

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОНТРОЛЛЕРА WI-FI
AQUARIUS WIRELESS LAN CONTROLLER
Руководство по доступу к программному обеспечению

Листов 23

СОДЕРЖАНИЕ

1. Требования к составу технических и программных средств	3
1.1. Минимальные требования к составу технических средств.....	3
1.2. Минимальный состав программных средств	3
2. Структура программы	4
2.1. Сведения о структуре программы.....	4
2.2. Сведения о составных частях программы	5
2.3. Сведения о связях между составными частями программы	5
2.4. Сведения о связях с другими программами	6
3. Настройка программы.....	7
3.1. Подготовка к настройке	7
3.2. Настройка программы.....	7
4. Проверка программы	17
4.1. Проверка запуска АС	17
4.2. Проверка подключения WTP	18
4.3. Проверка веб-интерфейса	18
4.4. Проверка модульных тестов	19
Приложение 1 Руководство по компиляции и сборке	20
Перечень сокращений	23

1. ТРЕБОВАНИЯ К СОСТАВУ ТЕХНИЧЕСКИХ И ПРОГРАММНЫХ СРЕДСТВ

1.1. Минимальные требования к составу технических средств

Требования к составу и характеристикам технических средств представлены в таблице 1.

Таблица 1 — Требования к составу ТС

ТС	Требования к ТС
Процессор	Архитектура x86-64 или ARM64, тактовая частота не менее 1 ГГц
Объем ОЗУ	Не менее 8 ГБ
Объем ПЗУ	Не менее 1 ГБ свободного места
Сетевой интерфейс	Ethernet, поддержка TUN/TAP (модуль ядра TUN)
ОС	Linux (ядро 5.x и выше), поддержка epoll, UDP-сокеты, TUN/TAP

1.2. Минимальный состав программных средств

Требования к минимальному составу программных средств для работы ПО представлены в таблице 2.

Таблица 2 — Минимальный состав ПО

Компонент	Версия	Назначение
Linux	5.x и выше	Операционная система
OpenSSL	1.1.x / 3.x	DTLS-шифрование, X.509-сертификаты
g++	7.x и выше	Компилятор C++17 (только для сборки)
Node.js	22.x	Сборка Angular frontend (только для сборки)
npm	10.x	Менеджер пакетов Node.js (только для сборки)

2. СТРУКТУРА ПРОГРАММЫ

2.1. Сведения о структуре программы

ПО AQ-WLC состоит из двух самостоятельных исполняемых модулей: AQ-WLC-AC (Access Controller, контроллер беспроводной сети) и AQ-WLC-WTP (Wireless Termination Point, агент точки доступа).

Взаимодействие между AC и WTP осуществляется по протоколу CAPWAP через UDP-порты 5246 (Control Plane) и 5247 (Data Plane). Управление AC выполняется через HTTPS веб-интерфейс (порт 443) или через интерактивную консоль.

Структура ПО представлена на рис. 1.

Структура AQ-WLC

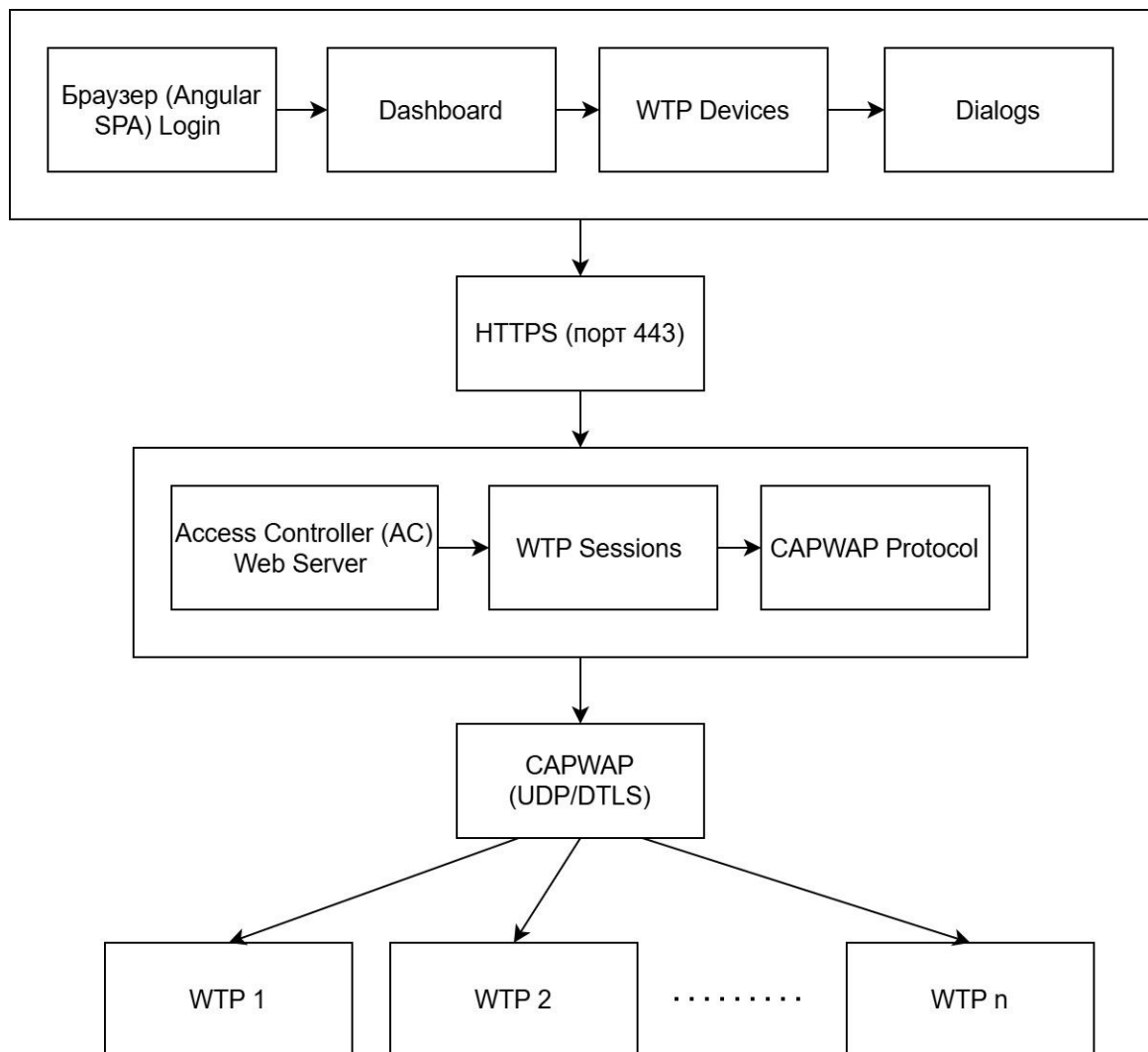


Рисунок 1

2.2. Сведения о составных частях программы

Компонент AQ-WLC-AC (Access Controller) состоит из следующих модулей:

- Dispatcher — принимает входящие UDP-пакеты на портах 5246 и 5247 через epoll, демультимплексирует трафик: направляет Discovery Request на регистрацию, DTLS ClientHello — в ThreadPool для создания сессии, Data Plane пакеты — в существующую сессию;
- ThreadPool / Worker — пул рабочих потоков, каждый реализует Event Loop на базе epoll, обрабатывает события всех назначенных WTP-сессий;
- WtpSession — иерархия классов сессий: базовый Control Plane, Clear Data Channel, DTLS Data Channel; реализует конечный автомат состояний CAPWAP;
- WtpConfigService — удаленная конфигурация WTP через проприетарный протокол VendorSpecificPayload;
- FirmwareManagement / ConfigurationManagement — управление обновлением прошивки и конфигурацией WTP;
- Web Server (Crow) — HTTPS-сервер, предоставляет REST API и обслуживает Angular SPA.

Компонент AQ-WLC-WTP (Wireless Termination Point) состоит из следующих модулей:

- ControlPlaneTask — основной поток управления, реализует конечный автомат состояний CAPWAP со стороны WTP;
- DataPlaneTask — поток передачи данных, обеспечивает инкапсуляцию и декапсуляцию Ethernet-фреймов через TAP-туннель;
- DiscoveryService — сбор адресов AC из нескольких источников с поддержкой приоритетов: Static Configuration, AC Referral, DNS, DHCP, Broadcast;
- ConfigHandler — обработчик входящих запросов конфигурации от AC через VendorSpecificPayload.

Внешняя библиотека capwap-protocol (submodule) реализует 47 типов Message Elements протокола CAPWAP в соответствии с RFC 5415 и RFC 5416.

2.3. Сведения о связях между составными частями программы

Взаимодействие компонентов ПО AQ-WLC происходит следующим образом:

- веб-браузер взаимодействует с AC Web Server по HTTPS (порт 443). Аутентификация осуществляется по TOTP-коду с выдачей JWT-токена в защищенном HttpOnly cookie;
- AC Web Server через WtpSession передает команды конфигурации, обновления прошивки и удаленного выполнения команд на WTP через CAPWAP Control Plane;

- Dispatcher на AC принимает все входящие UDP-пакеты и маршрутизирует их в соответствующие Worker-поток по принципу наименьшей загрузки;
- AC и WTP обмениваются данными через два независимых канала: Control Plane (UDP порт 5246, DTLS) и Data Plane (UDP порт 5247, Clear или DTLS);
- Data Plane реализует TAP-туннель: Ethernet-фреймы читаются из TapDevice WTP, инкапсулируются в CAPWAP UDP-пакеты, передаются на AC, где деинкапсулируются и записываются в TapDevice AC.

2.4. Сведения о связях с другими программами

ПО AQ-WLC взаимодействует со следующими внешними компонентами:

- OpenSSL — используется для реализации DTLS-шифрования каналов управления и данных, а также для генерации и проверки X.509-сертификатов. Единый контекст SSL_CTX разделяется между всеми сессиями;
- Linux TUN/TAP subsystem (модуль ядра TUN) — используется для создания виртуальных сетевых интерфейсов TAP для туннелирования Ethernet-фреймов;
- Linux epoll — используется для асинхронной обработки событий сокетов в Dispatcher и Worker;
- Angular SPA (frontend) — статические файлы Angular-приложения обслуживаются встроенным Crow HTTP-сервером.

3. НАСТРОЙКА ПРОГРАММЫ

3.1. Подготовка к настройке

Перед запуском ПО необходимо выполнить сборку из исходного кода и сгенерировать сертификаты. Для сборки необходим Linux-хост с компилятором g++ (C++17), библиотекой OpenSSL (libssl-dev), утилитой ccache (опционально), Node.js 22.x и npm 10.x.

Сборка выполняется следующими командами:

```
make certs          # Генерация сертификатов (первый запуск)
make all            # Сборка всех компонентов
```

Для сборки отдельных компонентов используются команды:

```
make build_AC       # Только Access Controller
make build_WTP      # Только Wireless Termination Point
make build_tests    # Только тесты
```

После сборки исполняемые файлы размещаются в директориях output/ac/ и output/wtp/. Генерируемые сертификаты помещаются в директорию certs/.

Для корректной работы Data Plane необходимо убедиться в наличии модуля ядра TUN и достаточных буферов сокетов:

```
# Проверка системных лимитов сокетных буферов
sysctl net.core.rmem_max net.core.wmem_max

# При необходимости увеличить (должно быть >= 8 МБ):
sudo sysctl -w net.core.rmem_max=8388608
sudo sysctl -w net.core.wmem_max=8388608
```

3.2. Настройка программы

Настройка AC выполняется через параметры командной строки и файлы конфигурации ac/ac_config.cpp и ac/ac_constants.cpp. Запуск AC:

```
sudo output/ac/AQ-WLC-AC <ip_адрес_AC>
```

Параметры командной строки AC представлены в таблице 3.

Таблица 3 — Параметры командной строки AC

Параметр	Описание
ac_ip	IP-адрес AC (обязательный)
--control-port=PORT	CAPWAP Control Port (по умолчанию: 5246)
--data-port=PORT	CAPWAP Data Port (по умолчанию: 5247)
--data-clear	Использовать Clear Text Data Channel
--data-dtls	Использовать DTLS Data Channel
--daemon	Режим демона (без интерактивного stdin)

Параметры командной строки WTP представлены в таблице 4.

Таблица 4 — Параметры командной строки WTP

Параметр	Описание
ac_ip	IP-адрес AC (опциональный; без него — Discovery)
--ac-control-port=PORT	AC Control Port (по умолчанию: 5246)
--ac-data-port=PORT	AC Data Port (по умолчанию: 5247)
--allow-commands=cmd1,cmd2	Режим allowlist — разрешены только указанные команды
--block-commands=cmd1,cmd2	Режим blocklist — запрещены только указанные команды
--daemon	Режим демона (без интерактивного stdin)
--model-number=STRING	Модель WTP
--serial-number=STRING	Серийный номер WTP
--software-version=STRING	Версия программного обеспечения WTP

3.2.2. Настройка аутентификации (TOTP)

После запуска AC необходимо настроить TOTP-секрет для доступа к веб-интерфейсу. В интерактивной консоли AC выполнить:

```
totp-r      # Генерация нового TOTP-секрета
totp-w <BASE32_SECRET>  # Сохранение секрета
```

Команда `totp-r` выводит Base32-секрет и otpauth URI. Полученный секрет необходимо занести в authenticator-приложение (Google Authenticator, Authy или аналог) путем сканирования QR-кода или ввода Base32-строки вручную.

3.2.3. Настройка веб-интерфейса

Веб-интерфейс доступен по адресу `https://<ip_адрес_АС>/` после запуска АС. Используется самоподписанный сертификат, поэтому браузер может отображать предупреждение безопасности.

Вход в систему выполняется вводом 6-значного TOTP-кода из authenticator-приложения (рисунок 2).

Страница входа в веб-интерфейс

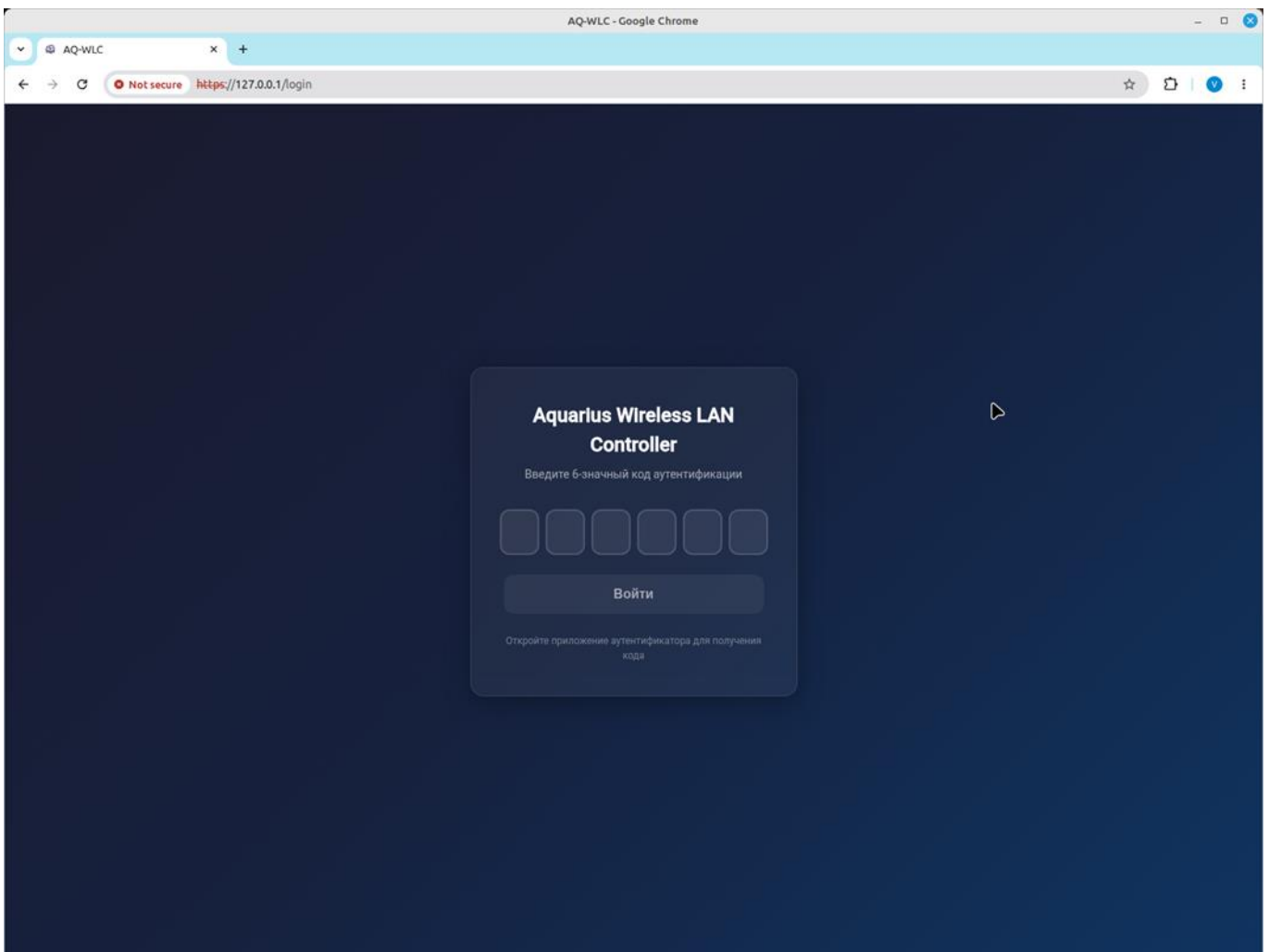


Рисунок 2

После успешной аутентификации открывается главная страница управления WTP-устройствами (рисунок 3).

Список WTP-устройств

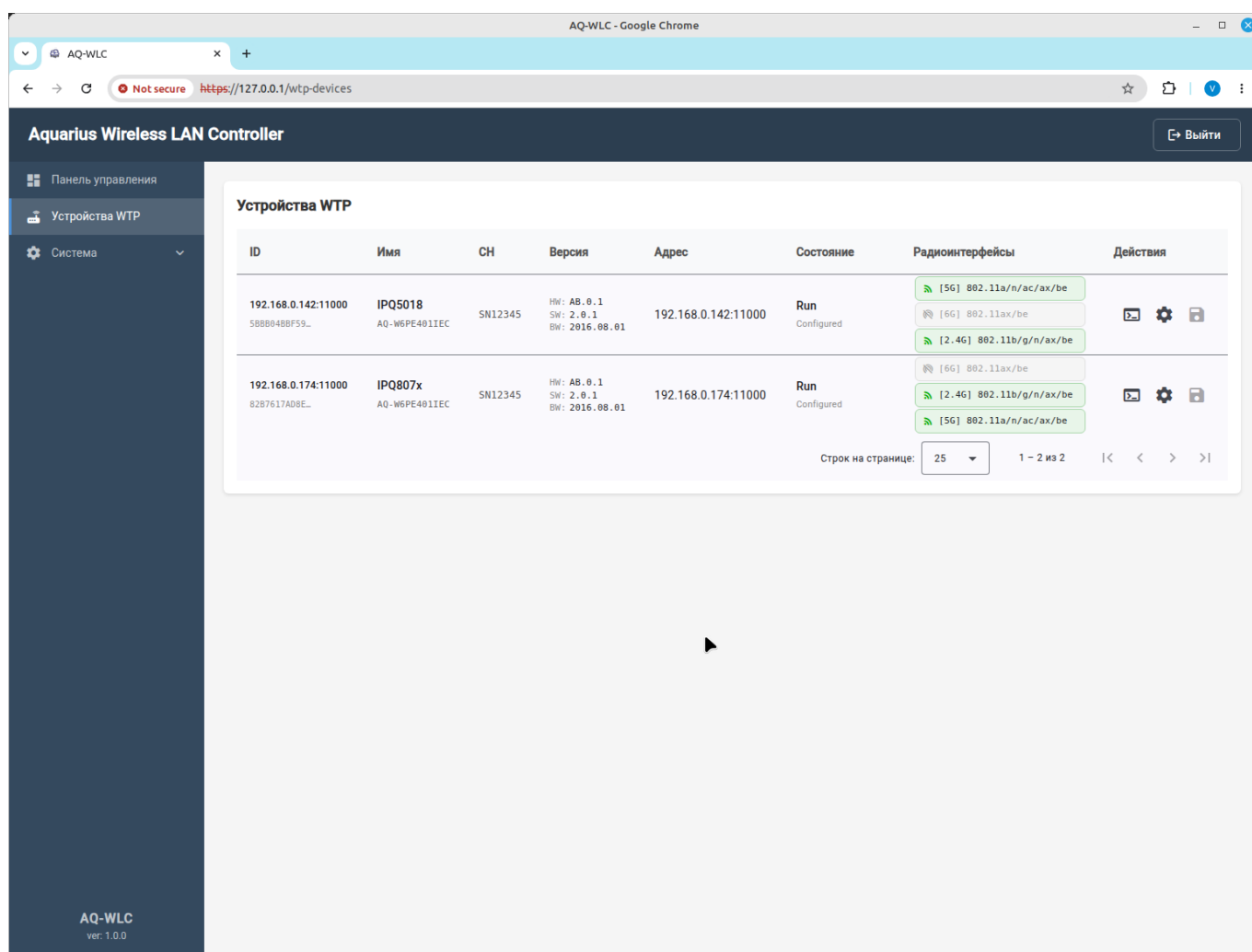


Рисунок 3

3.2.4. Настройка радиомодулей WTP

Для настройки радиомодуля необходимо нажать на его индикатор в колонке «Радиоинтерфейсы» в таблице «Устройства WTP». Откроется диалог «Конфигурация радио» (рисунок 4).

Диалог настройки радиомодуля

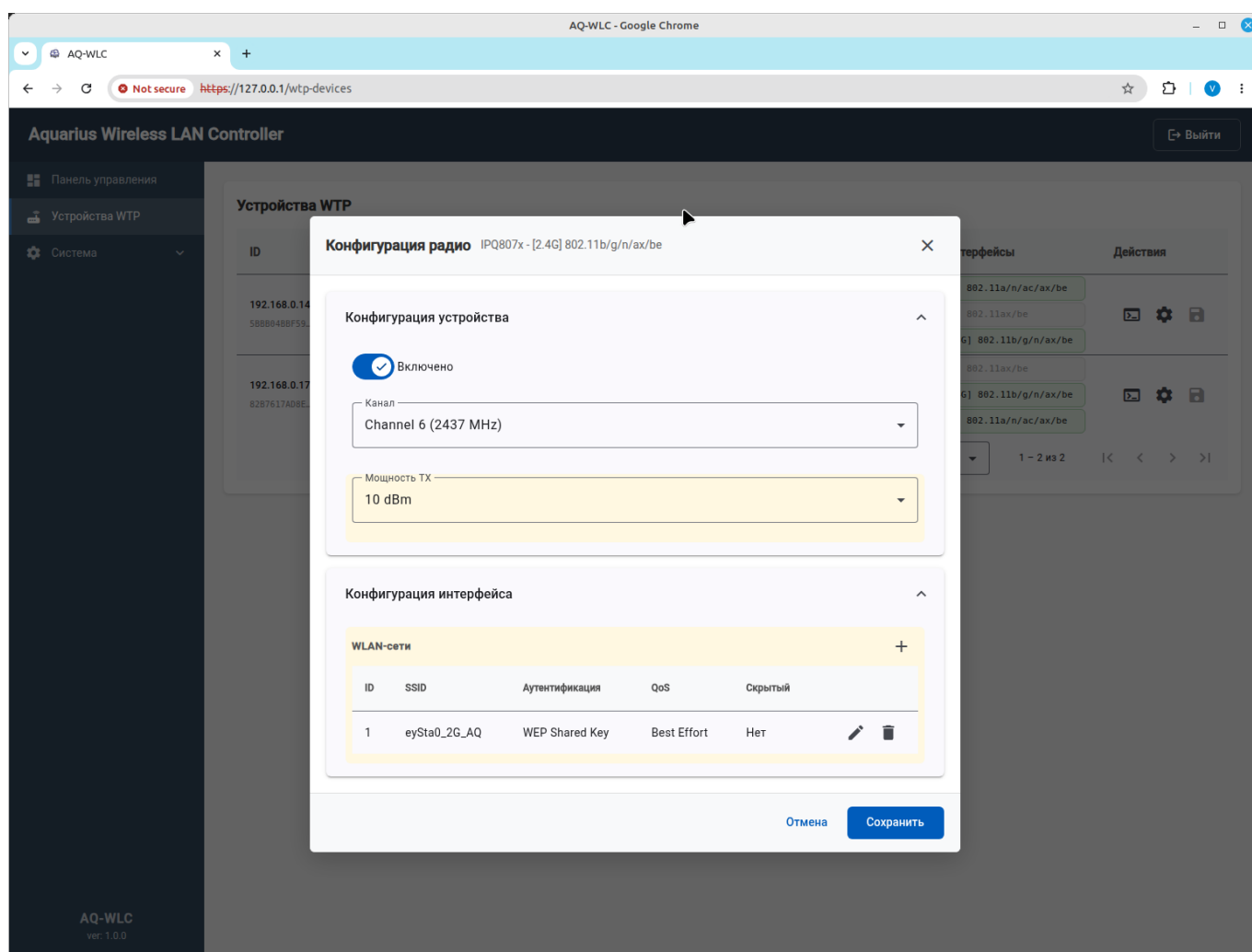


Рисунок 4

В диалоге доступны параметры, указанные в таблице 5.

Таблица 5 — Параметры настройки радиомодуля

Группа	Параметр	Описание
Конфигурация устройства	Включено	Включение/выключение радиомодуля
Конфигурация устройства	Канал	Выбор канала (2,4 ГГц: 1–13, 5 ГГц: 36–165)
Конфигурация устройства	Мощность TX	Мощность передатчика
Конфигурация интерфейса	ID	Режим работы: Access Point, Client, Ad-Hoc
Конфигурация интерфейса	SSID	Имя сети (до 32 символов)

Группа	Параметр	Описание
Конфигурация интерфейса	BSSID	MAC-адрес точки доступа
Конфигурация интерфейса	Encryption	Тип шифрования: WPA2, WPA3, WPA2/WPA3 Mixed
Конфигурация интерфейса	Key	Ключ шифрования

Измененные параметры подсвечиваются желтым цветом. После внесения изменений необходимо нажать кнопку «Сохранить».

3.2.5. Настройка конфигурации WTP

Для редактирования конфигурации WTP необходимо нажать кнопку «Настройки» в колонке «Действия». Откроется диалог «Конфигурация WTP» с группами параметров (рисунок 5).

Диалог конфигурации WTP

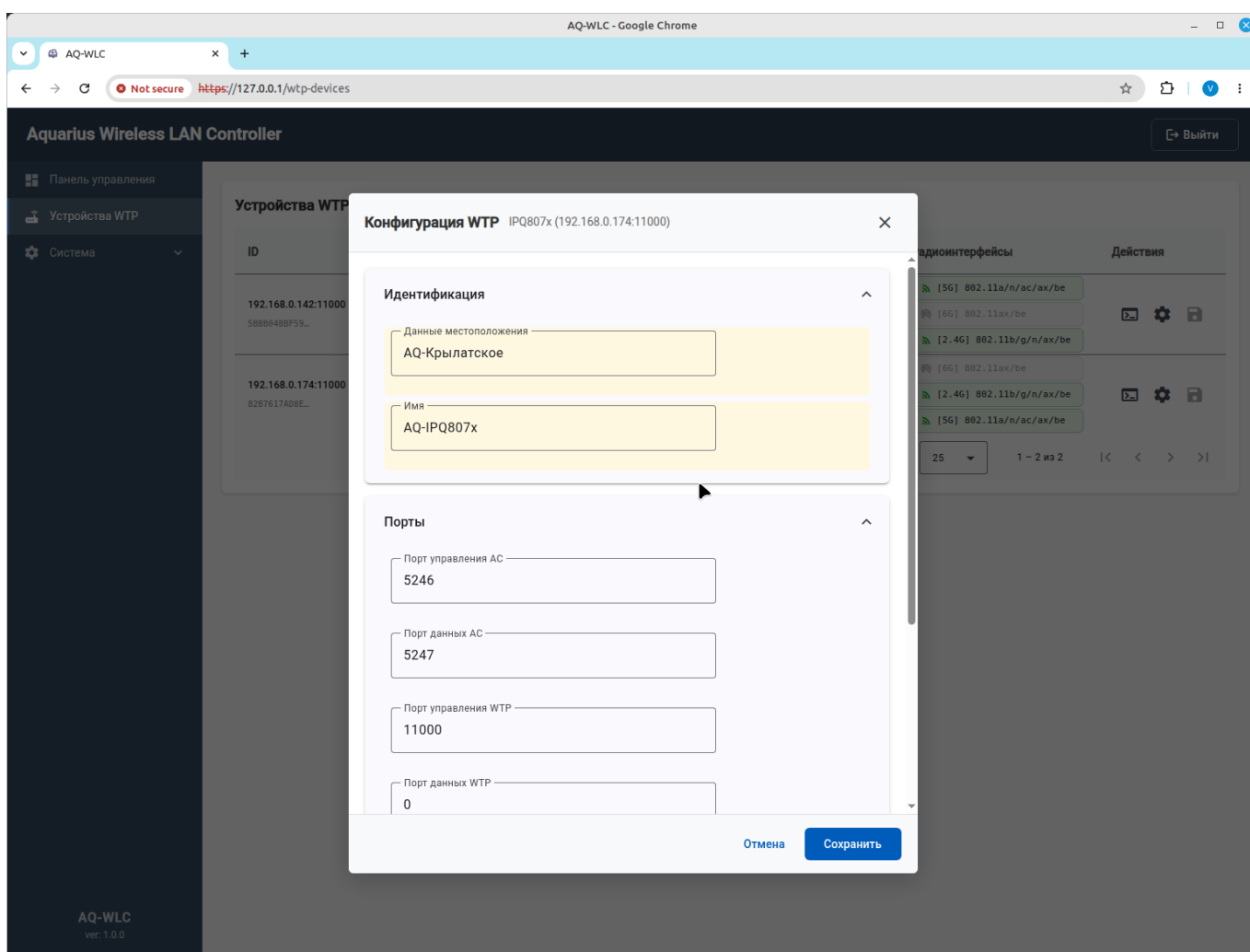


Рисунок 5

Группы параметров конфигурации WTP

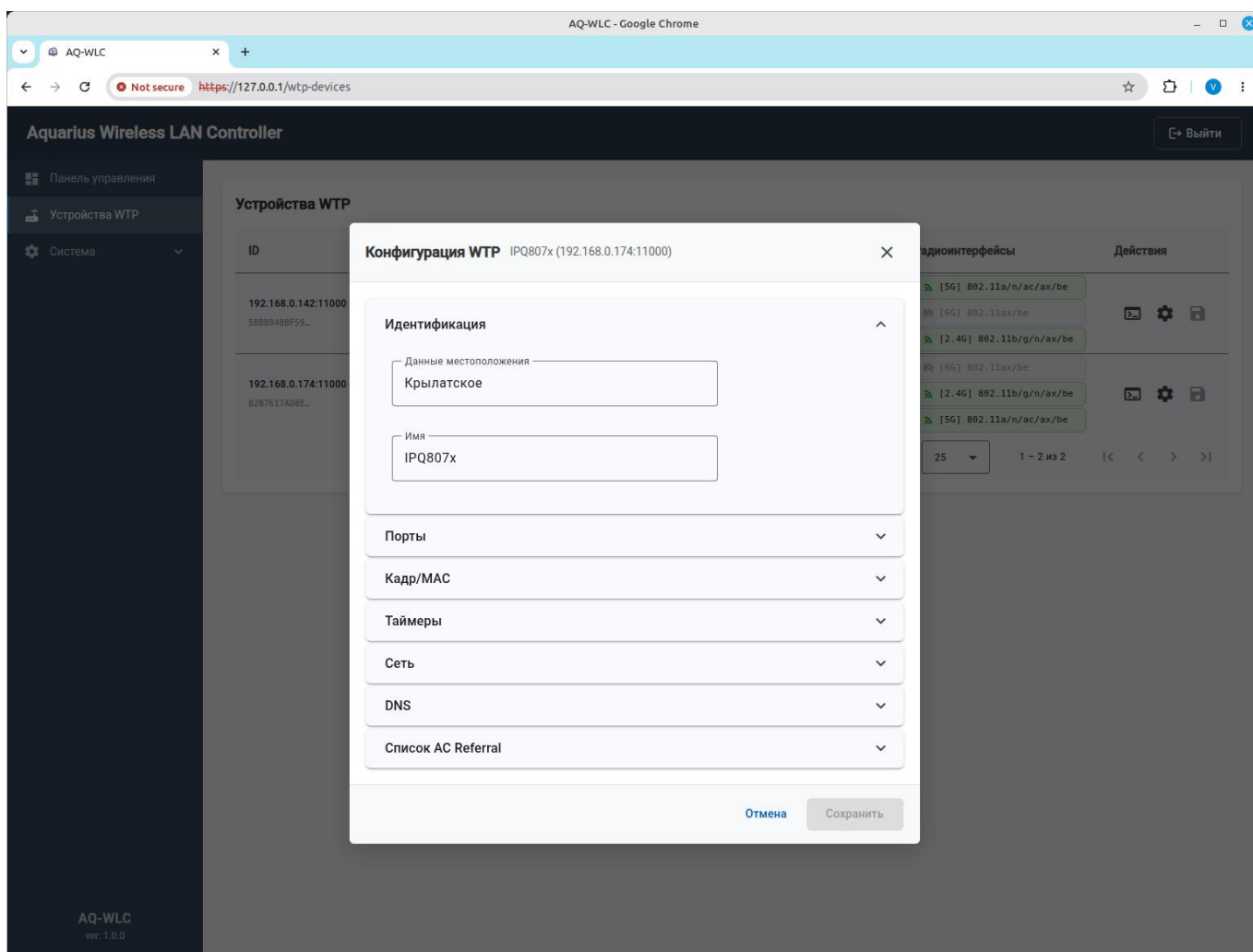


Рисунок 6

Группы параметров конфигурации WTP представлены в таблице 6.

Таблица 6 — Группы параметров конфигурации WTP

Группа	Параметры
Идентификация	Имя (имя устройства), Данные местоположения (местоположение)
Порты	Порт управления AC, Порт данных AC, Порт управления WTP, Порт данных WTP
Кадр/MAC	Локальное переключение, Кадры Native IEEE 802.3, Беспроводные кадры Native, Тип MAC, Поддержка ECN

Группа	Параметры
Таймеры	Таймер статистики (с), Интервал повтора (мс), Макс. повторов, Keep Alive канала данных (мс), Dead Interval канала данных (мс)
Сеть	Длина TAP MTU, Размер буфера сокета, Broadcast-настройки
DNS	Список DNS-серверов (до 4 записей)
Список AC Referral	Список AC-серверов для обнаружений (до 8 записей)

Для сохранения изменений в постоянную память WTP необходимо нажать кнопку «Применить изменения» (кнопка в колонке «Действия» становится активной при наличии несохраненных изменений).

3.2.6. Удаленный терминал WTP

Для выполнения shell-команд на WTP необходимо нажать кнопку «Терминал» в колонке «Действия». Откроется диалог «Терминал» (рисунок 7).

Интерактивный терминал WTP

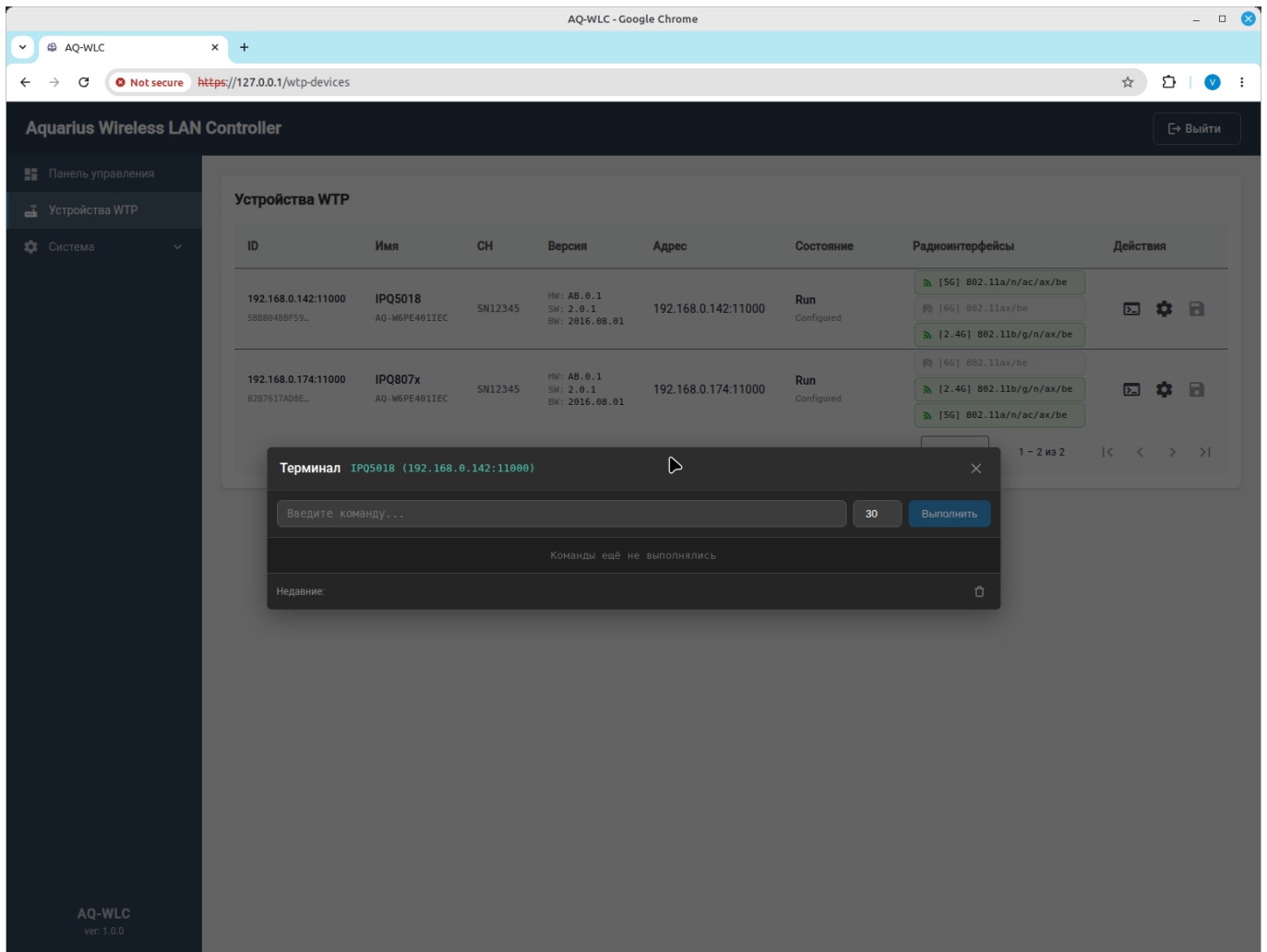


Рисунок 7

Выполнение команд через терминал

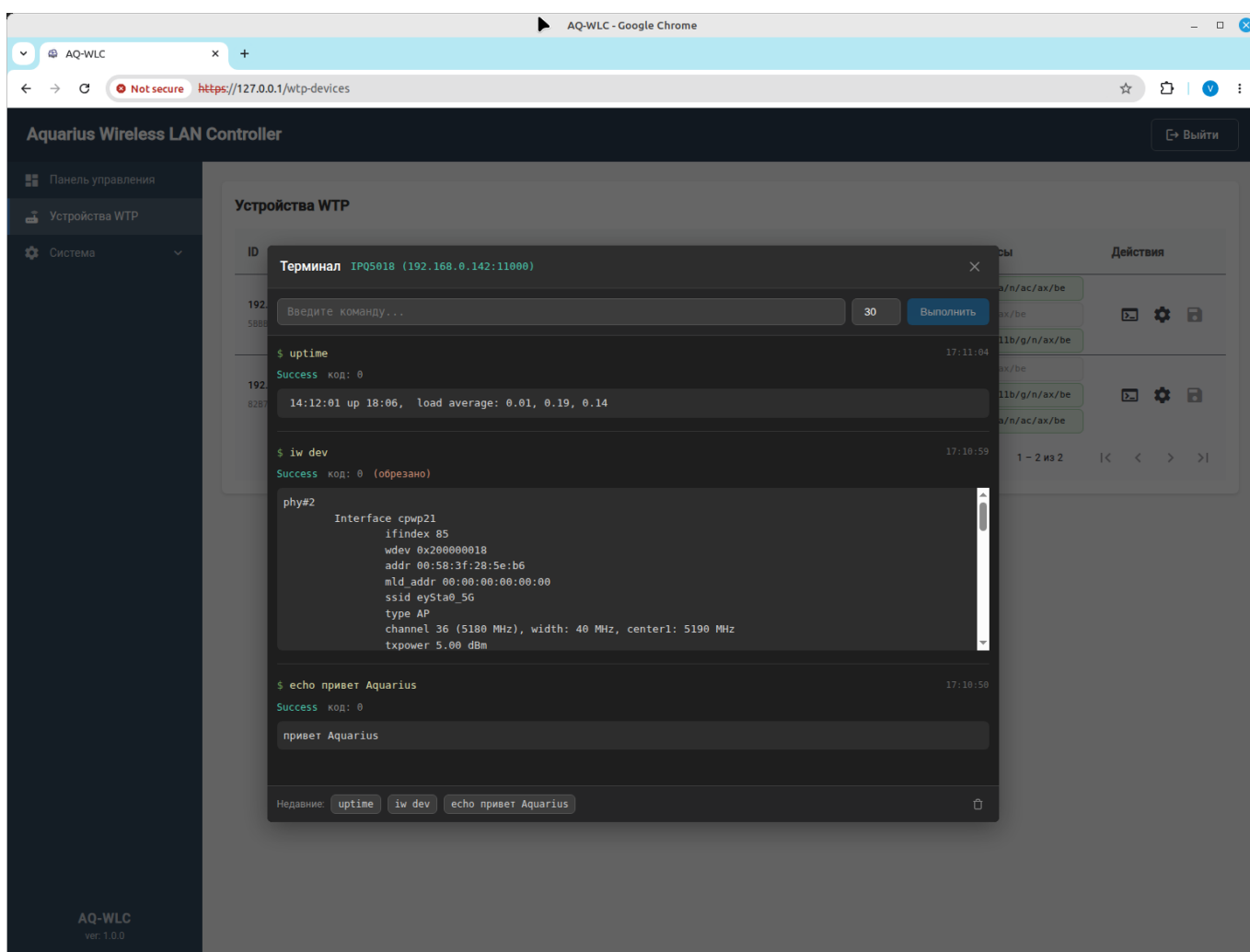


Рисунок 8

В поле ввода вводится команда, в числовом поле указывается таймаут в секундах (до 10 800 с, то есть до 3 часов), затем нажимается кнопка «Выполнить». Вывод команды (stdout и stderr) отображается в области результатов. Недавние команды доступны в строке «Недавние».

Список разрешенных или запрещенных команд задается при запуске WTP через параметры `--allow-commands` или `--block-commands`.

4. ПРОВЕРКА ПРОГРАММЫ

После запуска и настройки ПО необходимо выполнить следующие проверки.

4.1. Проверка запуска АС

Запустить АС и убедиться в отсутствии ошибок в выводе консоли. Индикатором успешного запуска является сообщение вида:

```
I (21 17:01:46.618) ----- Start AC-----
Enter command ('s' or EOF to stop):

I (21 17:01:46.618) TOTP secret loaded successfully
I (21 17:01:46.618) AC: (wrk_#0) Create stop_event_fd: 4
I (21 17:01:46.618) AC: (wrk_#0) Create add_event_fd: 5
I (21 17:01:46.618) AC: (wrk_#0) created epoll_fd: 3,
add_event_fd: 5, stop_event_fd: 4
...
I (21 17:01:46.618) AC: (wrk_#7) Create add_event_fd: 26
I (21 17:01:46.618) AC: (wrk_#7) created epoll_fd: 24,
add_event_fd: 26, stop_event_fd: 25
I (21 17:01:46.618) AC ThreadPool: Create 8 workers
I (21 17:01:46.621) ClientSocket: sock_fd: 27, open socket,
buffer_size: 8388608
I (21 17:01:46.621) ClientSocket: sock_fd: 28, open socket,
buffer_size: 8388608
I (21 17:01:46.621) AC Dispatcher: Create stop_event_fd: 30
I (21 17:01:46.621) AC Dispatcher: control_socket fd: 27
I (21 17:01:46.621) AC Dispatcher: data_socket fd: 28
I (21 17:01:46.621) Set ReceiveMulticast sock_fd: 27, addr:
224.0.1.140
I (21 17:01:46.621) AC Dispatcher: created epoll_fd: 29,
stop_event_fd: 30
I (21 17:01:46.622) AC: (wrk_#1) Start listen events: 64
...
I (21 17:01:46.622) AC: (wrk_#5) Start listen events: 64
```

```

I (21 17:01:46.622) AC ThreadPool: Started 8 workers
I (21 17:01:46.622) AC Dispatcher: Start listen events: 8
I (21 17:01:46.622) ClientSocket: sock_fd: 27, bind to:
0.0.0.0:5246
I (21 17:01:46.622) ClientSocket: sock_fd: 28, bind to:
0.0.0.0:5247
I (21 17:01:46.622) AC: (wrk_#4) Start listen events: 64
...
I (21 17:01:46.622) AC: (wrk_#6) Start listen events: 64
I (21 17:01:46.622) AC: WebServer: starting HTTPS on port
443
I (21 17:01:46.622) AC: LoginController: registering routes
I (21 17:01:46.622) AC: WtpController: registering routes
I (21 17:01:46.622) AC: WtpConfigController: registering
routes
I (21 17:01:46.622) AC: MainController: registering routes,
static_dir: ac/web/browser

```

4.2. Проверка подключения WTP

Запустить WTP и убедиться, что устройство успешно прошло процедуру Discovery, DTLS Handshake и перешло в состояние Run. В веб-интерфейсе в разделе WTP Devices устройство должно появиться со статусом Online и состоянием Run.

4.3. Проверка веб-интерфейса

Открыть браузер и перейти по адресу `https://<ip_адрес_AC>/`. Убедиться в успешном открытии страницы входа. Ввести корректный TOTP-код и убедиться в успешном входе и отображении таблицы WTP Devices.

Проверить ввод некорректного кода:

```

# Ввести заведомо неверный 6-значный код
# Ожидаемый результат: сообщение об ошибке аутентификации
Invalid code. Please try again.

```

При повторных неудачных попытках должна активироваться эскалирующая блокировка (задержка 1 с, 2 с, 4 с и далее).

4.4. Проверка модульных тестов

Для комплексной автоматизированной проверки ПО необходимо выполнить набор тестов:

```
sudo output/tests/unit_tests/aq-wlc-tests
```

Тесты охватывают: сериализацию/десериализацию всех типов CAPWAP-сообщений, работу ThreadPool и Worker, DiscoveryService (DNS, приоритеты, дедупликация), AC Dispatcher (DTLS Protection, регистрация Discovery), интеграционные сценарии AC + WTP (Discovery — Join — Configuration — Run), параллельный Discovery с балансировкой нагрузки, Firmware Download в состоянии Run, Data Plane Keep-Alive, передачу Ethernet-фреймов через TAP-туннель в режимах Clear и DTLS.

Индикатором успешного прохождения всех тестов является вывод:

```
OK (N tests, M checks)
```

РУКОВОДСТВО ПО КОМПИЛЯЦИИ И СБОРКЕ

1.1. Требования

Для компиляции AQ-WLC необходимо:

- Linux (Ubuntu 20.04 LTS и выше или совместимый дистрибутив);
- g++ с поддержкой C++17 (g++ 9.x и выше);
- OpenSSL (libssl-dev);
- libasio-dev (или libboost-all-dev);
- make;
- ccache (опционально, для ускорения пересборки).

Установка зависимостей на Ubuntu:

```
sudo apt-get install g++ libssl-dev libasio-dev make ccache
```

Для кросс-компиляции WTP под aarch64 (Raspberry Pi 3, платформы AQ-W6):

```
sudo apt-get install gcc-aarch64-linux-gnu g++-aarch64-  
linux-gnu
```

1.2. Сборка

```
# Клонирование репозитория с submodule-ами  
git clone --recurse-submodules <URL> aq-wlc  
cd aq-wlc  
  
# Генерация сертификатов (первый запуск)  
make certs  
  
# Сборка всех компонентов (для x86_64)  
make all
```

Отдельные компоненты:

```
make build_AC      # Только AC  
make build_WTP     # Только WTP (x86_64)  
make build_tests   # Только тесты  
make build_benchmarks # Только бенчмарки
```

1.3. Сборка прошивки для точек доступа

Сборка прошивок OpenWrt (AQW6) выполняется в Docker-контейнере. Требования: Docker, SSH-ключ ~/.ssh/id_ed25519, исходный код AQW6 в директории aqw6/, SPF-пакет qca-networking-2022-spf-12-2_qca_oem/.

Для FTG-Enterprise-E W6PE401E (IPQ8072A / IPQ807x):

```
# Применяет патчи и запускает Docker-контейнер
Make

# Внутри Docker shell:
time make -j32 oldconfig
time make -j32

# Создание финального образа (в Docker shell, на уровень
выше) :
cd ..
./single_image.sh aqw6/
```

Результирующие образы:

Файл	Назначение
norplusnand-4k-ipq807x-apps.img	RootFS + Wi-Fi FW, обновление через sysupgrade
norplusnand-4k-ipq807x-single.img	Полный образ, только через U-Boot

Очистка:

```
sudo make clean
```

Для AQ-W6PE100I (IPQ5018):

```
# Применяет патчи и запускает Docker-контейнер
Make

# Внутри Docker shell:
time make -j32 oldconfig
time make -j32
```

```
# Создание финального образа (в Docker shell, на уровень
выше) :
```

```
cd ..
```

```
./single_image.sh aqw6/
```

Результирующие образы:

Файл	Назначение
norplusnand-ipq5018-apps.img	RootFS + Wi-Fi FW, обновление через sysupgrade
norplusnand-ipq5018-single.img	Полный образ, только через U-Boot

Очистка:

```
sudo make clean
```

Примечание. Образы с суффиксом `-single.img` содержат SBL, TZ, U-Boot и CDT. Они предназначены только для прошивки через U-Boot. Использование через sysupgrade не допускается. На IPQ5018 дополнительно не использовать образы с `-4k-` в названии через sysupgrade — устройство перестанет загружаться (несовместимость UBI vid_hdr_offset 4K vs 2K).

1.4. Расположение собранных файлов

```
output/
|-- ac/aq-wlc-AC                # Бинарный файл AC
|-- wtp/aq-wlc-WTP              # Бинарный файл WTP (x86_64)
|-- wtp-aarch64/aq-wlc-WTP      # Бинарный файл WTP (aarch64)
|-- tests/unit_tests/aq-wlc-tests
`-- tests/benchmarks/aq-wlc-benchmarks
```

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ОЗУ	–	оперативное запоминающее устройство
ПЗУ	–	постоянное запоминающее устройство
ПО	–	программное обеспечение
ТС	–	технические средства
АС	–	access controller, контроллер беспроводной сети
API	–	application programming interface, программный интерфейс приложения
AQ-WLC	–	программное обеспечение Aquarius Wireless LAN Controller RU.АМПР.00202-01
CAPWAP	–	control and provisioning of wireless access points, протокол управления точками доступа
DHCP	–	dynamic host configuration protocol, это сетевой протокол
DNS	–	domain name system, система для получения информации о доменах
DTLS	–	datagram transport layer security, защита транспортного уровня для датаграмм
HTTP	–	hypertext transfer protocol, сетевой протокол для обмена данными
HTTPS	–	hypertext transfer protocol secure, защищенная версия протокола HTTP
JWT	–	JSON web token, токен сессии в формате json
REST	–	representational state transfer, архитектурный стиль взаимодействия компонентов распределенного приложения в сети
RFC	–	request for comments, документы стандартов ietf
SPA	–	single page application, одностраничное веб-приложение
TAP	–	network TAP, виртуальный сетевой интерфейс уровня 2
TOTP	–	time-based one-time password, одноразовый пароль на основе времени
TUN	–	network TUN, виртуальный сетевой интерфейс IP уровня
UDP	–	user datagram protocol, протокол передачи датаграмм
WTP	–	wireless termination point, беспроводная точка доступа