Технико-правовое задание: Бот для безопасного взаимодействия заявителей с правозащитниками

Основной функционал

Архитектура взаимодействия:

Бот действует как **защищенный посредник** между заявителями (в публичном канале) и командой правозащитников (в приватной админ-группе). Заявитель видит только ответы администратора, не зная его личность, а администратор получает все вопросы через анонимизированный интерфейс бота.

Базовые функции:

Заявитель отправляет вопрос в канал (через реплай на специальное сообщение бота или в прямые сообщения боту). Бот фиксирует временную метку, генерирует уникальный ID заявки и отправляет сообщение в защищенную групповую администраторскую часть. Администратор-правозащитник отвечает в приватной группе, бот возвращает ответ заявителю без раскрытия личности отвечающего. Весь диалог хранится отдельно для каждой заявки с шифрованием на уровне базы данных.

Работа с файлами и медиаконтентом:

Заявитель может прикреплять файлы любого допустимого Telegram формата (документы, архивы, изображения, видео до размера, установленного администратором). Файлы передаются как через Telegram Bot API без сохранения в облаке провайдера — используется прямой download по file_id. Перед передачей правозащитнику **все метаданные (EXIF, GPS, временные метки, информация о устройстве) автоматически удаляются** с помощью библиотек piexif (для JPEG) и других. При необходимости файлы переупаковываются в формат без встроенной информации. Администратор при отправке файла заявителю получает версию без метаданных.

Навигация и коммуникация:

Клавиши быстрого доступа в боте: "Новая заявка", "История моих заявок", "Статус текущей заявки", "Контакты правозащитников". Возможность добавления гиперссылок в текст ответов (стандартные Telegram URL и внешние ссылки). Поддержка реакций (emoji) на сообщения внутри диалога заявитель-администратор через встроенные кнопки Telegram.

Администратор может использовать форматирование текста (жирный, курсив, моноширинный).

Функции безопасности и приватности

1. Двухфакторная аутентификация администраторов

Каждый правозащитник-администратор аутентифицируется при входе через **2FA по Telegram**:

- Первый фактор: username и пароль, сгенерированный администратором системы
- Второй фактор: секретный код, отправляемый боту (через OTP-генератор или специальное сообщение в приватный чат с админ-ботом)

Логирование всех попыток входа с фиксацией IP-адреса, времени, устройства. Блокировка после 5 неудачных попыток на 30 минут. Необходимо использовать **SSH-ключи для доступа к серверам** вместо паролей, отключить вход по паролю на уровне sshd.

2. Защита от раскрытия анонимности администратора

Маскирование идентификации:

- Бот переписывает ID администратора при отправке ответов, используя зашифрованный временный идентификатор (например, "Admin_12h34m_session_key")
- Все сообщения отправляются через встроенный API Telegram (не через скрипты на сервере администратора), чтобы IP администратора не проходил в метаданных
- Используется **прокси-слой**: сообщения от админа проходят через Tor-bridge или VPN на уровне сервера, если организация работает в опасных юрисдикциях
- Удаляются все служебные заголовки HTTP (User-Agent, X-Forwarded-For) при коммуникации между сервером бота и API Telegram

Защита от timing-атак (определение администратора по времени ответа):

- Бот добавляет случайную задержку перед отправкой ответа (от 30 секунд до 5 минут) для рассеивания временных шаблонов

- Администраторы могут отключить эту функцию в критических ситуациях

3. Шифрование данных

End-to-End шифрование на уровне приложения (так как Telegram Bot API не поддерживает E2E для ботов по умолчанию):

- Все сообщения заявителей, идущие в админ-группу, шифруются с помощью **AES-256-GCM**
- Ключ шифрования хранится на защищенном сервере (не в облаке Telegram), администраторы имеют отдельные ключи для расшифровки
- Сообщения расшифровываются только на безопасном локальном устройстве администратора (не на сервере)
- Для передачи ключей используется **HTTPS с валидным SSL-сертификатом** и проверкой HMAC-подписи входящих запросов

Автоматическое удаление:

- Все сообщения в интеграции бот-администратор удаляются автоматически через **30 дней** (настраивается)
- История заявок пользователя хранится отдельно в защищенной БД с логированием доступа и выдаётся только администраторам

4. Защита от утечки метаданных файлов

- При загрузке файла бот **автоматически удаляет EXIF-данные** (GPS, камера, дата создания, время), используя python-библиотеки piexif, Pillow
- Архивы (ZIP, RAR, 7z) проверяются на наличие скрытых файлов и перепаковываются
- Документы PDF проходят через утилиту ghghostscript для удаления встроенных метаданных
- Изображения переконвертируются в формат без встроенной информации (PNG без метаданных или JPEG после обработки через PIL с параметром exif=None)

- **Уведомление администратору при попытке загрузить файл с опасными метаданными**: система не блокирует, но предупреждает о необходимости очистки на стороне администратора

5. Верификация новых членов и защита от фейков

Квиз-система для верификации:

- Каждый новый заявитель в канале проходит **обязательную верификацию** перед отправкой первого вопроса
- Бот предлагает ответить на один из 3-5 вопросов для проверки идентичности (например, "Какой город вы указали в профиле", "Повторите слово из правил канала")
- После неудачной попытки верификации заявитель может повторить через 1 час
- Логирование всех попыток верификации и блокировка IP/аккаунтов после 10 неудачных попыток на 24 часа

6. Rate limiting и защита от DDoS/спама

- Ограничение частоты отправки: **не более 100 сообщений в 1 час для одного заявителя**
- Система слежения за быстрыми повторяющимися запросами: если 3 сообщения пришли менее чем за 10 секунд, очередное сообщение отклоняется с информацией о переполнении
- Использование **PyRateLimiter** библиотеки для Python для реализации алгоритма "протекающего ведра"
- Администраторы получают **дашборд с графиками** частоты запросов, чтобы обнаружить подозрительную активность

7. Модерация и фильтрация контента

- **Автоматическое удаление сообщений** с явным фишингом, ссылками на вредоносные сайты (проверка через VirusTotal API)
- Фильтрация стоп-слов, включенных в настройки (мат, спам, названия экстремистских организаций) с использованием **нечеткого совпадения**, чтобы обнаруживать модифицированные версии
- Логирование удаленных сообщений отдельно для аудита

- Очередь на ручное одобрение администратором при срабатывании фильтров (система не удаляет сразу, а ставит на предварительное рассмотрение)

8. Ролевая система доступа (RBAC)

Четыре уровня ролей с **градуированными правами** (как в официальных Telegram каналах):

Роль	**Права**	**Ограничения**
Владелец	Полный контроль, добавление/удаление администраторов, изменение правил, доступ к полным логам	-
Администратор высокого уровня	Просмотр заявок, отправка ответов, добавление новых модераторов, доступ к расширенным логам	Не может удалять администраторов, менят правила шифрования
Модератор	Просмотр заявок, фильтрация спама, отправка типовых ответов, видит анонимизированные данные	Не может видеть истори заявок старше 7 дней, не может изменять параме безопасности
Волонтер (наблюдатель)	Чтение заявок только, не может писать ответы	-

Каждый администратор получает **уникальный токен доступа**, который ротируется еженедельно. Попытка доступа с истекшим токеном — автоматическое логирование и уведомление владельца.

9. Аудит деятельности и логирование

Централизованный лог всех операций:

- Время создания заявки, ID заявителя (анонимизированный), текст вопроса (зашифрованный в логе)
- Администратор, обработавший заявку (анонимизированный ID)

- Время и содержание ответа (зашифрованное)
- Все операции с файлами (загрузка, скачивание, удаление)
- Попытки несанкционированного доступа, ошибки системы
- Экспорт логов в защищенный файл с подписью для последующего аудита

Логирование хранится **не менее 90 дней**, но может быть настроено на 1-3 года для организаций, требующих полной документации. Логи шифруются и хранятся отдельно от основной БД, доступ ограничен только владельцу.

10. Защита от скриншотов и пересылки сообщений

- **Встроенное уведомление** при попытке сделать скриншот (если поддерживается платформой) — администратор получает уведомление
- Запрет на пересылку сообщений из диалога между заявителем и администратором через встроенные права Telegram (опция "Disable message forwarding")
- Отсутствие возможности скопировать текст напрямую (форматирование через специальный шрифт)

12. Уведомление о попытках перехвата

- Если система обнаружит несколько попыток неправильного доступа с одного ID заявителя, администратор получит **красное уведомление** ("Возможная фишинг-атака")
- Автоматическая блокировка подозрительного аккаунта на 24 часа с вариантом восстановления через верификацию
- Логирование всех таких попыток для анализа паттернов атак

Дополнительные функции безопасности, специфичные для правозащиты

13. Защита от государственного мониторинга (в юрисдикциях с цензурой)

- **Скрытие факта существования бота**: канал должен быть описан нейтрально, без явной привязки к "правозащитным" функциям
- Использование **обфускации имен переменных** в коде бота (если работает на открытом хостинге)
- Опция для администраторов: **запущенный бот в режиме "молчания"** — внешне выглядит как обычный бот, но только избранные администраторы знают полные функции

14. Резервное копирование и восстановление данных

- **Автоматическое резервное копирование** всех данных каждые 6 часов
- Возможность **быстрого восстановления** после сбоя без потери данных

15. Страховка от утечки данных

- **Инструмент для быстрого удаления всех данных** если возникла угроза безопасности (кнопка "Emergency Wipe" с двойным подтверждением)
- Процедура: все сообщения, файлы, логи удаляются, база данных перезагружается с минимальной информацией
- Система уведомляет всех администраторов за 15 минут до удаления (время для отмены)

Техническое окружение - python, aiogram и проекта https://github.com/chekazoid/GPT-TelegramBot, плюс те технологии которые необходимы.