**Прокси-сервер с веб-интерфейсом**

**Обзор проекта**

Проект представляет собой многофункциональный прокси-сервер с веб-интерфейсом управления, реализованный на Python. Сервер поддерживает как HTTP, так и HTTPS протоколы, имеет систему блокировки доменов и функционал мониторинга в реальном времени.

**Основные возможности**

* Проксирование HTTP и HTTPS запросов
* Блокировка доменов через веб-интерфейс
* Мониторинг доступа в реальном времени
* Фильтрация и логирование запросов
* Веб-интерфейс управления на Flask
* Real-time обновления через WebSocket
* Управление списком заблокированных доменов

**Архитектура проекта**

**Структура директорий**

proxy\_server/

├── data/ # Директория с данными

│ ├── Sites.txt # Файл логов доступа

│ └── blocked\_domains.json # Список заблокированных доменов

├── web/ # Веб-интерфейс

│ ├── templates/ # HTML шаблоны

│ │ ├── monitor.html # Страница мониторинга

│ │ └── block.html # Страница управления блокировкой

│ └── app.py # Flask приложение

├── constants.py # Константы и менеджер доменов

├── simple\_proxy.py # Основной прокси-сервер

├── main.py # Главный файл для запуска

└── readme.md # Документация

Оглавление

[**Прокси-сервер с веб-интерфейсом** 1](#_Toc185630380)

[**Обзор проекта** 1](#_Toc185630381)

[**Основные возможности** 1](#_Toc185630382)

[**Архитектура проекта** 1](#_Toc185630383)

[**Структура директорий** 1](#_Toc185630384)

[**Компоненты системы** 1](#_Toc185630385)

[**Установка и запуск** 2](#_Toc185630386)

[**Требования** 2](#_Toc185630387)

[**Установка** 2](#_Toc185630388)

[**Запуск** 3](#_Toc185630389)

[**Конфигурация** 3](#_Toc185630390)

[**Настройка прокси-сервера** 3](#_Toc185630391)

[**Фильтрация доменов** 3](#_Toc185630392)

[**Файлы данных** 3](#_Toc185630393)

[**Безопасность** 3](#_Toc185630394)

[**Реализованные механизмы** 4](#_Toc185630395)

[**Рекомендации** 4](#_Toc185630396)

[**Мониторинг и обслуживание** 4](#_Toc185630397)

[**Логи** 4](#_Toc185630398)

[**Веб-интерфейс** 4](#_Toc185630399)

[**Устранение неполадок** 4](#_Toc185630400)

[**Частые проблемы** 4](#_Toc185630401)

[**Отладка** 5](#_Toc185630402)

**Компоненты системы**

**1. Прокси-сервер (simple\_proxy.py)**

* Основной компонент системы, обрабатывающий сетевой трафик
* Поддерживает HTTP и HTTPS протоколы
* Реализует фильтрацию доменов
* Ведет логирование доступа
* Использует многопоточную обработку соединений
* Настраиваемые параметры:
  + Хост: 127.0.0.1 (по умолчанию)
  + Порт: 8080 (по умолчанию)
  + Размер буфера: 65536 байт
  + Максимум соединений: 5
  + Таймаут соединения: 10 секунд

**2. Менеджер доменов (constants.py)**

* Управляет списком заблокированных доменов
* Хранит данные в JSON формате
* Основные методы:
  + get\_blocked\_domains(): получение списка заблокированных доменов
  + add\_domain(domain): добавление домена в блок-лист
  + remove\_domain(domain): удаление домена из блок-листа
* Автоматически создает файл конфигурации при первом запуске

**3. Веб-интерфейс (web/app.py)**

* Flask-приложение для управления прокси-сервером
* Два основных раздела:
  1. Мониторинг (/):
     + Отображение логов в реальном времени
     + WebSocket для real-time обновлений
     + Автопрокрутка логов
     + Индикатор состояния соединения
  2. Управление блокировкой (/block):
     + Добавление/удаление доменов
     + Просмотр списка заблокированных доменов
     + Мгновенное применение изменений

**4. Система логирования**

* Записывает все обращения к доменам
* Фильтрует служебные запросы и известные сервисы
* Формат лога: [timestamp] Accessing: domain
* Отдельные сессии разделяются визуально
* Поддерживает real-time мониторинг

**Установка и запуск**

**Требования**

* Python 3.6+
* Зависимости:
  + Flask
  + Flask-SocketIO
  + watchdog
  + psutil

**Установка**

# Клонирование репозитория

git clone [repository-url]

cd proxy\_server

# Установка зависимостей

pip install flask flask-socketio watchdog psutil

**Запуск**

Есть два способа запуска системы:

1. Через главный скрипт (рекомендуется):

python main.py

1. Запуск компонентов по отдельности:

# Запуск прокси-сервера

python simple\_proxy.py

# Запуск веб-интерфейса

python web/app.py

После запуска:

* Прокси-сервер доступен на http://127.0.0.1:8080
* Веб-интерфейс доступен на http://127.0.0.1:5000

**Конфигурация**

**Настройка прокси-сервера**

Основные параметры находятся в файле constants.py:

DEFAULT\_HOST = '127.0.0.1'

DEFAULT\_PORT = 8080

BUFFER\_SIZE = 65536

MAX\_CONNECTIONS = 5

CONNECTION\_TIMEOUT = 10

**Фильтрация доменов**

В файле simple\_proxy.py определен список FILTERED\_DOMAINS для фильтрации служебных доменов:

FILTERED\_DOMAINS = {

'google-analytics.com',

'googletagmanager.com',

# ... другие домены

}

**Файлы данных**

* Sites.txt: лог доступа к доменам
* blocked\_domains.json: список заблокированных доменов

**Безопасность**

**Реализованные механизмы**

1. Таймауты соединений
2. Ограничение максимального числа соединений
3. Фильтрация доменов
4. Безопасная работа с файлами
5. Обработка ошибок соединения

**Рекомендации**

1. Использовать только в доверенной сети
2. Регулярно проверять логи
3. Периодически обновлять список блокировок
4. Настроить файервол
5. Ограничить доступ к веб-интерфейсу

**Мониторинг и обслуживание**

**Логи**

* Расположение: data/Sites.txt
* Формат: [timestamp] Accessing: domain
* Автоматическая ротация не предусмотрена
* Рекомендуется периодическая очистка

**Веб-интерфейс**

* Мониторинг в реальном времени
* Управление блокировками
* Индикация состояния системы
* Автопрокрутка логов

**Устранение неполадок**

**Частые проблемы**

1. Порт уже занят:
   * Проверить, не запущен ли уже сервер
   * Изменить порт в настройках
2. Ошибки соединения:
   * Проверить сетевые настройки
   * Увеличить таймаут
   * Проверить файервол
3. Проблемы с логами:
   * Проверить права доступа к файлам
   * Освободить место на диске
   * Проверить кодировку файлов
4. Веб-интерфейс недоступен:
   * Проверить запуск Flask
   * Проверить порт 5000
   * Перезапустить сервис

**Отладка**

* Все компоненты имеют подробное логирование
* В веб-интерфейсе есть индикация состояния
* Проверка доступа к файлам и портам
* Мониторинг использования ресурсов