Документация к Telagram боту “CVE Crawler Bot from EDA Team”.

1. ***Эксплуатация продукта:***

Для начала работы необходимо написать команду /start, затем бот отправит приветственное сообщение. После чего будут доступны следующие команды:

- /help - выводит список доступных команд
- /find - поможет узнать всю информацию о CVE по cveId. Формат ввода cveId: год, четырехзначный номер. Пример: 1999 0001. Бот предоставляет информацию о CVE, написанных с 1999 г. По настоящее время.
- /recent - выводит CVE, которые появились за последнее время
- /recent_in - выводит cveId всех новых CVE зарегистрированных за последние несколько дней (количество дней пользователь выбирает сам. Формат ввода: число. Пример: 1).
- /last_activity - выводит cveId CVE, активность которых была зарегистрирована за последнее время.

2. ***Установка продукта:***

Необходимо скачать с GitHub (ссылка:)все файлы. Пользователю необходимо скачать базу данных с сайта CVE.ORG (прямая ссылка на базу данных - https://github.com/CVEProject/cvelistV5/archive/refs/heads/main.zip). Все необходимые библиотеки предустановлены в виртуальной среде проекта, поэтому для запуска кода пользователю необходимо просто вставить в необходимый файл API своего бота.

3. ***Конфигурация и архитектура ПО:***

Продукт был разработан и написан на языке программирования Python. Основной код находится в файле “main.py”, в котором создан Telegram бот “CVE Crawler Bot from EDA Team” с помощью библиотеки aiogram (подробнее о ней - https://docs.aiogram.dev/en/latest/index.html). Алгоритмы, которые включает в себя Telegram бот написаны в отдельных файлах, и выполнены как функции, которые были импортированы в основной файл “main.py”. Программа анализирует таблицу в формате json и по желанию пользователя выводит ту или иную информацию. Для работы с файлами в формате json была использована библиотека json. Принципы работы файлов:

1. texts.py - файл с переменными, которые хранят в себе заготовленные тексты с ответами бота.

2. finder.py - файл с алгоритмом, который ищет CVE по cveId. Алгоритм: происходит обращение к базе данных. Происходит обращение к необходимым полям базы, и данные записываются в переменные. Переменные конкатенируются между собой и готовая строка с текстом отправляется боту.
3. recent.py - файл с алгоритмом, который выводит последнюю добавленную CVE в базе данных. Схема реализации похожа с finder.py. Идет обращение к базе данных, к полю new, которое хранит нужную информацию. Результат также конкатенируются и отправляется боту.
4. find_recent_in.py - файл с алгоритмом, который выводит все новые CVE до определенного дня. В функцию подается количество дней
5. last_activity.py - файл с алгоритмом, который выводит cveId CVE, активность которых была зарегистрирована за последнее время. Программа обращается к файлу, в котором зарегистрированы активности CVE за определённый период и выводит последнюю.