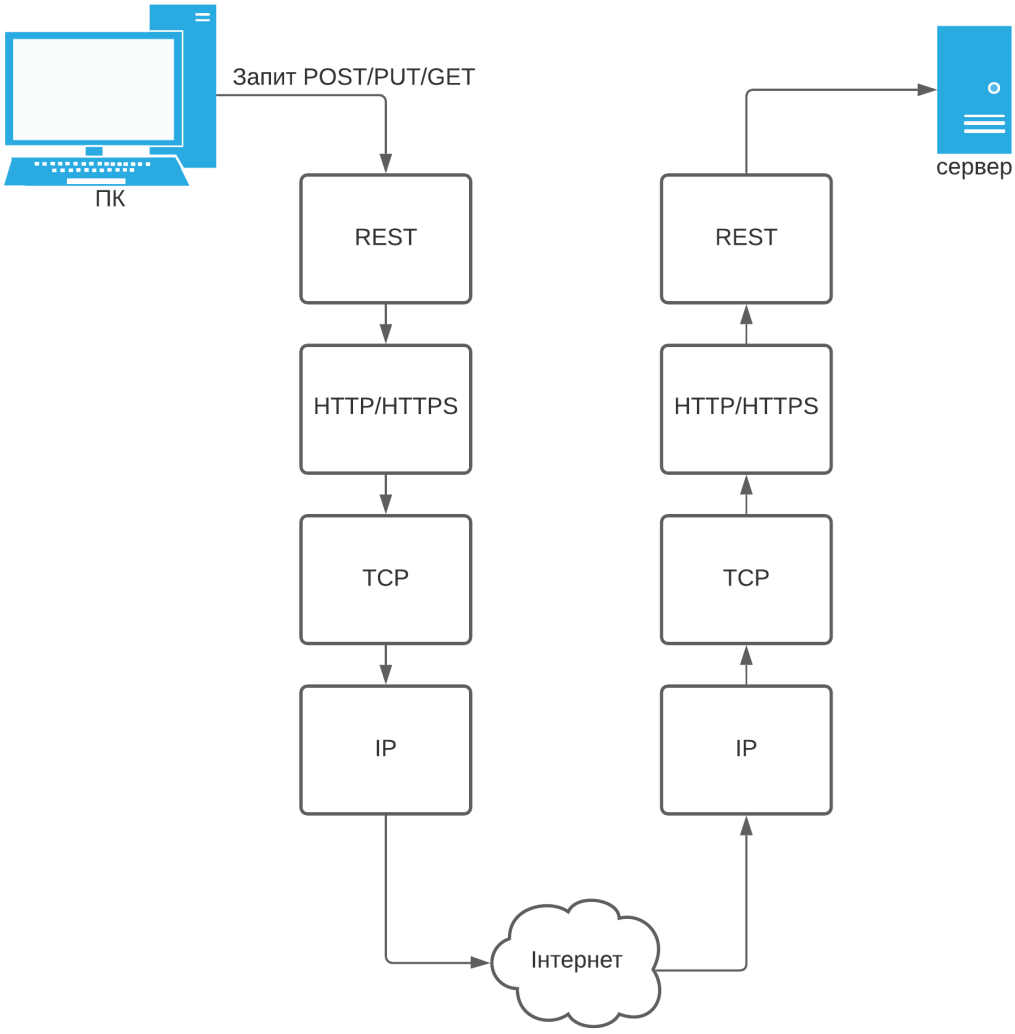

Запит POST/PUT/GET
ПК
REST
HTTP/HTTPS
TCP
IP
Інтернет
IP
TCP
HTTP/HTTPS
REST
сервер

Рисунок 3.1

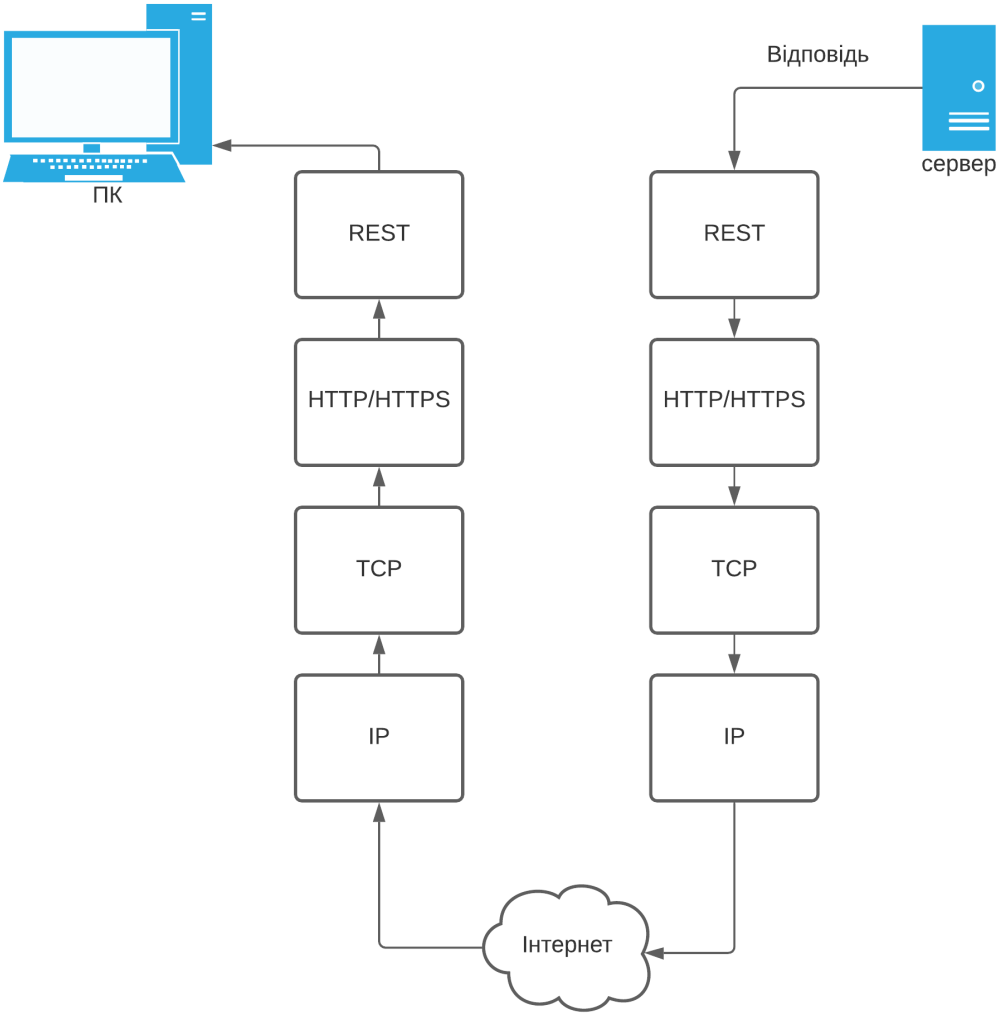


Рисунок 3.2

3.3 ФОРМАТ JSON

json (javascript object notation) - простий формат обміну даними,зручний для читання і написання як людиною, так і комп'ютером. Вінзаснований на підмножині мови програмування javascript,визначеного в стандарті ecma-262 3rd edition - december 1999. Json -текстовий формат, повністю незалежний від мови реалізації, але вінвикористовує угоди, знайомі програмістам с-подібних мов, такихяк с, с ++, с #, java, javascript, perl, python і багатьох інших. Ці властивостіроблять json ідеальним мовою обміну даними.json заснований на двох структурах даних:•колекція пар ключ / значення. У різних мовах, ця концепціяреалізована як об'єкт, запис, структура, словник, хеш, іменованийсписок або асоціативний

масив.•упорядкований список значень. У більшості мов цереалізовано як масив, вектор, список або послідовність.це універсальні структури даних. Майже всі сучасні мовипрограмування підтримують їх в будь-якій формі. Логічноприпустити,щоформатданих,незалежнийвідмовипрограмування, повинен бути заснований на цих структурах.в нотації json це виглядає так:•об'єкт - невпорядкований набір пар ключ / значення. Об'єктпочинається з відкриває фігурної дужки і закінчується закриваєфігурною дужкою. Кожне ім'я супроводжується двокрапкою, пара ключ / значення розділяються комою (рисунок 2.2).

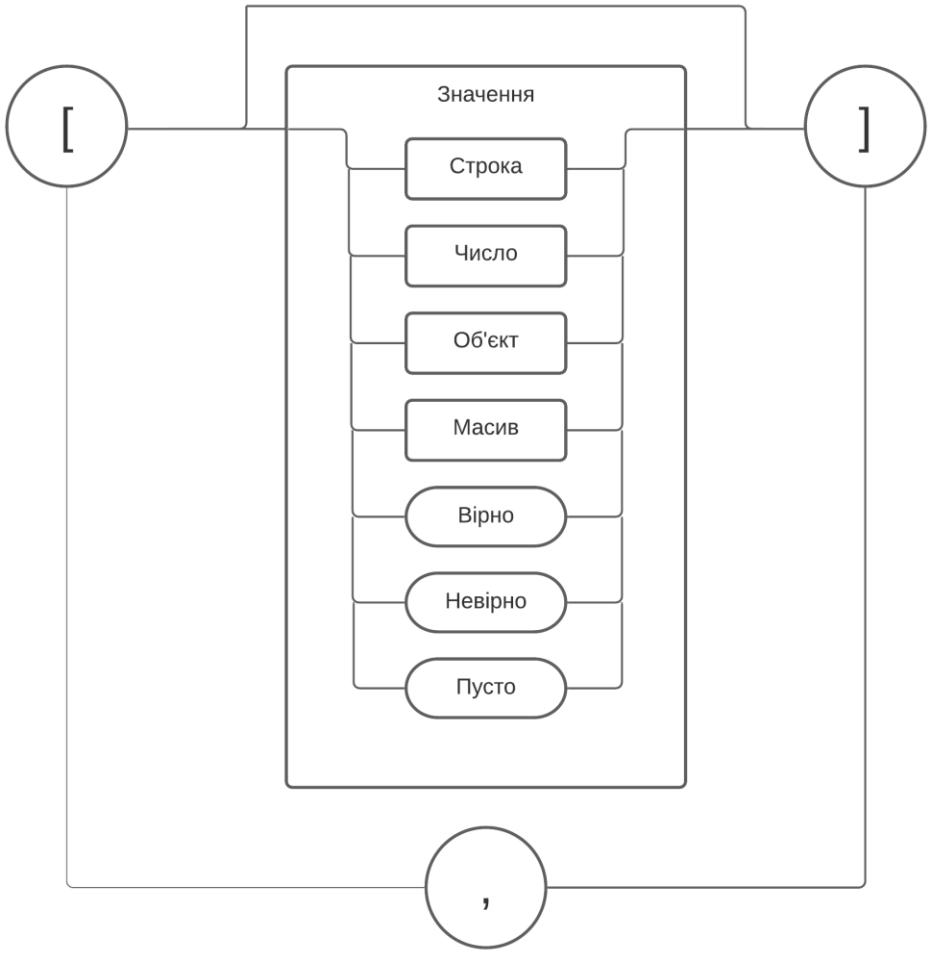


Рисунок 3.3

Також можна порівняти JSON з XML. Таке порівняння зображено на рисунку

XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<person>
 <name>Иван</name>
 <age>37</age>
 <mother>
  <name>Ольга</name>
  <age>58</age>
 </mother>
 <children>
  <child>Маша</child>
  <child>Игорь</child>
  <child>Таня</child>
 </children>
 <married>true</married>
 <dog null="true" />
</person>
```

JSON

```
{
 "person":{
  "name":"Иван",
  "age":37,
  "mother":{
   "name":"Ольга",
   "age":58
  },
  "children":[
   "Маша",
   "Игорь",
   "Таня"
  ],
  "married":true,
  "dog":null
 }
```

Рисунок 3.4

3.4 TELEGRAM BOT API

У документації telegram bot api виділяється два діаметральнопротилежних способу отримання оновлень: за допомогою періодичних запитів або установка вебхуков.

Вхідні оновлення зберігаються до тих пір, поки сервер не обробить його, але не довше 24 годин.

Незалежно від способу отримання оновлень, у відповідь отримуємо об'єкт update, серіалізований в json.

Перший і найбільш простий варіант полягає в періодичному опитуванні серверів telegram на предмет наявності нової інформації.

Все цездійснюється через так званний Long polling, тобто відкривається з'єднання нанетривалий час і всі оновлення тут же відправляються боту.просто, але не дуже надійно. По-перше, сервери telegram періодичнопочинають повертати помилку 504 (gateway timeout), через що деякі боти зупиняються. По-друге, якщо одночасно запущено кілька ботів, ймовірність зіткнутися з помилками зростає. Вебхуки працюють трохи

інакше. Під установкою вебхука мається на увазі, що тепер якщо в чат приходить повідомлення, то telegram сам говорить про це. Відпадає необхідність періодично опитуватисервери, тим самим, зникає причина падінь спамерських пошукових роботів. Однак за цедоводиться платити необхідністю установки повноцінного веб-серверана ту машину, на якій планується запускати спамерських пошукових роботів. Так само для роботитреба мати власний ssl-сертифікат (secure sockets layer), тому що вебхуків telegram працюють тільки по https [12].для роботи з telegram bot api була вивчена документація, в якійописані всі методи і передані параметри, всі відповіді приходять вjson-форматі. В ході написання чат-бота були протестовані івикористані наступні методи і типи

- метод polling() використовується для отримання оновлень через long polling. Відповідь повертається у вигляді масиву об'єктів update.

3.5 BotFather

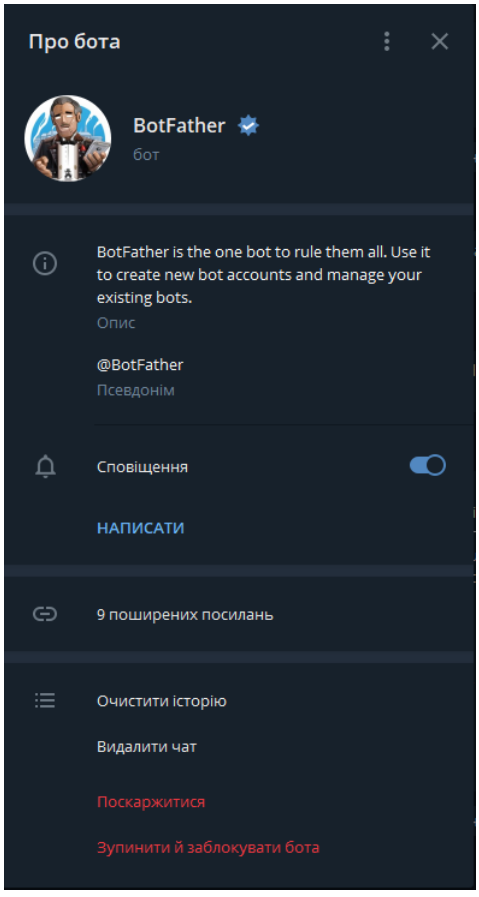


Рисунок 3.4

Взаємодія з BotFather здійснюється за допомогою простих команд. Наприклад, для того, щоб зареєструвати нового бота, досить відправити в чат команду / newbot і слідувати простим інструкціям:

Придумати ім'я бота, яке буде відображатися в чатах і контактах. Надалі його можна буде змінити. Тут все залежить тільки від вашої фантазії і вимог;

Придумати username - це вже складніше: ім'я повинно бути унікальним і закінчуватися на «bot». Допускаються букви латинського алфавіту, цифри і символ підкреслення (приклад - «killimgerbot»). Загальна кількість символів не менше 5 і не більше 32;

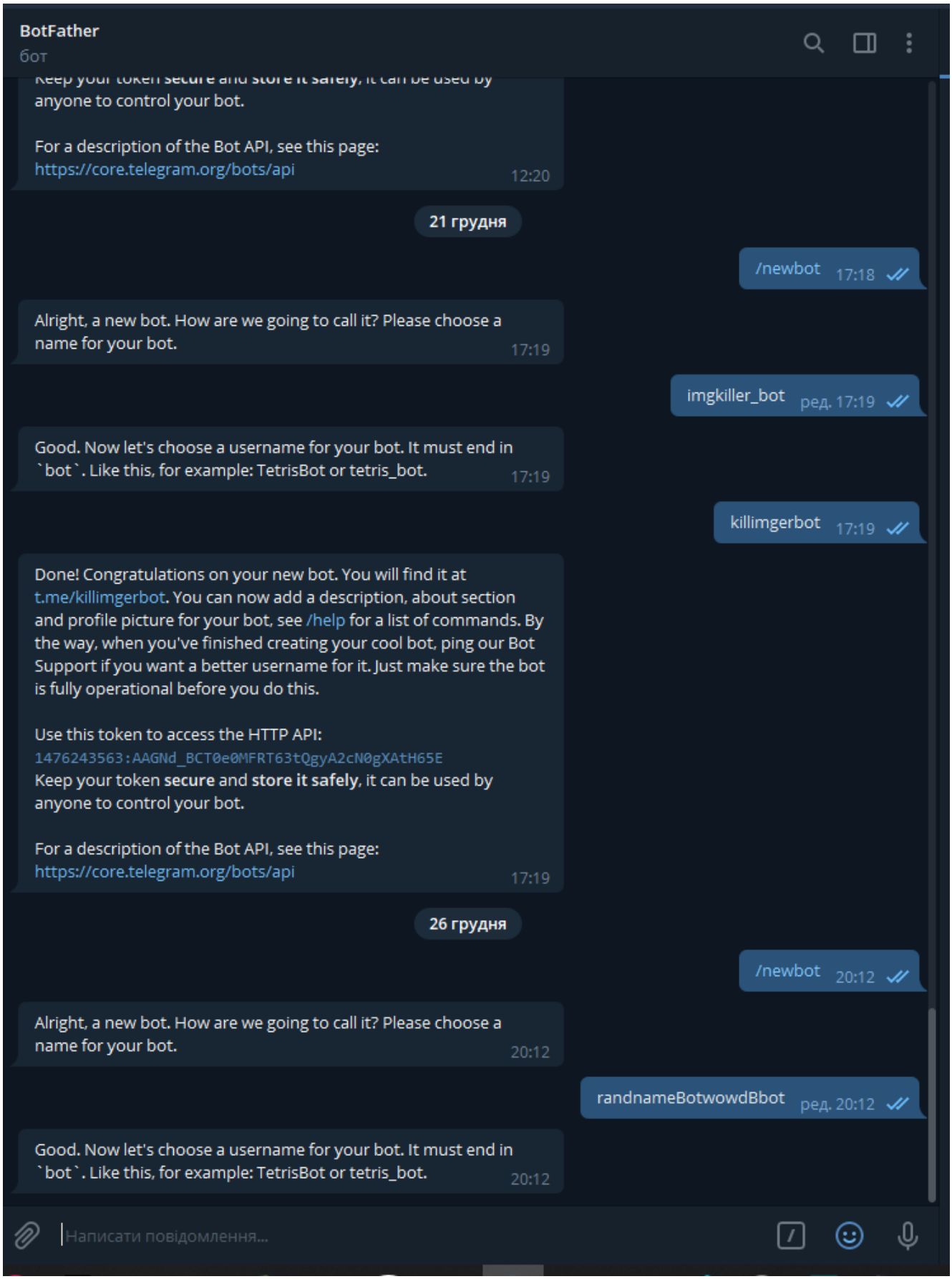

BotFather
бот
anyone to control your bot.
For a description of the Bot API, see this page:
https://core.telegram.org/bots/api
12:20
21 грудня
/newbot 17:18
Alright, a new bot. How are we going to call it? Please choose a name for your bot.
17:19
imgkiller_bot ред. 17:19
Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot.
17:19
killimgerbot 17:19
Done! Congratulations on your new bot. You will find it at t.me/killimgerbot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.
Use this token to access the HTTP API:
1476243563:AAGNd_BCT0e0MFRT63tQgyA2cN0gXAtH65E
Keep your token secure and store it safely, it can be used by anyone to control your bot.
For a description of the Bot API, see this page:
https://core.telegram.org/bots/api
17:19
26 грудня
/newbot 20:12
Alright, a new bot. How are we going to call it? Please choose a name for your bot.
20:12
randnameBotwowdBbot ред. 20:12
Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot.
20:12
Написати повідомлення...

Рисунок 3.5

Якщо все в порядку, то у відповідь ми отримаємо повідомлення з токеном. Токен необхідний для роботи з Bot API за допомогою http-протоколу. Не можна передавати його іншим і бажано не втрачати. Хорошим рішенням буде скопіювати його: зберегти в текстовий файл і покласти в надійне, завжди доступне місце - наприклад, хмарне сховище.

BotFather – це також інструмент для керування уже створеними ботами. Список доступних команд для редагування ботів наведено в таблиці:

Таблиця 3.1

BotFather, його команди та їх опис

Команда	Опис
/newbot	Створює нового бота з заданим ім'ям, що закінчується на bot, повертає токен новоствореного бота
/mybots	Повертає список ботів та дозволяє редагувати їх параметри за допомогою зручних елементів управління;
Редагування	
/setname	Змінює ім'я вже створеного бота
/setdescription	Змінює опис створеного бота, зазвичай у цьому описі вказують, що саме вміє робити бот, максимальний об'єм 512 символів

Продовження Таблиці 3.1

/setabouttext	Дозволяє змінити інформацію про бота – короткий текст до 120 символів. Користувач бачить цей текст на сторінці профілю бота.

/setuserpic	Редагувати зображення бота. Зображення бота відображається в пошуку,Тобто це аватар, який користувач бачить в діалогах і списку контактів
/setcommands	Редагування списоку команд, що підтримує бот. Користувач побачить ці команди як пропозиції, коли вони вводять «/» в чаті. Кожна команда має назву, яка повинна починатися зкосої риски «/», містити букви, цифри чи підкреслення та бути коротше 32 символів,а також містити параметри та текстовий опис;
/deletebot	Видаляє бота, дозволяє назвати іншого бота тим же ім'ям, що і видаленого
Налаштування боту	
/token	Генеруе та повертає токен для авторизації і роботи з ботом
/revoke	Відкликати поточний токен бота, допомогає в разі викрадення токену
/setinline	Вмикнути режим цитування
/setinlinegeo	Перемикати запити про розташування при використанні бота у вбудованому режимі.

Продовження Таблиці 3.1

/setinlinefeedback	Змінити налаштування зворотного зв'язку (збір статистики найбільш часто відправляються боту команд)
/setjoingroups	Визначає можливість додавання вашого бота в групи;

/setprivacy	Переключити режим конфіденційності в групах
Ігри	
/mygames	Режим редагування ігр
/newgame	Створити гру
/listgames	Повертає список створених ігр
/editgame	Редагувати гру
/deletegame	Видалити гру

Після отримання токену можна приступати до подальшого процесу розробки,

та слід пам’ятати ключові технічні відмінності ботів від користувачів:

а) У ботів немає позначки яка відображає статус підключення до мережі, а також відсутня позначка скільки часу прошло коли бот востаннє був в мережі. Натомість усі боти мають підпис «bot» під іменем, як, наприклад, «CommentsBot»

б) У ботів сталий об'єм хмарного сховища з повідомленямт, а тому

застарілі повідомлення будуть видалятися з сервера, як тільки користувач їх побачить. Проте програми можуть зберігати цю інформацію в базу данних на серверах де вони розгорнуті;

в) Боти не можуть самі розпочати діалог із користувачем. Користувач

повинен або додати їх до групи, або написати повідомлення в чат. . Користувачі можуть використовувати посилання telegram.me/<ім’я бота> або пошук для того щоб розпочати діалог із ботом;

г) Імена користувачів-ботів закінчуються на «bot»;

д) Після того як їх додали до групи, боти, за замовчуванням, не отримують

усі повідомлення із неї. Це можливо змінити у налаштуваннях.

Перед тим як обирати інструменти розробки, необхідно ознайомитися із прикладним програмним інтерфейсом (англ. Application Programming Interface, API), який надає месенджер для розробників. Прикладний програмний інтерфейс набір визначень взаємодії різнотипного програмного забезпечення. Якщо

простіше – це рівень абстракції необхідний для зручної комунікації декількох програм, при якій одна або обидві з них надають доступ до певних високорівневих функцій іншим, не розкриваючи внутрішньої їх реалізації. В той же час програма яка використовує API отримує доступ до функціоналу програми, яка надає його, без необхідності детально вивчати структуру та зв’язок її елементів. Інтерфейси прикладного програмування полегшують для розробників використання певних технологій при створенні додатків. Також API дозволяють приховати внутрішні особливості реалізації методів для покращення безпеки та забезпечення можливостей для зовнішньої взаємодії із програмою.

У попередніх розділах ми прийшли до висновку що найкращим варіантом для створення чат-бота є месенджер Телеграм. Телеграм пропонує два види API для розробників. Перше - це API Telegram, який призначений для створення сторонніх клієнтів для платформи Телеграм. В даній роботі він використовуватися не буде.

Другий - Bot API, що дозволяє легко створювати програми, які використовують повідомлення Телеграм як інтерфейс у спілкуванні із користувачем.

Власне Bot API – це те, що дозволяє легко та зручно створювати чат-ботів. Цей API дозволяє підключати ботів до системи Телеграм. З точки зору системи Телеграм, боти – це спеціальні облікові записи, для яких не потрібно встановлювати додатковий номер телефону. Ці облікові записи служать інтерфейсом для коду, який буде працювати на сервері.

Для цього не потрібно заглиблюватися в схему роботи протокола шифрування MTProto, який застосовується у системі Телеграм – їх проміжний сервер буде обробляти всі шифрування та спілкування з API Телеграм. Для зв'язку з цим серверомвикористовується простий HTTPS-інтерфейс, який пропонує спрощену версію API Телеграм. Будь-які запити до Bot API Телеграм повинні передаватися через протоколHTTPS і мати наступну форму: https://api.telegram.org/bot<ТОКЕН>/<ІМЯМЕТОДУ>. На місці <ТОКЕН>

відповідно повинен міститися токен авторизації бота, а на місці <ІМЯ МЕТОДУ> – метод який необхідно використати з API.

В API підтримуються GET і POST HTTP запити. GET-запит використовується за необхідності отримання певної інформації від API, а GET-запит – при необхідності передати якусь інформацію засобами Bot API.

Відповідь на будь-який запит містить об'єкт JSON, який завжди має логічне поле «ok», яке вказує на успішність запиту, а також може мати необов'язкове поле «description» з описом результату. Якщо «ok» встановлено в «true», запит бузуспішним, а результат запиту можна знайти в полі «result». У випадку невдалого запиту «ok» містить «false», а помилка описується в полі «description». Поле числового типу «error_code» вказує на код помилки. Деякі помилки також можуть мати необов'язкове поле «parameters», яке може допомогти автоматично опрацювати помилку.

Є два взаємовиключні способи отримання оновлень для бота – метод «getUpdates» і використання вебхуків, обробників подій. У випадку використання вебхуків сервер Телеграм, за наявності неопрацьованих оновлень, спровокує подію з якою пов'язаний веб-хук, а той відреагує на неї та отримає необхідні оновлення. Вхідні оновлення зберігаються на сервері до тих пір, поки бот не отримає їх у будьякий із вище описаних способів, але не більше 24 годин. Отже, необхідно щонайменше раз на 24 години отримувати вхідну інформацію. Насправді, це необхідно робити набагато частіше, оскільки частиною цієї інформації єповідомлення користувачів. Розглянемо основні типи які підтримує Bot API, та які можуть бути передані як у запиті бота, так і при відповіді на цей запит:

Таблиця 3.2

Короткий опис об’єктів Telegram API

User	Об'єкт являє собою користувача або бота;

Chat	Об'єкт являє собою чат чи групу;
Message	Об'єкт являє собою повідомлення. Використовується як для надсилання інформації користувачу, так і для отримання від нього відповідей;
MessageEntity	Один із елементів повідомлення (наприклад тег, ім’я користувача чи url-посилання);
PhotoSize	Відображає розмір фотографії, ескізу файлу чи стікера;
Audio	Аудіо файл, який можна програвати за допомогою вбудованого програвача;
Document	Файл будь-якого типу, окрім описаних;
Video	Відео файл;
Voice	Голосове повідомлення, яке надіслав користувач
VideoNote	Відео повідомлення, яке надіслав користувач.
Contact	Контакт з списку контактів;

Продовження Таблиці 3.2

Location	Певна точка на карті, кординати, або адреса;

Venue	Місце проведення події, тобто тимчасова точка на карті;
UserProfilePhotos	Фото(аватар) профілю користувача;
File	Файл, що можна завантажити. Максимальний розмір файлу - 20 мб;

Відповідно до наведених типів існують методи пов’язані із їх надсиланням та отриманням. Також існують методи для більш зручної роботи із чатами – вони дозволяють, наприклад, змінювати елементи, такі як фото, опис, та ін., заблокувати користувача чату чи розблокувати його та багато інших різноманітних функцій.

Важливо що Телеграм, як платформа, дозволяє ботам проводити різноманітні операції як із повідомленнями, так і з чатами, а також підтримує та навіть заохочує

використання ботів для різноманітної взаємодії. Факт того, що для покращення взаємодії із користувачами, вони обрали бота, як інтерфейс для створення інших ботів, тільки підтверджує те, що боти стали невід’ємною частиною сучасного спілкування. Крім того, як частина месенджера доступний бот для надсилання стікерів – спеціальних картинок, які заміняють emoji.

Гнучкість, зумовлена наявністю великої кількості доступних функцій, дозволяє персоналізувати ботів під потреби різних замовників. Для використання бота можна використовувати команди, повідомлення, інтерактивні клавіатури чи навіть голосові повідомлення. Відповіді користувачу можна наводити у формі текстових повідомлень, фото, відео чи файлів.

```
import telebot
#Імпортуємо бібліотеку для створення ботів
```

```
from argparse import ArgumentParser
#Імпортуємо бібліотеку для створення парсерів
bot = telebot.TeleBot('1476243563:AAGNd_BCT0e0MFRT63tQgyA2cN0gXAtH65E')
#Додаємо токен унікальний токен бота, при розгортанні на сервер використовується файл
#конфігурації в якому вказується токен, за для забезпечання його безпеки

@bot.message_handler(commands=['start'])
def start_message(message):
    #Створюємо змінну message, що реагує на команду /start
    bot.send_message(message.chat.id, 'Модерирую стикеры и картинки')
    #Створюємо повідомлення що виводиться після команди /start

@bot.message_handler(content_types=['photo', 'sticker',])
def handle_img(msgstimg):
    #Створюємо змінну msstimg, куди вноситься контент, що являє
    #себе джерелом спаму, в цьому випадку стікери та зображеня
    bot.delete_message(msgstimg.chat.id, msgstimg.message_id)
    #Метод що видаляє повідомлення що містить заданий контент
    bot.send_message(msgstimg.chat.id, 'Сообщение удалено')
    #Створюємо повідомлення що виводиться після виконання методу


def main():
    parser = ArgumentParser()
    #Викликаємо парсер
    parser.add_argument('--mode')
    #Задаємо параметр парсингу desc
    bot.polling()
```

```
if __name__ == '__main__':
    main()
    #Нескінченний цикл роботи бота
```

Розгоратання бота

Перш за все хочу зазначити, що боти працють за замовчуванням по 433 порту. Оскільки працює бот працює виключно через HTTPS то доступні порти: 443, 80, 88, 8443.

Для постійного отримання доступу до чат-боту, після реалізації йогонеобхідно розмістити на віддаленому сервері. В якості платформи був розглядається Heroku, а також розгорнути бота можна на власній машині, саме цей спосіб я хочу розібрати у білш уважно в кваліфікаційній роботі, оскільки він найменш освітленний, .

Heroku

Розміщення на хмарної PaaS-платформі Heroku схоже з роботою розподіленої системи контролю версій (Git). Існує три способи розгортання:

- Heroku Git і Container Registry використовуючи Heroku CLI;
- підключення аккаунта GitHub з автоматичним розгортанням;
- підключення аккаунта Dropbox з автоматичним розгортанням.

Для розгортання чат-бота були виконанітакі дії:

- реєстрація на хмарної SaaP-платформі Heroku;
- скачав і встановлений Heroku CLI;
- в терміналі операційної системи або вбудованому в IDE виконаний вхід в акаунт Heroku командо «heroku login»;
- клонований репозиторій з віддаленого Git-сервера Heroku налокальну машину за допомогою команди «heroku git: clone -a <APP_NAME>»;

- зафіксовані зміни в коді за допомогою команд «git add.» і«Git commit -am <COMMIT_NAME>»;
- всі зафіксовані зміни були відправлені на віддаленийсервер хмарних обчислень Heroku командою «git push heroku master».

Після виконання всіх зазначених команд в правильній послідовності починається передача даних на віддалені сервера. Якщо розгортання пройшло успішно, то на терміналі відобразиться довідкова інформація про стан програми та режим доступу до нього. Таким чином, чат-бот був розгорнутий на віддаленому сервері хмарних обчислень Heroku, що дозволить користувачам отримувати доступ безперебійно.

```
remote: -----> Build succeeded!
remote: -----> Discovering procces types
remote:        Procfile declare types       -> (none)
remote:        Default types for buildpack  -> web
remote:
remote: -----> Compressing...
remote:        Done: 2.1 M
remote: -----> Launching...
remote:        Realesed v1
remote:        https://killimgbotpydip.herokuapp.com/ deployed to Heroku
remote:        Verifying deploy... done.
remote:        https://git.heroku.com/killimgbotpydip.dip
* [new branch]         master -> master
```

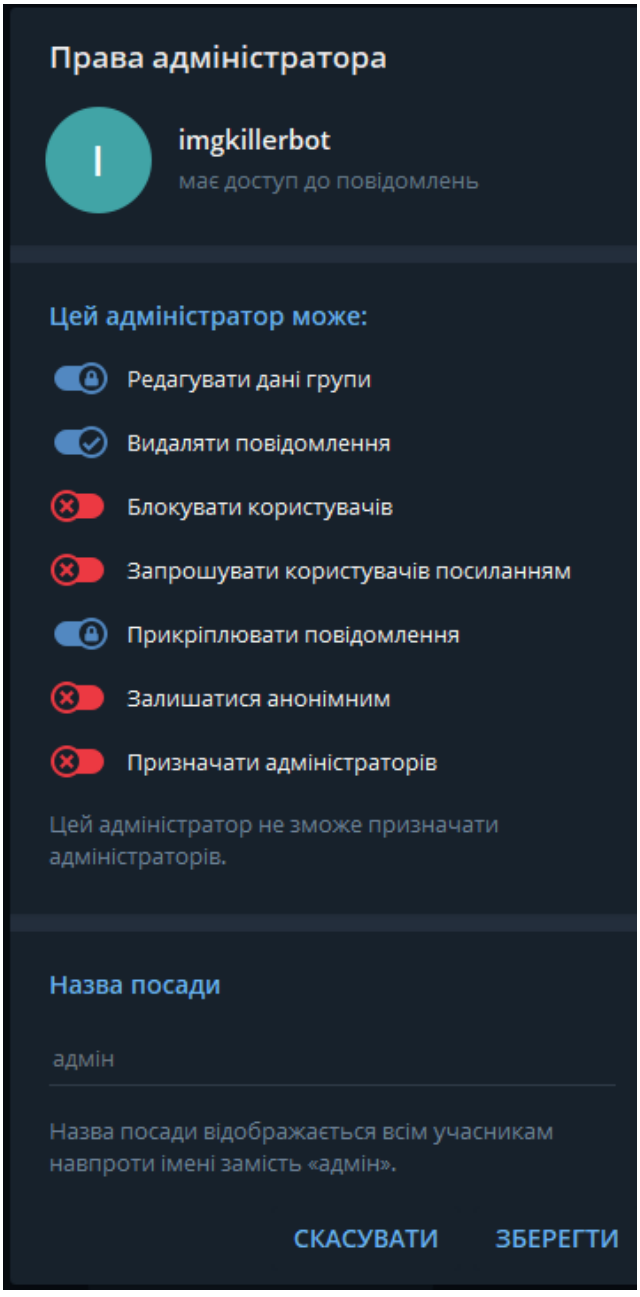
Розгортання бота на хмарній платформі Heroku Рисунок 3.6

Налаштування бота

Описую алгоритм для Windows desktop клієнта. Мається на увазі, що ви адміністратор чату і можете додавати в нього інших адміністраторів.

- Заходимо в чат, натискаємо на назву його назву зверху
- У вікні, зверху праворуч від напису "Інформація про групу" натискаємо на іконку з трьох точок
- У меню, натискаємо на "Управління групою"
- У меню, натискаємо на "Адміністратори"
- У вікні знизу натискаємо на "Додати адміністратора"
- У вікні, в рядку пошуку вводимо назву бота в нашому випадку imgkillerbot

- У открившемя вікні виставляємо боту права на видалення повідомлень, всі інші права відключаємо(Рисунок 3.7).
- Тиснемо "Зберегти"



Налаштування бота у групі Рисунок 3.7

Роботу бота продемонстровано на Рисунку 3.8 та Рисунку 3.9, тобто видно як бот видаляє стікер та залишає повідомлення про це.

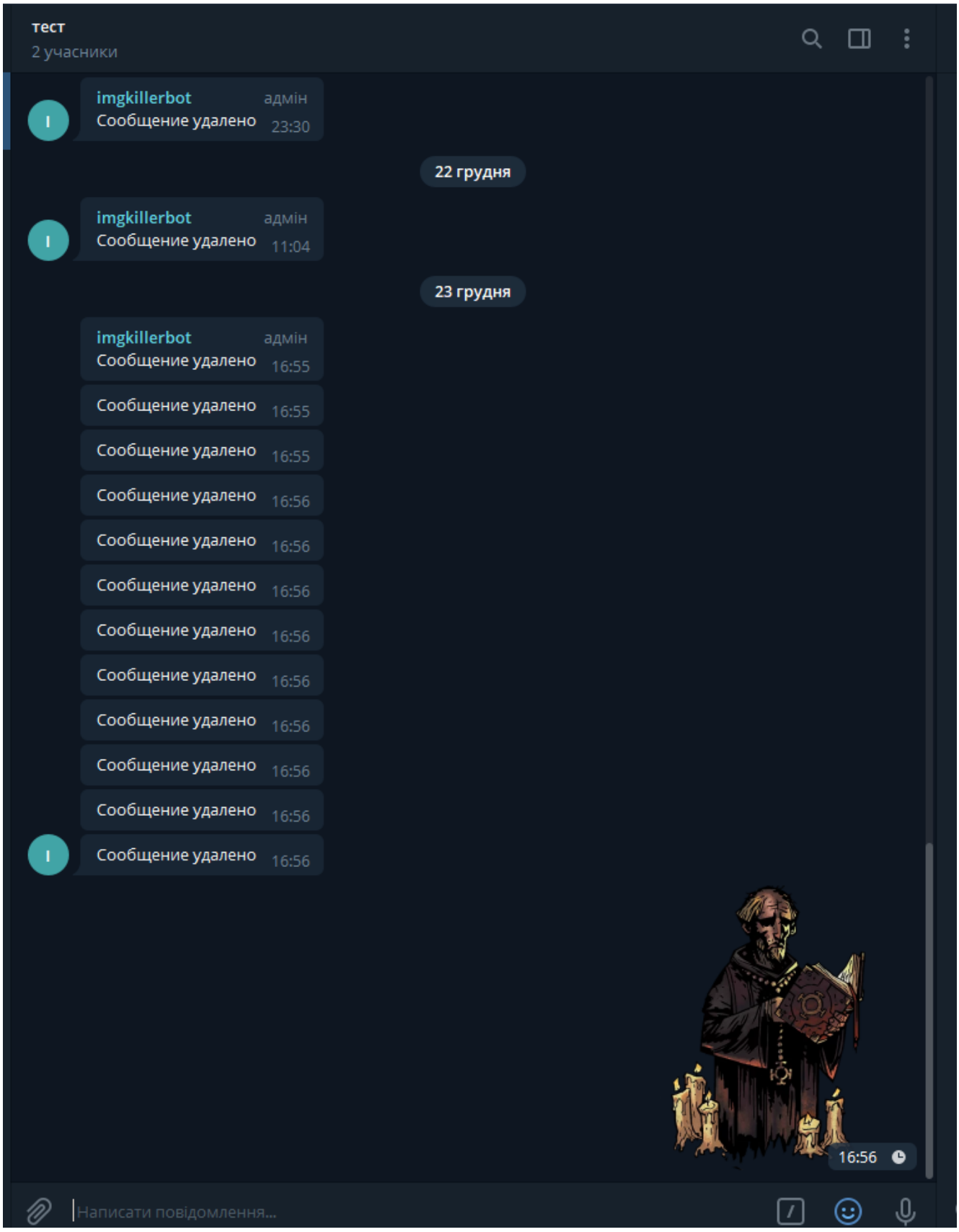

тест
2 учасники
imgkillerbot адмін
Сообщение удалено 23:30
22 грудня
imgkillerbot адмін
Сообщение удалено 11:04
23 грудня
imgkillerbot адмін
Сообщение удалено 16:55
Сообщение удалено 16:55
Сообщение удалено 16:55
Сообщение удалено 16:56
Сообщение удалено 16:56
Сообщение удалено 16:56
Сообщение удалено 16:56
Сообщение удалено 16:56
Сообщение удалено 16:56
Сообщение удалено 16:56
Сообщение удалено 16:56
Сообщение удалено 16:56
16:56
Написати повідомлення...

Рисунок 3.8

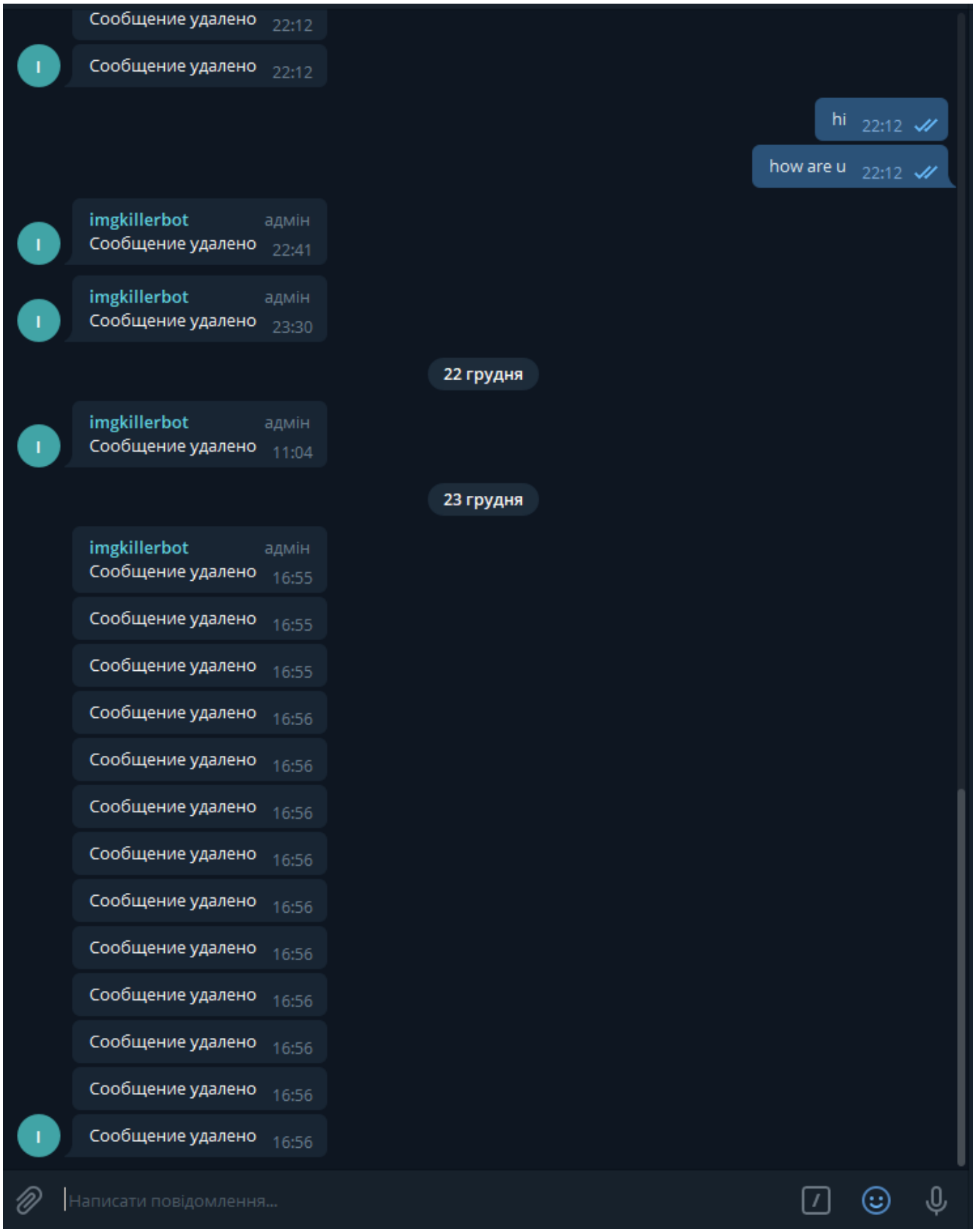

Сообщение удалено 22:12
Сообщение удалено 22:12
hi 22:12
how are u 22:12
imgkillerbot адмін
Сообщение удалено 22:41
imgkillerbot адмін
Сообщение удалено 23:30
22 грудня
imgkillerbot адмін
Сообщение удалено 11:04
23 грудня
imgkillerbot адмін
Сообщение удалено 16:55
Сообщение удалено 16:55
Сообщение удалено 16:55
Сообщение удалено 16:56
Сообщение удалено 16:56
Сообщение удалено 16:56
Сообщение удалено 16:56
Сообщение удалено 16:56
Сообщение удалено 16:56
Сообщение удалено 16:56
Сообщение удалено 16:56
Сообщение удалено 16:56
Сообщение удалено 16:56
Написати повідомлення...

Рисунок 3.9

Рисунок 3.10

Розгортання на власній машині

При адмініструванні комп'ютерів часто потрібно виконувати періодичні завдання обслуговування. Планувальник завдань Windows 10 є засобом для перегляду і управління запланованими завданнями. Оскільки при збоях, або нестабільному інтернет зьєднанні екземляр бота(процесс Python) буде завершуватися. Мною було прийнято рішення, виконавчий файл бота внести в планувальник завдань. Щоб процесс роботи в разі таких збоїв поновлювався.

Планувальник завдань - це оснащення mmc (Microsoft Management Console), за допомогою якої можна призначити різні завдання, які будуть проводитися в певний час або при виникненні певних подій. Як правило, такі завдання застосовуються для автоматизації окремих процесів:

- параметрическая автоматизація різних завдань, які виконуються на комп'ютері, наприклад:
- автоматичне створення контрольних точок відновлення в певний час
- очищення диска в певні дні
- запуск в певний час дефрагментації диска
- діагностичне тестування
- оптимізація процесу завантаження комп'ютера

Операційна система Windows 10 містить кілька інструментів для планування завдань, включаючи такі, як Планувальник завдань, інструмент командного рядка Schtasks і кілька командлетів консолі Windows PowerShell. Ці інструменти можна використовувати для планування завдань як на локальних, так і на віддалених робочих станціях.

Завдання можуть мати різні пов'язані з ними властивості, включаючи наступні:

- Тригери. За допомогою тригерів можна задати умови початку і завершення виконання різних завдань. Завдання можуть виконуватися за розкладом, при вході користувача в систему, при запуску комп'ютера і т.д. У параметри запуску завдань можна включити події, пов'язані з діями користувача. Використання тригерами подій значно розширюють можливості управління процесами.
- Дії. Параметр завдання Дії визначає особливості виконання запущеного процесу. Дозволяє процесу запускати програми, відправляти повідомлення електронної пошти або виводити повідомлення.

- Умови. Параметр завдання завдання умови та уточнює події, при яких активний процес запускається або зупиняється. Наприклад, при заданому умови можна запускати або зупиняти будь-яке завдання на основі тривалості простою комп'ютера. Умови можна використовувати, щоб вивести комп'ютер з режиму сну для виконання завдання. Можна налаштувати параметри умови для виконання завдання за умови роботи комп'ютера від мережі, і припинення виконання завдання при переході на живлення від батарей.

Відкривається планувальник завдань, за допомогою панелі управління, він знаходиться у розділі адміністрування. Після запуску Планувальника завдань слід створити зайвдання

Для того щоб додати бота до планувальника завдань потрібно виконити наступні дії:

- Натиснути «Створити задачу»
- В розділі «Загальні» запонити поле Ім’я задачі, та надати дозволи як на рисунку 3.11, це потрібно, щоб задача виконувалсь навідь тоді коли комп’ютер заблоковано.
- В розділі «Дії» додати виконавчий файл боту.
- В розділі «Умови» заповнити так як на рисунку 3.12
- Після цього запустити задачу.

Залишилось лише перевірити чи виконується завдання, для цього можна використати диспечер завдань, в розділі «Деталі» потрібно в нашому випадку знайти процесс Python.exe(Інтерпретатор Python) та дізнатися ідентифікатор процессу. Після цього залишиться лише виконати в командному рядку команду «netstat -n -p» та за ідентифікатором процесу знайти інформацію за яким портом працює бот, та інформацію про саме з’єднання(Рисунок 3.14

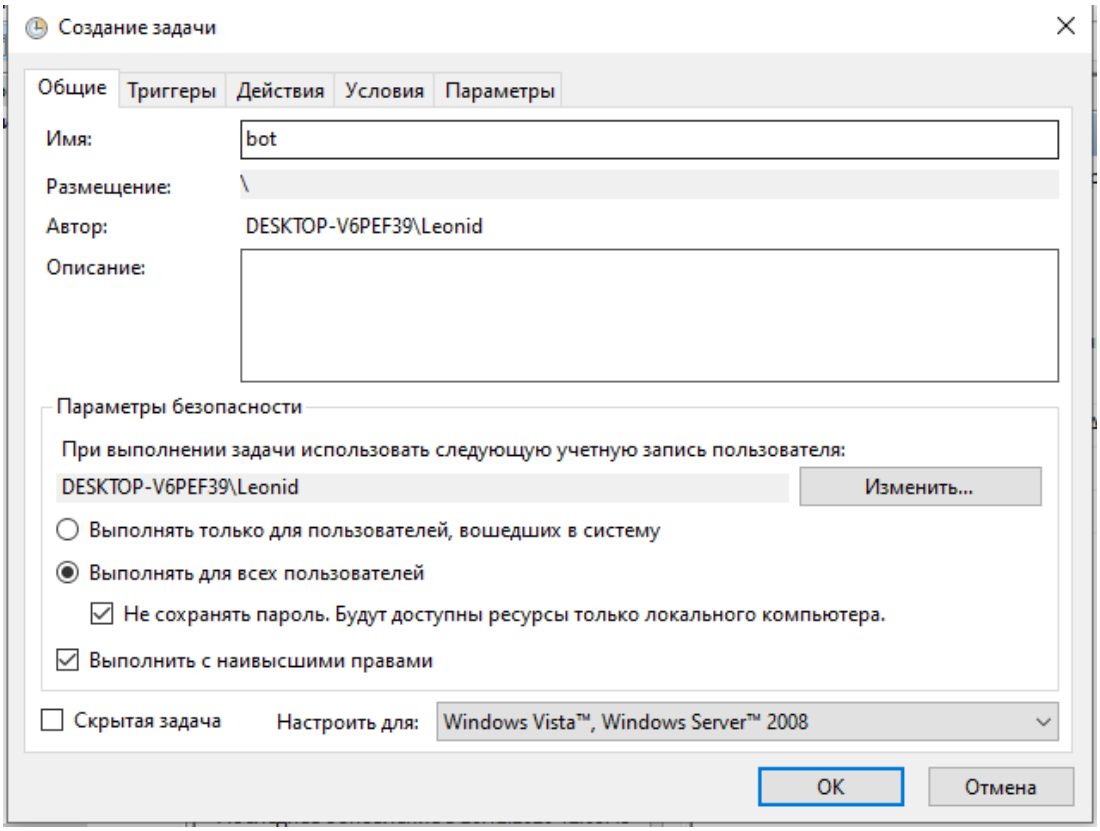

Создание задачи
Общие
Триггеры
Действия
Условия
Параметры
Имя:
bot
Размещение:
\
Автор:
DESKTOP-V6PEF39\Leonid
Описание:
Параметры безопасности
При выполнении задачи использовать следующую учетную запись пользователя:
DESKTOP-V6PEF39\Leonid
Изменить...
Выполнять только для пользователей, вошедших в систему
Выполнять для всех пользователей
Не сохранять пароль. Будут доступны ресурсы только локального компьютера.
Выполнить с наивысшими правами
Скрытая задача
Настроить для:
Windows Vista™, Windows Server™ 2008
OK
Отмена

Рисунок 3.11

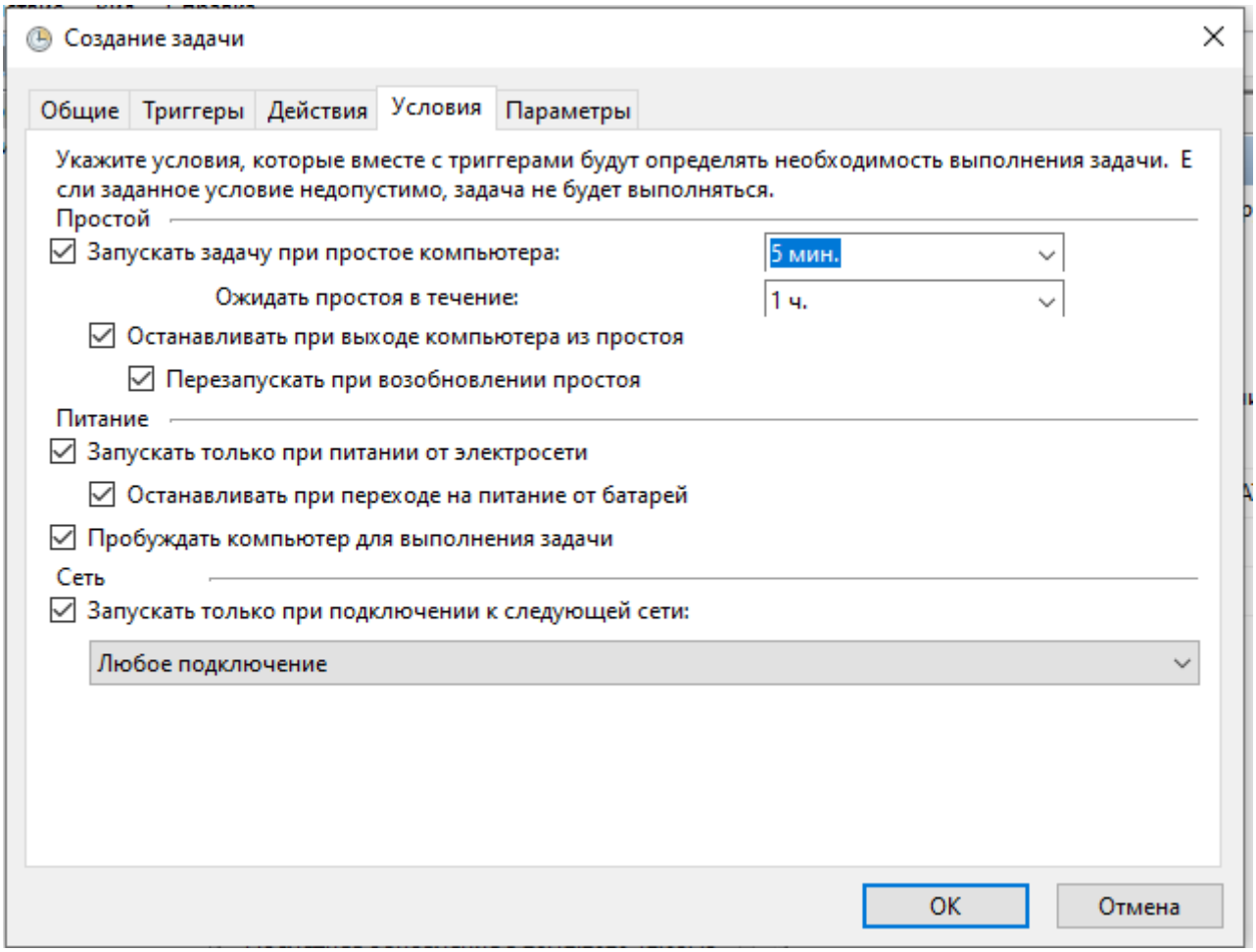
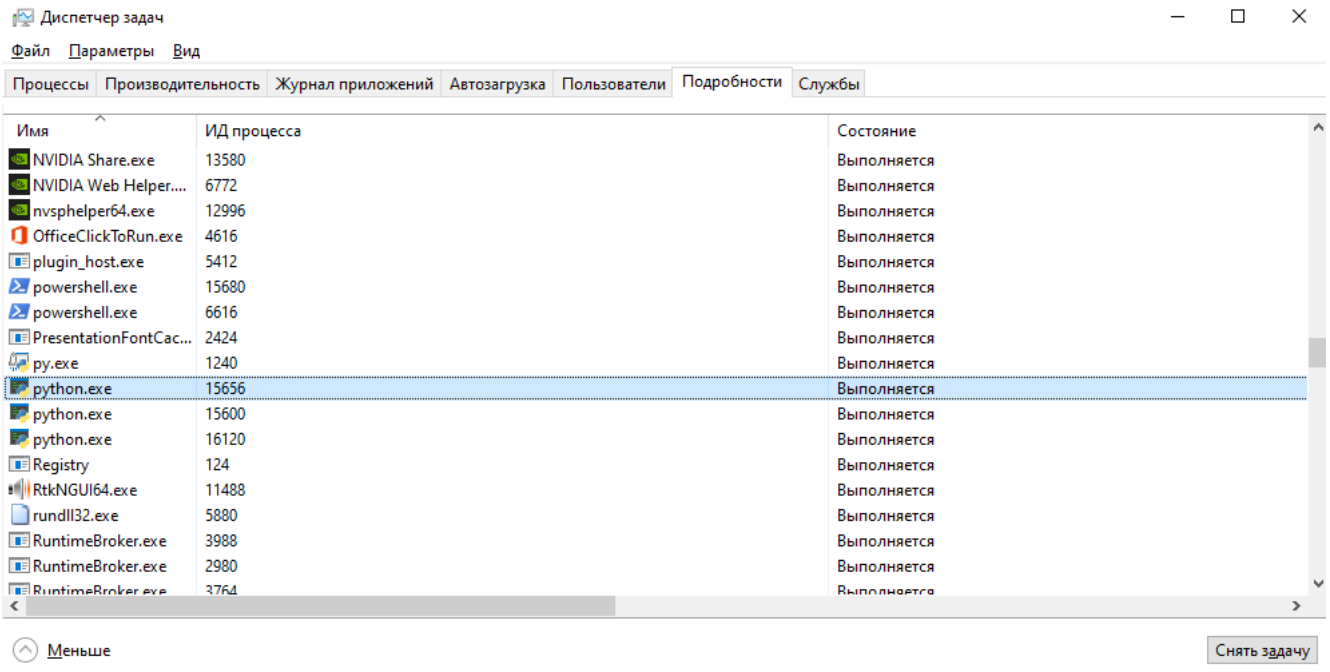
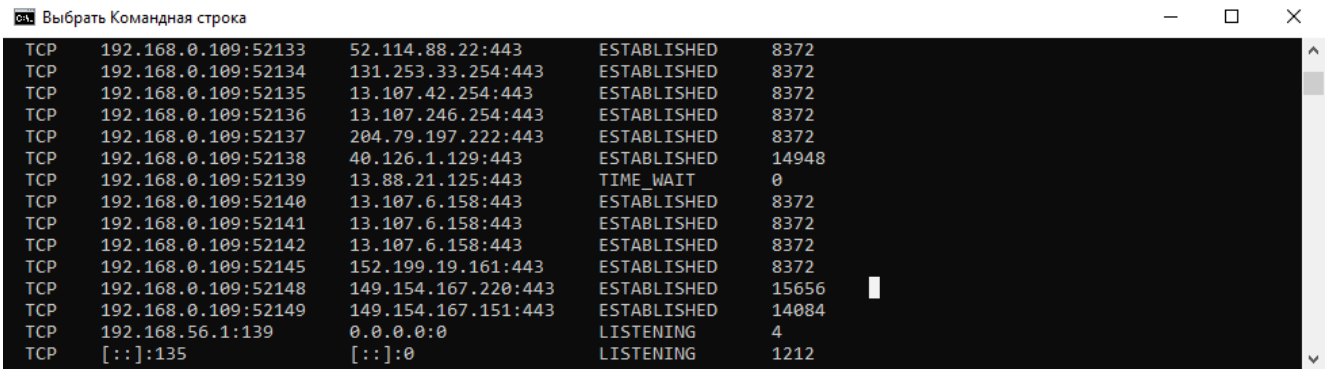


Рисунок3.12



Єкземляр боту в диспечері завдань Рисунок 3.13



Результат netstat Рисунок 3.14

Висновки до розділу 3

Проаналізовано….

Досліджено….

Виокремлено ……

Описано……

Підкреслено……..

В даний час популярність месенджерів як засобів спілкуваннянезмінно зростає. Компанії, сім'ї, друзі щодня користуютьсяможливостями обміну повідомленнями та медіаконтенту на відстані. так самоварто відзначити зростання популярності такого виду програмних продуктів як чат-боти, які працюють на платформах месенджерів. Цілодобова служба підтримки користувачів, конвертація документів і медіафайлів, замовлення таксі, пошук необхідних даних і багато іншого в даний час може бути реалізовано в рамках лише одного месенджера. Користувачам не доведеться завантажувати безліч додатківдля вирішення вузько завдань, тому що досить мати лишемесенджер і необхідний набір чат-ботів, які не займають місце впам'яті смартфона. В рамках випускний кваліфікаційної роботи були виконаніпоставлені завдання. По-перше, були вивчені месенджери. Було проведенопорівняння і аналіз переваг і недоліків, внаслідок чого був обраниймесенджер Telegram як найзручніший і доступний в плані документаціїTelegram Bot API. По-друге, були вивчені наявні аналоги чат-бота на платформіTelegram, а також виявлено їх переваги, недоліки та цікаві рішення. На

основі цього були виявлені вимоги для розробки чат-бота. В рамках останньої виконаної завданням було обрано технології та середовище для розробки чат-бота серед яких SaaP Heroku і VS Code.