

Амибиг

Руководство администратора

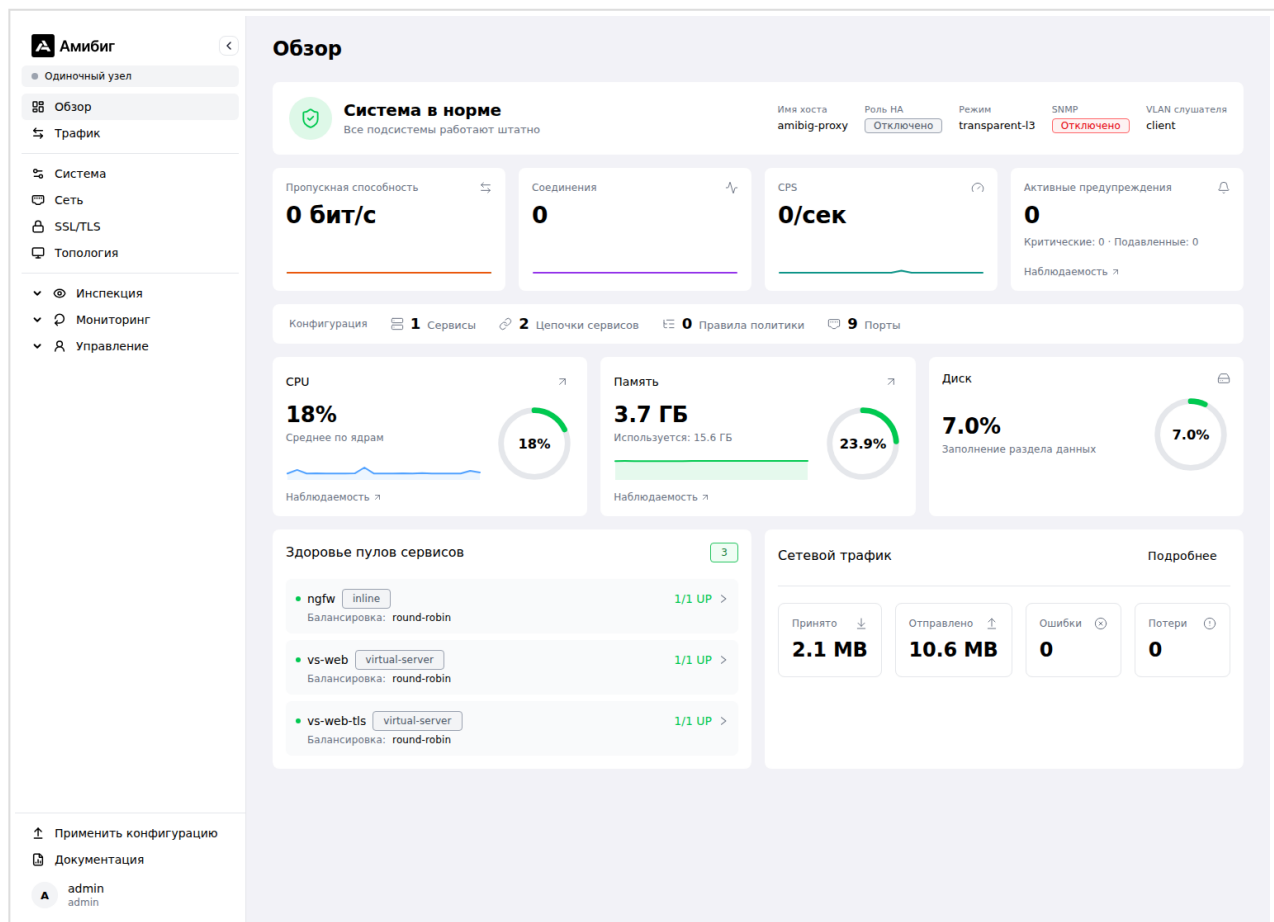
Содержание

Амибиг: руководство администратора	4
Перед началом работы.....	4
Применение конфигурации	5
Разделы руководства	5
Условные обозначения	7
Начало работы	8
Обзор Амибиг	8
Архитектура Амибиг.....	13
Системные требования	19
Установка	23
Быстрый старт	26
Сеть.....	40
Физические интерфейсы.....	40
VLAN	45
Адреса Self-IP.....	48
Статические маршруты.....	51
Агрегирование каналов (LACP)	56
Трансляция адреса источника в исходящих соединениях (SNAT)	65
Управление TCP MSS и MTU	68
SSL/TLS	72
Аппаратное ускорение TLS (Intel QAT).....	72
Обзор SSL/TLS.....	73
Управление сертификатами	79
Наборы шифров (Cipher Rules)	86
Доверенные УЦ	92
Клиентский SSL-профиль (client-profile)	96
Серверный SSL-профиль (server-profile).....	107
ГОСТ TLS	113
Сервисы	116
Обзор сервисов	116
Inline-сервисы (NGFW).....	121
ICAP-сервисы (DLP)	132
Пулы сервисов.....	141
Мониторинг доступности	148
Виртуальные серверы	157
GSLB (DNS-балансировка)	162
Цепочки сервисов	169
Обзор цепочек сервисов.....	169
Политики	177
Обзор политик.....	177
Правила политик	182
Критерии отбора трафика	193
Категории URL.....	200
Классы ограничения скорости.....	203
Идентификация	207
Идентификация пользователей	207
Каталог LDAP.....	212

Источники соответствий	217
Портал аутентификации	222
aRules	226
Обзор aRules	226
Справочник команд aRules	230
Справочник событий aRules	246
Примеры aRules	255
Топология	262
Прозрачная топология	262
Шаблоны развёртывания	265
Высокая доступность	269
Обзор высокой доступности	269
Настройка отказоустойчивого кластера	274
Состояние соединений при переключении	286
Виртуальные IP-адреса (VIP)	290
Управление	295
Веб-интерфейс	295
REST API	302
Управление пользователями	313
Идентификация устройства и лицензирование	320
Мониторинг по SNMP и trap-сообщения	326
Наблюдаемость: предупреждения, аналитика и экспорт	330
Телеметрия и метрики	337
Производительность	340
Резервное копирование и восстановление	344
Обновление системы	350
История конфигурации	359
Системные параметры	361
Миграция с F5 (импорт UCS)	364
Диагностика	368
Объяснение потока	368
Захват пакетов	371
Журнал событий и его экспорт	375
Мониторинг трафика	380
Устранение неполадок	383
Типовые проблемы и решения	383
Диагностика прохождения трафика	394
Справочник	400
Справочник конфигурационной схемы	400
YANG-модель конфигурации	428
Справочник командной строки администратора	434
Артефакты релиза	436
Системные лимиты	439
Коды сообщений	442

Амибиг: руководство администратора

Руководство предназначено для администраторов и инженеров внедрения Амибиг. В нём описаны установка устройства, настройка сети и TLS-инспекции, подключение внешних средств защиты, эксплуатация отказоустойчивой пары и диагностика трафика. Состав встроенной справки соответствует версии ПО, установленной на устройстве.



Перед началом работы

Выберите маршрут в зависимости от задачи:

1. Для новой установки проверьте системные требования, выполните установку, затем перейдите к быстрому старту.
2. Для переноса конфигурации F5 BIG-IP создайте резервную копию Амибиг и следуйте разделу Миграция с F5.
3. Для изменения рабочей конфигурации сначала ознакомьтесь с порядком применения и подтверждения изменений.

4. Для разбора неполадок начните с экрана Трафик, затем используйте Объяснение потока, захват трафика и журнал событий.

Применение конфигурации

Изменения, сохранённые в веб-интерфейсе или через REST API, сначала попадают в подготовленную (`candidate`) конфигурацию. До применения они не влияют на обработку трафика. Команда `Применить конфигурацию` переносит изменения в действующую (`running`) конфигурацию и запускает таймер подтверждения. Если администратор не подтвердит новую конфигурацию, Амибиг автоматически вернёт предыдущее состояние.

Перед применением проверяйте список изменений и предупреждения о перезапуске dataplane. После применения убедитесь, что интерфейс управления доступен, сервисы находятся в ожидаемом состоянии, а тестовый трафик обрабатывается нужным правилом.

Разделы руководства

Руководство организовано по этапам эксплуатации: от установки и настройки обработки трафика до сопровождения, диагностики и справочных сведений.

Установка и первичная настройка

Начните с этих разделов при вводе нового устройства в эксплуатацию:

Раздел	Содержание
Обзор продукта	Назначение Амибиг, сценарии применения и путь трафика
Архитектура	Dataplane, система управления и основные компоненты
Системные требования	Требования к серверу, сетевым интерфейсам и браузеру
Установка	Загрузочный носитель, разметка диска и первоначальная настройка
Быстрый старт	Минимальная конфигурация для первого потока трафика

Обработка трафика

Разделы расположены в порядке от сетевой конфигурации к правилам обработки трафика и отказоустойчивости:

Раздел	Содержание
Сеть	Физические интерфейсы, агрегированные каналы, VLAN, Self-IP и маршруты
SSL/TLS	Сертификаты, доверенные УЦ, наборы шифров и профили TLS
Сервисы	Inline- и ICAP-сервисы, пулы и мониторы доступности
Цепочки сервисов	Порядок передачи трафика внешним средствам защиты
Политики	Условия правил, действия и правило по умолчанию
Категории URL	Собственные списки доменов для правил политики
Идентификация	Пользователи и группы LDAP в условиях правил
aRules	Tcl-совместимые правила программируемой обработки трафика
Топология	Точка входа трафика и прозрачные режимы работы
Высокая доступность	Активный и резервный узлы, синхронизация и переключение

Эксплуатация

Используйте эти разделы для повседневного администрирования устройства:

Раздел	Содержание
Веб-интерфейс	Навигация, подготовленная и действующая конфигурации
REST API	Программное управление и коды ответов
Пользователи	Учётные записи, роли, права и список доступа к управлению
Лицензирование	Идентификатор устройства и установка лицензии
Наблюдаемость	Оповещения, история показателей и передача событий в SIEM
Резервное копирование	Создание и восстановление резервных копий

Раздел	Содержание
Обновление системы	Обновление A/B, откат и сброс к заводским настройкам
История конфигурации	Сравнение и восстановление сохранённых состояний
Системные параметры	Имя узла, время, DNS, NTP и адрес управления

Диагностика и справочные сведения

При диагностике сначала найдите симптом, затем сверяйте фактическую конфигурацию с ограничениями и справочными значениями:

Раздел	Содержание
Устранение неполадок	Симптомы, вероятные причины и порядок проверки
Справочник конфигурации	Структура <code>/config/amibig.json</code> и параметры секций
YANG-модель	Типы данных, ссылки и преобразование в JSON
Системные лимиты	Допустимые значения и ограничения текущей версии
Коды сообщений	Идентификаторы событий <code>AMB-NNNN</code>
Артефакты релиза	Состав автономной поставки и контрольные суммы

Условные обозначения

Моноширинным текстом выделены команды, пути, имена полей конфигурации и адреса REST API. Точные названия экранов, полей и кнопок веб-интерфейса также приводятся моноширинным текстом.

Начало работы

В этом разделе администратор знакомится с назначением и архитектурой Амибиг, проверяет требования к оборудованию и выполняет первоначальную настройку. Начните с обзора продукта, затем подготовьте оборудование и пройдите процедуру быстрого старта.

Материалы рассчитаны на первое развёртывание и демонстрационный стенд. Перед вводом устройства в промышленную эксплуатацию дополнительно настройте сеть, TLS-инспекцию, сервисные цепочки, учётные записи и резервное копирование в соответствующих разделах руководства.

Обзор Амибиг

Амибиг — система централизованной TLS-инспекции и построения сервисных цепочек, разработанная ООО «АМИКОН». Устройство устанавливается в разрыв сетевого трафика и передаёт расшифрованные данные средствам защиты информации: NGFW, DLP, IDS/IPS и антивирусам.

Большая часть корпоративного веб-трафика передаётся по TLS. Чтобы анализировать содержимое таких соединений и применять политики безопасности, средствам защиты требуется доступ к расшифрованному трафику.

При обработке соединения система выполняет следующие действия:

1. Принимает соединение в прозрачном режиме и определяет исходные IP-адреса источника и назначения.
2. Завершает клиентское TLS-соединение и расшифровывает данные.
3. Передаёт данные по сервисной цепочке в NGFW, DLP и другие средства защиты.
4. Устанавливает origin-соединение, повторно шифрует проверенные данные и передаёт их целевому серверу.

Для соединений с действием `intercept` клиент получает сертификат целевого сервера, подписанный сертификатом УЦ организации. Чтобы клиент доверял этому сертификату, сертификат УЦ Амибиг необходимо заранее установить в доверенное хранилище клиентских устройств. Соединения с действиями `bypass` и `passthrough` проходят без подмены сертификата.

Амибиг инспектирует соединения TLS 1.0–1.3 и ГОСТ-TLS. Для ГОСТ-TLS устройство принимает соединение клиента, устанавливает отдельное соединение с сервером и создаёт сертификат от собственного ГОСТ-УЦ (см. ГОСТ TLS).

Ориентир пропускной способности TLS-инспекции — не менее 20 Гбит/с. Базовый порог производительности — 1 млн новых соединений в минуту.

Сценарии использования

Амибиг применяется для инспекции корпоративного трафика, оркестрации средств защиты и замены существующих решений SSL Orchestrator.

Инспекция корпоративного трафика

На периметре или в ядре корпоративной сети Амибиг предоставляет средствам защиты расшифрованный трафик для решения следующих задач:

- проверка HTTPS-трафика сотрудников на наличие угроз с помощью NGFW;
- предотвращение утечки конфиденциальных данных через зашифрованные каналы с помощью DLP;
- обнаружение вредоносного ПО в TLS-трафике;
- контроль доступа к веб-ресурсам по категориям URL.

Миграция с F5 BIG-IP SSL Orchestrator

Для переноса с F5 BIG-IP SSL Orchestrator (SSLO) Амибиг предоставляет следующие возможности:

- импорт конфигурации из резервных копий F5 UCS с преобразованием объектов;
- выполнение совместимых с iRules сценариев в подсистеме aRules;
- перенос сервисных цепочек и политик в модель Амибиг.

Оркестрация сервисов безопасности

Амибиг расшифровывает соединение один раз, после чего последовательно передаёт данные нескольким средствам защиты:

- один цикл расшифрования обслуживает несколько средств защиты;
- политика определяет, какой трафик расшифровывать, а какой передавать без расшифрования;
- для разных категорий трафика можно назначить разные сервисные цепочки.

Ключевые возможности

Возможности продукта сгруппированы по этапам обработки трафика, управления политиками и эксплуатации устройства.

SSL/TLS-инспекция

Подсистема SSL/TLS-инспекции поддерживает следующие возможности.

Возможность	Описание
TLS 1.0–1.3	Поддержка указанных версий протокола
ГОСТ-TLS	Полный перехват соединений с российской криптографией: отдельный ГОСТ-УЦ, наборы шифров ГОСТ Р 34.12-2015 («Кузнечик»); см. ГОСТ TLS
Динамическая генерация сертификатов	Сертификат создаётся при установлении соединения с каждым целевым сервером
Настраиваемые наборы шифров	Правила Cipher Rules с минимальной и максимальной версией TLS, списками шифров и управлением ALPN
Передача без расшифрования	Исключение трафика из TLS-инспекции по SNI, IP-адресу или порту
Понижение версии HTTP/2	Переход с HTTP/2 на HTTP/1.1 для совместимости с инспекционными сервисами
Сертификаты УЦ RSA и ECDSA	Поддержка обоих алгоритмов для сертификата УЦ

Сервисы и сервисные цепочки

Расшифрованный трафик можно передавать в inline- и ICAP-сервисы.

Тип сервиса	Описание
Inline (NGFW)	Прозрачный L2-мост, трафик передаётся на внешний NGFW
ICAP (DLP)	Интеграция по протоколу ICAP: REQMOD для запросов, RESPMOD для ответов

Сервисная цепочка задаёт порядок передачи расшифрованного трафика через inline- и ICAP-сервисы. Одна цепочка может содержать не более восьми сервисов.

Для каждого сервиса настраиваются:

- **пул** экземпляров и способ распределения нагрузки: `round-robin`, `session-hash`, `least-connections` или `predictive`;
- **монитор доступности** TCP, ICMP или HTTP GET с настраиваемыми порогами;

- **действие при недоступности** (`service-down-action`): пропуск трафика без проверки или блокировка.

Политики трафика

Политики определяют, какой трафик расшифровывать и через какую сервисную цепочку направлять:

Критерий совпадения	Примеры
SNI (Server Name Indication)	<code>*.example.com</code> , <code>banking.corp.ru</code>
IP-адрес источника / назначения	<code>10.0.0.0/8</code> , <code>192.168.1.100</code>
Порт назначения	<code>443</code> , <code>8443</code> , <code>993</code>
Протокол	TCP, UDP, ICMP
URL-категория	Финансы, социальные сети, вредоносные

Действие `intercept` включает расшифрование и передачу в сервисную цепочку. `bypass` передаёт трафик без расшифрования через прозрачную цепочку, `passthrough` — без расшифрования и без цепочки, `reject` отклоняет соединение.

Подсистема сценариев aRules

aRules (Amicon Rules) — Tcl-совместимая подсистема обработки HTTP-трафика в сервисной цепочке. Она выполняет следующие операции:

- управление HTTP-заголовками (`HTTP::header insert/remove/replace`);
- условное включение и отключение ICAP-инспекции (`ADAPT::enable`);
- доступ к метаданным соединения (`IP::client_addr` , `HTTP::method` , `HTTP::uri`);
- привязка к сервисам в цепочке или к правилам политики.

Высокая доступность

Кластер высокой доступности работает по следующим правилам:

- кластер состоит из активного и резервного узлов;
- при отказе резервный узел принимает владение виртуальными IP-адресами (VIP) и рассылает GARP;
- применённая на активном узле конфигурация передаётся HA-партнёру;
- состояние установленных соединений между узлами не передаётся, поэтому клиенты переустанавливают соединения к тому же VIP (см. Состояние соединений при переключении);
- при `ssl.client-profile.session-ticket-mode = ha-sync` активный узел передаёт HA-партнёру ключи шифрования TLS-билетов (STЕК), чтобы после повторного подключения клиент мог возобновить TLS-сессию вместо полного TLS handshake;

- HA-канал по умолчанию защищён взаимной TLS-аутентификацией с TLS 1.3.

Управление и диагностика

Для настройки и контроля устройства предусмотрены следующие средства:

- **REST API** для автоматизации настройки и диагностики;
- **веб-интерфейс** на русском и английском языках;
- **ролевая модель** с ролями `admin`, `operator`, `viewer` и списками доступа по IP-адресам;
- **подготовленная (candidate) и действующая (running) конфигурации**, просмотр различий и таймер отката;
- **захват пакетов** на нескольких интерфейсах с сохранением в PCAP;
- **резервное копирование и восстановление конфигурации**, импорт F5 UCS;
- **обновление через слоты A и B** с проверкой подписи пакета и автоматическим откатом при сбое.

Перенос конфигурации с F5 BIG-IP SSLO

Амибиг импортирует резервные копии UCS и преобразует поддерживаемые объекты F5 BIG-IP в собственную JSON-конфигурацию. Перед применением импортированной конфигурации администратор должен проверить список преобразованных и пропущенных объектов.

При планировании переноса учитывайте различия продуктов:

- Амибиг поддерживает inline-сервисы L2 и ICAP, но не поддерживает TAP-сервисы и одновременную обработку трафика обоими узлами кластера;
- кластер Амибиг состоит из двух узлов: активного и резервного;
- aRules выполняют поддерживаемое подмножество Tcl-команд iRules;
- параметры F5, для которых нет прямого соответствия, указываются в отчёте импорта и требуют ручной настройки.

Полный порядок переноса и перечень поддерживаемых объектов приведены в разделе Перенос конфигурации F5 UCS.

Для кого предназначено руководство

Руководство предназначено для специалистов, которые проектируют, внедряют и сопровождают Амибиг:

- администраторы информационной безопасности настраивают политики инспекции, сертификаты и мониторинг;

- сетевые инженеры подключают устройство к сети и настраивают высокую доступность;
- специалисты по миграции переносят конфигурацию с F5 BIG-IP;
- руководители информационной безопасности используют обзор для планирования внедрения.

Архитектура Амибиг

Архитектура Амибиг разделена на подсистему обработки трафика (`dataplane`) и систему управления (`control plane`). Первая принимает пользовательский трафик, выполняет TLS-инспекцию и направляет данные по сервисной цепочке. Вторая хранит и применяет конфигурацию, предоставляет веб-интерфейс и REST API, собирает сведения о состоянии устройства.

Пользовательский и служебный трафик проходят через разные сетевые интерфейсы. Такое разделение не позволяет операциям управления влиять на обработку пользовательского трафика.

Схема высокого уровня

На схеме показаны основные этапы обработки трафика и компоненты управления. Подсистемы высокой доступности и лицензирования работают совместно с системой управления, но не участвуют непосредственно в передаче пользовательских пакетов.

Схема взаимодействия

Источник	Связь	Назначение
Приём трафика	→	TLS-инспекция
TLS-инспекция	→	Выбор политики
Выбор политики	→	Inline-сервис NGFW
Выбор политики	→	ICAP-сервис DLP или AV
Inline-сервис NGFW	→	Возврат трафика
ICAP-сервис DLP или AV	→	Возврат трафика
Веб-интерфейс	→	REST API
REST API	→	Конфигурация
Конфигурация	→	Применение конфигурации
Высокая доступность	Компонент	—

Источник	Связь	Назначение
Лицензирование	Компонент	—
Обработка трафика	Группа	Приём трафика, TLS-инспекция, Выбор политики, Inline-сервис NGFW, ICAP-сервис DLP или AV, Возврат трафика
Управление системой	Группа	Веб-интерфейс, REST API, Конфигурация, Применение конфигурации, Высокая доступность, Лицензирование

Обработка трафика

Пользовательский трафик обрабатывает служба `tforge` на базе DPDK. Одно соединение от начала до конца обслуживает один рабочий поток. Соединение не передаётся между потоками, а на пути обработки пакетов нет общего изменяемого состояния. Количество рабочих потоков зависит от числа выделенных ядер CPU и задаётся в параметрах подсистемы обработки трафика на экране **Система**.

Перехват трафика

Амибиг принимает TCP-соединения прозрачно для клиента и сервера и передаёт сведения об исходных IP-адресах источника и назначения в подсистему расшифрования. Адрес origin-соединения определяется настройками трансляции адреса источника.

TLS-инспекция

Для TLS-инспекции Амибиг завершает клиентское TLS-соединение, динамически создаёт сертификат целевого сервера и устанавливает отдельное origin-соединение. Расшифрованные данные передаются в подсистему применения политик. Для криптографических операций используется AES-NI; при наличии оборудования можно задействовать Intel QAT.

Применение политики

Подсистема применения политик сопоставляет трафик с правилами по SNI, IP-адресам, портам и протоколам. По результату проверки она выбирает действие и сервисную цепочку, а после проверки возвращает трафик в TLS-прокси.

Дополнительные компоненты выполняют следующие задачи:

- подсистема aRules исполняет Tcl-совместимые сценарии, изменяет HTTP-заголовки и управляет ICAP-инспекцией;
- диспетчер сессий хранит таблицу сессий, распределяет соединения между участниками пулов и учитывает состояние мониторов доступности.

Inline-сервис NGFW

Амибиг передаёт трафик на внешний NGFW через прозрачный L2-мост. После проверки NGFW возвращает трафик устройству. Если в пуле настроено несколько экземпляров NGFW, Амибиг распределяет соединения между ними.

ICAP-сервис DLP или антивируса

Встроенный клиент ICAP передаёт HTTP-запросы и ответы в системы DLP, антивирусы и другие средства анализа содержимого:

- `REQMOD` — проверка или изменение HTTP-запроса до отправки серверу;
- `RESPMOD` — проверка или изменение HTTP-ответа до отправки клиенту;
- поддерживаются ответ `204 No Modification`, предварительная передача данных (`preview`) и постоянные соединения (`keep-alive`);
- aRules управляют проверкой с помощью команд `ADAPT::enable request/response`.

Управление системой

Система управления хранит настройки, проверяет и применяет конфигурацию, предоставляет средства диагностики. Её компоненты не обрабатывают пользовательский трафик.

REST API

REST API предоставляет операции создания, чтения, изменения и удаления всех объектов конфигурации: сети, TLS-профилей, сервисов, цепочек, политик и aRules. Для аутентификации применяются подписанные Bearer-токены и роли `admin`, `operator`, `viewer`. Спецификация OpenAPI формируется автоматически.

Через API также выполняются проверка лицензии, загрузка и применение подписанных пакетов обновлений, а также переход от подготовленной (`candidate`) конфигурации к действующей (`running`).

Система управления конфигурацией

Применение конфигурации построено на следующих механизмах:

- **Формальная схема** описывает структуру и ограничения конфигурации.

- **Подготовленная (candidate) и действующая (running) конфигурации** разделяют редактирование и применение настроек.
- **Таймер отката** восстанавливает предыдущую конфигурацию, если администратор не подтвердил новую в установленный срок.
- **Проверка различий** показывает изменённые разделы перед применением.

Веб-интерфейс

Веб-интерфейс доступен на русском и английском языках. Через него администратор изменяет конфигурацию, контролирует состояние устройства и запускает диагностику.

Сервис высокой доступности

Сервис высокой доступности выполняет следующие задачи:

- узлы обмениваются контрольными сообщениями по TCP с настраиваемым интервалом;
- при переключении резервный узел принимает VIP, а клиенты переуставляют соединения, поскольку состояние установленных соединений не передаётся (см. Состояние соединений при переключении);
- активный узел передаёт HA-партнёру применённую конфигурацию;
- при `ssl.client-profile.session-ticket-mode = ha-sync` активный узел передаёт HA-партнёру ключи шифрования TLS-билетов (STEC), чтобы после повторного подключения клиент мог возобновить TLS-сессию вместо полного TLS handshake;
- HA-канал по умолчанию защищён взаимной TLS-аутентификацией с TLS 1.3.

Сервис лицензирования

Лицензия определяет доступный набор функций и срок их использования:

- редакции `demo` (30 дней), `basic`, `standard` и `premium` предоставляют разные наборы функций;
- лицензия привязана к аппаратному идентификатору сервера и защищена криптографической подписью;
- функции, не входящие в редакцию лицензии, недоступны;
- после окончания срока лицензии действует льготный период.

Потоковая обработка трафика

Ниже показана обработка зашифрованного соединения: от сообщения TLS ClientHello до отправки клиенту повторно зашифрованного ответа. Амибиг устанавливает два независимых TLS-соединения — клиентское и origin-

соединение. Между ними устройство передаёт расшифрованные данные по сервисной цепочке.

Обработка зашифрованного соединения

Диаграмма последовательности показывает обе TLS-сессии и этап инспекции между ними.

Последовательность взаимодействия

Источник	Связь	Назначение
Клиент, Амибиг	Этап	Клиентское TLS-соединение
Клиент	TLS ClientHello	Амибиг
Амибиг	ServerHello + созданный сертификат	Клиент
Клиент	Зашифрованные данные	Амибиг
Амибиг	Этап	Расшифрование и проверка
Амибиг	Этап	Политика → aRules → NGFW → ICAP
Амибиг, Сервер	Этап	Origin-соединение
Амибиг	TLS ClientHello	Сервер
Сервер	ServerHello + сертификат	Амибиг
Амибиг	Зашифрованные данные	Сервер
Сервер	Ответ	Амибиг
Амибиг	Этап	Инспекция ответа
Амибиг	Повторно зашифрованный ответ	Клиент

1. Клиент отправляет TLS ClientHello, и Амибиг принимает соединение.
2. Амибиг завершает клиентское TLS-соединение и отправляет клиенту ServerHello с динамически созданным сертификатом.
3. Амибиг расшифровывает прикладные данные клиента.
4. Подсистема политик выбирает действие и последовательно передаёт данные в aRules, inline-сервис NGFW и ICAP-сервис DLP, если они входят в цепочку.
5. Амибиг устанавливает origin-соединение с сервером.
6. Амибиг передаёт серверу прикладные данные и получает ответ.
7. Ответ проходит проверку, после чего Амибиг шифрует его и отправляет клиенту.

Этапы обработки

В таблице приведена последовательность обработки перехваченного соединения.

Этап	Описание
1	Прозрачный перехват TCP-соединения
2	Завершение клиентского TLS-соединения и создание сертификата
3	Расшифрование данных
4	Передача в подсистему политик вместе с IP-адресами клиента и сервера и значением SNI
5	Выбор действия: <code>intercept</code> , <code>bypass</code> или <code>reject</code>
6	Выполнение aRules: модификация заголовков, управление ICAP
7	Передача через inline-сервис NGFW в режиме L2-моста
8	Отправка ICAP-запросов <code>REQMOD</code> и <code>RESPMOD</code> в DLP или антивирус
9	Возврат проверенного трафика и установка origin-соединения

Изоляция трафика

Компоненты обработки трафика работают в отдельном сетевом окружении. Это позволяет:

- независимо управлять маршрутами пользовательского трафика;
- отделить пользовательский трафик от служебного;
- исключить конфликты адресации между подсистемами;
- использовать несколько сервисных цепочек одновременно.

Масштабируемость и производительность

Архитектура Амибиг спроектирована для масштабирования на многоядерных серверных платформах. Ориентиры производительности:

Параметр	Ориентир
Пропускная способность MITM SSL/TLS	≥ 20 Гбит/с
Новые соединения (базовый порог)	1 000 000 соединений в минуту ($\approx 16\,700$ в секунду)
Новые соединения	100 000 в секунду — хороший результат; 200 000 в секунду — целевой ориентир
Одновременные соединения	$\geq 1\,000\,000$

Фактическая производительность зависит от характеристик оборудования, алгоритма сертификата УЦ и глубины сервисной цепочки. Для ускорения криптографических операций система использует AES-NI; также поддерживается аппаратный криптоускоритель Intel QAT.

Системные требования

Амибиг работает на выделенных серверах архитектуры x86_64. Требуемые ресурсы зависят от числа новых и одновременных соединений, доли трафика с TLS-инспекцией, алгоритма сертификата УЦ и состава сервисных цепочек. Перед закупкой оборудования проверьте целевой профиль нагрузки на стенде.

Аппаратные конфигурации

Выберите конфигурацию в зависимости от назначения системы. Минимальные требования рассчитаны на лабораторную проверку, рекомендации для промышленной эксплуатации требуют подтверждения нагрузочными испытаниями.

Лабораторный или демонстрационный стенд

Минимальная конфигурация подходит для знакомства с продуктом и функциональной проверки настроек.

Компонент	Минимальное требование
CPU	4 ядра x86_64 с поддержкой AES-NI
Оперативная память	8 ГБ
Диск	32 ГБ
Сетевые интерфейсы	Не менее двух интерфейсов: для управления и пользовательского трафика
Режим загрузки	UEFI

Такая конфигурация предназначена для знакомства с функциями продукта и проверки настроек. Не используйте результаты лабораторного стенда как оценку производительности промышленной системы.

Демо-лицензия дополнительно ограничивает конфигурацию четырьмя сетевыми интерфейсами и одной сервисной цепочкой. Ограничения редакций приведены в разделе Лицензирование.

Рекомендуемая промышленная конфигурация

Для промышленной эксплуатации используйте конфигурацию не ниже указанной в таблице.

Компонент	Рекомендация
CPU	Не менее 16 ядер, современный Intel Xeon или сопоставимый процессор
Оперативная память	Не менее 64 ГБ
Диск	SSD объёмом не менее 128 ГБ; для интенсивной записи журналов рекомендуется NVMe
Сетевые интерфейсы	Порты класса 10/25 GbE для пользовательского трафика и отдельный интерфейс управления
Режим загрузки	UEFI
Питание	Резервированные блоки питания

Для устойчивой инспекции трафика на скоростях в десятки Гбит/с обычно требуются не менее 32 ядер, 128 ГБ оперативной памяти и сетевые адаптеры класса 25/100 GbE. Итоговая конфигурация определяется только после нагрузочных испытаний с типовым для организации трафиком.

Требования к процессору

Процессор должен поддерживать:

- 64-разрядную архитектуру x86_64;
- AES-NI;
- SSE 4.2.

Для высоких нагрузок рекомендуются AVX-512/VAES, SHA-NI и аппаратные источники случайных чисел RDRAND/RDSEED. Наличие расширения само по себе не

гарантирует конкретного прироста производительности: результат зависит от модели процессора, набора шифров и профиля трафика.

Проверить расширения можно командами:

```
uname -m  
grep -m1 -E 'aes|sse4_2|vaes|sha_ni|rdrand|rdseed' /proc/cpuinfo
```

Выбор алгоритма сертификата УЦ

Алгоритм сертификата УЦ для TLS-инспекции существенно влияет на число новых соединений, которое может обработать устройство:

- ECDSA P-256 рекомендуется как основной вариант для высокой производительности;
- RSA-2048 требует заметно больше процессорных ресурсов при той же частоте установления соединений;
- Intel QAT может снизить нагрузку в конфигурациях с большим объёмом операций RSA.

Сравнивайте варианты на собственном наборе TLS-профилей и сертификатов.

Требования к оперативной памяти

Потребление памяти растёт с числом одновременных соединений. Память используют состояние TLS, сетевые буферы, таблицы соединений, служебные процессы и hugerpages DPDK.

Практические ориентиры:

- 8 ГБ — небольшой лабораторный стенд;
- 32 ГБ — стенд для функциональных и интеграционных испытаний;
- 64 ГБ и более — промышленная система с большим числом соединений.

После запуска нагрузки контролируйте фактическое потребление памяти и наличие свободных hugerpages в средствах мониторинга Амибиг.

Требования к диску

Минимальный поддерживаемый объём диска — 32 ГБ. Установщик создаёт слоты А и В для обновлений, а также отдельные разделы конфигурации и данных.

Для промышленной эксплуатации используйте SSD объёмом не менее 128 ГБ. Дополнительное место требуется для журналов, пакетов обновлений, диагностических данных и файлов захвата трафика.

Сетевые требования

Минимальная схема включает:

- отдельный интерфейс управления;
- один или несколько портов обработки пользовательского трафика.

В промышленной схеме обычно выделяют клиентские и серверные порты, интерфейсы подключения inline-сервисов и отдельный HA-канал между узлами. Пропускная способность всех участков сервисной цепочки должна быть не ниже плановой скорости пользовательского трафика.

Используйте сетевые адаптеры, совместимость которых подтверждена для текущей версии Амибиг и установленного драйвера DPDK. Наличие драйвера в ядре Linux не означает совместимость адаптера с подсистемой обработки трафика. Перед закупкой уточните перечень проверенных моделей у поставщика.

Ограничение агрегированных каналов

В текущей версии агрегированный канал на портах обработки трафика работает только с одним рабочим потоком. Если `dataplane.dpdk.lcore-mask` включает несколько потоков, проверка конфигурации отклонит применение агрегированного канала. Это ограничение исключает одновременное использование агрегации и многопоточной обработки трафика; подробнее см. раздел Агрегирование каналов.

Требования к UEFI

Перед установкой проверьте следующие параметры встроенного программного обеспечения сервера.

Параметр	Требование
Режим загрузки	UEFI; Legacy BIOS не поддерживается
Secure Boot	Не требуется
AES-NI	Должен быть включён
IOMMU (VT-d)	Рекомендуется для DPDK, SR-IOV и Intel QAT

Операционная система

Амибиг поставляется с собственной операционной системой `amios` и не требует установки стороннего ПО. Установка продукта поверх Ubuntu, RHEL и других операционных систем не поддерживается.

Требования к браузеру

Для работы с веб-интерфейсом используйте актуальную поддерживаемую версию Google Chrome, Mozilla Firefox, Microsoft Edge или Safari. В браузере должны быть включены JavaScript и файлы cookie, а рабочая станция должна иметь доступ по HTTPS к интерфейсу управления Амибиг. Internet Explorer не поддерживается.

Требования к кластеру высокой доступности

Для кластера требуются:

- два узла с одинаковой версией Амибиг;
- одинаковая схема подключения портов обработки трафика;
- устойчивый HA-канал с IP-связностью между узлами;
- единый источник синхронизации времени;
- поддержка GARP сетевыми устройствами в сегментах VIP;
- подключение каждого inline-сервиса к обоим узлам.

Перед вводом кластера в эксплуатацию проверьте ручное переключение на резервный узел и прохождение трафика через все сервисные цепочки.

Установка

Амибиг устанавливается на сервер с загрузочного USB-носителя. Текстовый установщик размечает целевой диск, устанавливает операционную систему в слоты А и В и сохраняет первоначальные параметры интерфейса управления. При первой загрузке сценарий `firstboot` создаёт ключи, начальные пароли и цифровой паспорт устройства (`Device Identity`).

Установщик работает в консоли сервера или через IPMI/KVM. Для перехода между элементами используйте клавиши со стрелками и `Tab` , для подтверждения — `Enter` .

Подготовка носителя

Загрузочный образ установщика входит в комплект поставки — файл `amibig-installer.hdimg.gz` (см. Артефакты релиза). Распакуйте его и запишите на USB-носитель посекторно:

```
gunzip -c amibig-installer.hdimg.gz | dd of=/dev/sdX bs=4M status=progress
```

Замените `/dev/sdX` именем устройства USB-носителя.

Внимание. Команда уничтожит все данные на выбранном носителе. Перед запуском проверьте имя устройства.

После записи загрузите сервер с USB-носителя в режиме UEFI.

Шаги мастера установки

Мастер последовательно настраивает язык, сеть и целевой диск, после чего устанавливает систему и предлагает перезагрузить сервер.

1. **Выбор языка** — русский или английский.
2. **Приветствие** — экран **Установка системы**, кнопка **Начать установку**.
3. **Проверка установленной системы.** Если на одном из дисков уже установлена ОС Амибиг, выберите **Загрузить**, чтобы выйти из установщика и запустить систему, или **Переустановить**.
4. **Сетевой интерфейс.** Список обнаруженных портов с MAC-адресом, состоянием соединения и скоростью. Выберите порт управления и способ получения адреса: **DHCP (автоматически)** или **Статический адрес**. Для статического адреса установщик проверяет формат IP-адреса, непрерывность маски и принадлежность шлюза той же подсети. Кнопка **Пропустить** откладывает настройку сети; после установки её потребуется выполнить с консоли.
5. **Выбор диска.** Список дисков с размером, моделью и типом (SSD/HDD). Установочный носитель и съёмные USB-диски показываются в списке, но заблокированы для выбора — установить систему на них нельзя.
6. **Подтверждение.** Сводка выбора (диск, интерфейс, режим адреса) и предупреждение: все данные на выбранном диске будут уничтожены.
7. **Установка.** Индикатор показывает семь этапов: разметку диска, форматирование, установку системы, настройку загрузчика (ESP), подготовку разделов `config` и `data`, завершение установки.
8. **Завершение.** Проверьте созданные разделы, извлеките USB-носитель и нажмите **Перезагрузить**.

Разметка диска (A/B-слоты)

Установщик создаёт на целевом диске таблицу GPT из пяти разделов:

Раздел	Метка	Размер	Назначение
1	<code>esp</code>	128 МБ (FAT32)	загрузчик systemd-boot и ядра слотов A/B
2	<code>rootfs-a</code>	2 ГБ (ext4)	корневая файловая система, слот A

Раздел	Метка	Размер	Назначение
3	rootfs-b	2 ГБ (ext4)	корневая файловая система, слот В
4	config	512 МБ (ext4)	конфигурация (/ config): amibig.json , сертификаты, ключи, история
5	data	остаток диска (ext4)	журналы и пакеты обновлений (/ data)

Система ставится в слот А; слот В заполняется при первом обновлении. Разделы config и data общие для обоих слотов, поэтому конфигурация и журналы переживают обновления и откаты.

На раздел config установщик сразу записывает начальный amibig.json с выбранным портом управления (plane: management) и способом получения адреса (dhcp или static с адресом и шлюзом).

Первая загрузка (firstboot)

При первой загрузке сценарий firstboot выполняет следующие действия:

1. Инициализирует /config файлами по умолчанию.
2. Разворачивает доверенные ключи проверки подписи обновлений и КЕК, фиксирует «эпоху лицензии» (дату первого запуска).
3. Задаёт начальный пароль root и сохраняет его в файле /config/.initial-root-password (доступен только root).
4. Запрашивает в консоли имя узла и способ подготовки сертификата УЦ для TLS-инспекции. Можно создать самоподписанный сертификат с указанным полем Common Name (CN) или загрузить собственный позже через веб-интерфейс.
5. Генерирует самоподписанный сертификат веб-интерфейса.
6. Создаёт цифровой паспорт устройства (Device Identity) с предварительным серийным номером в файле /config/identity/device-identity.json . Серийный номер понадобится для запроса лицензии; подробнее см. в разделе Лицензирование.
7. Создаёт учётную запись admin со случайным начальным паролем. Пароль сохраняется в /config/.initial-admin-password , выводится в консоли и удаляется после первой смены. При первом входе система требует сменить пароль.
8. Создаёт маркер /config/.firstboot-done , чтобы не выполнять первоначальную настройку повторно.

Дальнейшие действия — вход в веб-интерфейс и базовая настройка — описаны в разделе Быстрый старт.

Установка без участия администратора (`auto-install.conf`)

Первоначальную настройку можно автоматизировать. Если в разделе `config` находится файл `/config/auto-install.conf`, сценарий `firstboot` читает параметры из него и не запрашивает значения в консоли. Установщик копирует этот файл на раздел `config` автоматически, если он включён в состав установочного носителя (`/installer-payload/auto-install.conf`).

Формат файла — shell-переменные `КЛЮЧ=значение` :

```
### /config/auto-install.conf
HOSTNAME=amibig-dc1
CERT_MODE=generate
CA_CN="AMIBIG CA"
```

Параметр	Значения	Описание
<code>HOSTNAME</code>	строка	имя хоста узла
<code>CERT_MODE</code>	<code>generate</code> (по умолчанию) / <code>skip</code>	создать самоподписанный сертификат УЦ для инспекции или пропустить этот шаг
<code>CA_CN</code>	строка	Common Name создаваемого сертификата УЦ (по умолчанию <code>AMIBIG CA</code>)

Ключи, пароли, цифровой паспорт устройства и сертификат веб-интерфейса создаются одинаково при интерактивной и автоматической установке.

Быстрый старт

В этом разделе приведена минимальная настройка Амибиг для TLS-инспекции. В результате устройство будет расшифровывать HTTPS-трафик и передавать его по сервисной цепочке с одним средством защиты.

Предварительные условия

До начала настройки подготовьте доступ к устройству, сертификат УЦ и подключённое средство защиты.

- сервер с установленной системой Амибиг (см. Системные требования);

- доступ к интерфейсу управления;
- актуальная версия поддерживаемого браузера;
- сертификат УЦ и закрытый ключ для TLS-инспекции;
- настроенное средство защиты, например NGFW или DLP, подключённое к Амибиг.

Перед началом настройки

Установщик Амибиг запускается с USB-носителя и последовательно запрашивает язык, интерфейс управления, способ адресации и целевой диск. После подтверждения установщик размечает диск, устанавливает систему и предлагает перезагрузить сервер. Подробная процедура приведена в разделе Установка.

При первой загрузке сценарий `firstboot` создаёт начальные пароли `admin` и `root`, выводит их в консоли вместе с адресом веб-интерфейса и формирует предварительный серийный номер устройства. Серийный номер потребуется для запроса лицензии (см. Лицензирование).

Шаг 1. Первоначальный вход

Для первого входа используйте адрес и временные учётные данные, которые сценарий `firstboot` выводит в локальной консоли.

1. Подключите рабочую станцию к интерфейсу управления Амибиг.
2. Узнайте IP-адрес и начальный пароль администратора. При первой загрузке `firstboot` печатает их на консоли сервера в виде:

```
WebUI:      https://<management-ip>/
Cert SHA-256: AA:BB:CC:DD:...:11:22:33:44
SSH:        root / <начальный-пароль-root>
WebUI:      admin / <начальный-пароль-admin>
```

Если IP получен через DHCP и недоступен — подключитесь к консоли сервера, баннер `/etc/issue` отображает текущий адрес, сгенерированные пароли и отпечаток сертификата веб-интерфейса. Также пароль `admin` сохранён в файле `/config/.initial-admin-password` на устройстве (только для `root`, удаляется при первой смене пароля).

3. Откройте браузер и перейдите по адресу:

```
https://<management-ip>/
```

Доверие сертификату. Амибиг использует самоподписанный сертификат веб-интерфейса, который генерируется на устройстве при первой загрузке. Браузер покажет предупреждение о ненадёжном

соединении. Чтобы убедиться, что подключение установлено именно к этому устройству, сравните отпечаток сертификата (SHA-256) с тем, что напечатан в локальной консоли в строке `Cert SHA-256`. Совпадение подтверждает подлинность.

Не добавляйте первоначальный сертификат в доверенное хранилище, пока не сверите его отпечаток. После входа замените сертификат веб-интерфейса сертификатом корпоративного УЦ.

4. Введите учётные данные:

Параметр	Значение
Логин	<code>admin</code>
Пароль	начальный пароль с консоли

5. Нажмите **Войти**.

Пароль администратора генерируется случайно при первой загрузке и уникален для каждого устройства. Учётные данные `admin/admin` или другие предсказуемые значения не используются.

Шаг 2. Смена пароля

При первом входе система обязательно потребует сменить пароль — интерфейс открывает форму изменения пароля до доступа к остальной функциональности.

1. В форме смены пароля введите:

- **Текущий пароль** — начальный пароль `admin` с консоли сервера.
- **Новый пароль** — постоянный пароль администратора.
- **Повторите новый пароль**.

2. Нажмите **Сохранить**.

Пароль должен содержать не менее 8 символов и символы не менее чем из трёх категорий:

- буквы нижнего регистра;
- буквы верхнего регистра;
- цифры;
- спецсимволы.

Дополнительно: пароль не должен совпадать с одним из последних ранее использованных (история паролей) и не должен входить в встроенный список заведомо слабых паролей.

После успешной смены пароля файл `/config/.initial-admin-password` удаляется, начальный пароль больше нигде не хранится.

Шаг 3. Настройка сети

Настройте сетевые интерфейсы для прохождения трафика.

3.1 Интерфейс управления

Проверьте параметры интерфейса управления, через который открыт текущий сеанс веб-интерфейса.

1. Перейдите в раздел **Сеть → Физ. интерфейсы**.
2. Выберите интерфейс управления, обычно `eth0`.
3. Проверьте IP-адрес и маску подсети.
4. При необходимости измените параметры и сохраните их.

Если нужных интерфейсов ещё нет в списке, добавьте их кнопкой **Обнаружить**. Обнаружение добавляет интерфейсы IOMMU-группами: сетевая карта добавляется целиком, выбрать из группы отдельный порт нельзя.

3.2 VLAN обработки трафика

Для этого сценария создайте тегированные клиентский и серверный VLAN. На смежных портах коммутатора настройте те же теги 802.1Q.

1. Перейдите в раздел **Сеть → VLAN**.
2. Нажмите **+ Добавить** и создайте VLAN `client` на порту, обращённом к клиентской сети. В поле **Тег** укажите `100`.
3. Нажмите **Добавить** в форме.
4. Снова нажмите **+ Добавить**, создайте VLAN `server` на порту, обращённом к серверной сети, и укажите в поле **Тег** значение `200`. Нажмите **Добавить** в форме.
5. Нажмите **Сохранить** в разделе **Сеть**.

Для нетегированного сегмента не создавайте VLAN без тега в веб-интерфейсе. Назначьте Self-IP и слушатель топологии непосредственно порту или транку обработки трафика. Подробнее см. в разделах VLAN и Прозрачная топология.

3.3 Адреса Self-IP

Создайте Self-IP в VLAN обработки трафика:

1. Перейдите в раздел **Сеть → Self-IP**.
2. Нажмите **+ Добавить**.
3. Заполните параметры:

Параметр	Пример	Описание
Имя	client-side	Произвольное имя
Адрес и маска	10.1.1.1/24	Адрес на клиентской стороне в формате CIDR
VLAN	client	VLAN обработки трафика, обращённый к клиентам

4. Нажмите **Добавить** в форме.
5. Снова нажмите **+ Добавить** и задайте Self-IP серверной стороны:

Параметр	Пример
Имя	server-side
Адрес и маска	10.2.1.1/24
VLAN	server

6. Нажмите **Добавить** в форме.

3.4 Маршрут по умолчанию

Добавьте маршрут по умолчанию, если целевые серверы находятся за шлюзом серверного сегмента.

1. Перейдите в раздел **Сеть → Маршруты**.
2. Нажмите **+ Добавить**.
3. Укажите шлюз по умолчанию:

Параметр	Пример
Имя	default-server
Сеть назначения	0.0.0.0/0
Шлюз	10.2.1.254

4. Нажмите **Добавить** в форме.

Исходящие соединения и MSS. По умолчанию исходящие соединения к серверам отправляются с собственного адреса устройства (режим `automar`), а анонсируемый TCP MSS автоматически ограничивается — для быстрого старта менять эти настройки не требуется. При необходимости они настраиваются на экране **Система**: разделы «Egress (трансляция адреса источника)» и «Протокол (TCP / MSS)» (см. Egress / SNAT и MSS и MTU).

3.5 Применение сетевых настроек

Перенесите подготовленные сетевые настройки в действующую конфигурацию и подтвердите их после проверки доступа к устройству.

1. Нажмите **Сохранить** в разделе **Сеть**, чтобы записать Self-IP и маршрут в подготовленную конфигурацию.
2. Нажмите **Применить конфигурацию** в нижней части боковой панели.
3. Дождитесь сообщения об успешном применении.
4. Подтвердите конфигурацию до истечения таймера отката, который по умолчанию составляет 2 минуты.

Если после применения связь с интерфейсом управления потеряна, не подтверждайте конфигурацию. По истечении таймера система автоматически восстановит предыдущие настройки.

Шаг 4. Добавление сертификата УЦ

Амибиг использует сертификат УЦ для подписи сертификатов, которые создаются при TLS-инспекции.

1. Перейдите в раздел **SSL/TLS → Сертификаты**.
2. Нажмите **+ Добавить**.
3. Заполните форму **Добавить сертификат**:

Поле	Значение	Описание
Имя	<code>inspection-ca</code>	Имя для назначения сертификата профилю клиента

Поле	Значение	Описание
Тип	ca	Сертификат УЦ с CA: TRUE и Key Usage: Certificate Sign
Загрузить сертификат	Файл PEM (.crt или .pem)	Сертификат УЦ
Загрузить ключ	Файл PEM (.key или .pem)	Закрытый ключ УЦ

4. Нажмите **ОК**. Сертификат и ключ будут добавлены в подготовленную конфигурацию SSL/TLS.

Рекомендация. Для высокой производительности используйте ECDSA P-256. RSA-2048 требует больше процессорных ресурсов при той же частоте установления соединений.

Установите сертификат УЦ как доверенный на всех клиентских устройствах, трафик которых будет расшифровываться. Для централизованного распространения можно использовать групповые политики Active Directory или MDM.

Шаг 5. Настройка SSL/TLS-профилей

Настройте профили клиентского соединения и соединения с целевым сервером. На этих вкладках находятся общие профили устройства, а не списки именованных объектов.

5.1 Профиль клиентского соединения

Профиль определяет параметры клиентского TLS-соединения:

1. Перейдите в раздел **SSL/TLS → Профиль клиента**.
2. Укажите следующий параметр:

Параметр	Значение	Описание
CA-сертификат	inspection-ca	Сертификат для подписи сертификатов, создаваемых при TLS-инспекции

- Для быстрого старта оставьте остальные параметры без изменений. Ограничения версий TLS задаются правилом шифрования, назначенным профилю.

5.2 Профиль origin-соединения

Профиль определяет параметры TLS-соединения с целевым сервером:

- Перейдите в раздел **SSL/TLS → Профиль сервера**.
- Укажите следующий параметр:

Параметр	Значение	Описание
Проверка узла	Да	Проверять сертификат сервера

- Нажмите **Сохранить** в разделе SSL/TLS.

Шаг 6. Создание сервиса

Создайте описание средства безопасности, через которое будет проходить трафик.

Вариант A: Inline-сервис (NGFW)

До создания сервиса добавьте тегированные VLAN `to-ngfw` и `from-ngfw` с тегами 300 и 301 на портах, подключённых к соответствующим сторонам NGFW. Создайте для этих VLAN адреса Self-IP и сохраните раздел **Сеть**.

- Перейдите в раздел **Инспекция → Сервисы**.
- Нажмите **+ Добавить**.
- Заполните:

Параметр	Значение	Описание
Имя	ngfw-1	Имя сервиса
Тип	Inline	Прозрачный L2-мост

Параметр	Значение	Описание
Входящий VLAN	to-ngfw	VLAN от Амибиг к NGFW
Исходящий VLAN	from-ngfw	VLAN от NGFW к Амибиг
Действие при недоступности	Игнорировать	Пропускать трафик без проверки при недоступности NGFW

4. Нажмите **ОК**, чтобы добавить сервис в таблицу.

5. Нажмите **Сохранить** в разделе **Сервисы**.

Вариант В: ICAP-сервис (DLP)

Этот вариант используйте, если средство защиты принимает запросы по протоколу ICAP.

1. Перейдите в раздел **Инспекция → Сервисы**.

2. Нажмите **+ Добавить**.

3. Заполните:

Параметр	Значение	Описание
Имя	dlp-1	Имя сервиса
Тип	ICAP	Протокол ICAP
ICAP URI (REQMOD)	icap://10.3.1.10:1344/reqmod	Адрес DLP для запросов
ICAP URI (RESPMOD)	icap://10.3.1.10:1344/respmo	Адрес DLP для ответов
Действие при недоступности	Игнорировать	Пропускать трафик без проверки при недоступности DLP

4. Нажмите **ОК**, чтобы добавить сервис в таблицу.

5. Нажмите **Сохранить** в разделе **Сервисы**.

Шаг 7. Создание сервисной цепочки

Сервисная цепочка определяет порядок прохождения трафика через сервисы.

1. Перейдите в раздел **Инспекция → Цепочки сервисов**.

2. Нажмите **+ Добавить**.

3. Заполните:

Параметр	Значение
Имя	inspection-chain
Сервисы	ngfw-1 (или dlp-1 , или оба)

4. Если в цепочке несколько сервисов, расположите их в требуемом порядке. Трафик проходит сервисы последовательно.

5. Нажмите **ОК**, чтобы добавить цепочку в таблицу.

6. Нажмите **Сохранить** в разделе **Цепочки сервисов**.

Шаг 8. Создание политики

Политика определяет, какой трафик расшифровывать и через какую цепочку направлять.

1. Перейдите в раздел **Инспекция → Политики**.

2. Нажмите **+ Добавить**.

3. Создайте правило для перехвата:

Параметр	Значение	Описание
Имя	intercept-all	Имя правила
Действие	Перехват (intercept)	Расшифровывать TLS-трафик и передавать его на проверку
Сервисная цепочка	inspection-chain	Цепочка для инспекции
Порт назначения	443	HTTPS-трафик

4. Нажмите **ОК**, чтобы добавить правило перехвата в таблицу.

5. При необходимости снова нажмите **+ Добавить** и создайте правило прямого пропуска для исключений:

Параметр	Значение	Описание
Имя	bypass-banking	Имя правила

Параметр	Значение	Описание
Действие	Прямой пропуск (passthrough)	Пропускать без расшифрования TLS- трафика и без сервисной цепочки
SNI	*.bank.ru, *.cbr.ru	Банковские сайты

- Нажмите **ОК**, чтобы добавить правило исключения.
- Расположите правила в требуемом порядке. Амибиг проверяет их сверху вниз и применяет первое правило, условиям которого соответствует трафик.
- Нажмите **Сохранить** в разделе **Политики**.

Шаг 9. Настройка слушателя

Слушатель определяет, на каком VLAN Амибиг принимает трафик для инспекции.

- Перейдите в раздел **Топология**.
- Выберите режим `transparent-l3`.
- В разделе **Слушатель** укажите:

Параметр	Значение
VLAN	<code>client</code>
Адрес	<code>0.0.0.0</code>
Порт	<code>443</code>
Протокол	TCP

- Убедитесь, что в поле **Привязанная политика** указана политика с правилом `intercept-all`.
- Нажмите **Сохранить**.

Шаг 10. Применение конфигурации

До применения ещё раз проверьте список изменений. Подтверждайте новую конфигурацию только после проверки доступа к устройству.

- Нажмите **Применить конфигурацию** в боковой панели.
- Проверьте список изменений.

3. Нажмите **Применить** и дождитесь завершения операции.
4. Нажмите **Подтвердить** до истечения 2-минутного таймера отката.

Если после применения конфигурации связь с устройством потеряна, не подтверждайте изменения. По истечении таймера система автоматически восстановит предыдущую конфигурацию.

Шаг 11. Проверка работы

После применения проверьте состояние сервисов, результат TLS-инспекции и фактическое прохождение трафика.

Проверка статуса сервисов

Сначала убедитесь, что подсистема инспекции и средство защиты готовы обрабатывать соединения.

1. Перейдите на экран **Обзор**.
2. Убедитесь, что подсистема инспекции запущена, а все сервисы доступны.
3. Если сервис недоступен, откройте его карточку и проверьте результат монитора доступности.

Проверка инспекции трафика

Проверьте подмену сертификата на клиентском устройстве, которому доверен сертификат УЦ Амибиг.

1. На клиентской рабочей станции с установленным сертификатом УЦ откройте браузер.
2. Перейдите на HTTPS-сайт, который соответствует правилу `intercept`.
3. Откройте сведения о сертификате сайта.
4. Убедитесь, что сертификат подписан УЦ Амибиг.

Проверка через захват трафика

Захват пакетов позволяет подтвердить путь тестового соединения через выбранные интерфейсы.

1. Перейдите в раздел **Мониторинг → Диагностика** и найдите блок **Новый захват**.
2. Выберите интерфейсы.
3. Запустите захват и сформируйте тестовый трафик с клиента.
4. Остановите захват и сохраните PCAP-файл для анализа в Wireshark.

Следующие шаги

После завершения быстрого старта рекомендуется:

Задача	Раздел документации
Настройка VLAN и агрегированных каналов	Сеть
Трансляция адреса источника в исходящих соединениях	Трансляция адреса источника (SNAT)
Ограничение TCP MSS и настройка MTU портов	MSS и MTU
Тонкая настройка шифров и TLS-профилей	SSL/TLS
Инспекция ГОСТ-TLS	ГОСТ TLS
Добавление дополнительных сервисов и цепочек	Сервисы, Цепочки
Создание детальных политик по SNI и IP	Политики
Написание aRules для обработки HTTP	aRules
Настройка HA-кластера	Высокая доступность
Настройка резервного копирования	Управление
Импорт конфигурации из F5 BIG-IP	Перенос конфигурации F5 UCS

Типичные проблемы при первоначальной настройке

В таблице приведены проверки для наиболее частых ошибок первоначальной настройки.

Проблема	Возможная причина	Решение
Браузер показывает ошибку сертификата	Сертификат УЦ не установлен на клиенте	Установите сертификат УЦ в доверенное хранилище ОС или браузера
Сайт не открывается	Ошибка маршрутизации	Проверьте Self-IP, маршруты и доступность сервера
Сервис находится в состоянии Down	Монитор доступности не получает ответ	Проверьте подключение NGFW или DLP, IP-адреса и порты

Проблема	Возможная причина	Решение
Конфигурация восстановлена автоматически	Новая конфигурация не подтверждена в срок	Повторно примените настройки и подтвердите их до истечения таймера
Низкая производительность	Недостаточно процессорных ресурсов или памяти	Проверьте загрузку CPU и использование памяти на экране Обзор

Сеть

В этом разделе администратор настраивает физические порты, VLAN, Self-IP, статические маршруты и агрегированные каналы Амибиг. Эти объекты определяют, через какие интерфейсы устройство принимает пользовательский трафик, взаимодействует с сервисами безопасности и обеспечивает доступ к системе управления.

До создания политик и сервисных цепочек назначьте портам требуемые роли, привяжите VLAN и проверьте маршруты по фактической схеме подключения. Ошибка на этом этапе может привести к потере доступа к устройству или нарушению прохождения трафика.

Физические интерфейсы

На вкладке **Сеть → Физ. интерфейсы** администратор назначает роли физическим портам Амибиг и контролирует состояние соединения. К этим портам привязываются VLAN, Self-IP, слушатели (`listener`) и VIP.

Амибиг

Одиночный узел

Обзор

Трафик

Система

Сеть

SSL/TLS

Топология

Инспекция

Мониторинг

Управление

Применить конфигурацию

Документация

admin

Сеть

Физ. интерфейсы

Агрегирование каналов

VLAN

Self-IP

Маршруты

Имя, подсистема или MAC-адрес

Фильтры

Обнаружить

Добавить

ИМЯ	ПОДСИСТЕМА	НАЗНАЧЕНИЯ	СОСТО...	СКОР...	МЕТОД
☒ eth0	Management c...	Control / M...	—	1G	—
☒ dpdk0	Dataplane port...	VLAN client,... ещё 1	—	1G	—
☒ dpdk1	Dataplane port...	VLAN to-ngf... ещё 4	Прив...	1G	—
☒ dpdk2	Dataplane port...	не задействован	Прив...	1G	—
☒ dpdk3	Dataplane port...	VLAN from-... ещё 4	—	1G	—
☒ dpdk4	Dataplane port...	не задействован	—	1G	—
☒ dpdk5	Dataplane port...	не задействован	—	1G	—
☒ dpdk6	Dataplane port...	не задействован	—	1G	—
☒ dpdk7	Dataplane port...	не задействован	—	1G	—

Всего строк: 9

Сбросить

Отмена

Сохранить

eth0

Control / Management

PCI-АДРЕС

MAC

52:54:00:ac:10:05

Описание

Management control plane

Назначения

Привязок нет

Статистика

Входящий

Исходящий

Ошибки

Потери

Счётчики появятся после запуска дата-плейна

Назначение портов

В Амибиг используются два класса портов:

- **управление и НА:** `plane = management` или `plane = ha`, `binding = kernel`;
- **обработка трафика:** `plane = data`, `binding = dpdk`.

При назначении портов действуют следующие ограничения:

- порты ядра ОС используются только для управления и НА;
- DPDK-порты используются только для клиентского и сервисного трафика;
- VLAN, Self-IP, слушатели и VIP пользовательского трафика нельзя размещать на портах ядра ОС.

Таблица интерфейсов

Вкладка построена по схеме «список слева — сведения справа». Каждый интерфейс занимает отдельную строку; агрегированные каналы отображаются в том же списке. Подробнее см. в разделе Агрегирование каналов. В строке указаны:

- переключатель административного включения (у агрегированного канала собственного переключателя нет);
- имя интерфейса; описание — второй строкой под именем;
- **Подсистема** — назначение порта: управление, обработка трафика или НА;
- **Назначения** — объекты, использующие интерфейс: агрегированный канал, VLAN, Self-IP, слушатель (`listener`) или inline-сервис. При большом числе привязок часть списка сворачивается в **ещё N**. Порт данных без привязок помечается как **не задействован**; для нового порта это нормальное состояние, а не ошибка;
- **Состояние** — текущее состояние соединения;
- **Скорость** — фактическая скорость или значение из конфигурации;
- **Метод** — доступный способ включения светового индикатора порта;
- меню строки «`:`» с действиями **Изменить**, **Мигнуть индикатором порта** и **Удалить**.

Под списком выводится счётчик «Всего строк». Над списком — текстовый поиск (подсказка «Имя, подсистема или MAC-адрес») и панель «Фильтры» с отбором по статусу, подсистеме, назначениям, состоянию, скорости и методу идентификации; поиск и фильтры действуют одновременно.

Панель справа показывает состояние и назначение выбранного интерфейса, PCI- и MAC-адреса, имя интерфейса в ОС, драйвер, привязанные объекты и счётчики принятого и переданного трафика, ошибок и потерь. Счётчики появляются после запуска подсистемы обработки трафика.

Объекты на портах обработки трафика

Только DPDK-порты и агрегированные каналы из таких портов могут использоваться для:

- VLAN;
- Self-IP;
- `topology.listener`;
- `ha.vip`;
- VLAN inline-сервиса (`to-service` и `from-service`);
- виртуальных серверов и адресов приёма.

Если порт имеет `binding = kernel`, веб-интерфейс и REST API не позволят привязать к нему клиентский или сервисный трафик.

Поля порта

Для физического интерфейса задаются следующие параметры.

Поле	Назначение
<code>name</code>	Логическое имя порта в конфигурации
<code>plane</code>	<code>management</code> , <code>ha</code> или <code>data</code>
<code>binding</code>	<code>kernel</code> для управления и HA, <code>dpdk</code> для обработки трафика
<code>pci-addr</code>	PCI-адрес адаптера для DPDK-порта
<code>speed</code>	Заданная скорость, если фактическое значение недоступно
<code>mtu</code>	Необязательный MTU DPDK-порта от 576 до 1500; допускается только уменьшение
<code>enabled</code>	Административное состояние
<code>description</code>	Понятное администратору назначение порта

Имена интерфейсов и агрегированных каналов должны быть уникальны. Веб-интерфейс и API отклоняют имя, которое уже используется. Это исключает неоднозначные ссылки на участника агрегации, родительский интерфейс VLAN или сторону inline-сервиса.

Добавление и редактирование

В окнах «Добавить физ. интерфейс» и «Редактировать физ. интерфейс»:

- при редактировании в верхней части окна отображаются состояние, драйвер и метод идентификации; эти сведения нельзя изменить. Там же находится кнопка **Мигнуть индикатором порта**;
- поля сгруппированы в секции **Основное** (имя, описание, подсистема, скорость, PCI-адрес, MAC-адрес и состояние интерфейса) и **Параметры IPv4** (адрес, широковещательный адрес, шлюз и MTU);
- адрес и маска вводятся одним полем **Адрес и маска** в нотации CIDR, например 192.168.57.10/24 . В конфигурации значение по-прежнему хранится двумя полями — `address` и `netmask` . Адрес без маски отклоняется с подсказкой «Укажите адрес вместе с маской, например 192.168.57.10/24».

MTU порта

Для порта обработки трафика можно задать MTU: поле «MTU порта» в секции **Параметры IPv4** карточки интерфейса. Допустимый диапазон — 576..1500, значение может только понижать MTU относительно физического. Если поле не заполнено, порт работает с MTU интерфейса по умолчанию (1500).

При заданном MTU анонсируемый TCP MSS для соединений через этот порт равен `mtu - 40` . Для агрегированного канала действует минимальный MTU среди участников, у которых он задан. Стратегия управления MSS и рекомендации по выбору значений описаны в разделе Управление TCP MSS и MTU.

Регистрация изменений состояния

Подключение и отключение DPDK-портов регистрируются в журнале событий на экране **Просмотр журналов**. Если настроен SNMP, по этим событиям отправляются trap-сообщения. Для участников агрегированного канала дополнительно фиксируются переходы в активное и неактивное состояние LACP — см. Агрегирование каналов.

Обнаружение интерфейсов

Кнопка **Обнаружить** открывает окно «Обнаруженные физические интерфейсы» с двумя вкладками: **Доступные** — интерфейсы, которых ещё нет в конфигурации, и **Добавленные** — уже добавленные, они показываются только для просмотра.

Строки в окне сгруппированы по IOMMU-группам, а флажок выбора доступен только для группы. Интерфейсы одной сетевой карты добавляются вместе; выбрать отдельный порт группы нельзя. Это связано с тем, что при привязке к DPDK вся IOMMU-группа передаётся из ядра ОС в подсистему обработки трафика. Порт без IOMMU-группы отображается отдельной группой «Без IOMMU-группы».

Обнаружение работает через API:

- GET /api/v1/config/ports
- GET /api/v1/config/ports/discovery

В списке обнаружения указывается рекомендуемая роль порта:

- интерфейс маршрута по умолчанию обычно предлагается как `management` ;
- остальные физические порты предлагаются как `data` .

Перед добавлением проверьте предложенную роль: изменение привязки порта к DPDK может прервать доступ через ядро ОС.

Идентификация порта

Чтобы найти физический разъём порта, запустите мигание его индикатора:

- веб-интерфейс: пункт **Мигнуть индикатором порта** в меню строки «`:`» или в заголовке карточки редактирования;
- API: POST /api/v1/config/ports/{name}/identify

Метод подсветки система выбирает по возможностям порта и показывает в колонке «Метод»:

- **ethtool (безопасно)** — штатное мигание через драйвер ядра, трафик не прерывается;
- **kernel-rebind (с обрывом трафика)** — DPDK-порт временно возвращается драйверу ядра ОС, поэтому трафик на нём прерывается. Веб-интерфейс запрашивает подтверждение в окне «Идентификация прервёт трафик»;
- **sysfs LED** — управление индикатором напрямую через sysfs.

Если ни один метод недоступен, пункт меню неактивен: для идентификации необходим PCI-адрес или имя интерфейса в ОС. Во время операции над таблицей отображается сообщение «Идентификация портов в процессе» с оставшимся временем. Закрытие сообщения не прекращает мигание. Агрегированный канал не имеет собственного индикатора; идентифицируйте его порты-участники по отдельности.

Пример

В примере интерфейсы разделены по назначению: управление, обработка трафика и НА-канал.

```
{
  "interfaces": {
    "interface": [
      {
        "name": "eth0",
        "description": "Интерфейс управления",
        "plane": "management",
        "binding": "kernel",
        "speed": "1G",
```

```

        "enabled": true
    },
    {
        "name": "dpdk0",
        "description": "Клиентский трафик",
        "plane": "data",
        "binding": "dpdk",
        "pci-addr": "0000:03:00.0",
        "speed": "10G",
        "enabled": true
    },
    {
        "name": "dpdk1",
        "description": "Трафик сервисной цепочки",
        "plane": "data",
        "binding": "dpdk",
        "pci-addr": "0000:03:00.1",
        "speed": "10G",
        "enabled": true
    },
    {
        "name": "ha0",
        "description": "HA-канал",
        "plane": "ha",
        "binding": "kernel",
        "speed": "1G",
        "enabled": true
    }
  ]
}

```

Рекомендации

При назначении портов соблюдайте следующие рекомендации:

- Используйте понятные логические имена: `dpdk0`, `dpdk1`, `ha0`.
- Не размещайте клиентские VLAN и слушатели на портах `management` и `ha`.
- Для DPDK-портов задавайте `speed`, если после передачи порта DPDK фактическую скорость определить нельзя.
- Перед удалением порта проверьте столбец «Назначения» и перенастройте связанные VLAN, Self-IP и inline-сервисы.

VLAN

VLAN в Амибиг — тегированный логический сегмент пользовательского трафика, привязанный к порту или агрегированному каналу подсистемы обработки трафика. Для нетегированного трафика Self-IP, слушатель, плечо inline-сервиса или VIP можно назначить непосредственно порту или транку без объекта VLAN.

Правила привязки VLAN

VLAN пользовательского трафика нельзя назначать порту управления или HA-порту.

Допустимые родители VLAN:

- порт обработки трафика (`plane = data` , `binding = dpdk`);
- агрегированный канал, состоящий только из портов обработки трафика.

Если попытаться привязать VLAN к порту управления или HA-порту, конфигурация не пройдёт проверку при применении.

Где VLAN используются

Тегированный VLAN связывает следующие объекты обработки трафика:

- входящий клиентский сегмент (`topology.listener.vlan`);
- сервисные сегменты inline-цепочки (`to-service` , `from-service`);
- Self-IP подсистемы обработки трафика;
- VIP кластера и адреса виртуальных серверов через связанные Self-IP.

Поля

Для VLAN задаются следующие параметры.

Поле	Назначение
<code>name</code>	логическое имя VLAN
<code>interface</code>	Порт или агрегированный канал обработки трафика
<code>tag</code>	тег 802.1Q от 1 до 4094 в веб-интерфейсе; в модели JSON/YANG поле необязательное
<code>description</code>	описание

Настройка в веб-интерфейсе

Веб-интерфейс создаёт тегированные VLAN и требует тег от 1 до 4094. В списке родительских объектов доступны только:

- порты обработки трафика;
- агрегированные каналы из портов обработки трафика.

Если нужный объект отсутствует в списке, проверьте его `plane` , `binding` и состав агрегированного канала.

Для нетегированного сегмента не создавайте VLAN в веб-интерфейсе. Выберите порт обработки трафика или транк непосредственно в форме Self-IP, слушателя топологии, плеча inline-сервиса или VIP. Модель JSON/YANG по-прежнему допускает VLAN без поля `tag` , но новые нетегированные привязки в веб-интерфейсе задаются непосредственно.

API

VLAN управляются через секцию `network`:

- `GET /api/v1/config/network`
- `PUT /api/v1/config/network`

Пример

В примере клиентский и сервисные VLAN используют теги 802.1Q.

```
{
  "vlangs": {
    "vlan": [
      {
        "name": "client",
        "interface": "dpdk0",
        "tag": 10,
        "description": "Входящий клиентский трафик"
      },
      {
        "name": "to-ngfw",
        "interface": "dpdk1",
        "tag": 100,
        "description": "Входящий трафик inline-сервиса"
      },
      {
        "name": "from-ngfw",
        "interface": "dpdk2",
        "tag": 101,
        "description": "Исходящий трафик inline-сервиса"
      }
    ]
  }
}
```

VLAN на агрегированном канале

Чтобы передавать тегированный трафик через агрегированный канал, укажите его имя в поле `interface`.

```
{
  "trunks": {
    "trunk": [
      {
        "name": "dataplane-trunk",
        "interfaces": ["dpdk0", "dpdk1"],
        "lacp": "enabled",
        "lacp-mode": "active"
      }
    ]
  },
  "vlangs": {
    "vlan": [
      {
        "name": "client",
        "interface": "dataplane-trunk",
        "tag": 200
      }
    ]
  }
}
```

```
}
}
```

Рекомендации

При настройке VLAN соблюдайте следующие рекомендации:

- Используйте имена по назначению: `client`, `to-ngfw`, `from-ngfw`, `server`.
- Не создавайте VLAN управления в секции пользовательского трафика. Адрес интерфейса управления задаётся в `system.management`.
- Если VLAN используется inline-сервисом, заранее создайте для него Self-IP.

Адреса Self-IP

Self-IP — это адрес Амибиг в сегменте обработки трафика. Его назначают VLAN для тегированного трафика либо непосредственно порту или агрегированному каналу для нетегированного трафика. Self-IP используется для маршрутизации, подключения inline-сервисов, проверки адреса слушателя (`listener`) и размещения VIP.

Разделение адресов управления и обработки трафика

Self-IP не используется для доступа к системе управления. Адрес и шлюз интерфейса управления задаются отдельно:

- `system.management.address`
- `system.management.gateway`

Веб-интерфейс, API и SSH доступны через интерфейс управления ядра ОС. Self-IP относится только к подсистеме обработки трафика.

Правила привязки

Self-IP должен быть привязан ровно к одному объекту второго уровня: VLAN в поле `vlan` или порту либо агрегированному каналу в поле `interface`. Указать оба поля одновременно нельзя. До создания Self-IP привяжите VLAN к порту или агрегированному каналу обработки трафика.

Недопустимо:

- создавать Self-IP в VLAN на порту управления;
- использовать Self-IP как адрес интерфейса управления;
- размещать слушатель или VIP в VLAN ядра ОС через Self-IP.

Поля

Объект Self-IP содержит следующие поля.

Поле	Назначение
name	логическое имя объекта
address	IPv4/IPv6 адрес в CIDR
vlan	VLAN обработки трафика; взаимоисключимо с <code>interface</code>
interface	Порт или агрегированный канал обработки трафика; взаимоисключимо с <code>vlan</code>
gateway	Необязательный шлюз сегмента

Поле **Шлюз** задаёт шлюз для трафика, который покидает сегмент через этот Self-IP. Подсистема обработки трафика использует значение при передаче через inline-сервисы и построении таблиц маршрутизации слушателей. Параметр `system.management.gateway` относится только к интерфейсу управления.

Таблица Self-IP

Таблица на вкладке **Self-IP** экрана **Сеть** показывает привязку каждого адреса в двух столбцах:

- **Привязка** — тип привязки (VLAN или Интерфейс) и имя объекта;
- **Порт** — физический порт или агрегированный канал, через который эта привязка работает: для VLAN — его родительский интерфейс, для прямой привязки — сам выбранный интерфейс.

Столбец «Порт» позволяет определить физический путь VLAN. Прямая нетегированная привязка также отображается в столбце «Назначения» соответствующего порта.

Где Self-IP используется

Адрес Self-IP участвует в следующих операциях подсистемы обработки трафика:

- шлюз для пути через inline-сервис;
- маршрутизация в подсистеме обработки трафика;
- проверка допустимого адреса слушателя;
- проверка подсети виртуального сервера и адреса приёма;
- адрес источника исходящих соединений к origin-серверам в режиме SNAT `automap` — см. Трансляция адреса источника (SNAT).

API

Объекты Self-IP находятся в `network.self-ips.self-ip`:

- GET `/api/v1/config/network`
- PUT `/api/v1/config/network`

Пример

В примере отдельный адрес Self-IP назначен каждому VLAN сервисного пути.

```
{
  "self-ips": {
    "self-ip": [
      {
        "name": "client",
        "address": "192.168.56.10/24",
        "vlan": "client"
      },
      {
        "name": "to-ngfw",
        "address": "192.168.57.10/24",
        "vlan": "to-ngfw"
      },
      {
        "name": "from-ngfw",
        "address": "192.168.58.10/24",
        "vlan": "from-ngfw"
      }
    ]
  }
}
```

Связь со слушателем и VIP

При проверке конфигурации система учитывает следующие зависимости:

- `topology.listener.vlan` должен указывать на VLAN обработки трафика;
- если `listener.address` задан явно, он должен входить в подсеть одного из адресов Self-IP этого VLAN;
- адрес виртуального сервера или VIP должен принадлежать подсети обработки трафика.

Рекомендации

При планировании адресов Self-IP соблюдайте следующие рекомендации:

- Создавайте Self-IP для каждого VLAN, который участвует в маршрутизации или сервисной цепочке.
- Не используйте Self-IP для доступа к системе управления.
- Для двухстековой сети можно назначить IPv4- и IPv6-адреса Self-IP одному VLAN обработки трафика.

Статические маршруты

На экране **Сеть → Маршруты** администратор задаёт шлюзы для сетей назначения. До применения маршрута убедитесь, что его шлюз доступен через настроенный интерфейс или Self-IP.

Когда нужны статические маршруты

Статические маршруты решают две основные задачи:

- **маршрут по умолчанию** задаёт шлюз для трафика, который не соответствует более точному маршруту;
- **маршрут к конкретной сети** направляет трафик выбранной подсети через заданный шлюз.

Маршруты из секции `network.routes` используются подсистемой обработки трафика. Маршрут интерфейса управления задаётся отдельно параметром `system.management.gateway`.

Параметры маршрута

Каждый статический маршрут содержит имя, сеть назначения и адрес следующего шлюза.

Параметр	Тип	Обязательный	Описание
<code>name</code>	строка	да	Уникальное логическое имя маршрута (например, <code>default</code> , <code>to-servers</code> , <code>branch-office</code>).
<code>network</code>	строка	нет	Сеть назначения в нотации CIDR. По умолчанию <code>0.0.0.0/0</code> (маршрут по умолчанию). Примеры: <code>10.0.0.0/8</code> , <code>172.16.0.0/12</code> , <code>192.168.10.0/24</code> .

Параметр	Тип	Обязательный	Описание
gateway	строка	да	IP-адрес следующего шлюза. Шлюз должен быть доступен через один из интерфейсов устройства.

Маршрут по умолчанию

Значение `network: "0.0.0.0/0"` задаёт маршрут по умолчанию. Через указанный шлюз проходит весь трафик, для которого нет более точного маршрута. В конфигурации допускается только один маршрут по умолчанию для IPv4.

Настройка в веб-интерфейсе

Веб-интерфейс позволяет просмотреть и изменить маршруты в подготовленной конфигурации перед их применением.

Просмотр маршрутов

Перейдите в **Сеть → Маршруты**. В таблице отображаются все подготовленные статические маршруты:

Столбец	Описание
Имя	Логическое имя маршрута
Сеть	Сеть назначения в CIDR
Шлюз	IP-адрес следующего шлюза

Добавление маршрута

Добавьте маршрут в подготовленную конфигурацию и сохраните раздел.

1. Нажмите **+ Добавить** в таблице маршрутов.
2. Заполните поля в модальном окне:
 - **Имя** — логическое имя маршрута.
 - **Сеть** — сеть назначения в CIDR (по умолчанию `0.0.0.0/0`).
 - **Шлюз** — IP-адрес шлюза.
3. Нажмите **Добавить** в форме.
4. Нажмите **Сохранить** в карточке секции.

Редактирование маршрута

Нажмите **Изм.** в строке маршрута, измените параметры и нажмите **Сохранить** в форме. Затем нажмите **Сохранить** в карточке раздела.

Удаление маршрута

Нажмите кнопку удаления в строке маршрута. Маршрут будет удалён из подготовленной (candidate) конфигурации после сохранения раздела.

Управление через REST API

Через REST API маршруты передаются как часть полной секции `network`.

Получение конфигурации маршрутов

Чтобы получить секцию сетевой конфигурации, выполните запрос:

```
GET /api/v1/config/network
```

Маршруты находятся в поле `routes.route` ответа:

```
{
  "routes": {
    "route": [
      {
        "name": "default",
        "network": "0.0.0.0/0",
        "gateway": "192.168.0.1"
      },
      {
        "name": "to-servers",
        "network": "10.0.0.0/8",
        "gateway": "10.0.1.254"
      }
    ]
  }
}
```

Обновление конфигурации

Чтобы заменить сетевую конфигурацию, выполните запрос:

```
PUT /api/v1/config/network
Content-Type: application/json
```

Тело запроса должно содержать полную секцию `network`, включая `routes`.

Применение конфигурации

Чтобы перенести маршруты в действующую конфигурацию:

1. **Применить** (`POST /api/v1/config/apply`) — запускается 2-минутный таймер отката.
2. **Подтвердить** (`POST /api/v1/config/confirm`) — в течение 2 минут.

Внимание. Ошибка в маршруте по умолчанию может привести к потере связи с устройством. Если доступ пропал, не подтверждайте новую конфигурацию: по истечении таймера система восстановит предыдущую.

Примеры конфигурации

Примеры ниже показывают маршрут по умолчанию и более точные маршруты к отдельным сетям назначения.

Только маршрут по умолчанию

Минимальная конфигурация маршрутизации:

```
{
  "routes": {
    "route": [
      {
        "name": "default",
        "network": "0.0.0.0/0",
        "gateway": "192.168.0.1"
      }
    ]
  }
}
```

Маршрут по умолчанию и маршрут к серверной сети

В примере трафик к серверной сети направляется через отдельный шлюз, а для остальных сетей используется маршрут по умолчанию.

```
{
  "routes": {
    "route": [
      {
        "name": "default",
        "network": "0.0.0.0/0",
        "gateway": "192.168.0.1"
      },
      {
        "name": "to-datacenter",
        "network": "10.0.0.0/8",
        "gateway": "10.0.1.254"
      }
    ]
  }
}
```

```
}
}
```

После применения трафик к сетям `10.0.0.0/8` пойдёт через шлюз `10.0.1.254`, остальной трафик — через `192.168.0.1`.

Маршруты к нескольким сетям

В следующем примере для каждой удалённой сети указан собственный шлюз.

```
{
  "routes": {
    "route": [
      {
        "name": "default",
        "network": "0.0.0.0/0",
        "gateway": "192.168.0.1"
      },
      {
        "name": "servers-dc1",
        "network": "10.1.0.0/16",
        "gateway": "10.0.1.254"
      },
      {
        "name": "servers-dc2",
        "network": "10.2.0.0/16",
        "gateway": "10.0.2.254"
      },
      {
        "name": "branch-offices",
        "network": "172.16.0.0/12",
        "gateway": "192.168.0.254"
      }
    ]
  }
}
```

Взаимодействие с другими компонентами

Статические маршруты используются совместно с Self-IP и настройками трансляции адреса источника, но не заменяют внутренние маршруты сервисных цепочек.

Маршрутизация в сервисных цепочках

Амибиг автоматически создаёт внутренние маршруты сервисных цепочек. Они не отображаются как пользовательские статические маршруты и не требуют ручной настройки.

Связь с Self-IP

Шлюз маршрута должен быть доступен из подсети одного из Self-IP адресов. Например, если маршрут по умолчанию указывает на шлюз `192.168.0.1`, на устройстве должен быть Self-IP в подсети `192.168.0.0/24` (например, `192.168.0.250/24`).

Возвратный маршрут при SNAT `none`

При `origin-snat: none` целевой сервер видит адрес клиента и отправляет ответ непосредственно ему. Соединение установится только в том случае, если обратный маршрут проходит через Амибиг. Подробнее см. в разделе Трансляция адреса источника (SNAT).

Рекомендации

При настройке маршрутизации соблюдайте следующие рекомендации:

- Настройте маршрут по умолчанию, если устройству требуется доступ за пределы непосредственно подключённых подсетей.
- Используйте максимально специфичные маски для маршрутов к конкретным сетям. Маршрут с длинной маской (/24) имеет приоритет над маршрутом с короткой (/8) при совпадении адреса назначения.
- Убедитесь, что шлюз каждого маршрута доступен через один из настроенных Self-IP. Недоступный шлюз приведет к потере трафика.
- При изменении маршрута по умолчанию подтвердите конфигурацию только после проверки связи с устройством.
- Давайте маршрутам осмысленные имена, отражающие назначение сети (`to-datacenter` , `branch-offices` , `dmz-servers`).

Агрегирование каналов (LACP)

На вкладке **Сеть → Агрегирование каналов** администратор объединяет физические интерфейсы по протоколу LACP (IEEE 802.3ad). Агрегированный канал используется как один логический интерфейс и сохраняет связь при отказе отдельного порта, пока доступен хотя бы один исправный участник.

Ограничение текущей версии. Агрегированный канал на DPDK-портах работает только с одним рабочим потоком. Если `dataplane.dpdk.lcore-mask` включает несколько потоков, система отклонит конфигурацию при проверке. Агрегацию нельзя одновременно использовать с многопоточной обработкой трафика.

Обзор

LACP позволяет объединить несколько физических интерфейсов. К агрегированному каналу можно привязать VLAN, Self-IP и сетевые объекты сервисных цепочек так же, как к обычному порту.

Агрегирование каналов обеспечивает:

- использование совокупной пропускной способности участников в пределах производительности одного рабочего потока; одно соединение обычно проходит через один физический порт;
- сохранение связи при отказе отдельного порта;
- распределение потоков по заданному алгоритму хеширования.

Параметры агрегированного канала

Для агрегированного канала задаются участники и параметры LACP.

Параметр	Тип	Обязательный	По умолчанию	Описание
<code>name</code>	строка	да	—	Уникальное имя агрегированного канала (например, <code>bond0</code> , <code>uplink-trunk</code>).
<code>interfaces</code>	массив строк	да	—	Список имен физических интерфейсов-участников. Интерфейсы должны быть предварительно добавлены на вкладке «Физ. интерфейсы».
<code>lacp</code>	перечисление	нет	<code>enabled</code>	Включение LACP. Значения: <code>enabled</code> (режим 802.3ad), <code>disabled</code> (статическая агрегация <code>balance-xor</code>).

Параметр	Тип	Обязательный	По умолчанию	Описание
<code>lACP-mode</code>	перечисление	нет	<code>active</code>	Параметр режима LACP (<code>active</code> или <code>passive</code>). В текущей реализации DPDK значение сохраняется, но не передаётся bonding PMD.
<code>lACP-timeout</code>	перечисление	нет	<code>short</code>	Параметр периода LACP (<code>short</code> или <code>long</code>). В текущей реализации DPDK значение сохраняется, но не передаётся bonding PMD.
<code>distribution-hash</code>	перечисление	нет	<code>src-dst-ipport</code>	Алгоритм распределения трафика между звеньями.
<code>description</code>	строка	нет	—	Произвольное текстовое описание.

Имена агрегированных каналов и физических интерфейсов должны быть уникальны. Веб-интерфейс и API отклоняют имя, которое уже используется.

Алгоритмы распределения (`distribution-hash`)

Алгоритм определяет набор полей пакета, по которому выбирается физический порт.

Значение	Уровень	Описание
<code>dst-mac</code>	L2	Хеширование по MAC-адресу назначения. Минимальная гранулярность.

Значение	Уровень	Описание
<code>src-dst-mac</code>	L2+L3	Хеширование по MAC-адресам + IP-адресам источника и назначения.
<code>src-dst-ipport</code>	L3+L4	Хеширование по IP-адресам + портам TCP/UDP источника и назначения. Максимальная гранулярность, рекомендуется для большинства сценариев.

Выбор алгоритма влияет на равномерность распределения трафика:

- `src-dst-ipport` обеспечивает наилучшее распределение, так как использует максимум информации из заголовков пакетов. Рекомендуется по умолчанию.
- `dst-mac` имеет минимальную гранулярность и может привести к неравномерной загрузке при небольшом количестве MAC-адресов назначения.
- `src-dst-mac` является компромиссным вариантом для сценариев, когда L4-информация недоступна (например, для фрагментированных пакетов).

Параметры режима и периода LACP

Поля `lacp-mode` и `lacp-timeout` предусмотрены схемой конфигурации и используются при импорте настроек. В текущей реализации агрегированных каналов DPDK эти значения не передаются в bonding PMD. Не рассчитывайте время обнаружения отказа по значениям этих полей: после применения проверьте фактическое согласование LACP с подключённым коммутатором.

Участники агрегированного канала

Интерфейс, включённый в агрегированный канал, нельзя одновременно использовать как самостоятельный порт. Проверка конфигурации отклоняет прямую привязку VLAN, Self-IP, VIP, слушателя или стороны inline-сервиса к порту-участнику. Привязывайте эти объекты к самому агрегированному каналу. В сообщении об ошибке система указывает канал, в который уже входит порт.

Отображение на вкладке «Физ. интерфейсы»

Агрегированный канал отображается отдельной строкой на вкладке **Физ. интерфейсы**, рядом с портами-участниками. В строке и на панели сведений указаны участники, параметры LACP, текущее состояние и привязанные VLAN и Self-IP. Для портов-участников в столбце **Назначения** такие привязки имеют пометку **через**, например `client-vlan (через bond0)`.

У агрегированного канала нет собственного светового индикатора. Чтобы найти физические разъёмы, запустите мигание индикатора отдельно для каждого порта-участника.

Настройка в веб-интерфейсе

В веб-интерфейсе можно просмотреть состояние агрегации, изменить её состав и проверить зависимости перед удалением.

Просмотр агрегированных каналов

Откройте вкладку **Агрегирование каналов** на экране **Сеть**. Таблица показывает подготовленные агрегированные каналы, их участников и текущее состояние. Описание отображается под именем канала.

Столбец	Описание
Имя	Имя агрегированного канала; описание — второй строкой
Участники	Список интерфейсов-участников через запятую
Активно	Текущее состояние участников по данным телеметрии
LACP	Состояние LACP (<code>enabled</code> / <code>disabled</code>)
Режим LACP	Режим согласования (<code>active</code> / <code>passive</code>)
Тайм-аут LACP	Периодичность LACPDU (<code>short</code> / <code>long</code>)
Хэш распределения	Алгоритм распределения трафика

Создание агрегированного канала

Создайте канал и добавьте его в подготовленную конфигурацию.

1. Нажмите кнопку **Добавить** на вкладке «Агрегирование каналов».
2. В разделе **Основное** укажите имя и описание канала, затем выберите порты-участники. Список содержит только свободные DPDK-порты.
3. В разделе **Параметры LACP** задайте `lacp`, `lacp-mode`, `lacp-timeout` и алгоритм распределения трафика.
4. Нажмите **Добавить**.
5. Нажмите **Сохранить** в карточке раздела.

Редактирование агрегированного канала

Откройте карточку «Редактировать агрегированный канал» кнопкой редактирования в строке таблицы. Измените параметры и нажмите **Сохранить**.

Удаление агрегированного канала

Нажмите кнопку удаления. Удаление заблокировано, если на агрегированный канал ссылаются VLAN (поле `interface` в настройках VLAN). Система покажет список зависимых VLAN. Сначала перенастройте или удалите зависимые VLAN.

Управление через REST API

REST API используется для получения и полной замены секции сетевой конфигурации с параметрами агрегированных каналов.

Получение конфигурации агрегированных каналов

Получите сетевую конфигурацию следующим запросом:

```
GET /api/v1/config/network
```

Агрегированные каналы находятся в поле `trunks.trunk` ответа. В текущей версии объединять можно только порты обработки трафика:

```
{
  "trunks": {
    "trunk": [
      {
        "name": "trunk1",
        "interfaces": ["dpdk0", "dpdk1"],
        "lacp": "enabled",
        "lacp-mode": "active",
        "lacp-timeout": "short",
        "distribution-hash": "src-dst-ipport",
        "description": "Подключение к коммутатору"
      }
    ]
  }
}
```

Обновление конфигурации

Для замены секции сетевой конфигурации выполните запрос:

```
PUT /api/v1/config/network
Content-Type: application/json
```

Тело запроса должно содержать полную секцию `network`, включая `trunks`.

Применение конфигурации

При применении конфигурации Амибиг создаёт агрегированный канал из указанных участников и настраивает LACP. Система автоматически учитывает канал при расчёте памяти `hugepages`; отдельно резервировать память не требуется.

Процесс применения с таймером отката:

1. **Применить** (`POST /api/v1/config/apply`) — запускается 2-минутный таймер.
2. **Подтвердить** (`POST /api/v1/config/confirm`) — в течение 2 минут.

Подробности ограничения по рабочим потокам

Количество рабочих потоков задаётся маской `lcore-mask`: один установленный бит соответствует одному потоку. Для конфигурации с агрегированным каналом оставьте в маске один установленный бит. Если требуется несколько рабочих потоков, не используйте агрегацию.

Распределение агрегированного DPDK-порта между несколькими рабочими потоками в текущей версии не поддерживается. Проверка при применении предотвращает запуск конфигурации, в которой часть потоков не получает трафик агрегированного канала.

Проверяйте 802.3ad на физическом оборудовании с коммутатором, на котором настроен LAG, либо между двумя непосредственно соединёнными устройствами. Стандартный стенд KVM/virtio не предоставляет подключённого устройства с поддержкой LACP, поэтому на нём нельзя проверить согласование и переключение при отказе участника.

Диагностика агрегированного канала

Для захвата трафика выберите агрегированный канал на экране **Диагностика**, а не отдельные порты-участники. DPDK обслуживает участников как единый канал, поэтому выбрать их для захвата по отдельности нельзя.

Изменения состояния агрегации регистрируются на экране **Просмотр журналов**: переходы участников между активным и неактивным состояниями LACP, а также подключение и отключение физических портов. Используйте эти записи для проверки согласования с коммутатором. Если настроен SNMP, по событиям портов отправляются trap-сообщения.

Примеры конфигурации

В примерах показаны типовые варианты агрегации портов и привязки к ней VLAN и Self-IP.

Агрегированный канал из двух портов обработки трафика

В примере объединены два DPDK-порта с активным режимом LACP.

```
{
  "trunks": {
    "trunk": [
      {
        "name": "trunk1",
        "interfaces": ["dpdk0", "dpdk1"],
        "lACP": "enabled",
```

```

        "lacp-mode": "active",
        "lacp-timeout": "short",
        "distribution-hash": "src-dst-ipport",
        "description": "Агрегация для клиентского трафика"
    }
}
}

```

Агрегированный канал из четырёх портов

Следующий пример объединяет четыре порта класса 25 GbE.

```

{
  "trunks": {
    "trunk": [
      {
        "name": "trunk-uplink",
        "interfaces": ["ens3f0", "ens3f1", "ens4f0", "ens4f1"],
        "lacp": "enabled",
        "lacp-mode": "active",
        "lacp-timeout": "short",
        "distribution-hash": "src-dst-ipport",
        "description": "4 × 25 Гбит/с к ядру сети"
      }
    ]
  }
}

```

Номинальная суммарная полоса четырёх портов составляет 100 Гбит/с. Фактическая пропускная способность зависит от распределения потоков, накладных расходов протоколов и ограничения текущей версии одним рабочим потоком. Одно соединение не использует полосу нескольких участников одновременно.

Статический агрегированный канал без LACP (`balance-xor`)

Если подключённое устройство не поддерживает LACP, можно использовать статическую агрегацию с распределением по хешу:

```

{
  "trunks": {
    "trunk": [
      {
        "name": "trunk-legacy",
        "interfaces": ["dpdk2", "dpdk3"],
        "lacp": "disabled",
        "description": "Статическая агрегация без LACP"
      }
    ]
  }
}

```

В этом режиме пакеты распределяются по алгоритму `distribution-hash`. Поля `lacp-mode` и `lacp-timeout` не используются.

Агрегированный канал с VLAN и Self-IP

Объекты создаются в последовательности: агрегированный канал → VLAN → Self-IP.

```
{
  "trunks": {
    "trunk": [
      {
        "name": "trunk1",
        "interfaces": ["dpdk0", "dpdk1"],
        "lacp": "enabled",
        "lacp-mode": "active",
        "lacp-timeout": "short",
        "distribution-hash": "src-dst-ipport"
      }
    ]
  },
  "vlans": {
    "vlan": [
      {
        "name": "client-vlan",
        "interface": "trunk1",
        "tag": 100
      }
    ]
  },
  "self-ips": {
    "self-ip": [
      {
        "name": "client-ip",
        "address": "10.0.1.1/24",
        "vlan": "client-vlan"
      }
    ]
  }
}
```

При применении конфигурации система автоматически создаст агрегированный канал из `dpdk0` и `dpdk1`, настроит на нем VLAN с тегом 100 и назначит IP-адрес `10.0.1.1/24`.

Требования к настройке коммутатора

До применения конфигурации подготовьте подключённый коммутатор:

- Объедините подключённые интерфейсы коммутатора в LAG с включённым LACP.
- Настройте на коммутаторе режим 802.3ad и проверьте согласование LACP после применения конфигурации Амибиг.
- Алгоритмы хеширования на коммутаторе и Амибиг могут различаться: каждая сторона распределяет исходящий трафик независимо.

Рекомендации

При настройке агрегированных каналов соблюдайте следующие рекомендации:

- Выбирайте `distribution-hash: src-dst-ipport` для равномерного распределения потоков в большинстве сценариев с TCP- и UDP-трафиком.

- Добавляйте в агрегированный канал интерфейсы одинаковой скорости. Смешивание интерфейсов разных скоростей (например, 1 Гбит/с и 10 Гбит/с) приведет к неравномерной загрузке.
- Проверьте состояние LACP после применения конфигурации. Если коммутатор не согласовал агрегацию, трафик может не проходить.
- При изменении состава участников будьте готовы к кратковременному прерыванию трафика на время перенастройки агрегации.
- Перед удалением агрегированного канала перенесите или удалите связанные VLAN.

Трансляция адреса источника в исходящих соединениях (SNAT)

При TLS-инспекции Амибиг завершает клиентское соединение и устанавливает отдельное origin-соединение с целевым сервером. Режим SNAT определяет, какой адрес источника увидит сервер в этом соединении. Настройка доступна на экране **Система**, в разделе «Egress (трансляция адреса источника)», и через REST API (секция `egress`: ключи `origin-snat`, `origin-source-port`, `snat-pool`). Это параметр обработки трафика, а не настройка DPDK.

Для отдельного правила на экране **Политики** доступно переопределение обоих режимов — правило без переопределения наследует глобальное значение.

The screenshot shows the Amibig configuration interface. On the left is a sidebar with navigation links: 'Одиночный узел', 'Обзор', 'Трафик', 'Система', 'Сеть', 'SSL/TLS', 'Топология', 'Инспекция', 'Мониторинг', and 'Управление'. The main panel is titled 'NTP' and contains several sections: 'Серверы' with a 'Добавить NTP-серверы' button; 'Dataplane (DPDK)' with fields for 'Путь к dataplane config' (set to '/config/dataplane.conf') and 'Hugepages (МБ)' (set to '2048'), and a 'Lcore Mask' field (set to '0xfe'); 'Протокол (TCP / MSS)' with 'MSS clamp (глобальный)' (set to '1360') and 'MSS clamp: клиентское соединение' (set to '0'), both with '0 — выводить из MTU' and '0 — выключено' options; 'Proxy-MSS (динамический MSS origin-соединения)' and 'Обнаружение «чёрной дыры» PMTUD' (both with toggle switches); 'Egress (трансляция source-адреса)' with 'Трансляция source-адреса (SNAT)' (set to 'automap — self-IP устройства') and 'Source-порт origin-соединения' (set to 'preserve — порт клиента, при конфликте новый'); 'Адреса SNAT-пула' with a 'Добавить IP-адрес' button; and 'Привязка CPU' with a toggle switch 'Включить привязку CPU'. At the bottom are buttons for 'Сбросить', 'Отмена', and 'Сохранить'.

Режимы трансляции адреса (origin-snat)

Режим выбирается из трёх вариантов. По умолчанию используется `automap` — он не требует дополнительной настройки и подходит для большинства внедрений.

- **automap** — исходящее соединение устанавливается с собственного адреса устройства (Self-IP). Ответ целевого сервера возвращается на Амибиг штатной маршрутизацией, поэтому режим работает и тогда, когда origin-сервер находится за шлюзом.
- **none** — трансляция не выполняется: целевой сервер видит адрес клиента. Ответ вернётся на Амибиг только если устройство стоит на пути возврата трафика. Если целевой сервер доступен только через шлюз и путь возврата идёт мимо устройства, соединение не установится — предупреждение об этом выдаётся уже на этапе применения конфигурации. Во время работы Амибиг регистрирует такие попытки в журнале и завершает соединение.
- **pool** — адрес источника выбирается из списка `snat-pool`. Список обязан содержать хотя бы один адрес; выбор адреса для соединения детерминирован по адресу клиента.

Режимы трансляции порта источника (`origin-source-port`)

Режим определяет, какой порт источника используется в исходящем соединении. По умолчанию используется `preserve`.

- **`preserve`** — использовать порт клиента; если он занят, автоматически выбрать новый. Каждый такой переход учитывается счётчиком `origin_snat_port_exhaustion` в телеметрии.
- **`preserve-strict`** — использовать строго порт клиента, без запасного варианта. Вместе с `origin-snat: none` даёт полное совпадение адреса и порта клиента в исходящем соединении.
- **`change`** — всегда выбирать новый порт. Этот режим сохраняет распределение соединений по рабочим потокам подсистемы обработки трафика и рекомендуется при работе в несколько рабочих потоков.

Трансляция UDP и ICMP

Режимы трансляции действуют не только на перехватываемые TCP-соединения. Для транзитных UDP-датаграмм и ICMP Echo по IPv4 Амибиг применяет тот же режим `origin-snat`: и при прохождении через сервисную цепочку, и при её обходе. Без трансляции пакеты уходят к целевому серверу с внутренним адресом клиента, поэтому обратный трафик может пройти в обход Амибиг.

Ответные датаграммы, пришедшие на адрес трансляции, транслируются обратно и доставляются клиенту. Это относится и к ICMP-ошибкам, которые сеть отправляет со стороны целевого сервера: `destination unreachable` (включая `fragmentation needed` для PMTUD) и `time exceeded`. Амибиг находит по вложенной в ошибку датаграмме исходный поток, переписывает адреса в ошибке и доставляет её клиенту.

Прочие протоколы, кроме TCP, UDP и ICMP Echo, передаются без трансляции адреса, как в режиме `none`.

Соответствие настройкам F5

Для администраторов, переносящих конфигурацию с F5 BIG-IP (SSL Orchestrator), режимы Амибиг соответствуют настройкам F5 следующим образом.

Задача	F5	Амибиг
Адрес устройства (self-IP)	SNAT Auto Map	<code>origin-snat: automap</code> (по умолчанию)
Полное сохранение адреса и порта клиента	<code>source-address-translation: none + source-port: preserve-strict</code>	<code>origin-snat: none + origin-source-port: preserve-strict</code>

Задача	F5	Амибиг
Адрес из пула	SNAT Pool	<code>origin-snat: pool + список snat-pool</code>

В отличие от F5 SSL Orchestrator, где выбор действует на всю топологию, Амибиг позволяет переопределить оба режима для отдельного правила политики.

Предотвращение исчерпания портов

Один адрес источника ограничен числом эфемерных портов. При большом количестве одновременных соединений к одному целевому серверу свободные порты заканчиваются, и новые соединения не устанавливаются. Пул из нескольких адресов увеличивает доступное число комбинаций адреса и порта. Ситуация, когда запрошенный порт оказался занят и был заменён, фиксируется счётчиком `origin_snat_port_exhaustion`; по порогу этого счётчика настраивается правило оповещения типа `port-exhaustion` на экране **Наблюдаемость**.

Требования для отказоустойчивой пары

В кластере высокой доступности список адресов `snat-pool` должен быть одинаковым на обоих узлах. Иначе после переключения исходящие соединения пойдут с других адресов, и origin-сервер или межсетевой экран за устройством увидят смену адреса отправителя. В режиме `automap` используется виртуальный IP-адрес кластера, поэтому адрес источника при переключении не меняется.

Параметры, требующие перезапуска

Секция `egress` — режимы `origin-snat`, `origin-source-port` и список `snat-pool` — применяется при перезапуске подсистемы обработки трафика. После изменения этих параметров необходимо перезапустить службу `tforged` или устройство. Уже установленные соединения при перезапуске завершаются, поэтому изменение режима трансляции планируйте на окно обслуживания.

Управление TCP MSS и MTU

Амибиг устанавливается в разрыв трафика и завершает TCP-соединения, поэтому устройство согласует размер сегментов отдельно для клиентского и origin-соединения. Если на пути к клиенту или целевому серверу есть участок с уменьшенным MTU (туннели, VPN, спутниковые каналы), сегменты полного размера теряются, и соединение может остановиться на TLS handshake или передаче данных. Подсистема обработки трафика на F-Stack/DPDK не получает ICMP-сообщения `fragmentation needed`, поэтому Амибиг ограничивает объявляемое значение MSS вместо определения MTU пути (PMTUD). Настройка доступна на экране **Система**, в разделе «Протокол (TCP / MSS)», и через REST API (секция `protocol`); MTU отдельного порта — в окне настроек порта на экране **Сеть**.

Амибиг

Одиночный узел

- Обзор
- Трафик
- Система**
- Сеть
- SSL/TLS
- Топология
- Инспекция
- Мониторинг
- Управление

Применить конфигурацию

Документация

admin

Система
Глобальные настройки системы

Основные

Имя хоста: amibig-proxy

Часовой пояс: UTC

Управление

Способ получения адреса: Статический адрес

Адрес: 192.168.122.230/24

Укажите адрес с префиксом и шлюз для интерфейса управления.

Шлюз: 192.168.122.1

DNS

DNS-серверы: 8.8.8.8

NTP

Серверы: Добавить NTP-серверы

Dataplane (DPDK)

Путь к dataplane config: /config/dataplane.conf

Hugepages (MB): 2048

Lcore Mask: 0xfe

Протокол (TCP / MSS)

MSS clamp (глобальный): 1360

0 — выводить из MTU

MSS clamp: клиентское соединение: 0

0 — выключено

MSS clamp: origin-соединение: 0

0 — выключено

☒ Proxy-MSS (динамический MSS origin-соединения)

Общее ограничение MSS (tcp-mss-clamp)

Ограничивает MSS, анонсируемый обеим сторонам соединения. По умолчанию — **1360**: это значение учитывает служебные данные распространённых туннельных протоколов при MTU 1500. Значение **0** отключает ограничение — тогда MSS рассчитывается из MTU порта и маршрута. Допустимый диапазон — 536..1460.

Ограничения по направлениям (tcp-mss-clamp-client , tcp-mss-clamp-origin)

Ограничивают MSS отдельно для клиентского соединения (анонс в SYN-ACK клиенту) и для origin-соединения (анонс в исходящем SYN). По умолчанию оба значения — **0**, то есть выключены; общее ограничение при этом продолжает действовать. Раздельные значения нужны, когда уменьшенный MTU есть только с одной стороны устройства — например, туннель только на соединении с origin-серверами.

Согласование MSS между соединениями (proxy-mss)

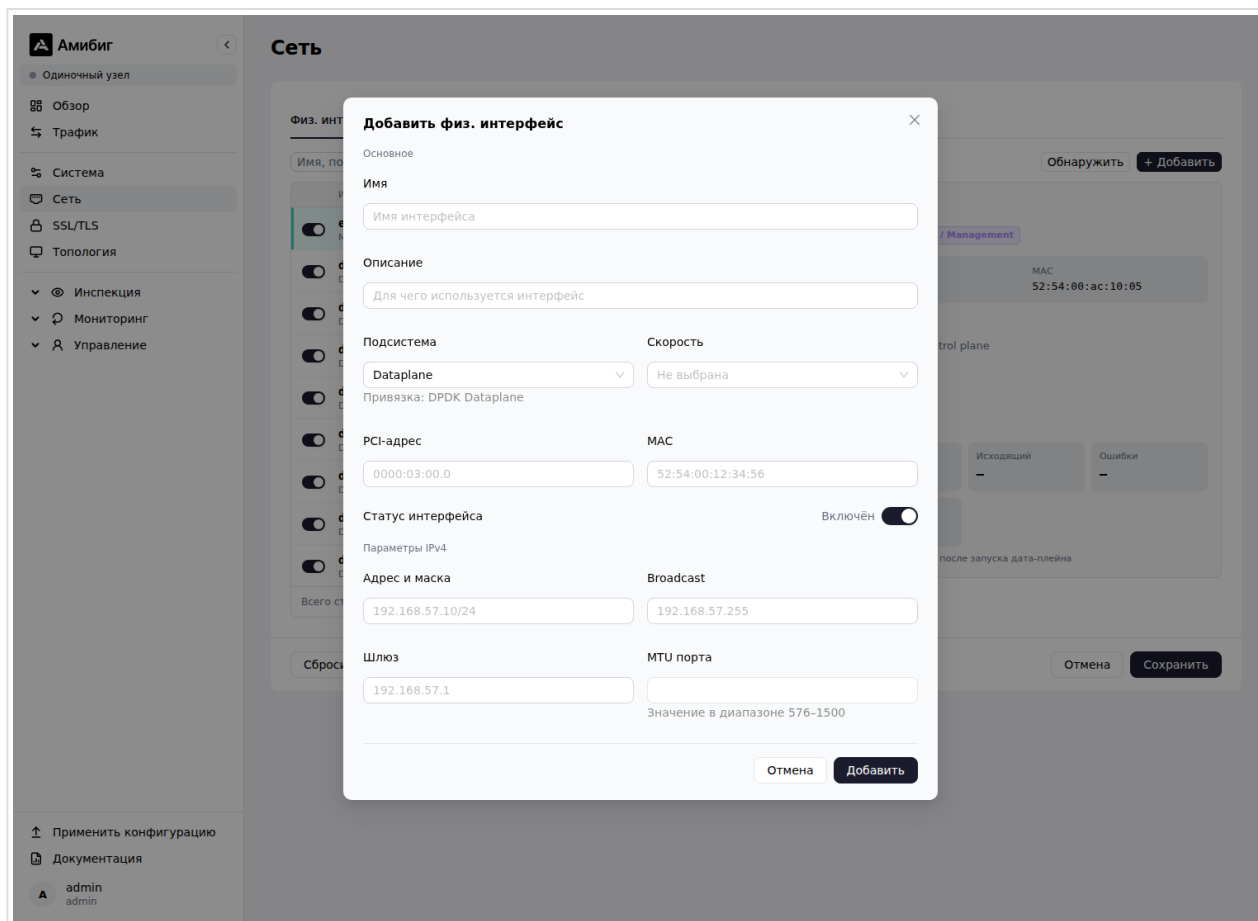
По умолчанию параметр включён: целевому серверу объявляется меньшее из значений MSS клиентского и origin-соединения. Это исключает ситуацию, когда целевой сервер отправляет сегменты крупнее, чем может принять клиентская сторона. Параметр соответствует настройке `proxy-mss` профиля TCP в F5.

Обнаружение сбоя PMTUD (`pmtud-blackhole-detect`)

По умолчанию **включено**. Если сегменты повторно теряются без ICMP-ответа, что указывает на сбой PMTUD, эффективный MSS соединения временно снижается до **1200** и восстанавливается после нормализации. Этот механизм не требует дополнительной настройки.

MTU порта

MTU отдельного порта обработки трафика задаётся полем «MTU» в окне настроек порта на экране **Сеть** (ключ `network.interface.mtu`). Диапазон — 576..1500, значение может только понижать MTU относительно физического; значения по умолчанию нет — незаполненное поле оставляет MTU интерфейса по умолчанию (1500). При заданном MTU анонсируемый TCP MSS для соединений через порт равен `mtu - 40`. Подробнее о портах — в разделе Порты.



Соответствие настройкам F5

Для администраторов, переносящих конфигурацию с F5 BIG-IP, настройки Амибиг соответствуют параметрам профиля TCP следующим образом.

Задача	F5	Амибиг
Фиксированное ограничение MSS	<code>mss-override</code> в профиле TCP	<code>tcp-mss-clamp</code>
Разные значения по направлениям	clientside/serverside профили TCP	<code>tcp-mss-clamp-client</code> / <code>tcp-mss-clamp-origin</code>
Анонс минимума двух ног	<code>proxy-mss</code> в профиле TCP	<code>proxy-mss</code> (по умолчанию включено)
Обработка сбоя PMTUD	переменные БД <code>tm.pathmtudiscovery</code>	<code>pmtud-blackhole-detect</code> (по умолчанию включено)

Параметры, требующие перезапуска

Секция `protocol` читается при запуске подсистемы обработки трафика. После изменения ограничений MSS, `proxy-mss` или `pmtud-blackhole-detect` необходимо перезапустить службу `tforged` или устройство. MTU порта также применяется при перезапуске `tforged` вместе с остальными параметрами портов.

SSL/TLS

Раздел описывает настройки SSL/TLS-инспекции: сертификаты, доверенные CA, наборы шифров, клиентские и серверные профили. Эти параметры определяют, как Амибиг завершает TLS-соединение клиента, проверяет целевой сервер и выпускает динамический MITM-сертификат.

Перед включением инспекции распространите CA-сертификат Амибиг на клиентские устройства и проверьте политику проверки сертификатов целевых серверов.

Аппаратное ускорение TLS (Intel QAT)

Установление TLS-соединения требует ресурсоёмких операций асимметричной криптографии (подпись и проверка RSA/ECDSA, согласование ключей ECDH). На большом потоке новых соединений именно эти операции ограничивают число соединений в секунду. Амибиг умеет переносить их на встроенный аппаратный ускоритель Intel QuickAssist (QAT), освобождая процессорные ядра под обработку трафика. Ускорение включается параметром `acceleration-mode` серверного профиля SSL и по умолчанию выключено — базовый режим всегда безопасен и не зависит от наличия ускорителя.

Параметр `acceleration-mode` принимает три значения:

Значение	Поведение
<code>cpu</code>	Вся криптография TLS считается на процессоре (значение по умолчанию).
<code>auto</code>	Использовать ускоритель при его наличии; если стек ускорителя недоступен — автоматически продолжить на процессоре.
<code>qat</code>	Требовать ускоритель. Если стек недоступен, обработка TLS всё равно продолжается на процессоре, но состояние помечается как заблокированное для диагностики.

Требования к оборудованию

Для аппаратного ускорения нужен процессор Intel Xeon со встроенным ускорителем QuickAssist четвёртого поколения (например, Xeon 6787P) либо совместимая PCIe-карта QAT. В прошивке материнской платы должна быть включена аппаратная виртуализация ввода-вывода (VT-d); соответствующие

компоненты (драйвер ядра, микрокод, служебные библиотеки) входят в состав образа Амибиг.

Проверка состояния

Итоговое решение о выборе криптопути Амибиг записывает в файл `/config/tls-acceleration-runtime.json` и в стартовую строку журнала подсистемы `dataplane`. Возможны четыре состояния:

Состояние	Значение
<code>supported</code>	Запрошен режим <code>qat</code> , стек ускорителя обнаружен, выбран аппаратный путь.
<code>fallback</code>	Запрошен режим <code>auto</code> , стек недоступен — обработка продолжается на процессоре.
<code>blocked</code>	Запрошен режим <code>qat</code> , стек недоступен. Это не прерывает работу: обработка TLS продолжается на процессоре, состояние служит сигналом администратору проверить оборудование и стек.
<code>disabled</code>	Запрошен режим <code>cpu</code> — ускорение не используется.

Поведение при сбое ускорителя

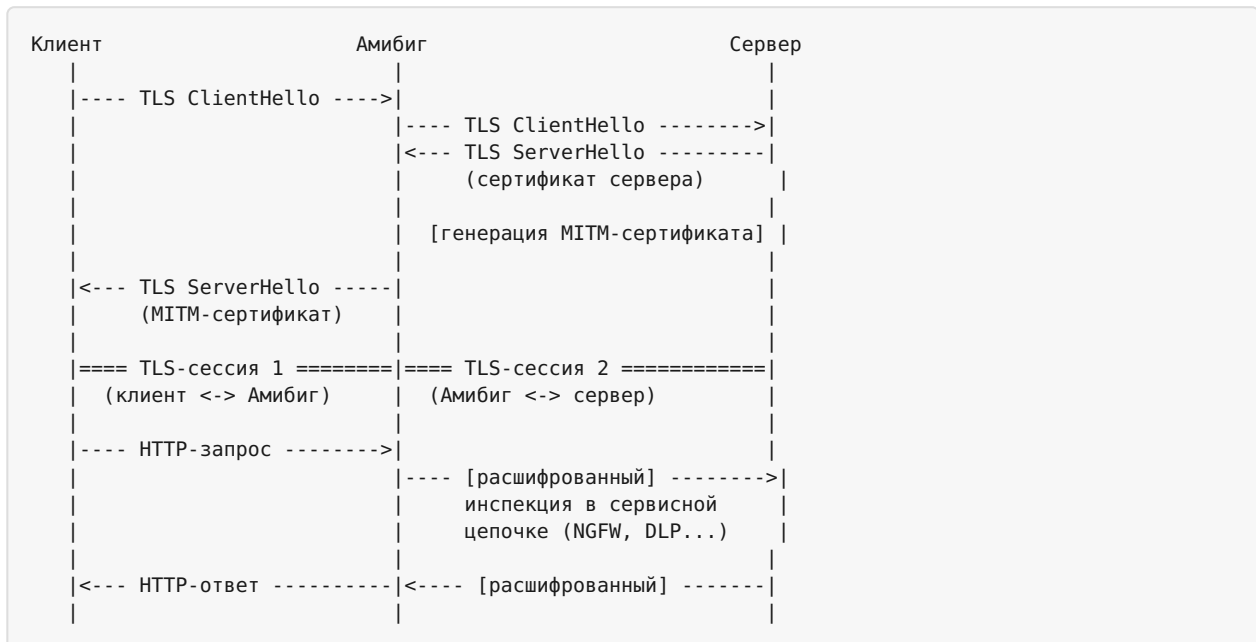
Если во время работы операция на ускорителе завершается ошибкой или очередь заданий переполнена, Амибиг автоматически выполняет эту операцию на процессоре, не разрывая устанавливаемое соединение. Число таких переходов отражается в телеметрии, а при массовой деградации в журнал записывается событие `AMB-3022` «Ускорение TLS (QAT) деградировало — операции выполняются на CPU». Служебные параметры очереди заданий ускорителя система задаёт автоматически при применении конфигурации; ручная настройка не требуется.

Обзор SSL/TLS

Амибиг выполняет роль SSL/TLS-прокси с функцией Man-in-the-Middle (MITM) — перехватывает зашифрованный трафик пользователей, расшифровывает его для инспекции средствами ИБ (NGFW, DLP, ICAP) и повторно шифрует перед отправкой на целевой сервер. Это позволяет применять политики безопасности к содержимому HTTPS-соединений, которые иначе были бы недоступны для анализа.

Принцип работы MITM-инспекции

При прохождении TLS-соединения через Амибиг устройство разделяет одно сквозное TLS-соединение на два независимых:



Последовательность взаимодействия

Источник	Связь	Назначение
Клиент, Амибиг	Этап	Клиентская TLS-сессия
Клиент	TLS ClientHello	Амибиг
Амибиг	Этап	Генерация MITM-сертификата (подпись УЦ Амибиг)
Амибиг	TLS ServerHello + MITM-сертификат	Клиент
Амибиг, Сервер	Этап	Серверная TLS-сессия
Амибиг	TLS ClientHello	Сервер
Сервер	TLS ServerHello + сертификат сервера	Амибиг
Клиент, Амибиг	Этап	TLS-сессия 1 (клиент — Амибиг)
Амибиг, Сервер	Этап	TLS-сессия 2 (Амибиг — сервер)
Клиент	HTTP-запрос (зашифрован)	Амибиг
Амибиг	Этап	Расшифрование + инспекция (NGFW, DLP)

Источник	Связь	Назначение
Амибиг	HTTP-запрос (повторно зашифрован)	Сервер
Сервер	HTTP-ответ (зашифрован)	Амибиг
Амибиг	Этап	Расшифрование + инспекция
Амибиг	HTTP-ответ (повторно зашифрован)	Клиент

Последовательность обработки

Последовательность охватывает установление двух TLS-соединений, инспекцию открытого трафика и его передачу получателю.

- 1. Перехват соединения.** Клиентское TLS-соединение перехватывается прозрачно. Амибиг принимает TCP-соединение и начинает TLS handshake от имени целевого сервера.
- 2. Подключение к серверу.** Параллельно Амибиг устанавливает собственное TLS-соединение с целевым сервером, используя параметры серверного профиля (проверка сертификата, набор шифров, версия TLS).
- 3. Генерация MITM-сертификата.** На основе сертификата, полученного от реального сервера, Амибиг динамически генерирует подменный сертификат. Этот сертификат подписывается локальным CA-сертификатом из клиентского профиля.
- 4. Завершение TLS handshake с клиентом.** Клиент получает MITM-сертификат и завершает TLS handshake. Если CA-сертификат Амибиг установлен в доверенное хранилище клиента, соединение считается доверенным.
- 5. Инспекция трафика.** Расшифрованный трафик проходит через сервисную цепочку — NGFW, DLP, ICAP-сервисы — в соответствии с политикой.
- 6. Повторное шифрование.** После инспекции трафик повторно шифруется и передаётся по назначению.

Роль профилей SSL

Конфигурация SSL/TLS в Амибиг разделена на два профиля, соответствующих двум сторонам MITM-соединения:

Клиентский профиль (client-profile)

Управляет TLS-соединением между **клиентом и Амибиг** (внутренняя сторона):

- Какой CA-сертификат используется для подписи генерируемых MITM-сертификатов
- Какие шифры и версии TLS предлагаются клиенту
- Как обрабатываются расширения сертификата (SAN, Basic Constraints)
- Режим ALPN-согласования (HTTP/2, HTTP/1.1)
- Политика для не-SSL трафика и обхода forward proxy

Подробнее: Профиль клиента

Серверный профиль (server-profile)

Управляет TLS-соединением между **Амибиг и целевым сервером** (внешняя сторона):

- Какой набор доверенных CA используется для проверки сертификата сервера
- Включена ли проверка подлинности сервера (verify-peer)
- Какие шифры и версии TLS используются при подключении к серверу
- Политика реагирования на ошибки сертификатов (просроченный, недоверенный, отозванный)

Подробнее: Профиль сервера

Цепочка сертификатов

Для корректной работы MITM-инспекции необходимо обеспечить доверие к генерируемым сертификатам на стороне клиентов.

Структура доверия

Клиент доверяет сформированному сертификату через УЦ инспекции, сертификат которого установлен в доверенном хранилище.

```

Корневой СА организации
|
+-- УЦ инспекции Амибиг (подписывающий сертификат)
    |
    +-- Динамический leaf-сертификат (для каждого сайта)
  
```

Требования

Для инспекции необходимы сертификат и закрытый ключ УЦ, а также набор доверенных УЦ для проверки origin-серверов.

Компонент	Расположение	Назначение
CA-сертификат (cert)	Указывается в <code>cert-path</code> записи типа <code>ca</code>	Подпись MITM-сертификатов
CA-ключ (key)	Указывается в <code>key-path</code> записи типа <code>ca</code>	Приватный ключ для подписи
Доверенный CA-бандл	Указывается в <code>cert-path</code> записи типа <code>bundle</code>	Проверка сертификатов серверов
CA-сертификат на клиентах	Хранилище доверенных сертификатов ОС/браузера	Доверие к MITM-сертификатам

Распространение CA-сертификата

Чтобы клиенты не получали предупреждения о недоверенном сертификате, CA-сертификат Амибиг необходимо распространить на все рабочие станции.

Типичные способы:

- **Active Directory GPO** — автоматическое распространение через групповые политики домена
- **MDM-системы** — для мобильных устройств (Intune, JAMF)
- **Ручная установка** — для небольших сетей или тестовых сред

Внимание. Без установки CA-сертификата на клиентских устройствах пользователи будут видеть предупреждения браузера о недоверенном соединении. Некоторые приложения с certificate pinning (банковские приложения, мессенджеры) могут полностью отказать в соединении — для них следует настроить правила обхода (bypass) в политике.

Обход инспекции (bypass)

Не весь трафик подлежит MITM-инспекции. Амибиг поддерживает несколько механизмов обхода:

Механизм	Описание
Правила политики с <code>action: bypass</code>	Обход по SNI, IP-адресу источника/назначения, порту
<code>allow-non-ssl</code> в клиентском профиле	Пропуск не-TLS трафика без разрыва соединения
<code>forward-proxy-bypass</code>	Обход для соединений, не поддерживающих MITM

Подробнее о настройке обхода: Политики

Поддерживаемые версии TLS

Поддерживаемый диапазон охватывает TLS 1.0–1.3; для рабочей среды рекомендуется использовать TLS 1.2 или TLS 1.3.

Версия	Поддержка	Примечание
TLS 1.0	Да	Устаревший, не рекомендуется
TLS 1.1	Да	Устаревший, не рекомендуется
TLS 1.2	Да	Рекомендуемый минимум
TLS 1.3	Да	Требуется лицензию Standard или Premium

Минимальная и максимальная версии TLS настраиваются в правилах шифрования и применяются через клиентский и серверный профили.

Перехват ГОСТ TLS

Помимо обычного TLS Амибиг выполняет полный перехват ГОСТ-TLS: принимает соединение от ГОСТ-клиента, устанавливает собственное исходящее соединение к origin-серверу по ГОСТ-алгоритмам и формирует подменный сертификат, подписанный ГОСТ-УЦ инспекции. Выбор удостоверяющего центра выполняется для каждого соединения по криптографии клиента: обычный клиент получает сертификат от обычного СА, ГОСТ-клиент — от ГОСТ-УЦ. По умолчанию перехват ГОСТ выключен; ГОСТ-соединения при этом пропускаются без инспекции.

Подробнее: ГОСТ TLS

Связанные разделы

Настройка сертификатов, профилей и алгоритмов шифрования описана в связанных разделах руководства.

- Управление сертификатами — загрузка СА-сертификатов и ключей
- Наборы шифров — настройка допустимых алгоритмов шифрования
- Профиль клиента — параметры TLS-соединения на стороне клиента
- Профиль сервера — параметры TLS-соединения на стороне сервера
- ГОСТ TLS — перехват ГОСТ-TLS и настройка ГОСТ-УЦ

Управление сертификатами

Сертификаты необходимы для SSL/TLS-инспекции в Амибиг. Система использует CA-сертификат для подписи динамически генерируемых сертификатов и набор доверенных УЦ для проверки подлинности origin-серверов. Настроить сертификаты можно через веб-интерфейс или REST API.

Типы сертификатов

Амибиг поддерживает три типа записей сертификатов:

Тип	Значение type	Назначение	Требуется ключ
CA-сертификат	ca	Подпись MITM-сертификатов при перехвате TLS	Да
Серверный сертификат	server	Предъявление клиентам (зарезервировано)	Да
Бандл доверенных CA	bundle	Проверка сертификатов целевых серверов	Нет

CA-сертификат

CA-сертификат и соответствующий приватный ключ используются Амибиг для динамической генерации подменных сертификатов при MITM-инспекции. Для каждого перехваченного TLS-соединения Амибиг создаёт сертификат с CN/SAN целевого сервера, подписанный этим CA.

Требования к CA-сертификату:

- Расширение Basic Constraints: CA:TRUE
- Расширение Key Usage: Certificate Sign, CRL Sign
- Формат: PEM (Base64-кодированный X.509)
- Рекомендуемый алгоритм ключа: RSA 2048+ или ECDSA P-256/P-384
- Рекомендуемый срок действия: 5–10 лет

Бандл доверенных CA

Файл, содержащий набор корневых и промежуточных CA-сертификатов, которым Амибиг доверяет при проверке сертификатов целевых серверов. Аналогичен хранилищу ca-certificates в операционных системах.

Основной сценарий в текущей версии — автоматическая сборка доверенного набора: базовый набор прошивки объединяется с загруженными сертификатами, из результата вычитаются исключения, а итоговый файл записывается по пути записи-цели. Ручная запись типа `bundle` при этом остаётся как цель сборки — в её файл движок инспекции и читает готовый набор. Удаление такой записи, пока на неё ссылается серверный профиль (`server-profile.trusted-ca`), блокируется, чтобы сборка не осталась без файла-цели. Подробнее — на странице Доверенные УЦ.

ГОСТ-УЦ

Удостоверяющий центр для перехвата ГОСТ-TLS загружается через тот же список сертификатов записью типа `ca` : сертификат должен содержать ключ ГОСТ Р 34.10-2012 — система проверяет это при загрузке. На запись ссылается по имени параметр `ssl.gost-ca` . Подробнее: ГОСТ TLS.

Формат конфигурации

Сертификаты описываются в секции `ssl.certificates` конфигурационного файла:

```
{
  "ssl": {
    "certificates": {
      "certificate": [
        {
          "name": "mitm-ca",
          "type": "ca",
          "cert-path": "mitm-ca.crt",
          "key-path": "mitm-ca.key",
          "description": "MITM CA for forward proxy"
        },
        {
          "name": "trusted-ca-bundle",
          "type": "bundle",
          "cert-path": "ca-bundle.crt",
          "description": "Trusted CAs for upstream verification"
        }
      ]
    }
  }
}
```

Параметры записи сертификата

Параметры записи связывают логическое имя сертификата с файлами сертификата и закрытого ключа на устройстве.

Параметр	Тип	Обязательный	Описание
<code>name</code>	string	Да	Уникальное имя записи. Используется для ссылок из профилей

Параметр	Тип	Обязательный	Описание
<code>type</code>	string	Да	Тип сертификата: <code>ca</code> , <code>server</code> или <code>bundle</code>
<code>cert-path</code>	string	Да	Путь к файлу сертификата на устройстве
<code>key-path</code>	string	Нет	Путь к файлу приватного ключа. Обязателен для типов <code>ca</code> и <code>server</code>
<code>description</code>	string	Нет	Текстовое описание назначения сертификата

Загрузка сертификатов

Сертификаты можно загрузить через веб-интерфейс или REST API; в обоих случаях Амибиг проверяет формат перед сохранением файла.

Через веб-интерфейс

Для загрузки создайте запись сертификата и укажите файлы сертификата и закрытого ключа.

1. Перейдите в раздел **SSL/TLS > Сертификаты**.
2. Нажмите **Добавить сертификат**.
3. Укажите имя, тип и описание.
4. Загрузите файл сертификата (`.cert`, `.pem`) и ключа (`.key`, `.pem`).
5. Сохраните запись и примените конфигурацию.

Файл сертификата должен быть в формате PEM (Base64-кодированный X.509). Файл в формате DER отклоняется с сообщением и подсказкой команды конвертации:

```
openssl x509 -inform DER -in cert.der -out cert.pem -outform PEM
```

Формат DER не конвертируется автоматически, потому что и движок инспекции, и сборщик доверенного набора читают только PEM: принятый молча DER лёг бы в хранилище нерабочим файлом.

Бандл (несколько сертификатов в одном файле) проверяется целиком, по каждому PEM-блоку. Если сломан второй или последующий блок, в тексте отказа указывается его номер — иначе бандл со сломанным вторым блоком прошёл бы

проверку, потому что стандартные средства разбирают только первый сертификат файла.

Предпросмотр «Разбор сертификата»

После выбора файла и по щелчку на записи в списке веб-интерфейс показывает разбор сертификата:

- субъект;
- издатель;
- срок действия «с — по» (даты в русской локале);
- серийный номер;
- отпечаток SHA-256;
- тип и длина ключа;
- альтернативные имена (SAN);
- признак «Просрочен».

Кириллица в субъекте российских удостоверяющих центров отображается корректно. Разбор помогает убедиться, что загружен именно тот сертификат, прежде чем сохранять запись и применять конфигурацию.

Через REST API

Загрузка файлов сертификата и ключа:

```
curl -X POST "https://<management-ip>/api/v1/amitls/upload-cert" \
  -H "Authorization: Bearer $TOKEN" \
  -F "cert_path=mitm-ca.crt" \
  -F "key_path=mitm-ca.key" \
  -F "cert_file=@mitm-ca.crt" \
  -F "key_file=@mitm-ca.key"
```

Получение информации о загруженном сертификате:

```
curl "https://<management-ip>/api/v1/amitls/cert-info?cert_path=mitm-ca.crt" \
  -H "Authorization: Bearer $TOKEN"
```

Ответ содержит те же поля разбора, что и предпросмотр в веб-интерфейсе:

```
{
  "exists": true,
  "valid": true,
  "subject": "CN=AMIBIG MITM CA, O=Example Corp",
  "issuer": "CN=AMIBIG MITM CA, O=Example Corp",
  "not_before": "Jan 1 00:00:00 2026 GMT",
  "expiry": "Dec 31 23:59:59 2030 GMT",
  "serial": "1A2B3C4D",
  "fingerprint_sha256": "9f2c...e14a",
  "key_type": "RSA",
  "key_bits": 4096,
  "san": ["DNS:example.com", "DNS:www.example.com"],
```

```
"expired": false
}
```

Применение СА-сертификата

После загрузки файлов необходимо применить СА-сертификат к движку инспекции:

```
curl -X POST "https://<management-ip>/api/v1/amitls/apply" \
-H "Authorization: Bearer $TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "cert_path": "mitm-ca.crt",
  "key_path": "mitm-ca.key"
}'
```

Эта команда обновляет конфигурацию движка инспекции и перезапускает его.

Обе операции фиксируются в журнале событий: загрузка сертификата — событием `AMB-5010`, применение СА-сертификата к движку инспекции — событием `AMB-5011`. Полный перечень кодов приведён в разделе [Каталог кодов сообщений](#).

Хранение сертификатов на устройстве

Загруженные сертификаты и ключи хранятся в персистентном хранилище Амибиг и сохраняются между перезагрузками. Права доступа к приватным ключам устанавливаются системой автоматически при загрузке через веб-интерфейс или REST API.

Переименование записи

При переименовании записи путь к файлу на устройстве не пересчитывается по новому имени, пока файл не загружен заново: сохранённый путь остаётся прежним. Так переименование не уводит конфигурацию на несуществующий файл.

Если на переименованную запись ссылались роли (`client-profile.ca-certificate`, `server-profile.trusted-ca`, `gost-ca`), ссылки автоматически переносятся на новое имя, и в веб-интерфейсе выводится уведомление о том, какие именно ссылки обновлены. Это исключает висячие ссылки, которые иначе сломали бы применение конфигурации.

Контроль срока действия

Срок действия сертификатов отслеживается подсистемой оповещений. Предустановленное правило `default-certificate-expiry` формирует

предупреждение за 30 дней до истечения срока действия сертификата и критическое оповещение — за 7 дней. Пороги правила можно изменить на экране наблюдаемости или через REST API.

Динамическая генерация MITM-сертификатов

При перехвате TLS-соединения Амибиг генерирует leaf-сертификат для каждого целевого домена:

Процесс генерации

Для каждого нового домена Амибиг получает параметры сертификата origin-сервера, создаёт сертификат для клиента и сохраняет его в кеше.

- 1. Извлечение информации.** Амибиг подключается к целевому серверу и получает его оригинальный сертификат.
- 2. Копирование атрибутов.** Из оригинального сертификата извлекаются:
 - Common Name (CN)
 - Subject Alternative Names (SAN)
 - Расширения, указанные в параметре `cert-extensions` клиентского профиля
- 3. Установка сроков действия.** Срок жизни задаётся параметром `cert-lifespan` клиентского профиля (в днях): дата начала действия выставляется на сутки раньше текущего момента, дата окончания — на заданный срок вперёд.
- 4. Подпись.** Сертификат подписывается СА-ключом из клиентского профиля. Если для подписи используется подчинённый СА, промежуточный сертификат передаётся клиенту во время TLS handshake и сохраняется при горячем применении конфигурации.
- 5. Кэширование.** Сгенерированный сертификат кэшируется в памяти для повторного использования при последующих соединениях к тому же домену.

Расширения сертификата

Параметр `cert-extensions` клиентского профиля определяет, какие расширения X.509 копируются из оригинального сертификата в генерируемый MITM-сертификат:

Расширение	Значение в конфигурации	Описание
Basic Constraints	<code>basic-constraints</code>	Ограничения CA/non-CA
Subject Alternative Name	<code>subject-alternative-name</code>	Альтернативные имена (DNS, IP)
Extended Key Usage	<code>extended-key-usage</code>	Назначение ключа (serverAuth, clientAuth)

Генерация CA-сертификата

Если у вашей организации нет подходящего CA-сертификата, его можно создать с помощью OpenSSL:

Генерация самоподписанного CA

Команды создают закрытый ключ RSA и самоподписанный CA-сертификат сроком действия десять лет.

```
### Создание приватного ключа CA (RSA 4096)
openssl genrsa -out mitm-ca.key 4096

### Создание самоподписанного CA-сертификата (10 лет)
openssl req -new -x509 -days 3650 -key mitm-ca.key \
  -out mitm-ca.crt \
  -subj "/C=RU/O=Example Corp/CN=AMIBIG MITM CA"
```

Генерация подчинённого CA

Если в организации существует корпоративный PKI, рекомендуется выпустить подчинённый CA:

```
### Создание приватного ключа
openssl genrsa -out mitm-ca.key 4096

### Создание запроса на подпись (CSR)
openssl req -new -key mitm-ca.key -out mitm-ca.csr \
  -subj "/C=RU/O=Example Corp/CN=AMIBIG MITM Subordinate CA"

### Подпись CSR корневым CA организации (выполняется администратором PKI)
openssl x509 -req -in mitm-ca.csr \
  -CA root-ca.crt -CAkey root-ca.key -CAcreateserial \
  -days 1825 -out mitm-ca.crt \
  -extensions v3_ca \
  -extfile <(echo "[v3_ca]
basicConstraints = critical,CA:TRUE,pathlen:0
keyUsage = critical,keyCertSign,cRLSign")
```

Рекомендация. Использование подчинённого CA предпочтительнее самоподписанного: если корневой CA организации уже установлен на рабочих станциях, дополнительное распространение сертификатов не потребуется.

При подписи подменных сертификатов подчинённым CA Амибиг передаёт клиенту промежуточный сертификат во время TLS handshake. Клиент выстраивает цепочку доверия до корневого CA организации. Цепочка сохраняется и при горячем применении конфигурации: перевыпускать сертификаты и переустанавливать доверие на клиентах не требуется.

Проверка сертификата

Для проверки корректности загруженного сертификата:

```
### Информация о сертификате
openssl x509 -in mitm-ca.crt -noout -text

### Проверка соответствия ключа и сертификата
openssl x509 -in mitm-ca.crt -noout -modulus | openssl md5
openssl rsa -in mitm-ca.key -noout -modulus | openssl md5
### Хэши должны совпадать
```

Связанные разделы

Дополнительные параметры доверия и генерации сертификатов описаны в связанных разделах руководства.

- Обзор SSL/TLS — принцип работы MITM-инспекции
- Профиль клиента — выбор CA-сертификата и параметры генерации
- Профиль сервера — настройка доверенного бандла CA
- ГОСТ TLS — загрузка ГОСТ-УЦ и перехват ГОСТ-TLS

Наборы шифров (Cipher Rules)

Правила шифрования (cipher rules) определяют допустимые алгоритмы и версии TLS. Именованное правило содержит единый набор криптографических параметров, на который могут ссылаться клиентский и серверный профили.

Формат конфигурации

Правила описываются в секции `ssl.cipher-rules`:

```
{
  "ssl": {
    "cipher-rules": {
      "rule": [
        {
          "name": "default",
          "ciphers": "ECDHE+AESGCM:ECDHE+CHACHA20:DHE+AESGCM:!aNULL:!MD5:!DSS",
          "min-version": "TLS1.2",
          "max-version": "TLS1.3",
          "description": "Default secure ciphers"
        },
        {
          "name": "strict",
          "ciphers": "ECDHE+AESGCM:!RSA:!SHA1",
          "min-version": "TLS1.2",
          "max-version": "TLS1.3",
          "description": "Strict compliance ciphers"
        }
      ]
    }
  }
}
```

```

    }
  }
}

```

Параметры правила

Параметры правила задают строку шифров для TLS 1.2 и более ранних версий, отдельные шифронаборы TLS 1.3 и допустимый диапазон версий протокола.

Параметр	Тип	По умолчанию	Описание
<code>name</code>	string	—	Уникальное имя правила. Используется для ссылок из профилей. Обязательно; пустое или пробельное имя отклоняется
<code>ciphers</code>	string	—	Строка шифров TLS 1.2 и ниже в формате OpenSSL
<code>ciphersuites</code>	string	—	Шифронаборы TLS 1.3 (см. раздел ниже)
<code>min-version</code>	string	<code>TLS1.2</code>	Минимальная допустимая версия TLS
<code>max-version</code>	string	<code>TLS1.3</code>	Максимальная допустимая версия TLS
<code>description</code>	string	—	Текстовое описание назначения правила

Примечание. При создании правила в веб-интерфейсе поле «Шифры» предзаполняется строкой `ECDHE+AESGCM:ECDHE+CHACHA20:DHE+AESGCM:!aNULL:!MD5:!DSS`. Умолчания приходят с устройства запросом `GET /api/v1/config/ssl/cipher-defaults`, поэтому набор по умолчанию совпадает с тем, что применяет система.

Строка шифров (cipher string)

Параметр `ciphers` использует формат строки шифров OpenSSL. Строка состоит из спецификаторов, разделённых двоеточиями (:), которые определяют, какие алгоритмы включены, а какие исключены.

Синтаксис

Префикс перед спецификатором определяет, как соответствующие шифры включаются в итоговый список или исключаются из него.

Префикс	Значение	Пример
(нет)	Включить шифры	<code>ECDHE+AESGCM</code>
<code>!</code>	Навсегда исключить	<code>!aNULL</code>
<code>-</code>	Временно убрать из списка	<code>-RC4</code>
<code>+</code>	Переместить в конец списка	<code>+SHA</code>

Основные ключевые слова

Ключевые слова OpenSSL обозначают алгоритм обмена ключами, шифрования или контроля целостности.

Ключевое слово	Описание
<code>ECDHE</code>	Обмен ключами на эллиптических кривых (forward secrecy)
<code>DHE</code>	Обмен ключами Diffie-Hellman (forward secrecy)
<code>RSA</code>	Обмен ключами RSA (без forward secrecy)
<code>AESGCM</code>	Шифрование AES в режиме GCM (AEAD)
<code>CHACHA20</code>	Шифрование ChaCha20-Poly1305 (AEAD)
<code>AES256</code>	AES с 256-битным ключом
<code>AES128</code>	AES с 128-битным ключом
<code>SHA384</code>	HMAC на SHA-384
<code>SHA256</code>	HMAC на SHA-256
<code>SHA</code>	HMAC на SHA-1 (устаревший)
<code>aNULL</code>	Шифры без аутентификации (небезопасно)

Ключевое слово	Описание
eNULL	Шифры без шифрования (небезопасно)
MD5	HMAC на MD5 (устаревший)
ALL	Все шифры кроме eNULL
HIGH	Шифры с высокой криптостойкостью

Комбинации ключевых слов

Символ `+` между ключевыми словами (без пробелов) означает логическое И:

- `ECDHE+AESGCM` — шифры с обменом ключами ECDHE **и** шифрованием AES-GCM
- `ECDHE+CHACHA20` — шифры с обменом ключами ECDHE **и** шифрованием ChaCha20

Версии TLS

Параметры `min-version` и `max-version` определяют диапазон допустимых версий протокола TLS.

Значение	Протокол	Статус
TLS1.0	TLS 1.0 (RFC 2246)	Устаревший. Отключён в большинстве современных браузеров
TLS1.1	TLS 1.1 (RFC 4346)	Устаревший. Отключён в большинстве современных браузеров
TLS1.2	TLS 1.2 (RFC 5246)	Рекомендуемый минимум для рабочей среды
TLS1.3	TLS 1.3 (RFC 8446)	Наиболее безопасный. Требуется лицензию Standard/Premium

Внимание. Поддержка TLS 1.3 доступна только при наличии лицензии уровня Standard или Premium. На лицензии Basic параметр `max-version: TLS1.3` будет отклонён при применении конфигурации.

Особенности TLS 1.3

TLS 1.3 использует собственный набор шифронаборов (cipher suites), не совместимый со строкой шифров TLS 1.2. Шифронаборы TLS 1.3 выбираются в правиле явно — параметром `ciphersuites`. В веб-интерфейсе это мультिवыбор из пяти значений:

- `TLS_AES_256_GCM_SHA384`
- `TLS_CHACHA20_POLY1305_SHA256`
- `TLS_AES_128_GCM_SHA256`
- `TLS_AES_128_CCM_SHA256`
- `TLS_AES_128_CCM_8_SHA256`

При создании правила предвыбраны первые три шифронабора. Параметр `ciphersuites` действует только тогда, когда `max-version` допускает TLS 1.3; правила, ограниченные TLS 1.2 (например, ГОСТ-профили), его игнорируют.

Строка `ciphers` применяется только для версий TLS 1.2 и ниже.

ГОСТ-шифры

Наборы шифров ГОСТ настраиваются отдельно от правил этого раздела — параметром `ssl.gost-default-ciphers`. По умолчанию используется семейство ГОСТ Р 34.12-2015 («Кузнечик» и «Магма»); устаревшее семейство ГОСТ 2001 отвергается, пока не включён параметр `gost-allow-legacy`. Подробнее: ГОСТ TLS.

Рекомендуемые наборы шифров

Готовые примеры предназначены для типовых требований к безопасности и совместимости.

Для стандартного использования

Набор подходит для современных клиентов и origin-серверов с поддержкой TLS 1.2 и TLS 1.3.

```
{
  "name": "default",
  "ciphers": "ECDHE+AESGCM:ECDHE+CHACHA20:DHE+AESGCM:!aNULL:!MD5:!DSS",
  "min-version": "TLS1.2",
  "max-version": "TLS1.3"
}
```

Обеспечивает forward secrecy, поддерживает все современные браузеры и серверы. Исключает небезопасные алгоритмы (NULL-аутентификация, MD5, DSS).

Для строгого соответствия требованиям

Набор исключает обмен ключами RSA и алгоритм SHA-1 для сред с повышенными требованиями к криптографической защите.

```
{
  "name": "strict",
  "ciphers": "ECDHE+AESGCM:!RSA:!SHA1",
  "min-version": "TLS1.2",
  "max-version": "TLS1.3"
}
```

Только ECDHE для обмена ключами, только AES-GCM для шифрования. Исключены RSA key exchange и SHA-1.

Для совместимости с устаревшими системами

Набор сохраняет устаревшие версии TLS и алгоритмы только для систем, которые нельзя перевести на современные параметры.

```
{
  "name": "legacy",
  "ciphers": "ECDHE+AESGCM:ECDHE+CHACHA20:DHE+AESGCM:ECDHE+AES:DHE+AES:RSA+AESGCM:RSA+AES:!aNULL:!MD5",
  "min-version": "TLS1.0",
  "max-version": "TLS1.3"
}
```

Внимание. Использование TLS 1.0/1.1 и RSA key exchange снижает уровень безопасности. Применяйте этот набор только при необходимости поддержки устаревших систем.

Проверка строки шифров

Для проверки, какие конкретные шифры соответствуют строке, используйте OpenSSL на устройстве:

```
### Список шифров для TLS 1.2
openssl ciphers -v 'ECDHE+AESGCM:ECDHE+CHACHA20:DHE+AESGCM:!aNULL:!MD5:!DSS'

### Список cipher suites для TLS 1.3
openssl ciphers -v -tls1_3
```

Использование в профилях

Правило шифрования указывается по имени в клиентском и серверном профилях:

```

{
  "ssl": {
    "client-profile": {
      "cipher-rule": "default"
    },
    "server-profile": {
      "cipher-rule": "default"
    }
  }
}

```

Можно использовать разные правила для клиентского и серверного профилей. Например, строгий набор для соединений с клиентами и более совместимый — для подключений к внешним серверам.

Связанные разделы

Применение правил шифрования в профилях и параметры ГОСТ описаны в связанных разделах руководства.

- Обзор SSL/TLS — принцип работы MITM-инспекции
- Профиль клиента — параметры TLS на стороне клиента
- Профиль сервера — параметры TLS на стороне сервера
- ГОСТ TLS — наборы шифров ГОСТ

Доверенные УЦ

Хранилище доверенных УЦ содержит сертификаты, по которым Амибиг проверяет цепочку доверия сервера назначения. Хранилище используется, если в серверном профиле задано `verify-peer: true`. Его состав настраивается на вкладке **Доверенные УЦ** экрана **SSL/TLS**.

При включённом переключателе «Собирать набор автоматически» Амибиг объединяет базовый набор из образа с загруженными сертификатами, удаляет исключения и формирует итоговый PEM-файл. Если автоматическая сборка выключена, состав PEM-файла определяет администратор.

SSL/TLS

Сертификаты Правила шифрования Профиль клиента Профиль сервера **Доверенные УЦ** ГОСТ Сертификат WebUI

Собирать набор автоматически: ☐

Набор собирается из базового набора прошивки и загруженных сертификатов типа `ca` и `bundle`: загрузили сертификат — он в доверии. Записи с назначенной ролью (подписывающий CA перехвата, ГОСТ-CA, сама цель сборки) в набор не входят. Итоговый файл записывается по пути записи из «Профиль сервера → Доверенный CA»; при выключенном режиме файл управляется оператором вручную, как раньше.

Состав итогового набора

Поиск по CN, субъекту или отпечатку ☐ Показывать снятые

Доверять	Субъект (CN)	Источник	Действует до
☒	DigiCert TLS ECC P384 Root G5	Базовый набор прошивки	14.01.2046
☒	D-TRUST BR Root CA 2 2023	Базовый набор прошивки	09.05.2038
☒	TrustAsia TLS RSA Root CA	Базовый набор прошивки	15.05.2044
☒	D-TRUST EV Root CA 1 2020	Базовый набор прошивки	11.02.2035
☒	emSign Root CA - C1	Базовый набор прошивки	18.02.2043
☒	COMODO ECC Certification Authority	Базовый набор прошивки	18.01.2038
☒	GlobalSign	Базовый набор прошивки	19.01.2038
☒	Amazon Root CA 3	Базовый набор прошивки	26.05.2040
☒	SwissSign RSA TLS Root CA 2022 - 1	Базовый набор прошивки	08.06.2047
☒	Amazon Root CA 2	Базовый набор прошивки	26.05.2040
☒	SSL.com EV Root Certification Authority ECC	Базовый набор прошивки	12.02.2041
☒	Telia Root CA v2	Базовый набор прошивки	29.11.2043
☒	Izenpe.com	Базовый набор прошивки	13.12.2037
☒	GlobalSign	Базовый набор прошивки	10.12.2034
☒	Starfield Root Certificate Authority - G2	Базовый набор прошивки	31.12.2037
☒	TunTrust Root CA	Базовый набор прошивки	26.04.2044
☒	SSL.com EV Root Certification Authority RSA R2	Базовый набор прошивки	30.05.2042

↑ Применить конфигурацию
📖 Документация
admin

Источники набора

Состав набора вычисляется автоматически, вручную источники не перечисляются. В набор входят:

- **базовый набор из образа** — набор Mozilla NSS, число сертификатов в котором зависит от версии образа;
- **все загруженные записи** типов `ca` и `bundle`, у которых нет назначенной роли.

Записи с назначенной ролью в набор **не входят**:

- запись, указанная в `server-profile.trusted-ca`, в которую записывается собранный набор;
- УЦ для SSL/TLS-инспекции из `client-profile.ca-certificate`;
- ГОСТ-УЦ — запись из `ssl.gost-ca`.

УЦ для SSL/TLS-инспекции исключается из хранилища. Иначе Амибиг мог бы считать доверенным сертификат сервера назначения, выпущенный этим УЦ.

Таблица состава

Под переключателем показывается состав итогового набора. Каждая строка — один сертификат:

- галочка **Доверять**;
- **Субъект (CN)** сертификата;
- **Источник** (базовый набор прошивки или имя загруженной записи);
- **Действует до**.

Если снять флажок «Доверять», SHA-256-отпечаток сертификата добавляется в список исключений. Такой сертификат не входит в итоговый набор. Поиск в таблице работает по CN, субъекту и отпечатку. Переключатель «Показывать снятые» добавляет в таблицу сертификаты, которые не вошли в набор:

- **включён** — сертификат входит в набор;
- **исключён** — отпечаток попал в список исключений;
- **заменён** — вместо сертификата из базового набора используется загруженный сертификат того же УЦ.

В итоговой строке указаны общее число сертификатов и число исключений. Кнопка «Скачать набор» сохраняет собранный набор в формате PEM.

Исключения

Снять доверие к конкретному публичному УЦ можно, не удаляя его сертификат из системы, — через исключение по отпечатку. Основной способ — снять галочку «Доверять» в таблице состава; ручной ввод отпечатка доступен в блоке «Дополнительно».

В конфигурации исключения хранятся в `ssl.trusted-store.exclusions.exclusion[]`; каждая запись содержит SHA-256-отпечаток (`fingerprint`) и необязательное описание (`description`):

```
{
  "ssl": {
    "trusted-store": {
      "exclusions": {
        "exclusion": [
          {
            "fingerprint": "9f2c...e14a",
            "description": "Отозванный публичный УЦ"
          }
        ]
      }
    }
  }
}
```

Правила объединения

При сборке набора действуют следующие правила:

- **Дедупликация по отпечатку.** Один и тот же сертификат из базового набора и из загруженной записи в итог попадает один раз.
- **Кросс-подписанные корни сохраняются оба.** Если у сертификатов совпадает субъект, но различаются ключи (кросс-подпись), в наборе остаются оба.
- **Приоритет пользовательского сертификата.** Если у базового и загруженного сертификатов совпадают и субъект, и ключ, в наборе остаётся пользовательский, а базовый помечается как вытесненный — так администратор видит, почему базового сертификата в наборе нет.
- **Исключения вычитаются последними** — уже после объединения источников.

Применение конфигурации

Набор формируется до перезапуска dataplane. Если сборка завершается ошибкой, применение конфигурации прекращается.

Если базовый набор из образа не найден, Амибиг фиксирует предупреждение и формирует набор только из загруженных сертификатов. В таком состоянии число доверенных УЦ может оказаться неполным. Восстановите пакет с базовым набором и повторно примените конфигурацию.

Итоговый набор записывается в файл, указанный в `server-profile.trusted-ca`. Если автоматическая сборка выключена, администратор самостоятельно управляет составом этого файла.

API

Работу с доверенным набором обслуживают два запроса:

Запрос	Назначение
<code>GET /api/v1/config/ssl/trusted-store</code>	Предпросмотр состава набора: перечень сертификатов с состояниями и счётчиками. Проблемы сборки возвращаются в поле <code>problems[]</code> , а не ошибкой запроса
<code>GET /api/v1/config/ssl/trusted-store/bundle</code>	Скачивание собранного набора PEM-файлом. Если набор собрать нельзя, возвращается код <code>409</code> с причиной

Предпросмотр считается по сохранённой конфигурации, поэтому после правок секцию нужно сохранить, чтобы состав пересчитался.

Связанные разделы

Настройка серверного профиля и управление сертификатами описаны в следующих разделах:

- Обзор SSL/TLS
- Управление сертификатами
- Профиль сервера — проверка сертификата origin-сервера по доверенному набору
- ГОСТ TLS

Клиентский SSL-профиль (client-profile)

Клиентский профиль управляет TLS-соединением между клиентом и Амибиг. При SSL/TLS-инспекции Амибиг предъявляет клиенту сертификат, сформированный для запрошенного имени и подписанный выбранным УЦ.

Формат конфигурации

Клиентский профиль задаётся одним объектом `ssl.client-profile` и применяется ко всем соединениям, выбранным для SSL/TLS-инспекции.

```
{
  "ssl": {
    "client-profile": {
      "ca-certificate": "mitm-ca",
      "cipher-rule": "default",
      "allow-non-ssl": true,
      "forward-proxy-bypass": true,
      "close-notify": "propagate",
      "handshake-timeout": 10,
      "alpn": {
        "mode": "negotiate"
      },
      "cert-lifespan": 30,
      "cert-extensions": [
        "basic-constraints",
        "subject-alternative-name",
        "extended-key-usage"
      ]
    }
  }
}
```

Параметры

Параметры профиля определяют УЦ для выпуска сертификатов, допустимые версии TLS и поведение при обработке клиентского соединения.

Основные параметры

Основные параметры связывают профиль с записью УЦ и правилом шифрования.

Параметр	Тип	По умолчанию	Описание
<code>ca-certificate</code>	string	—	Имя записи сертификата типа <code>ca</code> из секции <code>ssl.certificates</code> . Используется для подписи генерируемых MITM-сертификатов
<code>cipher-rule</code>	string	—	Имя правила шифрования из секции <code>ssl.cipher-rules</code> . Определяет допустимые шифры и версии TLS для клиентского соединения

Поведение при обработке трафика

Эти параметры задают действия для незашифрованного трафика, ошибок инспекции и завершения TLS-соединения.

Параметр	Тип	По умолчанию	Описание
<code>allow-non-ssl</code>	bool	<code>true</code>	Если <code>true</code> — не-TLS трафик (plain TCP) пропускается без разрыва соединения. Если <code>false</code> — не-TLS соединения отклоняются

Параметр	Тип	По умолчанию	Описание
<code>forward-proxy-bypass</code>	bool	<code>true</code>	Если <code>true</code> , соединение пропускается без расшифрования, когда SSL/TLS-инспекцию нельзя продолжить, например при запросе клиентского сертификата. Если <code>false</code> , соединение закрывается
<code>close-notify</code>	string	<code>propagate</code>	Отправлять ли клиенту сигнал <code>close_notify</code> при завершении перехваченной сессии TLS. В веб-интерфейсе — поле «Завершение сессии TLS»
<code>handshake-timeout</code>	uint16	<code>10</code>	Время на установление TLS с клиентом, в секундах. Диапазон 0–300, значение <code>0</code> отключает ограничение. В веб-интерфейсе — поле «Время на установление TLS»

Допустимые значения `close-notify`:

Значение	Описание
<code>propagate</code>	Амибиг отправляет <code>close_notify</code> клиенту только в том случае, если origin-сервер завершил свою TLS-сессию сам. Так клиент может отличить полный ответ от оборванного
<code>never</code>	<code>close_notify</code> не отправляется никогда. Поведение прежних версий Амибиг и умолчание F5

ALPN (Application-Layer Protocol Negotiation)

Настройки ALPN определяют согласование прикладного протокола с клиентом при TLS handshake.

Параметр	Тип	По умолчанию	Описание
<code>alpn.mode</code>	string	<code>negotiate</code>	Режим согласования протокола прикладного уровня

Допустимые значения `alpn.mode` :

Значение	Описание
<code>negotiate</code>	Согласование в пределах поддерживаемых протоколов. Если клиент предложил <code>http/1.1</code> , выбирается он. Если клиент предложил только <code>h2</code> , протокол не навязывается: ALPN-расширение в ответе не отправляется, и клиент сам решает, продолжать ли по HTTP/1.1
<code>downgrade-h1</code>	Принудительное понижение до HTTP/1.1. В ответе всегда указывается <code>http/1.1</code> , даже если клиент предлагал только HTTP/2
<code>disable</code>	ALPN-расширение не отправляется ни при каких условиях. Протокол определяется по содержимому

Примечание. Инспекция выполняется по HTTP/1.x, поэтому протокол HTTP/2 не выбирается ни в одном из режимов. Различие между `negotiate` и `downgrade-h1` проявляется на клиенте, который предложил только `h2`.

Внимание. В режиме `downgrade-h1` ответ `http/1.1` отправляется даже клиенту, который этот протокол не предлагал. Это осознанное отступление от RFC 7301 ради выполнения требования о понижении HTTP/2 до HTTP/1.1. Строгие клиенты (Go `crypto/tls`, BoringSSL, gRPC) могут разорвать TLS handshake, получив невыбранный ими протокол. Если такие клиенты должны работать, используйте режим `negotiate`.

Параметры генерации сертификатов

Параметры задают срок действия и набор расширений сертификатов, которые Амибиг формирует для клиентов.

Параметр	Тип	По умолчанию	Описание
<code>cert-lifespan</code>	uint16	30	Срок жизни генерируемых MITM-сертификатов в днях. Диапазон 1–825
<code>cert-extensions</code>	list	[]	Список расширений X.509, копируемых из оригинального сертификата сервера

Допустимые значения в списке `cert-extensions` :

Значение	Расширение X.509	Описание
<code>basic-constraints</code>	Basic Constraints	Флаг CA/non-CA и длина цепочки
<code>subject-alternative-name</code>	Subject Alternative Name (SAN)	Альтернативные имена хоста (DNS, IP)
<code>extended-key-usage</code>	Extended Key Usage (EKU)	Назначение ключа (serverAuth, clientAuth)

Рекомендация. Для корректной работы с современными браузерами настоятельно рекомендуется включить как минимум `subject-alternative-name`. Браузеры Chrome и Firefox игнорируют поле CN и полагаются исключительно на SAN для проверки имени хоста.

Производительность и кеши

Следующие параметры управляют стоимостью установления перехваченных соединений: выбором набора криптографических алгоритмов и кешами, которые избавляют от повторной работы при последующих соединениях к тем же серверам. В веб-интерфейсе они собраны на вкладке «Профиль клиента»: поле «Профиль криптографии перехвата» и группа «Кешы и возобновление сессий».

Параметр	Тип	По умолчанию	Описание
<code>mitm-crypto-profile</code>	string	<code>balanced</code>	Профиль криптографии перехвата: <code>compatibility</code> , <code>balanced</code> или <code>throughput</code>
<code>forge-cache-enabled</code>	bool	<code>true</code>	Кеш выпущенных сертификатов: не выпускать сертификат заново для уже виденного имени сервера
<code>forge-cache-max-entries</code>	uint32	<code>4096</code>	Максимальное число записей в кеше выпущенных сертификатов
<code>ctx-cache-enabled</code>	bool	<code>true</code>	Кеш контекстов TLS: переиспользовать готовый контекст для следующих соединений к тому же имени сервера
<code>ctx-cache-max-entries</code>	uint32	<code>4096</code>	Максимальное число записей в кеше контекстов
<code>session-cache-enabled</code>	bool	<code>true</code>	Возобновление сессий с клиентами: повторное подключение по сокращённой процедуре, без полного согласования ключей
<code>session-cache-timeout-seconds</code>	uint32	<code>300</code>	Время жизни сессии в кеше, в секундах
<code>session-ticket-mode</code>	string	<code>local</code>	Билеты сессий (session tickets): <code>local</code> , <code>ha-sync</code> или <code>disabled</code>

Допустимые значения `mitm-crypto-profile`:

Значение	Описание
<code>compatibility</code>	Самый широкий набор алгоритмов — подключаются в том числе устаревшие клиенты
<code>balanced</code>	Сбалансированный набор. Значение по умолчанию
<code>throughput</code>	Узкий набор быстрых алгоритмов: установление соединений обходится дешевле, но клиенты без их поддержки подключиться не смогут

Допустимые значения `session-ticket-mode` :

Значение	Описание
<code>local</code>	Билеты действительны только на этом узле
<code>ha-sync</code>	Ключ билетов общий для HA-пары: возобновление сессии переживает переключение на HA-партнёра
<code>disabled</code>	Билеты не выдаются

Детальное описание параметров

В подразделах ниже пояснено влияние основных параметров на обработку клиентского TLS-соединения.

ca-certificate

Ссылка на запись в `ssl.certificates` с типом `ca`. Этот сертификат и его ключ используются для подписи всех динамически генерируемых MITM-сертификатов.

При применении конфигурации Амибиг:

1. Ищет запись с указанным именем в `ssl.certificates`.
2. Извлекает пути `cert-path` и `key-path`.
3. Применяет указанные сертификат и ключ к движку инспекции.

Внимание. До смены УЦ для SSL/TLS-инспекции распространите его сертификат на клиентские устройства. Иначе после применения конфигурации клиенты не будут доверять сформированным сертификатам.

allow-non-ssl

Определяет поведение Амибиг при обнаружении не-TLS трафика на перехваченном порту.

Значение	Поведение
true	Не-TLS TCP-соединения (plain HTTP, другие протоколы) пропускаются напрямую без модификации
false	Не-TLS соединения разрываются (TCP RST)

Этот параметр полезен при работе в режиме `autossll`, когда на один порт может приходить как зашифрованный, так и незашифрованный трафик.

forward-proxy-bypass

Параметр задаёт действие, если SSL/TLS-инспекцию нельзя продолжить.

К таким случаям относятся запрос клиентского сертификата сервером (mutual TLS) и отсутствие УЦ для формирования сертификата.

Certificate pinning проверяется клиентским приложением после получения сертификата. Амибиг не может заранее определить такую проверку и автоматически перейти в режим без расшифрования.

Значение	Поведение
true	Соединение пропускается напрямую (passthrough) без расшифрования
false	Соединение разрывается

close-notify

Определяет, отправляет ли Амибиг клиенту сигнал `close_notify` — корректное завершение сессии TLS. В веб-интерфейсе параметру соответствует поле «Завершение сессии TLS».

В режиме `propagate` (по умолчанию) Амибиг отправляет `close_notify` клиенту только в том случае, если origin-сервер завершил свою TLS-сессию сам. Признак конца потока переносится с origin-соединения на клиентское: сервер завершил сессию корректно — клиент получает `close_notify`; сервер оборвал соединение — клиент видит обрыв. Только так клиент может отличить полный ответ от оборванного, когда длина ответа задана концом потока. По этой же причине варианта «отправлять всегда» не существует: собственный `close_notify` поверх оборванного ответа origin-сервера выдал бы усечённые данные за полные.

В режиме `never` сигнал `close_notify` не отправляется. Клиент не может отличить корректное завершение TLS-сессии от обрыва транспортного соединения. Некоторые TLS-клиенты считают такое завершение ошибкой.

handshake-timeout

Параметр ограничивает время ожидания ClientHello и TLS handshake с клиентом. По истечении времени соединение закрывается. Значение `0` отключает ограничение. В веб-интерфейсе параметру соответствует поле «Время на установление TLS».

Отсчёт начинается при переходе в фазу установления соединения и не сбрасывается при получении данных. Такое поведение ограничивает время, в течение которого медленный клиент занимает ресурсы TLS.

Срабатывания таймаута считаются в статистике отдельно по каждой стороне: по клиентским соединениям и по origin-соединениям (см. одноимённый параметр серверного профиля).

cert-lifespan

Параметр задаёт срок жизни сертификатов, сформированных для SSL/TLS-инспекции. Дата `Not Before` устанавливается на одни сутки раньше текущего времени, чтобы учесть возможное расхождение системных часов. Дата `Not After` рассчитывается от `Not Before` по заданному сроку.

Допустимый диапазон — от 1 до 825 дней, значение по умолчанию — 30 дней. В веб-интерфейсе параметру соответствует поле «Срок жизни сертификата (дн.)». Значения вне диапазона отклоняются при проверке конфигурации.

При горячем применении конфигурации кеш сформированных сертификатов создаётся заново. После применения новые сертификаты формируются с изменённым сроком.

cert-extensions

Параметр задаёт список расширений X.509, которые копируются из сертификата сервера назначения в сертификат для SSL/TLS-инспекции.

Пустой список означает, что расширения не копируются — генерируется минимальный сертификат с CN и базовыми атрибутами.

Рекомендуемый набор для рабочей среды:

```
{
  "cert-extensions": [
    "basic-constraints",
    "subject-alternative-name",
    "extended-key-usage"
  ]
}
```


mitm-crypto-profile

Выбор между совместимостью и производительностью для перехватываемых соединений. Профиль `compatibility` включает самый широкий набор алгоритмов — подключиться смогут в том числе устаревшие клиенты. Профиль `throughput` оставляет узкий набор быстрых алгоритмов: установление соединений обходится дешевле, но клиенты без поддержки этих алгоритмов подключиться не смогут. Умолчание — `balanced`.

Кеши и возобновление сессий

Установление перехваченного TLS-соединения — самая дорогая часть инспекции, и группа параметров «Кеши и возобновление сессий» позволяет не повторять уже выполненную работу:

- **Кеш выпущенных сертификатов** (`forge-cache-enabled`). На каждое имя сервера Амибиг выпускает собственный сертификат. Кеш избавляет от повторного выпуска для уже виденного имени — это заметная доля стоимости установления соединения.
- **Кеш контекстов TLS** (`ctx-cache-enabled`). Готовый контекст TLS переиспользуется для следующих соединений к тому же имени сервера вместо сборки заново.
- **Возобновление сессий с клиентами** (`session-cache-enabled`). Клиент, уже проходивший TLS handshake, подключается по сокращённой процедуре — без полного согласования ключей. Время жизни сессии в кеше задаёт `session-cache-timeout-seconds`.
- **Билеты сессий** (`session-ticket-mode`). Билет позволяет клиенту возобновить сессию без хранения состояния на устройстве. В режиме `ha-sync` ключ билетов общий для пары узлов, поэтому возобновление переживает переключение на HA-партнёра.

Пример полной конфигурации

Пример связывает клиентский профиль с УЦ и правилом шифрования и включает рекомендуемые параметры генерации сертификатов.

```
{
  "ssl": {
    "certificates": {
      "certificate": [
        {
          "name": "mitm-ca",
          "type": "ca",
          "cert-path": "mitm-ca.crt",
          "key-path": "mitm-ca.key",
          "description": "Corporate MITM CA"
        }
      ]
    },
    "cipher-rules": {
```

```

    "rule": [
      {
        "name": "modern",
        "ciphers": "ECDHE+AESGCM:ECDHE+CHACHA20:!aNULL:!MD5",
        "min-version": "TLS1.2",
        "max-version": "TLS1.3"
      }
    ],
    "client-profile": {
      "ca-certificate": "mitm-ca",
      "cipher-rule": "modern",
      "allow-non-ssl": true,
      "forward-proxy-bypass": true,
      "close-notify": "propagate",
      "handshake-timeout": 10,
      "alpn": {
        "mode": "negotiate"
      },
      "cert-lifespan": 30,
      "cert-extensions": [
        "basic-constraints",
        "subject-alternative-name",
        "extended-key-usage"
      ]
    }
  }
}

```

Типичные сценарии

Готовые варианты профиля показывают настройки для прозрачной обработки, строгого контроля и совместимости с устаревшими средствами защиты.

Максимальная прозрачность

Цель: минимальные отличия от оригинального TLS-соединения.

```

{
  "ca-certificate": "mitm-ca",
  "cipher-rule": "default",
  "allow-non-ssl": true,
  "forward-proxy-bypass": true,
  "alpn": { "mode": "negotiate" },
  "close-notify": "propagate",
  "cert-lifespan": 30,
  "cert-extensions": [
    "basic-constraints",
    "subject-alternative-name",
    "extended-key-usage"
  ]
}

```

Строгий режим

Цель: максимальный контроль, не-TLS трафик запрещён.

```

{
  "ca-certificate": "mitm-ca",

```

```

    "cipher-rule": "strict",
    "allow-non-ssl": false,
    "forward-proxy-bypass": false,
    "alpn": { "mode": "downgrade-h1" },
    "cert-lifespan": 7,
    "cert-extensions": [
      "subject-alternative-name"
    ]
  }

```

Совместимость с устаревшими инспекторами

Цель: сервисная цепочка содержит устройства, не поддерживающие HTTP/2.

```

{
  "ca-certificate": "mitm-ca",
  "cipher-rule": "default",
  "allow-non-ssl": true,
  "forward-proxy-bypass": true,
  "alpn": { "mode": "downgrade-h1" },
  "cert-lifespan": 30,
  "cert-extensions": [
    "basic-constraints",
    "subject-alternative-name",
    "extended-key-usage"
  ]
}

```

Связанные разделы

Дополнительные сведения о сертификатах, шифрах и origin-соединениях приведены в связанных разделах.

- Обзор SSL/TLS — принцип работы MITM-инспекции
- Управление сертификатами — загрузка CA-сертификатов
- Наборы шифров — настройка алгоритмов шифрования
- Профиль сервера — параметры TLS на стороне сервера

Серверный SSL-профиль (`server-profile`)

Серверный профиль управляет TLS-соединением между Амибиг и сервером назначения (origin-сервером). Профиль задаёт набор доверенных УЦ, набор шифров, версии TLS и действия при ошибках сертификата. Результат проверки сертификата определяет, продолжать ли SSL/TLS-инспекцию клиентского соединения.

Формат конфигурации

Серверный профиль задаётся объектом `ssl.server-profile` и применяется к origin-соединениям, выбранным для SSL/TLS-инспекции.

```

{
  "ssl": {
    "server-profile": {
      "trusted-ca": "trusted-ca-bundle",
      "cipher-rule": "default",
      "verify-peer": true,
      "on-expired-cert": "drop",
      "on-untrusted-cert": "drop",
      "on-revoked-cert": "ignore",
      "handshake-timeout": 10,
      "acceleration-mode": "cpu",
      "session-resumption-enabled": true,
      "session-cache-timeout-seconds": 300,
      "handshake-backpressure-limit": 0,
      "handshake-overflow-action": "defer"
    }
  }
}

```

Поведение параметров

Таблица описывает фактическое поведение каждого параметра в текущей версии.

Параметр	Поведение
<code>trusted-ca</code>	Ссылка на запись типа <code>bundle</code> в <code>ssl.certificates</code> . Указанный PEM-файл используется как набор доверенных УЦ для проверки сертификата сервера назначения
<code>cipher-rule</code>	Задаёт набор шифров и диапазон версий TLS для исходящего соединения
<code>verify-peer: true</code>	Включает проверку цепочки доверия и имени хоста для сертификата origin-сервера
<code>verify-peer: false</code>	Отключает проверку сертификата origin-сервера; TLS handshake завершается даже с недоверенным сертификатом
<code>on-expired-cert</code>	Действие для вердикта <code>expired</code> (просроченный сертификат)
<code>on-untrusted-cert</code>	Действие для вердикта <code>untrusted</code> (недоверенный сертификат)
<code>on-revoked-cert</code>	Поле политики; в текущей версии запросы OCSP/CRL не выполняются, поэтому вердикт <code>revoked</code> не формируется по данным внешней инфраструктуры отзыва
<code>handshake-timeout</code>	Максимальное время подключения к серверу назначения и TLS handshake с ним, в секундах (0–300, по умолчанию 10; 0 — без ограничения)

Параметр	Поведение
<code>acceleration-mode</code>	Использование аппаратного ускорителя Intel QAT для криптографических операций TLS: <code>cpu</code> , <code>auto</code> или <code>qat</code>
<code>session-resumption-enabled</code>	Возобновление ранее установленных сессий TLS с origin-серверами по сокращённой процедуре
<code>session-cache-timeout-seconds</code>	Время жизни сессии в кеше возобновления, в секундах (по умолчанию 300)
<code>handshake-backpressure-limit</code>	Порог одновременных TLS handshake с серверами назначения для одного рабочего процесса (0 — без ограничения)
<code>handshake-overflow-action</code>	Действие для нового соединения при достижении порога: <code>defer</code> , <code>drop</code> или <code>log-only</code>

Набор доверенных УЦ (`trusted-ca`)

Параметр ссылается на запись типа `bundle` в `ssl.certificates`. Её PEM-файл содержит набор УЦ, которым Амибиг доверяет при проверке сертификатов серверов назначения.

При автоматической сборке Амибиг объединяет базовый набор из образа с загруженными сертификатами и удаляет отмеченные исключения. Поле «Доверенный УЦ» в веб-интерфейсе в этом режиме недоступно для редактирования. Состав набора указан под полем. Подробнее см. в разделе «Доверенные УЦ».

Если автоматическая сборка выключена, администратор самостоятельно загружает и обновляет PEM-файл записи.

Проверка сертификата origin-сервера

При `verify-peer: true` Амибиг выполняет две проверки на исходящей стороне:

1. Строит цепочку доверия по набору `trusted-ca`.
2. Проверяет имя хоста по SNI, уже известному к моменту установления исходящего соединения.

Если SNI присутствует, оно используется как имя для проверки. Если SNI отсутствует, произвольное имя не подставляется: соединение помечается состоянием `hostname-unavailable`, и это состояние отдельно видно в журнале вердиктов.

Матрица действий

После TLS handshake с origin-сервером результат проверки классифицируется в один из вердиктов:

- `ok`
- `expired`
- `untrusted`
- `revoked`
- `unknown`

Действие применяется после получения вердикта и до передачи прикладных данных:

Действие	Поведение
<code>drop</code>	Соединение закрывается до передачи прикладных данных
<code>ignore</code>	Инспекция продолжается
<code>log-only</code>	Поведение как у <code>ignore</code> , но событие отдельно фиксируется в журнале

Для вердикта `unknown` в текущей версии используется консервативная ветка — действие для `untrusted`.

Таймаут установления (`handshake-timeout`)

Параметр ограничивает общее время установления origin-соединения: TCP handshake и TLS handshake с сервером назначения. Отсчёт начинается при переходе к установлению соединения и не сбрасывается при получении данных. По истечении времени соединение закрывается. Диапазон — 0–300 секунд, значение по умолчанию — 10. Значение 0 отключает ограничение. В веб-интерфейсе параметру соответствует поле «Время на установление TLS».

Для клиентского и origin-соединений задаются независимые значения. Это позволяет отдельно учитывать медленных клиентов и серверы назначения, которые приняли TCP-соединение, но не продолжили TLS handshake. Срабатывания учитываются отдельно для каждой стороны.

Аппаратное ускорение криптографии (`acceleration-mode`)

Управляет использованием аппаратного ускорителя Intel QAT для криптографических операций TLS. В веб-интерфейсе параметру соответствует поле «Аппаратное ускорение криптографии».

Значение	Поведение
<code>cpu</code>	Ускоритель не используется, вся криптография выполняется на процессоре. Значение по умолчанию
<code>auto</code>	Ускоритель используется, если он найден и исправен; при его недоступности работа продолжается на процессоре
<code>qat</code>	Запрашивать QAT. Если ускоритель не обнаружен или не исправен, рабочий процесс фиксирует предупреждение и продолжает выполнять криптографические операции на процессоре

Возобновление сессий с origin-серверами

При `session-resumption-enabled: true` Амибиг возобновляет ранее установленную сессию TLS с тем же origin-сервером по сокращённой процедуре вместо полного TLS handshake. Время жизни сессии в кеше задаёт `session-cache-timeout-seconds` (по умолчанию 300 секунд). В веб-интерфейсе параметры собраны в группе «Кеши и возобновление сессий» вкладки «Профиль сервера».

Ограничение одновременных установлений

Порог позволяет ограничить число TLS handshake, одновременно выполняемых одним рабочим процессом. Для ограничения нагрузки нужно выбрать действие `drop`.

Параметр `handshake-backpressure-limit` задаёт порог для одного рабочего процесса. Значение `0` (по умолчанию) отключает порог. Параметр `handshake-overflow-action` задаёт действие при его достижении:

Значение	Поведение
<code>defer</code>	Событие учитывается, а TLS handshake продолжается без постановки в очередь. Значение по умолчанию
<code>drop</code>	Отбросить соединение
<code>log-only</code>	Событие фиксируется в журнале, а TLS handshake продолжается

Действие при превышении порога имеет смысл только при включённом ограничении: конфигурация с `handshake-backpressure-limit: 0` и действием, отличным от `defer`, отклоняется при проверке.

В текущей реализации `defer` не откладывает обработку. Если нужно ограничить нагрузку, используйте `drop`; для наблюдения без влияния на трафик — `log-only`.

Ограничения текущей версии

Серверный профиль текущей версии не выполняет:

- запросы OCSP/CRL и принудительное применение статуса отзыва;
- поле F5 `unknown-cert-status-response-control` ;
- расширенные поля SNI-политики F5 (`server-name` , `sni-default` , `sni-require`);
- прочие поля политики SSL forward проху, не сводящиеся к набору доверенных CA, `verify-peer` и матрице действий.

При импорте UCS-конфигурации проверьте предупреждения о неподдерживаемых полях. Импорт такого поля не означает, что Амибиг применяет его к трафику.

Практические сценарии

Примеры показывают строгую проверку сертификатов, наблюдение без блокировки и тестовый режим.

Строгая проверка origin-сервера

Этот вариант разрывает соединение при просроченном или недоверенном сертификате origin-сервера.

```
{
  "trusted-ca": "trusted-ca-bundle",
  "cipher-rule": "default",
  "verify-peer": true,
  "on-expired-cert": "drop",
  "on-untrusted-cert": "drop",
  "on-revoked-cert": "ignore"
}
```

Рекомендуемая базовая конфигурация для эксплуатации: доверять только известному набору CA и разрывать соединение при просроченных или недоверенных сертификатах.

Мониторинг без разрыва соединений

Этот вариант фиксирует ошибки сертификата в журнале, не влияя на передачу трафика.

```
{
  "trusted-ca": "trusted-ca-bundle",
  "cipher-rule": "default",
  "verify-peer": true,
  "on-expired-cert": "log-only",
  "on-untrusted-cert": "log-only",
  "on-revoked-cert": "ignore"
}
```

Подходит для поэтапного внедрения: сначала оценить объём проблемных сертификатов origin-серверов, не влияя на трафик.

Тестовый режим без проверки

Этот вариант отключает проверку сертификата и предназначен только для изолированных испытательных сред.

```
{
  "trusted-ca": "trusted-ca-bundle",
  "cipher-rule": "default",
  "verify-peer": false,
  "on-expired-cert": "ignore",
  "on-untrusted-cert": "ignore",
  "on-revoked-cert": "ignore"
}
```

В этом режиме TLS handshake с origin-сервером завершается даже при самоподписанном сертификате. Использовать только в тестовых средах, где риск понятен и принят.

Связанные разделы

Дополнительные параметры SSL/TLS-инспекции описаны в следующих разделах:

- Обзор SSL/TLS
- Управление сертификатами
- Доверенные УЦ
- Наборы шифров
- Профиль клиента

ГОСТ TLS

Амибиг выполняет полный перехват ГОСТ-TLS — так же, как для обычного TLS. Устройство принимает соединение от ГОСТ-клиента, устанавливает собственное исходящее соединение к origin-серверу по ГОСТ-алгоритмам и формирует подменный сертификат, подписанный ГОСТ-ключом удостоверяющего центра инспекции. Расшифрованный трафик проходит политику и цепочку сервисов наравне с остальными соединениями. Это отличает Амибиг от решений, которые ГОСТ-трафик только пропускают без инспекции.

Криптографические операции выполняет открытый движок gost-engine версии 3.0.3, входящий в состав образа системы; дополнительная установка не требуется. Настройки размещаются в секции `ssl` конфигурации и применяются через REST API или файл конфигурации — отдельного экрана для ГОСТ-параметров в веб-интерфейсе нет.

Включение (gost-enabled)

Перехват ГОСТ-TLS управляется флагом `ssl.gost-enabled`. По умолчанию флаг выключен (`false`): соединение, в котором клиент предлагает ГОСТ-шифры,

пропускается без перехвата (passthrough) — инспекция для него не выполняется. После включения флага Амибиг перехватывает ГОСТ-соединения наравне с обычным TLS.

Включение `gost-enabled` требует указания удостоверяющего центра `gost-ca`: конфигурация с `gost-enabled: true` без `gost-ca` не пройдёт проверку при применении.

Удостоверяющий центр (gost-ca)

Параметр `ssl.gost-ca` — ссылка **по имени** на запись из списка сертификатов (`ssl.certificates`), а не путь к файлу. Указанный сертификат должен содержать ключ ГОСТ Р 34.10-2012 — система управления проверяет это при загрузке. Загрузка ГОСТ-УЦ выполняется тем же способом, что и для обычного CA-сертификата, — см. Управление сертификатами.

Выбор удостоверяющего центра выполняется для каждого соединения по криптографии клиента: соединение RSA/ECDSA-клиента подписывается обычным CA инспекции, соединение ГОСТ-клиента — ГОСТ-УЦ. Оба центра работают одновременно, поэтому смешанный трафик (обычный TLS и ГОСТ-TLS через одно устройство) не требует отдельных политик.

Наборы шифров

По умолчанию используется семейство ГОСТ Р 34.12-2015 — шифры «Кузнечик» и «Мagma» (RFC 9189). Политика шифров задаётся параметром `ssl.gost-default-ciphers`; значение по умолчанию:

```
GOST2012-KUZYNECHIK-KUZYNECHIKOMAC:GOST2012-MAGMA-MAGMAOMAC
```

Устаревшее семейство ГОСТ 2001 / 28147-89 по умолчанию отвергается: клиент, предлагающий только эти шифры, не пройдёт TLS handshake. Такое поведение защищает от понижения стойкости шифрования. Если в инфраструктуре остались устаревшие клиенты, семейство включается параметром `ssl.gost-allow-legacy: true`. Использовать его следует только как временную меру на период перевода клиентов на ГОСТ Р 34.12-2015 — постоянная работа на устаревших шифрах снижает защищённость трафика.

Параметр `ssl.gost-leaf-paramset` задаёт набор параметров (paramset) кривой ГОСТ Р 34.10-2012 для ключей подменных сертификатов; значение по умолчанию — `A` (набор ТК26, 256 бит).

Пример конфигурации

Минимальная конфигурация перехвата ГОСТ-TLS: запись ГОСТ-УЦ в списке сертификатов и включённый флаг с ссылкой на неё по имени.

```
{
  "ssl": {
```

```
"gost-enabled": true,  
"gost-ca": "gost-mitm-ca",  
"certificates": {  
  "certificate": [  
    {  
      "name": "gost-mitm-ca",  
      "type": "ca",  
      "cert-path": "gost-mitm-ca.crt",  
      "key-path": "gost-mitm-ca.key",  
      "description": "GOST MITM CA"  
    }  
  ]  
}
```

Параметры `gost-default-ciphers`, `gost-leaf-paramset` и `gost-allow-legacy` в примере не указаны — действуют значения по умолчанию, описанные выше.

Связанные разделы

Дополнительные сведения о сертификатах и правилах шифрования приведены в связанных разделах руководства.

- Обзор SSL/TLS — принцип работы MITM-инспекции
- Управление сертификатами — загрузка ГОСТ-УЦ через список сертификатов
- Наборы шифров — правила выбора шифров для обычного TLS

Сервисы

Раздел описывает внешние средства защиты, через которые Амибиг направляет расшифрованный трафик. В текущей версии поддерживаются inline-сервисы и ICAP-сервисы, а пулы и проверки доступности помогают управлять несколькими экземплярами одного сервиса.

Настройку сервисов выполняйте до создания сервисных цепочек: цепочка ссылается на уже описанные сервисы и их пулы.

Обзор сервисов

В разделе описаны типы внешних средств защиты, состав сервиса и связь сервисов с политиками и цепочками обработки трафика.

Назначение

Сервисы в Амибиг представляют собой средства информационной безопасности (СЗИ), через которые проходит расшифрованный трафик для инспекции. Каждый сервис соответствует конкретному устройству или программному решению — межсетевому экрану нового поколения (NGFW), DLP-системе, антивирусному шлюзу или иному средству анализа трафика.

Амибиг выступает в роли Service Chain Gateway: система расшифровывает SSL/TLS-трафик, направляет его через заданную последовательность сервисов, после чего повторно зашифровывает и отправляет получателю. Сервисы не видят зашифрованный трафик — они работают исключительно с открытым текстом, что позволяет им выполнять глубокий анализ содержимого.

Типы сервисов

Амибиг поддерживает два типа интеграции с внешними СЗИ:

Тип	Описание	Протокол	Типичные устройства
Inline	Прозрачный L2-мост — трафик проходит через устройство на канальном уровне	Ethernet (L2)	NGFW, IPS/IDS

Тип	Описание	Протокол	Типичные устройства
ICAP	Интеграция по протоколу ICAP — HTTP-запросы и ответы передаются на ICAP-сервер для модификации или проверки	ICAP (RFC 3507)	DLP, антивирус, контент-фильтры

Inline-сервисы

Inline-сервис работает в режиме прозрачного L2-моста. Амибиг направляет расшифрованный трафик через пару интерфейсов (to-service / from-service), между которыми размещается инспектирующее устройство. Устройство анализирует, модифицирует или блокирует трафик на канальном уровне.

Подробнее: Inline-сервисы (NGFW)

ICAP-сервисы

ICAP-сервис интегрируется с внешним ICAP-сервером по протоколу RFC 3507. Амибиг извлекает HTTP-запросы и ответы из расшифрованного потока и передает их на ICAP-сервер для проверки (request modification, response modification). ICAP-сервер возвращает вердикт — пропустить, модифицировать или заблокировать контент.

Подробнее: ICAP-сервисы (DLP)

Общая архитектура

Политика определяет, какой трафик попадает в цепочку сервисов; цепочка задаёт порядок прохождения через сервисы; каждый сервис описывает конкретное средство защиты и способ подключения к нему. Схема ниже показывает эти связи.

Схема взаимодействия

Источник	Связь	Назначение
Политика	→	Цепочка сервисов
Цепочка сервисов	→	Сервис 1 Inline NGFW
Цепочка сервисов	→	Сервис 2 ICAP DLP
Сервис 1 Inline NGFW	→	Возврат в Амибиг
Сервис 2 ICAP DLP	→	Возврат в Амибиг

Движок инспекции --> Сервисная цепочка --> Inline NGFW / Inline IPS / ICAP DLP

Компоненты сервиса

Каждый сервис в Амибиг включает следующие компоненты:

Компонент	Описание	Обязательный
Тип (<code>type</code>)	Режим работы: <code>inline</code> или <code>icap</code>	Да
Пул (<code>pool</code> или <code>pool-ref</code>)	Группа серверов/устройств с балансировкой нагрузки. Задаётся внутри сервиса или ссылкой на именованный пул с экрана Пулы	Да
Монитор (<code>monitor</code>)	Проверка доступности участников пула. Задаётся внутри пула или ссылкой на именованный монитор с экрана Мониторы	Да
Действие при недоступности (<code>service-down-action</code>)	Поведение при недоступности всех участников пула	Да
Inline-параметры (<code>inline</code>)	Настройки to-service/from-service (только для inline)	Для inline
ICAP-параметры (<code>icap</code>)	URI модификации, preview (только для ICAP)	Для ICAP

Общие параметры сервиса

Независимо от типа, каждый сервис имеет набор общих параметров:

Параметр	Тип	По умолчанию	Описание
<code>name</code>	string	—	Уникальное имя сервиса (например, <code>ngfw</code> , <code>dlp</code>)
<code>type</code>	enum	—	Тип сервиса: <code>inline</code> или <code>icap</code>
<code>description</code>	string	—	Описание назначения сервиса
<code>enabled</code>	boolean	<code>true</code>	Включен ли сервис

Параметр	Тип	По умолчанию	Описание
<code>service-down-action</code>	enum	<code>ignore</code>	Действие при недоступности: <code>ignore</code> , <code>drop</code> , <code>reset</code> (<code>bypass</code> — устаревший синоним <code>ignore</code>)
<code>weight</code>	uint16 (1-65535)	<code>100</code>	Вес сервиса в оценке работоспособности узла при аварийном переключении
<code>min-healthy-members</code>	uint8 (1-255)	<code>1</code>	Минимум доступных участников пула; ниже — сервис считается деградировавшим
<code>on-degrade</code>	enum	<code>drop</code>	Вклад деградировавшего сервиса в оценку здоровья узла: <code>drop</code> или <code>bypass</code>

Действие при недоступности сервиса (`service-down-action`)

Когда все участники пула сервиса становятся недоступными, Амибиг выполняет одно из следующих действий с соединениями клиентов:

Значение	Поведение
<code>ignore</code>	Трафик пропускается мимо сервиса без инспекции этим сервисом. Обеспечивает непрерывность связи
<code>bypass</code>	Устаревший синоним <code>ignore</code> . Принимается только для импорта старых конфигураций
<code>drop</code>	Трафик отбрасывается. Обеспечивает безопасность — без инспекции трафик не проходит
<code>reset</code>	Клиенту отправляется TCP RST. Соединение принудительно разрывается

Important

`ignore` (и устаревший `bypass`) — режим fail-open: при отказе средства защиты трафик продолжает проходить без проверки этим сервисом. Для политик, где отказ средства защиты не должен приводить к пропуску трафика без инспекции, используйте `drop` или `reset`.

Посервисное аварийное переключение

В отказоустойчивой паре состояние каждого сервиса влияет на право узла оставаться активным при аварийном переключении (failover). Для этого у сервиса задаются вес и порог деградации:

- **weight** — вес сервиса в оценке работоспособности узла. Оценка считается как сумма весов сервисов в состоянии UP, делённая на сумму весов участвующих сервисов. Чем выше вес, тем сильнее состояние сервиса влияет на оценку узла.
- **min-healthy-members** — минимальное число доступных участников пула. Если доступных участников меньше, сервис считается деградировавшим при расчёте оценки работоспособности узла.
- **on-degrade** — вклад деградировавшего сервиса в оценку: `drop` (по умолчанию) — сервис критичный, его вес обнуляется, но сервис остаётся в расчёте и тянет оценку узла вниз; `bypass` — сервис некритичный и исключается из расчёта полностью.

Параметр `on-degrade` определяет только вклад сервиса в оценку здоровья узла для аварийного переключения. За судьбу соединений клиентов при недоступности сервиса отвечает отдельный параметр `service-down-action`.

Порог оценки здоровья (`minimum-healthy-percent`), гистерезис переключения и режим немедленного переключения при падении критичного сервиса (`immediate-fail`) настраиваются на экране **Высокая доступность**. Подробнее: Active/Standby.

Связь с другими компонентами

Политики (правило --> `service-chain`) --> Цепочки сервисов (`chain` --> [`service1`, `service2`, ...]) --> Сервисы (`service` --> `pool` --> [`member1`, `member2`, ...]) --> Пул + Монитор (`health check`) --> Внешние устройства (NGFW, DLP, AV)

1. **Политики** определяют, какой трафик попадает в какую цепочку сервисов.
2. **Цепочки сервисов** задают порядок прохождения трафика через сервисы.
3. **Сервисы** описывают конкретные устройства и способ взаимодействия с ними.
4. **Пулы** обеспечивают балансировку нагрузки и отказоустойчивость.
5. **Мониторы** следят за доступностью каждого участника пула.

Порядок настройки

Рекомендуемый порядок создания конфигурации:

1. Настроить сетевые интерфейсы и VLAN (для inline-сервисов).
2. Создать сервисы, указав тип, параметры пула и монитора.
3. Создать цепочку сервисов, включив в нее сервисы в нужном порядке.
4. Создать политику, привязав цепочку сервисов к критериям отбора трафика.
5. Применить конфигурацию.

Ограничения

При планировании конфигурации учитывайте лицензионные и эксплуатационные ограничения сервисов.

Ограничение	Значение
Максимальное количество сервисов	Определяется лицензией
Максимальное количество участников в пуле	Без ограничений
Максимальная глубина цепочки	Определяется производительностью
Протоколы для inline	TCP для intercept/decrypt path; UDP/ICMP для bypass-style packet steering
Протоколы для ICAP	HTTP/HTTPS (внутри расшифрованного потока)

Inline-сервисы (NGFW)

В разделе описаны подключение inline-устройств, параметры сервисных плеч и контроль прохождения трафика через внешнее средство защиты.

Назначение

Inline-сервис обеспечивает прозрачную передачу расшифрованного трафика через внешнее устройство безопасности на канальном уровне (L2). Типичное применение — интеграция с межсетевыми экранами нового поколения (NGFW), системами предотвращения вторжений (IPS/IDS) и другими устройствами, работающими в режиме моста.

Амибиг размещает inline-устройство «в разрыв» потока расшифрованного трафика. Устройство получает открытый текст, анализирует его, может блокировать или модифицировать пакеты, после чего возвращает трафик обратно в Амибиг для повторного зашифрования.

Принцип работы

Inline-сервис передаёт расшифрованный трафик на внешнее устройство и принимает результат обработки через отдельное возвратное плечо.

Схема прохождения трафика

Расшифрованный трафик выходит из Амибиг через плечо to-service, проходит через инспектирующее устройство и возвращается через плечо from-service:

Схема взаимодействия

Источник	Связь	Назначение
Расшифрование	→ to-service	NGFW
NGFW	→ from-service	Зашифрование

Клиент --TLS--> Амибиг (расшифрование) --> to-service (VLAN) --> NGFW (инспекция) --> from-service (VLAN) --> обратно в цепочку --> Амибиг (зашифрование) --TLS--> Сервер

Этапы обработки

Обработка включает пять последовательных этапов от завершения клиентского TLS-соединения до возврата трафика в цепочку.

1. **Расшифрование** — Амибиг завершает TLS-сессию клиента, получая открытый текст.
2. **Передача to-service** — система направляет расшифрованный трафик на интерфейс to-service (VLAN) в сторону inline-устройства.
3. **Инспекция** — NGFW/IPS получает трафик в открытом виде, выполняет анализ, может заблокировать или модифицировать.
4. **Возврат from-service** — после инспекции трафик возвращается через интерфейс from-service (VLAN) обратно в Амибиг.
5. **Продолжение цепочки** — трафик передается в следующий сервис цепочки или на повторное зашифрование.

Плечи to-service и from-service

Каждый inline-сервис использует пару плеч, образующих логический мост:

Плечо	Направление	Описание
to-service	Амибиг -> устройство	Передача трафика на инспекцию. Аналог dirty VLAN

Плечо	Направление	Описание
from-service	Устройство -> Амибиг	Возврат проинспектированного трафика. Аналог clean VLAN

Каждое плечо привязывается либо к tagged VLAN, либо к untagged DPDK-порту или транку — необходимо указать ровно один из двух вариантов:

Параметр	Тип	Обязательный	Описание
vlan	vlan-ref	Одно из двух	Ссылка на VLAN (tagged), созданный в разделе сетевой конфигурации
interface	interface-ref	Одно из двух	Untagged DPDK-порт или транк на стороне сервиса
gateway	IPv4-адрес	Нет	Шлюз на стороне сервиса (IPv4)
ipv6-gateway	IPv6-адрес	Нет	Шлюз на стороне сервиса (IPv6)
self-ip	IP-адрес	Нет	Собственный адрес аппарата на этом плече (см. ниже)

Если плечо привязано к VLAN, применённая конфигурация может дополнительно содержать родительский интерфейс этого VLAN — система заполняет его автоматически при применении. Такая обогащённая форма принимается при импорте и повторном применении конфигурации и не считается конфликтом привязки.

Требования к привязке плеч

До создания сервиса подготовьте отдельные привязки для прямого и возвратного направлений трафика.

- Для inline-сервиса с tagged-привязкой необходимо создать два VLAN: один для to-service, второй для from-service.
- VLAN-интерфейсы создаются в разделе **Сеть > VLAN** до создания сервиса; каждый VLAN назначается на физический интерфейс или транк.
- Привязки to-service и from-service должны быть разными.

Проверка next-hop при применении

У каждого плеча inline-сервиса обязан быть шлюз (next-hop) — без него соединения к сервису будут падать, а цепочка не будет работать. Шлюз может быть задан в одном из трёх мест: в самом плече (gateway), в self-IP сегмента или в порту-родителе. Конфигурация, в которой шлюз не задан ни в одном из них, отклоняется при проверке применения — сообщение об ошибке указывает имя сервиса, VLAN или интерфейс плеча, self-IP сегмента и цепочки, которые перестанут работать.

Собственный адрес на плече (self-ip)

Параметр `self-ip` («Собственный адрес на плече» в форме сервиса на экране Сервисы) задаёт адрес аппарата на каждом плече. Если поле пусто, берётся первый self-IP, объявленный на VLAN или интерфейсе этого плеча.

Значение задаётся отдельно для каждого узла: при HA-синхронизации на партнёра оно не переносится.

Адреса плеч использует сквозная проба (см. ниже): с адреса `to-service.self-ip` уходит запрос пробы, а на адресе `from-service.self-ip` отвечает встроенный ответчик — поэтому в режиме петли адрес на плече from-service обязателен (заданный явно или взятый из self-IP сегмента).

Адресация сервисного сегмента в отказоустойчивой паре

В HA-кластере адресация сервисного сегмента (общая для пары или отдельная на каждом узле) задаётся глобальным режимом на экране Высокая доступность ; для отдельного сервиса режим можно переопределить параметром `service-addressing-mode` . Подробнее: Active/Standby.

Параметры inline-сервиса

Конфигурация inline-сервиса состоит из общих свойств, параметров сервисных плеч и настроек пула.

Основные параметры

Основные параметры задают имя, административное состояние и действие при недоступности сервиса.

Параметр	Тип	По умолчанию	Описание
<code>name</code>	string	—	Уникальное имя сервиса
<code>type</code>	enum	—	Всегда <code>inline</code>
<code>enabled</code>	boolean	<code>true</code>	Включен ли сервис

Параметр	Тип	По умолчанию	Описание
<code>service-down-action</code>	enum	<code>ignore</code>	Действие при отказе: <code>ignore</code> , <code>drop</code> , <code>reset</code> . Значение <code>bypass</code> принимается как устаревший синоним <code>ignore</code>

Inline-параметры

Inline-параметры связывают прямое и возвратное плечи с VLAN или физическими интерфейсами.

Параметр	Тип	По умолчанию	Описание
<code>to-service.vlan / to-service.interface</code>	vlan-ref / interface-ref	—	Привязка плеча передачи трафика на устройство (ровно одно из двух)
<code>to-service.gateway</code>	IPv4-адрес	—	IPv4-шлюз на стороне устройства (to)
<code>to-service.ipv6-gateway</code>	IPv6-адрес	—	IPv6-шлюз на стороне устройства (to)
<code>from-service.vlan / from-service.interface</code>	vlan-ref / interface-ref	—	Привязка плеча возврата трафика от устройства (ровно одно из двух)
<code>from-service.gateway</code>	IPv4-адрес	—	IPv4-шлюз на стороне устройства (from)
<code>from-service.ipv6-gateway</code>	IPv6-адрес	—	IPv6-шлюз на стороне устройства (from)
<code>to-service.self-ip</code>	IP-адрес	первый self-IP сегмента	Собственный адрес на плече to-service; с него уходит сквозная проба. Пер-узловое поле

Параметр	Тип	По умолчанию	Описание
<code>from-service.self-ip</code>	IP-адрес	первый self-IP сегмента	Собственный адрес на плече from-service; на нём отвечает встроенный ответчик сквозной пробы. Пер-узловое поле
<code>protocols</code>	list	<code>[tcp]</code>	Протоколы, которые прозрачный inline-сервис обрабатывает в dataplane. <code>tcp</code> — путь расшифрования (intercept), <code>udp</code> и <code>icmp</code> — прозрачное перенаправление пакетов без расшифрования

Параметры пула

Параметры пула определяют состав внешних устройств, метод балансировки и проверку доступности.

Параметр	Тип	По умолчанию	Описание
<code>pool.member[].address</code>	IP-адрес	—	Адрес участника пула
<code>pool.member[].port</code>	0-65535	—	Порт участника пула
<code>pool.member[].enabled</code>	boolean	<code>true</code>	Включен ли участник
<code>pool.load-balancing</code>	enum	<code>session-hash</code>	Метод балансировки
<code>pool.monitor.type</code>	enum	<code>tcp</code>	Тип проверки: <code>tcp</code> , <code>icmp</code> , <code>http-get</code>
<code>pool.monitor.interval</code>	секунды	<code>5</code>	Интервал проверок
<code>pool.monitor.timeout</code>	секунды	<code>16</code>	Таймаут ответа

Вместо встроенного пула сервис может ссылаться на именованный пул (`pool-ref`) с экрана Пулы — тогда встроенный блок `pool` игнорируется.

Important

Для inline-сервисов активную проверку доступности выполняет монитор типа `icmp`. Мониторы `tcp` и `http-get` для inline-сервисов активную пробу не выполняют: адрес участника указывает на само устройство, и TCP-проверка к нему не подтверждает прохождение трафика через мост. Подробнее: Мониторинг доступности.

Действие при отказе (service-down-action)

Когда все участники пула inline-сервиса становятся недоступными, Амибиг применяет настроенное действие. Этот параметр определяет, будет ли трафик остановлен или продолжит проходить без проверки этим сервисом.

Important

`ignore` и устаревшее значение `bypass` являются fail-open режимом. При таком режиме трафик продолжает проходить через цепочку без проверки отказавшим сервисом. Для политик, где отказ средства защиты не должен приводить к пропуску трафика без инспекции, используйте `drop` или `reset`.

Значение	Поведение	Когда использовать
<code>ignore</code>	Трафик пропускается без проверки этим сервисом	Приоритет — доступность сети
<code>bypass</code>	Устаревший синоним <code>ignore</code>	Только для импорта старых конфигураций
<code>drop</code>	Трафик отбрасывается без уведомления	Приоритет — безопасность, fail-closed
<code>reset</code>	Клиенту отправляется TCP RST	Требуется явное уведомление клиента

Типовые схемы развертывания

Схему подключения выбирают по числу устройств и требуемой последовательности инспекции.

Один NGFW

Базовая конфигурация с одним межсетевым экраном:

```
Амибиг (to-service, VLAN 100) --> NGFW eth0 ... NGFW eth1 --> (from-service, VLAN 200)
```

Амибиг

- to-service: VLAN 100, gateway 198.19.64.1
- from-service: VLAN 200, gateway 198.19.64.129

Два NGFW с балансировкой

Отказоустойчивая конфигурация с двумя устройствами:

```
Амибиг Pool (session-hash) --> NGFW-1 (VLAN 100/200) или NGFW-2 (VLAN 100/200) -->
```

Амибиг

- Балансировка: session-hash (привязка сессий)
- Монитор: icmp, interval=5, timeout=16
- service-down-action: ignore

NGFW + IPS в цепочке

Последовательная инспекция через два inline-устройства:

```
Трафик --> Амибиг --> NGFW (VLAN 100/200) --> IPS (VLAN 300/400) --> Амибиг --> Сервер
```

Пример конфигурации (YANG)

Пример создаёт inline-сервис с двумя VLAN, одним участником пула и ICMP-монитором.

```
services:
  service:
    - name: ngfw
      type: inline
      enabled: true
      service-down-action: ignore
      inline:
        to-service:
          vlan: vlan100
          gateway: "198.19.64.1"
        from-service:
          vlan: vlan200
          gateway: "198.19.64.129"
      protocols:
        - tcp
      pool:
        member:
          - address: "198.19.64.7"
            port: 0
        load-balancing: session-hash
      monitor:
        type: icmp
        interval: 5
        timeout: 16
```



```
fail-threshold: 3
recover-threshold: 2
```

Поведение при отказе устройства в цепочке

Если inline-устройство становится недоступным, судьба соединений клиентов определяется параметром `service-down-action` этого сервиса. При `drop` соединение завершается, и в журнале потоков фиксируется вердикт `DROP` с причиной `CHAIN_FAIL` — по этой причине удобно отбирать соединения, потерянные из-за отказа звена цепочки. Подробнее о порядке прохождения цепочки: Цепочки сервисов.

Постоянную активную проверку прохождения трафика через сервис система выполняет при включённой сквозной пробе (см. ниже). Без неё непрерывный мониторинг проверяет только доступность участников пула, а отказ пересылки внутри цепочки фиксируется по фактическому трафику. Разовую проверку прохождения можно выполнить вручную (см. раздел «Диагностика»).

Сквозная проба (traverse-probe)

Обычная проверка доступности обращается к самому сервисному устройству, а оно стоит в одной подсети с плечом — пакет уходит напрямую, и маршрут для этого не нужен. Рабочий трафик идёт к внешним адресам, и маршрут ему нужен. Отсюда класс отказов, при котором проверка показывает «1 из 1 доступен», а через цепочку не проходит ни одно соединение — например, из-за неверного шлюза.

Сквозная проба закрывает этот пробел: она открывает настоящее TCP-соединение к адресу **за** сервисом тем же путём, которым идёт рабочий трафик, и проверяет ровно то, что делает трафик — запрос дошёл до сервиса, сервис его пропустил, ответ вернулся.

По умолчанию цель не задаётся и работает **режим петли**: запрос уходит с плеча `to-service`, пересекает сервис, а целью служит собственный адрес плеча `from-service`, где отвечает встроенный ответчик. Внешних зависимостей у такой пробы нет — она работает и на изолированном стенде.

Параметр	Тип	По умолчанию	Описание
<code>traverse-probe.enabled</code>	boolean	<code>false</code>	Включить сквозную пробу

Параметр	Тип	По умолчанию	Описание
<code>traverse-probe.target</code>	IP-адрес	пусто	Адрес за сервисом, куда проба открывает соединение. Пусто — режим петли через встроенный ответчик. Явный адрес обязан лежать вне подсети плеча to-service: адрес из той же подсети достижим напрямую, проба уйдёт мимо сервиса и будет проходить даже при полностью неработающем сервисе (такая конфигурация отклоняется при проверке применения)
<code>traverse-probe.port</code>	1-65535	7778	Порт, к которому подключается проба; в режиме петли — порт встроенного ответчика на плече from-service
<code>traverse-probe.path</code>	string	/	Путь HTTP-запроса пробы
<code>traverse-probe.host</code>	string	пусто	Заголовок Host. Пусто — подставляется адрес цели. Нужен, когда за сервисом отвечает виртуальный хост
<code>traverse-probe.receive</code>	string	пусто	Ожидаемая подстрока в ответе. Пусто — в режиме петли ожидается метка встроенного ответчика, а с явной целью достаточно успешного кода ответа (2xx)

В веб-интерфейсе проба настраивается в форме inline-сервиса на экране **Сервисы**, в блоке «Сквозная проба».

```
inline:
  traverse-probe:
    enabled: true
    # target не задан — режим петли через встроенный ответчик
    port: 7778
    path: /
```

Результаты сквозной пробы учитываются при определении доступности участников пула наравне с обычными проверками, а переходы участника между состояниями записываются в журнал с указанием вида пробы (сквозная или обычная).

Диагностика

Диагностика позволяет проверить оба сервисных плеча, доступность участников и прохождение пробного трафика.

Проверка inline-сервиса

На экране **Сервисы** для inline-сервиса доступна кнопка **Проверить Inline**. Проверка выполняет последовательность шагов и показывает результат каждого:

1. **Интерфейсы** — плечи to-service и from-service подняты.
2. **ICMP Ping** — доступность адреса каждого участника пула.
3. **Трафик (UDP)** — прохождение пробного UDP-пакета через устройство (to-service -> from-service).
4. **Трафик (TCP)** — прохождение пробного TCP-сегмента через устройство.
5. **Монитор доступности** — текущее состояние участников по данным непрерывного мониторинга.

Карантин участника пула

Помимо активных проверок, на состояние участника влияет проверка пути данных: если реальное соединение к участнику пула не удалось установить, участник переходит в карантин и отображается как недоступный — даже если активная проба к нему проходит. Так исключается ситуация, когда экран показывает «1 из 1 доступен», а ни одно соединение через сервис не проходит.

Переходы участника между состояниями «работает» и «не работает» записываются в журнал с указанием вида пробы (сквозная или обычная) и счётчиков отказов и восстановлений.

Типичные проблемы

В таблице приведены основные признаки ошибок подключения и настройки действия при отказе.

Симптом	Возможная причина	Решение
Участник пула в статусе DOWN	VLAN не настроен или устройство выключено	Проверьте VLAN и физическое подключение
Трафик не проходит через устройство	Некорректные шлюзы to-service/from-service	Проверьте IP-адреса шлюзов
Все участники DOWN, трафик идёт без проверки этим сервисом	<code>service-down-action: ignore</code>	Это штатное поведение fail-open; проверьте устройства
Соединения сбрасываются при отказе сервиса	<code>service-down-action: reset</code>	Измените значение на <code>ignore</code> , если приоритет — доступность
В журнале потоков вердикт DROP с причиной CHAIN_FAIL	Отказ звена цепочки при <code>service-down-action: drop</code>	Проверьте устройство и состояние участников пула

ICAP-сервисы (DLP)

В разделе описаны режимы ICAP-инспекции, параметры подключения серверов анализа контента и способы диагностики интеграции.

Назначение

ICAP-сервис обеспечивает интеграцию Амибиг с внешними системами контент-анализа по протоколу ICAP (Internet Content Adaptation Protocol, RFC 3507). Типичное применение — подключение DLP-систем (предотвращение утечек данных), антивирусных шлюзов и контент-фильтров.

В отличие от inline-сервисов, ICAP-сервис не требует выделения VLAN и физического размещения устройства в разрыв канала. Амибиг извлекает HTTP-запросы и ответы из расшифрованного потока и передает их на ICAP-сервер по сети (IP-соединение). ICAP-сервер возвращает вердикт: пропустить, модифицировать или заблокировать контент.

Протокол ICAP

ICAP позволяет передавать HTTP-запросы и ответы на внешнюю проверку без размещения анализатора непосредственно в разрыве канала.

Основные концепции

ICAP (RFC 3507) — протокол адаптации контента, работающий поверх HTTP. Он позволяет Амибиг (ICAP-клиент) передавать HTTP-сообщения на внешний ICAP-сервер для проверки и модификации.

- **REQMOD:** Амибиг (расшифрованный HTTP-запрос) --REQMOD--> ICAP-сервер (проверка URI, заголовков, тела) --вердикт--> Амибиг
- **RESPMOD:** Амибиг (расшифрованный HTTP-ответ) --RESPMOD--> ICAP-сервер (проверка контента, MIME, вложений) --вердикт--> Амибиг

Режимы работы

Режим определяет, передаёт ли Амибиг на проверку запрос клиента или ответ origin-сервера.

Режим	Метод ICAP	Описание
Request Modification	REQMOD	HTTP-запрос клиента передается на ICAP-сервер до отправки на целевой сервер. Позволяет блокировать, перенаправить или модифицировать запрос
Response Modification	RESPMOD	HTTP-ответ сервера передается на ICAP-сервер до отправки клиенту. Позволяет проверить и модифицировать содержимое ответа

Каждый режим настраивается отдельно и может быть включен или отключен независимо.

Preview

Механизм Preview (предварительный просмотр) позволяет отправить на ICAP-сервер только начальную часть тела сообщения. ICAP-сервер на основании preview-данных принимает решение:

- **100 Continue** — требуется передать полное тело для анализа.
- **204 No Content** — проверка завершена, модификация не требуется (быстрый путь).
- **200 OK** — возвращается модифицированное сообщение.

Preview существенно снижает нагрузку на ICAP-сервер, так как большая часть трафика проходит проверку без передачи полного тела.

Параметр	Тип	По умолчанию	Описание
<code>preview-length</code>	uint32 (байты)	1024	Количество байт тела, отправляемых в preview. Значение 0 отключает preview

Параметры ICAP-сервиса

Конфигурация ICAP-сервиса включает общие свойства, отдельные параметры REQMOD и RESPMOD, а также пул серверов.

Основные параметры

Основные параметры задают имя, административное состояние и действие при недоступности сервиса.

Параметр	Тип	По умолчанию	Описание
<code>name</code>	string	—	Уникальное имя сервиса
<code>type</code>	enum	—	Всегда <code>icap</code>
<code>enabled</code>	boolean	<code>true</code>	Включен ли сервис
<code>service-down-action</code>	enum	<code>ignore</code>	Действие при недоступности: <code>ignore</code> , <code>drop</code> , <code>reset</code> (<code>bypass</code> — устаревший синоним <code>ignore</code>)

Значение `ignore` — режим fail-open: при недоступности всех ICAP-серверов трафик продолжает проходить без проверки этим сервисом. Для политик, где отказ средства защиты не должен приводить к пропуску трафика без инспекции, используйте `drop` или `reset`.

Request Modification (REQMOD)

Параметры REQMOD управляют передачей HTTP-запросов клиента на ICAP-сервер.

Параметр	Тип	По умолчанию	Описание
<code>icap.request-mod.enabled</code>	boolean	<code>true</code>	Включить модификацию запросов

Параметр	Тип	По умолчанию	Описание
<code>icap.request-mod.uri</code>	string	—	URI ICAP-сервиса для REQMOD (например, <code>icap://dlp-server/reqmod</code>)
<code>icap.request-mod.preview-length</code>	uint32	1024	Размер preview для запросов (байты)

Response Modification (RESPMOD)

Параметры RESPMOD управляют передачей ответов origin-сервера на ICAP-сервер.

Параметр	Тип	По умолчанию	Описание
<code>icap.response-mod.enabled</code>	boolean	true	Включить модификацию ответов
<code>icap.response-mod.uri</code>	string	—	URI ICAP-сервиса для RESPMOD (например, <code>icap://dlp-server/respmod</code>)
<code>icap.response-mod.preview-length</code>	uint32	1024	Размер preview для ответов (байты)

HTTP-параметры

Настройки обработки HTTP-заголовков и chunked-кодирования при передаче на ICAP-сервер:

Параметр	Тип	По умолчанию	Описание
<code>icap.http.max-header-count</code>	uint32	128	Максимальное количество HTTP-заголовков
<code>icap.http.max-header-size</code>	uint32	65536	Максимальный суммарный размер заголовков (байты)
<code>icap.http.request-chunking</code>	enum	sustain	Обработка chunked-кодирования запросов

Параметр	Тип	По умолчанию	Описание
<code>icap.http.response-chunking</code>	enum	<code>unchunk</code>	Обработка chunked-кодирования ответов

Режимы chunked-кодирования:

Значение	Описание
<code>sustain</code>	Сохранить исходное кодирование (chunked или обычное) без изменений
<code>rechunk</code>	Перекодировать в chunked-формат
<code>unchunk</code>	Раскодировать из chunked-формата в обычный (content-length)

Параметры пула

Пул задаёт адреса ICAP-серверов, способ распределения соединений и параметры проверки доступности.

Параметр	Тип	По умолчанию	Описание
<code>pool.member[].address</code>	IP-адрес	—	Адрес ICAP-сервера
<code>pool.member[].port</code>	0-65535	—	Порт ICAP-сервера (обычно 1344)
<code>pool.member[].enabled</code>	boolean	<code>true</code>	Включен ли участник
<code>pool.load-balancing</code>	enum	<code>session-hash</code>	Метод балансировки нагрузки
<code>pool.monitor.type</code>	enum	<code>tcp</code>	Тип проверки: <code>tcp</code> , <code>icmp</code> , <code>http-get</code>
<code>pool.monitor.interval</code>	секунды	<code>5</code>	Интервал проверок
<code>pool.monitor.timeout</code>	секунды	<code>16</code>	Таймаут ответа

Вместо встроенного пула сервис может ссылаться на именованный пул (`pool-ref`) с экрана Пулы — тогда встроенный блок `pool` игнорируется. Именованные пулы поддерживают также ссылку на именованный монитор с экрана Мониторы (включая тип `http-post`) и шлюз участника `to-gateway` для ICAP-серверов вне датаплейн-подсети (см. «Маршрутизируемый адрес ICAP-сервера»).

Схема взаимодействия

Схема показывает последовательность проверки запроса и ответа в рамках одного клиентского соединения.

Полный цикл ICAP-инспекции

Полный цикл включает расшифрование, запросы REQMOD и RESPMOD, обращение к origin-серверу и повторное зашифрование.

Последовательность взаимодействия

Источник	Связь	Назначение
Клиент	HTTP-запрос (TLS)	Амибиг
Амибиг	Этап	Расшифрование
Амибиг, ICAP-сервер	Этап	REQMOD — проверка запроса
Амибиг	REQMOD (HTTP-запрос)	ICAP-сервер
ICAP-сервер	200 OK / 204 No Content	Амибиг
Амибиг	HTTP-запрос	Сервер
Сервер	HTTP-ответ	Амибиг
Амибиг, ICAP-сервер	Этап	RESPMOD — проверка ответа
Амибиг	RESPMOD (HTTP-ответ)	ICAP-сервер
ICAP-сервер	200 OK / 204 No Content	Амибиг
Амибиг	Этап	Зашифрование
Амибиг	HTTP-ответ (TLS)	Клиент

1. Клиент --TLS--> Амибиг (расшифровывает)
2. Амибиг извлекает HTTP-запрос
3. Амибиг --ICAP REQMOD--> DLP-сервер
4. DLP-сервер проверяет запрос и возвращает вердикт: 204 (чист, передать серверу), 200 (модифицирован), 403 (заблокировать)
5. Амибиг --HTTP--> целевой сервер
6. Сервер --HTTP--> Амибиг получает ответ
7. Амибиг --ICAP RESPMOD--> DLP-сервер
8. DLP-сервер проверяет ответ и возвращает вердикт
9. Амибиг --TLS--> Клиент (зашифровывает)

Топология подключения

Амибиг (ICAP-клиент) --IP-сеть (L3)--> DLP-1 :1344 / DLP-2 :1344 / AV :1344 (пул ICAP-серверов с балансировкой)

В отличие от inline-сервисов, ICAP-серверы подключаются по IP-сети и не требуют выделенных VLAN.

Маршрутизируемый адрес ICAP-сервера

ICAP-серверу не требуется находиться в датаплейн-подсети Амибиг. Адрес участника пула может лежать вне её и достигаться через шлюз участника — поле «Шлюз к сервису» (`to-gateway`) участника именованного пула. Сам шлюз при этом обязан быть on-link, то есть принадлежать одной из датаплейн-подсетей. Так же адресуются участники inline-сервисов, и так же устроена модель F5, где ICAP-сервер — обычный член пула, достижимый штатной L3-маршрутизацией.

Пустой шлюз означает прежнюю семантику on-link: адрес участника должен принадлежать датаплейн-подсети. Участник с адресом вне датаплейн-подсети и без достижимого шлюза отклоняется при проверке применения.

Типичные сценарии интеграции

Направление проверки выбирают по задаче внешней системы: контроль исходящих запросов, входящих ответов или обоих направлений.

DLP (предотвращение утечек данных)

DLP-система проверяет исходящие HTTP-запросы на наличие конфиденциальной информации:

Параметр	Значение
<code>request-mod.enabled</code>	<code>true</code>
<code>request-mod.uri</code>	<code>icap://dlp-server/reqmod</code>
<code>response-mod.enabled</code>	<code>false</code>
<code>preview-length</code>	<code>4096</code>

Антивирусный шлюз

Антивирус проверяет входящие HTTP-ответы на наличие вредоносного кода:

Параметр	Значение
<code>request-mod.enabled</code>	<code>false</code>

Параметр	Значение
response-mod.enabled	true
response-mod.uri	icap://av-server/avscan
preview-length	1024

DLP + антивирус (оба направления)

Полная инспекция запросов и ответов:

Параметр	Значение
request-mod.enabled	true
request-mod.uri	icap://dlp-server/reqmod
response-mod.enabled	true
response-mod.uri	icap://dlp-server/respmod
preview-length	1024

Пример конфигурации (YANG)

Пример включает проверку запросов и ответов на двух ICAP-серверах с циклическим распределением соединений.

```

services:
  service:
    - name: dlp
      type: icap
      enabled: true
      service-down-action: ignore
      icap:
        request-mod:
          enabled: true
          uri: "icap://198.19.97.10/reqmod"
          preview-length: 1024
        response-mod:
          enabled: true
          uri: "icap://198.19.97.10/respmod"
          preview-length: 1024
      http:
        max-header-count: 128
        max-header-size: 65536
        request-chunking: sustain
        response-chunking: unchunk
      pool:
        member:
          - address: "198.19.97.10"
            port: 1344
          - address: "198.19.97.11"
            port: 1344
        load-balancing: round-robin
      monitor:
        type: tcp
        interval: 5

```

```
timeout: 16
fail-threshold: 3
recover-threshold: 2
```

Диагностика

Диагностику начинают с проверки сетевой доступности ICAP-сервера и ответа на запрос OPTIONS.

Проверка доступности ICAP-сервера

На экране **Сервисы** для ICAP-сервиса доступна кнопка **Проверить ICAP**. Проверка выполняет по указанному URI два шага и показывает результат каждого:

- **Ping** — ICMP-проверка доступности адреса ICAP-сервера.
- **ICAP OPTIONS** — отправка ICAP-запроса OPTIONS на указанный URI. Успешный ответ подтверждает, что ICAP-сервис доступен и функционирует.

Вердикты ICAP-сервера по конкретным запросам (пропустить/модифицировать/заблокировать) в журнал событий не выносятся — результат работы ICAP виден по фактическому поведению трафика.

Типичные проблемы

В таблице приведены основные ошибки сетевого подключения, URI и обработки HTTP-сообщений.

Симптом	Возможная причина	Решение
ICAP-сервер не отвечает на OPTIONS	Сервер выключен или недоступен	Проверить доступность по IP, порт 1344
ICAP возвращает 500	Ошибка конфигурации на стороне сервера	Проверить URI и настройки ICAP-сервера
Высокая задержка при инспекции	Preview отключен или слишком мал	Увеличить <code>preview-length</code>
Большие файлы не проходят проверку	Превышен лимит на стороне ICAP-сервера	Увеличить лимиты на ICAP-сервере
Chunked-ответы некорректны	Несовместимость chunked-кодирования	Изменить <code>response-chunking</code> на <code>unchunk</code>
Применение отклонено с ошибкой <code>not in any F-Stack subnet</code>	Адрес ICAP-сервера вне датаплейн-подсети, а шлюз участника не задан или сам не on-link	Задать участнику именованного пула шлюз (<code>to-gateway</code>) из датаплейн-подсети. Для участника с корректным шлюзом эта ошибка больше не возникает

Пулы сервисов

В разделе описаны состав пулов, методы распределения нагрузки и контроль доступности участников.

Назначение

Пул (pool) — это группа серверов или устройств, обеспечивающих работу сервиса. Каждый сервис в Амибиг (inline или ICAP) использует пул с одним или несколькими участниками. Пул обеспечивает балансировку нагрузки между участниками и автоматическое исключение неработоспособных узлов на основании данных мониторинга.

Пул задаётся одним из двух способов:

- **встроенный пул** — блок `pool` внутри сервиса;
- **именованный пул** — отдельный объект на экране Пулы, на который сервис ссылается параметром `pool-ref`. Именованный пул можно переиспользовать в нескольких сервисах; если `pool-ref` задан, встроенный блок `pool` игнорируется.

Архитектура

Сервис направляет трафик участникам пула по выбранному методу балансировки; монитор постоянно проверяет каждого участника и исключает недоступных:

Схема взаимодействия

Источник	Связь	Назначение
Сервис	→	Пул
Пул	→	M1 (UP)
Пул	→	M2 (UP)
Пул	→	M3 (DOWN)
Пул	→	M4 (UP)
Монитор	↔	M1 (UP)
Монитор	↔	M2 (UP)
Монитор	↔	M3 (DOWN)
Монитор	↔	M4 (UP)

Сервис --> Пул (Балансировщик: round-robin / session-hash / predictive / least-connections) --> Участники: M1 (UP), M2 (UP), M3 (DOWN), M4 (UP) <-- Монитор (tcp / icmp / http-get)

Участники пула

Каждый участник пула — это конкретный адрес и порт устройства или сервера.

Параметры участника

Параметры участника определяют его сетевой адрес и административное состояние.

Параметр	Тип	По умолчанию	Описание
<code>address</code>	IP-адрес	—	IP-адрес участника (IPv4 или IPv6)
<code>port</code>	0-65535	—	Порт. Для inline-устройств обычно 0, для ICAP — 1344
<code>enabled</code>	boolean	<code>true</code>	Административное состояние. Если <code>false</code> — участник исключен из балансировки. Переключатель участника в веб-интерфейсе (экран Сервисы для встроенного пула, экран Пулы для именованного) позволяет вывести узел на обслуживание без удаления записи и потери настроек
<code>to-gateway</code>	IP-адрес	—	Шлюз для маршрутизации к участнику (только в именованных пулах)

Состояния участника

Каждый участник пула может находиться в одном из следующих состояний:

Состояние	Описание	Участвует в балансировке
UP	Монитор успешно проверяет доступность	Да

Состояние	Описание	Участвует в балансировке
DOWN	Монитор обнаружил недоступность (превышен fail-threshold)	Нет
DISABLED	Администратор выключил участника (<code>enabled: false</code>)	Нет
RECOVERING	Участник восстанавливается (не достиг recover-threshold)	Нет

Переход между состояниями определяется пороговыми значениями монитора:

`UP --(fail-threshold последовательных неудач)--> DOWN --(recover-threshold последовательных успехов)--> UP`

Пороги настраиваются в веб-интерфейсе: для встроенного пула — в форме сервиса на экране **Сервисы** (поля «Отказов подряд до вывода» и «Успехов подряд до возврата»), для именованного пула — на экране **Пулы**.

Кроме результатов монитора, на состояние участника влияет проверка пути данных: если реальное соединение к участнику не удалось установить, участник переходит в карантин и отображается как недоступный — даже если активная проба к нему проходит. Переходы участника между состояниями UP и DOWN записываются в журнал с указанием вида пробы. Подробнее: [Мониторинг доступности](#).

Балансировка нагрузки

Метод балансировки определяет выбор доступного участника пула для нового соединения.

Методы балансировки

Амибиг поддерживает циклическое распределение, привязку по хешу сессии и методы, учитывающие текущую нагрузку.

Метод	Описание	Когда использовать
<code>round-robin</code>	Циклическое распределение запросов между участниками	ICAP-серверы с одинаковой производительностью
<code>session-hash</code>	Привязка сессии к участнику на основе хеша (IP-адреса клиента)	Inline-устройства с сохранением состояния сессий
<code>predictive</code>	Выбор участника на основе предсказания нагрузки	Разнородные серверы с переменной загрузкой

Метод	Описание	Когда использовать
<code>least-connections</code>	Направление на участника с наименьшим числом активных соединений	ICAP-серверы с разной производительностью

Рекомендации по выбору метода

Выбор метода зависит от того, хранит ли внешнее средство защиты состояние обрабатываемой сессии.

Тип сервиса	Рекомендуемый метод	Обоснование
Inline (NGFW)	<code>session-hash</code>	NGFW ведет таблицу состояний; переключение сессии на другое устройство приведет к потере контекста
Inline (IPS)	<code>session-hash</code>	IPS анализирует потоки; необходима привязка сессии
ICAP (DLP)	<code>round-robin</code>	Каждый HTTP-запрос/ответ проверяется независимо
ICAP (антивирус)	<code>least-connections</code>	Время проверки зависит от размера контента

Плавный ввод в эксплуатацию

При восстановлении участника пула после отказа можно настроить период плавного увеличения нагрузки:

Параметр	Тип	По умолчанию	Описание
<code>slow-ramp-time</code>	секунды	0	Период плавного увеличения нагрузки после восстановления. Значение 0 отключает

В течение периода `slow-ramp` участник получает меньшую долю трафика, пропорционально времени, прошедшему с момента восстановления. Это предотвращает перегрузку только что восстановленного узла.

В веб-интерфейсе параметр задаётся полем «Плавный ввод под нагрузку»: для встроенного пула — в форме сервиса на экране `Сервисы`, для именованного пула — на экране `Пулы`.

Влияние пула на аварийное переключение

В отказоустойчивой паре состояние участников пула влияет на оценку работоспособности узла через параметры сервиса:

- `min-healthy-members` — минимальное число доступных участников пула (1–255, по умолчанию 1). Если доступных участников меньше, сервис считается деградировавшим при расчёте оценки работоспособности узла.
- `weight` — вес сервиса в оценке работоспособности узла. Чем выше вес, тем сильнее деградация сервиса влияет на право узла оставаться активным.

При импорте конфигурации F5 значение `min-up-members` пула, превышающее 255, приводится к 255 — максимальному значению модели Амибиг. Подробнее о посервисном аварийном переключении: Обзор сервисов, Active/Standby.

Связь пулов с типами сервисов

Состав пула и способ проверки участников зависят от типа внешнего сервиса.

Пул для inline-сервиса

В inline-сервисе участники пула представляют физические или виртуальные устройства безопасности:

- **Inline-сервис "ngfw"**
 - to-service: VLAN 100
 - from-service: VLAN 200
 - pool:
 - участник: 198.19.64.7:0 (NGFW-1)
 - участник: 198.19.64.8:0 (NGFW-2)
 - load-balancing: session-hash
 - monitor: icmp, interval=5

Порт 0 означает, что трафик передается на уровне L2 без привязки к конкретному порту.

Пул для ICAP-сервиса

В ICAP-сервисе участники пула — это ICAP-серверы, доступные по IP-сети:

- **ICAP-сервис "dlp"**
 - request-mod: icap://server/reqmod
 - response-mod: icap://server/respmo
 - pool:
 - участник: 198.19.97.10:1344 (DLP-1)
 - участник: 198.19.97.11:1344 (DLP-2)

- load-balancing: round-robin
- monitor: tcp, interval=5

Стандартный порт ICAP-сервера — 1344 .

Пример конфигурации (YANG)

Примеры показывают конфигурацию пулов для inline- и ICAP-сервисов.

Пул inline-сервиса

В пуле inline-сервиса используется привязка по хешу сессии и ICMP-монитор.

```
pool:
  member:
    - address: "198.19.64.7"
      port: 0
      enabled: true
    - address: "198.19.64.8"
      port: 0
      enabled: true
  load-balancing: session-hash
  monitor:
    type: icmp
    interval: 5
    timeout: 16
    fail-threshold: 3
    recover-threshold: 2
    slow-ramp-time: 30
```

Пул ICAP-сервиса

В пуле ICAP-сервиса соединения распределяются циклически, а доступность проверяется по TCP.

```
pool:
  member:
    - address: "198.19.97.10"
      port: 1344
      enabled: true
    - address: "198.19.97.11"
      port: 1344
      enabled: true
  load-balancing: round-robin
  monitor:
    type: tcp
    interval: 5
    timeout: 16
    fail-threshold: 3
    recover-threshold: 2
    slow-ramp-time: 0
```

Управление через веб-интерфейс

В веб-интерфейсе можно создать встроенный или именованный пул и проверить состояние его участников.

Создание встроенного пула

Встроенный пул настраивается непосредственно в форме сервиса.

1. Перейдите в раздел **Сервисы**.
2. Создайте или откройте существующий сервис.
3. В разделе **Пул** выберите **Встроенный** и нажмите **Добавить участника**.
4. Укажите IP-адрес и порт.
5. Выберите метод балансировки.
6. Настройте монитор: тип, интервал, таймаут, пороги «Отказов подряд до вывода» и «Успехов подряд до возврата» (см. Мониторинг доступности).
7. При необходимости задайте «Плавный ввод под нагрузку» (`slow-ramp-time`).
8. Нажмите **Сохранить**.

Создание именованного пула

Именованный пул создаётся отдельно и может использоваться несколькими сервисами.

1. Перейдите в раздел **Пулы**.
2. Нажмите **Добавить пул**, укажите имя и участников.
3. Выберите метод балансировки и монитор — именованный (с экрана **Мониторы**) или встроенный.
4. В сервисе выберите **Именованный пул** и укажите созданный пул.

Мониторинг состояния

В веб-интерфейсе состояние каждого участника отображается цветовым индикатором:

Цвет	Состояние
Зеленый	UP — участник доступен
Красный	DOWN — участник недоступен
Серый	DISABLED — участник выключен администратором
Желтый	RECOVERING — участник восстанавливается

Диагностика

При диагностике проверьте состояние участников, параметры монитора и выбранный метод балансировки.

Типичные проблемы

В таблице приведены причины недоступности участников и неравномерного распределения нагрузки.

Симптом	Возможная причина	Решение
Все участники DOWN	Некорректный тип монитора	Проверить тип монитора, порт, интервал
Неравномерное распределение	<code>session-hash</code> при малом числе клиентов	Рассмотреть <code>round-robin</code> или <code>least-connections</code>
Участник постоянно переключается UP/DOWN	Таймаут монитора слишком мал	Увеличить <code>timeout</code> и <code>fail-threshold</code>
Новый участник перегружается при вводе	<code>slow-ramp-time</code> не настроен	Установить <code>slow-ramp-time</code> (например, 30 секунд)

Мониторинг доступности

В разделе описаны типы мониторов доступности, правила изменения состояния участников пула и рекомендации по выбору параметров проверки.

Назначение

Монитор доступности автоматически проверяет участников пула сервиса. На основании результатов Амибиг включает или исключает участника из балансировки нагрузки. Состояние участников также учитывается в оценке работоспособности узла при аварийном переключении — см. Обзор сервисов.

Монитор задаётся одним из двух способов:

- **именованный монитор** — отдельный объект на экране `Мониторы`, на который пул ссылается по имени. Именованный монитор можно переиспользовать в нескольких пулах;
- **встроенный монитор** — блок настроек внутри пула (для встроенного пула сервиса доступны типы `tcp`, `icmp`, `http-get`).

Проверка выполняется по тому сетевому пути, по которому адрес участника достижим со стороны dataplane.

Схема взаимодействия

Источник	Связь	Назначение
Монитор	→	Результат

Источник	Связь	Назначение
Результат	→ OK	UP
Результат	→ FAIL	DOWN
UP	→	Монитор
DOWN	→	Монитор

Типы мониторов

Амибиг поддерживает четыре типа проверок:

Тип	Протокол	Описание	Типичное применение
tcp	TCP	Проверка установления TCP-соединения на указанный адрес и порт	ICAP-серверы, серверы приложений
icmp	ICMP	Отправка ICMP Echo Request (ping) и ожидание ответа	Inline-устройства
http-get	HTTP	Отправка HTTP GET-запроса и проверка ответа	ICAP-серверы с HTTP-интерфейсом
http-post	HTTP	Отправка HTTP POST-запроса с телом и проверка кода и содержимого ответа	Серверы с прикладной проверкой состояния (только именованные мониторы)

TCP-монитор

TCP-монитор выполняет попытку установления TCP-соединения (TCP handshake) на заданный адрес и порт участника пула.

Амибиг — SYN --> Участник пула — SYN+ACK --> Амибиг — ACK --> Участник пула — FIN --> Закрытие. Результат: UP

Проверка считается успешной, если TCP-соединение установлено в пределах заданного таймаута. После успешного подключения соединение сразу закрывается.

Когда использовать: универсальный монитор для сервисов, доступных по IP-сети, — например, ICAP-серверов (порт 1344). Для inline-сервисов TCP-монитор

активную пробу не выполняет (см. «Особенности мониторинга inline-сервисов» ниже).

ICMP-монитор

ICMP-монитор отправляет ICMP Echo Request (ping) на адрес участника пула.

Амибиг — Echo Request --> Участник пула — Echo Reply --> Амибиг. Результат: UP (RTT)

Когда использовать: inline-устройства. Обратите внимание, что ICMP-мониторинг проверяет только сетевую доступность, но не работоспособность сервиса.

HTTP GET-монитор

HTTP GET-монитор отправляет HTTP GET-запрос на указанный URL и проверяет код ответа. Для именованных мониторов строка запроса задаётся полем `send-string`, ожидаемая подстрока ответа — полем `receive-string`.

Амибиг — GET / --> Участник пула — 200 OK --> Амибиг. Результат: UP

Когда использовать: ICAP-серверы, предоставляющие HTTP-интерфейс для проверки состояния. Для ICAP-серверов также доступна ручная проверка ICAP OPTIONS (кнопка **Проверить ICAP** на экране `Сервисы`).

HTTP POST-монитор

HTTP POST-монитор (только именованные мониторы) отправляет HTTP POST-запрос с заданным телом и проверяет ответ по коду состояния и содержимому.

Параметр	Описание
<code>send-body</code>	Тело POST-запроса
<code>content-type</code>	Значение заголовка Content-Type для тела запроса
<code>expect-status</code>	Ожидаемый HTTP-код: пусто (любой 2xx), точный код, список через запятую или диапазон <code>lo-hi</code>
<code>receive-disable</code>	Подстрока ответа, при наличии которой участник принудительно переводится в DOWN
<code>request-header</code>	Дополнительные заголовки запроса (имя/значение)

Когда использовать: серверы, у которых состояние проверяется прикладным запросом, например ресурс проверки, требующий тело запроса.

Особенности мониторинга inline-сервисов

Для inline-сервисов активную проверку выполняет только монитор типа `icmp`. Мониторы `tcp` и `http-get` для inline-сервисов активную пробу не выполняют: адрес участника указывает на само инспектирующее устройство, и TCP-проверка к нему не подтверждает прохождение трафика через мост.

Постоянную активную проверку прохождения трафика через inline-сервис выполняет сквозная проба: она открывает настоящее TCP-соединение к адресу за сервисом тем же путём, которым идёт рабочий трафик, — см. Inline-сервисы (NGFW). Если сквозная проба не включена, отказ пересылки внутри цепочки фиксируется по фактическому трафику: соединение, потерянное из-за отказа звена, записывается в журнал потоков с причиной `CHAIN_FAIL`. Разовая ручная проверка прохождения пробного трафика через inline-устройство доступна кнопкой **Проверить Inline** на экране `Сервисы`.

Параметры монитора

Параметры монитора задают способ проверки, её периодичность и пороги изменения состояния участника.

Параметр	Тип	По умолчанию	Описание
<code>type</code>	enum	<code>tcp</code>	Тип проверки: <code>tcp</code> , <code>icmp</code> , <code>http-get</code> , <code>http-post</code>
<code>interval</code>	uint16 (секунды)	<code>5</code>	Интервал между проверками
<code>timeout</code>	uint16 (секунды)	<code>16</code>	Максимальное время ожидания ответа
<code>fail-threshold</code>	uint8	<code>3</code>	Количество последовательных неудач для перевода в DOWN
<code>recover-threshold</code>	uint8	<code>2</code>	Количество последовательных успехов для перевода в UP
<code>send-string</code>	string	—	Строка запроса для <code>http-get</code>
<code>receive-string</code>	string	—	Ожидаемая подстрока ответа для <code>http-get</code>

В веб-интерфейсе пороги задаются полями «Отказов подряд до вывода» (`fail-threshold`) и «Успехов подряд до возврата» (`recover-threshold`): для встроенного монитора пула — в форме сервиса на экране `Сервисы` , для именованного монитора — на экране `Мониторы` , для именованного пула — на экране `Пулы` .

Соотношение `interval` и `timeout`

Параметр `timeout` определяет суммарное время ожидания ответа на одну проверку. Параметр `interval` определяет частоту запуска проверок.

Рекомендуемое соотношение:

```
timeout >= interval * (fail-threshold - 1) + interval
```

Это гарантирует, что за время таймаута будет выполнено достаточное количество проверок для принятия решения.

Пример по умолчанию:

```
interval = 5 секунд
timeout  = 16 секунд
fail-threshold = 3
```

```
Время обнаружения отказа: 5 * 3 = 15 секунд (3 неудачных проверки)
Время восстановления:      5 * 2 = 10 секунд (2 успешных проверки)
```

Логика определения состояния

Состояние участника изменяется только после достижения заданного порога последовательных успешных или неуспешных проверок.

Переход UP -> DOWN

Переход в состояние DOWN происходит после `fail-threshold` последовательных неудач.

```
Проверка 1: FAIL (счетчик неудач = 1)
Проверка 2: FAIL (счетчик неудач = 2)
Проверка 3: FAIL (счетчик неудач = 3, достигнут fail-threshold)
--> Участник переведен в DOWN
```

При первой же успешной проверке счетчик неудач сбрасывается.

Переход DOWN -> UP

Возврат в состояние UP происходит после `recover-threshold` последовательных успешных проверок.


```

Проверка 1: OK      (счетчик успехов = 1)
Проверка 2: OK      (счетчик успехов = 2, достигнут recover-threshold)
--> Участник переведен в UP

```

При первой же неудачной проверке счетчик успехов сбрасывается.

Диаграмма состояний

Основные переходы между состояниями зависят от результатов проверок и административного признака `enabled`.

- **UP** — при достижении `fail-threshold` последовательных неудач --> переход в **DOWN**
- **DOWN** — при достижении `recover-threshold` последовательных успехов --> переход в **UP**
- **DISABLED** — при установке `enabled = false` из любого состояния; при `enabled = true` --> возврат в **UP**

Карантин по вердикту пути данных

Кроме активных проверок, на статус здоровья влияет сам путь данных: если реальное соединение к участнику пула не удалось установить, участник уходит в карантин и показывается недоступным — даже когда активная проба к нему проходит. Это закрывает класс отказов, при котором проверка доступности успешна (например, ICMP до соседа в своей же подсети), а рабочие соединения через сервис не проходят.

Журналирование переходов

Каждый переход участника между состояниями UP и DOWN записывается в журнал с указанием сервиса, адреса участника, вида пробы (сквозная или обычная) и счётчиков отказов и восстановлений.

Рекомендации по настройке

Параметры проверки выбирают с учётом типа сервиса и требуемого времени обнаружения отказа.

Для inline-сервисов (NGFW)

Для inline-сервисов используйте ICMP-проверку с порогами, которые исключают вывод участника из пула из-за единичной потери пакета.

Параметр	Рекомендация	Обоснование
<code>type</code>	<code>icmp</code>	Единственный тип с активной пробой для inline-сервисов

Параметр	Рекомендация	Обоснование
<code>interval</code>	5 секунд	Баланс между скоростью обнаружения и нагрузкой
<code>timeout</code>	16 секунд	Запас для трех проверок
<code>fail-threshold</code>	3	Защита от ложных срабатываний
<code>recover-threshold</code>	2	Быстрое восстановление

Для ICAP-сервисов (DLP)

Для ICAP-сервиса достаточно проверять установление TCP-соединения на стандартный порт сервера.

Параметр	Рекомендация	Обоснование
<code>type</code>	<code>tcp</code>	Проверка порта 1344
<code>interval</code>	5 секунд	Стандартный интервал
<code>timeout</code>	16 секунд	Стандартный таймаут
<code>fail-threshold</code>	3	Защита от ложных срабатываний
<code>recover-threshold</code>	2	Быстрое восстановление

Для критичных сервисов (быстрое обнаружение)

Для критичного сервиса интервал и порог отказа можно уменьшить, сохранив более осторожный порог возврата в работу.

Параметр	Рекомендация	Обоснование
<code>type</code>	<code>tcp</code>	Наиболее точный метод
<code>interval</code>	2 секунды	Частые проверки
<code>timeout</code>	8 секунд	Сокращенный таймаут
<code>fail-threshold</code>	2	Быстрый переход в DOWN
<code>recover-threshold</code>	3	Осторожное восстановление

Пример конфигурации (YANG)

Примеры показывают базовые наборы параметров для разных способов проверки доступности.

Стандартный TCP-монитор

Этот вариант проверяет установление TCP-соединения каждые пять секунд.

```
monitor:
  type: tcp
  interval: 5
  timeout: 16
  fail-threshold: 3
  recover-threshold: 2
```

ICMP-монитор для inline-устройств

Этот вариант проверяет сетевую доступность inline-устройства с помощью ICMP.

```
monitor:
  type: icmp
  interval: 5
  timeout: 16
  fail-threshold: 3
  recover-threshold: 2
```

Агрессивный мониторинг (быстрое обнаружение)

Этот вариант сокращает время обнаружения отказа за счёт интервала в две секунды и порога из двух неудач.

```
monitor:
  type: tcp
  interval: 2
  timeout: 8
  fail-threshold: 2
  recover-threshold: 3
```

Взаимодействие с service-down-action

Когда монитор переводит **всех** участников пула в состояние DOWN, пул считается недоступным. В этом случае к соединениям клиентов применяется настройка `service-down-action` сервиса:

- **ignore** — трафик идет в обход недоступного сервиса без проверки этим сервисом (`fail-open`; `bypass` — устаревший синоним)
- **drop** — трафик отбрасывается
- **reset** — клиенту отправляется TCP RST

Кроме того, падение числа доступных участников ниже `min-healthy-members` переводит сервис в деградировавшее состояние при расчёте оценки работоспособности узла для аварийного переключения — см. Обзор сервисов.

Управление через веб-интерфейс

Веб-интерфейс показывает состояние участников пула и предоставляет команды для разовой проверки сервисов.

Просмотр состояния мониторинга

В разделе **Сервисы > [имя сервиса]** для каждого участника пула отображается:

- Текущее состояние (UP / DOWN / DISABLED).
- Время последней проверки.
- Время нахождения в текущем состоянии.
- RTT последней успешной проверки (для ICMP).

Ручная диагностика

На экране **Сервисы** доступны ручные проверки:

Проверка	Описание
Проверить ICAP	Ping адреса ICAP-сервера и отправка ICAP OPTIONS-запроса по указанному URI (для ICAP-сервисов)
Проверить Inline	Пошаговая проверка inline-сервиса: интерфейсы плеч, ping участников, прохождение пробного UDP- и TCP-трафика через устройство, статус монитора (для inline-сервисов)

Диагностика

При диагностике сопоставьте состояние участника с сетевой доступностью, типом монитора и установленными порогами.

Типичные проблемы

В таблице приведены распространённые причины перехода участников в состояние DOWN и действия администратора.

Симптом	Возможная причина	Решение
Участник постоянно DOWN	Порт закрыт, устройство недоступно	Проверить сетевое подключение и настройки межсетевого экрана
Участник переключается UP/ DOWN	Нестабильная сеть, таймаут слишком мал	Увеличить <code>timeout</code> и <code>fail-threshold</code>

Симптом	Возможная причина	Решение
Все участники DOWN после применения конфигурации	Некорректный тип монитора	Проверить тип монитора (<code>tcp</code> или <code>icmp</code>) и порт
Ping проходит, но TCP-монитор показывает DOWN	Порт закрыт на устройстве	Проверить, что сервис слушает на указанном порту
Долгое восстановление	Высокий <code>recover-threshold</code>	Уменьшить <code>recover-threshold</code>

Виртуальные серверы

В разделе описаны публикация сервисов на VIP-адресах, балансировка соединений и дополнительные параметры виртуальных серверов.

Назначение

Виртуальный сервер публикует сервис на VIP-адресе и распределяет клиентские соединения между участниками пула. Амибиг принимает клиентское соединение, выбирает участника и открывает к нему origin-соединение.

Этим виртуальный сервер отличается от сервисов и цепочек инспекции:

- **Сервисы и цепочки** обрабатывают транзитный трафик прозрачно: клиент обращается к настоящему серверу назначения, а Амибиг стоит на пути трафика и незаметно для сторон проводит его через средства защиты. Адрес назначения не меняется.
- **Виртуальный сервер** — явная точка входа: клиенты обращаются к адресу самого Амибиг, а настоящие серверы скрыты за ним в пуле.

Виртуальные серверы применяют, когда нужно опубликовать ферму серверов под одним адресом: балансировка веб-приложения, распределение нагрузки между узлами внутреннего сервиса, терминация TLS на Амибиг (SSL offload). Для прозрачной инспекции проходящего трафика виртуальный сервер не нужен — это задача политики и цепочек сервисов.

Не путайте VIP виртуального сервера с VIP HA-кластера: HA VIP — это плавающий адрес отказоустойчивой пары, который мигрирует между узлами при переключении. В HA-паре адрес виртуального сервера обслуживает активный узел.

Обработка соединения

Каждое входящее соединение проходит одинаковый путь: проверку лимитов, поиск персистентности, выбор участника пула и подключение к нему.

Схема взаимодействия

Источник	Связь	Назначение
Клиент	→	VIP-слушатель
VIP-слушатель	→	Проверка лимитов
Проверка лимитов	→	Персистентность
Персистентность	→	Выбор участника
Выбор участника	→	Участник M1
Выбор участника	→	Участник M2

Порядок обработки:

1. Соединение принимается слушателем на `address:port` виртуального сервера.
2. Проверяется `connection-limit`: при превышении соединение закрывается.
3. Проверяется `rate-limit`: при превышении числа новых соединений за текущую секунду соединение закрывается.
4. Если настроена персистентность и для клиента есть действующая запись — выбирается закреплённый участник (если он доступен).
5. Иначе участник выбирается методом балансировки среди доступных (UP) участников пула. Если доступных нет — соединение закрывается.
6. Открывается `origin`-соединение к участнику, и данные передаются между клиентским и `origin`-соединением.

Важно. В текущей версии виртуальные серверы обслуживают только TCP. Поле `protocol` принимает значение `udp`, но такой виртуальный сервер не обслуживает трафик.

Параметры виртуального сервера

Параметры определяют адрес слушателя, пул серверов и ограничения для входящих соединений.

Параметр	Тип	По умолчанию	Описание
<code>name</code>	string	—	Уникальное имя виртуального сервера (ключ)
<code>description</code>	string	—	Текстовое описание

Параметр	Тип	По умолчанию	Описание
<code>enabled</code>	boolean	<code>true</code>	Административное состояние. <code>false</code> — слушатель не создаётся
<code>address</code>	IP-адрес	—	VIP-адрес (IPv4 или IPv6), обязательный
<code>port</code>	0-65535	—	Порт слушателя, обязательный
<code>protocol</code>	enum	<code>tcp</code>	Протокол: <code>tcp</code> или <code>udp</code> . Обработка трафика реализована только для <code>tcp</code>
<code>pool-ref</code>	string	—	Ссылка на именованный пул с экрана Пулы
<code>pool-config</code>	container	—	Встроенный пул (используется, когда <code>pool-ref</code> не задан)
<code>persistence</code>	container	<code>none</code>	Настройки персистентности
<code>connection-limit</code>	uint32	<code>0</code>	Максимум одновременных соединений (<code>0</code> — без ограничений)
<code>rate-limit</code>	uint32	<code>0</code>	Максимум новых соединений в секунду (<code>0</code> — без ограничений)
<code>ssl</code>	container	выключен	Терминация TLS на виртуальном сервере (SSL offload)

Всего можно определить до 32 виртуальных серверов.

Пул и методы балансировки

Пул задаётся одним из двух способов — так же, как в сервисах:

- **именованный пул** (`pool-ref`) — отдельный объект с экрана Пулы , переиспользуемый между виртуальными серверами и сервисами; вместе с пулом виртуальный сервер получает его метод балансировки и монитор;

- **встроенный пул** (`pool-config`) — список участников прямо в виртуальном сервере, для простых схем.

Каждый участник встроенного пула описывается адресом, портом, признаком `enabled` и весом `weight` (1–100) для взвешенных методов. Методы балансировки:

Метод	Описание
<code>round-robin</code>	Циклическое распределение соединений между доступными участниками
<code>session-hash</code>	Привязка по хешу IP-адреса клиента: один клиент стабильно попадает на одного участника
<code>least-connections</code>	Новое соединение направляется к участнику с наименьшим числом активных соединений
<code>ratio</code>	Взвешенное распределение пропорционально <code>weight</code> участников
<code>dynamic-ratio</code>	Взвешенное распределение по динамическим весам, получаемым от мониторинга; при отсутствии данных используется статический <code>weight</code>

Недоступные участники исключаются из балансировки по данным монитора доступности. Подробнее см. в разделах «Мониторы доступности» и «Пулы».

Персистентность

Персистентность закрепляет клиента за конкретным участником пула, чтобы последовательные соединения одного клиента попадали на один и тот же сервер (важно для приложений, хранящих сессию локально).

Параметр	Тип	По умолчанию	Описание
<code>type</code>	enum	<code>none</code>	<code>none</code> , <code>source-address</code> или <code>cookie</code>
<code>timeout</code>	секунды	<code>300</code>	Срок жизни записи персистентности
<code>cookie-name</code>	string	<code>AMIBIG_PERSIST</code>	Имя cookie. В текущей версии используется значение по умолчанию

Типы персистентности:

- **По адресу источника (`source-address`)** — привязка по IP-адресу клиента. Запись «клиент → участник» хранится в таблице персистентности и обновляется при каждом соединении. Если закреплённый участник перешёл в состояние DOWN, выбирается новый.
- **Cookie** — привязка по cookie `AMIBIG_PERSIST`. В первый HTTP-ответ участника Амибиг добавляет заголовок `Set-Cookie`, по которому последующие запросы клиента возвращаются на того же участника. Cookie-персистентность предназначена для нешифрованного HTTP-трафика; для виртуального сервера с SSL offload используйте привязку по адресу источника (`source-address`).

Ограничение соединений

Два независимых лимита ограничивают входящую нагрузку на виртуальный сервер:

- `connection-limit` — максимальное число одновременных соединений. Новое соединение сверх лимита закрывается сразу после приёма.
- `rate-limit` — максимальное число новых соединений в секунду. Подсчёт ведётся в окне длительностью одну секунду; соединения сверх лимита закрываются.

Значение `0` в обоих случаях означает отсутствие ограничения.

SSL-настройки

Виртуальный сервер может завершать клиентский TLS (SSL offload). В этом режиме клиентское соединение защищено TLS, а к участникам пула передаются незашифрованные прикладные данные.

Параметр	Тип	По умолчанию	Описание
<code>ssl.enabled</code>	boolean	<code>false</code>	Включить терминацию TLS
<code>ssl.certificate</code>	string	—	Путь к сертификату в формате PEM
<code>ssl.key</code>	string	—	Путь к закрытому ключу в формате PEM

SSL-настройки задаются в секции `virtual-servers` через REST API; в веб-интерфейсе этих полей нет. Перед открытием слушателя Амибиг проверяет сертификат и закрытый ключ. Если проверка не пройдена, применение конфигурации завершается ошибкой и слушатель не открывается.

Создание через экран «Виртуальные серверы»

На экране «Виртуальные серверы» можно настроить слушатель, пул, персистентность и ограничения соединений.

1. Перейдите в раздел **Виртуальные серверы**.
2. Нажмите **Добавить виртуальный сервер**, укажите имя и описание.
3. В разделе **Слушатель** задайте VIP-адрес, порт и протокол.
4. В разделе **Пул** выберите именованный пул или оставьте поле пустым и заполните встроенный пул: участники (адрес, порт, вес) и метод балансировки.
5. В разделе **Персистентность** выберите тип и таймаут.
6. В разделе **Ограничения** при необходимости задайте максимальное число одновременных и новых соединений.
7. Нажмите **Сохранить**, затем примените конфигурацию на экране **Применить конфигурацию**.

Пример конфигурации

Пример публикует веб-приложение на TCP-порту 443 и распределяет соединения между двумя участниками встроенного пула.

```
{
  "virtual-servers": {
    "virtual-server": [
      {
        "name": "web-farm",
        "description": "Публикация веб-приложения",
        "enabled": true,
        "address": "192.0.2.100",
        "port": 443,
        "protocol": "tcp",
        "pool-config": {
          "member": [
            { "address": "10.10.1.11", "port": 8443, "enabled": true, "weight": 2 },
            { "address": "10.10.1.12", "port": 8443, "enabled": true, "weight": 1 }
          ],
          "load-balancing": "ratio"
        },
        "persistence": { "type": "source-address", "timeout": 600 },
        "connection-limit": 10000,
        "rate-limit": 500
      }
    ]
  }
}
```

Секция доступна через REST API: GET /api/v1/config/virtual-servers и PUT /api/v1/config/virtual-servers ; изменения вступают в силу после POST /api/v1/config/apply .

GSLB (DNS-балансировка)

В разделе описаны настройка DNS-слушателя GSLB, методы выбора площадки и использование географических данных.

Назначение

GSLB (Global Server Load Balancing) распределяет клиентов между площадками на уровне DNS. В DNS-ответе Амибиг возвращает VIP-адрес одной из площадок. Адрес выбирается циклически, по весу, по региону источника или по порядку приоритета. Это позволяет обслуживать одно доменное имя, например `www.example.com`, на нескольких площадках.

Основной объект GSLB — **wide-IP**: соответствие полного доменного имени (FQDN) пулу VIP-адресов. Для каждого запроса этого имени слушатель выбирает один адрес и включает его в DNS-ответ.

Схема взаимодействия

Источник	Связь	Назначение
Клиент	→	DNS-запрос <code>www.example.com</code>
DNS-запрос <code>www.example.com</code>	→	DNS-слушатель Амибиг
DNS-слушатель Амибиг	→	Wide-IP
Wide-IP	→	Пул <code>msk 192.0.2.100</code>
Wide-IP	→	Пул <code>spb 198.51.100.100</code>

DNS-слушатель

DNS-запросы обслуживает собственный слушатель в датаплейне — отдельный DNS-сервер разворачивать не нужно. Слушатель работает по UDP и обрабатывает запросы типов A и AAAA.

Параметр	Тип	По умолчанию	Описание
<code>enabled</code>	boolean	<code>false</code>	Включение GSLB (создаёт DNS-слушатель)
<code>dns-listener.address</code>	IP-адрес	<code>0.0.0.0</code>	Адрес слушателя
<code>dns-listener.port</code>	0-65535	<code>53</code>	Порт слушателя

Поведение слушателя:

- имя, для которого определён включённый wide-IP, получает ответ с **одним** адресом и TTL из настроек wide-IP;
- неизвестное имя получает ответ NXDOMAIN;

- запрос A для wide-IP с IPv6-адресами (и наоборот AAAA для IPv4) получает пустой ответ NOERROR;
- запросы других типов не обрабатываются.

Чтобы запросы поступали на слушатель Амибиг, делегируйте ему зону или подзону с помощью NS-записей. Обычное имя хоста без границы зоны через NS-запись не делегируется.

Wide-IP

Каждый wide-IP описывает одно доменное имя и пул адресов для него. Имя сравнивается с FQDN запроса точно (без учёта регистра), шаблоны не поддерживаются.

Параметр	Тип	По умолчанию	Описание
<code>name</code>	string	—	Уникальное имя wide-IP (ключ)
<code>fqdn</code>	string	—	Полное доменное имя, обязательное
<code>enabled</code>	boolean	<code>true</code>	Выключенный wide-IP не отвечает на запросы
<code>lb-method</code>	enum	<code>round-robin</code>	Метод выбора пула
<code>ttl</code>	секунды	<code>30</code>	TTL DNS-ответа
<code>pool</code>	list	—	Пулы — кандидаты для ответа
<code>last-resort</code>	IP-адрес	—	Резервный адрес, когда ни один пул не доступен

Каждый элемент списка `pool` — это одна площадка:

Параметр	Тип	По умолчанию	Описание
<code>name</code>	string	—	Имя пула (на него ссылаются topology-record)
<code>address</code>	IP-адрес	—	VIP-адрес, возвращаемый в DNS-ответе
<code>enabled</code>	boolean	<code>true</code>	Выключенный пул исключается из выбора

Параметр	Тип	По умолчанию	Описание
<code>ratio</code>	uint16	1	Вес для метода <code>ratio</code>
<code>vs-pool-ref</code>	string	—	Зарезервировано. В текущей версии поле не влияет на выбор пула и не получает состояние из LTM-пула

По умолчанию TTL равен 30 секундам. Малое значение сокращает время кеширования ответа. Фактический срок зависит от политики рекурсивного DNS-сервера и клиента.

Методы выбора (lb-method)

Метод `lb-method` определяет, как Амибиг выбирает площадку из списка включённых пулов.

Метод	Описание
<code>round-robin</code>	Пулы выбираются по кругу — равномерное распределение клиентов между площадками
<code>ratio</code>	Взвешенный выбор пропорционально <code>ratio</code> пулов: пул с весом 3 получает втрое больше ответов, чем пул с весом 1
<code>topology</code>	Выбор по географии клиента: регион источника запроса ищется в списке <code>topology-record</code> , и возвращается предпочтительный пул региона. Если запись для региона не найдена или её пул недоступен — откат к <code>round-robin</code>
<code>global-availability</code>	Всегда возвращается первый доступный пул по порядку списка. Пулы образуют приоритетный список: вторая площадка используется, только когда первая недоступна

Во всех методах участвуют только пулы с `enabled: true`. Если таких пулов нет, возвращается `last-resort`.

Важно. Если нет включённых пулов и не задан `last-resort`, слушатель не отправляет DNS-ответ. Клиент ожидает истечения таймаута.

Гео-база

Для метода `topology` (и для гео-критериев политики) нужна база соответствия IP-адресов регионам. Путь к файлу задаётся полем `gslb.geo-database`; база загружается при старте датаплейна. Поддерживаются два формата:

- **MaxMind MMDB** (например, GeoLite2 Country) — доступен в сборках с поддержкой MaxMind; регионом считается двухбуквенный код страны (`RU` , `US` и т. д.);
- **текстовая карта CIDR** — простой файл, каждая строка которого содержит подсеть и код региона, разделённые пробелом; строки с `#` — комментарии; до 256 записей. Такая карта удобна для стендов и для собственных регионов (например, `EU` или код филиала):

```
### подсеть      регион
10.10.0.0/16     RU
192.0.2.0/24     EU
2001:db8::/32    NA
```

Если формат MMDB не поддерживается сборкой или файл не распознан как MMDB, Амибиг пытается загрузить его как текстовую карту CIDR.

Topology-record

Записи топологии сопоставляют регион клиента предпочтительному пулу wide-IP.

Параметр	Тип	По умолчанию	Описание
<code>name</code>	string	—	Имя записи (ключ)
<code>source-region</code>	string	—	Код региона источника запроса (например, <code>RU</code> , <code>EU</code>)
<code>preferred-pool</code>	string	—	Имя предпочтительного пула wide-IP
<code>weight</code>	uint16	<code>100</code>	Зарезервировано; при нескольких подходящих записях используется первая по порядку

Регион определяется по адресу **источника DNS-запроса**. Если клиенты используют центральный корпоративный DNS-сервер, слушатель определяет регион по адресу этого сервера, а не клиента. Учитывайте размещение DNS-серверов при настройке топологии.

Настройка через экран «GSLB»

Основные параметры GSLB настраиваются на одноимённом экране веб-интерфейса.

1. Перейдите в раздел **GSLB** и включите переключатель **Включено**.
2. В разделе **DNS-слушатель** задайте адрес и порт (по умолчанию `0.0.0.0:53`).
3. Нажмите **Добавить Wide-IP**: укажите имя, FQDN, метод балансировки, TTL и при необходимости резервный IP.
4. В разделе **Пулы** добавьте площадки: имя, VIP-адрес и вес.
5. Нажмите **Сохранить**, затем примените конфигурацию на экране **Применить конфигурацию**.

Гео-база и topology-record задаются в секции `gslb` конфигурации через REST API (`PUT /api/v1/config/gslb`).

Пример конфигурации

Пример включает DNS-слушатель и направляет запросы из региона `RU` в пул `msk`.

```
{
  "gslb": {
    "enabled": true,
    "dns-listener": { "address": "192.0.2.53", "port": 53 },
    "wide-ip": [
      {
        "name": "www",
        "fqdn": "www.example.com",
        "enabled": true,
        "lb-method": "topology",
        "ttl": 30,
        "pool": [
          { "name": "msk", "address": "192.0.2.100", "enabled": true, "ratio": 1 },
          { "name": "spb", "address": "198.51.100.100", "enabled": true, "ratio": 1 }
        ],
        "last-resort": "192.0.2.100"
      }
    ],
    "geo-database": "/config/geo/GeoLite2-Country.mmdb",
    "topology-record": [
      { "name": "ru-to-msk", "source-region": "RU", "preferred-pool": "msk" }
    ]
  }
}
```

Проверить работу можно обычными DNS-инструментами: `dig @192.0.2.53 www.example.com` должен вернуть один из адресов пулов (при `round-robin` повторные запросы перебирают площадки), а запрос неизвестного имени — `NXDOMAIN`.

Критерий гео в политиках

Та же гео-база используется политикой обработки трафика: критерии `source-geo` и `dest-geo` правила сопоставляют страну или регион клиента и сервера назначения (например, отдельное правило для трафика в определённые страны).

Конфигурация с гео-критериями без заданного `gslb.geo-database` не пройдёт проверку при применении. Подробнее: Критерии отбора.

Цепочки сервисов

Цепочка сервисов задаёт порядок прохождения трафика через внешние средства защиты. Правила политики выбирают цепочку, а dataplane последовательно передаёт поток каждому сервису и возвращает его в Амибиг для дальнейшей обработки.

Используйте этот раздел после настройки сетевых объектов и сервисов. Для диагностики выбранной цепочки применяйте раздел [Объяснение потока](#).

Обзор цепочек сервисов

В разделе описаны назначение цепочек сервисов, порядок прохождения трафика и действия Амибиг при отказе отдельного сервиса.

Назначение

Цепочка сервисов (Service Chain) — это упорядоченная последовательность сервисов, через которые проходит расшифрованный трафик. Цепочка определяет порядок инспекции: трафик последовательно проходит через каждый сервис в цепочке — от первого к последнему — после чего возвращается в Амибиг для повторного зашифрования.

Цепочки сервисов — ключевой механизм Service Function Chaining (SFC) в Амибиг. Они позволяют комбинировать разные средства безопасности (NGFW, DLP, антивирус) в единый конвейер обработки трафика.

В цепочке поддерживаются два типа сервисов: inline (прозрачный L2-участок) и ICAP. Сервис TAP-типа, который только зеркалирует трафик, не находясь на его пути, в текущей версии dataplane не реализован.

Архитектура

Архитектура цепочки определяет место внешних средств защиты в обработке трафика и последовательность передачи данных между ними.

Место цепочек в системе

Цепочка применяется после выбора правила политики и до повторного зашифрования трафика.

```
Политика (правило matching) --> Цепочка сервисов --> [NGFW --> DLP --> AV] --> Амибиг
(зашифрование) --> отправка
```

Прохождение трафика через цепочку

Схема показывает полный путь соединения от расшифрования до передачи трафика origin-серверу.

Схема взаимодействия

Источник	Связь	Назначение
Клиент	→ TLS	Амибиг расшифрование
Амибиг расшифрование	→	Сервис 1 Inline NGFW
Сервис 1 Inline NGFW	→	Сервис 2 ICAP DLP
Сервис 2 ICAP DLP	→	Амибиг зашифрование
Амибиг зашифрование	→ TLS	Сервер

Клиент --TLS--> Амибиг (расшифрование) --> 1. NGFW (inline) --> 2. DLP (ICAP) --> Амибиг (зашифрование) --TLS--> Сервер

Трафик проходит через сервисы строго в заданном порядке. Если один из сервисов заблокирует трафик (например, NGFW отбросит пакет или DLP вернет ICAP-блокировку), дальнейшее прохождение по цепочке прекращается.

Через inline-звенья цепочки проходит не только TCP-трафик: пакеты ICMP и UDP, попавшие под правило политики с этой цепочкой, также направляются через inline-сервисы — при включённых параметрах политики `udp-chain` и `icmp-chain` (по умолчанию включены; переключатели «UDP через сервисную цепочку» и «ICMP через сервисную цепочку» на экране [Топология](#)). При выключенном параметре датаграммы соответствующего протокола идут к origin-серверу в обход цепочки; трансляция адреса источника (`origin-SNAT`) при этом сохраняется. Подробнее о направлении трафика в inline-сервис см. раздел [Inline-сервисы](#).

Параметры цепочки

Для создания цепочки необходимо задать её имя и упорядоченный список сервисов.

Параметр	Тип	Обязательный	Описание
<code>name</code>	string	Да	Уникальное имя цепочки (например, <code>default</code> , <code>full-inspection</code> , <code>ngfw-only</code>)
<code>description</code>	string	Нет	Текстовое описание назначения цепочки

Параметр	Тип	Обязательный	Описание
<code>services</code>	ordered list	Да	Упорядоченный список имен сервисов. Порядок определяет последовательность прохождения

Упорядоченность списка сервисов

Список `services` имеет тип `ordered-by user` в YANG-модели. Расположите сервисы в требуемом порядке: трафик проходит через них последовательно, сверху вниз.

Типовые цепочки

Состав цепочки выбирают по необходимым видам инспекции и требованиям к порядку обработки трафика.

Только NGFW

Минимальная цепочка с одним inline-сервисом:

```
service-chains:
  chain:
    - name: ngfw-only
      description: "Только межсетевой экран"
      services:
        - ngfw
```

Трафик --> NGFW --> обратно

NGFW + DLP

Стандартная цепочка с последовательной inline- и ICAP-инспекцией:

```
service-chains:
  chain:
    - name: full-inspection
      description: "NGFW + DLP"
      services:
        - ngfw
        - dlp
```

Трафик --> NGFW (inline) --> DLP (ICAP) --> обратно

NGFW + IPS + DLP + Антивирус

Развернутая цепочка для максимальной инспекции:

```

service-chains:
  chain:
    - name: maximum-security
      description: "Полная инспекция всеми СЗИ"
      services:
        - ngfw
        - ips
        - dlp
        - antivirus

```

Трафик --> NGFW --> IPS --> DLP --> AV --> обратно

Несколько цепочек для разного трафика

Различные цепочки для разных категорий трафика:

```

service-chains:
  chain:
    - name: standard
      description: "Стандартный трафик — только NGFW"
      services:
        - ngfw
    - name: sensitive
      description: "Чувствительный трафик — NGFW + DLP"
      services:
        - ngfw
        - dlp
    - name: downloads
      description: "Загрузки — NGFW + антивирус"
      services:
        - ngfw
        - antivirus

```

Политики (Policy) определяют, какой трафик направляется в какую цепочку.

Связь с политиками

Цепочка сервисов назначается в правиле политики. Каждое правило определяет критерии отбора трафика и целевую цепочку:

- Правило 1: источник=10.0.0.0/8 --> цепочка "sensitive"
- Правило 2: категория=downloads --> цепочка "downloads"
- Правило 3: (по умолчанию) --> цепочка "standard"

Подробнее о политиках см. раздел Политики.

Поведение при отказе сервиса в цепочке

Если сервис в цепочке становится недоступным (все участники пула в DOWN), судьба клиентского соединения определяется параметром `service-down-action` этого сервиса:

service-down-action	Влияние на цепочку
<code>ignore</code> (по умолчанию)	Трафик пропускает этот сервис и переходит к следующему в цепочке
<code>bypass</code>	Устаревший синоним <code>ignore</code> ; поведение идентично
<code>drop</code>	Трафик отбрасывается, остальные сервисы цепочки не обрабатывают его
<code>reset</code>	Клиенту отправляется TCP RST, цепочка прерывается

Пример: цепочка [ngfw, dlp] при отказе NGFW

Пример сопоставляет результат обработки соединения для каждого значения `service-down-action`.

```

Сценарий 1: ngfw.service-down-action = ignore
Трафик --> (NGFW пропущен) --> DLP --> обратно
Результат: трафик проходит DLP без NGFW-инспекции

Сценарий 2: ngfw.service-down-action = drop
Трафик --> (отброшен)
Результат: весь трафик, направляемый в эту цепочку, блокируется

Сценарий 3: ngfw.service-down-action = reset
Трафик --> (TCP RST клиенту)
Результат: соединения клиентов разрываются

```

Фиксация отказа звена в журнале потоков

Если звено цепочки отказывает уже в процессе передачи трафика (например, `inline`-сервис перестаёт отвечать в середине соединения), соединение завершается, а в журнале потоков (`flow`-журнале) запись о нём получает причину закрытия `chain-fail`. По этой причине закрытия удобно отбирать соединения, пострадавшие от отказа звена, при разборе инцидентов.

Влияние на переключение отказоустойчивой пары

Помимо судьбы отдельного соединения (`service-down-action`), деградация сервиса влияет на решение о переключении пары высокой доступности. Для каждого сервиса задаются `min-healthy-members` (минимум работоспособных участников пула, по умолчанию 1) и `on-degrade` — вклад деградировавшего сервиса в оценку работоспособности узла: `drop` (по умолчанию) — сервис критичный, его деградация снижает оценку узла и может привести к переключению; `bypass` — сервис исключается из оценки. Эти параметры управляют только решением о переключении и не влияют на обработку соединений в цепочке. Подробнее: Режим Active/Standby.

Управление через веб-интерфейс

Веб-интерфейс позволяет создать цепочку, изменить порядок сервисов и проверить их текущее состояние.

Создание цепочки

Для создания цепочки укажите её имя и добавьте сервисы в требуемом порядке.

1. Перейдите в раздел **Цепочки сервисов**.
2. Нажмите **Создать цепочку**.
3. Укажите имя и описание.
4. Добавьте сервисы из списка доступных, расположив их в нужном порядке.
5. Нажмите **Сохранить**.

Изменение порядка сервисов

В веб-интерфейсе порядок сервисов в цепочке изменяется перетаскиванием. Для каждого сервиса указаны тип (inline или ICAP) и текущее состояние.

Визуализация цепочки

Веб-интерфейс отображает цепочку в виде последовательной диаграммы:

Цепочка full-inspection: NGFW (inline, UP) --> DLP (ICAP, UP) --> AV (ICAP, DOWN, ignore)

Для каждого сервиса в цепочке отображается:

Элемент	Описание
Имя сервиса	Уникальный идентификатор
Тип	inline или ICAP
Состояние	UP (все участники пула доступны) / DEGRADED (часть участников DOWN) / DOWN (все участники DOWN)
service-down-action	Действие при отказе (если статус DOWN)

Пример полной конфигурации

Комплексный пример: два сервиса, одна цепочка, привязка к политике.

```
### 1. Сервисы
services:
  service:
    - name: ngfw
      type: inline
      enabled: true
```

```

service-down-action: ignore
inline:
  to-service:
    vlan: vlan100
    gateway: "198.19.64.1"
  from-service:
    vlan: vlan200
    gateway: "198.19.64.129"
pool:
  member:
    - address: "198.19.64.7"
      port: 0
  load-balancing: session-hash
  monitor:
    type: tcp
    interval: 5
    timeout: 16

- name: dlp
  type: icap
  enabled: true
  service-down-action: ignore
  icap:
    request-mod:
      enabled: true
      uri: "icap://198.19.97.10/reqmod"
      preview-length: 1024
    response-mod:
      enabled: true
      uri: "icap://198.19.97.10/respmo"
      preview-length: 1024
  pool:
    member:
      - address: "198.19.97.10"
        port: 1344
    load-balancing: round-robin
    monitor:
      type: tcp
      interval: 5
      timeout: 16

### 2. Цепочка
service-chains:
  chain:
    - name: default
      description: "NGFW + DLP для всего трафика"
      services:
        - ngfw
        - dlp

### 3. Политика (пример)
policy:
  rules:
    - name: all-traffic
      service-chain: default

```

Ограничения

При проектировании цепочки учитывайте ограничения на её состав и порядок сервисов.

Ограничение	Описание
Один сервис может входить в несколько цепочек	Один и тот же сервис (например, <code>ngfw</code>) может использоваться в разных цепочках

Ограничение	Описание
Сервис не может дублироваться в одной цепочке	Каждый сервис может встречаться в цепочке только один раз
Пустая цепочка	Цепочка должна содержать хотя бы один сервис
Порядок фиксирован	После создания порядок изменяется только через редактирование цепочки

Диагностика

Диагностику начинают с проверки привязки цепочки к политике и доступности всех включённых в неё сервисов.

Типичные проблемы

В таблице приведены основные признаки ошибок конфигурации и способы их устранения.

Симптом	Возможная причина	Решение
Трафик не проходит инспекцию	Цепочка не привязана к политике	Проверить правила политик
Часть трафика обходит цепочку	Правило политики не покрывает весь трафик	Проверить критерии matching в политике
Сервис в цепочке пропускается	Все участники пула DOWN, <code>service-down-action: ignore</code>	Проверить доступность участников пула
Соединения обрываются	Сервис в цепочке с <code>service-down-action: reset</code>	Проверить состояние сервиса, изменить действие
Высокая задержка	Слишком много сервисов в цепочке	Оптимизировать цепочку, убрать избыточные сервисы

Политики

Политика определяет способ обработки соединений: расшифрование TLS-трафика, передача в сервисную цепочку, сквозной пропуск, блокирование или перенаправление на портал аутентификации. Правила сопоставляют трафик по SNI, IP-адресам, портам, категориям URL, пользователям и другим признакам.

Сначала настройте явное правило по умолчанию, затем добавьте обычные правила в порядке от более точных условий к более общим. При обработке соединения применяется первое совпавшее правило.

Обзор политик

Раздел описывает место политики в обработке трафика, порядок выбора правил и связь с другими объектами конфигурации.

Назначение

Политика (policy) — центральный механизм управления трафиком в Амибиг. Политика определяет, что происходит с каждым соединением, проходящим через систему: расшифровывать ли SSL/TLS, направлять ли трафик в сервисную цепочку, блокировать или пропустить без инспекции.

Политика состоит из упорядоченного набора правил. Каждое правило содержит критерии отбора трафика и действие, которое применяется при совпадении. Правила настраиваются на экране [Политики](#).

Политики

Порядок обработки
Правила проверяются сверху вниз. Срабатывает первое подходящее включённое правило. Если ни одно правило не подошло, применяется явное правило по умолчанию.

Имя политики: demo2003-policy

Правило по умолчанию
demo-default-bypass
No automatic inspection before the show starts
Изменить правило по умолчанию

[Прямой пропуск](#) [Без сервисной цепочки](#)

[+ Добавить](#)

#	Имя	Условия	Действие	Цепочка сервисов	Включено	
1	demo-site-0001-ngfw1-dlp	port 443 TCP SNI site-0001.bench.local	Перехват	demo-ngfw1-dlp	Нет	Выше
2	demo-site-0002-ngfw1	port 8443 TCP SNI site-0002.bench.local	Перехват	demo-ngfw1	Нет	Выше
3	demo-site-0003-ngfw2	port 12443 TCP SNI site-0003.bench.local	Перехват	demo-ngfw2	Нет	Выше
4	demo-site-0004-ngfw2-dlp	port 10443 TCP SNI site-0004.bench.local	Перехват	demo-ngfw2-dlp	Нет	Выше
5	demo-site-0005-ngfw1-ngfw2	port 11443 TCP SNI site-0005.bench.local	Перехват	demo-ngfw1-ngfw2	Нет	Выше
6	demo-ftp-ngfw1	port 990 TCP	Перехват	demo-ngfw1	Нет	Выше
7	demo-mail-smtps	port 465 TCP	Перехват	demo-ngfw1	Нет	Выше

[Сбросить](#) [Отмена](#) [Сохранить](#)

[↑ Применить конфигурацию](#)
[📄 Документация](#)
admin

Принцип работы

Политика последовательно проверяет обычные правила, а затем применяет явное правило по умолчанию.

Порядок обработки правил

Правила политики обрабатываются **сверху вниз** в порядке, заданном администратором. Срабатывает **первое совпавшее правило** — для каждого соединения выполняется действие первого включённого правила, критерии которого совпали с параметрами трафика. Остальные правила не проверяются.

Соединение --> Правило 1 (нет совпадения) --> Правило 2 (совпадение!) --> Действие
v
Остальные правила не проверяются

Если ни одно правило не совпало, соединение обрабатывается явным правилом по умолчанию (policy.default-rule). Это отдельный объект, который не участвует в сортировке обычных правил и всегда применяется последним.

На экране **Политики** это отражено напрямую: обычные правила находятся в переупорядочиваемой таблице, а правило по умолчанию показывается отдельной карточкой и редактируется вне списка.

Оценка критериев в рамках одного правила

Внутри одного правила все указанные критерии отбора объединяются логическим **И** (AND). Соединение должно удовлетворять всем указанным критериям одновременно, чтобы правило сработало. Подробнее: Критерии отбора трафика.

Действия (actions)

Каждому правилу назначается одно из следующих действий:

Действие	Описание
<code>intercept</code>	Расшифровать SSL/TLS (MITM) и направить расшифрованный трафик в сервисную цепочку для инспекции
<code>bypass</code>	Не расшифровывать TLS, но провести трафик через выбранную прозрачную сервисную цепочку
<code>passthrough</code>	Не расшифровывать TLS и не отправлять трафик в сервисную цепочку
<code>reject</code>	Отклонить соединение (TCP RST). Соединение разрывается на стороне клиента

Важно: полный путь расшифрования (`intercept`) поддерживается только для TCP/TLS. Для UDP и ICMP доступны действия `bypass`, `passthrough` и `reject`. Для остальных IP-протоколов (`other`) доступны только `passthrough` и `reject`.

Типичные сценарии использования действий

Выбор действия зависит от необходимости расшифрования TLS-трафика и передачи его в сервисную цепочку:

- **intercept** — основной режим работы. Используется для HTTPS-трафика, который необходимо расшифровать и проверить средствами защиты информации (NGFW, DLP, антивирус).
- **bypass** — для трафика, который нельзя расшифровывать, но который всё равно должен пройти через прозрачную inline-цепочку.
- **passthrough** — для трафика, который нельзя или не нужно расшифровывать и который не должен попадать ни в одну сервисную цепочку.
- **reject** — для блокировки нежелательных соединений на уровне политики.

Блокировка QUIC

Глобальный параметр `reject-quic` по умолчанию включён. При его включении Амибиг блокирует распознанные начальные пакеты QUIC. Это позволяет

клиентским приложениям, поддерживающим резервный протокол, перейти на HTTPS поверх TCP. Возможность такого перехода зависит от клиентского приложения.

Блокировка основана не на номере порта, а на DPI-классификации. Отбрасываются пакеты, распознанные как QUIC Initial версии 1 или 2, на любом UDP-порту. Остальной UDP-трафик, в том числе на порту 443, не блокируется этим параметром. Из перехваченных HTTP-ответов также удаляется заголовок `Alt-Svc` с предложением `h3`.

Переключатель «Блокировка QUIC» находится на экране «Топология», а ключ — в секции политики:

```
{
  "policy": {
    "reject-quit": true
  }
}
```

UDP и ICMP через сервисную цепочку

Параметры `udp-chain` и `icmp-chain` (по умолчанию оба: `true`) определяют, проходит ли транзитный UDP- и ICMP-трафик через inline-звенья сервисной цепочки. Соответствующие переключатели «UDP через сервисную цепочку» и «ICMP через сервисную цепочку» находятся на экране `Топология`.

При выключенном параметре пакеты соответствующего протокола направляются к серверу назначения в обход цепочки. Трансляция адреса источника (`origin-SNAT`) при этом сохраняется.

Отключение `icmp-chain` воспроизводит поведение F5 SSL Orchestrator, который не проводит ICMP через цепочку, и делает служебный трафик — определение MTU пути (PMTUD), трассировку маршрута — независимым от доступности цепочки.

```
{
  "policy": {
    "udp-chain": true,
    "icmp-chain": true
  }
}
```

Связь с другими компонентами

Политика использует ссылки на топологию, сервисные цепочки и дополнительные объекты обработки трафика.

Привязка к топологии

Политика явно привязывается к топологии через `topology.policy-ref`. Это позволяет видеть на экране `Топология`, какая именно политика обслуживает текущую точку входа трафика.

Сервисные цепочки

Правило политики может ссылаться на сервисную цепочку (`service-chain`). При срабатывании правила трафик направляется в указанную цепочку для обработки последовательностью сервисов (NGFW, DLP, ICAP и др.).

Это относится к `TCP/TLS` для `intercept` и к `UDP/ICMP` для прозрачного `bypass` через `inline`-цепочку. Для остальных IP-протоколов (`other`) сервисные цепочки не поддерживаются.

Трансляция адреса источника (SNAT)

Глобальные режимы трансляции адреса и порта источника для `origin`-соединений задаются в секции `egress` . Для отдельного правила их можно переопределить параметрами `origin-snat` и `origin-source-port` . Подробнее см. в разделах «Правила политик» и «Трансляция адреса источника (SNAT)».

Классы ограничения скорости

К правилу можно привязать класс ограничения скорости (`rate-class`). В текущей версии эта связь только проверяется в модели конфигурации; `dataplane` не ограничивает полосу. Подробнее см. в разделе «Классы ограничения скорости».

aRules

К правилу политики можно привязать одно или несколько `aRules` — Tcl-скриптов для тонкой настройки обработки HTTP-запросов (модификация заголовков, управление ICAP-адаптацией и др.). Подробнее см. раздел `aRules`.

Пример конфигурации политики

Пример задаёт два обычных правила, явное правило по умолчанию и привязку политики к топологии:

```
{
  "policy": {
    "name": "main-policy",
    "reject-quit": true,
    "default-rule": {
      "name": "default-intercept",
      "description": "Расшифрование всего остального трафика",
      "action": "intercept",
      "service-chain": "basic-chain"
    },
  },
  "rule": [
    {
      "name": "bypass-banking",
      "description": "Пропуск банковского трафика без расшифрования",
      "enabled": true,
      "match": {
        "sni": ["*.sberbank.ru", "*.tinkoff.ru", "*.vtb.ru"]
      },
      "action": "passthrough"
    },
  ],
}
```

```

    "name": "intercept-internal",
    "description": "Расшифрование и инспекция внутреннего трафика",
    "enabled": true,
    "match": {
      "source-ip": ["10.0.0.0/8", "172.16.0.0/12"]
    },
    "action": "intercept",
    "service-chain": "full-inspection"
  }
]
},
"topology": {
  "policy-ref": "main-policy"
}
}

```

В примере выше правила дают следующий результат:

1. Трафик к банковским сервисам пропускается без расшифрования и без сервисной цепочки.
2. Трафик из внутренних сетей расшифровывается и проходит через полную цепочку инспекции.
3. Весь остальной трафик обрабатывается через явное `default-rule`.

Правила политик

Раздел описывает структуру правил, доступные действия и порядок применения подготовленной конфигурации политики.

Структура правила

Обычное правило содержит условия сопоставления и действие, которое Амибиг выполняет при совпадении. Правила проверяются сверху вниз; применяется первое совпавшее включённое правило.

Амибиг

Одиночный узел

Обзор

Трафик

Система

Сеть

SSL/TLS

Топология

Инспекция

Мониторы

Пулы

Виртуальные серверы

Сервисы

Цепочки сервисов

Политики

Идентификация

Категории URL

aRules

Шейпинг

GSLB

Мониторинг

Управление

Применить конфигурацию

Документация

admin

Политики

Порядок обработки

Правила проверяются сверху вниз. Срабатывает первое подходящее включённое правило. Если ни одно правило не подошло, применяется явное правило по умолчанию.

Имя политики: demo2003-policy

Правило по умолчанию

demo-default-bypass

No automatic inspection before the show starts

Изменить правило по умолчанию

Прямой пропуск

Без сервисной цепочки

+ Добавить

#	Имя	Условия	Действие	Цепочка сервисов	Включено	
1	demo-site-0001-ngfw1-dlp	port 443 TCP SNI site-0001.bench.local	Перехват	demo-ngfw1-dlp	Нет	Выше
2	demo-site-0002-ngfw1	port 8443 TCP SNI site-0002.bench.local	Перехват	demo-ngfw1	Нет	Выше
3	demo-site-0003-ngfw2	port 12443 TCP SNI site-0003.bench.local	Перехват	demo-ngfw2	Нет	Выше
4	demo-site-0004-ngfw2-dlp	port 10443 TCP SNI site-0004.bench.local	Перехват	demo-ngfw2-dlp	Нет	Выше
5	demo-site-0005-ngfw1-ngfw2	port 11443 TCP SNI site-0005.bench.local	Перехват	demo-ngfw1-ngfw2	Нет	Выше
6	demo-ftp-ngfw1	port 990 TCP	Перехват	demo-ngfw1	Нет	Выше
7	demo-mail-smtps	port 465 TCP	Перехват	demo-ngfw1	Нет	Выше

Сбросить

Отмена

Сохранить

Параметр	Тип	Обязательный	Описание
name	string	да	Уникальное имя правила (ключ)
description	string	нет	Текстовое описание назначения правила
enabled	boolean	нет	Активность правила (true по умолчанию)
match	container	да	Критерии отбора трафика для обычного правила. Правило «на весь остальной трафик» оформляется через policy.default-rule
action	enum	да	Действие: intercept , bypass , passthrough , reject , portal-redirect

183 / 451

Параметр	Тип	Обязательный	Описание
<code>service-chain</code>	string	нет	Имя сервисной цепочки для обработки трафика
<code>rate-class</code>	string	нет	Ссылка на класс ограничения скорости. Поле сохраняется в конфигурации, но текущий dataplane не применяет ограничение
<code>origin-snat</code>	enum	нет	Переопределение трансляции адреса источника: <code>automap</code> , <code>none</code> , <code>pool</code> . Если поле отсутствует, используется значение секции <code>egress</code>
<code>origin-source-port</code>	enum	нет	Переопределение порта источника: <code>preserve</code> , <code>preserve-strict</code> , <code>change</code> . Если поле отсутствует, используется значение секции <code>egress</code>
<code>insert-identity-headers</code>	boolean	нет	Добавлять в запрос, уходящий в сервисную цепочку, заголовки с именем опознанного пользователя (<code>false</code> по умолчанию)
<code>identity-header-format</code>	enum	нет	Формат имени в заголовке: <code>username</code> (по умолчанию) или <code>ldap-uri</code>

Параметр	Тип	Обязательный	Описание
<code>identity-header-encoding</code>	enum	нет	Кодирование значения заголовка: <code>plain</code> (по умолчанию) или <code>base64</code>
<code>arules</code>	list[string]	нет	Список имён aRules для выполнения при срабатывании правила

Создание правила

Создайте правило в подготовленной конфигурации, проверьте его положение в списке и примените конфигурацию. Следующие примеры показывают обязательные условия и дополнительные параметры.

Минимальное правило

Обычное правило должно иметь имя, действие и хотя бы один критерий отбора. Явное правило по умолчанию настраивается отдельно через `policy.default-rule`.

```
{
  "name": "inspect-web",
  "match": {
    "dest-port": [443],
    "protocol": "tcp"
  },
  "action": "intercept",
  "service-chain": "main-chain"
}
```

Правило с критериями отбора

В одном правиле можно объединить несколько критериев. В следующем примере должны одновременно совпасть SNI и адрес источника:

```
{
  "name": "bypass-finance",
  "description": "Пропуск трафика к финансовым сервисам",
  "enabled": true,
  "match": {
    "sni": ["*.bank.example.com", "*.finance.example.com"],
    "source-ip": ["10.10.0.0/16"]
  },
  "action": "passthrough"
}
```

Правило с полным набором параметров

В расширенном правиле можно назначить сервисную цепочку и правила aRules:

```
{
  "name": "intercept-web",
  "description": "Инспекция веб-трафика из внутренней сети",
  "enabled": true,
  "match": {
    "source-ip": ["192.168.0.0/16"],
    "dest-port": [443, 8443],
    "protocol": "tcp"
  },
  "action": "intercept",
  "service-chain": "ngfw-dlp-chain",
  "rules": ["inject_auth", "log_requests"]
}
```

Действия (action)

Выберите действие с учётом протокола и требуемого пути трафика. Действия `intercept` и `bypass` используют сервисную цепочку, а `passthrough` и `reject` не допускают её указания.

intercept

Расшифровывает TLS-трафик методом MITM и направляет открытый трафик в сервисную цепочку. Действие поддерживается для TCP/TLS.

До применения правила настройте профиль SSL/TLS и сертификат УЦ для выпуска подменных сертификатов. Укажите `service-chain`: без сервисной цепочки dataplane выполняет сквозной пропуск без расшифрования.

```
{
  "name": "inspect-all-https",
  "match": { "dest-port": [443], "protocol": "tcp" },
  "action": "intercept",
  "service-chain": "full-inspection"
}
```

bypass

Передаёт трафик без расшифрования через прозрачную inline-сервисную цепочку.

Поле `service-chain` обязательно. Цепочка должна содержать хотя бы один прозрачный inline-сервис и поддерживать протокол правила.

```
{
  "name": "bypass-pinned",
  "match": {
    "sni": ["*.apple.com", "*.googleapis.com"]
  },
  "action": "bypass",
  "service-chain": "inline-ngfw"
}
```

passthrough

Передаёт трафик напрямую к назначению без расшифрования и без сервисной цепочки.

Поле `service-chain` указывать нельзя. Действие применяется, например, к приложениям с `certificate pinning` и другим исключениям из инспекции.

```
{
  "name": "passthrough-banking",
  "match": {
    "sni": ["*.bank.example.com"]
  },
  "action": "passthrough"
}
```

reject

Прекращает обработку совпавшего трафика. TCP-соединение закрывается, а дейтаграмма отбрасывается. Поле `service-chain` указывать нельзя.

```
{
  "name": "block-forbidden",
  "match": {
    "sni": ["*.gambling.example.com"]
  },
  "action": "reject"
}
```

portal-redirect

Перенаправляет неопознанного пользователя на портал аутентификации. До применения правила включите портал в секции `identity.portal`; конфигурация с выключенным порталом будет отклонена.

```
{
  "name": "authenticate-unknown-users",
  "match": { "source-user-class": "unknown", "protocol": "tcp" },
  "action": "portal-redirect"
}
```

После совпадения HTTP-запрос получает ответ 302 с адресом настроенного портала.

Назначение сервисной цепочки

Параметр `service-chain` указывает имя существующей сервисной цепочки. При совпадении правила трафик проходит через сервисы в порядке, заданном в цепочке.

```
{
  "name": "full-inspect",
```

```

"match": { "dest-port": [443], "protocol": "tcp" },
"action": "intercept",
"service-chain": "ngfw-chain"
}

```

Если `service-chain` не указан для действия `intercept`, dataplane выполняет сквозной пропуск без расшифрования. Для действия `bypass` поле обязательно, иначе конфигурация не пройдет проверку.

Переопределение SNAT для правила

Для отдельного правила можно переопределить трансляцию адреса и выбор порта источника в origin-соединении. Если параметры `origin-snat` и `origin-source-port` не заданы, используются общие значения секции `egress`.

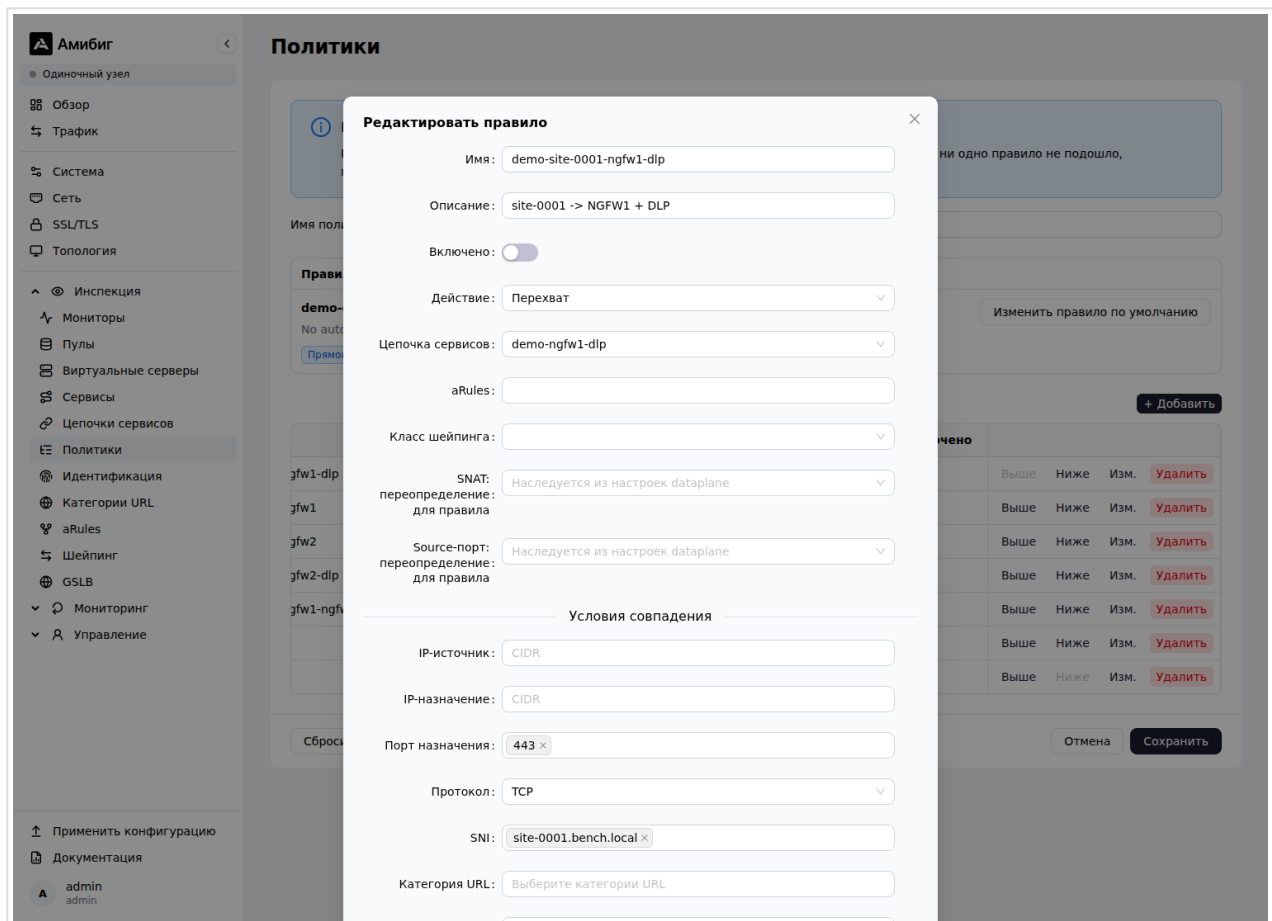
Параметр	Значения	Поведение при отсутствии
<code>origin-snat</code>	<code>automap</code> , <code>none</code> , <code>pool</code>	Наследуется глобальный <code>egress.origin-snat</code>
<code>origin-source-port</code>	<code>preserve</code> , <code>preserve-strict</code> , <code>change</code>	Наследуется глобальный <code>egress.origin-source-port</code>

Значения совпадают с общими режимами. Подробное описание и требования к маршрутизации приведены в разделе Трансляция адреса источника (SNAT).

Учитывайте следующие ограничения:

- `origin-snat: none` сохраняет адрес клиента. Устройство должно находиться на пути ответного трафика от сервера назначения.
- `origin-snat: pool` использует глобальный список `snat-pool` секции `egress`.

В веб-интерфейсе откройте форму правила на экране `Политики` и выберите значения в полях «SNAT: переопределение для правила» и «Порт источника: переопределение для правила». Значение «Наследуется из настроек dataplane» сохраняет общие параметры.



В F5 SSL Orchestrator режим SNAT задаётся на всю топологию инспекции; Амибиг дополнительно позволяет переопределить оба режима для отдельного правила политики.

```
{
  "name": "preserve-client-ip",
  "description": "Сохранение адреса клиента для серверов в защищаемом сегменте",
  "match": {
    "dest-ip": ["203.0.113.0/24"]
  },
  "action": "intercept",
  "service-chain": "full-inspection",
  "origin-snat": "none",
  "origin-source-port": "preserve-strict"
}
```

Передача имени пользователя в цепочку

Если подсистема идентификации определила пользователя, Амибиг может добавить его имя в HTTP-заголовки запроса к сервисной цепочке. Межсетевой экран или DLP может использовать эти сведения в собственной политике. Передача настраивается отдельно для каждого правила Амибиг.

Параметр	Значения	По умолчанию	Описание
<code>insert-identity-headers</code>	boolean	<code>false</code>	Включить добавление заголовков с именем пользователя
<code>identity-header-format</code>	<code>username</code> , <code>ldap-uri</code>	<code>username</code>	Формат имени: <code>username</code> — исходное имя; <code>ldap-uri</code> — строка вида <code>LDAP://domain/user</code>
<code>identity-header-encoding</code>	<code>plain</code> , <code>base64</code>	<code>plain</code>	Кодирование значения: <code>plain</code> — текстом как есть; <code>base64</code> — когда имя может содержать символы, недопустимые в заголовке HTTP (национальные алфавиты, пробелы)

Для неопознанного пользователя заголовки не добавляются. Одноимённые заголовки из клиентского запроса удаляются. Заголовки, добавленные устройством, также удаляются после выхода из сервисной цепочки и не передаются серверу назначения.

В веб-интерфейсе откройте форму правила на экране **Политики**. В блоке «Передача имени пользователя в цепочку» включите параметр «Добавлять имя пользователя в заголовки», затем выберите формат имени и способ кодирования.

```
{
  "name": "intercept-with-identity",
  "match": {
    "source-user-class": "known-user"
  },
  "action": "intercept",
  "service-chain": "ngfw-dlp-chain",
  "insert-identity-headers": true,
  "identity-header-format": "username",
  "identity-header-encoding": "base64"
}
```

Назначение класса ограничения скорости

Поле `rate-class` предусмотрено моделью конфигурации и может ссылаться на существующий класс ограничения скорости. В текущей версии dataplane не применяет это поле и не ограничивает полосу пропускания. Не используйте `rate-class` как средство обеспечения QoS.

Следующий фрагмент показывает только формат конфигурации. Его применение не изменит скорость трафика:

```
{
  "name": "rate-limited-browsing",
  "match": {
    "url-category": ["streaming", "social-media"]
  },
  "action": "intercept",
  "service-chain": "basic-chain",
  "rate-class": "low-priority"
}
```

Привязка aRules

К правилу политики можно привязать одно или несколько правил aRules. Они выполняются при обработке HTTP-трафика, соответствующего правилу политики.

```
{
  "name": "intercept-with-headers",
  "match": { "dest-port": [443], "protocol": "tcp" },
  "action": "intercept",
  "service-chain": "dlp-chain",
  "arules": ["inject_auth", "size_limit"]
}
```

aRules из правила политики объединяются с aRules из сервисов цепочки. Порядок выполнения:

1. aRules из сервисов цепочки (в порядке расположения сервисов).
2. aRules из правила политики.

Повторяющиеся имена исключаются: если одно правило aRules привязано и к сервису, и к правилу политики, оно выполняется один раз.

Управление порядком правил

Правила обрабатываются строго в порядке расположения в списке (`ordered-by user`). Перед применением конфигурации убедитесь, что частные исключения находятся выше общих правил.

Рекомендации по порядку

Расположите правила в следующем порядке:

1. **Точные правила** — конкретные SNI, отдельные адреса и узкие подсети.
2. **Общие правила** — категории URL и широкие подсети.
3. **Правило по умолчанию** — отдельный объект `policy.default-rule`, который проверяется после обычных правил.

Пример: порядок правил

В примере сначала исключается банковский трафик, затем блокируется отдельная категория, после чего применяется общее правило для внутренней сети:

```
{
  "policy": {
    "name": "main-policy",
    "default-rule": {
      "name": "default-intercept",
      "action": "intercept",
      "service-chain": "basic"
    },
    "rule": [
      { "name": "bypass-banking", "match": { "sni": ["*.bank.ru"] }, "action": "passthrou
gh" },
      { "name": "block-gambling", "match": { "url-category": ["gambling"] }, "action":
"reject" },
      { "name": "inspect-internal", "match": { "source-ip": ["10.0.0.0/8"] }, "action": "intercept
", "service-chain": "full" }
    ]
  }
}
```

Явное правило по умолчанию

Объект `policy.default-rule` определяет действие для трафика, который не совпал ни с одним обычным правилом. Он:

- всегда применяется после всех обычных правил;
- не имеет блока `match`;
- не перемещается внутри упорядоченного списка правил;
- отображается в веб-интерфейсе и в инструменте `Объяснение потока`.

```
{
  "name": "default-intercept",
  "description": "Инспекция остального трафика",
  "action": "intercept",
  "service-chain": "basic"
}
```

Перестановка правил

На экране `Политики` переместите обычное правило вверх или вниз. После изменения порядка примените конфигурацию и проверьте ожидаемый поток инструментом `Объяснение потока`.

`policy.default-rule` в этот список не входит и не перемещается вместе с обычными правилами.

Включение и отключение правил

Параметр `enabled` временно исключает правило из обработки без удаления. После применения конфигурации отключённое правило (`"enabled": false`) пропускается, и трафик проверяется по следующему правилу.

```
{
  "name": "test-rule",
  "enabled": false,
  "match": { "sni": ["*.test.example.com"] },
  "action": "intercept",
  "service-chain": "test-chain"
}
```

Критерии отбора трафика

Раздел описывает доступные критерии политики и правила их сочетания в одном условии.

Общие принципы

Критерии отбора определяют, какой трафик соответствует правилу политики. Все критерии внутри одного правила объединяются логическим **И** (AND) — соединение должно удовлетворять каждому указанному критерию одновременно.

Если критерий задан как список значений, внутри списка значения объединяются логическим **ИЛИ** (OR) — достаточно совпадения хотя бы с одним значением.

Критерий `protocol` — единственное значение, не список.

```
Правило срабатывает, если:
(source-ip[0] OR source-ip[1] OR ...)
AND (dest-ip[0] OR dest-ip[1] OR ...)
AND (dest-port[0] OR dest-port[1] OR ...)
AND (sni[0] OR sni[1] OR ...)
AND (url-category[0] OR url-category[1] OR ...)
AND (source-geo[0] OR source-geo[1] OR ...)
AND (dest-geo[0] OR dest-geo[1] OR ...)
AND (source-user[0] OR source-user[1] OR ...)
AND (source-user-group[0] OR source-user-group[1] OR ...)
AND source-user-class
AND protocol
```

Обычное правило обязано иметь хотя бы один критерий отбора. Правило «на весь остальной трафик» без критериев оформляется через явное правило по умолчанию (`policy.default-rule`) — конфигурация с обычным правилом без критериев не пройдёт проверку при применении.

Критерии

Для каждого критерия ниже указаны формат значения, логика сопоставления и пример конфигурации.

source-ip — IP-адрес источника

Совпадение по IP-адресу клиента. Значения задаются в формате CIDR.

Параметр	Описание
Тип	список, ip-prefix
Формат	CIDR-нотация (например, 10.0.0.0/8 , 192.168.1.100/32)
Логика	OR между значениями

```
{
  "match": {
    "source-ip": ["10.0.0.0/8", "172.16.0.0/12", "192.168.0.0/16"]
  }
}
```

Одиночный хост задаётся с префиксом /32 :

```
{
  "match": {
    "source-ip": ["192.168.1.100/32"]
  }
}
```

dest-ip — IP-адрес назначения

Совпадение по IP-адресу сервера назначения. Формат аналогичен source-ip .

Параметр	Описание
Тип	список, ip-prefix
Формат	CIDR-нотация
Логика	OR между значениями

```
{
  "match": {
    "dest-ip": ["203.0.113.0/24", "198.51.100.0/24"]
  }
}
```

dest-port — порт назначения

Совпадение по порту назначения TCP/UDP-соединения.

Параметр	Описание
Тип	список, port-number
Формат	Целое число (1-65535)
Логика	OR между значениями

```
{
  "match": {
    "dest-port": [443, 8443, 9443]
  }
}
```

Для задания диапазона портов перечислите все порты диапазона:

```
{
  "match": {
    "dest-port": [443, 8443, 8444, 8445]
  }
}
```

protocol — протокол транспортного уровня

Совпадение по протоколу. В отличие от остальных критериев, `protocol` принимает **одно значение**, а не список.

Параметр	Описание
Тип	enumeration (единственное значение)
Значения	<code>tcp</code> , <code>udp</code> , <code>icmp</code> , <code>other</code> , <code>any</code>

```
{
  "match": {
    "protocol": "tcp"
  }
}
```

Значение `any` соответствует отсутствию явного протокольного отбора: правило применяется к любому протоколу, а допустимость действия проверяется по фактическому семейству протокола соединения.

Важно: поддержка действий зависит от протокола:

- `tcp` поддерживает полный путь расшифрования (`intercept`) и сервисные цепочки;
- `udp` и `icmp` поддерживают `bypass`, `passthrough` и `reject`;
- `other` поддерживает только `passthrough` и `reject`.

sni — Server Name Indication

Совпадение по значению TLS SNI (Server Name Indication) — имени сервера, передаваемому клиентом в TLS ClientHello. Поддерживаются подстановочные шаблоны.

Параметр	Описание
Тип	список, string
Формат	Доменное имя или шаблон с *
Логика	OR между значениями

Синтаксис шаблонов SNI

Таблица показывает результат сопоставления точных имён и шаблонов SNI:

Шаблон	Совпадает с	Не совпадает с
example.com	example.com	www.example.com , sub.example.com
*.example.com	example.com , www.example.com , a.b.example.com	example.net

Подстановочный знак поддерживается только в префиксе *. . Шаблоны вида example.* и *.*.example.com не используйте.

```
{
  "match": {
    "sni": ["*.sberbank.ru", "*.tinkoff.ru", "online.vtb.ru"]
  }
}
```

Примечание: SNI доступен только для TLS-соединений. Для не-TLS трафика критерий SNI не применяется.

url-category — категория URL

Совпадение по категории URL. Категории — управляемые объекты конфигурации: они создаются на экране Категории URL и содержат списки доменных шаблонов (точное имя или шаблон *.example.com). Правило ссылается на категорию по имени; ссылка на несуществующую категорию не пройдёт проверку при применении.

Параметр	Описание
Тип	список, string
Формат	Имя категории, определённой на экране Категории URL
Логика	OR между значениями

```
{
  "match": {
    "url-category": ["banking", "healthcare", "government"]
  }
}
```

source-geo и dest-geo — географический регион

Совпадение по стране или региону клиента (`source-geo`) и сервера назначения (`dest-geo`). Значения — коды стран или регионов заглавными буквами (например, RU , US , EMEA).

Параметр	Описание
Тип	список, string
Формат	Код страны/региона заглавными буквами (2–8 символов)
Логика	OR между значениями

```
{
  "match": {
    "source-geo": ["RU"],
    "dest-geo": ["US", "EMEA"]
  }
}
```

Для работы географического отбора в конфигурации должен быть указан файл базы GeoIP (`gs1b.geo-database` , формат MaxMind MMDb). Конфигурация с гео-критериями без указанной базы не пройдёт проверку при применении.

source-user — имя пользователя

Критерий совпадает с именем идентифицированного пользователя. Подсистема идентификации получает соответствия из syslog, PAN User-ID, REST API или портала аутентификации и определяет имя по адресу источника. LDAP используется как каталог пользователей и групп, но не как источник соответствий IP-адресов. В форме правила на экране «Политики» критерий задаётся полем «Пользователь».

Параметр	Описание
Тип	список, string
Формат	Точное имя пользователя (например, <code>ivanov</code> , <code>DOMAIN\petrov</code>)
Логика	OR между значениями

```
{
  "match": {
    "source-user": ["ivanov", "DOMAIN\petrov"]
  }
}
```

Если для адреса источника пользователь не определён, критерий не совпадает.

source-user-group — группа пользователей

Критерий проверяет членство идентифицированного пользователя в группе LDAP-каталога. Имя пользователя определяется по адресу источника. В форме правила на экране «Политики» критерий задаётся полем «Группа пользователей».

Параметр	Описание
Тип	список, string
Формат	Имя группы (например, <code>Domain Users</code> , <code>Finance</code>)
Логика	OR между значениями

```
{
  "match": {
    "source-user-group": ["Finance", "Domain Admins"]
  }
}
```

source-user-class — состояние опознания

Совпадение по самому факту опознания пользователя, без перечисления имён. В форме правила на экране `Политики` критерий задаётся полем «Состояние опознания».

Параметр	Описание
Тип	enumeration (единственное значение)
Значения	<code>known-user</code> — для адреса источника известен пользователь; <code>unknown</code> — пользователь не определён

```
{
  "match": {
    "source-user-class": "unknown"
  }
}
```

Типичное применение — отдельное правило для неопознанного трафика: например, направить его на портал входа или назначить более строгую проверку. Критерий вынесен в отдельное поле намеренно: внутри `source-user` «магических» значений нет.

Комбинирование критериев

Следующие примеры показывают правила с несколькими критериями и разной шириной охвата трафика.

Пример: узкое правило

Правило срабатывает только для TCP-трафика из подсети `10.10.0.0/16` на порт 443 к доменам `*.internal.corp`:

```
{
  "name": "inspect-internal-https",
  "match": {
    "source-ip": ["10.10.0.0/16"],
    "dest-port": [443],
    "protocol": "tcp",
    "sni": ["*.internal.corp"]
  },
  "action": "intercept",
  "service-chain": "internal-chain"
}
```

Пример: широкое правило passthrough

Пропуск всего трафика к банковским и медицинским сервисам по категориям URL:

```
{
  "name": "passthrough-sensitive",
  "match": {
    "url-category": ["banking", "healthcare"]
  },
  "action": "passthrough"
}
```

Пример: блокировка по IP назначения

Блокировка соединений к известным вредоносным адресам:

```
{
  "name": "block-malicious-ips",
  "match": {
    "dest-ip": ["198.51.100.0/24", "203.0.113.50/32"]
  },
}
```

```
"action": "reject"
}
```

Порядок приоритета критериев

Система не назначает приоритетов отдельным критериям. Приоритет определяется исключительно позицией правила в списке (срабатывает первое совпавшее правило). Для корректной работы размещайте правила с более специфичными критериями выше правил с более широкими критериями.

Правильно:

- | | | |
|-------------------|-------------------------|---------------------------|
| 1. bypass-banking | (SNI: *.bank.ru) | <-- специфичный SNI |
| 2. inspect-office | (source-ip: 10.0.0.0/8) | <-- широкая подсеть |
| 3. default-rule | (правило по умолчанию) | <-- весь остальной трафик |

Неправильно:

- | | | |
|--------------------|--------------------------|--|
| 1. Широкое правило | (`source-ip: 0.0.0.0/0`) | совпадает первым; остальные правила не проверяются |
| 2. bypass-banking | (SNI: *.bank.ru) | |
| 3. inspect-office | (source-ip: 10.0.0.0/8) | |

Категории URL

Раздел описывает создание доменных категорий и их применение в критериях политики.

Назначение

Категория URL — это именованный список доменных шаблонов, например для банковских, государственных или социальных ресурсов. Правила политики ссылаются на категорию по имени, поэтому список доменов можно изменять без редактирования каждого правила. Категории обычно применяются для исключения отдельных ресурсов из расшифрования TLS-трафика или для их блокирования.

Сопоставление выполняется по имени сервера из SNI во время клиентского TLS handshake. Если клиент не передал SNI, критерий категории URL не совпадает.

Создание категории на экране «Категории URL»

Создайте категорию в подготовленной конфигурации и затем примените изменения:

1. Перейдите в раздел **Категории URL**.
2. Нажмите **+** **Добавить**. В форме **Добавить категорию URL** укажите имя (например, `banking`) и описание.

3. В поле **URL-шаблоны** введите домены. Добавляйте каждый шаблон отдельным элементом.
4. Нажмите **ОК**, чтобы добавить категорию в таблицу.
5. Нажмите **Сохранить** в разделе, затем примените конфигурацию на экране **Применить конфигурацию**.

После применения конфигурации категория становится доступна в условиях правил. Чтобы временно исключить категорию из сопоставления, выключите параметр **Включено**. Ссылки из правил при этом сохраняются.

Параметр	Тип	По умолчанию	Описание
name	string	—	Уникальное имя категории (ключ). На него ссылается критерий url-category правила
description	string	—	Текстовое описание
enabled	boolean	true	Выключенная категория никогда не совпадает
entry	list[string]	—	Список доменных шаблонов

Форматы шаблонов доменов

Шаблон описывает имя хоста. Допустимы две формы:

Форма	Пример	С чем совпадает
Точное имя	example.com	Только example.com
Wildcard	*.example.com	Сам example.com и любой его поддомен на любой глубине: www.example.com , a.b.example.com

Правила записи:

- шаблон можно ввести в виде URL, например `https://*.example.com/path` ; при сохранении остаётся только имя хоста, а схема, порт, путь и параметры отбрасываются;
- звёздочка допустима только в начале шаблона и только в виде префикса `*` ; шаблоны вида `example.*` или `www.*.com` отклоняются;
- регистр не учитывается;
- записи, дублирующие уже добавленный хост, отбрасываются при сохранении;

- запись, не являющаяся корректным доменом, не пройдет проверку при сохранении.

Лимиты

Dataplane обрабатывает ограниченное число категорий и ссылок. Не превышайте следующие значения:

Ограничение	Значение
Категорий в конфигурации	64
Шаблонов в одной категории	16
Категорий в критерии одного правила	8

При превышении лимита dataplane использует только первые 64 категории, первые 16 шаблонов в категории и первые 8 ссылок в правиле. Имена категорий должны быть уникальны; конфигурация с повторяющимся именем не пройдет проверку.

Использование в правилах политики

Правило ссылается на категории списком `match.url-category`. Между значениями действует логика «ИЛИ»: достаточно совпадения с одной из перечисленных категорий.

```
{
  "name": "bypass-banking",
  "match": {
    "url-category": ["banking", "healthcare"]
  },
  "action": "bypass",
  "service-chain": "transparent-chain"
}
```

Критерий срабатывает только при наличии SNI в клиентском TLS handshake. Ссылка на несуществующую категорию не пройдет проверку, а веб-интерфейс не позволит удалить категорию, используемую правилом. Правила сочетания условий приведены в разделах Критерии отбора и Правила политик.

Приоритет при пересечении категорий

Категории не имеют собственного приоритета. Если домен входит в несколько категорий, применяется первое сверху правило с совпавшими условиями. Поэтому точное правило, например `reject` для отдельной категории, размещайте выше общего. Внутри одного правила категории объединяются по логике «ИЛИ».

Классы ограничения скорости

Раздел описывает модель классов ограничения скорости и её ограничения в текущей версии Амибиг.

Назначение

Класс ограничения скорости (rate-shaping class) хранит параметры полосы пропускания. Класс создаётся как отдельный объект и связывается с правилом политики через `rate-class`. Один класс можно использовать в нескольких правилах.

Важно. В текущей версии dataplane не применяет параметры класса к трафику. Классы проходят проверку, сохраняются в конфигурации и привязываются к правилам, но не гарантируют и не ограничивают скорость.

Параметры класса

Для каждого класса обязательны имя и параметр `rate`. Остальные поля можно не задавать.

Параметр	Тип	По умолчанию	Описание
<code>name</code>	string	—	Уникальное имя класса. На него ссылается <code>rate-class</code> правила
<code>description</code>	string	—	Описание класса
<code>rate</code>	string	—	Заданная базовая скорость (CIR). Примеры: <code>500kbps</code> , <code>10mbps</code> , <code>1gbps</code>
<code>ceiling</code>	string	равен <code>rate</code>	Заданная максимальная скорость (PIR)
<code>max-burst</code>	string	—	Заданный максимальный объём кратковременного превышения <code>rate</code> . Примеры: <code>64kb</code> , <code>2mb</code>

Параметр	Тип	По умолчанию	Описание
<code>direction</code>	enum	<code>both</code>	Направление трафика
<code>drop-policy</code>	enum	<code>tail</code>	Политика сброса пакетов при переполнении очереди
<code>queue</code>	enum	<code>pfifo</code>	Дисциплина обслуживания очереди

Скорости записываются числом с суффиксом `kbps` , `mbps` или `gbps` ; объём кратковременного превышения — с суффиксом `kb` или `mb` . В веб-интерфейсе значения отображаются, например, как «10 Мбит/с».

Направление (`direction`)

Параметр задаёт направление, к которому относится класс:

Значение	Описание
<code>to-client</code>	В сторону клиента
<code>to-server</code>	В сторону сервера назначения
<code>both</code>	В обоих направлениях

Политика сброса (`drop-policy`)

Параметр задаёт порядок сброса пакетов при переполнении очереди:

Значение	Описание
<code>tail</code>	Tail drop: при заполнении очереди отбрасываются новые пакеты
<code>red</code>	Random Early Detection: пакеты отбрасываются с заданной вероятностью до полного заполнения очереди

Дисциплина очереди (`queue`)

Параметр задаёт алгоритм обслуживания очереди:

Значение	Описание
pfifo	Пакеты обслуживаются в порядке поступления
sfq	Stochastic Fairness Queueing: полоса распределяется между потоками

Создание класса

Класс создаётся на экране «Шейпинг трафика». После сохранения изменений нужно применить подготовленную (candidate) конфигурацию.

1. Перейдите на экран **Шейпинг трафика**.
2. Нажмите **+ Добавить** и укажите имя в форме **Добавить класс**.
3. Задайте **Скорость (CIR)**, например 10mbps . При необходимости укажите **Потолок (PIR)** и **Макс. всплеск**.
4. Выберите **Направление**, **Политику сброса** и **Очередь**.
5. Нажмите **ОК**, чтобы добавить класс в таблицу.
6. Нажмите **Сохранить** в разделе, затем примените конфигурацию на экране **Применить конфигурацию**.

Привязка к правилу политики

Класс привязывается к обычному правилу или к правилу по умолчанию (policy.default-rule) параметром rate-class . Сейчас эта ссылка влияет только на проверку конфигурации.

```
{
  "rate-shaping": {
    "class": [
      {
        "name": "guest-limit",
        "description": "Полоса гостевого сегмента",
        "rate": "10mbps",
        "ceiling": "15mbps",
        "max-burst": "2mb",
        "direction": "both",
        "drop-policy": "tail",
        "queue": "pfifo"
      }
    ]
  },
  "policy": {
    "rule": [
      {
        "name": "guest-traffic",
        "match": { "source-ip": ["10.200.0.0/16"] },
        "action": "passthrough",
        "rate-class": "guest-limit"
      }
    ]
  }
}
```

Ссылка проверяется при применении. Правило с несуществующим классом не пройдёт проверку. Веб-интерфейс также не позволит удалить класс, на который ссылается правило. Подробнее см. в разделе «Правила политик».

Проверка конфигурации

Отдельных счётчиков для классов нет. Поэтому можно проверить только корректность модели конфигурации:

- конфигурация применяется без ошибок проверки;
- класс отображается на экране «Шейпинг трафика», а правило — со ссылкой на класс на экране «Политики»;
- секция доступна через REST API: `GET /api/v1/config/rate-shaping`.

Идентификация

Раздел описывает настройку подсистемы идентификации пользователей. Амибиг сопоставляет IP-адрес клиента с учётной записью, после чего политика может учитывать имя пользователя и членство в группах.

Соответствия «адрес — пользователь» поступают из syslog-событий входа, PAN User-ID, портала аутентификации или REST API. Каталог LDAP используется отдельно: он предоставляет сведения о пользователях и членстве в группах, но не связывает пользователя с IP-адресом. Настройте каталог и хотя бы один источник соответствий, если правила должны учитывать группы пользователей.

Идентификация пользователей

Раздел объясняет, как Амибиг определяет пользователя по клиентскому адресу и применяет эти сведения в правилах политики.

Назначение

Подсистема идентификации связывает IP-адрес клиента с учётной записью пользователя. После этого политика может учитывать имя пользователя и членство в группах LDAP: например, исключать из расшифрования TLS-трафик группы `Domain Admins` или перенаправлять неопознанных пользователей на портал аутентификации.

Идентификация открывает следующие возможности:

- **Критерии политики по пользователям** — `source-user` (точное имя учётной записи), `source-user-group` (членство в группе LDAP) и `source-user-class` (`known-user` или `unknown`, в зависимости от наличия соответствия для IP-адреса клиента). Подробнее см. в разделе Критерии политики.
- **Действие `portal-redirect`** — перенаправление неопознанного пользователя на портал аутентификации.
- **Передача имени пользователя в сервисную цепочку** — заголовки `X-Authenticated-User` и `X-Client-IP` в запросах к NGFW или DLP (параметр правила `insert-identity-headers`). Одноимённые заголовки из клиентского запроса удаляются, чтобы клиент не мог подменить имя пользователя.

Архитектура

Управляющая подсистема принимает события от внешних источников, синхронизирует справочник пользователей и групп с каталогом LDAP и хранит таблицу соответствий. Dataplane периодически получает эту таблицу в виде

компактного бинарного снимка и определяет пользователя по IP-адресу и, при необходимости, порту источника. Обращений к LDAP и внешним системам при обработке соединения нет.

Схема взаимодействия

Источник	Связь	Назначение
REST API	→	Таблица IP → пользователь
Syslog	→	Таблица IP → пользователь
PAN User-ID	→	Таблица IP → пользователь
Портал	→	Таблица IP → пользователь
Каталог LDAP (группы)	→ синхронизация	Таблица IP → пользователь
Таблица IP → пользователь	→ бинарный снимок	Dataplane: критерии политики
Источники соответствий	Группа	REST API, Syslog, PAN User-ID, Портал

Таблица содержит записи двух видов:

Вид записи	Ключ	Применение
Точная	IP-адрес клиента	Обычное рабочее место: один адрес — один пользователь
С диапазоном портов	IP-адрес и диапазон портов источника	Терминальные серверы и NAT: за одним адресом работают несколько пользователей, различаемых по диапазонам портов

При поиске точная запись имеет приоритет: сначала проверяется соответствие по IP-адресу, и только если его нет — записи с диапазонами портов. Для адресов, у которых соответствия не нашлось, ведётся отрицательный кэш: повторные соединения с того же адреса в течение его срока жизни не вызывают повторный поиск.

Управляющая подсистема периодически сохраняет таблицу на диск и восстанавливает её после перезагрузки. В HA-паре активный узел передаёт соответствия резервному. После переключения ролей новый активный узел использует уже заполненную таблицу. Каждый узел синхронизирует справочник LDAP самостоятельно.

Источники соответствий

Соответствие «адрес — пользователь» может появиться в таблице четырьмя путями. Все источники независимы и могут работать одновременно; подробная настройка каждого описана на странице Источники соответствий.

Источник	Как работает	Типовой сценарий
REST API	Внешняя система отправляет события входа/выхода через <code>POST /api/v1/identity/mappings</code>	Интеграция с SIEM, скрипты, контроллеры доменов с агентом
Syslog	Амибиг принимает syslog-события входа и извлекает имя и адрес регулярными выражениями	Wi-Fi контроллеры, NAC, 802.1x
PAN User-ID	Амибиг опрашивает межсетевой экран Palo Alto по XML API и забирает готовые соответствия	В сети уже развёрнут PAN User-ID
Портал	Неопознанный пользователь входит на портале аутентификации с доменной учётной записью	Гостевые сегменты, устройства без агентов

Каталог LDAP при этом не является источником соответствий: он предоставляет справочник пользователей и групп, по которому вычисляется членство для критериев `source-user-group`. Настройка каталога описана на странице Каталог LDAP.

Время жизни записей

Записи автоматически удаляются по истечении срока действия. Это исключает применение старого соответствия после того, как DHCP назначил адрес другому устройству.

Параметр	Диапазон	По умолчанию	Описание
<code>user-ttl-minutes</code>	1-1440	45	Время жизни соответствия, минуты. Применяется к записям от syslog, к записям PAN, не принёсшим собственного времени жизни, и к REST-записям без явного <code>timeout_minutes</code>
<code>negative-ttl-seconds</code>	0-65535	60	Время жизни записи отрицательного кэша для неизвестного IP, секунды. 0 отключает отрицательный кэш
<code>portal.idle-ttl-seconds</code>	≥ 1	900	TTL простоя для записей, созданных порталом; обновляется при повторном входе
<code>portal.max-ttl-seconds</code>	≥ 1	3600	Абсолютный предел жизни порталной записи, отсчитывается от первого входа

REST-запись с `timeout_minutes: 0` создаётся бессрочной — она живёт до явного `logout` или до очистки таблицы.

Включение подсистемы

Подсистема идентификации выключена по умолчанию. Откройте экран `Идентификация` и включите параметр «Включить идентификацию пользователей» в карточке «Конфигурация идентификации». Для настройки через REST API добавьте секцию `identity` в подготовленную конфигурацию:

```
{
  "amibig:config": {
    "identity": {
      "enabled": true,
      "ldap-server": [
        {
          "host": "dc1.corp.local",
          "port": 636,
          "use-tls": true,
          "tls-verify": true,

```

```

        "bind-dn": "CN=svc-amibig,CN=Users,DC=corp,DC=local",
        "bind-password": "<пароль сервисной учётной записи>",
        "base-dn": "DC=corp,DC=local"
    }
],
"sync-interval": 3600,
"user-ttl-minutes": 45,
"group-include": ["VPN-Users", "Domain Admins"]
}
}
}

```

Проверка состояния

Откройте экран **Идентификация** в разделе меню «Инспекция». На экране отображаются следующие сведения:

- **Статус синхронизации LDAP** — включён ли модуль, доступен ли LDAP-клиент, число пользователей и групп в справочнике, время последней синхронизации, отметка синхронизации (uSNChanged) и последняя ошибка.
- **Соответствия IP → пользователь** — общее число записей, распределение по источникам, число диапазонов портов и состояние репликации в HA-паре.
- **Поиск IP ↔ пользователь** — поиск записей по адресу и/или имени: источник, диапазон портов и время истечения каждой записи.

Те же данные доступны через REST API:

- `GET /api/v1/identity/mappings?ip=<адрес>&user=<имя>` — поиск соответствий; оба параметра необязательны;
- `GET /api/v1/identity/status` — статус синхронизации и счётчики по источникам.

В ответе на запрос поиска возвращаются найденные записи и их количество:

```

{
  "items": [
    {
      "ip": "10.20.0.15",
      "user": "ivanov",
      "source": "syslog",
      "expires_at": 1767105300.0,
      "event_ts": 1767102600.0
    }
  ],
  "count": 1
}

```

Просмотр доступен всем ролям. Администратор и оператор могут изменять таблицу, а полностью очистить её методом `DELETE /api/v1/identity/mappings` может только администратор.

Связанные разделы

Следующие разделы описывают настройку каталога, источников соответствий и применение полученных данных в политике:

- Каталог LDAP — подключение к контроллерам домена, синхронизация групп, диагностика.
- Источники соответствий — syslog-приёмник, коннектор PAN User-ID, ручные записи через REST API.
- Портал аутентификации — вход неопознанных пользователей с доменной учётной записью.
- Критерии политики — как использовать пользователей и группы в правилах.

Каталог LDAP

В этом разделе описаны подключение Амибиг к корпоративному каталогу, синхронизация групп и проверка состояния соединения.

Назначение

Каталог LDAP (Active Directory или совместимый с ним Samba AD DC) предоставляет сведения о пользователях и членстве в группах. Эти данные используются критерием политики `source-user-group` и порталом аутентификации при проверке доменных учётных данных. Каталог не создаёт соответствия «адрес — пользователь»: IP-адреса поступают из источников соответствий.

Откройте экран `Идентификация`, затем в карточке «Конфигурация идентификации» найдите блок «Серверы LDAP (Active Directory)». Серверы опрашиваются в указанном порядке; синхронизация выполняется через первый доступный сервер.

Параметры сервера

Для каждого сервера задаются реквизиты подключения и параметры защиты канала:

Параметр	Поле в веб-интерфейсе	По умолчанию	Описание
<code>host</code>	Хост	—	IP-адрес или FQDN контроллера домена. Значение должно быть уникальным в списке серверов

Параметр	Поле в веб-интерфейсе	По умолчанию	Описание
<code>port</code>	Порт	<code>636</code>	Порт LDAP, 1-65535. <code>636</code> соответствует LDAPS; при отключённом TLS обычно <code>389</code>
<code>use-tls</code>	TLS	<code>true</code>	Подключаться по LDAPS. Включено по умолчанию: портал аутентификации отправляет через это подключение доменный пароль пользователя, и на открытом порту 389 он шёл бы в открытом виде
<code>tls-verify</code>	Проверять сертификат	<code>true</code>	Проверять сертификат контроллера домена. Без проверки канал шифруется, но подлинность контроллера не подтверждается — пароли и членство в группах можно подменить. Отключать только на лабораторном стенде
<code>tls-ca</code>	CA контроллера домена	системное хранилище	Ссылка на сертификат типа <code>ca</code> или <code>bundle</code> из раздела SSL/TLS — удостоверяющий центр, выпустивший сертификат контроллера домена. Пустое значение — используется системное хранилище доверия

Параметр	Поле в веб-интерфейсе	По умолчанию	Описание
<code>bind-dn</code>	Bind DN	—	DN сервисной учётной записи для чтения каталога, например <code>CN=svc- amibig,CN=Users,DC=corp,DC=local</code>
<code>bind-password</code>	Пароль bind	—	Пароль сервисной учётной записи. Не записывается в журнал и не возвращается в сведениях о состоянии
<code>base-dn</code>	Base DN	—	База поиска пользователей и групп, например <code>DC=corp,DC=local</code>

Сервисной учётной записи достаточно права чтения каталога — административные права ей не нужны.

Фильтры поиска и членство в группах

Фильтры поиска заданы в системе и не настраиваются. Полная синхронизация выбирает пользователей запросом `(objectClass=user)`, а группы — запросом `(objectClass=group)` в пределах `base-dn`. Данные запрашиваются страницами по 1000 записей в соответствии с RFC 2696.

Членство вычисляется по атрибуту `tokenGroups`, который учитывает вложенные и основные группы пользователя. Если `tokenGroups` отсутствует, система определяет основную группу по `primaryGroupID`. Поэтому критерий `source-user-group: VPN-Users` срабатывает и для пользователя, включённого в `VPN-Users` через вложенную группу.

Отслеживаемые группы

Укажите в списке «Отслеживаемые группы LDAP» (`group-include`) только те группы, которые используются в правилах политики:

Параметр	Ограничение	Описание
<code>group-include</code>	не более 640 имён	Имена групп LDAP (<code>sAMAccountName</code>), доступные критериям политики

Список содержит не более 640 групп. Dataplane проверяет членство при обработке каждого соединения, поэтому число групп ограничено. Если группа не включена в список, правило с критерием `source-user-group` для неё не сработает, даже если группа существует в каталоге.

Интервал синхронизации

Параметр `sync-interval` определяет период между плановыми синхронизациями справочника:

Параметр	Диапазон	По умолчанию	Описание
<code>sync-interval</code>	60-86400	3600	Интервал синхронизации групп, секунды

После включения выполняется полная синхронизация. Затем система запрашивает записи, изменившиеся после предыдущего прохода, по атрибуту `uSNChanged`. Отметка синхронизации относится к конкретному контроллеру домена. При переходе на другой сервер система повторно выполняет полную синхронизацию. После ошибки новая попытка начинается не позднее чем через 60 секунд, независимо от значения `sync-interval`.

Если каталог недоступен, узел продолжает использовать последний успешно синхронизированный справочник. Причина сбоя отображается в состоянии синхронизации.

Ниже приведён пример секции с двумя контроллерами домена:

```
{
  "amibig:config": {
    "identity": {
      "enabled": true,
      "ldap-server": [
        {
          "host": "dc1.corp.local",
          "port": 636,
          "use-tls": true,
          "tls-verify": true,
          "tls-ca": "corp-root-ca",
          "bind-dn": "CN=svc-amibig,CN=Users,DC=corp,DC=local",
          "bind-password": "<пароль>",
          "base-dn": "DC=corp,DC=local"
        },
        {
          "host": "dc2.corp.local",
          "bind-dn": "CN=svc-amibig,CN=Users,DC=corp,DC=local",
          "bind-password": "<пароль>",
          "base-dn": "DC=corp,DC=local"
        }
      ],
      "sync-interval": 3600,
      "group-include": ["VPN-Users", "Finance", "Domain Admins"]
    }
  }
}
```

Как проверить подключение

Откройте экран **Идентификация** и проверьте карточку «Статус синхронизации LDAP». Данные в карточке обновляются автоматически:

- **Каталог LDAP: доступен / недоступен** — наличие LDAP-клиента на узле. Состояние «Недоступен» указывает на ошибку установки системы, а не на сетевую недоступность контроллера домена. Обратитесь в техническую поддержку Amicon.
- **Последняя синхронизация** — время последнего успешного прохода. Состояние «Ещё не выполнялась» допустимо сразу после применения конфигурации. Если время не появилось в течение двух минут, проверьте поле «Последняя ошибка».
- **Пользователей / Групп** — размер синхронизированного справочника. Нули при успешной синхронизации означают, что `base-dn` указывает не туда либо список отслеживаемых групп пуст.
- **Отметка синхронизации (uSNChanged)** — контроллер домена и позиция инкрементальной синхронизации. Заполненная отметка подтверждает завершение полной синхронизации.
- **Последняя ошибка** — текст последней ошибки подключения или синхронизации.

Те же данные возвращает метод `GET /api/v1/identity/status` в полях `ldap_available`, `ldap.last_sync`, `ldap.users`, `ldap.groups`, `ldap.last_error` и `watermark`. Пароль сервисной учётной записи и API-ключи в ответ не включаются.

Типичные ошибки

Найдите симптом в таблице, устраните указанную причину и дождитесь следующей попытки синхронизации:

Симптом	Причина и действие
В «Последней ошибке» — <code>invalid credentials</code> или <code>bind failed</code>	Проверьте <code>bind-dn</code> и пароль сервисной учётной записи. После исправления повторная попытка выполняется в течение минуты
Ошибка проверки сертификата (<code>certificate verify failed</code>)	Сертификат контроллера домена выпущен внутренним УЦ, которого нет в системном хранилище. Загрузить сертификат УЦ в разделе SSL/TLS и выбрать его в поле «CA контроллера домена». Отключение «Проверять сертификат» — только для лабораторного стенда

Симптом	Причина и действие
Подключение обрывается сразу после установления	Проверьте сочетание порта и режима TLS: для LDAPS (<code>use-tls: true</code>) обычно используется порт 636, для подключения без TLS — порт 389
Ошибка содержит <code>returned no usable entries</code>	Проверьте <code>base-dn</code> и состояние контроллера домена. Система не заменяет действующий справочник пустым результатом и автоматически повторяет синхронизацию
Правило по группе не срабатывает, хотя пользователь в ней состоит	Проверьте, что точное имя группы (<code>sAMAccountName</code>) внесено в список «Отслеживаемые группы LDAP». Членство в остальных группах не загружается
Пользователи есть, но соответствий нет	Каталог предоставляет только справочник; адреса приходят из источников соответствий. Проверить, что хотя бы один источник настроен и создаёт записи

Источники соответствий

В этом разделе приведены способы получения соответствий между клиентскими адресами и учётными записями пользователей.

Обзор

Соответствия «адрес — пользователь» поступают из четырёх независимых источников: syslog-событий входа, PAN User-ID, портала аутентификации и REST API. Источники можно использовать одновременно. Для каждой записи сохраняется её источник, а на экране **Идентификация** отображаются отдельные счётчики. На этой странице описана настройка syslog, PAN User-ID и REST API; портал рассмотрен в отдельном разделе.

Источник создаёт точное соответствие по IP-адресу либо соответствие по IP-адресу и диапазону портов источника. Второй вариант предназначен для терминальных серверов и сетей с NAT. Если источник не передал срок действия, используется параметр `user-ttl-minutes` (по умолчанию 45 минут). Подробнее см. в разделе **Идентификация пользователей**.

Syslog-приёмник событий входа

Амибиг принимает по UDP syslog-события от контроллеров Wi-Fi, NAC, серверов 802.1x и прокси-серверов. Имя пользователя и IP-адрес извлекаются из текста события с помощью регулярных выражений. Настройка находится в блоке «Источник Syslog» на экране **Идентификация**.

Syslog по UDP не проверяет подлинность отправителя. До включения приёмника ограничьте доступ к выбранному порту средствами сетевой инфраструктуры и заполните список разрешённых источников.

Параметр	Поле в веб-интерфейсе	По умолчанию	Описание
<code>syslog.enabled</code>	Принимать события syslog	<code>false</code>	Включить приёмник
<code>syslog.port</code>	Порт	514	UDP-порт приёмника, 1-65535
<code>syslog.allow-sources</code>	Разрешённые источники (IP)	пусто	IP-адреса доверенных отправителей. Сообщения с других адресов отбрасываются до разбора. При пустом списке приёмник не принимает сообщения
<code>syslog.rule</code>	Правила извлечения	пусто	Список regex-правил разбора

Датаграммы длиннее 2048 байт отбрасываются без разбора.

Правила разбора

Для каждого правила задаются имя и регулярное выражение. Правила проверяются по порядку. После первого совпадения создаётся запись, остальные правила не проверяются.

Параметр	Ограничение	Описание
<code>name</code>	—	Имя правила, например <code>wifi-login</code>
<code>regex</code>	1-512 символов	Регулярное выражение. Обязано содержать именованные группы <code>user</code> и <code>ip</code> — по ним извлекаются учётная запись и адрес

Длина выражения ограничена, чтобы сложный шаблон не задерживал обработку последующих событий.

В следующем примере контроллер Wi-Fi отправляет сообщения вида `... login ok user=ivanov ip=10.20.0.15 ...`:

```

{
  "amibig:config": {
    "identity": {
      "syslog": {
        "enabled": true,
        "port": 514,
        "allow-sources": ["10.0.5.1", "10.0.5.2"],
        "rule": [
          {
            "name": "wifi-login",
            "regex": "login ok user=(?P<user>\\S+) ip=(?P<ip>\\S+)"
          }
        ]
      }
    }
  }
}

```

Проверьте результат методом `GET /api/v1/identity/status`. В блоке `syslog` счётчик `matched` показывает число событий, по которым созданы записи, а `dropped` — число отброшенных событий. Если увеличивается только `dropped`, проверьте адрес отправителя в списке разрешённых источников, длину сообщения и соответствие регулярного выражения фактическому тексту события.

Интеграция PAN-OS (User-ID)

При интеграции с Palo Alto Networks User-ID Амибиг получает готовые соответствия с межсетевого экрана PAN-OS. Коннектор периодически выполняет по HTTPS запрос XML API `show user ip-user-mapping all`. Настройка находится в блоке «Коннектор PAN User-ID» на экране `Идентификация`.

Параметр	Поле в веб-интерфейсе	По умолчанию	Описание
<code>pan.enabled</code>	Опрашивать Palo Alto	<code>false</code>	Включить коннектор
<code>pan.host</code>	Хост	—	Адрес межсетевого экрана или агента User-ID. При включённом коннекторе поле обязательно
<code>pan.api-key</code>	API-ключ	—	Ключ XML API PAN-OS. Не записывается в журнал и не возвращается в сведениях о состоянии

Параметр	Поле в веб-интерфейсе	По умолчанию	Описание
<code>pan.verify-tls</code>	Проверять TLS-сертификат	<code>true</code>	Проверять подлинность сертификата при подключении к API. Отключайте только на изолированном испытательном стенде
<code>pan.poll-interval</code>	Интервал опроса, с	<code>60</code>	Период опроса соответствий, секунды (не менее 1)

```

{
  "amibig:config": {
    "identity": {
      "pan": {
        "enabled": true,
        "host": "10.0.0.1",
        "api-key": "<ключ XML API>",
        "verify-tls": true,
        "poll-interval": 60
      }
    }
  }
}

```

Коннектор использует время простоя из ответа PAN-OS и округляет его вверх до целых минут. Если PAN-OS не вернул это значение, применяется общий параметр `user-ttl-minutes`. Созданные записи учитываются в счётчике «PAN User-ID» на экране **Идентификация**.

Ручные соответствия через API

Метод `POST /api/v1/identity/mappings` принимает набор событий входа и выхода в формате PAN User-ID: действие `login` создаёт соответствие, а `logout` удаляет его. Метод предназначен для интеграции с SIEM и агентами, а также для создания временных тестовых записей администратором.

Право на добавление есть у администратора и оператора; полная очистка таблицы (`DELETE /api/v1/identity/mappings`) — только у администратора.

Поля записи:

Поле	Обязательное	Описание
<code>action</code>	нет	<code>login</code> (по умолчанию) или <code>logout</code>

Поле	Обязательное	Описание
<code>user</code>	да	Имя учётной записи. Управляющие символы запрещены — имя дальше попадает в HTTP/ICAP-заголовки
<code>ip</code>	да	IPv4- или IPv6-адрес клиента
<code>timeout_minutes</code>	нет	Время жизни записи, минуты, 0–1440. Не указано — используется <code>user-ttl-minutes</code> ; 0 — запись бессрочная
<code>port_start</code> , <code>port_end</code>	нет	Диапазон портов источника, 1–65535. Поля задаются только вместе; <code>port_start</code> не должен превышать <code>port_end</code>

В примере ниже создаются две записи, в том числе запись для терминального сервера с диапазоном портов, и удаляется третья запись:

```
{
  "entries": [
    {
      "action": "login",
      "user": "ivanov",
      "ip": "10.20.0.15",
      "timeout_minutes": 480
    },
    {
      "action": "login",
      "user": "petrov",
      "ip": "10.20.0.200",
      "port_start": 20000,
      "port_end": 29999
    },
    {
      "action": "logout",
      "user": "sidorov",
      "ip": "10.20.0.77"
    }
  ]
}
```

Ответ содержит счётчики `applied` и `removed`, а также поле `ha_role`. В HA-паре отправляйте записи на активный узел. Запись, принятая резервным узлом, остаётся локальной до смены ролей и может быть заменена при репликации с активного узла. В этом случае ответ содержит поле `warning`.

Повторное действие `login` для того же адреса заменяет существующую запись и обновляет срок её действия. При поиске сначала проверяется точное соответствие по IP-адресу, затем — соответствие по диапазону, в который входит порт источника соединения.

Проверьте созданную запись методом `GET /api/v1/identity/mappings?ip=10.20.0.15` или с помощью поиска «IP ↔ пользователь» на экране Идентификация. В ответе должны отображаться имя пользователя, источник и срок действия записи.

Портал аутентификации

Раздел описывает настройку портала для пользователей, которых не удалось определить по данным внешних источников.

Назначение

Портал аутентификации (captive portal) идентифицирует пользователей, для которых нет данных от syslog, PAN User-ID или другого источника. Правило политики перенаправляет такого пользователя на страницу входа. После проверки учётных данных в LDAP Амибиг создаёт соответствие между IP-адресом и именем пользователя. Последующие соединения обрабатываются с учётом имени пользователя.

Для работы с порталом не требуется устанавливать программный агент на устройство пользователя.

Сценарий перенаправления

Перенаправление задаётся правилом политики с действием `portal-redirect`. Обычно правило сочетается с критерием `source-user-class: unknown` — «клиентский адрес без соответствия» (см. Критерии политики):

```
{
  "amibig:config": {
    "policy": {
      "rule": [
        {
          "name": "unknown-to-portal",
          "match": {
            "source-user-class": "unknown"
          },
          "action": "portal-redirect"
        }
      ]
    }
  }
}
```

После применения правила портал работает следующим образом:

1. Неидентифицированный пользователь открывает веб-страницу. Соединение соответствует правилу с действием `portal-redirect`.
2. Для HTTPS Амибиг выполняет SSL/TLS-инспекцию и на первый HTTP-запрос отвечает кодом `302` с адресом портала. Параметр `orig` формируется из разобранных заголовка `Host` и URI исходного запроса. Значение проходит

проверку и кодируется перед передачей portalу; клиент не может задать произвольный адрес перенаправления.

3. Пользователь входит на портале. Появляется соответствие «адрес — пользователь» с источником `portal`.
4. Следующие соединения с этого адреса под правило для неизвестных уже не попадают и обрабатываются политикой по имени пользователя, а портал возвращает пользователя на исходную страницу.

Действие `portal-redirect` требует включённого портала и заданного `portal.url`. Конфигурация с выключенным порталом или пустым адресом будет отклонена при проверке. Отдельно убедитесь, что включена подсистема `identity`: иначе `dataplane` не сможет определить пользователя и запишет предупреждение при загрузке конфигурации.

Параметры портала

Портал настраивается в блоке «Портал аутентификации» на экране «Идентификация». Страница входа доступна по пути `/portal` без токена веб-интерфейса. Поэтому доступ к ней следует ограничить клиентскими сегментами, для которых нужна аутентификация.

Параметр	Подпись в UI	По умолчанию	Описание
<code>portal.enabled</code>	Включить портал	<code>false</code>	Включить портал
<code>portal.url</code>	URL портала	—	HTTPS-адрес перенаправления, например <code>https://portal.example.com/portal</code> . Адрес должен быть доступен из клиентских сегментов. При включённом портале поле обязательно
<code>portal.idle-ttl-seconds</code>	TTL простоя, с	<code>900</code>	Время жизни портальной записи; каждый повторный успешный вход отсчитывает его заново

Параметр	Подпись в UI	По умолчанию	Описание
<code>portal.max-ttl-seconds</code>	Максимальный TTL, с	3600	Абсолютный предел жизни записи, отсчитывается от первого входа. По истечении пользователь входит заново, даже если активно работал

Запись удаляется при истечении любого из двух сроков: времени простоя или максимального времени жизни. По умолчанию срок составляет 15 минут простоя, но не более одного часа с момента входа.

Для проверки пароля портал выполняет LDAP bind от имени пользователя. Имя передаётся в формате UPN (`имя@домен`), а домен составляется из компонентов `DC=` параметра `base-dn`. Без настроенного LDAP-сервера портал не выполняет аутентификацию.

Что видит пользователь

Страница входа — карточка с логотипом Амибиг, пояснением «Доступ в сеть предоставляется после входа с вашей корпоративной учётной записью», полями «Имя пользователя» и «Пароль» и кнопкой «Войти». Пользователь вводит доменную учётную запись — короткое имя, без `@домен`.

После успешного входа отображается страница «Доступ разрешён»; если пользователь пришёл по перенаправлению, через две секунды браузер автоматически возвращается на исходную страницу (есть и ссылка «перейти сейчас»).

Возможные сообщения об ошибках:

Сообщение	Причина
«Укажите корректное имя пользователя» / «Укажите пароль»	Пустое поле, имя длиннее 63 символов или недопустимые символы в имени
«Неверное имя пользователя или пароль»	Каталог отверг учётные данные
«Слишком много попыток входа. Повторите позже.»	Сработало ограничение попыток (см. ниже)
«LDAP-сервер не настроен» / «Служба идентификации недоступна»	Портал включён, но подсистема идентификации не настроена: проверить, что на экране Идентификация задан хотя бы один LDAP-сервер и конфигурация применена

Ограничение попыток

Портал — страница без предварительной аутентификации, поэтому попытки входа ограничиваются: не более 5 попыток за 60 секунд с одного клиентского адреса. При превышении пользователь получает сообщение «Слишком много попыток входа» (HTTP 429 с заголовком `Retry-After`), и счётчик сбрасывается по мере устаревания неудачных попыток. Ограничение действует на каждый адрес отдельно и ведётся независимо от ограничения попыток входа в веб-интерфейс: перебор паролей на портале не блокирует вход администратора в интерфейс управления.

Защита от подмены

При размещении портала учитывайте следующие механизмы защиты:

- Адрес для соответствия берётся только из адреса клиентского соединения с порталом — никакие параметры запроса или формы на него не влияют, «зарегистрировать» чужой IP невозможно.
- Пароль передаётся исключительно в LDAP bind: он не сохраняется, не сравнивается локально и не попадает в журналы. Поэтому подключение к каталогу по умолчанию защищено LDAPS с проверкой сертификата — см. Каталог LDAP.
- Параметр возврата `orig` принимает относительный путь или абсолютный адрес HTTP/HTTPS. Значения с другой схемой, внедрёнными учётными данными или управляющими символами отбрасываются. В текущей версии абсолютный адрес не ограничен настроенным узлом, поэтому формировать ссылки на портал с `orig` должны только доверенные компоненты.

Что видит администратор

Состояние портала и результаты входа можно проверить тремя способами:

- **Экран Идентификация** — счётчик «Портал» в карточке «Соответствия IP → пользователь»; поиск «IP ↔ пользователь» показывает записи с источником `portal` и временем истечения.
- **Журнал** — события категории `auth: Portal login successful` и `Portal login failed` с именем пользователя и адресом клиента (пароль в журнал не попадает никогда). Серии `Portal login failed` с одного адреса — признак перебора паролей.
- **REST API** — `GET /api/v1/identity/mappings?user=<имя>` для проверки конкретной записи, `GET /api/v1/identity/status` для счётчиков по источникам.

В HA-паре записи портала, как и остальные, реплицируются с активного узла на резервный; вход, принятый порталом резервного узла, остаётся локальным для него до смены ролей.

aRules

aRules — Tcl-совместимый механизм для программируемой обработки HTTP-трафика. Он используется для изменения заголовков, выбора пула, ведения журнала и других действий, которые невозможно выразить только стандартными правилами политики.

Раздел содержит обзор модели выполнения, справочник поддерживаемых команд, события и практические примеры. Команды F5 iRules, не описанные в справочнике, в текущей версии не реализованы.

Обзор aRules

Раздел знакомит с моделью выполнения aRules, способами привязки правил и ограничениями текущей версии.

Назначение

aRules (Amicon Rules) — встроенный Tcl-совместимый механизм обработки HTTP-трафика. Он изменяет HTTP-заголовки, управляет ICAP-адаптацией, выбирает пул назначения и записывает сведения о запросах в журнал.

Совместимость с F5 iRules

aRules использует синтаксис F5 iRules. Правила можно переносить из F5 BIG-IP SSL Orchestrator без перехода на другой язык, но перед импортом необходимо проверить используемые события и команды по справочникам Амибиг.

Поддерживается совместимость на уровне:

- **Синтаксис Tcl** — стандартные конструкции `if`, `switch`, `foreach`, `proc`, `set`, `expr`, `string`, `regex` и другие.
- **Событийная модель** — обработчики `RULE_INIT`, `HTTP_REQUEST`, `HTTP_RESPONSE`, `LB_FAILED` и `ICAP_RESPONSE`. Обработчик `HTTP_REQUEST_DATA` распознаётся, но в текущей версии не вызывается.
- **Команды F5** — реализованы группы `HTTP::*`, `ADAPT::*`, `ICAP::*`, `IP::*`, `LB::*`, `IDENT::*`, команды `pool`, `node`, `class`, `table`, `connect`, `send`, `recv`, `close`, `event` и `log`.

Полный перечень приведён в справочнике команд и справочнике событий. Команды и события F5 iRules, отсутствующие в справочниках, не реализованы. До применения импортированного правила замените или удалите такие конструкции.

Пример совместимости

Оригинальный F5 iRule:

```
when HTTP_REQUEST {
    if { [HTTP::method] eq "POST" } {
        if { [HTTP::header exists Content-Length] } {
            set clen [HTTP::header value Content-Length]
            if { $clen <= 1000 } {
                ADAPT::enable request false
            }
        }
    }
}
```

В этом примере используются только поддерживаемые команды, поэтому код можно применить в Амибиг без изменений.

Интерпретатор Tcl

aRules выполняются встроенным Tcl-совместимым интерпретатором. Если библиотека интерпретатора недоступна или не прошла проверку совместимости, dataplane продолжает работу без выполнения aRules и записывает причину в журнал.

Безопасная среда выполнения

Интерпретатор регистрирует ограниченный набор команд. Прямой доступ к файловой системе, запуск программ и загрузка внешних расширений недоступны. Для исходящих TCP-запросов используются только предусмотренные команды sideband-соединений.

Доступны безопасные команды Tcl: `set`, `expr`, `if`, `proc`, `return`, `string`, `regexp`, `switch`, `foreach`, `list`, `catch`, `format`, `split`, `join`, `append`, `incr` и другие.

Поток обработки

Для совпавшего правила dataplane собирает aRules из сервисной цепочки и самого правила политики, затем выполняет обработчики в контексте HTTP-сообщения:

```
HTTP-запрос
|
v
Оценка политики --> совпавшее правило, действие, цепочка
|
v
Сбор aRules: aRules из сервисов цепочки + aRules из правила политики
|
v
RULE_INIT --> инициализация переменных и процедур
|
v
HTTP_REQUEST --> обработка запроса (модификация заголовков, ADAPT-флаги)
|
v
Применение изменений к HTTP-данным
```

```
|
v
Передача в сервисную цепочку (ICAP, inline NGFW)
```

Область видимости переменных

Для каждого события создаётся новый Tcl-интерпретатор, а перед обработчиком выполняется `RULE_INIT`. Глобальные переменные (`::var`) из `RULE_INIT` доступны обработчику, но их изменения не сохраняются для следующего события или соединения. Для общего состояния используйте команду `table`.

Привязка aRules

aRules можно привязать к сервису в цепочке или к правилу политики.

К сервису в цепочке

aRule выполняется, когда трафик проходит через сервисную цепочку с указанным сервисом.

```
{
  "name": "dlp-service",
  "type": "icap",
  "arules": ["size_limit"]
}
```

К правилу политики

aRule выполняется, когда соединение совпадает с правилом политики.

```
{
  "name": "inspect-web",
  "match": {
    "dest-port": [443],
    "protocol": "tcp"
  },
  "action": "intercept",
  "service-chain": "main-chain",
  "arules": ["inject_auth", "log_requests"]
}
```

Порядок выполнения

Если aRules привязаны в нескольких местах, они выполняются в следующем порядке:

При наличии aRules в нескольких точках привязки порядок выполнения:

1. aRules из сервисов цепочки (в порядке расположения сервисов в цепочке).
2. aRules из совпавшего правила политики.

Повторяющиеся имена исключаются. Если одно правило aRules привязано к сервису и правилу политики, оно выполняется один раз.

Типичные сценарии использования

Выбирайте aRules для операций, которые нельзя задать стандартными условиями и действиями политики:

Сценарий	Описание
Исключение малых запросов из DLP-проверки	Отключение ICAP-адаптации для PUT/POST с телом менее заданного размера
Добавление заголовков аутентификации	Добавление <code>X-Authenticated-User</code> в HTTP-запросы
Удаление служебных заголовков	Удаление внутренних заголовков перед отправкой на внешние сервисы
Условная маршрутизация	Управление ICAP-адаптацией на основе URI или метода запроса
Журналирование запросов	Запись сведений о запросах в системный журнал

Валидация синтаксиса

Перед применением конфигурации система проверяет синтаксис каждого правила aRules:

1. Парсинг структуры `when EVENT { body }`.
2. Компиляция тела обработчика через пробный `proc`.

При ошибке синтаксиса конфигурация не применяется. Исправьте правило по диагностическому сообщению и повторите проверку.

Управление через API

aRules хранятся в отдельной секции конфигурации. Для управления и предварительной проверки доступны следующие методы REST API:

Метод	Путь	Описание
GET	<code>/api/v1/config/arules</code>	Получить секцию aRules
PUT	<code>/api/v1/config/arules</code>	Записать секцию aRules (создание, изменение, удаление правил)
POST	<code>/api/v1/arules/validate</code>	Проверить Tcl-синтаксис кода aRule без применения

Изменения вступают в силу после применения конфигурации (`POST /api/v1/config/apply`).

Ограничения

При разработке и переносе правил учитывайте ограничения текущей версии:

- aRules выполняются только для **HTTP-трафика** (не для FTP, SMTP, raw TCP).
- Глобальные переменные не разделяются между обработчиками и соединениями; состояние между запросами хранится через `table`.
- Обрабатывается первый HTTP-запрос в соединении; последующие запросы в keep-alive соединении не вызывают повторного выполнения aRule.
- Время выполнения Tcl-кода не ограничивается. Бесконечный цикл остановит обработку трафика соответствующим рабочим потоком dataplane. Проверяйте aRules на испытательном устройстве до применения в рабочей конфигурации.

Справочник команд aRules

Справочник содержит точный синтаксис, область применения и результат поддерживаемых команд aRules.

Обзор

Команды aRules расширяют Tcl-интерпретатор и используют синтаксис F5 iRules. Они работают с текущим соединением и доступны в обработчиках, указанных в справочнике событий.

Справочник перечисляет только команды, зарегистрированные в текущей версии. Команды, которых нет в этом справочнике, в Амибиг не реализованы — правило, использующее их, завершится ошибкой выполнения (она записывается в журнал, обработка трафика при этом продолжается).

Группы команд:

- `HTTP::` — работа с HTTP-запросами и ответами.
- `ADAPT::` — управление ICAP-адаптацией.
- `ICAP::` — инспекция ответа ICAP-сервера.
- `IP::` — адреса текущего соединения.
- `IDENT::` — имя пользователя и отслеживаемые группы LDAP.
- `LB::` — состояние членов пула и повторный выбор.
- `pool` , `node` — переопределение выбора пула или узла назначения.
- `class` , `table` — именованные наборы данных и таблица «ключ — значение».
- `connect` , `send` , `recv` , `close` (`SIDEBAND::*`) — вспомогательные sideband-соединения.
- `event` — сведения о текущем событии.

- `log` — запись в системный журнал.

HTTP-команды

Команды `HTTP::` читают и изменяют данные текущего HTTP-запроса или ответа.

HTTP::method

Возвращает HTTP-метод текущего запроса.

Параметр	Значение
Синтаксис	<code>HTTP::method</code>
Возвращает	Строка: <code>GET</code> , <code>POST</code> , <code>PUT</code> , <code>DELETE</code> , <code>HEAD</code> , <code>OPTIONS</code> , <code>PATCH</code> и др.
Доступен в	<code>HTTP_REQUEST</code>

```
if { [HTTP::method] eq "POST" } {
    log local0. "POST-запрос обнаружен"
}
```

HTTP::uri

Возвращает URI текущего запроса (путь + строка запроса).

Параметр	Значение
Синтаксис	<code>HTTP::uri</code>
Возвращает	Строка, например <code>/api/data?page=1</code>
Доступен в	<code>HTTP_REQUEST</code>

```
if { [string match "/api/internal/*" [HTTP::uri]] } {
    ADAPT::enable request false
}
```

HTTP::host

Возвращает значение заголовка `Host`.

Параметр	Значение
Синтаксис	<code>HTTP::host</code>
Возвращает	Строка, например <code>www.example.com</code>
Доступен в	<code>HTTP_REQUEST</code>

```
if { [HTTP::host] eq "admin.internal.corp" } {
    HTTP::header insert X-Admin-Access "true"
}
```

HTTP::version

Возвращает версию HTTP из строки запроса.

Параметр	Значение
Синтаксис	<code>HTTP::version</code>
Возвращает	Строка, например <code>HTTP/1.1</code>
Доступен в	<code>HTTP_REQUEST</code>

HTTP::status

Возвращает код статуса HTTP-ответа. С аргументом — переопределяет код статуса, который будет отправлен клиенту.

Параметр	Значение
Синтаксис	<code>HTTP::status</code> или <code>HTTP::status <code></code>
Возвращает	Целое число, например <code>200</code> , <code>404</code> , <code>500</code>
Доступен в	<code>HTTP_RESPONSE</code>

```
when HTTP_RESPONSE {
    if { [HTTP::status] >= 500 } {
        log local0. "Ошибка сервера: [HTTP::status]"
    }
}
```

HTTP::header

Группа подкоманд для работы с HTTP-заголовками. В `HTTP_REQUEST` команды работают с заголовками запроса, в `HTTP_RESPONSE` — с заголовками ответа.

HTTP::header exists

Проверяет наличие заголовка.

Параметр	Значение
Синтаксис	<code>HTTP::header exists <name></code>
Возвращает	<code>1</code> (есть) или <code>0</code> (нет)

Параметр	Значение
Доступен в	HTTP_REQUEST , HTTP_RESPONSE

```
if { [HTTP::header exists X-Forwarded-For] } {
    log local0. "Запрос через прокси: [HTTP::header value X-Forwarded-For]"
}
```

HTTP::header value

Возвращает значение заголовка.

Параметр	Значение
Синтаксис	HTTP::header value <name>
Возвращает	Строка — значение заголовка. Пустая строка, если заголовок не найден
Доступен в	HTTP_REQUEST , HTTP_RESPONSE

```
set content_type [HTTP::header value Content-Type]
if { [string match "application/json*" $content_type] } {
    log local0. "JSON-запрос к [HTTP::uri]"
}
```

HTTP::header insert

Добавляет новый заголовок в запрос/ответ.

Параметр	Значение
Синтаксис	HTTP::header insert <name> <value>
Возвращает	—
Доступен в	HTTP_REQUEST , HTTP_RESPONSE

```
HTTP::header insert X-Inspected-By "AMIBIG"
HTTP::header insert X-Client-IP [IP::client_addr]
```

HTTP::header remove

Удаляет заголовок из запроса/ответа.

Параметр	Значение
Синтаксис	<code>HTTP::header remove <name></code>
Возвращает	—
Доступен в	<code>HTTP_REQUEST</code> , <code>HTTP_RESPONSE</code>

```
HTTP::header remove X-Authenticated-User
HTTP::header remove X-Debug-Info
```

HTTP::header replace

Заменяет значение существующего заголовка. Если заголовок отсутствует, добавляет новый.

Параметр	Значение
Синтаксис	<code>HTTP::header replace <name> <value></code>
Возвращает	—
Доступен в	<code>HTTP_REQUEST</code> , <code>HTTP_RESPONSE</code>

```
HTTP::header replace Host "backend.internal.corp"
```

HTTP::header names

Возвращает список имён всех заголовков.

Параметр	Значение
Синтаксис	<code>HTTP::header names</code>
Возвращает	Строка с именами заголовков через пробел
Доступен в	<code>HTTP_REQUEST</code> , <code>HTTP_RESPONSE</code>

```
set headers [HTTP::header names]
log local0. "Заголовки: $headers"
```

HTTP::redirect

Выполняет HTTP-перенаправление клиента на указанный URL (ответ 302), не отправляя запрос дальше.

Параметр	Значение
Синтаксис	<code>HTTP::redirect <url></code>
Возвращает	—
Доступен в	<code>HTTP_REQUEST</code>

```
if { [HTTP::host] eq "old.example.com" } {
    HTTP::redirect "https://new.example.com[HTTP::uri]"
}
```

HTTP::respond

Отправляет ответ клиенту напрямую. В `HTTP_REQUEST` — вместо обращения к серверу; в `HTTP_RESPONSE` — вместо ответа сервера. Необязательно принимает тело (`content <body>`) и пары `имя значение`, которые добавляются как заголовки ответа.

Параметр	Значение
Синтаксис	<code>HTTP::respond <code> [content <body>] [<header> <value> ...]</code>
Возвращает	—
Доступен в	<code>HTTP_REQUEST</code> , <code>HTTP_RESPONSE</code>

```
if { [HTTP::uri] eq "/blocked" } {
    HTTP::respond 403 content "Access Denied" Content-Type "text/plain"
}
```

HTTP::payload

Возвращает тело текущего HTTP-сообщения или его длину. Доступны байты тела, полученные вместе с заголовками в разобранном буфере сообщения: для больших тел, приходящих отдельными сегментами, команда возвращает только начальную часть. Отдельного механизма дочитывания тела заданного размера в текущей версии нет.

Параметр	Значение
Синтаксис	<code>HTTP::payload</code> или <code>HTTP::payload length</code>
Возвращает	Строка (тело) или число (длина в байтах)
Доступен в	<code>HTTP_REQUEST</code> (тело запроса), <code>HTTP_RESPONSE</code> (тело ответа)

```
when HTTP_REQUEST {
    if { [HTTP::payload length] > 0 } {
```

```

    log local0. "Тело запроса: [HTTP::payload length] байт"
  }
}

```

HTTP::collect

Запрашивает сбор тела HTTP-запроса. По модели F5 iRules после этой команды накапливается указанный объём тела, и когда собранные данные становятся доступны, наступает событие `HTTP_REQUEST_DATA`, в обработчике которого тело читается командой `HTTP::payload`.

Параметр	Значение
Синтаксис	<code>HTTP::collect [<размер>]</code>
Возвращает	—
Доступен в	<code>HTTP_REQUEST</code>

В текущей версии команда принимается для совместимости с переносимыми правилами F5 (в том числе с аргументом размера), но дочитывание тела отдельными сегментами не выполняется: доступной остаётся часть тела, полученная вместе с заголовками, и событие `HTTP_REQUEST_DATA` не вызывается. Тело в пределах доступной части читают командой `HTTP::payload` прямо в `HTTP_REQUEST`.

```

when HTTP_REQUEST {
    if { [HTTP::method] eq "POST" } {
        HTTP::collect 1024
    }
}

when HTTP_REQUEST_DATA {
    log local0. "Собрано [HTTP::payload length] байт тела запроса"
}

```

ADAPT-команды

Команды `ADAPT::` включают или отключают отдельные фазы обработки ICAP для текущего соединения.

ADAPT::enable

Включает или отключает ICAP-адаптацию для текущего соединения в режиме REQMOD или RESPMOD.

Параметр	Значение
Синтаксис	<code>ADAPT::enable request <bool></code> или <code>ADAPT::enable response <bool></code>

Параметр	Значение
Значения bool	<code>true</code> , <code>false</code> (также <code>1</code> , <code>yes</code>)
Доступен в	<code>HTTP_REQUEST</code>

```
### Отключить ICAP-обработку запроса
ADAPT::enable request false

### Отключить ICAP-обработку ответа
ADAPT::enable response false
```

Когда `ADAPT::enable request false` установлен, запрос передаётся без отправки на ICAP-сервер. Аналогично для `response` . Если оба флага установлены в `false` , ICAP-обработка полностью пропускается для данного соединения.

ADAPT::disable

Отключает адаптацию для указанной фазы.

Параметр	Значение
Синтаксис	<code>ADAPT::disable request</code> или <code>ADAPT::disable response</code>
Доступен в	<code>HTTP_REQUEST</code>

`ADAPT::disable request` эквивалентно `ADAPT::enable request false` . Чтобы отключить обе фазы, команда вызывается дважды — для `request` и для `response` .

ICAP-команды

Команды инспекции ответа ICAP-сервера. Работают только в событии `ICAP_RESPONSE` ; вызов в другом событии завершается ошибкой выполнения.

ICAP::status

Возвращает код статуса ответа ICAP-сервера.

Параметр	Значение
Синтаксис	<code>ICAP::status</code>
Возвращает	Целое число, например <code>200</code> , <code>204</code>
Доступен в	<code>ICAP_RESPONSE</code>

```
when ICAP_RESPONSE {
  if { [ICAP::status] != 204 } {
    log local0. "ICAP модифицировал сообщение: [ICAP::status]"
  }
}
```

```
}
}
```

ICAP::header

Возвращает значение заголовка ответа ICAP-сервера или проверяет его наличие.

Параметр	Значение
Синтаксис	ICAP::header <name> или ICAP::header exists <name>
Возвращает	Значение заголовка (пустая строка, если нет) или 1 / 0 для exists
Доступен в	ICAP_RESPONSE

```
when ICAP_RESPONSE {
  if { [ICAP::header exists X-Infection-Found] } {
    log local0. "DLP-вердикт: [ICAP::header X-Infection-Found]"
  }
}
```

IP-команды

Команды IP:: возвращают адреса сторон проксируемого соединения.

IP::client_addr

Возвращает IP-адрес клиента.

Параметр	Значение
Синтаксис	IP::client_addr
Возвращает	Строка, например 10.1.1.50
Доступен в	HTTP_REQUEST , HTTP_RESPONSE , LB_FAILED , ICAP_RESPONSE

```
log local0. "Запрос от клиента [IP::client_addr]"
```

IP::server_addr

Возвращает IP-адрес сервера назначения.

Параметр	Значение
Синтаксис	IP::server_addr

Параметр	Значение
Возвращает	Строка, например <code>10.2.2.100</code>
Доступен в	<code>HTTP_REQUEST</code> , <code>HTTP_RESPONSE</code> , <code>LB_FAILED</code> , <code>ICAP_RESPONSE</code>

```
if { [IP::server_addr] eq "10.0.0.1" } {
    HTTP::header insert X-Internal "true"
}
```

Команды идентификации

Команды `IDENT::` возвращают данные, найденные подсистемой идентификации для IP-адреса клиента. Если идентификация выключена или пользователь не определён, команды возвращают пустую строку.

`IDENT::user`

Возвращает имя пользователя текущего соединения.

Параметр	Значение
Синтаксис	<code>IDENT::user</code>
Возвращает	Имя пользователя или пустую строку
Доступен в	События с контекстом соединения

```
when HTTP_REQUEST {
    set user [IDENT::user]
    if { $user ne "" } {
        HTTP::header insert X-Authenticated-User $user
    }
}
```

`IDENT::groups`

Возвращает Tcl-список отслеживаемых групп LDAP, в которые входит пользователь.

Параметр	Значение
Синтаксис	<code>IDENT::groups</code>
Возвращает	Список групп или пустую строку
Доступен в	События с контекстом соединения

```
when HTTP_REQUEST {
    if { [lsearch [IDENT::groups] "Domain Admins"] >= 0 } {
        log local0. "Запрос пользователя из группы Domain Admins"
    }
}
```

LB-команды

Команды управляют участниками пулов сервисов. Допустимость команды проверяется во время выполнения; вызов в неподдерживаемом событии завершается ошибкой.

LB::status

Возвращает состояние текущего или указанного участника пула.

Параметр	Значение
Синтаксис	LB::status или LB::status pool <name> member <addr> <port> [<state>]
Возвращает	up, down или session_disabled; с аргументом <state> — 1 или 0
Доступен в	HTTP_RESPONSE, LB_FAILED

```
when LB_FAILED {
    log local0. "Участник пула недоступен: [LB::status]"
}
```

LB::down

Помечает текущего или указанного участника пула недоступным для дальнейшего выбора.

Параметр	Значение
Синтаксис	LB::down или LB::down pool <name> member <addr> <port>
Возвращает	—
Доступен в	HTTP_RESPONSE, LB_FAILED

LB::detach

Отсоединяет текущее соединение от выбранного члена пула.

Параметр	Значение
Синтаксис	<code>LB::detach</code>
Возвращает	—
Доступен в	<code>LB_FAILED</code>

LB::reselect

Запрашивает повторный выбор участника после неудачного подключения, при необходимости из другого пула.

Параметр	Значение
Синтаксис	<code>LB::reselect [pool <name>]</code>
Возвращает	—
Доступен в	<code>LB_FAILED</code>

```
when LB_FAILED {
  LB::down
  LB::reselect
}
```

Команды pool и node

Команды `pool` и `node` переопределяют назначение, выбранное сервисной цепочкой.

pool

Направляет соединение в указанный пул вместо пула, выбранного политикой.

Параметр	Значение
Синтаксис	<code>pool <pool-name></code>
Возвращает	—
Доступен в	<code>HTTP_REQUEST</code>

```
when HTTP_REQUEST {
  if { [string match "/api/v2/*" [HTTP::uri]] } {
    pool api-v2-pool
  }
}
```

node

Направляет соединение на конкретный адрес и порт, минуя выбор участника пула. Порт задаётся отдельным аргументом либо после двоеточия (`addr:port` , для IPv6 — `[addr]:port`).

Параметр	Значение
Синтаксис	<code>node <addr> [<port>]</code> или <code>node <addr>:<port></code>
Возвращает	—
Доступен в	<code>HTTP_REQUEST</code>

```
when HTTP_REQUEST {
  if { [HTTP::host] eq "admin.internal.corp" } {
    node 10.10.0.20 8080
  }
}
```

Команды class и table

Команда `class` читает именованные наборы данных, а `table` хранит кратковременное состояние между выполнениями aRules.

class

Команда выполняет поиск по data-group — именованному набору записей «ключ — значение». Наборы задаются в секции `arules.data-group` и могут быть импортированы из UCS-конфигурации F5.

Параметр	Значение
Синтаксис	<code>class match <item> equals <data-group></code> или <code>class lookup <item> <data-group></code>
Возвращает	<code>match</code> — 1/0; <code>lookup</code> — значение записи (пустая строка, если нет)
Доступен в	Все события

```
when HTTP_REQUEST {
  if { [class match [HTTP::host] equals blocked_hosts] } {
    HTTP::respond 403 content "Blocked"
  }
}
```

table

Таблица «ключ — значение» хранится отдельно в каждом рабочем потоке dataplane. Записи не синхронизируются между рабочими потоками или узлами HA. Таблица содержит не более 1024 записей; длина ключа ограничена 127 символами, значения — 255 символами. Время жизни по умолчанию составляет 300 секунд.

Параметр	Значение
Синтаксис	<code>table lookup <key> / table set <key> <value> [<timeout>] / table delete <key> / table incr <key> [<delta>] [<timeout>]</code>
Возвращает	<code>lookup</code> — значение (пустая строка, если нет); <code>set</code> — записанное значение; <code>incr</code> — новое значение счётчика
Доступен в	Все события

```
when HTTP_REQUEST {
  set count [table incr "req:[IP::client_addr]" 1 60]
  if { $count > 100 } {
    HTTP::respond 429 content "Too Many Requests"
  }
}
```

Sideband-команды

Sideband-команды открывают вспомогательное TCP-соединение, например для запроса к внешнему сервису авторизации. Формы `connect / send / recv / close` и `SIDEBAND::connect / SIDEBAND::send / SIDEBAND::recv / SIDEBAND::close` эквивалентны. Для одного контекста доступно не более четырёх соединений, размер ответа ограничен 8192 байтами.

connect

Открывает sideband-соединение. Целью может быть имя включённого сервиса типа `sideband` или IP-адрес с портом. Для IPv6 используйте форму `[addr]:port`.

Параметр	Значение
Синтаксис	<code>connect [-timeout <ms>] <target></code>
Возвращает	Идентификатор соединения (handle), например <code>sb1</code>
Доступен в	<code>HTTP_REQUEST</code>

send

Отправляет данные в открытое sideband-соединение.

Параметр	Значение
Синтаксис	<code>send <handle> <payload></code>
Возвращает	Число отправленных байт
Доступен в	<code>HTTP_REQUEST</code>

recv

Читает ответ из sideband-соединения. Ответ разбирается как HTTP: доступны статус, тело и отдельные заголовки.

Параметр	Значение
Синтаксис	<code>recv <handle> [status body header <name> raw]</code>
Возвращает	Без подкоманды и с <code>raw</code> — весь ответ; <code>status</code> — код статуса; <code>body</code> — тело; <code>header <name></code> — значение заголовка
Доступен в	<code>HTTP_REQUEST</code>

close

Закрывает sideband-соединение.

Параметр	Значение
Синтаксис	<code>close <handle></code>
Возвращает	—
Доступен в	<code>HTTP_REQUEST</code>

```

when HTTP_REQUEST {
  set sb [connect -timeout 500 auth-service]
  send $sb "GET /check?user=[HTTP::header value X-User] HTTP/1.0\r\n\r\n"
  if { [recv $sb status] != 200 } {
    HTTP::respond 403 content "Denied"
  }
  close $sb
}

```

Команда event

Команда `event` возвращает дополнительные сведения о текущем событии.

event info

Возвращает причину отказа при неудачном подключении к участнику пула.

Параметр	Значение
Синтаксис	<code>event info</code>
Возвращает	Строка с причиной отказа (пустая вне контекста отказа)
Доступен в	<code>LB_FAILED</code>

```
when LB_FAILED {
    log local0. "Отказ подключения: [event info]"
}
```

Команда log

Команда `log` записывает диагностическое сообщение aRules в журнал dataplane.

log

Записывает сообщение в системный журнал.

Параметр	Значение
Синтаксис	<code>log <facility>. <message></code>
Facility	<code>local0</code> , <code>local1</code> , ..., <code>local7</code>
Доступен в	Все события, включая <code>RULE_INIT</code>

```
log local0. "Пропуск проверки для [HTTP::method] [HTTP::host][HTTP::uri] от [IP::client_addr]"
```

Форма с `facility` и точкой совместима с F5 iRules. Значение `facility` принимается, но не влияет на маршрутизацию: сообщения записываются в журнал dataplane с префиксом `[aRule]`.

```
[aRule] Пропуск проверки для GET www.example.com/api/data от 10.1.1.50
```

Встроенные команды Tcl

Помимо расширений aRules, в безопасном интерпретаторе доступны стандартные команды Tcl:

Команда	Описание
<code>set</code>	Присвоение переменной
<code>expr</code>	Арифметические и логические выражения
<code>if / elseif / else</code>	Условное ветвление
<code>switch</code>	Множественный выбор
<code>foreach</code>	Цикл по списку
<code>proc</code>	Определение процедуры
<code>return</code>	Возврат значения
<code>string</code>	Операции со строками (<code>match</code> , <code>tolower</code> , <code>length</code> , <code>range</code> и др.)
<code>regexp</code>	Регулярные выражения
<code>list / lindex / llength</code>	Операции со списками
<code>split / join</code>	Разделение и объединение строк
<code>format / scan</code>	Форматирование строк
<code>append / incr</code>	Добавление к строке / инкремент
<code>catch</code>	Перехват ошибок
<code>info exists</code>	Проверка существования переменной
<code>binary scan</code>	Работа с бинарными данными

Справочник событий aRules

Раздел описывает поддерживаемые события aRules, порядок их вызова и доступный контекст обработки.

Обзор

Код aRules состоит из обработчиков событий соединения. Обработчик объявляется конструкцией `when EVENT { ... }`.

```
when HTTP_REQUEST {
    # код обработчика
}
```

Одно правило aRules может содержать обработчики нескольких событий:

```
when RULE_INIT {
    set ::threshold 1000
}

when HTTP_REQUEST {
    if { [HTTP::header exists Content-Length] } {
        set clen [HTTP::header value Content-Length]
        if { $clen <= $::threshold } {
            ADAPT::enable request false
        }
    }
}
```

В текущей версии выполняются события `RULE_INIT`, `HTTP_REQUEST`, `HTTP_RESPONSE`, `LB_FAILED` и `ICAP_RESPONSE`. Обработчики других событий F5 iRules не выполняются. Исключение составляет `HTTP_REQUEST_DATA`: обработчик распознаётся и проходит проверку, но событие не возникает. Перед переносом правила замените или удалите неподдерживаемые обработчики.

Порядок вызова событий

События возникают по мере обработки HTTP-запроса и ответов сервисов. `LB_FAILED` и `ICAP_RESPONSE` вызываются только при наличии соответствующего сервиса и результата обработки.

Последовательность взаимодействия

Источник	Связь	Назначение
HTTP_REQUEST	Этап	Получен HTTP-запрос клиента
—	Условие	Не удалось подключиться к участнику пула
HTTP_REQUEST	Отказ подключения	LB_FAILED
LB_FAILED	Этап	LB::down / LB::reselect
—	Условие	Получен ответ ICAP-сервера
HTTP_REQUEST	Ответ ICAP	ICAP_RESPONSE
ICAP_RESPONSE	Этап	ICAP::status / ICAP::header
HTTP_REQUEST	Получен HTTP-ответ сервера	HTTP_RESPONSE

1. **HTTP_REQUEST** — при получении и разборе HTTP-запроса клиента.

2. **LB_FAILED** — при неудачном подключении к участнику пула.
3. **ICAP_RESPONSE** — при получении ответа ICAP-сервера (если сервис ICAP участвует в цепочке).
4. **HTTP_RESPONSE** — при получении HTTP-ответа сервера.

Обработчик `RULE_INIT` выполняется перед обработчиком события — см. ниже.

RULE_INIT

Событие `RULE_INIT` подготавливает переменные и процедуры для следующего обработчика.

Описание

Обработчик задаёт глобальные переменные и определяет вспомогательные процедуры. Он выполняется перед каждым обработчиком этого правила aRules в новом Tcl-интерпретаторе. Переменные и процедуры из `RULE_INIT` доступны текущему обработчику, но изменения не сохраняются для следующего события. Для общего состояния используется команда `table`.

Когда вызывается

Непосредственно перед выполнением обработчика события (`HTTP_REQUEST` , `HTTP_RESPONSE` , `LB_FAILED` , `ICAP_RESPONSE`).

HTTP-контекст

Не используйте HTTP-команды в `RULE_INIT` : доступные данные зависят от события, перед которым выполняется инициализация.

Доступные команды

В `RULE_INIT` используйте стандартные команды Tcl и запись в журнал:

Команда	Описание
<code>set</code>	Инициализация глобальных переменных (<code>::var</code>)
<code>proc</code>	Определение процедур
<code>log</code>	Запись в системный журнал
<code>expr</code>	Вычисления
Стандартные Tcl	<code>if</code> , <code>switch</code> , <code>foreach</code> , <code>string</code> , <code>list</code> и др.

Пример

В примере задаётся порог и вспомогательная процедура для последующего обработчика:

```
when RULE_INIT {
  # Значения доступны в каждом обработчике этого aRule
  set ::max_payload_size 1000
  set ::bypass_methods "OPTIONS HEAD"

  # Вспомогательная процедура
  proc should_bypass {method} {
    if { [lsearch $::bypass_methods $method] >= 0 } {
      return 1
    }
    return 0
  }
}
```

Глобальные переменные

Переменные, объявленные с префиксом `::` (например, `::max_size`), являются глобальными в пределах интерпретатора и доступны из процедур. Поскольку интерпретатор создаётся заново для каждого выполнения обработчика, глобальные переменные из `RULE_INIT` — это константы и настройки, а не разделяемое состояние. Для счётчиков и состояния между запросами используется команда `table`.

HTTP_REQUEST

Событие `HTTP_REQUEST` предназначено для анализа и изменения клиентского HTTP-запроса.

Описание

Основное событие обработки. Вызывается при получении HTTP-запроса, после разбора заголовков.

Когда вызывается

После получения и разбора HTTP-запроса от клиента. Вызывается до отправки запроса в сервисную цепочку.

HTTP-контекст

Полный доступ к данным HTTP-запроса: метод, URI, заголовки, IP-адреса, доступная часть тела.

Доступные команды

В обработчике запроса доступны следующие группы команд:

Группа	Команды
HTTP	HTTP::method , HTTP::uri , HTTP::host , HTTP::version , HTTP::header * , HTTP::payload , HTTP::redirect , HTTP::respond
ADAPT	ADAPT::enable , ADAPT::disable
Маршрутизация	pool , node
Sideband	connect , send , recv , close
IP	IP::client_addr , IP::server_addr
Идентификация	IDENT::user , IDENT::groups
Данные	class , table
log	log
Tcl	Разрешённые стандартные команды

Пример

Пример записывает сведения о запросе в журнал, добавляет заголовок и при необходимости отключает ICAP:

```
when HTTP_REQUEST {
  # Запись POST-запросов в журнал
  if { [HTTP::method] eq "POST" || [HTTP::method] eq "PUT" } {
    set clen "unknown"
    if { [HTTP::header exists Content-Length] } {
      set clen [HTTP::header value Content-Length]
    }
    log local0. "[IP::client_addr] [HTTP::method] [HTTP::host][HTTP::uri] CL=$clen"
  }

  # Добавление заголовка
  HTTP::header insert X-Forwarded-By "AMIBIG"

  # Условное исключение из DLP-проверки
  if { [string match "/api/internal/*" [HTTP::uri]] } {
    ADAPT::enable request false
    ADAPT::enable response false
  }
}
```

HTTP_REQUEST_DATA

Событие `HTTP_REQUEST_DATA` сохранено для синтаксической совместимости с F5 iRules, но не вызывается в текущей версии.

Описание

Событие сбора тела запроса. По модели F5 iRules наступает, когда тело HTTP-запроса, запрошенное командой `HTTP::collect` в обработчике `HTTP_REQUEST`, собрано и доступно для чтения командой `HTTP::payload`.

Когда вызывается

В текущей версии событие не вызывается. Обработчик `when HTTP_REQUEST_DATA { ... }` распознаётся и проходит проверку, но dataplane не дочитывает тело отдельными сегментами. Перенесите необходимую логику в `HTTP_REQUEST` и учитывайте, что `HTTP::payload` возвращает только часть тела, уже находящуюся в буфере.

HTTP-контекст

Как в `HTTP_REQUEST`: данные запроса и собранная часть тела (`HTTP::payload`).

Пример

Следующий перенесённый обработчик проходит проверку, но блок `HTTP_REQUEST_DATA` не выполняется:

```
when HTTP_REQUEST {
    if { [HTTP::method] eq "POST" } {
        HTTP::collect 1024
    }
}

when HTTP_REQUEST_DATA {
    # В текущей версии обработчик не вызывается; тело в пределах
    # доступной части читают в HTTP_REQUEST через HTTP::payload
    log local0. "Собрано [HTTP::payload length] байт тела запроса"
}
```

HTTP_RESPONSE

Событие `HTTP_RESPONSE` предназначено для анализа и изменения HTTP-ответа перед отправкой клиенту.

Описание

Вызывается при получении HTTP-ответа от сервера (через сервисную цепочку).

Когда вызывается

После того как ответ получен от сервера и прошёл через сервисную цепочку, до отправки клиенту.

HTTP-контекст

Доступны код статуса, заголовки и находящаяся в буфере часть тела ответа. Метод, URI и заголовок Host исходного запроса в этом событии недоступны.

Доступные команды

В обработчике ответа доступны следующие группы команд:

Группа	Команды
HTTP	<code>HTTP::status</code> , <code>HTTP::header *</code> (заголовки ответа), <code>HTTP::payload</code> , <code>HTTP::respond</code>
LB	<code>LB::status</code> , <code>LB::down</code>
IP	<code>IP::client_addr</code> , <code>IP::server_addr</code>
Идентификация	<code>IDENT::user</code> , <code>IDENT::groups</code>
Данные	<code>class</code> , <code>table</code>
log	<code>log</code>
Tcl	Разрешённые стандартные команды

Пример

Пример записывает серверные ошибки в журнал и удаляет служебные заголовки:

```
when HTTP_RESPONSE {
    # Запись ошибок сервера в журнал
    if { [HTTP::status] >= 500 } {
        log local0. "Ошибка [HTTP::status] от [IP::server_addr]"
    }

    # Удаление серверных заголовков
    HTTP::header remove Server
    HTTP::header remove X-Powered-By
}
```

LB_FAILED

Событие `LB_FAILED` позволяет обработать отказ подключения к участнику пула.

Описание

Вызывается, когда не удалось подключиться к выбранному участнику пула сервиса. Обработчик может пометить участника недоступным, отсоединить соединение или запросить повторный выбор, в том числе из другого пула.

Когда вызывается

При отказе или истечении времени ожидания подключения к участнику пула.

HTTP-контекст

Данные HTTP-сообщения недоступны. Доступны адреса соединения и сведения о неудачном выборе.

Доступные команды

Для обработки отказа доступны команды управления балансировкой и общие команды контекста:

Группа	Команды
LB	LB::status , LB::down , LB::detach , LB::reselect
Событие	event info
IP	IP::client_addr , IP::server_addr
Идентификация	IDENT::user , IDENT::groups
Данные	class , table
log	log
Tcl	Разрешённые стандартные команды

Пример

Пример помечает текущего участника недоступным и запрашивает повторный выбор:

```
when LB_FAILED {
    log local0. "Отказ участника пула: [event info]"
    LB::down
    LB::reselect
}
```

ICAP_RESPONSE

Событие `ICAP_RESPONSE` предоставляет результат обработки сообщения ICAP-сервером.

Описание

Вызывается при получении ответа ICAP-сервера в режимах REQMOD или RESPMOD. Обработчик читает код статуса и заголовки ICAP, например для записи вердикта DLP в журнал.

Когда вызывается

После получения ответа ICAP-сервера, до продолжения обработки сообщения.

HTTP-контекст

Доступны код статуса и заголовки ответа ICAP-сервера (`ICAP::status` , `ICAP::header`), а также адреса соединения.

Доступные команды

Для анализа ответа ICAP доступны следующие группы команд:

Группа	Команды
ICAP	<code>ICAP::status</code> , <code>ICAP::header</code>
IP	<code>IP::client_addr</code> , <code>IP::server_addr</code>
Идентификация	<code>IDENT::user</code> , <code>IDENT::groups</code>
Данные	<code>class</code> , <code>table</code>
log	<code>log</code>
Tcl	Разрешённые стандартные команды

Пример

Пример записывает нестандартный ответ ICAP и заголовок с вердиктом в журнал:

```
when ICAP_RESPONSE {
  if { [ICAP::status] != 204 } {
    log local0. "ICAP-вердикт [ICAP::status] для [IP::client_addr]"
    if { [ICAP::header exists X-Infection-Found] } {
      log local0. "DLP: [ICAP::header X-Infection-Found]"
    }
  }
}
```

Сводная таблица

Таблица показывает назначение и основной контекст каждого поддерживаемого события:

Событие	Когда	Контекст	Ключевые команды
RULE_INIT	Перед каждым обработчиком события	Нет	set , proc , log
HTTP_REQUEST	При получении HTTP-запроса	Запрос	HTTP::method , HTTP::uri , HTTP::host , HTTP::header , HTTP::redirect , HTTP::respond , ADAPT::enable , pool , node , connect / send / recv / close , IP::* , IDENT::* , log
HTTP_REQUEST_DATA	Не вызывается в текущей версии (сбор тела после HTTP::collect)	Запрос	HTTP::payload , log
HTTP_RESPONSE	При получении HTTP-ответа	Ответ	HTTP::status , HTTP::header , HTTP::respond , LB::status , LB::down , IP::* , IDENT::* , log
LB_FAILED	При отказе подключения к участнику пула	Соединение	LB::down , LB::detach , LB::reselect , event info , IDENT::* , log
ICAP_RESPONSE	При получении ответа ICAP-сервера	Ответ ICAP	ICAP::status , ICAP::header , IDENT::* , log

Примеры aRules

Примеры показывают типовые способы обработки HTTP-трафика и ограничения, которые необходимо учитывать перед применением aRules.

Исключение малых запросов из DLP-проверки

Правило отключает ICAP-адаптацию, если значение Content-Length не превышает заданный порог. Заголовок передаёт клиент и может не соответствовать

фактическому размеру тела, поэтому применяйте этот пример только там, где риск обхода DLP принят политикой безопасности.

```
when RULE_INIT {
    set ::max_payload_size 1000
}

when HTTP_REQUEST {
    if { [HTTP::method] eq "PUT" || [HTTP::method] eq "POST" } {
        if { [HTTP::header exists Content-Length] } {
            set clen [HTTP::header value Content-Length]
            if { $clen <= $::max_payload_size } {
                ADAPT::enable request false
                ADAPT::enable response false
                log local0. "DLP-проверка отключена: $clen байт, клиент [IP::client_addr]"
            }
        }
    }
}
```

Привязка: к DLP-сервису в сервисной цепочке.

```
{
  "name": "dlp-service",
  "type": "icap",
  "arules": ["size_limit"]
}
```

Передача имени пользователя в заголовке

Для стандартной передачи имени используйте параметр политики `insert-identity-headers`. `aRules` требуется, если принимающий сервис ожидает другое имя заголовка. Команда `IDENT::user` возвращает пустую строку для неопознанного пользователя.

```
when HTTP_REQUEST {
    set user [IDENT::user]
    if { $user ne "" } {
        HTTP::header insert X-Corporate-User $user
        log local0. "Добавлено имя пользователя для [HTTP::method] [HTTP::uri]"
    }
}
```

Привязка: к правилу политики.

```
{
  "name": "inspect-web",
  "match": {
    "dest-port": [443],
    "protocol": "tcp"
  },
  "action": "intercept",
  "service-chain": "main-chain",
  "arules": ["add_corporate_user"]
}
```


Удаление служебных заголовков

Правило удаляет внутренние заголовки перед передачей запроса в сервисную цепочку. Для `X-Authenticated-User` и `X-Client-IP` Амибиг выполняет такую очистку автоматически при включённой подсистеме идентификации.

```
when HTTP_REQUEST {
  if { [HTTP::header exists X-Authenticated-User] } {
    HTTP::header remove X-Authenticated-User
    log local0. "Удалён X-Authenticated-User из [HTTP::method] [HTTP::uri]"
  }
  if { [HTTP::header exists X-Internal-Token] } {
    HTTP::header remove X-Internal-Token
  }
}
```

URL-редирект

Перенаправление клиентов со старого домена на новый с сохранением URI.

```
when HTTP_REQUEST {
  if { [HTTP::host] eq "old.example.com" } {
    HTTP::redirect "https://new.example.com[HTTP::uri]"
    log local0. "Перенаправление old.example.com на new.example.com"
  }
}
```

Блокировка запросов по URI

Блокировка доступа к определённым путям с возвратом HTTP 403.

```
when HTTP_REQUEST {
  set blocked_paths [list "/admin" "/debug" "/internal"]
  foreach path $blocked_paths {
    if { [string match "${path}*" [HTTP::uri]] } {
      HTTP::respond 403 content "Access Denied"
      log local0. "Заблокирован доступ к [HTTP::uri] для [IP::client_addr]"
      return
    }
  }
}
```

Исключение URI из ICAP-обработки

Отключение ICAP-адаптации для внутренних API-запросов, которые не требуют DLP-инспекции.

```
when HTTP_REQUEST {
  if { [string match "/api/internal/*" [HTTP::uri]] } {
    ADAPT::enable request false
    ADAPT::enable response false
    log local0. "ICAP-обработка отключена для [HTTP::uri]"
  }
}
```

Логирование POST-запросов

Правило записывает в журнал адрес клиента, URI и заявленный размер тела для запросов POST и PUT.

```
when HTTP_REQUEST {
  if { [HTTP::method] eq "POST" || [HTTP::method] eq "PUT" } {
    set clen "unknown"
    if { [HTTP::header exists Content-Length] } {
      set clen [HTTP::header value Content-Length]
    }
    log local0. "[IP::client_addr] [HTTP::method] [HTTP::host][HTTP::uri] CL=$clen"
  }
}
```

Замена заголовка Host

Правило заменяет заголовок `Host` перед передачей запроса внутреннему серверу.

```
when HTTP_REQUEST {
  if { [HTTP::host] eq "public-api.example.com" } {
    HTTP::header replace Host "backend-api.internal.corp"
    log local0. "Host rewrite: public-api -> backend-api"
  }
}
```

Отключение ICAP для выбранных методов

Правило отключает ICAP для методов OPTIONS и HEAD. Применяйте такое исключение только в соответствии с принятой политикой: наличие тела и его обработка зависят от клиента и сервера.

```
when RULE_INIT {
  set ::safe_methods [list "OPTIONS" "HEAD"]
}

when HTTP_REQUEST {
  if { [lsearch $::safe_methods [HTTP::method]] >= 0 } {
    ADAPT::enable request false
    ADAPT::enable response false
    log local0. "ICAP отключён для метода [HTTP::method]"
  }
}
```

Инспекция тела запроса

Пример ищет признаки нежелательного содержимого в доступной части тела POST-запроса. `HTTP::payload` возвращает только байты, уже полученные вместе с заголовками, поэтому такое правило не заменяет DLP, WAF или антивирусную проверку.

```
when HTTP_REQUEST {
  if { [HTTP::method] eq "POST" && [HTTP::payload length] > 0 } {
    set body [HTTP::payload]
    set body_len [HTTP::payload length]
  }
}
```

```

# Поиск диагностических признаков
if { [regexp {<script[^>]*>} $body] } {
    log local0. "XSS pattern detected in POST from [IP::client_addr] to [HTTP::uri]"
}

if { [string match "*DROP TABLE*" $body] || [string match "*/; DELETE *" $body] } {
    log local0. "SQL injection pattern from [IP::client_addr] to [HTTP::uri]"
}

log local0. "Body inspection: $body_len bytes, [HTTP::method] [HTTP::uri]"
}
}

```

Ограничение частоты запросов по клиенту

Команда `table` подсчитывает запросы клиента и возвращает ответ 429 после превышения порога. Счётчик локален для рабочего потока dataplane и не синхронизируется между рабочими потоками или узлами НА, поэтому пример не обеспечивает точного общего ограничения скорости.

```

when HTTP_REQUEST {
    set count [table incr "req:[IP::client_addr]" 1 60]
    if { $count > 100 } {
        HTTP::respond 429 content "Too Many Requests"
        log local0. "Rate limit: [IP::client_addr] ($count запросов за 60 с)"
    }
}

```

Обработка отказа участника пула

Правило помечает недоступного участника пула и запрашивает повторный выбор после неудачного подключения.

```

when LB_FAILED {
    log local0. "Отказ подключения к участнику пула: [event info]"
    LB::down
    LB::reselct
}

```

Блокировка по data-group

Проверка узла назначения по именованному набору записей (data-group), например импортированному из UCS-конфигурации F5.

```

when HTTP_REQUEST {
    if { [class match [HTTP::host] equals blocked_hosts] } {
        HTTP::respond 403 content "Blocked"
        log local0. "Blocked host [HTTP::host] for [IP::client_addr]"
    }
}

```

Добавление зоны по префиксу адреса клиента

Пример добавляет имя зоны по текстовому префиксу IPv4-адреса. Это не проверка CIDR и не подходит для произвольных масок подсети.

```
when HTTP_REQUEST {
  set client [IP::client_addr]

  if { [string match "10.10.*" $client] } {
    HTTP::header insert X-Security-Zone "trusted"
  } elseif { [string match "10.20.*" $client] } {
    HTTP::header insert X-Security-Zone "dmz"
  } else {
    HTTP::header insert X-Security-Zone "external"
  }
}
```

Комплексное правило с несколькими действиями

Пример объединяет запись в журнал, добавление заголовков и условное управление ICAP.

```
when RULE_INIT {
  set ::max_body 2048
  set ::internal_nets [list "10.0.*" "172.16.*" "192.168.*"]
}

when HTTP_REQUEST {
  set client [IP::client_addr]
  set is_internal 0

  # Определение: внутренний или внешний клиент
  foreach net $::internal_nets {
    if { [string match $net $client] } {
      set is_internal 1
      break
    }
  }

  # Добавление заголовков
  HTTP::header insert X-Forwarded-By "AMIBIG"
  HTTP::header insert X-Request-Time [clock seconds]

  if { $is_internal } {
    HTTP::header insert X-Client-Zone "internal"
  } else {
    HTTP::header insert X-Client-Zone "external"
  }

  # Исключение малых POST-запросов внутренних клиентов из DLP-проверки
  if { $is_internal && ([HTTP::method] eq "POST" || [HTTP::method] eq "PUT") } {
    if { [HTTP::header exists Content-Length] } {
      set clen [HTTP::header value Content-Length]
      if { $clen <= $::max_body } {
        ADAPT::enable request false
        log local0. "DLP-проверка отключена для внутреннего POST: $clen байт"
      }
    }
  }

  # Запись в журнал
  log local0. "$client [HTTP::method] [HTTP::host][HTTP::uri] zone=[expr {$is_internal ? "interna
```

```
l" : "external"]]"
}
```

Конфигурация aRules в JSON

Ниже приведён пример секции с несколькими правилами aRules:

```
{
  "arules": {
    "arule": [
      {
        "name": "size_limit",
        "description": "Исключение малых PUT/POST из DLP-проверки",
        "enabled": true,
        "code": "when RULE_INIT {\n    set ::max_payload_size 1000\n}\n\nwhen HTTP_REQUEST {\n
if { [HTTP::method] eq \"PUT\" || [HTTP::method] eq \"POST\" }
{\n    if { [HTTP::header exists Content-Length] } {\n        set clen [HTTP::header value
Content-Length]\n        if { $clen <= $::max_payload_size } {\n            ADAPT::enable
request false\n            ADAPT::enable response false\n            log local0. \"DLP-
проверка отключена: $clen байт, клиент [IP::client_addr]\"\n        }\n    }\n}"
      },
      {
        "name": "add_corporate_user",
        "description": "Передача имени пользователя в заголовке",
        "enabled": true,
        "code": "when HTTP_REQUEST {\n    set user [IDENT::user]\n    if { $user ne \"\" }
{\n        HTTP::header insert X-Corporate-User $user\n        log local0. \"Добавлено имя
пользователя для [HTTP::method] [HTTP::uri]\"\n    }\n}"
      },
      {
        "name": "remove_auth",
        "description": "Удаление заголовка аутентификации",
        "enabled": true,
        "code": "when HTTP_REQUEST {\n    if { [HTTP::header exists X-Authenticated-User] }
{\n        HTTP::header remove X-Authenticated-User\n        log local0. \"Удалён X-Authenticated-
User\"\n    }\n}"
      }
    ]
  }
}
```

Топология

В этом разделе администратор настраивает слушатель и выбирает схему развёртывания Амибиг. Топология связывает слушатель (`listener`), режим подсистемы обработки трафика, политику и подготовленные сетевые объекты.

Шаблоны позволяют быстро подготовить демонстрационный стенд или основу типового внедрения. Перед применением конфигурации проверьте созданные объекты в разделах сети, сервисов, сервисных цепочек и политик.

Прозрачная топология

Секция `topology` определяет, где подсистема обработки трафика принимает пользовательские соединения. После приёма Амибиг передаёт их в подсистему политик, которая выбирает действие и сервисную цепочку. Топология настраивается на экране **Топология**.

В текущей реализации поддерживаются два прозрачных режима: `transparent-l2` для включения в разрыв без изменения маршрутизации и `transparent-l3` для приёма трафика через маршрутизируемый путь.

Архитектурные ограничения

Для слушателя (`topology.listener`) задаётся ровно одна привязка подсистемы обработки трафика: тегированный VLAN или прямой нетегированный интерфейс.

Из этого следуют требования:

- `listener.vlan` ссылается на тегированный VLAN, привязанный к DPDK-порту или агрегированному каналу из DPDK-портов;
- `listener.interface` ссылается непосредственно на порт обработки трафика или агрегированный канал, если входящий трафик не тегирован;
- `listener.address`, если он отличается от `0.0.0.0` и `::`, должен принадлежать подсети Self-IP на выбранном VLAN или интерфейсе;
- слушатель нельзя разместить на интерфейсе управления или HA-интерфейсе.

Поля

Основные параметры топологии находятся в секции `topology`.

Поле	Назначение
<code>name</code>	имя профиля топологии

Поле	Назначение
<code>mode</code>	<code>transparent-l2</code> или <code>transparent-l3</code>
<code>dataplane-mode</code>	Используемая реализация подсистемы обработки трафика (<code>dpdk</code>)
<code>listener.vlan</code>	тегированный VLAN входящего трафика; взаимоисключающий с <code>listener.interface</code>
<code>listener.interface</code>	порт или агрегированный канал для нетегированного входящего трафика; взаимоисключающий с <code>listener.vlan</code>
<code>listener.address</code>	Любой адрес или конкретный адрес в подсети Self-IP
<code>listener.port</code>	порт или <code>0</code> для всех портов
<code>listener.protocol</code>	<code>tcp</code> , <code>udp</code> или <code>any</code>

Для UDP действует ограничение: в текущей версии полный путь инспекции и применения сервисной цепочки рассчитан прежде всего на TCP. Значение `udp` в поле `listener.protocol` само по себе не включает все функции TCP-инспекции для UDP.

Параметры DPDK не входят в `topology` . Маска рабочих потоков (`lcore-mask`), размер `hugepages` и путь к конфигурации `dataplane` находятся в отдельной секции верхнего уровня `dataplane.dpdk` . В веб-интерфейсе эти параметры находятся на экране **Система**, а не на экране **Топология**. См. Производительность.

Общие параметры политики

Помимо параметров слушателя, на экране **Топология** находятся три переключателя, которые действуют на весь трафик независимо от правил политики:

- «**Блокировка QUIC**» (`reject-quic`) — отбрасывать пакеты QUIC Initial, вынуждая браузеры вернуться на HTTPS поверх TCP;
- «**UDP через сервисную цепочку**» (`udp-chain`) — проводить транзитный UDP через inline-звенья сервисной цепочки;
- «**ICMP через сервисную цепочку**» (`icmp-chain`) — проводить транзитный ICMP через inline-звенья сервисной цепочки.

В конфигурации эти параметры хранятся в секции `policy` , а не `topology` . Подробнее: Обзор политик.

Режимы

Режим топологии выбирается по способу включения Амибиг в сеть: в разрыв канала второго уровня либо в маршрутизируемый путь.

transparent-l2

Амибиг включается в существующий сегмент в разрыв линии без изменения маршрутизации L3. Трафик принимается в тегированном VLAN или непосредственно на нетегированном интерфейсе обработки трафика, проходит через подсистему политик и назначенную сервисную цепочку. Режим подходит для пилотных и промышленных внедрений, в которых маршрутизация должна остаться неизменной.

transparent-l3

В этом режиме трафик поступает в подсистему обработки через маршрутизацию. Используется в виртуальных стендах, демонстрационных и лабораторных схемах и в установках, где L3-путь явно направляет поток на Амибиг.

Пример

Слушатель задаётся в секции `topology`, а параметры подсистемы обработки трафика — в отдельной секции `dataplane`:

```
{
  "topology": {
    "name": "demo-topology",
    "mode": "transparent-l3",
    "dataplane-mode": "dpdk",
    "listener": {
      "vlan": "client",
      "address": "0.0.0.0",
      "port": 0,
      "protocol": "any"
    }
  },
  "dataplane": {
    "dpdk": {
      "config-path": "/config/dataplane.conf",
      "hugepages-mb": 2048,
      "lcore-mask": "0xfe"
    }
  }
}
```

Связь с остальной конфигурацией

Диаграмма ниже показывает порядок обработки: топология определяет слушатель, а дальнейшие решения принимает подсистема политики.

Схема взаимодействия

Источник	Связь	Назначение
Клиентский трафик	→	Слушатель
Слушатель	→	Подсистема политики
Подсистема политики	→	Решение TLS
Решение TLS	→	Цепочка сервисов
Цепочка сервисов	→	Сервер назначения

Подсистема политики решает:

- передача без расшифрования (`bypass`) или TLS-инспекция (`intercept`);
- выбор цепочки сервисов;
- поведение TLS;
- выбор конкретных участников пула.

Шаблоны развёртывания

Для первоначальной настройки используйте шаблоны развёртывания. Мастер создаёт основу топологии и подбирает VLAN и порты для слушателя и сервисного пути.

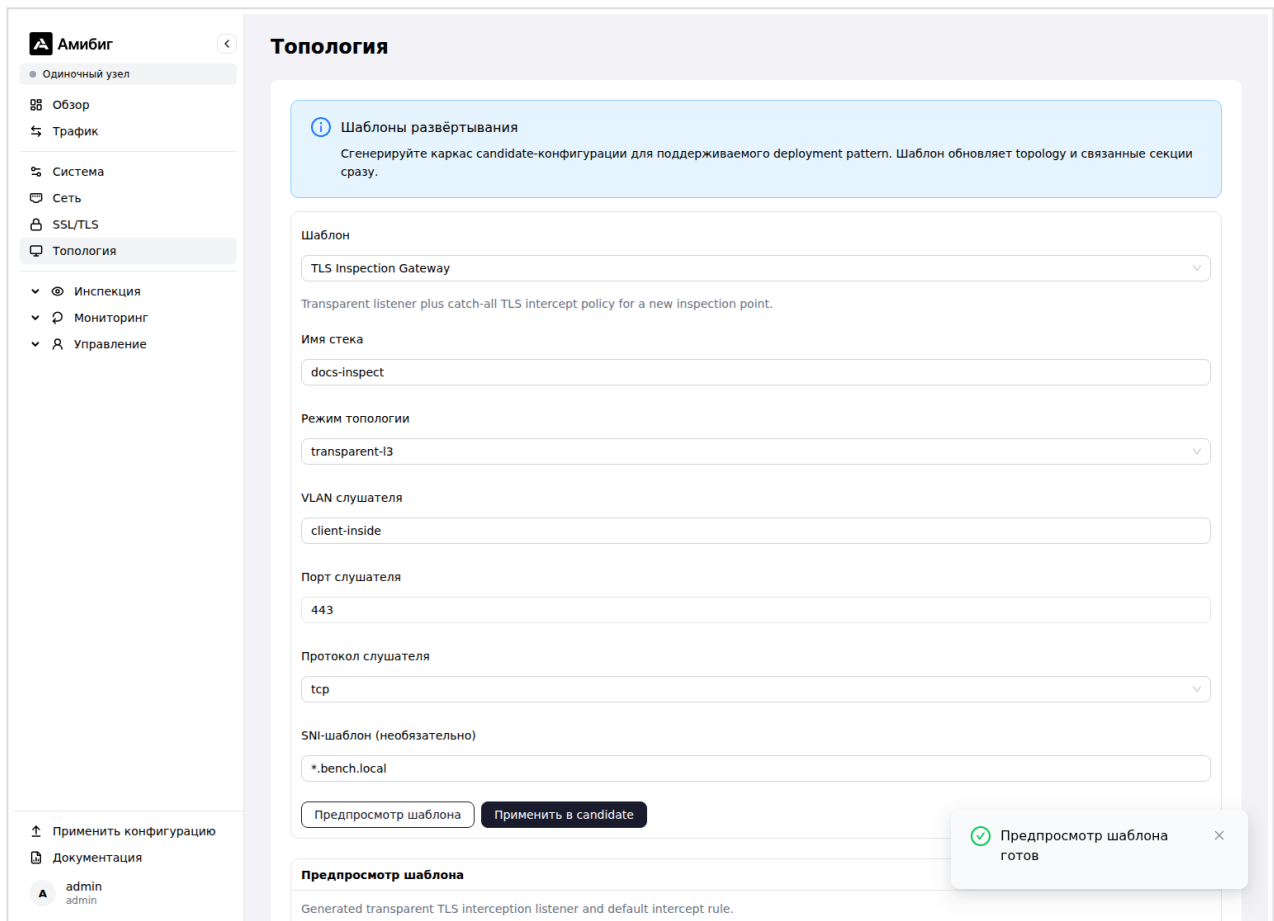
Рекомендации

При выборе режима топологии соблюдайте следующие рекомендации:

- Для виртуальных стендов и демонстраций используйте `transparent-l3`.
- Для пилотных внедрений в разрыв без изменения маршрутизации используйте `transparent-l2`.
- Перед применением конфигурации проверяйте путь потока через Объяснение потока.
- Если задаёте конкретный `listener.address`, убедитесь, что нужный адрес Self-IP подсистемы обработки трафика уже существует.

Шаблоны развёртывания

Мастер шаблонов на экране **Топология** создаёт согласованный набор сетевых объектов, сервисов, цепочек и правил. Используйте его для первоначальной настройки стенда, а затем проверьте и дополните подготовленную (`candidate`) конфигурацию. Сам мастер не применяет конфигурацию к устройству.



Когда использовать шаблон

Шаблон сокращает первоначальную настройку в следующих сценариях:

- пилотный проект или демонстрационный стенд;
- начальная конфигурация для новой инсталляции;
- подготовка каркаса топологии перед ручной донастройкой;
- воспроизводимая сборка типовых сценариев для лабораторных испытаний.

Какие объекты создаёт шаблон

После заполнения параметров веб-интерфейс:

- показывает предпросмотр будущих изменений (кнопка `Предпросмотр шаблона`);
- перечисляет затронутые секции конфигурации;
- показывает сгенерированные объекты по секциям;
- добавляет объекты в подготовленную (`candidate`) конфигурацию после нажатия **Применить в candidate**.

Мастер также создаёт или обновляет:

- имя политики `policy.name`;

- правило по умолчанию `policy.default-rule` для трафика, который не соответствует другим правилам;
- привязку `topology.policy-ref`, чтобы связь топологии и политики была видна сразу.

После создания объектов проверьте различия, укажите недостающие параметры и только затем перенесите подготовленную (`candidate`) конфигурацию в действующую (`running`).

Поддерживаемые шаблоны

Точный список доступных шаблонов возвращает REST API `GET /api/v1/config/topology-templates` и показывает веб-интерфейс.

Шаблон	Назначение
<code>tls-inspection-gateway</code>	Прозрачный шлюз TLS-инспекции со слушателем и правилом <code>intercept</code>
<code>inline-ngfw-chain</code>	Inline-сервис NGFW с цепочкой сервисов
<code>icap-dlp-chain</code>	Каркас под ICAP/DLP-инспекцию
<code>reverse-proxy-vip</code>	Основа виртуального сервера для обратного прокси

Создание конфигурации по шаблону

Предварительный просмотр показывает изменения до записи в подготовленную конфигурацию.

1. Выберите шаблон.
2. Заполните обязательные поля.
3. Нажмите `Предпросмотр шаблона`.
4. Проверьте:
 - какие секции будут затронуты;
 - какие объекты создаются;
 - какие следующие шаги предлагает интерфейс.
5. Если структура подходит, нажмите **Применить в candidate**.

После создания объектов:

1. Откройте список изменений и проверьте секции `network`, `services`, `chains`, `topology`.
2. Укажите адреса, шлюзы, сертификаты и мониторы доступности.
3. Проверьте путь трафика через Объяснение потока.
4. Примените конфигурацию и проверьте созданные объекты.

REST API

Получение списка шаблонов:

```
GET /api/v1/config/topology-templates
```

Предварительный просмотр и запись в подготовленную конфигурацию:

```
POST /api/v1/config/topology-templates/compose
```

Метод возвращает перечень затронутых секций, список сгенерированных объектов и предварительный просмотр. Переход от подготовленной (`candidate`) к действующей (`running`) конфигурации выполняется отдельно через штатную процедуру применения.

Ограничения

Перед применением созданной конфигурации учитывайте следующие ограничения:

- Мастер создаёт каркас, а не готовую к промышленной эксплуатации конфигурацию: реальные адреса, шлюзы, сертификаты, мониторы доступности и детальные правила политики обычно требуют ручной донастройки.
- Повторное применение шаблона рассчитано на стабильные имена объектов; конфликты именования лучше разрешать заранее.

Высокая доступность

В этом разделе администратор настраивает кластер высокой доступности из двух узлов Амибиг: активного и резервного. При отказе активного узла кластер передаёт резервному узлу виртуальные IP-адреса (VIP) и общую конфигурацию. Состояние TCP-соединений между узлами не передаётся, поэтому клиенты подключаются заново.

До настройки кластера подготовьте одинаковую сетевую схему на обоих узлах, выделенный HA-канал и единый источник синхронизации времени.

Обзор высокой доступности

Амибиг устанавливается в разрыв сетевого трафика, поэтому отказ единственного устройства может прервать обработку соединений или лишить средства защиты расшифрованного трафика. Кластер высокой доступности (HA) из двух узлов сокращает перерыв: при отказе активного узла виртуальные IP-адреса переходят на резервный.

Активный и резервный узлы

Диаграмма показывает распределение ролей и HA-канал между узлами.

Схема взаимодействия

Источник	Связь	Назначение
Виртуальный IP-адрес	→	Узел А: трафик обрабатывается
Узел А: трафик обрабатывается	↔ HA-канал с mTLS	Узел В: пассивный режим
Активный узел	Группа	Узел А: трафик обрабатывается
Резервный узел	Группа	Узел В: пассивный режим

Кластер Амибиг состоит из двух узлов:

- **активный узел** владеет VIP и обрабатывает клиентский трафик;
- **резервный узел** получает конфигурацию от HA-партнёра, а при режиме `session-ticket-mode = ha-sync` — также ключи TLS-сессий. Служба `tforged` на нём запущена, но не владеет VIP и не принимает соединения для TLS-инспекции (`role.txt=backup`, файл `vip-state.txt` пуст).

	Узел А, активный	Узел В, резервный
VIP	10.1.0.1	—
Адрес узла	10.1.0.2	10.1.0.3
<code>tforge</code>	Запущен, владеет VIP	Запущен, не владеет VIP
HA-канал	Контроль связи и синхронизация конфигурации через mTLS с TLS 1.3; STEK передаётся только в режиме <code>ha-sync</code>	Контроль связи и синхронизация конфигурации через mTLS с TLS 1.3; STEK принимается только в режиме <code>ha-sync</code>

Оба узла подключаются к клиентской, сервисной и серверной сетям по одинаковой схеме.

В каждый момент времени пользовательский трафик обрабатывает только один узел. На резервном узле `tforge` уже запущен, поэтому при переключении достаточно передать ему VIP и обновить таблицы соседних сетевых устройств.

Резервный узел не принимает пользовательский трафик: его слушатели и подключения ICAP остаются в пассивном режиме до переключения. Мониторы доступности и синхронизация конфигурации продолжают работать.

Проверка роли в веб-интерфейсе

Роль текущего узла постоянно отображается в боковой панели: **Активный узел**, **Резервный узел** или **Одиночный узел**. Если роль определить не удалось, индикатор становится жёлтым. Значение обновляется по данным `GET /api/v1/ha/status` без перезагрузки страницы.

Роль также отображается в блоке состояния на экране **Обзор**.

Автоматическое переключение на резервный узел

Кластер переключает роль после потери связи с активным узлом или при недоступности критичных сервисов. Решение зависит от состояния HA-партнёра и оценки работоспособности обоих узлов.

Обнаружение отказа

Узлы обмениваются контрольными сообщениями по выделенному HA-каналу. Если сообщение от HA-партнёра не получено в течение `dead-interval`, который по умолчанию равен 3 секундам, партнёр считается недоступным. Система также контролирует локальные сервисы. Недоступность критичных компонентов на активном узле может запустить переключение.

Оценка работоспособности узла (HealthScore)

Отказ отдельного участника пула не всегда должен приводить к переключению. Узел рассчитывает оценку работоспособности по весам сервисов: оценка равна сумме весов сервисов в состоянии UP , делённой на сумму весов участвующих сервисов. Вес (weight), минимальное число доступных участников (min-healthy-members) и поведение при деградации (on-degrade) задаются в настройках каждого сервиса; глобальный порог оценки (minimum-healthy-percent , по умолчанию 50 %) — на странице **Мониторинг → НА**. Состояние участников пула определяют мониторы доступности.

Если число доступных участников сервиса ниже заданного минимума, сервис либо уменьшает общую оценку (on-degrade = drop), либо исключается из расчёта (on-degrade = bypass). Когда локальная оценка ниже порога, роль передаётся НА-партнёру только в том случае, если его оценка выше на значение failover-bonus . Такой гистерезис предотвращает частую смену ролей. Параметр immediate-fail позволяет переключиться при недоступности критичного сервиса без учёта failover-bonus , если НА-партнёр доступен и готов принять активную роль.

Последовательность переключения

После обнаружения отказа резервный узел выполняет следующие действия:

Шаг	Действие
1	Система обнаруживает потерю активного узла и выбирает новый активный узел
2	ha-daemon записывает role.txt=master и обновляет vip-state.txt
3	Рабочий поток vip_anchor включает intercept , выполняет vip_assign и рассылает GARP
4	Сетевые устройства обновляют таблицы соседей и CAM/FDB для нового MAC-адреса
5	Новый активный узел принимает новые соединения через уже запущенный tforge
6	Клиенты повторно устанавливают TCP-соединения; в режиме ha-sync общий STEK позволяет возобновить TLS-сессии

При значениях по умолчанию потеря НА-партнёра обнаруживается через 3 секунды. Полное время переключения зависит также от обновления сетевых таблиц на соседних устройствах.

Поведение соединений при переключении

Результат зависит от протокола и режима обработки соединения.

Тип соединения	Результат
Существующие TCP-соединения	Разрываются; клиент устанавливает новое соединение к тому же VIP
TLS-сессии	В режиме <code>ha-sync</code> после повторного подключения возможно возобновление с общим STEK; состояние соединения между узлами не переносится
Потоки UDP/ICMP с действиями <code>bypass</code> , <code>passthrough</code> , <code>reject</code>	Поток создаётся заново после первого пакета клиента
Выполняющиеся ICAP-запросы	Прерываются; после повторного подключения создаётся новый запрос
Передача через inline-сервис	Для TCP обычно требуется повторное подключение; передача без потерь не гарантируется

Возврат роли после восстановления

Восстановленный узел не получает VIP обратно автоматически. Чтобы вернуть ему активную роль, на текущем активном узле выберите **Мониторинг → НА → Передать active партнёру**. Ручная передача роли вызывает ещё одно кратковременное прерывание соединений.

Что синхронизируется между узлами

До переключения кластер передаёт резервному узлу следующие данные.

Данные	Частота
Конфигурация	После применения изменений
Ключи TLS-сессий (STEK)	Каждые 5 мин, только при <code>session-ticket-mode = ha-sync</code>

Сертификаты УЦ для НА автоматически не синхронизируются. При первоначальной настройке администратор вручную передаёт сертификат УЦ каждого узла НА-партнёру.

Следующие данные не синхронизируются и создаются заново после переключения:

- состояние TCP-соединений;
- состояние мониторов доступности;

- ICAP-сессии;
- пакеты, обработка которых не завершилась.

Какая часть настроек сервисов переносится на партнёра как есть, а какая задаётся на каждом узле отдельно, определяет режим адресации сервисов (`service-addressing-mode`) — см. Настройка отказоустойчивого кластера.

Мониторинг состояния кластера

Состояние кластера доступно через REST API (`GET /api/v1/ha/status`) и веб-интерфейс:

Параметр	Описание
Обработка трафика	Узел, который владеет VIP и обрабатывает трафик
Приоритет выборов	Узел с наивысшим приоритетом
Состояние партнёра	Состояние связи с HA-партнёром: Доступен , Недоступен или Подключение
Шифрование канала	Состояние mTLS на HA-канале
Сообщения (отпр/принят)	Счётчики отправленных и полученных сообщений
Последний heartbeat	Время последнего контрольного сообщения от HA-партнёра
Последняя синхронизация	Время последней синхронизации конфигурации

Смена роли, подключение и отключение HA-партнёра, истечение тайм-аута контрольных сообщений, синхронизация конфигурации и ключей регистрируются в системном журнале. Записи доступны на экране **Просмотр журналов**.

Границы отказоустойчивости

Амибиг передаёт резервному узлу VIP и заранее синхронизированные данные, но не переносит состояние установленных TCP-соединений. Кластер обеспечивает:

- синхронизацию конфигурации;
- синхронизацию STEK и возобновление TLS-сессий после переключения, если задано `session-ticket-mode = ha-sync` ;
- передачу VIP между активным и резервным узлами;
- отсутствие автоматического возврата роли после восстановления узла.

При переключении существующие TCP-соединения разрываются, после чего клиенты подключаются заново. При режиме `ha-sync` синхронизация STEK позволяет возобновить TLS-сессию после повторного подключения, но не

переносит состояние TCP-соединения. При значении по умолчанию `local` билеты сессий действуют только на выдавшем их узле.

Ограничения

При проектировании кластера учитывайте следующие ограничения:

- Поддерживается только кластер из активного и резервного узлов; одновременная обработка трафика обоими узлами не поддерживается.
- При переключении существующие TLS-соединения переустанавливаются; возобновление TLS-сессии на другом узле требует режима `ha-sync`.
- При первоначальной настройке сертификат УЦ HA необходимо вручную передать HA-партнёру.
- Inline-сервисы (NGFW) должны быть физически подключены к обоим узлам кластера.

Настройка отказоустойчивого кластера

В этом разделе описана настройка кластера из активного и резервного узлов. После настройки кластер передаёт виртуальные IP-адреса резервному узлу при отказе активного и синхронизирует между узлами общую часть конфигурации.

Предварительные требования

Для формирования HA-кластера необходимо:

- два узла Амибиг с одинаковой версией программного обеспечения;
- выделенный сетевой интерфейс на каждом узле для HA-канала: прямое соединение или отдельный VLAN;
- IP-связность между узлами по HA-каналу;
- одинаковый набор интерфейсов обработки трафика на обоих узлах;
- подключение inline-сервисов, например NGFW или DLP, к обоим узлам.

Параметры HA

Параметры кластера находятся в секции `ha`. Значения, которые относятся к конкретному узлу, необходимо задать отдельно на каждом устройстве.

Основные параметры

Основные параметры включают режим HA, идентификатор узла и правила адресации сервисного сегмента.

Параметр	Описание	По умолчанию
<code>ha.enabled</code>	Включение режима HA	<code>false</code>

Параметр	Описание	По умолчанию
<code>ha.mode</code>	Режим кластера	<code>active-standby</code>
<code>ha.local.node-id</code>	Уникальный идентификатор узла в кластере (1 или 2)	<code>1</code>
<code>ha.local.priority</code>	Приоритет узла (выше = предпочтительнее для активной роли)	<code>100</code>
<code>ha.peer.address</code>	IP-адрес HA-партнёра	—
<code>ha.service-addressing-mode</code>	Режим адресации сервис-сегмента: <code>shared</code> или <code>per-node</code>	<code>per-node</code>

HA-канал

Для основного и резервного HA-каналов задаются отдельные интерфейсы и IP-адреса.

Параметр	Описание	По умолчанию
<code>ha.ha-link.interface</code>	Сетевой интерфейс основного HA-канала	—
<code>ha.ha-link.address</code>	Локальный адрес на HA-канале с длиной префикса	—
<code>ha.ha-link.peer-address</code>	IP-адрес HA-партнёра на основном канале	—
<code>ha.ha-link-secondary.interface</code>	Интерфейс резервного HA-канала	—
<code>ha.ha-link-secondary.address</code>	Локальный адрес резервного канала с длиной префикса	—
<code>ha.ha-link-secondary.peer-address</code>	IP-адрес HA-партнёра на резервном канале	—

Резервный HA-канал настраивается в веб-интерфейсе на странице **Мониторинг → HA**. Для основного и резервного каналов предусмотрены отдельные блоки с выбором интерфейса. Служба `amibig-ha` поддерживает одно TCP-соединение с mTLS и при недоступности основного канала устанавливает его через резервный. Поэтому в таблице **HA-каналы** один канал обычно имеет состояние **Активен**, а второй — **Резерв**.

Режим адресации сервисов (`service-addressing-mode`)

Параметр `ha.service-addressing-mode` определяет, как узлы кластера адресуют сервисный сегмент — сеть, в которой находятся участники пулов, шлюзы inline-сервисов и ICAP-серверы:

Значение	В веб-интерфейсе	Назначение
<code>per-node</code> (по умолчанию)	Раздельная адресация по узлам	Каждый узел задаёт свои адреса участников пулов и шлюзов. При синхронизации конфигурации эти поля не перезаписываются значениями партнёра.
<code>shared</code>	Общая сервисная сеть	Оба узла используют одни и те же адреса сервисов. Участники пулов, шлюзы inline-сервисов и URI ICAP-серверов передаются с активного узла без изменения.

Режим выбирается на странице **Мониторинг → НА** (поле «Режим адресации сервисов»). Для отдельного сервиса глобальный режим можно переопределить параметром `services.service[].service-addressing-mode`; пустое значение означает наследование глобального.

Что остаётся локальным для узла в каждом режиме:

- **per-node** : адреса участников пулов (`pool.member`), шлюзы inline-сервиса (`inline.to-service.gateway` , `inline.from-service.gateway`), Self-IP inline-сервиса, URI ICAP-сервиса (`icap.request-mod.uri` , `icap.response-mod.uri`);
- **shared** : только адрес Self-IP inline-сервиса, чтобы исключить дублирование IP-адресов в одном L2-сегменте; остальные поля сервисного сегмента приходят от активного узла;
- в обоих режимах строго локальны: секция `ha` , сетевые настройки узла (`network.interfaces` , `network.vlans` , `network.self-ips`) и параметры интерфейса управления.

В режиме `shared` при применении конфигурации система проверяет, что каждый синхронизируемый адрес участника пула или шлюза входит в подсеть одного из локальных адресов Self-IP. Если условие не выполнено, после переключения участники пула будут недоступны. Такая конфигурация не применяется, а система выводит диагностическое сообщение.

Проверка полноты настройки резервного узла

При раздельной адресации (`per-node`) часть полей задаётся отдельно на каждом узле. Поэтому после синхронизации резервный узел может оказаться настроен не

полностью. На странице **Мониторинг → НА** блок «**Доконфигурировать на узле-партнёре**» показывает, каких локальных значений не хватает на текущем узле:

- сервис или пул без локального участника пула (**не задан member**);
- inline-сервис без локального шлюза (**не задан шлюз**);
- несовпадение типа сервиса с партнёром (например, `icap` против `inline`).

Когда все поля режима `per-node` заполнены, блок показывает **Конфигурация полная — недостающих полей нет**. Пока в блоке указаны пропущенные значения, переключение на этот узел приведёт к деградации перечисленных сервисов.

Параметры переключения

Переключение по состоянию сервисов управляется двумя группами параметров: глобальные пороги задаются в `ha.failover`, а вклад каждого сервиса в оценку работоспособности узла — в настройках сервиса (`services.service[]`).

Параметр	Описание	По умолчанию
<code>ha.failover.heartbeat-interval</code>	Интервал отправки контрольных сообщений, с	1
<code>ha.failover.dead-interval</code>	Время ожидания контрольного сообщения от НА-партнёра, с	3
<code>ha.failover.minimum-healthy-percent</code>	Порог оценки работоспособности узла (<code>HealthScore</code>), %. При значении 0 проверка отключена	50
<code>ha.failover.immediate-fail</code>	Переключение без учёта <code>failover-bonus</code> , если число доступных участников критичного сервиса (<code>on-degrade = drop</code>) ниже заданного минимума и НА-партнёр готов принять роль	false
<code>ha.failover.failover-bonus</code>	Гистерезис: роль передаётся НА-партнёру, только если его оценка выше локальной более чем на указанное значение	10
<code>services.service[].weight</code>	Вес сервиса в оценке работоспособности узла (1-65535)	100

Параметр	Описание	По умолчанию
<code>services.service[].min-healthy-members</code>	Минимальное число доступных участников пула	1
<code>services.service[].on-degrade</code>	Вклад деградировавшего сервиса в оценку: <code>drop</code> — критичный, <code>bypass</code> — некритичный	<code>drop</code>

Оценка работоспособности узла (HealthScore)

Оценка рассчитывается по весам сервисов:

$$\text{HealthScore} = (\sum \text{весов сервисов в состоянии UP}) / (\sum \text{весов участвующих сервисов}) \times 100$$

Участвующие сервисы — все, кроме переведённых в `bypass` (деградировали и объявлены некритичными) и выключенных (`enabled = false`).

Вес сервиса учитывается только целиком. Если число доступных участников не меньше `min-healthy-members`, сервис добавляет к оценке полный вес. Если число ниже минимума и задано `on-degrade = drop`, вес остаётся в общей сумме, но не входит в сумму доступных сервисов. При `on-degrade = bypass` сервис полностью исключается из расчёта.

По умолчанию сервис считается критичным (`on-degrade = drop`): снижение числа доступных участников ниже заданного минимума уменьшает оценку узла и может вызвать переключение. Значение `bypass` следует задавать только для некритичных сервисов, недоступность которых не должна приводить к переключению на резервный узел.

Если оценка опускается ниже `minimum-healthy-percent`, узел теряет право на автоматический выбор активной роли. Передача роли партнёру выполняется, только если он доступен, пригоден для обработки трафика и его оценка выше локальной более чем на `failover-bonus` процентов. Гистерезис предотвращает частую смену ролей при близких оценках.

Различие параметров `on-degrade` и «Действие при недоступности»

Это разные настройки с похожими словами. «Действие при недоступности» (`service-down-action`) определяет действие с клиентским соединением, когда сервис недоступен: пропустить без проверки, отбросить или сбросить. Параметр `on-degrade` управляет вкладом сервиса в оценку работоспособности узла при аварийном переключении и на конкретные соединения не влияет.

Восстановившийся узел не получает активную роль автоматически.

Шифрование

Шифрование HA-канала включено по умолчанию.

Параметр	Описание	По умолчанию
<code>ha.tls.enabled</code>	Шифрование HA-канала (mTLS, TLS 1.3)	<code>true</code>
<code>ha.tls.allow-insecure</code>	Разрешение незашифрованного TCP-соединения при отсутствии сертификатов	<code>false</code>

Виртуальные IP

В списке `ha.vip` задаются адреса, которые переходят на резервный узел.

Параметр	Описание	По умолчанию
<code>ha.vip</code>	Список виртуальных IP-адресов (адрес + интерфейс)	<code>[]</code>

Пошаговая настройка кластера

Настройте сетевую связность, затем задайте параметры каждого узла и примените конфигурацию. После применения проверьте роль узлов и состояние HA-канала.

Шаг 1. Подготовка сетевых интерфейсов

Настройте HA-интерфейс на каждом узле и проверьте IP-связность с HA-партнёром. Не передавайте по HA-каналу обрабатываемый трафик.

Шаг 2. Настройка узла A

Добавьте следующие значения в подготовленную конфигурацию первого узла:

```
{
  "ha": {
    "enabled": true,
    "mode": "active-standby",
    "local": {
      "node-id": 1,
      "priority": 100
    },
    "peer": {
      "address": "10.10.0.2"
    },
    "ha-link": {
      "interface": "enp4s0",
      "address": "10.10.0.1/24",
```



```

    "peer-address": "10.10.0.2"
  },
  "failover": {
    "heartbeat-interval": 1,
    "dead-interval": 3
  },
  "tls": {
    "enabled": false
  },
  "vip": [
    { "address": "192.168.56.100/24", "interface": "dpdk0" }
  ]
}

```

В поле `ha-link.interface` укажите назначенное Linux имя физического интерфейса, например `enp4s0` или `eno2np1`. Имя `ha0` из файла `yang/example-config.json` приведено только как пример. В веб-интерфейсе нужный порт выбирается на странице **Мониторинг → НА**, в поле **Канал НА → Физический интерфейс**.

Шаг 3. Настройка узла В

На втором узле задайте другой `node-id`, меньший приоритет и обратные адреса НА-канала:

```

{
  "ha": {
    "enabled": true,
    "mode": "active-standby",
    "local": {
      "node-id": 2,
      "priority": 50
    },
    "peer": {
      "address": "10.10.0.1"
    },
    "ha-link": {
      "interface": "enp0s8",
      "address": "10.10.0.2/24",
      "peer-address": "10.10.0.1"
    },
    "failover": {
      "heartbeat-interval": 1,
      "dead-interval": 3
    },
    "tls": {
      "enabled": false
    },
    "vip": [
      { "address": "192.168.56.100/24", "interface": "dpdk0" }
    ]
  }
}

```

Шаг 4. Применение конфигурации

Примените конфигурацию на обоих узлах с помощью команды **Применить конфигурацию** в веб-интерфейсе или через REST API:

В примерах `/path/to/amibig-webui.pem` — путь к проверенному сертификату веб-интерфейса или сертификату выдавшего его УЦ. Порядок проверки сертификата приведён в разделе Первоначальная настройка.

```
curl --cacert /path/to/amibig-webui.pem \
-X POST https://node-a.example.com/api/v1/config/apply \
-H "Authorization: Bearer $TOKEN" \
-H "Content-Type: application/json" \
-d '{"description": "Включение HA-кластера"}'
```

После применения служба `amibig-ha` создаёт на каждом узле сертификат УЦ и сертификат узла. Пока TLS выключен, HA-канал не зашифрован. Сразу после первоначального применения выполните следующий шаг.

Шаг 5. Включение шифрования HA-канала

На каждом узле скачайте сертификат УЦ командой **HA → Скачать СА этого узла**. Загрузите сертификат УЦ узла А на узел В, а сертификат УЦ узла В — на узел А командой **HA → Загрузить СА партнёра**. Затем задайте `ha.tls.enabled = true` на обоих узлах и примените конфигурацию.

Подробнее порядок обмена сертификатами описан в подразделе «Обмен сертификатами УЦ».

Шаг 6. Проверка состояния кластера

Убедитесь, что кластер сформирован:

```
curl --cacert /path/to/amibig-webui.pem \
https://node-a.example.com/api/v1/ha/status \
-H "Authorization: Bearer $TOKEN"
```

Ожидаемый результат на активном узле:

```
{
  "enabled": true,
  "role": "active",
  "peer-state": "up",
  "tls-active": true,
  "uptime": 24892.4,
  "last-heartbeat": 24891.8,
  "last-config-sync": 24710.2
}
```

Поля `uptime`, `last-heartbeat` и `last-config-sync` содержат значения монотонных часов процесса в секундах, а не календарные метки времени. Например, возраст последнего контрольного сообщения равен разности `uptime` и `last-heartbeat`.

Приоритет и выбор активного узла

Приоритет (`priority`) определяет, какой узел предпочтителен для активной роли:

- узел с более высоким значением `priority` становится активным при первоначальном формировании кластера;
- значения `priority` на двух узлах должны различаться;
- рекомендуемые значения: `100` для предпочтительного активного узла и `50` для резервного.

Восстановившийся узел не получает VIP автоматически. Поэтому приоритет влияет на первоначальный выбор активного узла при одновременном запуске, но не вызывает автоматический возврат роли.

HA-канал

По HA-каналу узлы передают следующие данные:

- контрольные сообщения для проверки доступности HA-партнёра;
- изменения общей части конфигурации;
- ключи TLS-сессий, если в профиле клиентского соединения задано `session-ticket-mode = ha-sync`.

Состояние TCP-соединений по HA-каналу не передаётся. После переключения клиенты устанавливают соединения заново.

Ручная передача активной роли

На текущем активном узле откройте **Мониторинг → HA** и нажмите **Передать active партнёру**.

- Конфигурация кластера при этом не меняется.
- Текущий активный узел временно снижает свой приоритет и переходит в состояние `backup`; HA-партнёр становится владельцем VIP.
- Исходный узел остаётся резервным до следующей ручной передачи роли или автоматического переключения.

После передачи роли убедитесь, что новый активный узел владеет VIP, а прежний активный узел перешёл в резерв.

Требования к HA-каналу

HA-канал должен соответствовать следующим требованиям.

Параметр	Рекомендация
Тип соединения	Выделенный интерфейс: прямое соединение или отдельный VLAN

Параметр	Рекомендация
Пропускная способность	1 Гбит/с минимум, 10 Гбит/с рекомендуется
Задержка	< 1 мс
Использование	Не передавать обрабатываемый трафик

Интервалы контрольных сообщений

Интервалы определяют, как быстро узел обнаруживает потерю связи с HA-партнёром. При выборе значений учитывайте задержку и возможные кратковременные потери пакетов в HA-канале.

Интервал отправки

Параметр `failover.heartbeat-interval` задаёт интервал отправки контрольных сообщений в секундах. Значение по умолчанию — 1 секунда.

- Меньшее значение обеспечивает более быстрое обнаружение отказа, но увеличивает нагрузку на HA-канал.
- Рекомендуемый диапазон: 1–5 секунд.

Время ожидания

Параметр `failover.dead-interval` задаёт время ожидания контрольного сообщения от HA-партнёра. Если за это время сообщение не получено, узел считает партнёра недоступным. Значение по умолчанию — 3 секунды.

- Значение должно быть больше `heartbeat-interval`.
- Рекомендуемое соотношение: `dead-interval >= 3 * heartbeat-interval`.
- Слишком малое значение может приводить к ложным срабатываниям при кратковременных задержках на HA-канале.

Пример быстрого обнаружения отказа

В следующем примере используется минимальный рекомендуемый интервал.

```
{
  "failover": {
    "heartbeat-interval": 1,
    "dead-interval": 3
  }
}
```

Узел обнаружит потерю HA-партнёра не позднее чем через 3 секунды. Общее время переключения также зависит от обновления сетевых таблиц на соседних устройствах.

Пример для сети с нестабильной задержкой

Увеличенный интервал подходит для HA-канала с кратковременными задержками.

```
{
  "failover": {
    "heartbeat-interval": 2,
    "dead-interval": 10
  }
}
```

Узел обнаружит потерю HA-партнёра не позднее чем через 10 секунд. Такой интервал снижает вероятность ложного переключения при кратковременных задержках в HA-канале.

Шифрование HA-канала

При включении параметра `ha.tls.enabled` обмен между узлами защищается взаимной TLS-аутентификацией (mTLS). Минимальная версия протокола — TLS 1.3.

Автоматическая генерация сертификатов

При запуске `amibig-ha` (как только `ha.enabled = true`, независимо от `tls.enabled`) система автоматически генерирует:

- сертификат УЦ для HA: самоподписанный сертификат EC P-256 со сроком действия 10 лет, уникальный для каждого узла;
- сертификат узла, подписанный собственным УЦ; поле SAN содержит IP-адрес узла в HA-канале.

Сертификат УЦ создаётся даже при `tls.enabled = false`. Это позволяет передать его HA-партнёру до включения TLS: для установления защищённого соединения каждый узел должен доверять УЦ партнёра.

Обмен сертификатами УЦ

Каждый узел создаёт собственный самоподписанный сертификат УЦ. Для работы mTLS выполните следующие действия:

1. На обоих узлах включите `ha.enabled = true` и примените конфигурацию. TLS на этом этапе можно оставить выключенным.
2. На каждом узле выберите **HA → Скачать СА этого узла** или выполните запрос `GET /api/v1/ha/own-ca`.
3. На каждом узле загрузите сертификат УЦ HA-партнёра: выберите **HA → Загрузить СА партнёра** или отправьте запрос `POST /api/v1/ha/peer-ca` с параметром `file=...` в формате multipart либо с телом JSON `{"pem": "..."}.`
4. Включите `ha.tls.enabled = true` на обоих узлах и примените конфигурацию. При следующем подключении узлы установят соединение с mTLS.

Работа без шифрования

При `tls.enabled = false` HA-канал использует незашифрованное TCP-соединение. Если TLS включён, но сертификаты отсутствуют, служба не запускает HA-канал. Параметр `tls.allow-insecure = true` разрешает в этом случае перейти на незашифрованное соединение. Не применяйте эти режимы в промышленной эксплуатации.

Синхронизация конфигурации

При применении конфигурации через веб-интерфейс или API активный узел передаёт обновлённую конфигурацию резервному. Синхронизируются общие секции: профили SSL/TLS, сервисы, сервисные цепочки и политики.

Параметры конкретного узла сохраняются локально и не перезаписываются при синхронизации: секция `ha`, включая `node-id`, `priority` и адреса HA-канала; сетевые параметры узла (`network.interfaces`, `network.vlans`, `network.self-ips`); параметры интерфейса управления; поля сервисов режима `per-node`. Подробнее см. в подразделе «Режим адресации сервисов».

Ручную синхронизацию можно инициировать через веб-интерфейс или API:

```
curl --cacert /path/to/amibig-webui.pem \
-X POST https://node-a.example.com/api/v1/ha/sync \
-H "Authorization: Bearer $TOKEN"
```

Проверка доступности сервисов

Система контролирует доступность локальных сервисов на обоих узлах. Если оценка работоспособности активного узла опускается ниже порога, система может передать VIP резервному узлу.

На резервном узле проверки доступности продолжают выполняться. Сами по себе они не передают узлу активную роль: переключение происходит только при выполнении условий, заданных в секции `ha.failover`.

Состояние соединений при переключении

При переключении Амибиг передаёт резервному узлу виртуальные IP-адреса и часть данных, необходимых для возобновления работы. Состояние отдельных TCP-соединений между узлами не копируется. В этом разделе описано, какие соединения клиенты должны установить заново и какие данные кластер синхронизирует заранее.

Порядок переключения

Резервный узел постоянно контролирует активный узел по HA-каналу. После обнаружения отказа он принимает VIP и начинает обрабатывать новые соединения уже запущенной службой `tforged`. Существующие TCP-соединения при этом разрываются.

При значениях по умолчанию узлы отправляют контрольные сообщения раз в секунду и считают HA-партнёра недоступным через 3 секунды без ответа. После этого кластер передаёт VIP и обновляет таблицы соседних сетевых устройств. Полное время переключения зависит также от сходимости сети.

Поведение TCP-соединений

После переключения клиент повторно подключается к тому же VIP, и новое соединение обрабатывает ставший активным узел. Это относится в том числе к TCP-потокам в режимах `bypass` и `passthrough`: они также проходят через прокси и не переносятся между узлами.

Планируя отказоустойчивость прикладного сервиса, учитывайте возможность разрыва соединения и настройте повторное подключение на стороне клиента.

Синхронизация ключей TLS-сессий (STEK)

Синхронизация STEK позволяет возобновить TLS-сессию после повторного подключения к другому узлу, но не восстанавливает TCP-соединение.

Назначение синхронизации

Амибиг использует билеты TLS-сессий для их возобновления. По умолчанию задано `ssl.client-profile.session-ticket-mode = local`: билет действует только на выдавшем его узле, а STEK между узлами не передаётся.

Чтобы синхронизировать STEK, выполните следующие действия:

1. Откройте **SSL/TLS → Профиль клиента** и выберите профиль, используемый для TLS-инспекции.
2. В группе **Кеши и возобновление сессий** установите для поля **Билеты сессий (session tickets)** значение **Синхронизировать с HA-партнёром (ha-sync)**.
3. Сохраните и примените конфигурацию.
4. В ответе `GET /api/v1/ha/status` проверьте значение `runtime-preservation.session-ticket-mode`: оно должно быть равно `ha-sync`. Поле `last-stek-sync` показывает время последней передачи ключа.

В режиме `ha-sync` активный узел создаёт ключ шифрования билетов TLS-сессий (STEK) и передаёт его резервному узлу по HA-каналу.

Синхронизированный STEK позволяет клиенту после повторного подключения возобновить TLS-сессию вместо полного TLS handshake, если клиент и профиль TLS поддерживают соответствующий механизм. Это сокращает задержку повторного подключения, но не сохраняет TCP-соединение.

Порядок синхронизации

После включения режима `ha-sync` ключ передаётся в следующем порядке:

1. Активный узел создаёт STEK.
2. Ключ передаётся резервному узлу по HA-каналу.
3. Резервный узел сохраняет ключ для использования после переключения.
4. Ключи синхронизируются каждые 5 минут.

Поддерживаемые механизмы

В таблице указано, какие варианты возобновления TLS используют синхронизированный STEK.

Механизм	Поддержка	Примечание
Билеты сессий TLS 1.2	Да	Основной механизм возобновления TLS 1.2
PSK TLS 1.3	Да	PSK формируется на основе ключа билета сессии
Возобновление TLS 1.2 по идентификатору сессии	Нет	Требуется синхронизация кеша сессий
Незавершённый TLS handshake	Нет	Клиент устанавливает соединение заново

Проверка возобновления TLS-сессии

Сохраните сессию до переключения, затем подключитесь к тому же VIP с сохранёнными параметрами:

```
VIP=192.0.2.10

### Первое соединение: сохранить параметры сессии
openssl s_client -connect "${VIP}:443" -sess_out /tmp/sess.pem

### После переключения: попытаться возобновить сессию
openssl s_client -connect "${VIP}:443" -sess_in /tmp/sess.pem
```

При успешном возобновлении во втором выводе будет указано `Reused`, а не `New`.

Восстановление потоков UDP и ICMP (`loose-init`)

Состояние потоков UDP и ICMP не копируется между узлами. Когда после переключения новый активный узел получает первый пакет от клиента, он заново применяет политику и создаёт запись о потоке. Это относится к режимам `bypass`, `passthrough` и `reject`, в том числе при передаче через цепочку inline-сервисов.

Для этого механизма действуют следующие ограничения:

- поток создаётся только по пакету от клиента; ответный пакет, который пришёл раньше нового клиентского пакета, отбрасывается;
- механизм не применяется к режиму `intercept`, в котором соединение обрабатывается прокси;
- TCP-потоки, включая режим `bypass`, клиент устанавливает заново.

При создании потока система повторно применяет действующую политику. Если политика запрещает пакет, запись о потоке не создаётся. Неактивные записи удаляются по тайм-ауту.

Синхронизируемые данные

Кластер синхронизирует данные, которые можно подготовить до переключения.

Данные	Периодичность
Конфигурация	После применения изменений
Ключи TLS-сессий (STEK)	Каждые 5 минут при <code>session-ticket-mode = ha-sync</code>

Сертификаты УЦ для HA автоматически не синхронизируются. При первоначальной настройке администратор вручную передаёт сертификат УЦ каждого узла HA-партнёру.

Несинхронизируемые данные

Следующие данные создаются заново после переключения.

Данные	Причина	Результат переключения
Состояние TCP-соединений	Между узлами не копируется	Клиент повторно подключается
Состояние мониторов доступности	Рассчитывается локально	Новый активный узел использует результаты собственных проверок
ICAP-сессии	Устанавливаются каждым узлом самостоятельно	ICAP-соединение устанавливается заново

Данные	Причина	Результат переключения
Пакеты в обработке	Содержимое буферов не копируется	После повторного подключения приложение повторяет запрос или операцию

Виртуальные IP-адреса (VIP)

Клиенты и соседние сетевые устройства обращаются к кластеру по виртуальным IP-адресам (VIP), а не по адресам конкретных узлов. VIP в каждый момент принадлежит одному узлу. При переключении адрес переходит резервному узлу, поэтому менять маршруты и настройки клиентов не нужно.

Передача VIP между узлами

Оба узла держат службу `tforge` запущенной. Активный узел владеет VIP, отвечает на запросы ARP и ICMP к этим адресам и обрабатывает трафик. Резервный узел контролирует HA-партнёра, но не отвечает на запросы к VIP.

После обнаружения отказа резервный узел выполняет следующие действия:

1. Переходит в активную роль и назначает весь список `ha.vip`.
2. Если задан параметр `floating-mac`, назначает соответствующий служебный MAC-адрес.
3. Рассылает GARP, чтобы соседние устройства обновили таблицы ARP и FDB.
4. Начинает принимать новые соединения.

Все адреса из списка `ha.vip` передаются одновременно. Частичная передача списка между узлами не поддерживается.

На уровне реализации `ha-daemon` публикует роль в файле `/run/tforge/role.txt`, а список назначенных VIP — в файле `/run/tforge/vip-state.txt`. Рабочий поток `vip_anchor` применяет эти данные в сетевом стеке F-Stack.

Настройка VIP

Для каждого VIP необходимо задать адрес и однозначно указать сетевой сегмент, в котором кластер будет принимать трафик.

Параметры

VIP задаются в списке `ha.vip`. Для каждого адреса укажите VLAN, порт обработки трафика или агрегированный канал.

Поле	Тип	Описание
<code>address</code>	Строка	IP-адрес с маской подсети в формате CIDR, например <code>192.0.2.10/24</code>
<code>vlan</code>	Строка	Имя VLAN обработки трафика для тегированного размещения VIP
<code>interface</code>	Строка	Имя порта обработки трафика или агрегированного канала для нетегированного размещения VIP
<code>floating-mac</code>	Строка	Необязательный служебный MAC-адрес в формате <code>xx:xx:xx:xx:xx:xx</code>

Для каждого VIP задайте только одно из полей: `vlan` или `interface`. Конфигурация с обоими полями либо без обоих полей не применяется.

Параметр `floating-mac` позволяет передавать между узлами связку VIP и MAC-адреса. В этом случае GARP служит дополнительным уведомлением, а сходимость сети меньше зависит от обновления записи ARP по новому MAC-адресу.

Пример конфигурации

В следующем примере два VIP размещены в разных VLAN:

```
{
  "ha": {
    "vip": [
      {
        "address": "192.168.56.100/24",
        "vlan": "client",
        "floating-mac": "02:ab:56:00:00:64"
      },
      {
        "address": "10.10.0.100/24",
        "vlan": "internet",
        "floating-mac": "02:ab:70:00:00:64"
      }
    ]
  }
}
```

В схеме с отдельными клиентским, сервисным и серверным сегментами можно назначить собственный VIP в каждом сегменте. Сетевые устройства должны использовать VIP соответствующего сегмента как адрес следующего шлюза.

Ограничения

При настройке учитывайте следующие ограничения:

- поддерживается до 64 VIP;
- VIP должен входить в подсеть, настроенную на указанном VLAN или интерфейсе;
- VIP не должен совпадать с адресом какого-либо узла;
- список VIP и привязки к VLAN или интерфейсам должны совпадать на обоих узлах;
- значение `floating-mac` должно совпадать на обоих узлах;
- для всех VIP в одном VLAN или на одном физическом интерфейсе следует либо не задавать `floating-mac`, либо указывать один и тот же адрес.

Состояние кластера до и после переключения

В следующих примерах показано, какой узел владеет VIP при штатной работе, после отказа и после восстановления исходного активного узла.

Штатная работа

В штатном режиме узел А владеет VIP, а узел В контролирует его состояние:

Клиент → 192.168.56.100 → Узел А, активный, владеет VIP
Узел В, резервный

Узел А отвечает на запросы ARP к адресу 192.168.56.100 и обрабатывает направленный на него трафик.

Отказ узла А

После обнаружения отказа узел В принимает активную роль:

1. Кластер обнаруживает отказ узла А.
2. Узел В назначает VIP и служебный MAC-адрес.
3. Узел В рассылает GARP.
4. Коммутаторы обновляют таблицы ARP и FDB.
5. Клиентский трафик поступает на узел В.

Внутренняя последовательность операций на новом активном узле приведена в таблице.

Шаг	Действие
1	<code>ha-daemon</code> обновляет файлы <code>role.txt</code> и <code>vip-state.txt</code>

Шаг	Действие
2	<code>tforge</code> включает режим <code>intercept</code> и назначает VIP
3	Если задан <code>floating-mac</code> , узел назначает служебный MAC-адрес
4	Узел рассылает GARP для адресов обработки трафика и VIP
5	Сеть обновляет таблицы ARP и FDB
6	Новый активный узел принимает новые соединения
7	Узел использует синхронизированные данные там, где это поддерживается

Восстановление узла A

При использовании режима `nopreempt` восстановившийся узел A становится резервным, а VIP остаются на узле B. Чтобы вернуть активную роль узлу A, администратор должен выполнить ручное переключение.

Проверка состояния VIP

Владение VIP проверяется через REST API либо непосредственно по файлам состояния на узле.

Проверка через REST API

Запросите состояние HA на нужном узле:

В примере `/path/to/amibig-webui.pem` — путь к проверенному сертификату веб-интерфейса или сертификату выдавшего его УЦ.

```
curl --cacert /path/to/amibig-webui.pem \
  https://node.example.com/api/v1/ha/status \
  -H "Authorization: Bearer $TOKEN"
```

Поле `role` содержит текущую роль:

- `active` — узел владеет VIP;
- `standby` — узел не владеет VIP и контролирует активный узел.

Проверка через командную строку

VIP назначается в таблице адресов F-Stack и не отображается в выводе `ip addr`. Проверьте роль и список адресов по файлам состояния:

```
cat /run/tforge/role.txt      # master: узел владеет VIP; backup: не владеет
cat /run/tforge/vip-state.txt # список VIP текущего активного узла
```

На резервном узле файл `role.txt` содержит `backup`, а файл `vip-state.txt` пуст.

Чтобы проверить сетевую доступность, отправьте `ping` на VIP из клиентского сегмента. На запросы ICMP к VIP отвечает только активный узел. Адреса Self-IP остаются доступны на обоих узлах. Такая проверка не создаёт событий на экране **Трафик**.

Рекомендации

При планировании адресов и маршрутов соблюдайте следующие рекомендации:

- выделяйте для VIP отдельные адреса, которые не совпадают с адресами узлов;
- указывайте VIP как шлюз для клиентов, inline-сервисов и серверов в соответствующих сегментах;
- в маршрутизируемых сегментах направляйте обратный трафик на VIP, а не на Self-IP конкретного узла, иначе после переключения ответы могут поступать на прежний узел;
- в локальных L2-сегментах задавайте `floating-mac`, чтобы уменьшить зависимость от скорости обновления таблиц ARP;
- не используйте адреса интерфейса управления и HA-канала в качестве VIP;
- контролируйте состояние кластера через веб-интерфейс, REST API и системный журнал.

Защита от одновременной активности узлов

При потере всех HA-каналов оба узла могут считать себя активными и назначить одинаковые VIP. Чтобы снизить риск такого состояния, используйте выделенный основной HA-канал и настройте резервный канал по независимому сетевому пути. Контролируйте доступность обоих каналов средствами мониторинга инфраструктуры.

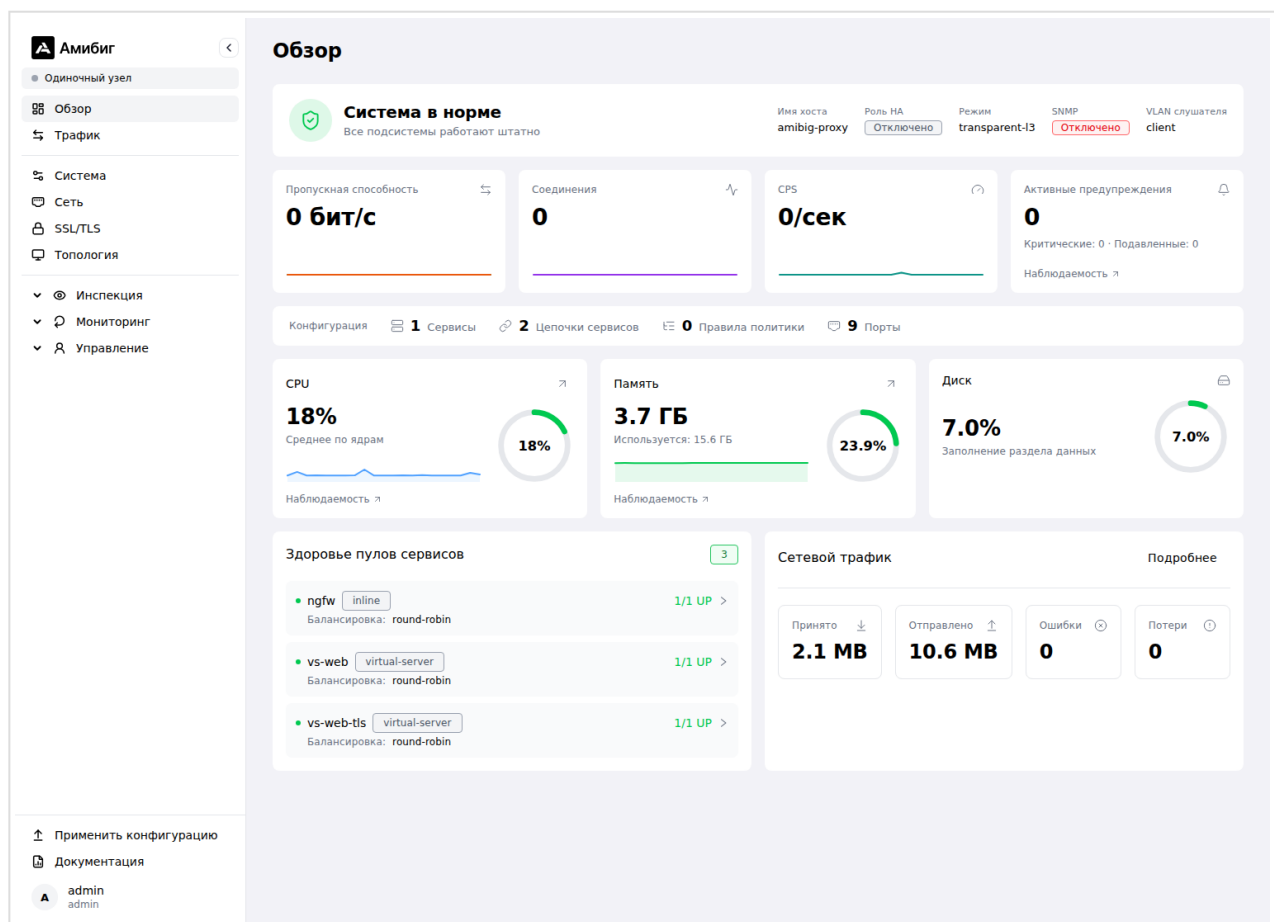
Управление

Раздел объединяет административные задачи: работу с веб-интерфейсом и REST API, учётные записи, лицензию, мониторинг, резервное копирование и обновления. Эти процедуры выполняет администратор устройства после установки и во время эксплуатации.

Для безопасной эксплуатации сначала настройте пользователей и доступ к интерфейсу управления, затем проверьте лицензию, резервное копирование и порядок обновления.

Веб-интерфейс

Веб-интерфейс Амибиг предназначен для настройки устройства и контроля его состояния. Он доступен на русском и английском языках и использует тот же REST API, который открыт для внешних систем автоматизации.

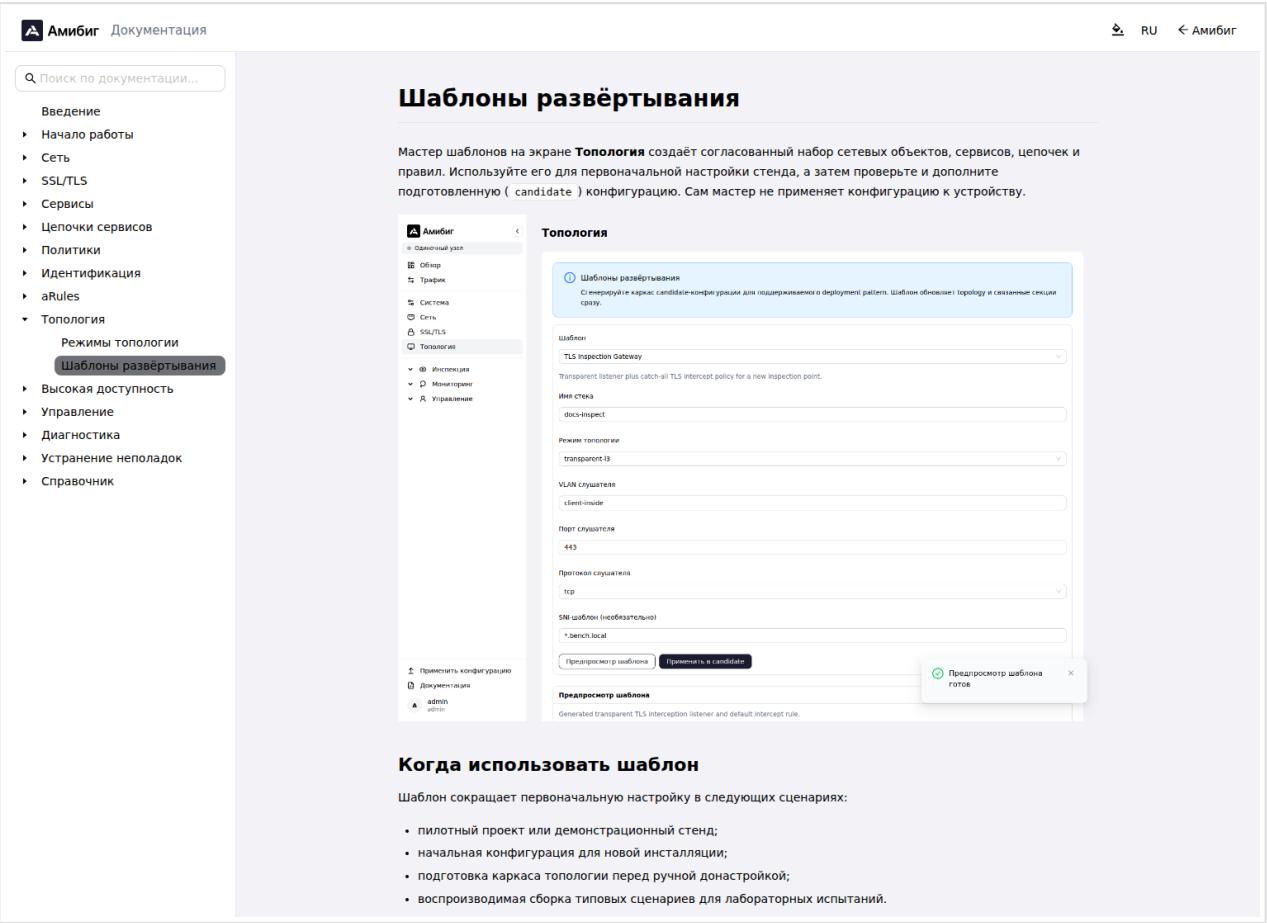


Доступ

Откройте веб-интерфейс по адресу интерфейса управления:

```
https://<management-ip>/
```

При открытии `http://<management-ip>/` устройство возвращает перенаправление 301 на `https://` . Если управление публикуется через обратный прокси-сервер или внешний VIP, используйте адрес, выделенный для доступа к управлению. Встроенная документация открывается по пути `/docs` и не требует отдельной внешней базы знаний.



Структура интерфейса

Верхний уровень бокового меню — основные рабочие экраны, остальные разделы собраны в три группы: «Инспекция», «Мониторинг» и «Управление».

Экраны верхнего уровня:

Экран	Что настраивается или отображается
Обзор	состояние системы, загрузка процессоров, рабочие потоки dataplane, доступность пулов и показатели работы

Экран	Что настраивается или отображается
Трафик	главный рабочий экран: события обработки соединений и решения политики в реальном времени
Система	адрес управления, параметры DPDK, трансляция адреса источника и параметры TCP/MSS
Сеть	физические интерфейсы, агрегирование каналов, VLAN, self-IP, маршруты
SSL/TLS	сертификаты, доверенные УЦ, наборы шифров, клиентские и серверные профили, сертификат веб-интерфейса
Топология	слушатель, режим dataplane, шаблоны развёртывания и общие параметры обработки UDP, ICMP и QUIC

Группа «Инспекция»:

Экран	Что настраивается или отображается
Мониторы	именованные мониторы доступности, которые можно назначать пулам
Пулы	пулы серверов, участники пулов и способ балансировки
Виртуальные серверы	виртуальные серверы и привязка пулов
Сервисы	inline-, ICAP- и sideband-сервисы, участники пулов и проверка доступности
Цепочки сервисов	цепочки сервисов и их визуальная диаграмма
Политики	правила, критерии совпадения, действия, цепочки, aRules, явное правило по умолчанию и порядок обработки
Идентификация	синхронизация с LDAP, соответствия «адрес — пользователь», портал аутентификации
Категории URL	категории URL для критериев политики
aRules	Tcl-правила обработки трафика
Шейпинг	классы управления трафиком и ограничения их применения в текущей версии
GSLB	глобальная балансировка (wide-IP)

Группа «Мониторинг»:

Экран	Что настраивается или отображается
HA	состояние кластера, роли узлов, синхронизация
Логирование	системный журнал, аудит, события SSL/TLS, передача событий в syslog и Kafka
Просмотр логов	поиск и экспорт событий
Наблюдаемость	оповещения, накопленные показатели, передача журнала соединений в SIEM и выборка событий
Диагностика	объяснение потока, захват пакетов и диагностический архив
SNMP	SNMP-агент и trap-приёмники

Группа «Управление»:

Экран	Что настраивается или отображается
Пользователи , Лицензия , Обслуживание (включая обновление ПО), Резервные копии , История	эксплуатационные операции

Индикатор роли узла

В боковой панели на любой вкладке постоянно отображается индикатор роли узла: активный (зелёный), резервный (синий), одиночный (серый) или неизвестно (жёлтый). В свёрнутой панели индикатор остаётся видимым как цветная точка. Состояние обновляется автоматически по данным `GET /api/v1/ha/status`. Индикатор позволяет проверить, к какому узлу подключён администратор, до выполнения операций обслуживания.

Панель фильтрации на экране Трафик

Над таблицей соединений расположена сворачиваемая панель фильтров: текстовые поля Клиент , Сервер , Хост , Цепочка , Правило (подстрока, без учёта регистра) и множественный выбор Протокол / Действие со значениями из загруженных событий. Условия объединяются по И; кнопка Сбросить фильтры возвращает полный список. Фильтры переживают автообновление таблицы.

Параметры исходящих соединений и TCP

Кроме адреса управления и движковых параметров DPDK, экран Система содержит две секции трафиковой политики:

- «**Egress (трансляция адреса источника)**» — режим `origin-snat` (`automap` , `none` или `pool`), правила выбора порта источника и SNAT-пул; подробности в разделе Трансляция адреса источника (SNAT);
- «**Протокол (TCP / MSS)**» — ограничение MSS (общее и по направлениям), `proxy-mss` и обнаружение недоступности PMTUD; подробности в разделе MSS и MTU.

Изменение этих параметров требует применения с перезапуском dataplane (`cold apply`). Активные соединения при перезапуске прерываются.

Сертификат HTTPS веб-интерфейса

На экране SSL/TLS есть отдельная вкладка сертификата веб-интерфейса. Это сертификат интерфейса управления; он не связан с сертификатами TLS-инспекции. Вкладка показывает текущий сертификат: CN, SAN, издателя, отпечаток, срок действия и источник сертификата. На вкладке можно:

- сгенерировать CSR и скачать его (приватный ключ не покидает устройство);
- загрузить выпущенный сертификат в PEM (сертификат, закрытый ключ и цепочка) или PKCS#12 с паролем;
- откатиться к самоподписанному сертификату.

После замены сертификата текущий сеанс завершается. Повторно откройте адрес управления, проверьте новый сертификат в браузере и выполните вход.

Темы и язык

Параметры отображения действуют только в текущем браузере и не изменяют конфигурацию устройства:

- Темы: `dark` , `light` , `turbo` .
- Языки: `ru` , `en` .
- Выбранные тема и язык сохраняются в `localStorage` браузера.
- Переключение языка влияет и на встроенную документацию.

Подготовка и применение конфигурации

Веб-интерфейс разделяет редактирование и применение конфигурации:

1. Администратор изменяет объекты в формах и таблицах.
2. Изменения сохраняются в **подготовленной (candidate) конфигурации**.
3. `Diff` показывает различия между подготовленной и действующей конфигурациями.

4. Применить конфигурацию делает подготовленную конфигурацию действующей (`running`).
5. После применения запускается **таймер отката**.
6. Подтвердить фиксирует конфигурацию; по истечении таймера выполняется автоматический откат.

После подтверждения проверьте состояние сервисов и прохождение тестового трафика. Если подтверждение не поступит до истечения таймера, Амибиг вернёт предыдущую действующую конфигурацию.

Применение с перезапуском dataplane

Некоторые параметры нельзя применить без перезапуска обработчика трафика: состав и привязка dataplane-портов, VLAN, агрегированные каналы, режим топологии, параметры DPDK, секции `egress` и `protocol`, отдельные per-node поля inline-сервисов.

На странице применения веб-интерфейс показывает предупреждение, если изменение требует перезапуска (`cold apply`). Перед применением такой конфигурации выведите трафик с узла, переведите кластер в окно обслуживания или заранее подтвердите допустимый перерыв обслуживания.

Важно

Таймер автоматического отката защищает доступ к управлению, но не предотвращает разрыв активных соединений при перезапуске. Если требуется сохранить клиентские сессии, сначала выполните вывод трафика или переключение на соседний узел.

Адрес интерфейса управления

В разделе Система параметр Способ получения адреса определяет, как Амибиг настраивает интерфейс управления после загрузки:

Значение	Поведение
ДНСР	Адрес и шлюз получает ДНСР-клиент. Поля Адрес и Шлюз не сохраняются в конфигурации.
Статический адрес	Администратор задаёт <code>system.management.address</code> и при необходимости <code>system.management.gateway</code> .

Адрес, полученный по DHCP, не сохраняется как статическое значение. После изменения способа получения адреса дождитесь применения конфигурации и проверьте доступность веб-интерфейса по новому адресу.

Практический порядок настройки

Следующая последовательность помогает создать зависимости в нужном порядке и проверить результат до применения конфигурации.

Базовая настройка

Создавайте связанные объекты в следующем порядке:

1. Настройте порты, VLAN и маршруты на экране `Сеть` .
2. Подготовьте CA и TLS-профили на экране `SSL/TLS` .
3. Создайте сервисы безопасности и пулы на экране `Сервисы` .
4. Соберите цепочки на экране `Цепочки сервисов` .
5. Определите логику решений на экране `Политики` .
6. Проверьте результат на экране `Диагностика` в блоке `Объяснение потока` .
7. Примените изменения, затем подтвердите их.

Быстрый старт через шаблоны

Если нужен типовой стенд или демонстрационная конфигурация, начните с шаблонов развёртывания. Конструктор создаёт скелет топологии сразу в нескольких секциях подготовленной конфигурации.

Встроенные инструменты

Перед применением и при диагностике используйте встроенные средства контроля:

- **Diff** — что изменится при применении.
- **История** — кто и что менял.
- **Диагностика** — объяснение потока (моделирование решения политики до подачи трафика), захват пакетов, диагностический архив для передачи в поддержку.
- **Наблюдаемость** — оповещения, динамика показателей и настройка передачи в SIEM (см. Наблюдаемость).
- **Документация** — встроенное руководство по продукту на `/docs` .

Роли и доступ

Доступные действия определяются правами роли текущей учётной записи. На экране `Диагностика` права различаются по операциям: объяснение потока доступно всем ролям, оператор может просматривать и скачивать созданные захваты, а

управление захватом и создание диагностического архива доступны только администратору.

Действие	admin	operator	viewer
Просмотр конфигурации и статуса	+	+	+
Редактирование подготовленной конфигурации	+	+	—
Применение / подтверждение / откат	+	—	—
Управление пользователями и лицензией	+	—	—
Объяснение потока	+	+	+
Просмотр и скачивание захватов пакетов	+	+	—
Запуск, остановка и удаление захвата пакетов	+	—	—
Создание диагностического архива	+	—	—

Подробнее о ролях и правах — в разделе Управление пользователями.

REST API

REST API Амибиг предназначен для автоматизации настройки и мониторинга. Все методы доступны по префиксу `/api/v1/` и требуют аутентификации, кроме явно указанных общедоступных методов.

Базовый URL:

```
https://<management-ip>/api/v1/
```

Все запросы и ответы используют формат JSON, если не указано иное (загрузка/скачивание файлов используют `multipart/form-data` и `application/octet-stream`).

Аутентификация

Для защищённых методов REST API используется Bearer-токен. Получите токен с учётными данными пользователя и передавайте его в заголовке каждого запроса.

Получение токена

Передайте имя учётной записи и пароль в метод входа:

```
POST /api/v1/auth/login
```

Тело запроса:

```
{
  "username": "admin",
  "password": "your-password"
}
```

Ответ:

```
{
  "token": "admin:admin:1709123456:a1b2c3d4...",
  "user": "admin",
  "role": "admin",
  "must_change_password": false
}
```

Использование токена

Токен передается в заголовке `Authorization` каждого запроса:

```
Authorization: Bearer <token>
```

Срок действия токена — 8 часов. После его окончания повторно выполните вход.

Смена пароля

Пользователь может изменить собственный пароль следующим запросом:

```
POST /api/v1/auth/password
```

```
{
  "old_password": "current-password",
  "new_password": "new-Passw0rd!",
  "confirm_password": "new-Passw0rd!"
}
```

Информация о текущем пользователе

Метод возвращает имя, роль и признак обязательной смены пароля:

```
GET /api/v1/auth/me
```

Конфигурация

Методы конфигурации читают и изменяют подготовленное состояние. Записанные изменения начинают влиять на устройство только после применения.

Получение полной конфигурации

Метод возвращает подготовленную (`candidate`) конфигурацию целиком:

```
GET /api/v1/config
```

Получение секции

Для чтения одной секции укажите её имя в пути:

```
GET /api/v1/config/{section}
```

Доступные секции: `system`, `network`, `ssl`, `monitors`, `pools`, `services`, `service-chains`, `virtual-servers`, `gslb`, `policy`, `egress`, `protocol`, `topology`, `dataplane`, `ha`, `logging`, `snmp`, `observability`, `users`, `rate-shaping`, `arules`, `url-categories`.

Секции `egress` (трансляция адреса источника для origin-соединений) и `protocol` (TCP/MSS-параметры) — верхнеуровневые: их ключи не входят в `dataplane.dpd`. Изменения этих секций применяются с перезапуском `dataplane`. В веб-интерфейсе такое применение обозначено как `cold apply`.

Обновление секции

Запрос заменяет указанную секцию в подготовленной конфигурации:

```
PUT /api/v1/config/{section}
```

Тело запроса — JSON-объект с полным содержимым секции.

Экспорт конфигурации

Метод выгружает подготовленную конфигурацию в файл JSON:

```
GET /api/v1/config/export
```


Возвращает полную конфигурацию в формате JSON-файла.

Импорт конфигурации

Импорт заменяет подготовленную конфигурацию содержимым файла JSON:

```
POST /api/v1/config/import
```

Передайте файл с типом `multipart/form-data`. Действующая конфигурация не изменится до отдельного применения.

Импорт F5 UCS

Метод преобразует поддерживаемые объекты UCS и помещает результат в подготовленную конфигурацию:

```
POST /api/v1/config/import-ucs
```

Передайте архив F5 BIG-IP UCS с типом `multipart/form-data`.

Сокращённый ответ включает преобразованную конфигурацию и отчёт:

```
{
  "config": {
    "system": {
      "hostname": "bigip1"
    }
  },
  "report": [
    { "severity": "info", "entity": "ssl/profile", "message": "..."},
    { "severity": "warning", "entity": "services/inline", "message": "..."}
  ]
}
```

Проверка конфигурации

Метод проверяет переданное содержимое без сохранения и применения:

```
POST /api/v1/config/validate
```

Применение и откат конфигурации

Перед применением сравните подготовленную и действующую конфигурации. После успешного применения подтвердите результат до окончания таймера, иначе Амибиг автоматически вернёт предыдущее состояние.

Просмотр изменений

Метод сравнивает подготовленную и действующую конфигурации:

```
GET /api/v1/config/diff
```

Ответ содержит признак изменений и оба состояния:

```
{
  "has_changes": true,
  "left": {
    "system": { "hostname": "amibig-old" }
  },
  "right": {
    "system": { "hostname": "amibig-new" }
  }
}
```

Применение конфигурации

Метод делает подготовленную конфигурацию действующей. По умолчанию после применения запускается таймер подтверждения:

```
POST /api/v1/config/apply
```

```
{
  "description": "Описание изменений",
  "skip_timer": false
}
```

Параметр	Тип	По умолчанию	Описание
description	string	""	Описание изменений (для истории)
skip_timer	bool	false	Не запускать таймер автоматического отката; используйте только при гарантированном доступе к устройству

Ответ при успехе:

```
{
  "success": true,
  "message": "Config applied successfully"
}
```

Возможные ошибки:

HTTP код	Ситуация
409	Блокировка лицензией (<code>license_check: true</code>)
409	Изменение отклонено проверкой безопасности применения (<code>safety_check: true</code>)
500	Ошибка применения конфигурации

Статус применения

Пока действует таймер, метод возвращает оставшееся время:

```
GET /api/v1/config/apply-status
```

```
{
  "pending_confirm": true,
  "seconds_left": 95
}
```

Подтверждение применения

Подтвердите доступность устройства и остановите таймер отката:

```
POST /api/v1/config/confirm
```

Останавливает таймер отката и фиксирует конфигурацию.

Откат конфигурации

Метод отменяет все неприменённые изменения и копирует действующую конфигурацию в подготовленную:

```
POST /api/v1/config/revert
```

Последовательность применения

Полный цикл с таймером подтверждения выглядит следующим образом:

1. `POST /api/v1/config/apply`
 - подготовленная конфигурация становится действующей
 - необходимые компоненты применяют изменения
 - запускается таймер отката (120 секунд)
2. `GET /api/v1/config/apply-status`
 - проверка: `pending_confirm=true`, `seconds_left=N`
3. `POST /api/v1/config/confirm`
 - таймер остановлен, конфигурация зафиксирована

ИЛИ: таймер истекает
→ автоматический откат к предыдущей конфигурации

События применения

Применение конфигурации фиксируется в событийном журнале (категория `audit / config`, коды `AMB-NNNN` — см. каталог кодов сообщений), в разделе История веб-интерфейса и SNMP trap-сообщениями `config-applied` и `config-failed`. При изменении секции `users` дополнительно записываются события по каждому созданному/удалённому пользователю и каждой смене роли — см. Управление пользователями.

Высокая доступность

Методы HA возвращают роль узла и состояние партнёра, а также позволяют запустить синхронизацию вручную. Изменяющие операции доступны только администратору.

Статус HA

Метод возвращает роль узла, состояние партнёра и время последней синхронизации:

```
GET /api/v1/ha/status
```

Ответ:

```
{
  "enabled": true,
  "role": "active",
  "peer-state": "up",
  "tls-active": true,
  "messages-sent": 12345,
  "messages-received": 12340,
  "last-heartbeat": "2026-02-26T10:00:01Z",
  "last-config-sync": "2026-02-26T10:00:00Z"
}
```

Ручная синхронизация

Запрос запускает внеочередную синхронизацию конфигурации с HA-партнёром:

```
POST /api/v1/ha/sync
```

Обновление системы

Для обновления, отката и перезагрузки используются следующие методы:

Адрес метода	Метод	Описание
/api/v1/system/update/upload	POST	Загрузка пакета .amibig
/api/v1/system/update/apply	POST	Применение к неактивному слоту
/api/v1/system/update/status	GET	Статус обновления
/api/v1/system/update/rollback	POST	Откат на другой слот
/api/v1/system/slots	GET	Информация о слотах A/B
/api/v1/system/reboot	POST	Перезагрузка системы
/api/v1/system/shutdown	POST	Штатное выключение с учётом роли HA

Подробности в разделе Обновление системы.

Сертификат HTTPS веб-интерфейса

Методы позволяют проверить и заменить сертификат интерфейса управления:

Адрес метода	Метод	Описание
/api/v1/system/webui-cert	GET	Сведения о текущем сертификате интерфейса управления
/api/v1/system/webui-cert.pem	GET	Скачивание текущего сертификата (PEM)
/api/v1/system/webui-cert/csr	POST	Создание CSR; закрытый ключ остаётся на устройстве
/api/v1/system/webui-cert	POST	Загрузка PEM (сертификат, закрытый ключ и цепочка) или PKCS#12
/api/v1/system/webui-cert/reset	POST	Откат к самоподписанному сертификату

Операции изменения требуют роли `admin`. После замены сертификата активные браузерные сессии разрываются.

Наблюдаемость и телеметрия

Состояние и накопленные показатели доступны через следующие группы методов:

Группа методов	Описание
GET /api/v1/telemetry*	Единый снимок телеметрии
GET /metrics	Prometheus/OpenMetrics
GET /api/v1/alerts*	Активные предупреждения, история, сводка
GET /api/v1/observability*	Накопленная аналитика (окна, серии, топы)

Подробности в разделах Телеметрия и метрики и Наблюдаемость.

Резервное копирование

Создание и восстановление ACF требуют роли `admin` :

Адрес	Метод	Описание
/api/v1/backup/export	GET	Экспорт архива <code>.acf</code>
/api/v1/backup/import	POST	Импорт архива <code>.acf</code>

Подробности в разделе Резервное копирование.

Лицензирование

Методы чтения доступны всем ролям, кроме отдельно указанного аппаратного ID:

Адрес	Метод	Роли	Описание
/api/v1/license/status	GET	admin, operator, viewer	Статус лицензии
/api/v1/license/hardware-id	GET	admin	Аппаратный ID устройства
/api/v1/license/features	GET	admin, operator, viewer	Лицензированные функции
/api/v1/license/upload	POST	admin	Загрузка <code>.lic</code> файла

Диагностика

Методы диагностики проверяют сервисы, объясняют обработку модельного соединения и управляют захватом пакетов. После запуска захвата используйте полученный идентификатор `id` как `capture_id` в последующих запросах.

Адрес	Метод	Описание
<code>/api/v1/config/check-icap</code>	POST	Проверка доступности ICAP-сервера
<code>/api/v1/config/check-inline</code>	POST	Проверка inline-сервиса
<code>/api/v1/diagnostics/interfaces</code>	GET	Список интерфейсов, доступных для захвата
<code>/api/v1/diagnostics/flow-explain</code>	POST	Объяснение обработки модельного соединения
<code>/api/v1/diagnostics/pcap</code>	POST	Запуск захвата пакетов
<code>/api/v1/diagnostics/pcap</code>	GET	Список захватов
<code>/api/v1/diagnostics/pcap/{capture_id}</code>	GET	Состояние захвата
<code>/api/v1/diagnostics/pcap/{capture_id}/stop</code>	POST	Остановка захвата
<code>/api/v1/diagnostics/pcap/{capture_id}/download</code>	GET	Скачивание исходного PCAP-файла
<code>/api/v1/diagnostics/pcap/{capture_id}/download?enrich=true</code>	GET	Скачивание PCAP-файла с метаданными Амибиг
<code>/api/v1/diagnostics/dissector</code>	GET	Скачивание Lua-диссектора метаданных для Wireshark
<code>/api/v1/diagnostics/pcap/{capture_id}</code>	DELETE	Удаление захвата и его файлов
<code>/api/v1/diagnostics/archive</code>	POST	Создание диагностического архива

При запросе с `enrich=true` API добавляет к пакетам трейлеры с метаданными Амибиг. Для их декодирования установите в Wireshark Lua-диссектор, доступный отдельным методом API.

Обработка ошибок

API использует стандартные HTTP-коды состояния:

Код	Описание
200	Успех
400	Некорректный запрос (неверные данные)
401	Не аутентифицирован (нет токена или токен истёк)
403	Недостаточно прав (роль не позволяет операцию) или лицензия истекла
404	Ресурс не найден
409	Операция отклонена лицензией или проверкой безопасности применения
422	Ошибка проверки данных, включая нарушение требований к паролю
500	Внутренняя ошибка сервера

Тело ошибки содержит поле `detail` с описанием:

```
{
  "detail": "Invalid username or password"
}
```

Для ошибок валидации пароля:

```
{
  "detail": {
    "password_policy_errors": {
      "admin": ["password.too_short", "password.complexity"]
    }
  }
}
```

Примеры использования

Примеры объединяют отдельные методы в законченные административные операции. Перед запуском замените адрес устройства и учётные данные, а после выполнения проверьте код ответа каждого запроса.

Полный цикл: получение токена, изменение, применение, подтверждение

Пример ниже добавляет изменение в подготовленную конфигурацию, применяет и подтверждает его. Подставьте доверенный адрес интерфейса управления:

```
### 1. Аутентификация
TOKEN=$(curl -s -X POST "https://<management-ip>/api/v1/auth/login" \
  -H "Content-Type: application/json" \
  -d '{"username":"admin","password":"Adm1n!pass"}' | jq -r '.token')
```



```

### 2. Получение текущей сетевой конфигурации
curl -s "https://<management-ip>/api/v1/config/network" \
  -H "Authorization: Bearer $TOKEN" | jq

### 3. Обновление секции (пример: добавление VLAN)
curl -X PUT "https://<management-ip>/api/v1/config/network" \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{ ... }'

### 4. Просмотр diff
curl -s "https://<management-ip>/api/v1/config/diff" \
  -H "Authorization: Bearer $TOKEN" | jq '.has_changes'

### 5. Применение
curl -X POST "https://<management-ip>/api/v1/config/apply" \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"description": "Add VLAN 100"}'

### 6. Проверка статуса
curl -s "https://<management-ip>/api/v1/config/apply-status" \
  -H "Authorization: Bearer $TOKEN"

### 7. Подтверждение
curl -X POST "https://<management-ip>/api/v1/config/confirm" \
  -H "Authorization: Bearer $TOKEN"

```

Управление пользователями

Амибиг использует локальные учётные записи и ролевую модель доступа. Роль определяет права пользователя, а список доступа к управлению ограничивает подключения к веб-интерфейсу и REST API по адресу источника.

Роли

Система предусматривает три роли с нисходящим уровнем прав:

Роль	Описание
admin	Полный доступ. Управление конфигурацией, пользователями, лицензиями, обновлениями, резервным копированием
operator	Просмотр и редактирование конфигурации. Не может применять, управлять пользователями, обновлять систему
viewer	Только просмотр. Не может вносить изменения

Матрица прав

В таблице показаны права каждой встроенной роли:

Операция	admin	operator	viewer
Просмотр конфигурации	+	+	+
Просмотр статуса и мониторинга	+	+	+
Просмотр логов и истории	+	+	+
Редактирование конфигурации	+	+	—
Применение конфигурации	+	—	—
Подтверждение и откат	+	—	—
Управление пользователями	+	—	—
Резервное копирование	+	—	—
Импорт F5 UCS	+	—	—
Обновление системы	+	—	—
Перезагрузка	+	—	—
Загрузка лицензии	+	—	—
Ручная синхронизация HA	+	—	—

Первоначальная учётная запись

При первой установке система создаёт учётную запись администратора:

Поле	Значение
Логин	admin
Пароль	случайный начальный пароль — выводится в сообщении консоли и сохраняется в <code>/config/.initial-admin-password</code>
Роль	admin

При первом входе система принудительно потребует смену пароля; после смены файл с начальным паролем удаляется.

Управление пользователями через веб-интерфейс

Изменения учётных записей сначала сохраняются в подготовленной конфигурации. Чтобы они вступили в силу, завершите операцию применением конфигурации.

Создание пользователя

Чтобы добавить локальную учётную запись:

1. Перейдите на экран **Пользователи** в боковом меню.
2. Нажмите **Добавить пользователя**.
3. Заполните поля:
 - имя пользователя (уникальное);
 - роль (**admin** , **operator** , **viewer**);
 - пароль (должен соответствовать политике);
 - необязательное описание.
4. Нажмите **Сохранить**.
5. Примените конфигурацию.

Редактирование пользователя

Чтобы изменить роль, описание или пароль:

1. На экране **Пользователи** выберите учётную запись для редактирования.
2. Измените необходимые поля.
3. Для смены пароля введите новый пароль в поле **Пароль**.
4. Нажмите **Сохранить**, затем примените конфигурацию.

Удаление пользователя

Перед удалением убедитесь, что в системе останется хотя бы одна учётная запись **admin** с паролем:

1. На экране **Пользователи** нажмите кнопку удаления у соответствующей учётной записи.
2. Подтвердите удаление.
3. Примените конфигурацию.

Управление пользователями через API

REST API предоставляет те же операции с учётными записями, что и веб-интерфейс. Ответы не содержат пароли и их хеши.

Получение списка пользователей

Метод возвращает учётные записи без открытых паролей:

```
GET /api/v1/config/users
```

Обновление пользователей

Запрос заменяет список пользователей в подготовленной конфигурации:

```
PUT /api/v1/config/users
```

```
{
  "user": [
    {
      "name": "admin",
      "role": "admin",
      "description": "System administrator"
    },
    {
      "name": "operator1",
      "role": "operator",
      "password": "Oper@t0r!pass",
      "description": "Network operator"
    },
    {
      "name": "auditor",
      "role": "viewer",
      "password": "Aud!t!view",
      "description": "Security auditor"
    }
  ]
}
```

Передавайте поле `password` только по HTTPS. API вычисляет хеш и не сохраняет открытый пароль.

Политика паролей

Система применяет строгую политику паролей для всех учётных записей.

Требования к паролю

Новый пароль должен одновременно соответствовать всем требованиям:

Параметр	Значение	Описание
Минимальная длина	8 символов	Пароль короче 8 символов отклоняется
Сложность	3 из 4 категорий	Требуется наличие символов минимум из трех категорий

Параметр	Значение	Описание
Совпадение с логином	Запрещено	Пароль не может быть равен имени пользователя
Общеизвестные пароли	Запрещены	Проверка по встроенному словарю распространенных паролей
Повторное использование	Запрещено	Нельзя использовать 4 последних пароля
Подтверждение	Обязательно	При смене пароля требуется повторный ввод

Категории сложности

Пароль должен содержать символы минимум из **трех** из четырех категорий:

Категория	Примеры
Строчные буквы	a - z
Прописные буквы	A - Z
Цифры	0 - 9
Специальные символы	!@#\$%^&* () и др.

Срок действия пароля

Срок и глубина истории заданы для всех локальных учётных записей:

Параметр	Значение	Описание
Срок действия	40 дней	По истечении пользователь обязан сменить пароль
Глубина истории	4 пароля	Нельзя повторно использовать последние 4 пароля

При входе с истекшим паролем система перенаправляет пользователя на страницу смены пароля. Доступ к остальным функциям блокируется до смены.

Коды ошибок политики

При нарушении политики API возвращает HTTP 422 со списком ошибок:

Код ошибки	Описание
<code>password.too_short</code>	Пароль короче минимальной длины (8 символов)
<code>password.complexity</code>	Недостаточная сложность (менее 3 из 4 категорий)
<code>password.equals_username</code>	Пароль совпадает с именем пользователя
<code>password.common</code>	Пароль найден в словаре распространённых
<code>password.reused</code>	Пароль совпадает с одним из 4 предыдущих
<code>password.mismatch</code>	Пароль и подтверждение не совпадают

События аудита

Действия с учётными записями и сеансами фиксируются в событийном журнале с кодами AMB-NNNN (категория `auth / audit`):

Событие	Код	Когда записывается
Успешный вход	AMB-2004	пользователь вошёл в систему
Выход из системы	AMB-2005	пользователь завершил сеанс
Неудачная попытка входа	AMB-2002	неверные учётные данные
Блокировка входа	AMB-2003	серия неудачных попыток — вход для адреса временно заблокирован
Ограничение частоты	AMB-2001	попытка входа отклонена ограничителем частоты
Смена пароля	AMB-2007 / AMB-2006	пароль изменён / смена не удалась

При применении изменений секции `users` через API или веб-интерфейс дополнительно записываются события по каждому изменению: создание пользователя, удаление пользователя, смена роли (с указанием старой и новой роли и автора изменения). Пароли, их хеши и история паролей в события не попадают.

События доступны на экране `Просмотр логов`, экспортируются в `syslog` и `Kafka` (см. `Наблюдаемость`) и описаны в каталоге кодов сообщений.

Список доступа к управлению

Список доступа к управлению (`management-acl`) ограничивает подключения к веб-интерфейсу и REST API по IP-адресам источника.

Параметры ACL

Каждая запись содержит сеть источника, необязательные порты и действие:

Поле	Тип	Описание
<code>source</code>	string	IP-адрес или подсеть в формате CIDR (<code>x.x.x.x/y</code>)
<code>ports</code>	list[int]	Список портов (пустой = все порты)
<code>action</code>	string	<code>allow</code> — разрешить, <code>deny</code> — запретить

Пример конфигурации

Следующий список разрешает управление только из двух административных подсетей:

```
{
  "user": [
    { "name": "admin", "role": "admin" }
  ],
  "management-acl": [
    { "source": "10.0.0.0/8", "action": "allow" },
    { "source": "192.168.1.0/24", "action": "allow" },
    { "source": "0.0.0.0/0", "action": "deny" }
  ]
}
```

В этом примере доступ к веб-интерфейсу и API разрешён только из подсетей 10.0.0.0/8 и 192.168.1.0/24. Все остальные подключения блокируются.

Порядок применения правил

Правила проверяются сверху вниз до первого совпадения. Если список содержит записи, но ни одна не совпала, доступ запрещается. Пустой список не ограничивает доступ.

Настройка через веб-интерфейс

Добавляйте разрешающие правила до запрещающих:

1. Перейдите на экран `Пользователи` .
2. В секции `ACL управления` добавьте правила.

3. Для каждого правила укажите:

- источник — IP-адрес или подсеть;
- действие — разрешить или запретить.

4. Расположите правила в нужном порядке.

5. Примените конфигурацию.

Проверка перед применением

Перед применением система проверяет, разрешает ли новый список текущий адрес и порт администратора. Если нет, применение блокируется. Эта проверка не заменяет резервный доступ через консоль устройства.

Хранение учётных данных

Пароли хранятся как PBKDF2-HMAC-SHA256 с индивидуальной случайной солью и 100 000 итераций. Открытые значения не сохраняются. Для запрета повторного использования система хранит хеши четырёх предыдущих паролей; при импорте конфигурации история сохраняется.

Идентификация устройства и лицензирование

На экране **Управление** → **Лицензия** отображаются идентификаторы устройства и состояние лицензии:

- **идентификация устройства** — серийный номер и отпечаток оборудования, привязанные к устройству и его разделу `/config`;
- **лицензия** — подписанный файл с разрешёнными функциями и ограничениями, привязанный к отпечатку оборудования.

[illegible]

Идентификация устройства

Карточка в верхней части страницы заполняется автоматически при первой загрузке. Она содержит следующие поля:

Поле	Значение
Серийный номер	<code>AMB-PYYWW-RAND10-CHK2</code> для временного номера или <code>AMB-YYWW-SEQ8-CHK2</code> для заводского
Тип серийного номера	Временный (полевой) или Заводской
Отпечаток оборудования	<code>sha256:<64 hex></code> — значение, рассчитанное по DMI, WWN загрузочного диска и <code>machine-id</code>
Device UUID	UUID v4, привязанный к разделу <code>/config</code>
Создан	дата и время создания в UTC

Временный и заводской серийные номера

Если у устройства нет заводской записи, при первой загрузке создаётся временный серийный номер с префиксом `АМВ-Р`. Веб-интерфейс показывает для него предупреждение:

Этот серийный номер сгенерирован в полевом режиме (до запуска заводской регистрации). Для получения постоянной лицензии обратитесь в техническую поддержку Amicon, передав серийный номер и отпечаток оборудования.

Заводской серийный номер не содержит префикс `Р` и регистрируется при производстве устройства.

Хранение идентификаторов

Идентификаторы хранятся в `/config/identity/device-identity.json`. Обновление A/B и очистка раздела `/data` не изменяют этот файл. Очистка `/config` создаёт новый временный серийный номер и новый отпечаток оборудования; прежняя лицензия после этого не действует.

Влияние операций обслуживания

Перед заменой компонентов или очисткой разделов проверьте последствия для лицензии:

Событие	Серийный номер	Отпечаток	Что нужно сделать
Перезагрузка	сохраняется	сохраняется	действий не требуется
Обновление A/B	сохраняется	сохраняется	действий не требуется
Очистка <code>/data</code>	сохраняется	сохраняется	действий не требуется
Очистка <code>/config</code>	новый	новый	заново запросить лицензию
Замена системной платы	сохраняется	меняется	перепривязать лицензию через техническую поддержку Amicon

Событие	Серийный номер	Отпечаток	Что нужно сделать
Замена сетевой карты	сохраняется	сохраняется	действий не требуется: сетевой адаптер не входит в отпечаток
Замена загрузочного диска	сохраняется	меняется	перепривязать лицензию через техническую поддержку Amicon

Получение лицензии для временного серийного номера

Для запроса и установки лицензии выполните следующие действия:

1. Дождитесь первой загрузки устройства и откройте экран **Управление** → **Лицензия**.
2. **Скопируйте** серийный номер и отпечаток оборудования кнопками **Копировать** рядом с полями идентификации устройства.
3. Передайте в техническую поддержку Amicon серийный номер **АМВ-Р...**, отпечаток **sha256:...**, требуемые функции и срок действия лицензии.
4. Получите подписанный файл **.lic**.
5. Загрузите файл в блоке **Загрузка лицензии**. Сервис проверит подпись Ed25519 по доверенному открытому ключу и привязку к оборудованию, затем вернёт рассчитанное состояние лицензии. Истёкший срок подписки причиной отказа **не является** — такая лицензия загружается и продолжает действовать (см. «Работа после истечения подписки»).
6. Убедитесь, что статус изменился на **Активна** или ожидаемое состояние подписки и что в карточке функций указаны приобретённые ограничения.

Состояние лицензии

Карточка показывает текущее состояние лицензии и действующие ограничения. В модели лицензии Амибиг срок подписки отделён от права эксплуатации: истечение подписки не останавливает устройство и не блокирует администрирование (подробнее — в разделе «Работа после истечения подписки»).

Статус	Что означает
Активна	Лицензия прошла проверку; доступны функции и ограничения из файла лицензии
Демо	Действует демонстрационный режим на 30 дней с ограниченным набором функций
Льготный период	Подписка истекла, идёт льготный период (в поле Осталось дней показан его остаток)

Статус	Что означает
Подписка истекла	Подписка и льготный период закончились. Устройство продолжает работать и администрироваться; нельзя добавлять объекты сверх приобретённых ограничений
Истекла	Закончился демо-период или срок права эксплуатации — операции записи блокируются. Купленной лицензии с истёкшей подпиской это не касается
Недействительна	Подпись не прошла проверку или лицензия выдана для другого оборудования

Веб-интерфейс показывает уведомление в верхней части каждой страницы с кнопкой **Управление лицензией** (переход на эту страницу) в следующих случаях:

- статус **Истекла**, **Подписка истекла** или **Льготный период**;
- до истечения подписки остаётся 30 дней и меньше;
- демо-режим, до конца демо-периода остаётся 7 дней и меньше.

Если в лицензии срок подписки не задан (поле `expires_at` пустое), подписка считается бессрочной: уведомление об истечении не выводится, а в полях **Истекает** и **Осталось дней** показывается **Бессрочно**.

Работа после окончания подписки

Срок подписки в лицензии отделён от бессрочного права эксплуатации устройства. Когда подписка и льготный период закончились, Амибиг сохраняет полную работоспособность: трафик обрабатывается, конфигурацию можно изменять и применять, все оплаченные функции продолжают действовать. Недоступным остаётся только рост сверх оплаченных лимитов — например, нельзя добавить интерфейсы или сервисные цепочки сверх разрешённого лицензией количества.

Право эксплуатации задаётся полем `operation_expires_at` в лицензии. Пустое или отсутствующее значение означает бессрочное право; лицензии прежнего формата, выданные без этого поля, остаются действительными. То же правило действует для срока подписки: пустое `expires_at` означает бессрчную подписку. Демо-режим строже: без файла лицензии через 30 дней операции записи блокируются полностью.

Оповещения формируются заранее: если до окончания подписки осталось не более 30 дней, веб-интерфейс показывает уведомление. За 30, 14 и 7 дней в журнал записывается событие и отправляется SNMP trap-сообщение `license-fault`. Событие и trap-сообщение отправляются при **смене** состояния, а не при каждом рестарте: перезагрузка устройства или обновление не порождает дубликат предупреждения для состояний **Льготный период**, **Подписка истекла** и «истекает».

через N дней». Состояния `Истекла` и `Недействительна`, при которых администрирование ограничено, сообщаются и при первом опросе после запуска.

Обновляемых подписных баз в текущей версии нет, поэтому истечение подписки не снижает функциональность: устройство продолжает работать на накопленных локальных данных и действующей конфигурации.

Лицензированные функции

Показывает разрешённые функции и числовые лимиты. Колонка `Значение` — зелёный тег `Включено` / серый `Отключено` / число для лимитов (например `Макс. пропускная способность: 1 Гбит/с`).

В демонстрационном режиме действуют ограничения: 1 Гбит/с, одна цепочка сервисов и четыре сетевых интерфейса. После установки лицензии применяются значения из подписанного файла.

Аппаратный ID

Отдельный SHA-256 (устаревший идентификатор; используется для вывода LUKS-ключа защиты диска и для совместимости с лицензиями ранних сборок). Не путать с **отпечатком оборудования** в карточке идентификации устройства: это разные артефакты, рассчитанные по разным правилам.

Кнопка `Копировать` помещает значение в буфер обмена, `Обновить` — пересчитывает из текущего оборудования.

Поведение при ошибке валидации лицензии

После загрузки `.lic` сервер возвращает HTTP 200 и то же состояние лицензии, что и `GET /api/v1/license/status`. Отдельного поля с причиной отказа в ответе нет — состояние читается по набору полей:

Поля ответа	Что произошло	Действие оператора
<code>tier: "invalid", hard_expired: true</code>	Подпись не прошла проверку ни одним доверенным ключом или файл повреждён	Запросить новый файл — старый мог быть подменён
<code>hardware_id_match: false, hard_expired: true</code>	Отпечаток в лицензии не совпал с текущим	Перепривязать лицензию через техническую поддержку Amicon; это может потребоваться после замены системной платы или загрузочного диска
<code>subscription_expired: true, valid: true</code>	Срок подписки истёк, право эксплуатации сохраняется	Продлить подписку; работа и администрирование не прерываются

Поля ответа	Что произошло	Действие оператора
<code>grace_period: true</code>	Срок подписки истёк, идёт льготный период	Продлить подписку до его окончания
<code>hard_expired: true</code> при <code>demo_mode: true</code>	Демо-период закончился	Загрузите действующую лицензию

Если `.lic` не разбирается как JSON или не соответствует схеме, API возвращает HTTP 400. Файл с недействительной подписью, несовпадающей привязкой к оборудованию или истёкшим правом эксплуатации не заменяет действующую лицензию. В ответе возвращается состояние отклонённого файла. Истечение срока подписки не препятствует установке лицензии.

Сброс идентификаторов устройства

Сброс используйте только по указанию технической поддержки Amicon. Команда требует доступа к консоли устройства:

```
sudo amibig-identity reset --confirm
```

Команда переносит `/config/identity/device-identity.json` в `.reset-backup`. При следующей загрузке система создаёт новые идентификаторы. Действующая лицензия после сброса становится недействительной; до выполнения команды согласуйте выпуск новой лицензии.

В веб-интерфейсе эта операция недоступна.

Мониторинг по SNMP и trap-сообщения

Амибиг публикует состояние устройства по SNMP: внешняя система мониторинга может опрашивать поддерево производителя и получать trap-сообщения о событиях. SNMP-агент читает те же данные, что и `GET /api/v1/telemetry` и `GET /metrics`, — отдельного источника состояния у него нет.

Ключи конфигурации

Секция `snmp` использует следующие ключи:

- `communities`
- `v3-users`
- `trap-targets`

Пример:

```
{
  "snmp": {
    "enabled": true,

```

```

        "contact": "noc@example.test",
        "location": "DC-1",
        "communities": [
            { "name": "monitoring-ro", "access": "ro", "source": "192.0.2.0/24" }
        ],
        "v3-users": [
            {
                "name": "operator",
                "auth-protocol": "sha-256",
                "auth-password": "<пароль>",
                "priv-protocol": "aes-256",
                "priv-password": "<пароль>",
                "access": "ro"
            }
        ],
        "trap-targets": [
            { "name": "nms-v2", "address": "192.0.2.10", "port": 162, "version": "v2c", "community": "public" },
            { "name": "nms-v3", "address": "192.0.2.20", "port": 162, "version": "v3", "community": "operator" }
        ]
    }
}

```

Для `v3` поле `trap-targets[].community` используется как селектор `security-name` и должно совпадать с именем из `v3-users`.

Для рабочих контуров используйте SNMPv3 с аутентификацией и шифрованием. Если системе мониторинга требуется SNMPv2c, ограничьте `source` её подсетью и не используйте стандартное имя `community`.

Enterprise OID

Поддерево Амибиг:

```
.1.3.6.1.4.1.61803
```

Основные группы опроса:

- `.1` — системные скаляры: версия, `hostname`, `contact`, `location`;
- `.2` — сводка по сервисам: всего, `up`, `degraded`, `down`, строки сервисов;
- `.3` — таблица участников пулов: состояние и счётчики соединений;
- `.4` — НА: включён ли и текущая роль;
- `.5` — интерфейсы: имя, состояние, `rx/tx` байты, ошибки, потери;
- `.6` — виртуальные серверы: имя, адрес, порт, доступность и число участников пулов;
- `.7` — скаляры устройства и TLS: активные и суммарные соединения, байтовые и TLS-счётчики.

Кодирование статусов при опросе фиксировано:

- `up = 1`
- `down = 2`
- `degraded = 3`

- `unknown = 4`

Машиночитаемый MIB-файл

Файл `AMIBIG-MIB.mib` — это формальное SMIV2-описание enterprise-поддерживаемого Амибиг. Он содержит дерево объектов опроса (системные скаляры, сводку по сервисам, участников пулов, HA, интерфейсы, виртуальные серверы, счётчики устройства и TLS), все 16 trap-сообщений и поля, которые передаются в уведомлениях. После импорта файла система мониторинга показывает имена объектов и текст уведомлений вместо числовых OID.

Получить файл можно двумя способами:

- кнопкой «Скачать MIB-файл» в разделе SNMP веб-интерфейса;
- на устройстве файл установлен по стандартному пути `net-snmp /usr/share/snmp/mibs/AMIBIG-MIB.mib`.

Проверить, что файл принят инструментами net-snmp, можно командой:

```
snmptranslate -m +AMIBIG-MIB -On AMIBIG-MIB::amibigMIB
```

Команда должна вывести числовой OID `.1.3.6.1.4.1.61803`.

Пример snmpwalk

Для SNMPv2c укажите настроенное имя community и адрес устройства:

```
snmpwalk -v2c -c monitoring-ro 192.0.2.50 .1.3.6.1.4.1.61803
```

Поддерживаемое строится из общего снимка телеметрии. Если REST-телеметрия временно недоступна, агент продолжает отдавать последнее валидное дерево до следующего окна обновления.

Trap-сообщения

Trap-сообщения отправляются из поддерживаемого уведомлений `.1.3.6.1.4.1.61803.0`:

Trap-сообщение	OID	Когда отправляется
<code>service-down / service-up</code>	<code>.0.1 / .0.2</code>	смена состояния сервиса
<code>ha-failover</code>	<code>.0.3</code>	переключение роли HA
<code>config-applied / config-failed</code>	<code>.0.4 / .0.12</code>	применение конфигурации
<code>system-boot</code>	<code>.0.5</code>	загрузка системы
<code>vip-down / vip-up</code>	<code>.0.6 / .0.7</code>	смена состояния VIP

Trap-сообщение	OID	Когда отправляется
auth-failure	.0.8	ошибка аутентификации
license-fault	.0.9	проблема лицензии
storage-fault	.0.10	проблема хранилища
tls-fault	.0.11	сбой TLS
alert-open / alert-clear	.0.13 / .0.14	открытие/закрытие предупреждения наблюдаемости
link-down / link-up	.0.15 / .0.16	смена состояния физического соединения порта dataplane

Семантика:

- trap-сообщения сервисов, VIP и HA формируются при обнаружении переходов состояния над общим снимком телеметрии;
- trap-сообщения аутентификации, конфигурации, лицензии, хранилища и TLS формируются из структурированных событий;
- предупреждения наблюдаемости публикуются как `alert-open` и `alert-clear` (см. Наблюдаемость);
- в сообщениях `link-down` и `link-up` имя порта передаётся в переменной `target`; события отправляются и для портов-участников агрегированных каналов;
- приёмники `v2c` и `v3` читаются из конфигурации устройства.

Пример приёмника trap-сообщений

Приёмники задаются в списке `trap-targets`:

```
{
  "snmp": {
    "enabled": true,
    "trap-targets": [
      { "name": "noc", "address": "192.0.2.10", "port": 162, "version": "v2c", "community": "public" }
    ]
  }
}
```

После применения конфигурации trap-сообщения начинают отправляться автоматически — дополнительная ручная настройка на устройстве не требуется.

Связанные разделы

Форматы телеметрии и журналирования описаны в следующих разделах:

- Наблюдаемость: предупреждения, аналитика и экспорт

- Телеметрия и метрики
- Журналирование и просмотр логов

Наблюдаемость: предупреждения, аналитика и экспорт

Подсистема наблюдаемости отвечает на три вопроса администратора: что происходит с устройством сейчас, что происходило с ним раньше и как передать эти данные во внешние системы мониторинга. Настройки собраны в секции `observability` конфигурации и на экране `Наблюдаемость` веб-интерфейса; экспорт событийного журнала настраивается на экране `Логирование` (секция `logging.export`).

В подсистему входят:

- секция `observability` конфигурации;
- API предупреждений `GET /api/v1/alerts*`;
- API накопленной аналитики `GET /api/v1/observability*`;
- экран `Наблюдаемость` в веб-интерфейсе — предупреждения, динамика показателей, SIEM-приёмники журнала соединений и выборка записей;
- маршрутизация предупреждений в локальный журнал, syslog и SNMP;
- передача журнала соединений в SIEM-приёмники;
- экспорт событийного журнала в syslog и Kafka;
- машиночитаемые данные наблюдаемости внутри диагностического архива.

Конфигурация `observability`

Пример секции с передачей журнала соединений в SIEM-приёмник:

```
{
  "observability": {
    "profile": "balanced",
    "alerts": {
      "enabled": true,
      "default-routes": ["local-log", "snmp", "syslog"],
      "rule": [
        {
          "name": "device-throughput",
          "type": "throughput",
          "scope": "device",
          "warning-threshold": 1000000,
          "critical-threshold": 2000000,
          "clear-threshold": 800000,
          "window-seconds": 300,
          "for-seconds": 60,
          "clear-for-seconds": 120,
          "enabled": true
        }
      ]
    },
    "history": {
      "enabled": true,

```

```

    "sample-period-seconds": 15,
    "raw-retention-minutes": 60
  },
  "flow-export": {
    "enabled": true,
    "mode": "cef",
    "receivers": [
      {
        "name": "siem-main",
        "host": "192.0.2.50",
        "port": 514,
        "transport": "udp",
        "format": "cef",
        "enabled": true
      }
    ],
    "sampling": { "mode": "all", "rate": 1, "adaptive-threshold-cps": 50000 }
  },
  "retention": {
    "events-days": 90,
    "alert-history-days": 90,
    "alert-history-max-rows": 20000
  }
}

```

Профили работы (`observability.profile`)

Профиль задаёт готовый набор значений по умолчанию для предупреждений, истории и экспорта. Явно заданные значения внутри `alerts`, `history`, `flow-export` и `support-bundle` переопределяют значения профиля.

- `minimal_telemetry` — минимальная нагрузка на устройство: предупреждения и история выключены, остаётся только базовая телеметрия.
- `balanced` — значение по умолчанию: предупреждения и история включены, передача журнала соединений выключена, пока администратор явно не задаст приёмники.
- `full_diagnostics` — максимальная детализация для отладки и нагрузочных испытаний; уменьшает запас производительности.

Предупреждения (alerts)

Правила предупреждений оцениваются поверх накопленной истории и текущего состояния. Каждое правило имеет пороги `warning` / `critical` / `clear`, окно оценки и выдержку времени до открытия и закрытия.

Поддерживаемые типы правил:

Тип	Что контролирует
<code>throughput</code>	пропускная способность (байт/с)
<code>connection-rate</code>	скорость установления соединений
<code>member-flaps</code>	частые смены состояния участников пулов

Тип	Что контролирует
<code>ha-health</code>	оценка состояния узла HA
<code>disk-pressure</code>	заполнение диска
<code>certificate-expiry</code>	приближение срока истечения сертификатов
<code>cpu</code>	загрузка CPU устройства
<code>memory</code>	использование памяти
<code>port-exhaustion</code>	исчерпание портов источника SNAT на origin-соединениях

Маршруты доставки (`routes` , по умолчанию — `alerts.default-routes`): `local-log` (локальный журнал событий), `snmp` (trap-сообщения `alert-open` и `alert-clear`), `syslog` (внешние syslog-приёмники).

Предустановленные правила

В начальной конфигурации устройства задан набор правил: `default-disk-pressure` , `default-certificate-expiry` , `default-ha-health` , `default-member-flaps` , `default-cpu` , `default-memory` . Предустановленный набор хранится в конфигурации: правила видны через `GET /api/v1/config` , а редактируются через `PUT /api/v1/config/{section}` или веб-интерфейс. Маркер `alerts.preset-version` защищает намеренно опустошённый список правил от повторного создания при перезапуске.

API предупреждений

Для сводки, активных предупреждений и истории доступны следующие методы:

- `GET /api/v1/alerts/summary` , `GET /api/v1/alerts/active` , `GET /api/v1/alerts/history` , `GET /api/v1/alerts`

Накопленная аналитика

История метрик хранится на устройстве с прореживанием по окнам `5m` , `1h` , `24h` , `7d` , `30d` (`observability.history`).

Основные методы API:

- `GET /api/v1/observability/windows`
- `GET /api/v1/observability/summary`
- `GET /api/v1/observability/series`
- `GET /api/v1/observability/top`
- `GET /api/v1/observability/ha/timeline`
- `GET /api/v1/observability/tls/failures`

- `GET /api/v1/observability/certificates`

Исторические графики на экране `Наблюдаемость`

Экран `Наблюдаемость` показывает накопленную историю отдельным графиком по каждой метрике за выбранное окно:

- скорость установления соединений (CPS);
- установленные TCP-соединения;
- инспектируемые HTTP(s)-соединения и новые HTTP(s)-соединения в секунду;
- запросы в секунду (RPS);
- загрузка интерфейсов: пропускная способность в Гбит/с и пакеты в секунду — с выбором конкретного интерфейса или суммарного значения;
- использование памяти в МБ;
- загрузка CPU в процентах.

Если для метрики настроено правило предупреждений (например, `cpu`, `memory`, `throughput` или `connection-rate`), поверх её графика отображаются горизонтальные линии порогов `warning` и `critical` — так видно, насколько текущие значения близки к срабатыванию предупреждения.

Помимо трафиковых метрик, в истории интерфейсов накапливаются счётчики потерь на уровне сетевой карты: `imissed` (пакеты, не принятые из-за переполнения очередей), `ierrors` (ошибки приёма) и `rx_nombuf` (нехватка буферов). Их скорость доступна через `GET /api/v1/observability/series` с `metric=imissed_rate`, `ierrors_rate` или `rx_nombuf_rate` и `scope=interface` — это первый инструмент диагностики потерь пакетов под нагрузкой.

Передача журнала соединений в SIEM (`observability.flow-export`)

Журнал содержит по одной записи на завершённое соединение с вердиктом политики, байтовыми счётчиками и TLS-деталью. Экспорт настраивается на экране `Наблюдаемость` в секции «Экспорт flow-лога в SIEM» или через секцию `observability.flow-export`.

Kafka

Включено: ☐

Брокеры:

Топик:

Формат:

Защита подключения к брокеру:

Категории событий:

Уровни важности:

Коды сообщений:

Назначения Syslog

[+ Добавить](#)

Имя	Адрес	Порт	Протокол	Формат
Пока пусто — добавьте первый элемент кнопкой «+ Добавить»				

Экспорт flow-лога в SIEM

Куда и в каком формате уезжает журнал соединений: приёмники и режим сэмплинга. **Неактивен**

☐ Включить экспорт flow-лога [Добавить приёмник](#)

Имя	Хост	Порт	Транспорт	Формат	Включён
Приёмники не заданы — экспорт неактивен.					

Сэмплирование

Режим: Делитель N: Порог adaptive, соединений/с:

[Сбросить](#) [Отмена](#) [Сохранить](#)

Поддерживаемые режимы (`mode`): `cef` , `rfc5424` , `json-lines` , `ipfix` ; режим `sflow` сохранён для совместимости с существующими конфигурациями.

Каждая запись содержит:

- адреса и порты источника и назначения, а также протокол (в CEF — поля `src` , `spt` , `dst` и `dpt`);
- код сообщения `AMB-NNNN` (см. каталог кодов);
- вердикт политики, имя правила и цепочки сервисов;
- детали X.509-сертификата сервера и JA3-отпечаток клиента для TLS-соединений.

Приёмники (`receivers`)

Экспорт активен, только когда `enabled: true` и включён хотя бы один приёмник. Поля приёмника:

Поле	Значение
<code>name</code>	имя приёмника
<code>host</code> , <code>port</code>	адрес и порт SIEM-приёмника
<code>transport</code>	<code>udp</code> , <code>tcp</code> или <code>tls</code>

Поле	Значение
<code>format</code>	<code>cef</code> , <code>rfc5424</code> , <code>json-lines</code> , <code>ipfix</code>
<code>enabled</code>	включён ли приёмник
<code>ca-trust</code>	путь к набору сертификатов УЦ (обязателен при <code>tls</code>)
<code>server-host</code>	ожидаемое имя сервера в сертификате (обязательно при <code>tls</code>)

Для транспорта `tls` поля `ca-trust` и `server-host` обязательны — без них сертификат приёмника не будет проверен, и конфигурация отклоняется.

Выборка записей (`sampling`)

При высокой интенсивности соединений выборка ограничивает нагрузку от передачи данных:

Поле	Значение
<code>mode</code>	<code>all</code> — каждая запись; <code>1-in-n</code> — каждая N-я; <code>aggregates</code> — только агрегаты
<code>rate</code>	N для режима <code>1-in-n</code>
<code>adaptive-threshold-cps</code>	порог скорости соединений, после которого экспорт автоматически переходит с <code>all</code> на прореживание

В режиме `all` передаются все записи до достижения порога CPS, после чего включается выборка. Значение порога подберите по результатам нагрузочных испытаний и пропускной способности приёмника.

Экспорт событийного журнала: `syslog` и `Kafka` (`logging.export`)

События устройства (аудит, система, обновления — все с кодами `AMB-NNNN`) передаются независимо от журнала соединений через секцию `logging.export`. Помимо `syslog`-форматов RFC 3164/5424 поддерживаются форматы SIEM-систем CEF и LEEF. Два типа приёмников:

- **syslog** — список назначений; форматы `rfc3164`, `rfc5424`, `cef`, `leef`; транспорт `udp` или `tcp`;
- **Kafka** — брокеры и топик; форматы `json`, `cef`, `leef`; по умолчанию используется подключение без шифрования; при необходимости доступны `security-protocol: SASL_SSL` и SASL-аутентификация.

Каждый приёмник имеет собственный фильтр (`filter`) с тремя критериями: `categories`, `severities`, `msg-ids`. Пустой фильтр пропускает все события. Для

непустого критерия событие должно входить в его список; критерии объединяются по И.

Пример:

```
{
  "logging": {
    "export": {
      "syslog": [
        {
          "name": "siem-events",
          "address": "192.0.2.60",
          "port": 514,
          "protocol": "udp",
          "format": "leef",
          "filter": { "severities": ["warning", "error", "critical"] }
        }
      ],
      "kafka": {
        "enabled": true,
        "brokers": ["kafka1.example.test:9092"],
        "topic": "amibig-events",
        "format": "cef",
        "filter": { "categories": ["audit", "update"], "msg-ids": [] },
        "security-protocol": "SASL_SSL",
        "sasl-mechanism": "PLAIN",
        "sasl-username": "amibig",
        "sasl-password": "<пароль>"
      }
    }
  }
}
```

Подробнее о каналах журналирования и локальном просмотре событий — в разделе Журналирование.

Хранение данных (observability.retention)

Локальные базы событий и истории предупреждений очищаются автоматически:

Поле	По умолчанию	Что ограничивает
events-days	90	возраст записей событийного журнала (events.db)
alert-history-days	90	возраст записей истории предупреждений
alert-history-max-rows	20000	максимальное число записей истории предупреждений

Диагностический архив

POST /api/v1/diagnostics/archive добавляет в архив машиночитаемые данные наблюдаемости: observability/manifest.json , observability/telemetry.json , observability/alerts-summary.json , observability/alerts-active.json , observability/

`history-summary.json` и, при включённом `include-certificates` — `observability/certificates.json`. Манифест содержит перечень разделов, версию схемы и краткую оценку состояния устройства. При обращении в техническую поддержку Amicon приложите диагностический архив к заявке.

Маршрутизация предупреждений

Жизненный цикл предупреждения не создаёт отдельного канала данных. Одно и то же событие:

- записывается локально в `events.db` с категорией `observability`;
- экспортируется через `syslog`-приёмники;
- преобразуется в SNMP trap-сообщения `alert-open` и `alert-clear`;
- попадает в диагностический архив как сводка для передачи в поддержку.

Связанные разделы

Настройка каналов передачи и форматов данных описана в следующих разделах:

- Журналирование и просмотр логов
- Каталог кодов сообщений
- Мониторинг по SNMP и trap-сообщения
- Телеметрия и метрики

Телеметрия и метрики

Амибиг предоставляет два внешних интерфейса для чтения текущего состояния устройства:

- `GET /api/v1/telemetry` и методы отдельных секций — единый снимок телеметрии в JSON;
- `GET /metrics` — текстовый формат Prometheus/OpenMetrics.

Оба интерфейса читают один и тот же общий снимок телеметрии: REST API и `/metrics` не собирают состояние независимо друг от друга.

REST API телеметрии

Полный снимок:

```
GET /api/v1/telemetry
```

Методы отдельных секций:

```
GET /api/v1/telemetry/device
GET /api/v1/telemetry/interfaces
GET /api/v1/telemetry/services
```

```
GET /api/v1/telemetry/virtual-servers
GET /api/v1/telemetry/ha
GET /api/v1/telemetry/tls
```

Полный снимок содержит:

- `schema_version`
- `collected_at`
- `device`
- `interfaces`
- `services`
- `virtual_servers`
- `ha`
- `tls`

Методы отдельных секций сохраняют `schema_version` и `collected_at` на верхнем уровне, а полезную нагрузку возвращают в поле `data`.

`/metrics`

Метод возвращает тот же снимок в формате, пригодном для сбора Prometheus:

```
GET /metrics
```

Формат ответа — Prometheus/OpenMetrics text exposition.

Основные семейства метрик:

- `amibig_info`
- `amibig_snapshot_collected_unixtime`
- `amibig_device_active_connections`
- `amibig_device_total_connections_total`
- `amibig_device_bytes_c2r_total`
- `amibig_device_bytes_r2c_total`
- `amibig_device_tls_connections`
- `amibig_interface_up`
- `amibig_interface_rx_bytes_total`
- `amibig_interface_tx_bytes_total`
- `amibig_interface_rx_errors_total`
- `amibig_interface_tx_errors_total`
- `amibig_service_member_up`
- `amibig_service_member_active_connections`
- `amibig_service_member_total_connections_total`
- `amibig_virtual_server_up`
- `amibig_virtual_server_active_connections`

- `amibig_virtual_server_total_connections_total`
- `amibig_ha_role`

Метки (labels)

Метки используют имена объектов конфигурации, а не внутренние идентификаторы среды выполнения. Примеры:

- `service`
- `member_address`
- `member_port`
- `virtual_server`
- `interface`
- `role`

Примеры

Опрос из Prometheus с проверкой сертификата интерфейса управления:

```
curl --cacert amibig-webui.pem -H "Authorization: Bearer <token>" \
  "https://<management-ip>/metrics"
```

Чтение полного снимка:

```
curl --cacert amibig-webui.pem -H "Authorization: Bearer <token>" \
  "https://<management-ip>/api/v1/telemetry"
```

Правила модели телеметрии

При интеграции учитывайте следующие свойства источника данных:

- запрос `/metrics` только читает уже собранное состояние и не запускает внешние команды;
- REST-телеметрия и `/metrics` читают один общий снимок;
- имена метрик и набор меток остаются стабильными между минорными обновлениями продукта.

Связь с SNMP и журналированием

Та же модель телеметрии обслуживает не только REST и `/metrics`:

- SNMP-поддерево `.1.3.6.1.4.1.61803` строится из того же снимка;
- SNMP trap-сообщения для сервисов, VIP и HA формируются при обнаружении переходов состояния над тем же снимком;
- экспорт событий в syslog отправляет те же структурированные события, которые хранятся локально.

Один снимок используется для REST, OpenMetrics, SNMP-опроса, trap-сообщений и syslog.

Связь с наблюдаемостью

Накопленная аналитика строится поверх той же модели:

- `GET /api/v1/observability/summary?window=5m` возвращает сводные показатели за выбранный интервал из накопленной истории;
- `GET /api/v1/alerts` оценивает правила предупреждений поверх накопленной истории и текущего состояния;
- экран **Наблюдаемость** веб-интерфейса использует методы наблюдаемости и предупреждений, не собирая отдельный набор показателей.

Связанные разделы

Настройка хранения, SNMP и экспорта событий описана в следующих разделах:

- Наблюдаемость: предупреждения, аналитика и экспорт
- Мониторинг по SNMP и trap-сообщения
- Журналирование и просмотр логов

Производительность

Раздел описывает целевые показатели производительности Амибиг, факторы, которые на неё влияют, и инструменты, позволяющие администратору наблюдать загрузку устройства и находить узкие места.

Целевые показатели

Целевые показатели продукта задают ориентир для приёмочных испытаний, но не заменяют измерения на оборудовании конкретного внедрения:

Показатель	Значение
Инспекция SSL/TLS (MITM)	не менее 20 Гбит/с
Базовый порог интенсивности соединений	1М CPM (соединений в минуту, $\approx 16,7$ тыс. соединений/с)

Фактические значения на конкретном внедрении зависят от характеристик оборудования, доли перехватываемого TLS-трафика, длины цепочек сервисов и включённых подсистем. Показатели конкретной инсталляции следует подтверждать нагрузочным тестированием на целевом оборудовании.

Модель обработки трафика

Dataplane Амибиг (`tforge`) построен по модели «одно соединение — один рабочий поток»: каждое соединение от начала до конца обрабатывается одним ядром без блокировок и без разделяемого изменяемого состояния на горячем пути.

Производительность масштабируется добавлением ядер dataplane; RSS сетевого адаптера распределяет соединения по рабочим потокам.

Из этой модели следуют практические правила:

- рост числа ядер dataplane увеличивает суммарную пропускную способность, но одно очень «тяжёлое» соединение по-прежнему ограничено производительностью одного ядра;
- равномерность загрузки рабочих потоков зависит от разнообразия трафика: малое число потоков с одинаковыми адресами ложится на малое число ядер.

Что влияет на производительность

При оценке запаса производительности учитывайте не только интенсивность соединений, но и включённые функции обработки:

Фактор	Влияние
Доля MITM-инспекции	Перехват TLS — самая дорогая операция: два TLS-соединения и криптография на каждый поток. Passthrough-трафик обрабатывается значительно дешевле
Длина цепочки сервисов	Каждый inline-сервис в цепочке добавляет два прохода трафика через устройство
Профиль наблюдаемости	<code>full_diagnostics</code> увеличивает детализацию и снижает запас производительности; <code>minimal_telemetry</code> — наоборот (см. Наблюдаемость)
Передача журнала соединений	В режиме <code>all</code> под высокой нагрузкой добавляет накладные расходы; порог <code>adaptive-threshold-cps</code> автоматически включает выборку событий
HA	Синхронизация конфигурации и ключей возобновления TLS-сессий добавляет служебный трафик и нагрузку на процессор. Состояние установленных TCP-соединений не синхронизируется
MSS/MTU	Неправильный MSS вызывает фрагментацию или повторные передачи (см. MSS и MTU)

Наблюдение за загрузкой

Оценивайте загрузку по системной статистике и показателям обработки трафика. Сопоставление этих данных помогает отличить нехватку вычислительных ресурсов от ограничений сервисной цепочки или сетевых интерфейсов.

Загрузка процессоров и рабочих потоков dataplane

Для оценки полезной нагрузки dataplane запросите системную статистику:

```
GET /api/v1/system/stats
```

Ответ содержит блок `cpu` с общей загрузкой и загрузкой по ядрам, блок `memory` и статистику интерфейсов. Когда dataplane запущен, в `cpu` добавляется блок `dataplane` — фактическая занятость рабочих потоков по учёту циклов `tforge`:

```
{
  "cpu": {
    "avg_pct": 12.4,
    "dataplane": {
      "workers": [ { "worker_id": 0, "busy_pct": 37.5 } ],
      "avg_pct": 37.5,
      "history_avg": [35.1, 36.8, 37.5]
    }
  }
}
```

Показания `/proc` для ядер dataplane не отражают полезную нагрузку: режим активного опроса (`busy-poll`) держит их на 100%. Используйте `cpu.dataplane : busy_pct` — доля циклов, потраченных на обработку пакетов.

Эти же данные видны на экране `Обзор` веб-интерфейса.

Потери пакетов на сетевой карте

Счётчики `imissed`, `ierrors` и `rx_nombuf` накапливаются в истории наблюдаемости и доступны через `GET /api/v1/observability/series (metric=imissed_rate|ierrors_rate|rx_nombuf_rate, scope=interface)`. Рост `imissed` под нагрузкой — признак того, что рабочие потоки не успевают считывать очереди сетевого адаптера. Это первый сигнал приближения к пределу производительности.

Предупреждения

Типы правил `cpu`, `memory` и `port-exhaustion` позволяют получать предупреждения о приближении к пределам до того, как начнутся потери (см. Наблюдаемость).

Параметры настройки

Движковые параметры DPDK задаются на экране `Система` (форма DPDK) или в секции `dataplane.dpdk`:

Параметр	Назначение
<code>lcore-mask</code>	маска ядер процессора, выделенных dataplane; определяет число рабочих потоков
<code>hugepages-mb</code>	объём hugepages для пулов пакетных буферов
<code>config-path</code>	путь к сгенерированному файлу конфигурации dataplane

Параметры движка применяются только с перезапуском dataplane (`cold apply`).
Веб-интерфейс предупреждает об этом на странице [Применить конфигурацию](#).

Сетевые параметры, влияющие на производительность (ограничение MSS, `proxy-mss`, обнаружение недоступности PMTUD), задаются на экране [Система](#) в секции «Протокол (TCP / MSS)» — см. MSS и MTU. Режим трансляции адреса источника и правила выбора порта источника для исходящих соединений задаются в секции «Egress (трансляция адреса источника)» — см. Трансляция адреса источника (SNAT).

Методика измерений

При самостоятельном нагрузочном тестировании учитывайте:

- **Генератор трафика может сам стать узким местом.** Прежде чем делать выводы о пределе устройства, убедитесь, что генератор не исчерпал собственные ресурсы (CPU, порты, файловые дескрипторы).
- **Считайте соединения на приёмной стороне.** Клиентские инструменты занижают счёт при ошибках; надёжнее считать установленные соединения на origin-сервере.
- **Фиксируйте контекст.** Результат без указания версии сборки, конфигурации, числа рабочих потоков, состояния НА и профиля наблюдаемости не воспроизводим и не пригоден для сравнения.
- **Разделяйте классы измерений.** Пропускная способность больших передач (Гбит/с) и интенсивность коротких соединений (CPS) — разные пределы; смешивать их в одну цифру нельзя.

Связанные разделы

Для настройки сбора показателей и сетевых параметров используйте следующие разделы:

- Наблюдаемость: предупреждения, аналитика и экспорт
- Телеметрия и метрики
- MSS и MTU
- Трансляция адреса источника (SNAT)

Резервное копирование и восстановление

Амибиг создаёт резервную копию действующей конфигурации и связанных с ней файлов в формате ACF. На том же экране можно восстановить ACF или импортировать архив UCS с F5 BIG-IP.

Доступные операции

Перед изменением конфигурации выберите операцию с учётом формата исходного файла и ожидаемого результата:

Операция	Описание
Экспорт ACF	Создание архива текущей конфигурации
Импорт ACF	Восстановление конфигурации из архива
Импорт F5 UCS	Конвертация и импорт конфигурации из F5 BIG-IP UCS-архива

Формат ACF

ACF (файл конфигурации Амибиг) — архив постоянных данных из раздела `/config`. Неприменённые изменения из подготовленной конфигурации в архив не попадают. В резервную копию входят:

Содержимое	Описание
Действующая конфигурация	Сохранённое состояние <code>/config/amibig.json</code>
SSL/TLS	Сертификаты и закрытые ключи SSL/TLS
HA	УЦ, сертификаты и закрытые ключи HA
Лицензия	Файл лицензии
Ключи	Ключи подписи обновлений и KEK

Имя файла формируется автоматически: `amibig-backup-YYYYMMDD-HHMMSS.acf`.

Экспорт конфигурации

Для экспорта требуется роль `admin`. Созданный ACF содержит закрытые ключи и учётные данные, поэтому храните его в защищённом хранилище.

Через веб-интерфейс

Экспорт через веб-интерфейс создаёт файл без изменения конфигурации:

1. Перейдите на экран **Резервные копии**.
2. Нажмите **Экспорт** в блоке **Экспорт резервной копии**.
3. Убедитесь, что браузер сохранил файл **.acf**.

Через REST API

Для автоматического экспорта выполните запрос с токеном пользователя **admin**:

```
curl -o backup.acf \
  "https://<management-ip>/api/v1/backup/export" \
  -H "Authorization: Bearer $TOKEN"
```

API возвращает файл **application/octet-stream**; имя файла указано в заголовке **Content-Disposition**.

Импорт конфигурации

Восстановление ACF перезаписывает постоянную конфигурацию, сертификаты, лицензию, ключи и учётные записи, а затем применяет восстановленную конфигурацию без таймера подтверждения. До импорта экспортируйте текущее состояние и убедитесь, что после операции сможете войти с учётными данными из резервной копии.

Через веб-интерфейс

Для восстановления файла ACF:

1. Перейдите на экран **Резервные копии**.
2. В блоке **Восстановление из резервной копии** выберите файл **.acf** и нажмите **Восстановить**.
3. Дождитесь завершения импорта.
4. После завершения повторно выполните вход и проверьте состояние сервисов.

Через REST API

Для восстановления через API передайте файл как **multipart/form-data**:

```
curl -X POST "https://<management-ip>/api/v1/backup/import" \
  -H "Authorization: Bearer $TOKEN" \
  -F "file=@backup.acf"
```

Ответ при успехе:

```
{
  "success": true,
```

```
"message": "Backup restored successfully"
}
```

Процесс восстановления

Во время восстановления Амибиг выполняет следующие операции:

1. Загруженный архив `.acf` распаковывается.
2. Проверяется формат архива и безопасность содержащихся в нём путей.
3. Конфигурация перезагружается в память.
4. Автоматически применяется конфигурация для активации.

Импорт F5 BIG-IP UCS

Импорт UCS выполняется отдельно от восстановления резервной копии Амибиг. Конвертер переносит только поддерживаемые объекты и формирует отчёт, который необходимо проверить до применения подготовленной конфигурации.

Назначение

Импорт UCS переносит поддерживаемые объекты F5 BIG-IP SSL Orchestrator в подготовленную конфигурацию Амибиг. UCS (User Configuration Set) — штатный формат резервной копии F5 BIG-IP. Импорт не применяет результат автоматически.

Что конвертируется

Конвертер сопоставляет поддерживаемые объекты следующим образом:

Компонент F5	Компонент Амибиг	Примечания
Клиентские SSL-профили	Клиентские профили SSL/TLS	Наборы шифров и версии протокола
Серверные SSL-профили	Серверные профили SSL/TLS	Наборы шифров и версии протокола
SSL-сертификаты	Сертификаты SSL/TLS	Сертификаты и ключи
Виртуальные серверы	Топология	Адреса и порты
Участники пулов	Пулы сервисов	Адреса участников пула
Политика SSLO	Правила политики	Условия и действия
Цепочки сервисов	Цепочки сервисов	Порядок сервисов
Self-IP	Self-IP Амибиг	IP-адреса интерфейсов
VLAN	VLAN Амибиг	Теги и интерфейсы
Маршруты	Сетевые маршруты	Статические маршруты

Компонент F5	Компонент Амибиг	Примечания
iRules	aRules	Частичная конвертация

Учётные записи и списки доступа управления F5 из UCS не переносятся. Конвертер создаёт штатную учётную запись `admin` и разрешающий список доступа. Если в действующей конфигурации уже есть учётная запись `admin` с хешем пароля, при импорте этот хеш сохраняется. Остальные учётные записи и списки доступа настройте вручную. Подробный разбор конвертации приведён на странице Миграция с F5 (импорт UCS).

Отчёт конвертации

После импорта система формирует отчёт. Уровень сообщения показывает, требуется ли ручная проверка объекта:

Уровень	Описание
info	Объект успешно конвертирован
warning	Объект конвертирован с ограничениями
skipped	Объект не конвертирован, требуется ручная настройка

Через веб-интерфейс

Для импорта UCS и проверки результата:

1. Перейдите на экран `Резервные копии`.
2. В блоке «Импорт из F5 BIG-IP» выберите файл `.ucs` и нажмите `Конвертировать и импортировать`.
3. Дождитесь завершения конвертации.
4. Просмотрите отчёт конвертации.
5. Проверьте конфигурацию в соответствующих разделах.
6. Примените конфигурацию.

Через REST API

Для автоматического импорта отправьте UCS следующим запросом:

```
curl -X POST "https://<management-ip>/api/v1/config/import-ucs" \
  -H "Authorization: Bearer $TOKEN" \
  -F "file=@bigip-config.ucs"
```

Сокращённый ответ:

```
{
  "config": {
    "system": {
      "hostname": "bigip1"
    }
  },
  "report": [
    {
      "severity": "info",
      "entity": "ssl/client-profile/clientssl",
      "message": "SSL client profile converted"
    },
    {
      "severity": "warning",
      "entity": "services/inline/ngfw",
      "message": "Inline service interfaces require manual mapping"
    }
  ]
}
```

Особенности импорта UCS

До начала миграции учтите ограничения конвертера:

- Объекты импортируются в **подготовленную (candidate) конфигурацию**. Действующая (running) конфигурация не изменяется до применения.
- SSL-сертификаты и ключи извлекаются из UCS и сохраняются в системе.
- Учётные записи и пароли пользователей F5 не переносятся. Конвертер создаёт только штатную учётную запись `admin` и сохраняет хеш пароля одноимённой существующей учётной записи, если он задан в действующей конфигурации.
- Параметры F5, для которых в Амибиг нет аналога, пропускаются с сообщением в отчёте.

Пошаговая процедура миграции с F5

Процедура сохраняет исходное состояние Амибиг и оставляет результат импорта в подготовленной конфигурации до отдельного применения:

1. **На F5 BIG-IP:** создайте UCS-архив:

```
tmsh save sys ucs /var/local/ucs/migration.ucs
```

2. **Скопируйте** UCS-файл на рабочую станцию.

3. **На Амибиг:** создайте резервную копию текущей конфигурации (если есть):

```
curl -o pre-migration-backup.acf \
  "https://<management-ip>/api/v1/backup/export" \
  -H "Authorization: Bearer $TOKEN"
```

4. **Импортируйте** UCS:

```
curl -X POST "https://<management-ip>/api/v1/config/import-ucs" \
-H "Authorization: Bearer $TOKEN" \
-F "file=@migration.ucs"
```

5. **Просмотрите отчёт** конвертации. Обратите внимание на записи с уровнем `warning` и `skipped`.

6. **Проверьте конфигурацию** в веб-интерфейсе:

- сетевые интерфейсы, VLAN, Self-IP и маршруты;
- профили и сертификаты SSL/TLS;
- inline- и ICAP-сервисы: привязку интерфейсов может потребоваться настроить вручную;
- Цепочки сервисов и политики.

7. **Примените** конфигурацию через веб-интерфейс или API.

8. **Проверьте** прохождение трафика и работу сервисов.

Рекомендации

План резервного копирования должен учитывать обновления, перенос между устройствами и защиту архива.

Регулярное резервное копирование

Создавайте и проверяйте резервные копии регулярно:

- Создавайте резервную копию перед каждым значимым изменением конфигурации.
- Создавайте резервную копию перед обновлением системы.
- Храните резервные копии на внешнем хранилище, отдельно от устройства.

Перенос конфигурации между устройствами

Экспорт/импорт ACF может использоваться для переноса конфигурации между устройствами Амибиг. Учитывайте:

- Лицензия привязана к отпечатку оборудования конкретного устройства и не переносится.
- Сетевые интерфейсы могут иметь разные имена на разных устройствах.
- При включенном HA секция `ha` содержит адреса, специфичные для каждого узла.

Обновление системы

Амибиг обновляется по схеме A/B. Устройство содержит два слота корневой файловой системы; один из них активен, а пакет записывается во второй. После перезагрузки система проверяет новый слот. Если проверка доступности не пройдена, устройство автоматически возвращается к предыдущему слоту.

См. также Артефакты релиза: пакет обновления — только часть полного комплекта поставки для изолированного контура. Описание артефактов поставки описывается файлами `amibig-release-manifest.json` и `SHA256SUMS`.

Экран «Обслуживание»

Точка входа для операций обновления и перезагрузки — экран `Обслуживание` веб-интерфейса.

Экран объединяет:

- текущую версию и номер сборки;
- активный слот и следующий слот загрузки;
- признак «нужна перезагрузка»;
- загруженный пакет обновления и действие `Активировать` (подготовка к следующей загрузке);
- цель отката и действие `Откатить`;
- действия `Перезагрузить`, `Запланировать перезагрузку` и отмену запланированной перезагрузки;
- статус текущей или последней операции обслуживания;
- явное состояние перепоключения после принятой перезагрузки.

После принятия команды перезагрузки веб-интерфейс и API становятся временно недоступны. Для контролируемой перезагрузки:

1. Подготовьте целевой слот.
2. Убедитесь, что для следующей загрузки выбран нужный слот.
3. Запустите перезагрузку.
4. Дождитесь восстановления доступа к интерфейсу управления.
5. Проверьте активный слот и версию ПО.

Процедура обновления

Обновление выполняется поэтапно: пакет сначала проверяется, затем записывается в неактивный слот. Перезагружайте устройство только после успешной записи и готовности целевого слота.

Пошаговый процесс

Обновление состоит из четырёх отдельных этапов:

1. Загрузка пакета → Проверка → Состояние ready
2. Активация → Запись в неактивный слот → Состояние applied
3. Перезагрузка → Загрузка с нового слота
4. Проверка доступности → Подтверждение или откат

Шаг 1. Загрузка пакета

Загрузите пакет `.amibig` через веб-интерфейс или API:

Через веб-интерфейс:

1. Перейдите на экран `Обслуживание`.
2. Нажмите `Загрузить` в блоке «ПО и слоты».
3. Выберите файл `.amibig`.
4. Дождитесь завершения проверки пакета.

Через API:

```
curl -X POST "https://<management-ip>/api/v1/system/update/upload" \
  -H "Authorization: Bearer $TOKEN" \
  -F "file=@amibig-update-v0.3.0.amibig"
```

Ответ:

```
{
  "status": "ok",
  "version": "0.3.0",
  "build_date": "2026-02-22T06:00:00Z",
  "target_slot": "b"
}
```

Проверка пакета

При загрузке система автоматически проверяет пакет обновления:

- сверяет контрольные суммы;
- проверяет криптографическую подпись;
- проверяет, не отозван ли ключ подписи.

Пакеты, не прошедшие проверку, отклоняются с описанием ошибки. Ошибка подписи (недоверенный или отозванный ключ, повреждённый пакет) фиксируется в событийном журнале с кодом `AMB-1012` — см. [События обновления](#).

Шаг 2. Применение обновления

Через веб-интерфейс:

1. После успешной проверки нажмите **Активировать**.
2. Дождитесь завершения записи в неактивный слот.

Через API:

```
curl -X POST "https://<management-ip>/api/v1/system/update/apply" \
  -H "Authorization: Bearer $TOKEN"
```

Ответ:

```
{
  "status": "ok",
  "version": "0.3.0",
  "target_slot": "b",
  "message": "Reboot to activate the new version"
}
```

Результат активации

Кнопка **Активировать** и запрос `/system/update/apply` не перезагружают устройство. Они выполняют следующие действия:

1. Определяется неактивный слот: если активен А, целевым становится В, и наоборот.
2. Целевой слот подготавливается и в него записывается обновлённая корневая файловая система.
3. Обновляется конфигурация загрузчика для загрузки с нового слота.
4. Обрабатываются директивы ротации ключей из манифеста (если есть).

Шаг 3. Перезагрузка

Через веб-интерфейс:

Используйте **Перезагрузить**, либо заранее запланируйте перезагрузку (**Запланировать**) и при необходимости отмените её. Рядом доступна кнопка **Выключить** для штатного выключения питания. После выключения устройство не запустится автоматически: потребуется IPMI, управляемый блок распределения питания или физическая кнопка включения.

Через API:

```
### Перезагрузка
curl -X POST "https://<management-ip>/api/v1/system/reboot" \
  -H "Authorization: Bearer $TOKEN"

### Выключение (графический эквивалент кнопки «Выключить»)
```



```
curl -X POST "https://<management-ip>/api/v1/system/shutdown" \
-H "Authorization: Bearer $TOKEN"
```

API подтверждает принятие операции структурированным ответом:

```
{
  "status": "accepted",
  "operation_id": "maint-abc123",
  "message": "Reboot accepted. Disconnect is expected.",
  "disconnect_expected": true,
  "eta_seconds": 10
}
```

После ответа со значением `disconnect_expected: true` дождитесь восстановления доступа к интерфейсу управления. Операции перезагрузки и выключения взаимоисключающие: пока одна уже принята, запрос второй возвращает `409 Conflict`. В HA-паре активный узел перед выполнением операции передаёт роль партнёру.

Шаг 4. Проверка доступности и подтверждение

После загрузки с нового слота Амибиг автоматически проверяет систему:

Проверка	Описание	Таймаут
Сетевой интерфейс	Хотя бы один физический сетевой интерфейс находится в состоянии UP	До 120 секунд
API	Порт управления доступен	До 120 секунд
Инспектор трафика	Сервис инспекции трафика запущен	До 120 секунд

Если все проверки пройдены:

- Обновление считается подтвержденным.
- Устройство продолжает работать на новом слоте.

Если проверки не пройдены:

- Выполняется автоматический откат: загрузчик перенастраивается на предыдущий слот, устройство перезагружается.

Статус обновления

Проверяйте статус после загрузки пакета, запуска установки и перезагрузки. Состояние операции доступно через REST API и веб-интерфейс.

Через API

Для контроля текущей операции запросите её состояние:

```
curl "https://<management-ip>/api/v1/system/update/status" \
-H "Authorization: Bearer $TOKEN"
```

Возможные состояния:

Состояние	Описание
idle	Нет активных операций
uploading	Загрузка пакета
validating	Валидация пакета
applying	Запись в целевой слот
rebooting	Перезагрузка
shutting_down	Выключение (poweroff)
error	Ошибка (описание в поле message)

Информация о слотах

Отдельный запрос возвращает активный слот, цель следующей загрузки и версии:

```
curl "https://<management-ip>/api/v1/system/slots" \
-H "Authorization: Bearer $TOKEN"
```

```
{
  "active": "a",
  "active_slot": "a",
  "next_boot_slot": "b",
  "next_boot_version": "0.3.0",
  "rollback_slot": "a",
  "rollback_version": "0.2.0",
  "reboot_required": true,
  "pending_update_version": "0.3.0",
  "slots": [
    {
      "slot": "a",
      "active": true,
      "version": "0.2.0",
      "build_date": "2026-02-15T10:00:00Z"
    },
    {
      "slot": "b",
      "active": false,
      "version": "0.3.0",
      "build_date": "2026-02-22T06:00:00Z"
    }
  ],
  "boot_try": null
}
```

Ручной откат

Ручной откат выбирает другой установленный слот для следующей загрузки. Перед операцией убедитесь, что на нём находится нужная версия.

Через веб-интерфейс:

1. Перейдите на экран **Обслуживание**.
2. Проверьте, какой слот отката сейчас доступен.
3. Нажмите **Откатить** и подтвердите переключение цели загрузки.
4. Перезагрузите устройство.

Через API:

```
### 1. Переключение на другой слот
curl -X POST "https://<management-ip>/api/v1/system/update/rollback" \
  -H "Authorization: Bearer $TOKEN"

### 2. Перезагрузка
curl -X POST "https://<management-ip>/api/v1/system/reboot" \
  -H "Authorization: Bearer $TOKEN"
```

Откат возможен только если альтернативный слот содержит установленную версию.

События обновления

Жизненный цикл обновления фиксируется в событийном журнале (категория **update**, коды **AMB-NNNN**):

Код	Уровень	Событие
AMB-1011	info	Пакет проверен и готов к установке (версия, целевой слот)
AMB-1012	error	Проверка пакета завершилась ошибкой — в том числе ошибка подписи: недоверенный ключ, отозванный ключ подписи или КЕК, несовпадение подписи манифеста или HMAC полезной нагрузки
AMB-1010	warning	Принят неподписанный пакет формата v1 (переходный период)
AMB-1013	info	Обновление установлено в неактивный слот

Код	Уровень	Событие
AMB-1014	error	Установка обновления завершилась ошибкой
AMB-1015	warning	Подготовлен откат на предыдущий слот
AMB-1016	info	Запрошена перезагрузка системы

События доступны на экране `Просмотр логов`, экспортируются в syslog и Kafka (см. `Наблюдаемость`) и описаны в каталоге кодов сообщений. Событие `AMB-1012` с текстом об отозванном или недоверенном ключе — признак либо устаревшего доверенного хранилища на устройстве, либо подменённого пакета: не устанавливайте такой пакет и сверьте его контрольную сумму с `SHA256SUMS` из комплекта поставки.

Защита обновлений

На устройствах с полнодисковым шифрованием корневая файловая система хранится в зашифрованном виде, а её целостность проверяется криптографически. Порядок обновления для администратора не меняется.

Полный цикл обновления через API

Следующий пример выполняет загрузку, активацию и перезагрузку последовательно. После каждого запроса проверьте HTTP-код и поле `status` :

```
### Аутентификация
TOKEN=$(curl -s -X POST "https://<management-ip>/api/v1/auth/login" \
  -H "Content-Type: application/json" \
  -d '{"username":"admin","password":"password"}' | jq -r '.token')

### 1. Загрузка
curl -X POST "https://<management-ip>/api/v1/system/update/upload" \
  -H "Authorization: Bearer $TOKEN" \
  -F "file=@amibig-update-v0.3.0.amibig"

### 2. Проверка статуса
curl "https://<management-ip>/api/v1/system/update/status" \
  -H "Authorization: Bearer $TOKEN"

### 3. Применение
curl -X POST "https://<management-ip>/api/v1/system/update/apply" \
  -H "Authorization: Bearer $TOKEN"

### 4. Проверка слотов
curl "https://<management-ip>/api/v1/system/slots" \
  -H "Authorization: Bearer $TOKEN"

### 5. Перезагрузка
curl -X POST "https://<management-ip>/api/v1/system/reboot" \
  -H "Authorization: Bearer $TOKEN"

### После перезагрузки: проверка версии
```

```
curl "https://<management-ip>/api/v1/system/slots" \
-H "Authorization: Bearer $TOKEN"
```

Ротация ключей через обновление

Пакет обновления может содержать следующие директивы замены криптографических ключей:

Директива	Описание
<code>add_signing_keys</code>	Добавить новые публичные ключи подписи
<code>revoke_keys</code>	Отозвать существующие ключи (по ID)

При активации обновления новые ключи добавляются в доверенное хранилище, а отозванные ключи блокируются. Последующие пакеты, подписанные отозванными ключами, отклоняются.

Сброс к заводским настройкам

Сброс возвращает конфигурацию Амибиг к начальному состоянию, сохранив доступ к управлению: узел после перезагрузки остаётся доступен по прежнему адресу, но встречает администратора первичной настройкой, как после установки.

Что стирается и что сохраняется

Стирается:

- вся рабочая конфигурация: сервисы, пулы, виртуальные серверы, цепочки сервисов, политики, GSLB, aRules, настройки HA;
- все учётные записи пользователей;
- список NTP-серверов;
- история конфигурации (`/config/history`);
- сертификаты HA-сопряжения (`/config/ha`), включая загруженный CA партнёра;
- привязка портов к DPDK: dataplane останавливается, NIC возвращаются ядру, чтобы следующая загрузка была детерминированно чистой.

Сохраняется:

- настройки управления: имя хоста, часовой пояс, способ получения адреса, адрес и шлюз управления, DNS-серверы, интерфейсы управления (`plane: management`);
- лицензия и идентификаторы устройства — сброс не затрагивает файлы, не относящиеся к конфигурации;
- журналы, SSH host-ключи и persist-таблицы;

- файлы сертификатов и ключей на разделе `/config` (кроме HA-сертификатов) физически не удаляются, но ссылающаяся на них секция `ssl` из конфигурации исчезает.

Через веб-интерфейс

Сброс требует роли `admin` и явного текстового подтверждения:

1. Откройте экран `Обслуживание`, панель действий питания.
2. Нажмите `Сбросить` в строке «Сброс к заводским настройкам».
3. В окне подтверждения введите фразу `FACTORY RESET` (заглавными) и нажмите `Подтвердить сброс`.

Если узел — активный в HA-паре, интерфейс дополнительно предупредит: сброс не каскадируется на партнёра. Партнёр обнаружит потерю heartbeat-сообщений и пометит узел недоступным, но трафик в момент сброса может прерваться.

Через API

Тот же сброс можно выполнить запросом с подтверждающей фразой:

```
curl -X POST "https://<management-ip>/api/v1/system/maintenance/factory-reset" \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"confirmation": "FACTORY RESET", "schedule_reboot": true}'
```

Поле `confirmation` обязано содержать строго `FACTORY RESET`, иначе запрос отклоняется с кодом `400`. Поле `schedule_reboot` (по умолчанию `true`) определяет, перезагружать ли узел сразу после сброса.

Ответ:

```
{
  "ok": true,
  "warnings": [],
  "ha_role": "standalone",
  "scheduled_reboot": true,
  "audit_event_id": "3f2a9c..."
}
```

Каждый этап сброса (начало операции, удаление конфигурации и планирование перезагрузки) фиксируется в журнале как событие аудита с единым `audit_event_id`, именем инициатора и адресом источника.

Через консоль устройства доступна команда `/usr/local/bin/amibig-factory-reset.py` — с тем же двойным подтверждением (фраза `FACTORY RESET` плюс явное согласие).

Результат сброса

Узел загружается с той же версией ПО (слоты A/B сброс не затрагивает) и с прежним адресом управления. Поскольку пользователи удалены, система заново

создаёт учётную запись `admin` со случайным начальным паролем. Пароль выводится в сообщении консоли и сохраняется в `/config/.initial-admin-password`; при первом входе пароль потребуется сменить. Дальше настройка выполняется как при первом запуске — см. Установку и Быстрый старт.

История конфигурации

Амибиг автоматически сохраняет снимок конфигурации при каждом успешном применении и при каждом импорте. Снимки образуют историю изменений: в любой момент можно посмотреть, что именно поменялось между версиями, и вернуть прежнюю версию конфигурации, не восстанавливая её из резервной копии.

Хранится до 50 последних снимков; при превышении лимита самые старые записи удаляются автоматически. Снимки лежат на устройстве в каталоге `/config/history` и не входят в пакет обновления — при обновлении системы история сохраняется, а при сбросе к заводским настройкам удаляется вместе с конфигурацией.

История дополняет, но не заменяет резервные копии: снимок содержит только конфигурацию (`amibig.json`), без сертификатов, лицензии и ключей. Полный перенос состояния устройства выполняется через резервное копирование.

Экран «История»

Экран `История` находится в группе «Управление» бокового меню. Он показывает таблицу снимков (новые сверху) со столбцами:

Столбец	Содержимое
Дата	момент создания снимка
Действие	источник снимка: <code>apply</code> — действующая конфигурация после успешного применения, <code>import</code> — подготовленная конфигурация после импорта файла или восстановления снимка
Описание	комментарий, указанный администратором при применении

Для каждой записи доступны действия `Сравнить с текущей`, `Восстановить` и `Удалить`. Просматривать историю и отличия могут все роли (`admin`, `operator`, `viewer`); восстановление и удаление — только `admin`.

Сравнение версий

Сравнить версии можно двумя способами:

- **С текущей конфигурацией** — кнопка `Сравнить с текущей` в строке снимка. Слева показывается снимок, справа — действующая (`running`) конфигурация.
- **Две произвольные версии** — блок «Сравнение версий» над таблицей: выберите снимок «от» и снимок «до» (или пункт «Текущая конфигурация») и нажмите `Сравнить` .

Окно сравнения показывает конфигурацию в формате JSON бок о бок: добавленные строки подсвечены зелёным, удалённые — красным, изменённые — жёлтым. Совпадающие фрагменты сворачиваются в строку «N скрытых строк» — щелчок по ней раскрывает контекст. Если версии идентичны, выводится сообщение об отсутствии изменений.

Восстановление прежней версии

Кнопка `Восстановить` в строке снимка (после подтверждения) загружает выбранный снимок как **подготовленную (`candidate`) конфигурацию**. Восстановление само по себе не меняет работу устройства: прежняя действующая конфигурация используется до тех пор, пока вы не выполните `Применить конфигурацию` .

Для восстановления:

1. Восстановите нужный снимок.
2. Просмотрите различия между подготовленной и действующей конфигурациями на странице применения.
3. Примените конфигурацию и подтвердите её.

Применение восстановленной версии проходит через стандартный механизм применения и подтверждения с таймером отката: если после применения не нажать `Подтвердить` , через 120 секунд выполняется автоматический откат. Это защищает от восстановления версии, в которой, например, изменён адрес управления. Подробнее о состояниях конфигурации и таймере отката — в разделе Веб-интерфейс.

Удаление записей

Кнопка `Удалить` в строке снимка удаляет запись истории вместе с файлом снимка. Операция необратима и доступна только роли `admin` . Удаление записи не влияет на действующую конфигурацию.

REST API

Все операции экрана доступны через REST API:

Метод и путь	Назначение	Роль
GET /api/v1/history	список записей (limit , offset)	любая
GET /api/v1/history/{id}	снимок целиком	любая
GET /api/v1/history/{id}/diff	пара «снимок — текущая конфигурация» для сравнения	любая
POST /api/v1/history/diff	пара произвольных снимков (id_from , id_to)	любая
POST /api/v1/history/{id}/restore	восстановить снимок в подготовленную конфигурацию	admin
DELETE /api/v1/history/{id}	удалить запись	admin

Пример восстановления снимка:

```
curl -X POST "https://<management-ip>/api/v1/history/20260817-142530-a1b2c3/restore" \
-H "Authorization: Bearer $TOKEN"
```

Ответ:

```
{
  "success": true,
  "message": "Snapshot restored as candidate"
}
```

Восстановление и удаление фиксируются в событийном журнале (категория config), поэтому по журналам всегда можно установить, кто и когда возвращал прежнюю версию.

Системные параметры

Базовые параметры узла — имя хоста, часовой пояс, DNS- и NTP-серверы, адрес интерфейса управления — собраны на экране Система и хранятся в секции system конфигурации. Изменения сохраняются в подготовленной конфигурации, показываются в списке различий и вступают в силу после Применить конфигурацию (см. порядок применения конфигурации).

Экран Система содержит и другие секции: параметры DPDK, «Egress (трансляция адреса источника)», «Протокол (TCP / MSS)» и привязку процессоров. Они описаны в своих разделах: Трансляция адреса источника (SNAT) и MSS и MTU; здесь рассматриваются только общесистемные параметры.

Секция «Основные»

Основные параметры определяют имя узла в сети и системное время:

Поле в UI	Ключ конфигурации	Описание
Имя хоста	<code>system.hostname</code>	имя узла
Часовой пояс	<code>system.timezone</code>	часовой пояс узла, по умолчанию UTC

Секция «Управление»

Здесь настраивается адрес интерфейса управления:

Поле в UI	Ключ конфигурации	Описание
Способ получения адреса	<code>system.management.method</code>	dhcp (по умолчанию) или static
Адрес	<code>system.management.address</code>	статический адрес с длиной префикса, например 10.1.0.10/24
Шлюз	<code>system.management.gateway</code>	шлюз по умолчанию для интерфейса управления

В режиме DHCP поля адреса и шлюза не сохраняются в конфигурации — их выдаёт DHCP-сервер. Сам порт управления (какой физический интерфейс используется для управления) выбирается на экране Сеть: у интерфейса задаётся `plane: management`. Первичное значение записывает инсталлер.

Механизм применения защищает от потери доступа: конфигурацию, в которой не остаётся ни одного включённого интерфейса управления, применить нельзя, а после применения действует таймер отката.

DNS и NTP

Адреса серверов задаются списками в соответствующих полях:

Поле в UI	Ключ конфигурации	Описание
DNS-серверы	<code>system.dns.nameservers</code>	список адресов DNS-серверов
Серверы (секция NTP)	<code>system.ntp.servers</code>	список NTP-серверов (адрес или имя)

Синхронизацию времени выполняет служба `amibig-ntpd` (клиент BusyBox `ntpd`). При применении конфигурации служба перенастраивается на указанный список серверов; если список пуст, служба удаляется и синхронизация отключается.

Точное время особенно важно для проверки сроков действия сертификатов при TLS-инспекции и для сопоставления событий журналов между узлами HA-пары — задайте NTP-серверы на обоих узлах.

ACL управления

Доступ к веб-интерфейсу и REST API по адресам источника ограничивает ACL управления. Он настраивается **не** на экране Система, а на экране Пользователи (блок «ACL управления»), потому что хранится в секции `users` конфигурации:

```
{
  "users": {
    "management-acl": {
      "entry": [
        {"source": "10.0.0.0/24", "action": "allow"},
        {"source": "0.0.0.0/0", "action": "deny"}
      ]
    }
  }
}
```

Правила проверяются сверху вниз до первого совпадения; у записи можно дополнительно указать список портов. Пустой ACL означает «разрешено всем». При применении система проверяет, что новый ACL не отрежет доступ тому адресу, с которого выполняется применение. Подробности — в разделе Управление пользователями.

Что требует перезапуска

Общесистемные параметры этой страницы — `system.hostname`, `system.timezone`, `system.management`, `system.dns`, `system.ntp` — относятся к «тёплому» классу изменений: они применяются транзакцией control plane, без перезапуска dataplane и без влияния на проходящий трафик.

Другие секции экрана Система требуют применения с перезапуском обработчика трафика (cold apply): переключатель `system.cpu-affinity-enabled` (привязка CPU), параметры DPDK, а также секции «Egress» и «Протокол». Страница применения предупреждает о таких изменениях заранее — см. изменения с перезапуском dataplane.

Поведение при сбросе к заводским настройкам

Имя хоста, часовой пояс, настройки управления, DNS-серверы и интерфейсы управления сохраняются при сбросе к заводским настройкам — узел остаётся доступным по прежнему адресу. Список NTP-серверов при сбросе не сохраняется.

Миграция с F5 (импорт UCS)

Амибиг импортирует конфигурацию F5 BIG-IP с SSL Orchestrator из стандартного архива резервной копии UCS (User Configuration Set). Конвертер разбирает TMSH-конфигурацию (`bigip.conf` , `bigip_base.conf`), переводит объекты в модель Амибиг, извлекает из архива SSL-сертификаты и ключи и формирует отчёт конвертации. Результат попадает в **подготовленную (candidate) конфигурацию**. До применения работа устройства не меняется, и у администратора есть время всё проверить.

Настройки доступа к управлению из UCS не заменяют настройки самого устройства. Амибиг сохраняет свой интерфейс управления, текущий адрес и шлюз. Адрес управления F5 конвертер записывает только в отчёт; если он отличается от текущего адреса Амибиг, отчёт содержит сообщение `warning` .

Если UCS повреждён в конце, конвертер обрабатывает читаемую часть и отмечает неполные объекты в отчёте. Не применяйте результат, пока не проверите все сообщения `warning` и `skipped` .

Шаг 1. Выгрузка UCS на F5 BIG-IP

На стороне F5 создайте архив и скопируйте его на рабочую станцию:

```
tmsl save sys ucs /var/local/ucs/migration.ucs
```

Перед импортом рекомендуется сделать резервную копию текущей конфигурации Амибиг — импорт полностью заменит подготовленную конфигурацию.

Шаг 2. Загрузка архива

Архив можно передать через веб-интерфейс или REST API. Оба способа запускают один конвертер и сохраняют результат в подготовленной конфигурации без автоматического применения.

Через веб-интерфейс

Для загрузки UCS и просмотра отчёта:

1. Откройте экран `Резервные копии` в группе «Управление».
2. В блоке «Импорт из F5 BIG-IP» выберите файл `.ucs` и нажмите `Конвертировать и импортировать` .
3. Дождитесь окончания загрузки и конвертации.
4. Изучите «Отчёт конвертации UCS» — таблицу с колонками «Важность», «Объект», «Сообщение».

Через REST API

Для автоматического импорта передайте UCS как `multipart/form-data`:

```
curl -X POST "https://<management-ip>/api/v1/config/import-ucs" \
-H "Authorization: Bearer $TOKEN" \
-F "file=@migration.ucs"
```

Операция доступна только роли `admin`. Сокращённый ответ содержит сконвертированную конфигурацию и отчёт:

```
{
  "config": {
    "system": {
      "hostname": "bigip1",
      "management": {
        "address": "192.0.2.10/24",
        "gateway": "192.0.2.1"
      }
    }
  },
  "report": [
    {
      "severity": "info",
      "entity": "system",
      "message": "Hostname: bigip1, management: 10.1.0.11/16"
    },
    {
      "severity": "warning",
      "entity": "system",
      "message": "F5 management address 10.1.0.11/16 not applied -- this appliance keeps its own
management addressing; change it deliberately after the migration if required"
    }
  ]
}
```

В этом примере `192.0.2.10/24` и `192.0.2.1` — действующие адрес и шлюз Амибиг. Импортированный адрес F5 `10.1.0.11/16` присутствует только в отчёте и не попадает в подготовленную конфигурацию.

Уровни сообщений отчёта:

Уровень	Значение
<code>info</code>	объект успешно сконвертирован
<code>warning</code>	объект сконвертирован с ограничениями или найден не полностью
<code>skipped</code>	объект намеренно пропущен (нет эквивалента или он не нужен)

Что конвертируется

Поддерживаемые объекты F5 преобразуются следующим образом:

Объект F5	Результат в Амибиг
<code>sys global-settings</code> , <code>sys dns</code> , <code>sys ntp</code> , часовой пояс	секция <code>system</code> : имя устройства, DNS-серверы, NTP-серверы и часовой пояс
<code>sys management-ip</code> , <code>sys management-route</code>	адрес и шлюз F5 фиксируются в отчёте, но не применяются; Амибиг сохраняет текущую адресацию управления
Интерфейсы, транки, VLAN, self-IP, маршруты	секция <code>network</code> ; интерфейс управления Амибиг сохраняется из действующей конфигурации
SSL-профили <code>client-ssl/server-ssl</code> топологий SSLO	клиентский и серверный TLS-профили, MITM CA
<code>sys file ssl-cert</code> / <code>ssl-key</code>	файлы сертификатов и ключей извлекаются из архива и сохраняются в <code>/config/ssl</code>
Сервисы SSLO (<code>ltm profile service</code> типов <code>icap</code> и <code>inline</code>) с их пулами и мониторами	ICAP- и inline-сервисы с участниками пулов
Цепочки сервисов SSLO	секция <code>service-chains</code>
Политика SSLO (правила, условия, действия, bypass)	секция <code>policy</code>
Виртуальные серверы и топология SSLO	секция <code>topology</code> (слушатель и режим dataplane)
Пользовательские iRules (<code>ltm rule</code>)	aRules: Tcl-код переносится как есть, правило включается с описанием «Imported from F5 iRule ...»
Пул удалённого syslog (<code>remote_syslog-ng</code>)	приёмники syslog в секции <code>logging</code>
SNMP-community (<code>sys snmp</code>)	community только для чтения в секции <code>snmp</code>

Секции, у которых нет эквивалента в UCS (например, `ha` и `rate-shaping`), создаются со значениями по умолчанию — HA-сопряжение после миграции настраивается заново.

Что не конвертируется

Следующие объекты требуют ручной настройки или не имеют аналога в Амибиг:

- **Интерфейс и адресация управления F5** — не применяются. После импорта Амибиг продолжает использовать свой текущий интерфейс управления, адрес и шлюз; сведения об адресе F5 остаются в отчёте.
- **Учётные записи и ACL управления F5** — вместо них создаются запись `admin` и разрешающий ACL по умолчанию. Если в действующей конфигурации уже

есть запись `admin` с хешем пароля, этот хеш сохраняется. Остальные учётные записи создайте заново на экране `Пользователи`.

- **Автосгенерированные iRules SSL Orchestrator** (`sslo_*`, `ssloS_*`, `ssloT_*` и т.п.) — их логику выполняет сам движок Амибиг, переносить их не нужно; в отчёте они помечены `skipped`.
- **Пустые iRules** — пропускаются.
- **Профили доступа APM** — эквивалента нет, помечаются `skipped`.
- Прочие объекты, специфичные для F5 и не имеющие аналога, попадают в отчёт с уровнем `skipped` или `warning`.

Проверка перед применением

После импорта действующая (`running`) конфигурация не изменяется. Перед применением подготовленной конфигурации:

1. Просмотрите отчёт конвертации, обратите внимание на `warning` и `skipped`.
2. Убедитесь, что в разделах `Система` и `Сеть` сохранены текущие интерфейс, адрес и шлюз управления Амибиг.
3. Проверьте секции в веб-интерфейсе: `Сеть` (интерфейсы, VLAN, self-IP, маршруты — имена портов на Амибиг отличаются от имён F5 и могут требовать ручной правки), `SSL/TLS` (профили и извлечённые сертификаты), `Сервисы`, `Цепочки сервисов`, `Политики`, `Топология`.
4. Просмотрите различия между подготовленной и действующей конфигурациями.
5. Проверьте ожидаемые сценарии на экране `Диагностика` в блоке `Объяснение потока`.
6. Примените конфигурацию и подтвердите её — работает стандартный механизм применения и подтверждения с таймером отката.

История импортов сохраняется на экране `История` — при неудачной миграции можно восстановить прежний снимок.

Ограничения совместимости aRules

iRules переносятся с исходным Tcl-кодом, но среда выполнения aRules поддерживает подмножество команд и событий F5 iRules. После миграции сверьте каждое правило с перечнем поддерживаемых событий и команд; обзор модели совместимости — в главе aRules. Неподдерживаемые команды выявляются при валидации правила на экране `aRules`.

Диагностика

Раздел помогает проверить, как Амибиг обрабатывает трафик: посмотреть события потоков, смоделировать решение политики, снять рсар-захват и собрать журналы. Инструменты диагностики полезны как при первичной настройке, так и при разборе инцидентов.

Начинайте с экрана **Трафик** и **Объяснение потока** , затем переходите к захвату пакетов и журналам, если нужно подтвердить поведение на уровне сети.

Объяснение потока

Блок **Объяснение потока** на экране **Диагностика** моделирует решение политики без передачи реального трафика. Результат показывает применённое правило, режим TLS, сервисную цепочку и участника пула.

Амибиг

Одиночный узел

Обзор

Трафик

Система

Сеть

SSL/TLS

Топология

Инспекция

Мониторинг

НА

Логирование

Просмотр логов

Наблюдаемость

Диагностика

SNMP

Управление

Применить конфигурацию

Документация

admin

admin

Диагностика

Диагностический архив

Плагин Wireshark

Захват пакетов и сетевая диагностика

Объяснение потока

Смоделируйте поток на текущей конфигурации AMIBIG и посмотрите точное правило, цепочку и выбор service/member, которые будут применены.

Область конфигурации

IP источника

Порт источника

IP назначения

Порт назначения

Протокол

SNI / Host

Гео источника

Гео назначения

Объяснить поток

Сбросить

default rule 'demo-default-bypass' => passthrough with no service chain

Область конфигурации	running	Совпавшее правило	demo-default-bypass
Действие по конфигу	passthrough	Фактическое действие	passthrough
TLS-режим	direct-forward-without-chain	Разрешённая цепочка	Нет
Гео источника	Нет	Гео назначения	Нет
Совпавшие URL-категории	Категории не совпали		

Трассировка правил

#1 MISS demo-site-0001-ngfw1-dlp

#2 MISS demo-site-0002-ngfw1

#3 MISS demo-site-0003-ngfw2

#4 MISS demo-site-0004-ngfw2-dlp

#5 MISS demo-site-0005-ngfw1-ngfw2

Зачем это нужно

Используйте моделирование, чтобы ответить на вопрос:

«Если клиент 10.1.1.50 пойдёт на site-0005.bench.local:11443 , что именно сделает Амибиг?»

В результате отображаются:

- сработавшее правило;
- было ли это обычное правило или `policy.default-rule` ;
- действие по конфигурации и фактическое действие;
- режим TLS;
- выбранную сервисную цепочку;
- выбранного участника пула;
- примечания о неподдерживаемых сочетаниях и запасных вариантах поведения;
- полную трассировку правил.

Для UDP и ICMP результат соответствует действиям dataplane: `bypass` , `passthrough` или `reject` . Для остальных IP-протоколов (`other`) показываются только `passthrough` и `reject` . Действие `intercept` для трафика, отличного от TCP, помечается как неподдерживаемое.

Выбор конфигурации

Моделирование можно выполнить для одного из двух состояний конфигурации:

Конфигурация	Значение
<code>running</code>	Действующая конфигурация
<code>candidate</code>	Подготовленная, ещё не применённая конфигурация

Сравните результаты для `running` и `candidate` , чтобы проверить влияние подготовленных изменений до их применения.

Поля входа

Минимально укажите:

- IP-адрес источника;
- IP-адрес назначения;
- порт назначения;
- протокол.

Дополнительно можно указать:

- порт источника;

- SNI ;
- геопризнак источника;
- геопризнак назначения.

Для сценариев HTTPS и SMTPS лучше заполнять SNI , иначе часть правил, использующих SNI в условии, не будет применена.

Как читать результат

Результат объяснения потока разделён на итоговое решение, прохождение сервисов и диагностические предупреждения. Сначала проверьте итог, затем сопоставьте его с выбранным правилом и состоянием сервисов.

Итог

Показывает итоговое решение:

- какое правило сработало;
- сработало ли явное default-rule ;
- какое действие задано в конфигурации;
- какое действие реально будет применено.

Прохождение сервисов

Показывает цепочку сервисов и их состояние:

- доступность;
- эффект сервиса;
- выбранного участника пула;
- причину выбора участника.

Целевой сервер

Если поток попадает на виртуальный сервер или выбор целевого сервера зависит от пула/persistence, этот блок показывает итоговую цель.

Трассировка правил

Самая полезная часть для разбора. Для каждого правила политики видно:

- имя правила и действие;
- включено правило или отключено;
- список проверок;
- почему условие совпало или не совпало.

Практический сценарий

Перед применением изменений проверьте ожидаемый поток следующим образом:

1. Сохраните изменения в подготовленной конфигурации.
2. Откройте экран `Диагностика`.
3. Выберите `candidate`.
4. Заполните адреса, порт и `SNI`.
5. Нажмите `Объяснить поток`.
6. Если результат не соответствует ожиданию, вернитесь на экраны `Политики`, `Цепочки сервисов` или `Топология`.
7. Повторите моделирование и применяйте конфигурацию только после получения ожидаемого результата.

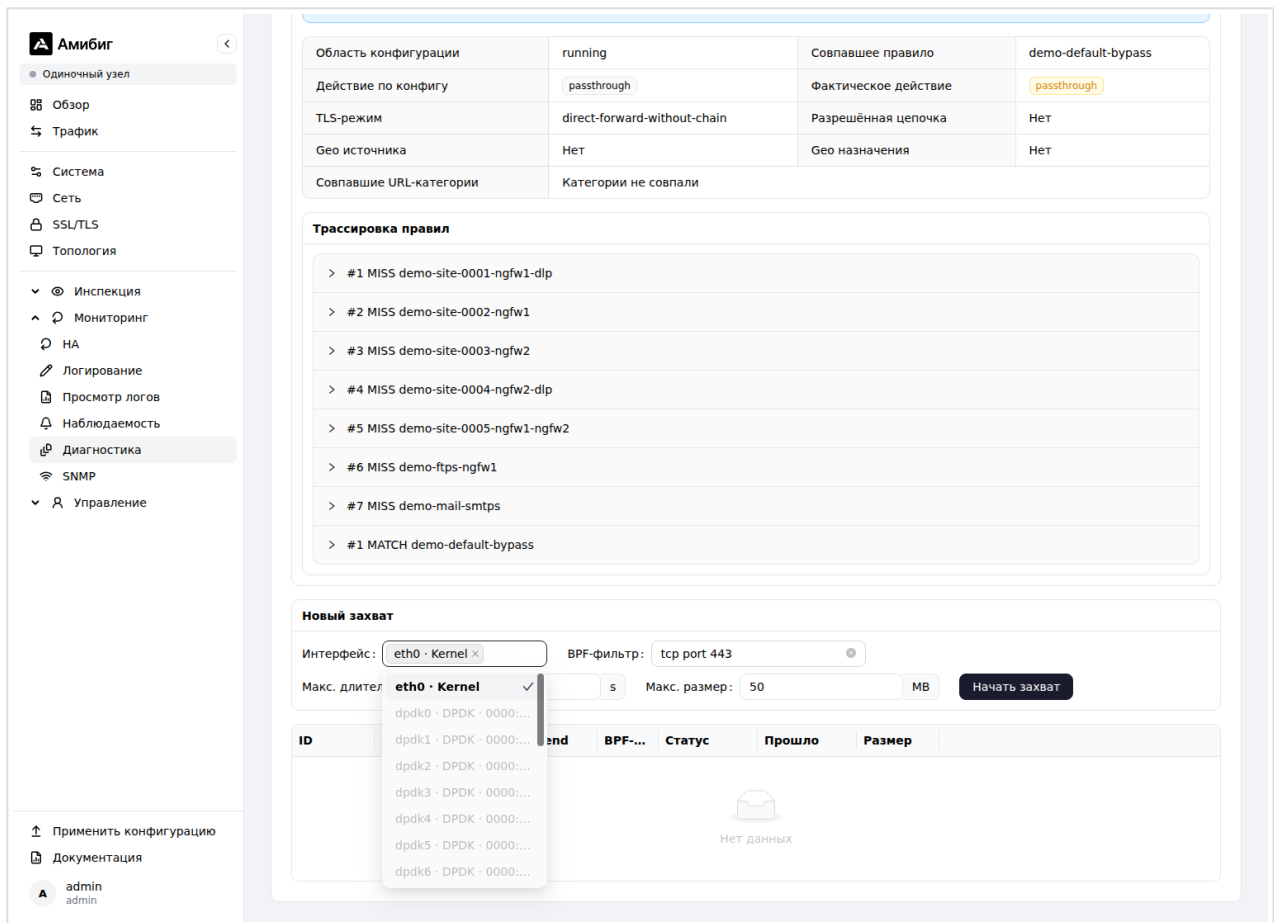
Сценарии применения

Используйте объяснение потока в следующих случаях:

- перед применением новой политики;
- при расследовании неожиданного `bypass / intercept`;
- при проверке действия при недоступности сервиса;
- при смене сервисной цепочки;
- при сравнении действующей и подготовленной конфигураций;
- при подготовке демонстрационных и нагрузочных сценариев.

Захват пакетов

Амибиг поддерживает встроенный захват пакетов на экране `Диагностика`. Захват можно запускать на интерфейсах, которые присутствуют в конфигурации, а результат скачивать в формате обычного PCAP или PCAP с метаданными.



На скриншоте показан блок подготовки захвата: выбор интерфейсов, BPF-фильтр и лимиты перед запуском.

Где это находится

Откройте экран **Диагностика** и используйте блок **Новый захват** под блоком **Объяснение потока**.

Доступные действия:

- выбрать один или несколько интерфейсов;
- задать BPF-фильтр;
- ограничить длительность захвата;
- ограничить размер захвата;
- остановить захват;
- скачать обычный PCAP или PCAP с метаданными;
- скачать Lua-диссектор для Wireshark.

Что нужно знать про интерфейсы

Список интерфейсов для захвата формируется из текущей конфигурации:

- интерфейсы ядра Linux из `network.interfaces`;

- DPDK-порты dataplane из `network.interfaces` ;
- агрегированные каналы (транки) из `network.trunks` .

Если интерфейс не описан в конфигурации, в списке он не появится. Имена VLAN нельзя использовать как точки захвата. Для проверки трафика VLAN выберите физический интерфейс или транк, назначенный родительским объектом в `network.vlans` .

Захват с агрегированного канала (LACP)

Захват с транка выполняется на самом bond-порту, а не на отдельных портах-участниках: в DPDK порты-участники подчинены драйверу агрегации и недоступны для захвата по отдельности. Bond-порт агрегирует трафик всех участников, поэтому в один захват попадают пакеты со всех входящих в транк портов (включая служебные кадры LACP).

Если агрегированный канал ещё не поднят в dataplane, запуск захвата на нём завершится ошибкой — сначала убедитесь, что транк активен (например, по телеметрии на экране `Сеть`).

Права и возможности ОС

Компоненту захвата нужны права на низкоуровневый доступ к выбранному интерфейсу Linux. Служба должна запускаться с необходимыми Linux capabilities, например `CAP_NET_RAW` и `CAP_NET_ADMIN` , либо вызывать `tcpdump` или `dumpcap` с такими правами.

Если прав недостаточно, форма в веб-интерфейсе остаётся доступной, но запуск захвата завершается ошибкой.

BPF и DPDK

Поддержка фильтра зависит от типа выбранного интерфейса:

- Для захвата на **интерфейсе ядра Linux** BPF-фильтр поддерживается.
- Для захвата на **DPDK-порте** BPF-фильтр не поддерживается, потому что захват идёт через управляющий сокет dataplane.
- Запрос с интерфейсами ядра Linux и DPDK будет отклонён. Для каждого backend запускайте отдельный захват.

Перед запуском через API запросите `GET /api/v1/diagnostics/interfaces` . В ответе для каждого интерфейса указаны `capture_backend` и признак `supports_bpf` .

Типовой рабочий сценарий

Чтобы записать минимальный воспроизводимый фрагмент трафика:

1. Откройте экран `Трафик` и убедитесь, что поток воспроизводим.
2. На экране `Диагностика` создайте захват на нужном интерфейсе.

3. При необходимости задайте BPF-фильтр, например `tcp port 443`.
4. Воспроизведите проблему.
5. Остановите захват.
6. Скачайте:
 - обычный PCAP для классического анализа пакетов;
 - PCAP с метаданными, если нужна привязка к решениям политики.

PCAP с метаданными

PCAP с метаданными особенно полезен, когда нужно связать пакеты с решением движка политики.

Он дополняет трафик метаданными, такими как:

- сервисная цепочка;
- SNI;
- действие;
- признаки направления;
- служебные метаданные Амибиг в конце записи.

Для анализа в Wireshark скачайте встроенный Lua-диссектор и положите его в каталог плагинов Wireshark.

Полезные примеры BPF

Используйте узкий фильтр, чтобы уменьшить размер файла и упростить анализ:

Задача	Фильтр
Только HTTPS	<code>tcp port 443</code>
Только ICAP	<code>tcp port 1344</code>
Один клиент	<code>host 10.1.1.50</code>
Пара клиент/сервер	<code>host 10.1.1.50 and host 192.168.70.20</code>
Исключить SSH и HTTPS-интерфейс управления	<code>not (tcp port 22 or tcp port 443)</code>

Практические рекомендации

При подготовке захвата учитывайте следующие рекомендации:

- Для проблем с TLS сопоставляйте результат блока `Объяснение потока` с захватом пакетов.
- Для проблем с сервисной цепочкой полезно снимать трафик одновременно на входе и на VLAN сервисной стороны.

- Скачивайте файлы до перезагрузки узла или очистки списка захватов.

Журнал событий и его экспорт

Амибиг разделяет две связанные, но разные задачи:

- **подсистема журналирования** в конфигурации устройства (секция `logging`);
- **экран Просмотр логов** в веб-интерфейсе для оперативной работы с событиями.

Просмотр логов

Все Загрузка Система Аудит Конфигурация НА Dataplane

Авто CSV Удалить

События (1ч) **11** Проблемы (1ч) **0** Аномалии DP **0**

Последний apply: 2026-08-19 16:41:27 - AMB-6030: Backup...
 Последний failover: Нет
 Главный повтор: Нет

Канал Категория Важность Поиск Начальная дата → Конечная дата Поиск

Время	Категория	Важность	Источник	Сообщение
> 2026-08-19 16:42:13	audit	info	api-audit	POST /api/v1/diagnostics/flow-explain -> 200
> 2026-08-19 16:42:12	audit	info	api-audit	POST /api/v1/config/topology-templates/compose -> 200
> 2026-08-19 16:41:37	audit	info	api-audit	POST /api/v1/diagnostics/flow-explain -> 200
> 2026-08-19 16:41:35	audit	info	api-audit	POST /api/v1/config/topology-templates/compose -> 200
> 2026-08-19 16:41:28	auth audit	info	auth	AMB-2004: Login successful
> 2026-08-19 16:41:27	audit	info	api-audit	POST /api/v1/diagnostics/archive -> 200
> 2026-08-19 16:41:27	audit	info	diagnostics	AMB-2030: Diagnostic archive created and served
> 2026-08-19 16:41:27	diagnostics	info	diagnostics	AMB-1045: Diagnostic archive created
> 2026-08-19 16:41:27	config	info	backup	AMB-6030: Backup exported
> 2026-08-19 16:41:27	auth audit	info	auth	AMB-2004: Login successful
> 2026-08-19 16:41:27	audit	info	api-audit	DELETE /api/v1/logs -> 200

< 1 >

↑ Применить конфигурацию
 📄 Документация
 admin admin

Где смотреть события

Текущая операционная работа выполняется на экране **Просмотр логов**.

Он поддерживает:

- фильтрацию по категории;
- фильтрацию по уровню важности;
- поиск по тексту;
- ограничение временного диапазона;
- выгрузку в CSV;
- очистку журнала;
- автообновление списка.

Категории событий

Категория указывает источник и назначение события:

Категория	Что входит
<code>system</code>	Запуск, состояние компонентов, системные сообщения
<code>config</code>	Изменения подготовленной и действующей конфигураций, применение и откат
<code>auth</code>	Вход в систему, списки доступа, смена пароля, отказ в доступе
<code>security</code>	TLS, лицензия, действия с обновлениями и безопасностью
<code>traffic</code>	Решения политики, мониторы доступности и события обработки трафика
<code>ha</code>	Переключение ролей и синхронизация HA
<code>observability</code>	Жизненный цикл оповещений и события подсистемы наблюдаемости

Уровни важности

Уровень помогает отделить штатные события от состояний, требующих реакции:

Уровень	Когда используется
<code>debug</code>	Низкоуровневая диагностика
<code>info</code>	Нормальные рабочие события
<code>warning</code>	Деградация или аномалия
<code>error</code>	Ошибка, влияющая на функциональность
<code>critical</code>	Потеря сервиса или необходимость срочного вмешательства

Коды сообщений

Операторские события снабжены стабильными кодами вида `AMB-NNNN` — код виден префиксом в тексте сообщения и в поле `msg_id` данных события. При обращении в техническую поддержку Amicon указывайте код события: он не меняется между версиями. Полный перечень кодов приведён на справочной странице Коды сообщений.

Кроме кода, события обработки трафика несут адреса и порты источника и назначения отдельными полями (`src_ip` , `src_port` , `dst_ip` , `dst_port`) — эти поля попадают и в экспортируемые записи, что упрощает корреляцию в SIEM.

Выгрузка и архив

Для передачи результатов диагностики используйте CSV-выгрузку событий или диагностический архив. CSV подходит для выбранных записей журнала, архив содержит дополнительные сведения о системе и конфигурации.

Выгрузка в CSV

С экрана `Просмотр логов` можно выгрузить отфильтрованный набор событий в CSV. Используйте выгрузку для:

- передачи в техническую поддержку Amicon;
- разбора инцидента вне устройства;
- аудита изменений конфигурации и событий входа в систему.

Диагностический архив

Для более полного набора артефактов используйте кнопку **Диагностический архив** на экране `Диагностика`.

Диагностический архив подходит, когда нужны одновременно:

- системные события;
- конфигурация;
- результаты захвата пакетов;
- служебные файлы для расследования.

Настройка журналирования

Через JSON/API секция `logging` отдельно управляет:

- уровнем системного журнала (`logging.system`);
- журналом аудита (`logging.audit`);
- журналированием SSL-событий (`logging.ssl-events`);
- экспортом на внешние syslog-приёмники (`logging.export.syslog`);
- экспортом в Kafka (`logging.export.kafka`).

Все перечисленные параметры доступны на экране `Логирование`, включая защищённое подключение к Kafka и фильтры передачи на приёмники.

Если задача — оперативно понять проблему, начинайте с экрана `Просмотр логов`. Если задача — отправить поток событий наружу, переходите к экрану `Логирование` или к конфигурации секции `logging`.

Экспорт на syslog-приёмники

После успешного применения конфигурации Амибиг перечитывает `logging.export.syslog` из действующей (`running`) конфигурации и начинает передавать новые операторские события на настроенные приёмники.

Поддерживаемые значения:

- `protocol` : `udp` или `tcp` ;
- `format` : `rfc3164` , `rfc5424` , `cef` или `leef` .

В форматах `rfc3164` и `rfc5424` тело сообщения — один JSON-объект с полями `timestamp` , `category` , `severity` , `source` , `message` , `data` и служебными идентификаторами (`request_id` , `txn_id` , поколения конфигурации). Форматы `cef` и `leef` предназначены для SIEM: код события `AMB-NNNN` попадает в идентификатор записи, адреса и порты источника/назначения — в стандартные атрибуты `src / spt / dst / dpt` .

Фильтр на приёмник

Для каждого приёмника можно ограничить перечень пересылаемых событий фильтром `filter` по категориям, уровням важности и кодам сообщений. Пустой или отсутствующий фильтр принимает все события; каждая непустая часть фильтра требует совпадения.

Пример приёмника с фильтром:

```
{
  "logging": {
    "export": {
      "syslog": [
        {
          "name": "siem-primary",
          "address": "192.0.2.30",
          "port": 514,
          "protocol": "udp",
          "format": "cef",
          "filter": {
            "categories": ["auth", "security"],
            "severities": ["warning", "error", "critical"],
            "msg-ids": []
          }
        }
      ]
    }
  }
}
```

В веб-интерфейсе фильтр настраивается на экране `Логирование` : у каждого приёмника есть блок «Отбор событий» с полями «Категории событий», «Уровни важности» и «Коды сообщений» (значения вида `AMB-1003`).

Поведение при доставке:

- локальная запись в `events.db` всегда выполняется первой;

- внешняя доставка не блокирует локальную запись: недоступность приёмника не приводит к потере событий в локальном журнале;
- смена приёмников применяется только после успешного применения конфигурации.

Экспорт в Kafka

Помимо syslog, события можно отправлять в Kafka — секция `logging.export.kafka`. Поддерживаются форматы `json` (по умолчанию), `cef` и `leef`; фильтр `filter` работает так же, как у syslog-приёмника.

```
{
  "logging": {
    "export": {
      "kafka": {
        "enabled": true,
        "brokers": ["kafka-1.example.net:9092", "kafka-2.example.net:9092"],
        "topic": "amibig-events",
        "format": "json",
        "filter": {
          "categories": [],
          "severities": [],
          "msg-ids": []
        }
      }
    }
  }
}
```

Для брокеров с аутентификацией доступны поля `security-protocol`, `sasl-mechanism`, `sasl-username` и `sasl-password` (например, `security-protocol: "SASL_SSL"`; учётные данные подставьте свои — в примерах они не приводятся). Без этих полей используется подключение без аутентификации.

Эти параметры настраиваются и на экране `Логирование`: поле «Защита подключения к брокеру» (значения `PLAINTEXT`, `SSL`, `SASL_PLAINTEXT`, `SASL_SSL`), а для SASL-вариантов — «Механизм проверки подлинности», «Учётная запись» и «Пароль». Там же задаётся фильтр «Отбор событий» для Kafka.

Как и для syslog, доставка в Kafka не блокирует локальную запись событий и применяется после успешного применения конфигурации.

Маршрутизация оповещений

Подсистема оповещений записывает события своего жизненного цикла в категорию `observability`, используя существующий канал журналирования:

- если у правила включён маршрут `local-log`, событие попадает в локальный журнал `events.db`;
- если включён маршрут `syslog`, та же запись экспортируется на настроенные syslog-приёмники;
- если включён маршрут `snmp`, параллельно отправляется SNMP trap-сообщение `alert-open` или `alert-clear`.

То есть оповещения не обходят подсистему журналирования и не заводят отдельный канал экспорта. Настройка правил оповещений описана в разделе Наблюдаемость.

Связанные разделы

Сведения о кодах событий и внешнем мониторинге приведены в следующих разделах:

- Коды сообщений
- Наблюдаемость: оповещения и аналитика
- SNMP-мониторинг и trap-сообщения

Мониторинг трафика

Экран **Трафик** показывает события, которые dataplane и вспомогательные компоненты формируют при обработке соединений. Используйте его для оперативной проверки применённого действия, правила и сервисной цепочки.

Амибиг

Одиночный узел

Обзор

Трафик

Система

Сеть

SSL/TLS

Топология

Инспекция

Мониторинг

Управление

Применить конфигурацию

Документация

admin

Трафик

Мониторинг соединений в реальном времени

Фильтры

16:42:17 Авто

Время	Клиент	Сервер	Хост	Протокол	Действие	Цепочка	Правило
> 2026-08-19 16:41:27	198.18.0.55 IPv4	203.0.113.10:443	updates.example.net	TLS	bypass	demo-bypass	
> 2026-08-19 16:41:27	198.18.0.44 IPv4	192.168.70.20:11443	site-0005.bench.local	TLS	intercept	demo-ngfw1-ngfw2	
> 2026-08-19 16:41:27	198.18.0.33 IPv4	192.168.70.20:10443	site-0004.bench.local	TLS	intercept	demo-ngfw2-dlp	
> 2026-08-19 16:41:27	198.18.0.22 IPv4	192.168.70.20:12443	site-0003.bench.local	TLS	intercept	demo-ngfw2	
> 2026-08-19 16:41:27	198.18.0.11 IPv4	192.168.70.20:8443	site-0002.bench.local	TLS	intercept	demo-ngfw1	
> 2026-08-19 16:41:27	198.18.0.5 IPv4	192.168.70.20:443	site-0001.bench.local	TLS	intercept	demo-ngfw1-dlp	

< 1 >

Что видно на экране

Для каждого потока отображаются:

- время;
- IP-адрес клиента;
- IP-адрес и порт сервера;
- SNI или хост;
- протокол;
- действие;
- сервисная цепочка.

По нажатию на строку можно развернуть полные метаданные события.

Панель фильтрации

Блок **Фильтры** над списком (разворачивается и сворачивается) позволяет сузить поток событий без выгрузки:

- адрес клиента;
- адрес сервера;
- SNI / хост;
- цепочка;
- правило;
- протокол — множественный выбор из TCP , UDP , ICMP , ICMPv6 (плюс значения, встреченные в живых событиях);
- действие — множественный выбор из intercept , bypass , passthrough , reject .

Кнопка **Сбросить фильтры** возвращает полный список. Фильтры применяются в браузере только к уже полученным событиям и не изменяют сбор данных на устройстве.

Откуда берутся данные

Экран агрегирует события, которые приходят в `/api/v1/traffic` :

- события потоков dataplane;
- события мониторов доступности;
- статистику пулов.

Поэтому экран полезен не только для решений по HTTP/TLS-потокам, но и для понимания состояния участников пулов.

Когда использовать экран **Трафик**

Откройте экран **Трафик**, когда требуется:

- проверить, что трафик проходит через Амибиг;
- увидеть действие `intercept`, `bypass`, `passthrough` или `reject`;
- определить выбранную цепочку без запуска захвата пакетов;
- проверить хост, SNI и правило для фактического соединения.

Связка с другими инструментами

Для последовательной диагностики сопоставьте события с результатами двух инструментов:

1. **Объяснение потока** — чтобы смоделировать ожидаемое поведение;
2. **захват пакетов** — чтобы увидеть реальные пакеты и детали TLS handshake.

Если фактическое событие отличается от результата моделирования, проверьте выбранное состояние конфигурации, входные адреса, порт, протокол и SNI. Затем сопоставьте время события с моментом последнего применения конфигурации.

Устранение неполадок

Раздел содержит типовые сценарии диагностики: проблемы доступа к веб-интерфейсу, ошибки TLS-инспекции, отсутствие трафика в сервисной цепочке, сбои обновлений и вопросы НА. Каждый сценарий связывает симптом с вероятной причиной и действием администратора.

Используйте этот раздел после проверки базовой конфигурации сети, политики и состояния сервисов.

Типовые проблемы и решения

Раздел описывает типовые проблемы эксплуатации Амибиг: симптом, вероятную причину и конкретное действие. Для пошаговой диагностики прохождения трафика используйте раздел Диагностика прохождения трафика.

Команды API предполагают, что сертификат интерфейса управления добавлен в доверенные, а действующий токен сохранён в переменной `TOKEN`.

Проблемы доступа к веб-интерфейсу

Начните с проверки HTTPS-соединения к адресу управления, затем проверьте сертификат и список доступа. Это позволяет отделить сетевую недоступность от ошибки доверия или блокирования по ACL.

Веб-интерфейс не открывается

Сначала определите, устанавливается ли HTTPS-соединение с адресом управления:

Симптом	Возможная причина	Решение
Браузер показывает <code>Connection refused</code>	Служба управления не запущена	Проверьте состояние служб через консоль устройства; если служба не запускается, обратитесь в техническую поддержку Amicon
Тайм-аут подключения	Указан неверный адрес интерфейса управления	Проверьте <code>system.management</code> через консоль устройства

Симптом	Возможная причина	Решение
Тайм-аут подключения	Межсетевой экран блокирует доступ	Разрешите HTTPS к порту 443 с рабочей станции администратора
Браузер не доверяет сертификату	Используется первоначальный самоподписанный сертификат	Установите сертификат корпоративного УЦ для веб-интерфейса или добавьте УЦ Амибиг в доверенные на рабочей станции
HTTP 403 при изменении конфигурации	Закончился демонстрационный период или лицензия недействительна	Загрузите действующую лицензию

Диагностика:

С рабочей станции проверьте готовность API и журнал. Примеры предполагают, что сертификат интерфейса управления добавлен в доверенные:

```
### Проверка доступности системы
curl -s "https://<management-ip>/api/v1/system/ready"
### Проверка журналов через API
curl "https://<management-ip>/api/v1/logs?category=system&severity=error&limit=20" \
-H "Authorization: Bearer $TOKEN"
```

Невозможно войти в систему

Если форма входа открывается, сопоставьте сообщение с причиной:

Симптом	Возможная причина	Решение
"Invalid credentials"	Неправильный пароль	Сбросьте пароль через консоль устройства
"Invalid credentials"	Пользователь не существует	Проверьте секцию <code>users</code> в конфигурации
Токен авторизации не принимается	Токен истёк	Повторите аутентификацию через <code>/api/v1/auth/login</code>

Замена сертификата HTTPS веб-интерфейса

Сертификат HTTPS-доступа заменяется на вкладке `Сертификат WebUI` экрана `SSL/TLS`. Он защищает только интерфейс управления и не влияет на сертификаты TLS-инспекции dataplane.

Симптом	Возможная причина	Решение
Браузер предупреждает о ненадёжном соединении	Используется самоподписанный сертификат, созданный при первой загрузке	Создайте CSR на вкладке Сертификат WebUI , подпишите его в корпоративном УЦ и загрузите выпущенный сертификат
Импорт отклонён с ошибкой про ECU	В расширении Extended Key Usage сертификата нет serverAuth	Перевыпустите сертификат с serverAuth в ECU (сертификат без расширения ECU принимается)
Импорт отклонён с ошибкой про цепочку	Приложенная цепочка промежуточных сертификатов не проходит проверку	Приложите корректную цепочку. Цепочка необязательна, но без неё браузер не построит доверие к сертификату промежуточного УЦ
Браузер не доверяет сертификату по IP-адресу	В SAN сертификата нет IP управления	При генерации CSR на устройстве адрес управляющего интерфейса подставляется в SAN автоматически (в том числе при DHCP); проверить SAN в карточке текущего сертификата

Поддерживается PEM с сертификатом, закрытым ключом и необязательной цепочкой, а также PKCS#12 с паролем. После замены текущий сеанс завершается. Повторно откройте интерфейс, проверьте новый сертификат и выполните вход.

Ошибки SSL/TLS

Сопоставьте сообщение клиента с этапом установления соединения: доверием к УЦ, согласованием версии и набора шифров или проверкой origin-сервера.

Клиент не может подключиться через Амибиг

Проверьте сообщение клиента и соответствующий параметр TLS:

Симптом	Возможная причина	Решение
ERR_SSL_PROTOCOL_ERROR	Система обработки SSL/TLS не запущена	Проверьте состояние устройства на экране Обзор и события категории security

Симптом	Возможная причина	Решение
ERR_CERT_AUTHORITY_INVALID	Сертификат УЦ Амибиг не установлен на клиенте	Распространите сертификат УЦ через GPO или MDM
ERR_SSL_VERSION_OR_CIPHER_MISMATCH	Несовместимые наборы шифров	Проверьте <code>ssl.client-profile.cipher-rule</code> и <code>min-version</code>
ERR_CERT_DATE_INVALID	Несинхронизированные часы на Амибиг	Настройте NTP в секции <code>system.ntp</code>
Сертификат с неправильным CN	Ошибка генерации MITM-сертификата	Проверьте сертификат УЦ в разделе <code>SSL/TLS</code>

Диагностика SSL:

```

### Проверка TLS handshake с клиентского сегмента через Амибиг
openssl s_client -connect target.example:443 -servername target.example

### Проверка SSL-логов через API
curl "https://<management-ip>/api/v1/logs?category=security&search=ssl&limit=20" \
-H "Authorization: Bearer $TOKEN"

```

Некоторые сайты не работают через Амибиг

Для приложений с особыми требованиями TLS может потребоваться отдельное правило без расшифрования:

Симптом	Возможная причина	Решение
Приложение использует certificate pinning	Проверка закреплённого сертификата несовместима с TLS-инспекцией	Добавьте правило <code>passthrough</code> по SNI или адресу назначения
Сайт требует клиентский сертификат	Соединение mTLS не должно расшифровываться	Добавьте правило <code>passthrough</code> по SNI
Соединения WebSocket разрываются	Истекает тайм-аут бездействия	Увеличьте тайм-аут или создайте правило <code>passthrough</code> для нужного SNI

Добавление passthrough-правила:

```

{
  "name": "passthrough-pinned-sites",
  "enabled": true,
  "match": {
    "sni": ["*.apple.com", "*.icloud.com", "updates.microsoft.com"]
  },
  "action": "passthrough"
}

```

Соединения ГОСТ-клиентов

Соединения ГОСТ-TLS поддерживают те же действия политики, включая `passthrough`.

Симптом	Возможная причина	Решение
ГОСТ-сайты проходят без расшифрования	Инспекция ГОСТ-TLS не включена	Включите <code>gost-enabled</code> в секции <code>ssl</code> и настройте ГОСТ-УЦ (см. ГОСТ TLS)
Клиент не доверяет подменному ГОСТ-сертификату	ГОСТ-УЦ не установлен как доверенный на клиенте	Распространите сертификат ГОСТ-УЦ на клиентские устройства

Проблемы прохождения трафика

Если соединение не устанавливается, последовательно проверьте маршрутизацию, трансляцию адреса источника и параметры протокола. Начинайте с условия, которое соответствует наблюдаемой схеме подключения.

Соединения к серверам за шлюзом не устанавливаются при отключённой трансляции

При `origin-snat: none` трансляция адреса источника отключена. Сервер назначения видит адрес клиента, поэтому обратный маршрут должен проходить через Амибиг.

Симптом	Возможная причина	Решение
Соединения к удалённым серверам не устанавливаются, в журнале есть предупреждения	Глобально включён <code>origin-snat: none</code> , а обратный маршрут проходит мимо Амибиг	Верните глобальный режим <code>automap</code> , а <code>none</code> задавайте только для правил с симметричным маршрутом
Серверу нужен реальный адрес клиента, но часть серверов за шлюзом	Один глобальный режим на весь трафик	Задайте переопределение в правиле: <code>none</code> для напрямую доступных серверов, <code>automap</code> для остальных (см. Трансляция адреса источника)

Устройство не отвечает на ping Self-IP или VIP

Амибиг отвечает на ICMP echo-запросы к Self-IP и активным VIP. Если ответа нет, выполните следующие проверки:

1. Для Self-IP убедитесь, что адрес назначен нужному VLAN или интерфейсу на экране **Сеть** → **Self-IP** и конфигурация применена.
2. Для VIP убедитесь, что запрос приходит на активный узел. Резервный узел не публикует VIP и не отвечает на такие запросы.
3. Проверьте связность L2: ARP-запрос к Self-IP или VIP должен получать ответ, для VIP — от активного узла.
4. Транзитный ICMP проверяйте по правилам политики: он не обрабатывается как запрос к адресу устройства.

Особенности поведения:

- ответы на запросы к адресам устройства не отображаются на экране **Трафик** ;
- фрагментированные echo-запросы к Self-IP обрабатываются сетевым стеком dataplane.

Проблемы ресурсов

Сведения о нехватке памяти и вычислительных ресурсов доступны в системной статистике и журнале запуска dataplane. Сравните фактические лимиты с параметрами конфигурации и ожидаемой нагрузкой.

Dataplane не запускается при недостатке оперативной памяти

Пул соединений dataplane ограничивается доступной оперативной памятью. Пул, huperages и системный резерв должны размещаться одновременно. Если заданное значение `max-concurrent-sessions` не помещается, в журнал записывается предупреждение и фактический лимит уменьшается.

Если рассчитанного лимита недостаточно для рабочей нагрузки, увеличьте объём оперативной памяти (см. Системные требования).

Проблемы с сервисами

Для внешнего сервиса проверьте административное состояние, результаты монитора доступности и физический путь трафика. Ошибка на любом из этих уровней может исключить сервис из цепочки.

Inline-сервис (NGFW) не получает трафик

Сопоставьте наблюдаемый симптом с состоянием конфигурации и сервиса:

Симптом	Возможная причина	Решение
NGFW не получает пакеты	Конфигурация не применена	Примените конфигурацию через веб-интерфейс или API
NGFW не получает пакеты	Не настроены VLAN сервиса	Проверьте <code>network.vlans</code>
Трафик поступает на NGFW, но не возвращается	Неверный шлюз на выходной стороне сервиса	Проверьте <code>from-service.gateway</code>
Монитор показывает <code>down</code>	NGFW не отвечает на проверку	Проверьте доступность адреса и порта сервиса

Диагностика inline-сервиса:

```

### Проверка мониторов доступности через API
curl "https://<management-ip>/api/v1/traffic/health" \
  -H "Authorization: Bearer $TOKEN"

### Проверка журналов сервисов
curl "https://<management-ip>/api/v1/logs?category=traffic&limit=20" \
  -H "Authorization: Bearer $TOKEN"

```

На экране **Диагностика** запустите захват на физических портах или транках, назначенных родительскими интерфейсами входного и выходного VLAN сервиса. Имена VLAN нельзя использовать как точки захвата.

ICAP-сервис (DLP) не работает

Проверьте доступность ICAP-сервера и ответ протокола:

Симптом	Возможная причина	Решение
ICAP connection refused	DLP-сервер не слушает порт 1344	Проверьте состояние сервиса на DLP-сервере
ICAP timeout	Сетевая недоступность DLP-сервера	Проверьте сетевую связность до DLP-сервера
ICAP 503 Service Unavailable	DLP-сервер перегружен	Проверьте нагрузку DLP и увеличьте пул
Трафик, отличный от HTTP, блокируется	ICAP применяется к трафику, отличному от HTTP	Проверьте конфигурацию: ICAP обрабатывает только HTTP

Диагностика ICAP:

```

### Проверка мониторов доступности
curl "https://<management-ip>/api/v1/traffic/health" \
  -H "Authorization: Bearer $TOKEN"

### Проверка журналов

```

```
curl "https://<management-ip>/api/v1/logs?category=traffic&search=icap&limit=20" \
-H "Authorization: Bearer $TOKEN"
```

Сервис находится в состоянии `down`, но доступен вручную

Проверьте параметры монитора доступности:

```
{
  "pool": {
    "monitor": {
      "type": "tcp",
      "interval": 5,
      "timeout": 16,
      "fail-threshold": 3,
      "recover-threshold": 2
    }
  }
}
```

- `fail-threshold: 3` — сервис переходит в состояние `down` после трёх неудачных проверок подряд.
- `recover-threshold: 2` — сервис возвращается в состояние `up` после двух успешных проверок.
- Убедитесь, что `timeout` больше `interval`.

Проблемы высокой доступности

Диагностика HA включает канал между узлами, взаимное доверие, роли и состояние синхронизации. Проверяйте их в этом порядке, чтобы не маскировать сетевую ошибку ручным запуском синхронизации.

HA не синхронизируется

Поля состояния API позволяют локализовать проблему канала HA:

Симптом	Возможная причина	Решение
<code>peer-state: down</code>	Канал HA не подключён	Проверьте физическое подключение интерфейса HA
<code>peer-state: down</code>	Неверно заданы адреса канала HA	Проверьте <code>ha.ha-link.address</code> и <code>ha.ha-link.peer-address</code>
<code>peer-state: connecting</code>	Не проходит TLS handshake	Проверьте сертификаты канала HA

Симптом	Возможная причина	Решение
<code>config-convergence.converged: false</code>	Поколения применённой конфигурации на узлах различаются	Проверьте <code>reason</code> , <code>mismatched-keys</code> и экран Просмотр логов на резервном узле

Диагностика HA:

```

### Статус HA через API
curl "https://<management-ip>/api/v1/ha/status" \
  -H "Authorization: Bearer $TOKEN"

### Проверка связи по HA link
ping -I eth5 10.10.0.2

### Логи HA через API
curl "https://<management-ip>/api/v1/logs?category=ha&limit=50" \
  -H "Authorization: Bearer $TOKEN"

```

Аварийное переключение не происходит

Проверьте роль узлов, публикацию VIP и синхронизацию конфигурации:

Симптом	Возможная причина	Решение
Резервный узел не публикует VIP	HA не включена	Проверьте <code>ha.enabled: true</code>
VRRP не переключается	Неверно задан приоритет	Проверьте приоритеты обоих узлов
VIP переключается, но трафик не проходит	Конфигурация не применена на резервном узле	Проверьте синхронизацию конфигурации
Оба узла активны	Потеряны все каналы HA	Восстановите каналы HA и исключите подачу клиентского трафика на один из узлов до согласования ролей

Конфигурация не синхронизируется

Для восстановления синхронизации:

- убедитесь, что на обоих узлах заданы `ha.enabled: true` и `peer-state: up`;
- проверьте поля `config-convergence`, `pending-config-txn`, `last-config-txn-ack` и `last-config-sync` в ответе `GET /api/v1/ha/status`;
- вносите изменения на активном узле: на резервном узле конфигурация доступна только для чтения;

- если `config-convergence.converged` остаётся `false`, запустите ручную синхронизацию через `POST /api/v1/ha/sync`, затем проверьте подтверждение и журнал событий.

Проблемы с лицензией

Причину отказа при загрузке лицензии определяйте по её статусу и журналу. Отдельно проверьте аппаратный идентификатор, подпись и срок действия.

Лицензия не принимается

Состояние лицензии содержит причину отказа:

Симптом	Возможная причина	Решение
Hardware ID mismatch	Лицензия создана для другого устройства	Передайте текущий аппаратный ID в техническую поддержку Amicon для перепривязки лицензии
"Signature verification failed"	Публичный ключ не установлен	Обратитесь в техническую поддержку Amicon для установки ключа
"License expired"	Срок действия истёк	Запросите новую лицензию в технической поддержке Amicon
"Invalid license format"	Повреждённый файл	Повторно получите и загрузите файл лицензии

Диагностика лицензии:

```
### Текущий статус
curl "https://<management-ip>/api/v1/license/status" \
-H "Authorization: Bearer $TOKEN"

### Аппаратный ID устройства
curl "https://<management-ip>/api/v1/license/hardware-id" \
-H "Authorization: Bearer $TOKEN"

### Лицензированные функции
curl "https://<management-ip>/api/v1/license/features" \
-H "Authorization: Bearer $TOKEN"
```

Демонстрационный период закончился

Демонстрационный режим действует 30 дней с момента первой загрузки. После окончания периода операции изменения через API блокируются с HTTP 403, а веб-

интерфейс доступен только для чтения. Загрузите действующую лицензию, чтобы возобновить изменение конфигурации.

Функция требует другой уровень лицензии

Лицензия ограничивает доступные функции по уровню:

Функция	Basic	Standard	Premium
TLS-инспекция	+	+	+
Максимальная пропускная способность	1 Гбит/с	10 Гбит/с	Без ограничений
Максимум цепочек сервисов	1	3	Без ограничений
Интеграция ICAP	—	+	+
HA active-standby	—	—	+
IPv6	—	+	+
TLS 1.3	—	+	+

Для получения недостающей функции обновите лицензию до соответствующего уровня через техническую поддержку Amicon.

Общие проблемы после применения конфигурации

При ошибке применения сохраните текст ответа и соответствующие события журнала. Затем исправьте подготовленную конфигурацию или верните её к действующему состоянию.

Применение конфигурации завершилось с ошибкой

Повторите запрос и сразу получите последние события категории `config`:

```
### Применение конфигурации через API
curl "https://<management-ip>/api/v1/config/apply" -X POST \
  -H "Authorization: Bearer $TOKEN"

### Просмотр журналов применения
curl "https://<management-ip>/api/v1/logs?category=config&limit=50" \
  -H "Authorization: Bearer $TOKEN"
```

Типичные ошибки:

- **Service 'ngfw' referenced in chain but not defined** — сервис не существует. Создайте сервис перед добавлением в цепочку.

- **VLAN 'dirty-ngfw' referenced but not defined** — VLAN не настроен. Добавьте VLAN в секцию `network.vlans`.
- **aRule 'xyz' has Tcl syntax error** — ошибка синтаксиса Tcl. Проверьте парные фигурные скобки и кавычки.

Откат конфигурации

Если после применения конфигурации система работает некорректно:

```
### Автоматический откат (если не подтверждена в течение 120 секунд)
### Конфигурация откатится автоматически

### Ручной откат
curl -X POST "https://<management-ip>/api/v1/config/revert" \
  -H "Authorization: Bearer $TOKEN"

### Подтверждение конфигурации (отмена таймера отката)
curl -X POST "https://<management-ip>/api/v1/config/confirm" \
  -H "Authorization: Bearer $TOKEN"
```

Диагностика прохождения трафика

Раздел задаёт последовательность проверки соединения от клиентского интерфейса до origin-сервера и обратно. Используйте следующие инструменты:

- Объяснение потока — моделирование правила, действия, цепочки и режима TLS;
- Захват трафика — PCAP на физических интерфейсах или транках, через которые проходит входной, сервисный и выходной трафик;
- экран **Трафик** (Мониторинг трафика) — фактические решения по каждому соединению с панелью фильтрации по протоколам и действиям;
- Логирование — журнал событий с идентификаторами `AMB-NNNN`.

Путь трафика через Амибиг

Проверяйте путь последовательно: вход, решение политики, сервисную цепочку и origin-соединение.

Схема взаимодействия

Источник	Связь	Назначение
Клиент	→	Входной интерфейс
Входной интерфейс	→	Обработка SSL/TLS
Обработка SSL/TLS	→	Политика
Политика	→	Inline-сервис

Источник	Связь	Назначение
Политика	→	ICAP-сервис
Inline-сервис	→	Origin-соединение
ICAP-сервис	→	Origin-соединение
Origin-соединение	→	Выходной интерфейс
Выходной интерфейс	→	Origin-сервер

На клиентском соединении Амибиг принимает трафик и определяет действие политики. При `intercept` устройство расшифровывает TLS-трафик, передаёт его в назначенную цепочку и устанавливает отдельное origin-соединение к серверу.

Шаг 1. Проверка поступления трафика

Убедитесь, что пакеты от клиента достигают Амибиг.

На экране **Диагностика** запустите захват на входном интерфейсе:

```
### Запуск захвата через API
curl -X POST "https://<management-ip>/api/v1/diagnostics/pcap" \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "interfaces": ["enp0s8"],
    "max_seconds": 30
  }'
```

Перед запуском запросите `GET /api/v1/diagnostics/interfaces`. Добавляйте BPF-фильтр только для интерфейса с `supports_bpf: true`: DPDK-захват отклоняет запросы с BPF.

Если пакеты не поступают, проверьте физическое подключение, коммутацию и маршрут клиента через VIP Амибиг.

Если входной интерфейс — LACP-транк, захват запускается на самом транке (bond-порту), а не на отдельных портах-участниках.

Шаг 2. Проверка обработки на входе

Убедитесь, что система перехватывает трафик. Проверьте журналы через API:

```
### Проверка системных журналов
curl "https://<management-ip>/api/v1/logs?category=system&limit=20" \
  -H "Authorization: Bearer $TOKEN"
```

Если трафик не обрабатывается, убедитесь, что конфигурация применена, а в секции `topology.listener` задан слушатель.

Шаг 3. Проверка решения политики

Политика назначает соединению действие `intercept`, `bypass`, `passthrough` или `reject`.

Через блок **Объяснение потока**

До захвата пакетов смоделируйте поток на экране **Диагностика**. Укажите адреса, порт и SNI, затем сравните результат для действующей (`running`) и подготовленной (`candidate`) конфигураций.

Через экран «Трафик»

На экране **Трафик** найдите соединение по адресу клиента. Столбцы **Действие**, **Цепочка** и **Правило** показывают фактическое решение.

```
### Поиск соединений конкретного клиента через API
curl "https://<management-ip>/api/v1/traffic?limit=100" \
-H "Authorization: Bearer $TOKEN"
```

Через журналы

Для поиска событий по адресу клиента:

```
### SSL-события в логах
curl "https://<management-ip>/api/v1/logs?category=security&search=10.1.1.50&limit=20" \
-H "Authorization: Bearer $TOKEN"
```

Порядок проверки правил

Policy engine проверяет правила в порядке, указанном в конфигурации. Первое совпавшее правило определяет действие. Проверьте:

1. Исключения расположены выше общих правил.
2. Условия `match` содержат правильные `source-ip`, `dest-ip`, `dest-port`, `sni`, `url-category` и `protocol`.
3. Правила включены (`enabled: true`).
4. Указана правильная сервисная цепочка (`service-chain`).

```
### Получить текущую политику
curl "https://<management-ip>/api/v1/config/policy" \
-H "Authorization: Bearer $TOKEN" | python3 -m json.tool
```

Шаг 4. Проверка сервисной цепочки

Проверьте каждый сервис в порядке, заданном цепочкой. Для inline- и ICAP-сервисов используются разные признаки доступности и разные точки захвата трафика.

Inline-сервис (NGFW)

Проверьте состояние участников пула по данным монитора доступности:

```
### Проверка мониторов доступности
curl "https://<management-ip>/api/v1/traffic/health" \
-H "Authorization: Bearer $TOKEN"
```

Найдите в `network.vlans` родительские интерфейсы входного (`dirty`) и выходного (`clear`) VLAN сервиса, затем запустите захват на этих физических портах или транках. API захвата не принимает имена VLAN. Если оба интерфейса используют один backend, их можно указать в одном запросе:

```
curl -X POST "https://<management-ip>/api/v1/diagnostics/pcap" \
-H "Authorization: Bearer $TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "interfaces": ["<service-ingress-port>", "<service-egress-port>"],
  "max_seconds": 30
}'
```

Если родительские интерфейсы используют разные backend, запустите отдельный захват для каждого. Не задавайте BPF-фильтр для DPDK. Если трафик поступает на входной порт сервиса, но не возвращается через выходной, проверьте политику NGFW, шлюз `from-service.gateway` и обратный маршрут.

ICAP-сервис (DLP)

Для ICAP проверьте монитор доступности и события обмена с сервером:

```
### Проверка мониторов доступности
curl "https://<management-ip>/api/v1/traffic/health" \
-H "Authorization: Bearer $TOKEN"

### Проверка журналов ICAP
curl "https://<management-ip>/api/v1/logs?category=traffic&search=icap&limit=20" \
-H "Authorization: Bearer $TOKEN"
```

Шаг 5. Проверка origin-соединения

После прохождения сервисной цепочки трафик шифруется и отправляется на сервер. Запустите захват на выходном интерфейсе:

```
curl -X POST "https://<management-ip>/api/v1/diagnostics/pcap" \
-H "Authorization: Bearer $TOKEN" \
-H "Content-Type: application/json" \
```

```
-d '{
  "interfaces": ["enp0s9"],
  "max_seconds": 30
}'
```

Комплексная диагностика соединения

Используйте полный сценарий, если отдельная проверка политики, маршрута или сервиса не выявила причину. Все шаги должны относиться к одному клиентскому соединению и одному интервалу времени.

Последовательность проверки

Для диагностики конкретного соединения (клиент 10.1.1.50 -> сервер 93.184.216.34:443):

1. Смоделируйте ожидаемое правило и действие в блоке `Объяснение потока`.
2. Запустите захват на входном интерфейсе. Фильтр `host 10.1.1.50 and tcp port 443` добавляйте только для интерфейса ядра Linux с поддержкой BPF.
3. Найдите соединение на экране `Трафик` и сравните действие с ожидаемым.
4. Проверьте монитор доступности участников сервисной цепочки.
5. Запустите захват на выходном интерфейсе. Фильтр `host 93.184.216.34` добавляйте только при поддержке BPF.
6. Проверьте экран `Просмотр логов` с фильтром по адресу клиента.

Использование PCAP-захвата для диагностики

Для комплексного анализа сначала сопоставьте VLAN каждого участка пути с его физическим портом или транком и запросите сведения об интерфейсах через `GET /api/v1/diagnostics/interfaces`. Сгруппируйте интерфейсы по `capture_backend`:

1. Несколько DPDK-портов можно указать в одном запросе без `bpf_filter`.
2. Интерфейсы ядра Linux укажите в отдельном запросе; для этой группы доступен BPF-фильтр.
3. Не смешивайте интерфейсы ядра Linux и DPDK в одном запросе.

Скачайте каждый PCAP и сопоставьте файлы по времени. Метаданные Амибиг покажут действие, цепочку и SNI для пакетов.

Диагностика aRules

Если aRules не срабатывают или дают неожиданный результат, проверьте события и ссылку на правило:

```
### Проверка журналов aRules
curl "https://<management-ip>/api/v1/logs?search=aRule&limit=20" \
```

```
-H "Authorization: Bearer $TOKEN"

### Проверка, что aRule привязан к сервису или правилу
curl -s "https://<management-ip>/api/v1/config/services" \
-H "Authorization: Bearer $TOKEN"
```

aRules срабатывают только для HTTP-трафика и только при обработке первого запроса в соединении. Последующие запросы по тому же keep-alive-соединению повторно не пропускаются через aRules.

Типичные сценарии

В этом разделе проверки сгруппированы по наблюдаемому симптому. Выполняйте их сверху вниз и фиксируйте первый этап, на котором фактическое поведение отличается от ожидаемого.

Трафик проходит без инспекции (всё bypass)

Проверьте действие политики и обработку QUIC:

1. Проверьте, что в policy есть правило с `action: intercept`.
2. Проверьте порядок: общее правило `bypass` выше `intercept` может исключить весь трафик из расшифрования.
3. Проверьте переключатель `Блокировка QUIC` на экране `Топология`. При `reject-quic: true` dataplane блокирует пакеты, распознанные как QUIC Initial, на любом UDP-порту.

Высокая задержка при инспекции

Разделите задержку Амибиг и внешних сервисов:

1. Проверьте мониторы доступности: недоступные участники могут перераспределить нагрузку на оставшийся сервис.
2. Измерьте задержку ICAP-сервера отдельно от задержки Амибиг.
3. Если политика безопасности допускает исключение, отключите передачу некритичного трафика в DLP отдельным правилом.

Соединения обрываются случайным образом

Сопоставьте момент разрыва с событиями и загрузкой устройства:

1. Проверьте ошибки на экране `Просмотр логов`.
2. Проверьте историю состояний участников пула.
3. Проверьте использование памяти и процессоров на экране `Обзор`.

Справочник

Справочник содержит структурированные сведения, которые нужны для проверки конфигурации и автоматизации: YANG-модель, JSON-схему, ключи командной строки, артефакты релиза, лимиты и константы.

Раздел рассчитан на администраторов, инженеров внедрения и техническую поддержку, когда требуется сверить точное имя поля, допустимое значение или состав поставочного артефакта.

Справочник конфигурационной схемы

Конфигурация Амибиг хранится в формате JSON и содержит следующие секции верхнего уровня.

```
{
  "system": { },
  "network": { },
  "ssl": { },
  "services": { },
  "service-chains": { },
  "policy": { },
  "egress": { },
  "protocol": { },
  "topology": { },
  "dataplane": { },
  "ha": { },
  "logging": { },
  "snmp": { },
  "observability": { },
  "users": { },
  "rate-shaping": { },
  "arules": { },
  "url-categories": { }
}
```

Параметры обработки трафика, ранее вложенные в `dataplane.dpdn`, вынесены в отдельные секции верхнего уровня: `egress` (трансляция адреса источника на пути к серверу) и `protocol` (TCP/MSS). Конфигурации и скрипты, использующие старые пути, необходимо обновить.

system

Глобальные системные настройки: имя узла, часовой пояс, адрес управления, DNS и NTP.

```
{
  "system": {
```



```
    "hostname": "amibig-01",
    "timezone": "UTC",
    "management": {
      "address": "10.0.0.10/24",
      "gateway": "10.0.0.1"
    },
    "dns": {
      "nameservers": ["8.8.8.8", "8.8.4.4"]
    },
    "ntp": {
      "servers": ["pool.ntp.org"]
    }
  }
}
```

Поле	Тип	Описание
hostname	string	Имя хоста системы
timezone	string	Временная зона (по умолчанию UTC)
management.address	ip-prefix	IP-адрес управления с длиной префикса
management.gateway	ip-address	Шлюз по умолчанию для интерфейса управления
dns.nameservers	list[ip-address]	Список DNS-серверов
ntp.servers	list[string]	Список NTP-серверов

observability

Настройки пороговых оповещений, хранения аналитики, передачи журнала соединений во внешнюю SIEM и диагностического архива.

```
{
  "observability": {
    "profile": "balanced",
    "alerts": {
      "enabled": true,
      "default-routes": ["local-log", "snmp", "syslog"],
      "rule": []
    },
    "history": {
      "enabled": true,
      "sample-period-seconds": 15,
      "raw-retention-minutes": 60,
      "rollup": [
        { "window": "5m", "bucket-seconds": 15, "retention-hours": 24 },
        { "window": "1h", "bucket-seconds": 60, "retention-hours": 72 },
        { "window": "24h", "bucket-seconds": 300, "retention-hours": 168 },
        { "window": "7d", "bucket-seconds": 3600, "retention-hours": 720 },
        { "window": "30d", "bucket-seconds": 14400, "retention-hours": 2160 }
      ]
    },
    "retention": {
      "events-days": 90,
    }
  }
}
```

```

        "alert-history-days": 90,
        "alert-history-max-rows": 20000
    },
    "flow-export": {
        "enabled": false,
        "mode": "rfc5424",
        "receivers": [
            {
                "name": "siem-1",
                "host": "10.0.0.50",
                "port": 514,
                "transport": "udp",
                "format": "rfc5424",
                "enabled": true
            }
        ],
        "sampling": {
            "mode": "all",
            "rate": 1,
            "adaptive-threshold-cps": 0
        }
    },
    "support-bundle": {
        "enabled": true,
        "manifest-version": 1,
        "history-window": "1h",
        "include-telemetry": true,
        "include-alert-summary": true,
        "include-certificates": true
    }
}

```

Поле	Тип	Описание
<code>profile</code>	enum	Профиль настроек: <code>minimal_telemetry</code> , <code>balanced</code> (по умолчанию), <code>full_diagnostics</code> ; явно заданные вложенные значения имеют приоритет над профилем
<code>alerts.default-routes</code>	list[enum]	Каналы доставки оповещений: <code>local-log</code> , <code>snmp</code> , <code>syslog</code>
<code>alerts.rule[]</code>	list[object]	Правила типов <code>throughput</code> , <code>connection-rate</code> , <code>member-flaps</code> , <code>ha-health</code> , <code>disk-pressure</code> , <code>certificate-expiry</code> , <code>cpu</code> , <code>memory</code> , <code>port-exhaustion</code>
<code>history.sample-period-seconds</code>	integer	Период сбора сырых выборок телеметрии
<code>history.raw-retention-minutes</code>	integer	Время хранения сырых выборок

Поле	Тип	Описание
<code>history.rollup[]</code>	<code>list[object]</code>	Окна агрегации <code>5m</code> , <code>1h</code> , <code>24h</code> , <code>7d</code> , <code>30d</code>
<code>retention.events-days</code>	<code>integer</code>	Срок хранения журнала событий (<code>events.db</code>), по умолчанию <code>90</code> дней
<code>retention.alert-history-days</code>	<code>integer</code>	Срок хранения истории оповещений, по умолчанию <code>90</code> дней
<code>retention.alert-history-max-rows</code>	<code>integer</code>	Потолок числа записей истории оповещений, по умолчанию <code>20000</code>
<code>flow-export.mode</code>	<code>enum</code>	Формат передачи журнала соединений: <code>sflow</code> , <code>cef</code> , <code>rfc5424</code> , <code>json-lines</code> , <code>ipfix</code>
<code>flow-export.receivers[]</code>	<code>list[object]</code>	SIEM-приёмники: <code>name</code> , <code>host</code> , <code>port</code> , <code>transport</code> (<code>udp</code> / <code>tcp</code> / <code>tls</code>), <code>format</code> (<code>cef</code> / <code>rfc5424</code> / <code>json-lines</code> / <code>ipfix</code>), <code>enabled</code> ; для <code>tls</code> обязательны <code>ca-trust</code> и <code>server-host</code>
<code>flow-export.sampling.mode</code>	<code>enum</code>	Выборка записей: <code>all</code> (каждая запись), <code>1-in-n</code> , <code>aggregates</code>
<code>flow-export.sampling.rate</code>	<code>integer</code>	Делитель N для режима <code>1-in-n</code>
<code>flow-export.sampling.adaptive-threshold-cps</code>	<code>integer</code>	Порог CPS автоматического перехода <code>all</code> → <code>1:N</code> (<code>0</code> = без перехода)
<code>support-bundle.manifest-version</code>	<code>integer</code>	Версия машинно-читаемого манифеста внутри диагностического архива
<code>support-bundle.history-window</code>	<code>enum</code>	Окно аналитики, включаемое в диагностический архив
<code>support-bundle.include-telemetry</code>	<code>boolean</code>	Включать снимок телеметрии в архив
<code>support-bundle.include-alert-summary</code>	<code>boolean</code>	Включать сводку и активные оповещения в архив

Поле	Тип	Описание
<code>support-bundle.include-certificates</code>	boolean	Включать состояние сертификатов в архив

`minimal_telemetry` снижает накладные расходы и оставляет только лёгкую локальную телеметрию. `balanced` — рекомендуемый профиль по умолчанию. `full_diagnostics` делает набор данных наблюдаемости максимально полным и предназначен для отладочных режимов с меньшим запасом производительности.

Передача журнала соединений активна только при `enabled: true` и хотя бы одним включённым приёмнике. Пустой список приёмников отключает передачу. Режим `sflow` использует прежние поля `collector-host`, `collector-port`, `agent-address`, `sampling-rate`, `counter-poll-seconds`, `export-service-metadata`.

network

Сетевая конфигурация: интерфейсы управления и dataplane, агрегированные каналы LACP, VLAN, Self-IP и статические маршруты.

interfaces

Список описывает физические интерфейсы и их назначение:

```
{
  "interfaces": {
    "interface": [
      {
        "name": "enp0s8",
        "description": "Client-facing interface",
        "plane": "data",
        "binding": "dpdk",
        "speed": "10G",
        "mtu": 1400,
        "enabled": true
      }
    ]
  }
}
```

Поле	Тип	Описание
<code>name</code>	string	Имя интерфейса (например, <code>enp0s8</code>)
<code>description</code>	string	Описание
<code>plane</code>	enum	Назначение порта: <code>management</code> , <code>data</code> , <code>ha</code> (по умолчанию <code>data</code>)

Поле	Тип	Описание
<code>binding</code>	enum	<code>kernel</code> (порт остаётся в ОС) или <code>dppk</code> (порт привязан к dataplane)
<code>speed</code>	enum	Скорость: <code>1G</code> , <code>10G</code> , <code>25G</code> , <code>40G</code> , <code>100G</code>
<code>mtu</code>	uint16 (576–1500)	MTU DPDK-порта (только уменьшение; без значения порт остаётся с MTU 1500). Для агрегированного канала действует минимум по портам-участникам
<code>enabled</code>	boolean	Включён (по умолчанию <code>true</code>)

trunks

Агрегация интерфейсов (LACP 802.3ad).

```
{
  "trunks": {
    "trunk": [
      {
        "name": "bond0",
        "interfaces": ["enp0s8", "enp0s9"],
        "lacp": "enabled",
        "lacp-mode": "active",
        "lacp-timeout": "short",
        "distribution-hash": "src-dst-ipport"
      }
    ]
  }
}
```

Поле	Тип	По умолчанию	Описание
<code>name</code>	string	—	Имя агрегированного канала
<code>interfaces</code>	list[string]	—	Физические интерфейсы-участники
<code>lacp</code>	enum	<code>enabled</code>	LACP: <code>enabled</code> / <code>disabled</code>
<code>lacp-mode</code>	enum	<code>active</code>	Режим: <code>active</code> / <code>passive</code>

Поле	Тип	По умолчанию	Описание
lasp-timeout	enum	short	Таймаут: short (3 сек) / long (90 сек)
distribution-hash	enum	src-dst-ipport	Алгоритм хеширования: dst-mac, src-dst-ipport, src-dst-mac

vlan

Каждый VLAN привязывается к физическому или агрегированному интерфейсу:

```
{
  "vlans": {
    "vlan": [
      {
        "name": "dirty-ngfw",
        "interface": "enp0s9",
        "tag": 100,
        "description": "NGFW dirty side"
      }
    ]
  }
}
```

Поле	Тип	Описание
name	string	Уникальное имя VLAN
interface	string	Родительский интерфейс или транк
tag	uint16 (1-4094)	Тег VLAN 802.1Q; без значения трафик передаётся без тега
description	string	Описание

self-ips

Self-IP задаёт собственный адрес Амибиг в выбранном VLAN:

```
{
  "self-ips": {
    "self-ip": [
      {
        "name": "client-ip",
        "address": "10.1.0.1/24",
        "vlan": "client-vlan"
      }
    ]
  }
}
```

```
}
}
```

routes

Статический маршрут содержит сеть назначения и адрес шлюза:

```
{
  "routes": {
    "route": [
      {
        "name": "default",
        "network": "0.0.0.0/0",
        "gateway": "10.0.0.1"
      }
    ]
  }
}
```

ssl

Сертификаты, правила шифрования, профили клиентской и серверной стороны SSL/TLS, а также параметры ГОСТ-TLS.

Параметры ГОСТ-TLS

В секции `ssl` находятся следующие параметры перехвата ГОСТ-TLS:

Поле	Тип	По умолчанию	Описание
<code>gost-enabled</code>	boolean	<code>false</code>	Включить поддержку ГОСТ-TLS (ГОСТ Р 34.10-2012, 34.11-2012)
<code>gost-ca</code>	cert-ref	—	Ссылка на сертификат ГОСТ-УЦ для TLS-инспекции; обязательна при <code>gost-enabled: true</code> , закрытый ключ должен соответствовать ГОСТ Р 34.10-2012

Поле	Тип	По умолчанию	Описание
<code>gost-default-ciphers</code>	string	<code>GOST2012-KUZNYECHIK-...</code>	Политика ГОСТ-шифров TLS 1.2; по умолчанию семейство ГОСТ Р 34.12-2015 («Кузнечик»/«Мagma»)
<code>gost-leaf-paramset</code>	string	<code>A</code>	Набор параметров кривой ГОСТ Р 34.10-2012 для создаваемых ключей подменных сертификатов
<code>gost-allow-legacy</code>	boolean	<code>false</code>	Разрешить устаревшие шифры ГОСТ Р 34.10-2001 / 28147-89 (по умолчанию запрещены для защиты от понижения стойкости)

certificates

Список содержит сертификаты, закрытые ключи и наборы доверенных УЦ:

```
{
  "certificates": {
    "certificate": [
      {
        "name": "mitm-ca",
        "type": "ca",
        "cert-path": "/config/ssl/ca.crt",
        "key-path": "/config/ssl/ca.key"
      },
      {
        "name": "trusted-bundle",
        "type": "bundle",
        "cert-path": "/config/ssl/ca-bundle.crt"
      }
    ]
  }
}
```

Поле	Тип	Описание
<code>name</code>	string	Уникальное имя сертификата

Поле	Тип	Описание
<code>type</code>	enum	<code>ca</code> (подписывающий УЦ TLS-инспекции), <code>server</code> (серверный сертификат), <code>bundle</code> (набор доверенных УЦ)
<code>cert-path</code>	string	Путь к PEM-файлу сертификата
<code>key-path</code>	string	Путь к закрытому ключу (для <code>ca</code> и <code>server</code>)

trusted-store

Автоматическая сборка набора доверенных УЦ для проверки серверных сертификатов. Само присутствие контейнера `trusted-store` в конфигурации включает режим сборки (presence-контейнер): при применении конфигурации базовый набор УЦ из состава ПО объединяется с пользовательскими сертификатами типа `ca` и `bundle` (кроме записей с назначенной ролью — подписывающий УЦ TLS-инспекции, ГОСТ-УЦ и сама цель сборки), из результата вычитаются исключения, и итоговый файл записывается по пути `cert-path` сертификата, на который ссылается `server-profile.trusted-ca`. Без контейнера администратор управляет этим файлом вручную.

```
{
  "trusted-store": {
    "exclusions": {
      "exclusion": [
        {
          "fingerprint": "aa11...ff99",
          "description": "Отозванный публичный УЦ"
        }
      ]
    }
  }
}
```

Поле	Тип	Описание
<code>exclusions.exclusion[]</code>	list	Список УЦ, вычитаемых из итогового набора после объединения
<code>exclusions.exclusion[].fingerprint</code>	string	Отпечаток сертификата SHA-256 в шестнадцатеричном виде (ключ списка; двоеточия и регистр нормализуются)
<code>exclusions.exclusion[].description</code>	string	Описание исключения

cipher-rules

Правило задаёт версии TLS и наборы шифров для профилей:

```
{
  "cipher-rules": {
    "rule": [
      {
        "name": "default-cipher",
        "ciphers": "ECDHE+AESGCM:ECDHE+CHACHA20",
        "min-version": "TLS1.2",
        "max-version": "TLS1.3"
      }
    ]
  }
}
```

Поле	Тип	Описание
ciphers	string	Строка наборов шифров OpenSSL для TLS 1.2 и ниже
ciphersuites	string	Строка шифронаборов TLS 1.3 в формате OpenSSL; пусто — умолчание OpenSSL. Действует только когда max-version допускает TLS1.3
min-version	enum	Минимальная версия TLS: от TLS1.0 до TLS1.3
max-version	enum	Максимальная версия TLS: от TLS1.0 до TLS1.3

client-profile

Профиль клиентского соединения управляет созданием подменных сертификатов и параметрами TLS между клиентом и Амибиг.

Поле	Тип	По умолчанию	Описание
ca-certificate	cert-ref	—	УЦ для подписи подменных сертификатов
cipher-rule	cipher-rule-ref	—	Правило шифрования
allow-non-ssl	boolean	true	Пропускать не-SSL трафик
forward-proxy-bypass	boolean	true	Разрешить bypass по решению policy

Поле	Тип	По умолчанию	Описание
<code>alpn.mode</code>	enum	<code>negotiate</code>	ALPN: <code>negotiate</code> , <code>downgrade-h1</code> , <code>disable</code>
<code>cert-lifespan</code>	uint16	<code>30</code>	Срок действия подменных сертификатов, 1–825 дней; <code>notBefore</code> устанавливается на один день раньше времени создания
<code>close-notify</code>	enum	<code>propagate</code>	Корректное завершение TLS на перехваченных соединениях: <code>propagate</code> — передавать признак конца потока (<code>close_notify</code> от origin-сервера транслируется клиенту, «голый» FIN остаётся «голым» FIN), <code>never</code> — всегда завершать без <code>close_notify</code>
<code>handshake-timeout</code>	uint16	<code>10</code>	Лимит фазы установления клиентского соединения (ожидание ClientHello и TLS handshake), секунды, диапазон 0–300; <code>0</code> — без ограничения. Отсчитывается от входа в фазу и не продлевается трафиком

Поле	Тип	По умолчанию	Описание
<code>cert-extensions</code>	<code>list[enum]</code>	—	Расширения для копирования: <code>basic-constraints</code> , <code>subject-alternative-name</code> , <code>extended-key-usage</code>
<code>forge-cache-enabled</code>	<code>boolean</code>	<code>true</code>	Кэшировать подменные сертификаты для повторных TLS handshake
<code>forge-cache-max-entries</code>	<code>integer</code>	<code>4096</code>	Максимальное число подменных сертификатов в кэше
<code>ctx-cache-enabled</code>	<code>boolean</code>	<code>true</code>	Кэшировать клиентские контексты <code>SSL_CTX</code> отдельно для каждого SNI
<code>ctx-cache-max-entries</code>	<code>integer</code>	<code>4096</code>	Максимальное число контекстов <code>SSL_CTX</code> в кэше
<code>session-cache-enabled</code>	<code>boolean</code>	<code>true</code>	Разрешить возобновление TLS-сессий на клиентском соединении
<code>session-cache-timeout-seconds</code>	<code>integer</code>	<code>300</code>	Срок хранения клиентской TLS-сессии в кэше, секунды
<code>session-ticket-mode</code>	<code>enum</code>	<code>local</code>	Режим ключа Session Ticket (STEK): <code>ha-sync</code> , <code>local</code> , <code>disabled</code>
<code>mitm-crypto-profile</code>	<code>enum</code>	<code>balanced</code>	Криптографический профиль TLS-инспекции: <code>compatibility</code> , <code>balanced</code> , <code>throughput</code>

server-profile

Профиль origin-соединения управляет TLS между Амибиг и сервером.

Поле	Тип	По умолчанию	Описание
<code>trusted-ca</code>	cert-ref	—	Набор доверенных УЦ для проверки сертификата сервера
<code>cipher-rule</code>	cipher-rule-ref	—	Правило шифрования
<code>verify-peer</code>	boolean	<code>true</code>	Требовать валидный сертификат сервера
<code>on-expired-cert</code>	enum	<code>drop</code>	Действие при истёкшем сертификате: <code>drop</code> , <code>ignore</code> , <code>log-only</code>
<code>on-untrusted-cert</code>	enum	<code>drop</code>	Действие при недоверенном сертификате
<code>on-revoked-cert</code>	enum	<code>ignore</code>	Действие при отозванном сертификате
<code>handshake-timeout</code>	uint16	<code>10</code>	Лимит фазы установления origin-соединения (подключение к серверу и TLS handshake), секунды, диапазон 0–300; 0 — без ограничения. Отдельный от клиентского лимит: молчащий origin-сервер и медленный клиент — разные отказы
<code>session-resumption-enabled</code>	boolean	<code>true</code>	Разрешить возобновление TLS-сессий на origin-соединении
<code>session-cache-timeout-seconds</code>	integer	<code>300</code>	Срок хранения origin-сессии в кэше, секунды

Поле	Тип	По умолчанию	Описание
handshake-backpressure-limit	integer	0	Максимальное число одновременно выполняемых TLS handshake на одном рабочем потоке; 0 снимает ограничение
handshake-overflow-action	enum	defer	Действие при достижении лимита: drop отклоняет новое соединение, defer и log-only продолжают обработку без постановки в очередь
acceleration-mode	enum	cpu	Режим ускорения: cpu, auto, qat; при недоступности QAT обработка продолжается на процессоре

services

Определение сервисов инспекции: inline (NGFW) и ICAP (DLP).

```
{
  "services": {
    "service": [
      {
        "name": "ngfw",
        "type": "inline",
        "enabled": true,
        "service-down-action": "bypass",
        "arules": [],
        "inline": {
          "to-service": { "vlan": "dirty-ngfw", "self-ip": "10.2.1.1", "gateway": "10.2.1.10" },
          "from-service": { "vlan": "clear-ngfw", "self-ip": "10.2.2.1", "gateway": "10.2.2.10" },
          "protocols": ["tcp"],
          "traverse-probe": { "enabled": true }
        },
        "pool": {
          "member": [
            { "address": "10.2.1.10", "port": 0, "enabled": true }
          ],
          "load-balancing": "session-hash",
          "monitor": {
            "type": "tcp",
            "interval": 5,

```

```

        "timeout": 16,
        "fail-threshold": 3,
        "recover-threshold": 2
    }
}
]
}
}

```

Поле	Тип	Описание
<code>type</code>	enum	<code>inline</code> (устройство L2/L3) или <code>icap</code> (ICAP-сервер)
<code>service-down-action</code>	enum	Действие при недоступности: <code>bypass</code> , <code>drop</code> , <code>reset</code>
<code>arules</code>	list[string]	Привязанные aRules (по имени)
<code>pool.load-balancing</code>	enum	Балансировка: <code>round-robin</code> , <code>session-hash</code> , <code>predictive</code> , <code>least-connections</code>
<code>pool.monitor.type</code>	enum	Тип проверки: <code>tcp</code> , <code>icmp</code> , <code>http-get</code>
<code>inline.to-service.self-ip</code>	ip-address	Собственный адрес устройства на сегменте к сервису; адрес, с которого уходит сквозная проба. Пер-узловое поле: при синхронизации НА не переносится
<code>inline.from-service.self-ip</code>	ip-address	Собственный адрес устройства на сегменте от сервиса; на нём слушает встроенный ответчик сквозной пробы. Пер-узловое поле: при синхронизации НА не переносится

inline.traverse-probe

Сквозная проба inline-сервиса: небольшой HTTP-запрос, отправляемый **через** сервис и обратно на устройство. Обычная проверка участника пула адресована самому устройству безопасности и доставляется в пределах подсети сегмента, поэтому может показывать сервис доступным, когда через цепочку не проходит ни одно соединение. Сквозная проба открывает реальное TCP-соединение к

адресу за сервисом тем же маршрутом, что и рабочий трафик. По умолчанию (режим петли) целью служит собственный `from-service.self-ip` устройства: запрос уходит через сегмент к сервису, пересекает сервис и обслуживается встроенным ответчиком на сегменте от сервиса — путь проверяется целиком без внешних зависимостей.

Поле	Тип	По умолчанию	Описание
<code>enabled</code>	boolean	<code>false</code>	Включить сквозную пробу
<code>target</code>	ipv4-address	—	Переопределение цели пробы; пусто — режим петли (цель — <code>from-service.self-ip</code>). Явная цель не должна попадать в подсеть сегмента к сервису, иначе проба пойдёт в обход сервиса
<code>port</code>	port-number	<code>7778</code>	TCP-порт встроенного ответчика и порт назначения пробы
<code>path</code>	string	<code>/</code>	Путь HTTP-запроса пробы
<code>host</code>	string	—	Заголовок Host; пусто — используется адрес цели
<code>receive</code>	string	—	Ожидаемая подстрока в ответе; пусто — достаточно статуса 2xx. В режиме петли сверяется маркер встроенного ответчика

service-chains

Упорядоченные последовательности сервисов.

```
{
  "service-chains": {
    "chain": [
```



```

    {
      "name": "ngfw-dlp-chain",
      "description": "NGFW + DLP inspection",
      "services": ["ngfw", "dlp"]
    }
  ]
}

```

Поле	Тип	Описание
name	string	Уникальное имя цепочки
services	list[service-ref]	Упорядоченный список сервисов

policy

Правила классификации трафика и назначения сервисных цепочек. Правила проверяются в указанном порядке, первое совпадение определяет действие.

```

{
  "policy": {
    "name": "main-policy",
    "reject-quic": true,
    "udp-chain": true,
    "icmp-chain": true,
    "default-rule": {
      "name": "default-intercept",
      "action": "intercept",
      "service-chain": "ngfw-dlp-chain"
    },
    "rule": [
      {
        "name": "bypass-internal",
        "enabled": true,
        "match": {
          "source-ip": ["10.0.0.0/8"],
          "dest-ip": ["10.0.0.0/8"]
        },
        "action": "passthrough"
      }
    ]
  }
}

```

Параметры уровня политики

Параметры определяют обработку QUIC, UDP и ICMP до проверки отдельных правил:

Поле	Тип	По умолчанию	Описание
<code>reject-quic</code>	boolean	<code>true</code>	Блокировать дейтаграммы, распознанные как QUIC Initial, на любом UDP-порту. Остальной трафик UDP/443 не блокируется этим параметром
<code>udp-chain</code>	boolean	<code>true</code>	Пропускать транзитный UDP через inline-цепочку сервисов; <code>false</code> — направлять UDP к origin-серверу в обход цепочки. Трансляция адреса источника действует в обоих случаях
<code>icmp-chain</code>	boolean	<code>true</code>	Пропускать транзитный ICMP через inline-цепочку сервисов; <code>false</code> — воспроизводит поведение F5 (маршрутизация в обход цепочки), служебный трафик (определение MTU пути, трассировка) не зависит от доступности цепочки

В веб-интерфейсе эти переключатели находятся на экране `Топология`.

Условия match

Условия одного правила объединяются логическим И, а значения внутри списка — логическим ИЛИ:

Поле	Тип	Описание
<code>source-ip</code>	<code>list[ip-prefix]</code>	IP/подсети клиента
<code>dest-ip</code>	<code>list[ip-prefix]</code>	IP/подсети назначения
<code>dest-port</code>	<code>list[port-number]</code>	Порты назначения

Поле	Тип	Описание
<code>protocol</code>	enum	<code>tcp</code> , <code>udp</code> , <code>icmp</code> , <code>other</code> , <code>any</code>
<code>sni</code>	list[string]	TLS SNI; поддерживаются шаблоны вида <code>*.example.com</code>
<code>url-category</code>	list[string]	URL-категории
<code>source-user</code>	list[string]	Точное имя аутентифицированного пользователя
<code>source-user-group</code>	list[string]	Членство аутентифицированного пользователя в LDAP-группе
<code>source-user-class</code>	enum	Наличие идентификации у клиентского IP: <code>known-user</code> — соответствие «адрес — пользователь» есть, <code>unknown</code> — нет

`default-rule`

Явное правило по умолчанию применяется после всех записей в `policy.rule`.

Поле	Тип	Описание
<code>name</code>	string	Имя правила по умолчанию
<code>action</code>	enum	<code>intercept</code> , <code>bypass</code> , <code>passthrough</code> , <code>reject</code>
<code>service-chain</code>	chain-ref	Цепочка сервисов для правила по умолчанию
<code>rate-class</code>	rate-class-ref	Ссылка на класс управления трафиком для правила по умолчанию; ограничение полосы в dataplane текущей версии не применяется

Действия

Действие первого совпавшего правила определяет дальнейший путь соединения:

Действие	Описание
<code>intercept</code>	Расшифрование TLS-трафика и передача в сервисную цепочку
<code>bypass</code>	Пропуск без расшифрования через прозрачную inline-сервисную цепочку
<code>passthrough</code>	Прямой пропуск без расшифрования и без сервисной цепочки
<code>reject</code>	Отклонение соединения (RST/DROP)

Переопределение SNAT на правило

Каждое правило может переопределить глобальные настройки секции `egress` для потоков, которые оно классифицирует:

Поле	Тип	Описание
<code>origin-snat</code>	enum	<code>automap</code> , <code>none</code> , <code>pool</code> — переопределяет <code>egress.origin-snat</code> ; без значения наследуется глобальная настройка
<code>origin-source-port</code>	enum	<code>preserve</code> , <code>preserve-strict</code> , <code>change</code> — переопределяет <code>egress.origin-source-port</code> ; без значения наследуется глобальная настройка

Идентификационные заголовки на правило

Правило может добавлять в запросы, идущие через сервисную цепочку, заголовки с данными об аутентифицированном пользователе:

Поле	Тип	По умолчанию	Описание
<code>insert-identity-headers</code>	boolean	<code>false</code>	Вставлять заголовки <code>X-Authenticated-User</code> / <code>X-Client-IP</code> в запросы, направляемые через цепочку. Независимо от значения, эти заголовки всегда удаляются на выходе из цепочки и из входящих клиентских запросов
<code>identity-header-format</code>	enum	<code>username</code>	Формат значения: <code>username</code> — имя пользователя, <code>ldap-uri</code> — формат <code>LDAP://домен/пользователь</code>
<code>identity-header-encoding</code>	enum	<code>plain</code>	Кодирование значения: <code>plain</code> или <code>base64</code> (требование Symantec DLP)

Dataplane текущей версии поддерживает следующие сочетания:

- `tcp` поддерживает действие `intercept` с расшифрованием TLS-трафика и сервисной цепочкой;
- `udp` и `icmp` поддерживают `bypass`, `passthrough` и `reject`;
- `other` поддерживает только `passthrough` и `reject`.

egress

Политика трансляции адреса источника на пути от Амибиг к серверу (аналог F5 SSLO Egress Setting). Секция верхнего уровня; переопределение на правило политики — через поля `origin-snat` и `origin-source-port` в `policy.rule`.

```
{
  "egress": {
    "origin-snat": "automap",
    "origin-source-port": "preserve",
    "snat-pool": [
      { "address": "192.168.70.11" },
      { "address": "192.168.70.12" }
    ]
  }
}
```

```
    ]  
  }  
}
```

Поле	Тип	По умолчанию	Описание
origin-snat	enum	automap	Режим адреса источника в origin-соединении: automap (Self-IP исходящего VLAN), none (сохранить адрес клиента), pool (адрес из snat-pool)
origin-source-port	enum	preserve	Режим порта источника: preserve (сохранить порт клиента, при конфликте выбрать эфемерный), preserve-strict (требовать порт клиента, при конфликте отклонить соединение), change (всегда выбирать эфемерный порт)
snat-pool[]	list[object]	—	Адреса трансляции для режима pool (ключ address)

Подробное описание режимов и диагностики — в разделе Трансляция адреса источника (SNAT).

protocol

Политика TCP/MSS для инспектируемых и пропускаемых соединений (аналог профиля TCP F5). MTU канального уровня задаётся для порта в network.interfaces.interface.mtu .

```
{  
  "protocol": {  
    "tcp-mss-clamp": 1360,  
    "tcp-mss-clamp-client": 0,  
    "tcp-mss-clamp-origin": 0,  
    "proxy-mss": true,  
    "pmtud-blackhole-detect": true
```

```

    }
}

```

Поле	Тип	По умолчанию	Описание
<code>tcp-mss-clamp</code>	uint16 (0 или 536–1460)	1360	Общее верхнее значение TCP MSS для клиентского и origin-соединений; 0 отключает ограничение
<code>tcp-mss-clamp-client</code>	uint16 (0 или 536–1460)	0	Верхнее значение MSS для клиентского соединения (SYN-ACK); 0 отключает отдельное значение, но не общее ограничение
<code>tcp-mss-clamp-origin</code>	uint16 (0 или 536–1460)	0	Верхнее значение MSS для origin-соединения (исходящий SYN); 0 отключает отдельное значение
<code>proxy-mss</code>	boolean	true	Объявлять серверу MSS, согласованный на клиентском соединении
<code>pmtud-blackhole-detect</code>	boolean	true	Обнаружение «чёрных дыр» PMTUD: при повторных таймаутах ретрансмиссии эффективный MSS временно снижается до запасного значения и затем восстанавливается

dataplane

Параметры DPDK (`dataplane.dpdk`): путь к сгенерированной конфигурации `dataplane` (`config-path`), объём hugepages (`hugepages-mb`), маска ядер (`lcore-mask`) и

другие настройки запуска. Параметры обработки трафика находятся в секциях `egress` и `protocol`.

topology

Точка входа трафика: режим работы и параметры перехвата.

```
{
  "topology": {
    "name": "main",
    "mode": "transparent-l3",
    "policy-ref": "main-policy",
    "listener": {
      "vlan": "client-vlan",
      "address": "0.0.0.0",
      "port": 0,
      "protocol": "any"
    }
  }
}
```

Поле	Тип	Описание
<code>mode</code>	enum	<code>transparent-l2</code> или <code>transparent-l3</code>
<code>listener.vlan</code>	vlan-ref	VLAN для перехвата
<code>listener.address</code>	ip-address	Адрес прослушивания (<code>0.0.0.0</code> = все)
<code>listener.port</code>	port-number	Порт (<code>0</code> = все)
<code>client-interface</code>	string	Физический интерфейс для перехвата трафика

ha

Пара высокой доступности Active/Standby.

```
{
  "ha": {
    "enabled": true,
    "mode": "active-standby",
    "local": { "node-id": 1, "priority": 100 },
    "peer": { "address": "10.10.0.2" },
    "ha-link": {
      "interface": "eth5",
      "address": "10.10.0.1/30",
      "peer-address": "10.10.0.2"
    },
    "failover": {
      "heartbeat-interval": 1,
      "dead-interval": 3
    }
  }
}
```



```

    },
    "tls": { "enabled": false }
  }
}

```

Поле	Тип	Описание
<code>mode</code>	enum	<code>active-standby</code> ; другие режимы в текущей версии не поддерживаются
<code>local.node-id</code>	uint8	ID узла
<code>local.priority</code>	uint8	Приоритет VRRP (выше = предпочтительнее)
<code>peer.address</code>	ip-address	IP-адрес HA-партнёра
<code>failover.heartbeat-interval</code>	uint8	Интервал heartbeat-сообщений, секунды
<code>failover.dead-interval</code>	uint8	Число пропущенных heartbeat-сообщений до аварийного переключения

Поля состояния, доступные только для чтения: `role` , `vrrp-state` , `peer-state` , `tls-active` , `messages-sent` , `messages-received` .

logging

Журналирование: системный лог (`system`), аудит (`audit`), SSL-события (`ssl-events`) и экспорт (`export`). Экспорт поддерживает список syslog-приёмников (форматы `rfc3164` , `rfc5424` , `cef` , `leef`) и Kafka (форматы `json` , `cef` , `leef`), у каждого приёмника — фильтр по категориям, уровням важности и кодам сообщений. Подробнее в разделе Журнал событий и его экспорт.

snmp

SNMP-мониторинг.

```

{
  "snmp": {
    "enabled": false,
    "communities": [
      { "name": "public", "access": "ro", "source": "0.0.0.0/0" }
    ],
    "v3-users": [],
    "trap-targets": []
  }
}

```

Канонические ключи секции: `communities`, `v3-users`, `trap-targets`.

users

Управление пользователями и ролями RBAC.

```
{
  "users": {
    "user": [
      {
        "name": "admin",
        "role": "admin",
        "description": "System administrator",
        "shell": "bash"
      }
    ]
  }
}
```

Роль	Описание
admin	Полный доступ: чтение, запись, управление пользователями
operator	Чтение и изменение конфигурации, без управления пользователями
viewer	Только чтение

rate-shaping

Секция хранит классы управления трафиком и проверяет ссылки на них из правил политики. **Dataplane текущей версии не ограничивает пропускную способность по этим значениям.** Не используйте секцию как средство обеспечения QoS или гарантированной полосы.

```
{
  "rate-shaping": {
    "class": [
      {
        "name": "low-priority",
        "rate": "10mbps",
        "ceiling": "50mbps",
        "max-burst": "2mb",
        "direction": "both",
        "drop-policy": "tail",
        "queue": "sfq"
      }
    ]
  }
}
```

Поле	Тип	Описание
rate	string	Настроенное значение CIR, например 10mbps , 1gbps , 500kbps ; в текущей версии не применяется
ceiling	string	Настроенное значение PIR; в текущей версии не применяется
max-burst	string	Настроенный размер всплеска, например 2mb или 64kb ; в текущей версии не применяется
direction	enum	to-client , to-server , both
drop-policy	enum	tail или red ; в текущей версии не применяется
queue	enum	pfifo или sfq ; в текущей версии не применяется

Класс указывается в правиле политики через поле `rate-class` . Ссылка проверяется при валидации конфигурации, но не изменяет обработку трафика.

arules

Tcl-совместимые правила обработки HTTP-трафика (совместимость с F5 iRules). Подробнее в разделе aRules.

```
{
  "arules": {
    "arule": [
      {
        "name": "size_limit",
        "description": "Bypass DLP for small PUT/POST",
        "enabled": true,
        "code": "when RULE_INIT {\n    set ::max_payload_size 1000\n}\n\nwhen HTTP_REQUEST {\n    if { [HTTP::method] eq \"PUT\" || [HTTP::method] eq \"POST\" } {\n        if { [HTTP::header exists Content-Length] } {\n            set clen [HTTP::header value Content-Length]\n            if { $clen <= $::max_payload_size } {\n                ADAPT::enable request false\n                ADAPT::enable response false\n            }\n        }\n    }\n  }\n}"
      ]
    }
  }
}
```

Поле	Тип	Описание
name	string	Уникальное имя aRule

Поле	Тип	Описание
<code>code</code>	string	Tcl-код с обработчиками событий (<code>when EVENT { body }</code>)
<code>description</code>	string	Описание
<code>enabled</code>	boolean	Включён (по умолчанию <code>true</code>)

aRules привязываются к сервисам (поле `arules` в `service`) или к правилам политики (поле `arules` в `policy rule`).

API для работы с конфигурацией

REST API предоставляет следующие операции чтения, проверки и применения:

Адрес	Метод	Описание
<code>/api/v1/config</code>	GET	Полная конфигурация
<code>/api/v1/config/{section}</code>	GET	Отдельная секция
<code>/api/v1/config/{section}</code>	PUT	Обновление секции
<code>/api/v1/config/validate</code>	POST	Проверка подготовленной конфигурации
<code>/api/v1/config/apply</code>	POST	Применение конфигурации
<code>/api/v1/config/confirm</code>	POST	Подтверждение (отмена таймера отката)
<code>/api/v1/config/revert</code>	POST	Возврат подготовленной конфигурации к действующей

Допустимые имена секций: `system`, `network`, `ssl`, `monitors`, `pools`, `services`, `service-chains`, `virtual-servers`, `gslb`, `policy`, `egress`, `protocol`, `topology`, `dataplane`, `ha`, `logging`, `snmp`, `observability`, `users`, `rate-shaping`, `arules`, `url-categories`.

YANG-модель конфигурации

Раздел объясняет связь YANG-модели с файлом `/config/amibig.json` и правила представления объектов в JSON. Он предназначен для проверки структуры конфигурации и разработки средств автоматизации.

Что такое YANG

YANG (RFC 7950) — язык моделирования данных для описания структуры конфигурации и состояния сетевых устройств. В Амибиг YANG-модель определяет:

- **структуру** — какие параметры существуют и как они организованы;
- **типы** — допустимые значения каждого параметра;
- **ссылки** — зависимости между объектами (`leafref`);
- **ограничения** — обязательные поля, диапазоны и перечисления.

Конфигурация Амибиг хранится в формате JSON. YANG-модель задаёт схему этого JSON: какие поля допустимы, каких типов, какие обязательны.

Модуль `amibig.yang`

Основной модуль использует пространство имён `Amicon` и стандартные типы IETF:

```
module amibig {
  yang-version 1.1;
  namespace "urn:amicon:amibig";
  prefix amibig;

  import ietf-yang-types { prefix yang; }
  import ietf-inet-types { prefix inet; }

  organization "000 AMICON";
  revision 2026-01-01 { description "Initial draft."; }
}
```

- `namespace urn:amicon:amibig` — уникальный идентификатор модуля;
- `prefix amibig` — сокращение для XPath-выражений;
- `import` — стандартные типы IETF: `inet:ip-address`, `inet:ipv4-prefix`, `yang:date-and-time`.

Текущий охват модели

Файл `yang/amibig.yang` пока описывает только основной HA-канал `ha.ha-link`. Резервный канал `ha.ha-link-secondary` поддерживается веб-интерфейсом, REST API и действующей конфигурацией, но ещё не представлен в YANG-модели. Поэтому средства автоматизации, которые строят схему только по YANG, не обнаружат поля резервного канала. Настройте их через экран **Мониторинг → HA** или REST API по правилам раздела Настройка отказоустойчивого кластера.

Пользовательские типы (typedef)

YANG-модель определяет типы для ссылок между объектами конфигурации:

Тип	Описание	Пример значения
<code>service-ref</code>	Ссылка на сервис по имени	<code>"ngfw"</code>

Тип	Описание	Пример значения
chain-ref	Ссылка на сервисную цепочку	"ngfw-dlp-chain"
cipher-rule-ref	Ссылка на правило шифрования	"default-cipher"
rate-class-ref	Ссылка на класс шейпинга	"low-priority"
cert-ref	Ссылка на сертификат	"mitm-ca"
pool-ref	Ссылка на пул	"web-pool"
monitor-ref	Ссылка на монитор доступности	"tcp-default"
vlan-ref	Ссылка на VLAN	"client-vlan"
interface-ref	Ссылка на физический интерфейс или транк	"enp0s8" , "bond0"
ip-prefix	IPv4 или IPv6 с длиной префикса	"10.0.0.0/24" , "fd00::/64"
port-number	Номер порта 0-65535	443

Каждый `*-ref` тип реализован как `leafref` — YANG-валидатор проверяет, что указанный объект действительно существует в конфигурации. Например, `service-ref` ссылается на путь `/amibig:config/services/service/name`.

Дерево контейнеров

Вся конфигурация находится внутри корневого контейнера `config`:

```

config
├── system                – имя узла, часовой пояс, управление, DNS, NTP
├── network
│   ├── interfaces       – физические интерфейсы и MTU каждого порта (list interface)
│   ├── trunks           – агрегированные каналы LACP dataplane (list trunk)
│   ├── vlans            – VLAN dataplane (list vlan)
│   ├── self-ips         – IP-адреса на dataplane VLAN или транке (list self-ip)
│   └── routes           – статические маршруты (list route)
├── ssl
│   ├── gost-*           – параметры перехвата ГОСТ-TLS
│   ├── certificates     – УЦ, серверные сертификаты и наборы (list certificate)
│   ├── trusted-store    – сборка набора доверенных УЦ (presence-контейнер),
│   │                     исключения по отпечатку (exclusions/exclusion)
│   ├── cipher-rules     – наборы шифров (list rule): ciphers (до TLS 1.2)
│   │                     и ciphersuites (TLS 1.3)
│   ├── client-profile   – профиль клиентского соединения: cert-lifespan,
│   │                     close-notify, handshake-timeout, ALPN и кэши
│   └── server-profile    – профиль серверной стороны: верификация сертификата,
│   │                     handshake-timeout
├── monitors             – мониторы доступности (list monitor)
├── pools                – пулы серверов (list pool)
└── services             – inline и ICAP сервисы (list service); у inline –
                        Self-IP в to-service/from-service (поля конкретного узла)
                        и сквозная проба traverse-probe

```

— service-chains	— цепочки сервисов (list chain)
— virtual-servers	— виртуальные серверы (list virtual-server)
— gslb	— глобальная балансировка (wide-IP)
— rate-shaping	— параметры классов трафика (list class), которые dataplane текущей версии не применяет
— policy	— правила классификации трафика (list rule) и переключатели уровня политики (reject-quick, udp-chain, icmp-chain); в match — критерии пользователя (source-user, source-user-group, source-user-class), в rule — identity-заголовки (insert-identity-headers, identity-header-format, identity-header-encoding)
— topology	— точка перехвата: слушатель на VLAN dataplane
— dataplane	— движок DPDK: config-path, hugepages, lcore-mask
— egress	— трансляция адреса источника к серверу (origin-snat, snat-pool)
— protocol	— политика TCP/MSS (tcp-mss-clamp, proxy-mss, pmtud)
— ha	— Active/Standby кластер
— logging	— системный лог, аудит, SSL, экспорт (syslog/Kafka)
— observability	— оповещения, передача журнала соединений в SIEM, выборка
— snmp	— мониторинг по SNMP
— users	— пользователи и список доступа к управлению (list user)
— arules	— Tcl-правила обработки (list arule)
— url-categories	— категории URL (list category)

Ключевые конструкции YANG

В модели используются стандартные конструкции YANG для одиночных значений, списков, ссылок и ограничений. Их представление в JSON зависит от типа конструкции.

list и leaf

Списки (list) содержат элементы с уникальным ключом:

```
list service {
  key "name";
  leaf name { type string; }
  leaf type { type enumeration { enum inline; enum icap; } }
  leaf enabled { type boolean; default true; }
  leaf service-down-action {
    type enumeration { enum bypass; enum drop; enum reset; }
  }
}
```

leafref (ссылки)

Ссылки гарантируют целостность — нельзя указать несуществующий объект:

```
leaf ca-certificate {
  type cert-ref; // → /config/ssl/certificates/certificate/name
}

leaf-list services {
  type service-ref; // → /config/services/service/name
  ordered-by user; // порядок задаётся пользователем
}
```

enumeration

Перечисления задают допустимые значения:

```
leaf mode {
  type enumeration {
    enum transparent-l2;
    enum transparent-l3;
  }
}
```

ordered-by user

Указывает, что порядок элементов списка имеет значение и определяется администратором (а не системой). Используется для цепочек сервисов и правил политики.

Преобразование YANG в JSON

YANG-модель транслируется в JSON-конфигурацию по следующим правилам:

YANG	JSON	Пример
container X	объект "X": {}	"system": { "hostname": "..."} }
leaf X	поле "X": value	"hostname": "amibig-01"
leaf-list X	массив "X": [...]	"nameservers": ["8.8.8.8"]
list X { key "name" }	массив объектов	"service": [{"name": "ngfw", ...}]
enumeration	строка	"type": "inline"
boolean	логическое значение	"enabled": true
uint16	число	"tag": 100

Пример: сервис в YANG и JSON

YANG-определение:

```
list service {
  key "name";
  leaf name { type string; }
  leaf type { type enumeration { enum inline; enum icap; } }
  leaf enabled { type boolean; default true; }
  leaf-list arules { type string; }
}
```

Соответствующий JSON:


```
{
  "services": {
    "service": [
      {
        "name": "ngfw",
        "type": "inline",
        "enabled": true,
        "arules": ["size_limit"]
      }
    ]
  }
}
```

Пример: leafref в действии

В YANG цепочка ссылается на сервисы:

```
list chain {
  key "name";
  leaf name { type string; }
  leaf-list services {
    type service-ref; // leafref → /config/services/service/name
    ordered-by user;
  }
}
```

В JSON — массив имён сервисов:

```
{
  "service-chains": {
    "chain": [{
      "name": "ngfw-dlp",
      "services": ["ngfw", "dlp"]
    }]
  }
}
```

Если сервис "dlp" не определён в services, валидация по YANG-модели выдаст ошибку.

Расположение файлов

Исходные модели и пример конфигурации находятся в каталоге yang/ :

Файл	Описание
yang/amibig.yang	Основная YANG-модель
yang/ietf-inet-types.yang	Стандартные сетевые типы IETF
yang/ietf-yang-types.yang	Базовые типы IETF
yang/example-config.json	Минимальный пример конфигурации

Дальнейшее чтение

Дополнительные сведения о полях и языке YANG:

- Схема конфигурации — полный справочник полей JSON-конфигурации
- [RFC 7950 — The YANG 1.1 Data Modeling Language](#)

Справочник командной строки администратора

Основные интерфейсы управления Амибиг — веб-интерфейс и REST API. В разделе приведены команды `curl`, которые можно выполнять с рабочей станции или из сеанса SSH. Примеры предполагают, что сертификат интерфейса управления добавлен в доверенные, а токен сохранён в переменной `TOKEN`.

Управление конфигурацией

Команды ниже охватывают проверку, применение и откат конфигурации:

```
### Просмотр текущей конфигурации
curl "https://<management-ip>/api/v1/config" \
  -H "Authorization: Bearer $TOKEN"

### Обновление секции конфигурации
curl -X PUT "https://<management-ip>/api/v1/config/policy" \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{ ... }'

### Проверка подготовленной конфигурации без применения
curl -X POST "https://<management-ip>/api/v1/config/validate" \
  -H "Authorization: Bearer $TOKEN"

### Применение конфигурации
curl -X POST "https://<management-ip>/api/v1/config/apply" \
  -H "Authorization: Bearer $TOKEN"

### Подтверждение конфигурации (отмена автоматического отката)
curl -X POST "https://<management-ip>/api/v1/config/confirm" \
  -H "Authorization: Bearer $TOKEN"

### Возврат подготовленной конфигурации к действующей
curl -X POST "https://<management-ip>/api/v1/config/revert" \
  -H "Authorization: Bearer $TOKEN"
```

Полный перечень имён секций приведён в разделе Схема конфигурации.

Диагностика

Для первичной диагностики используйте методы чтения версии, загрузки и журнала событий. Эти запросы не изменяют конфигурацию устройства.

Состояние системы

Запросите версию и текущую загрузку устройства:

```
### Версия и идентификация сборки
curl "https://<management-ip>/api/v1/system/version" \
  -H "Authorization: Bearer $TOKEN"

### Статистика системы (CPU, память, dataplane)
curl "https://<management-ip>/api/v1/system/stats" \
  -H "Authorization: Bearer $TOKEN"
```

Просмотр журналов

Параметры запроса позволяют ограничить объём и отобразить нужные события:

```
### Последние 50 системных событий
curl "https://<management-ip>/api/v1/logs?limit=50" \
  -H "Authorization: Bearer $TOKEN"

### Фильтрация по категории и уровню важности
curl "https://<management-ip>/api/v1/logs?category=security&severity=error&limit=100" \
  -H "Authorization: Bearer $TOKEN"
```

Состояние HA

Состояние локального узла и HA-партнёра возвращает следующий запрос:

```
curl "https://<management-ip>/api/v1/ha/status" \
  -H "Authorization: Bearer $TOKEN"
```

Мониторы доступности сервисов

Для проверки участников пулов запросите сводное состояние мониторов:

```
curl "https://<management-ip>/api/v1/traffic/health" \
  -H "Authorization: Bearer $TOKEN"
```

Управление лицензиями

API позволяет проверить состояние лицензии и загрузить новый файл:

```
### Текущий статус лицензии
curl "https://<management-ip>/api/v1/license/status" \
  -H "Authorization: Bearer $TOKEN"

### Аппаратный ID устройства
curl "https://<management-ip>/api/v1/license/hardware-id" \
  -H "Authorization: Bearer $TOKEN"

### Активные функции
```

```
curl "https://<management-ip>/api/v1/license/features" \
  -H "Authorization: Bearer $TOKEN"

### Загрузка лицензии
curl -X POST "https://<management-ip>/api/v1/license/upload" \
  -H "Authorization: Bearer $TOKEN" \
  -F "file=@device.lic"
```

Резервное копирование и восстановление

Создание и восстановление резервной копии требуют роли `admin` :

```
### Создание резервной копии
curl -o backup.acf "https://<management-ip>/api/v1/backup/export" \
  -H "Authorization: Bearer $TOKEN"

### Восстановление из резервной копии
curl -X POST "https://<management-ip>/api/v1/backup/import" \
  -H "Authorization: Bearer $TOKEN" \
  -F "file=@backup.acf"
```

Обновление системы

Сначала загрузите пакет и дождитесь успешной проверки, затем установите его в неактивный слот:

```
### Загрузка пакета обновления
curl -X POST "https://<management-ip>/api/v1/system/update/upload" \
  -H "Authorization: Bearer $TOKEN" \
  -F "file=@amibig-update.amibig"

### Применение обновления (установка в неактивный слот)
curl -X POST "https://<management-ip>/api/v1/system/update/apply" \
  -H "Authorization: Bearer $TOKEN"
```

Параметры `dataplane` задаются только через конфигурацию (секции `egress`, `protocol`, `dataplane`) и применяются через `config/apply` — отдельных флагов командной строки или переменных окружения для управления поведением обработки трафика не предусмотрено.

Артефакты релиза

Раздел описывает состав автономной поставки Амибиг и файлы, по которым проверяется её целостность. Используйте его при установке, обновлении и передаче релиза в изолированный контур.

Обзор

Для автономной поставки (без доступа в интернет) Амибиг использует один явный набор артефактов релиза. Администратору и технической поддержке не нужно восстанавливать состав релиза по служебным каталогам сборки или вспомогательным скриптам вручную.

Корневой машинно-читаемый файл описи релиза:

- `amibig-release-manifest.json`

Контрольные суммы артефактов верхнего уровня:

- `SHA256SUMS`

Поддерживаемый набор артефактов автономной поставки

В поставку для изолированного контура могут входить следующие файлы:

Файл	Роль	Назначение	Обычно нужен для
<code>amibig-installer.hdim.gz</code>	<code>installer-image</code>	Загрузочный образ автономного инсталлятора	первичная установка, носитель восстановления
<code>installer-vendor.tar.gz</code>	<code>installer-vendor</code>	Зафиксированные зависимости среды инсталлятора	пересборка инсталлятора, восстановление поддержкой
<code>amibig-offline-repo.tar.gz</code>	<code>offline-apt-repo</code>	Автономный репозиторий пакетов	установка без сети, подготовка зависимостей
<code>amibig-update-v<version>[-b<build>].amibig</code>	<code>update-package</code>	Пакет обновления для A/B-схемы слотов	обновление без сети

Артефакты для проверки релиза

Эти файлы не являются установочными носителями, но входят в состав релиза:

Файл	Роль	Назначение
<code>amibig-sbom.spdx.json</code>	<code>build-sbom</code>	SBOM для отслеживания происхождения компонентов и проверки релиза

Файл	Роль	Назначение
<code>amibig-release-manifest.json</code>	<code>release-manifest</code>	Опись верхнего уровня и статус каждого ожидаемого артефакта
<code>SHA256SUMS</code>	<code>checksum-set</code>	Контрольные суммы SHA-256 только для поддерживаемых артефактов верхнего уровня

Что публикует `amibig-release-manifest.json`

Манифест верхнего уровня фиксирует:

- `schema_version`
- `generated_at`
- `product.name`, `product.version`, `product.build_number`
- идентификацию сборки из `ci/build-manifest.env`
- `artifacts[]` с полями:
 - `role`
 - `filename`
 - `path`
 - `purpose`
 - `required_for`
 - `support_level`
 - `status`
 - `checksum` для присутствующих файлов

Манифест может пометить ожидаемый артефакт как `status: missing` — соглашение об именовании и составе описи при этом остаётся действительным; полноту набора проверяйте по полю `status` каждого артефакта.

Побочные файлы сборки

Следующие файлы или каталоги могут существовать рядом со сборкой, но не входят в состав поставки:

- `.manifest` в выходном каталоге сборщика автономного репозитория;
- `.image-id` в выходном каталоге сборщика автономного репозитория;
- распакованный каталог репозитория до упаковки в `amibig-offline-repo.tar.gz`;
- временные файлы этапов сборки;
- несжатые промежуточные образы, используемые только внутри конвейера сборки.

Такие файлы могут фигурировать в манифесте только как служебные (`helper-only`), но не как артефакты релиза верхнего уровня.

Практическое правило

Если нужно понять, что именно входит в автономную поставку конкретной сборки:

1. откройте `amibig-release-manifest.json` ;
2. сверьте присутствующие файлы поставки;
3. проверьте `SHA256SUMS` ;
4. уже потом смотрите встроенные манифесты внутри отдельных артефактов, таких как `.amibig` .

Системные лимиты

В разделе приведены ограничения модели конфигурации и ориентиры производительности текущей версии Амибиг. Для приёмочных испытаний сверяйте их с требованиями к конкретной аппаратной платформе.

Лимиты конфигурации

При превышении этих значений проверка конфигурации завершается ошибкой:

Параметр	Ограничение
Максимум сервисов (inline + ICAP)	32
Максимум участников пула на один сервис	16
Максимум сервисных цепочек	16
Максимум сервисов в одной цепочке	8
Максимум правил политики	64
Максимум шаблонов SNI в одном правиле	8
Максимум подсетей CIDR источника или назначения в одном правиле	8
Максимум портов назначения в одном правиле	8
Максимальная длина имени объекта	64 символа
Максимум определённых aRules	32
Максимум обработчиков событий в одном aRule	6

Параметр	Ограничение
Максимум aRule-ссылок на сервис или правило	8
Максимум адресов в <code>egress.snat-pool</code>	16
<code>min-healthy-members</code> сервиса	1-255

Протокол (TCP / MSS) и MTU

Встроенные значения и допустимые диапазоны сетевых параметров:

Параметр	Значение
<code>protocol.tcp-mss-clamp</code> по умолчанию	1360
Допустимый диапазон ограничения MSS	0 (выключено) или 536-1460
Направленные клампы (<code>tcp-mss-clamp-client</code> / <code>-origin</code>) по умолчанию	0 (выключены)
<code>proxy-mss</code> по умолчанию	включён
<code>pmtud-blackhole-detect</code> по умолчанию	включён
Резервное значение MSS при обнаружении недоступности PMTUD	1200
MTU DPDK-порта (<code>network.interface.mtu</code>)	576-1500 (только уменьшение; без значения — 1500)

Лимиты производительности

Показатели ниже являются целевыми ориентирами продукта, а не гарантией для любой конфигурации. Фактические значения подтверждаются нагрузочным тестированием на целевом оборудовании.

Параметр	Целевое значение
Базовый порог интенсивности соединений	1 млн соединений/мин (около 16,7 тыс. соединений/с)
Ориентир интенсивности соединений	100 тыс. соединений/с — хороший результат; 200 тыс. соединений/с — целевой
Пропускная способность TLS-инспекции	не менее 20 Гбит/с

Лимиты сети

Сетевая модель принимает следующие диапазоны:

Параметр	Ограничение
Тег VLAN	1-4094
Номер порта	0-65535

Лимиты веб-интерфейса и API

Ограничения защищают устройство от неограниченного роста оперативных данных:

Параметр	Значение
Буфер событий экрана <code>Трафик</code>	5 000 событий
Максимум записей в журнале событий	50 000
Максимум одновременных захватов пакетов	10
Длительность захвата по умолчанию	60 секунд
Размер захвата по умолчанию	50 МБ
Максимальный <code>limit</code> в запросах трафика и журнала	500
Таймаут автоматического отката конфигурации	120 секунд

Хранение событий и истории оповещений

Если значения не переопределены, применяются следующие сроки и объёмы:

Параметр	Значение по умолчанию
<code>observability.retention.events-days</code>	90 дней (минимум 1)
<code>observability.retention.alert-history-days</code>	90 дней (минимум 1)
<code>observability.retention.alert-history-max-rows</code>	20 000 записей (минимум 100)

Лимиты лицензирования

Для демонстрационного и льготного режимов действуют следующие сроки:

Параметр	Значение
Демонстрационный период	30 дней
Льготный период по умолчанию	30 дней

Лимиты обновлений

Механизм А/В использует следующие значения при проверке нового слота:

Параметр	Значение
Попытки загрузки до автоматического отката	3
Время ожидания проверки доступности при загрузке	120 секунд

Коды сообщений

Каждое операторское событие Амибиг имеет стабильный код вида `AMB-NNNN`. Код выводится префиксом в тексте сообщения (например, «AMB-2002: Login failed...») и полем `msg_id` в данных события — как в журнале веб-интерфейса, так и в записях, отправляемых на `syslog/SIEM`. Код не меняется между версиями: на него можно ссылаться при обращении в техническую поддержку Amicon и в правилах корреляции SIEM. Первая цифра кода определяет подсистему, уровень важности в таблице — типичный для события; фактический уровень отдельного события может отличаться в зависимости от контекста.

1xxx — Система и обновления

Коды диапазона 1xxx описывают жизненный цикл системы: запуск и остановку компонентов, установку обновлений, состояние системных служб и критические события ядра.

Код	Уровень	Подсистема	Описание
AMB-1001	warning	system	Создан пользователь admin с начальным паролем

Код	Уровень	Подсистема	Описание
AMB-1002	info	system	Плоскость управления готова к работе
AMB-1003	info	system	REST API запущен
AMB-1004	info	system	REST API останавливается
AMB-1010	warning	update	Принят неподписанный пакет обновления формата v1 (переходный период)
AMB-1011	info	update	Пакет обновления проверен и готов к установке
AMB-1012	error	update	Проверка пакета обновления завершилась ошибкой
AMB-1013	info	update	Обновление установлено в неактивный слот
AMB-1014	error	update	Установка обновления завершилась ошибкой
AMB-1015	warning	update	Подготовлен откат на предыдущий слот
AMB-1016	info	system	Запрошена перезагрузка системы
AMB-1020	warning	system	Изменилось состояние лицензии
AMB-1021	info	system	Установлен набор правил оповещений по умолчанию
AMB-1022	info	system	Служба AmiTLS перезапущена

Код	Уровень	Подсистема	Описание
AMB-1023	warning	system	Срок подписки лицензии истекает
AMB-1030	info	system	Запрошен запуск системной службы
AMB-1031	info	system	Системная служба готова к работе
AMB-1032	warning	system	Системная служба не подтвердила готовность за отведённое время
AMB-1033	info	system	Системная служба остановлена штатно
AMB-1034	error	system	Системная служба завершилась аварийно
AMB-1040	error	system	Сохранён core dump аварийно завершившегося процесса
AMB-1045	info	system	Создан диагностический архив
AMB-1050	info	system	Событие процедуры начальной загрузки
AMB-1051	critical	system	При загрузке обнаружены ошибки файловой системы
AMB-1060	critical	system	Сработал OOM-killer ядра
AMB-1061	error	system	Ошибка дискового ввода-вывода в ядре

2xxx — Аутентификация и аудит

Коды диапазона 2xxx фиксируют вход и выход пользователей, смену паролей, операции с учётными записями и другие действия, попадающие в журнал аудита.

Код	Уровень	Подсистема	Описание
АМВ-2001	warning	auth	Попытка входа отклонена ограничителем частоты
АМВ-2002	warning	auth	Неудачная попытка входа
АМВ-2003	warning	auth	Вход временно заблокирован после серии неудачных попыток
АМВ-2004	info	auth	Успешный вход пользователя
АМВ-2005	info	auth	Выход пользователя из системы
АМВ-2006	warning	auth	Не удалось сменить пароль пользователя
АМВ-2007	info	auth	Пароль пользователя изменён
АМВ-2010	warning	auth	Запрос заблокирован списком доступа управления
АМВ-2020	info	audit	Создана учётная запись пользователя
АМВ-2021	info	audit	Удалена учётная запись пользователя
АМВ-2022	info	audit	Изменена роль пользователя
АМВ-2030	info	audit	Диагностический архив создан и выдан пользователю

3xxx — Обработка трафика (dataplane)

Коды диапазона 3xxx относятся к dataplane: телеметрия, поколения конфигурации, счётчики ошибок, состояние интерфейсов, пулов и агрегированных каналов.

Код	Уровень	Подсистема	Описание
AMB-3001	info	dataplane	Установлено подключение к телеметрии dataplane
AMB-3002	info	dataplane	Телеметрия dataplane восстановлена
AMB-3003	error	dataplane	Телеметрия dataplane недоступна
AMB-3004	info	dataplane	Изменилось runtime-поколение dataplane
AMB-3005	info	dataplane	Dataplane применил новое поколение конфигурации
AMB-3006	warning	dataplane	Вырос счётчик ошибок dataplane
AMB-3007	warning	dataplane	Остаток данных TLS-соединения передан другому рабочему потоку
AMB-3008	warning	dataplane	Остаток данных незашифрованного соединения передан другому рабочему потоку
AMB-3009	info	dataplane	Изменился статус участника пула
AMB-3010	info	dataplane	Изменилось состояние link на интерфейсе
AMB-3011	info	dataplane	Изменился статус участника агрегированного канала

Код	Уровень	Подсистема	Описание
AMB-3020	error	dataplane	Ошибка выполнения aRules runtime
AMB-3021	critical	dataplane	Ошибка инициализации DPDK/F-Stack
AMB-3022	warning	dataplane	Ускорение TLS (QAT) деградировало — операции выполняются на CPU

4xxx — Кластер HA

Коды диапазона 4xxx описывают работу кластера HA: синхронизацию конфигурации, смену ролей узлов, виртуальные IP-адреса и связь с парным узлом.

Код	Уровень	Подсистема	Описание
AMB-4001	info	ha	Конфигурация синхронизирована с активного узла
AMB-4002	error	ha	Не удалось применить синхронизированную конфигурацию
AMB-4003	info	ha	Сгенерирован ключ STEK для возобновления TLS-сессий
AMB-4004	info	ha	Ключ STEK получен с активного узла
AMB-4005	error	ha	Истёк таймаут heartbeat от парного узла
AMB-4010	info	ha	Ручное переключение ролей выполнено или отменено
AMB-4011	warning	ha	Запрошено ручное переключение ролей
AMB-4012	warning	ha	Изменилась роль узла в HA-паре

Код	Уровень	Подсистема	Описание
АМВ-4013	info	ha	Виртуальный IP-адрес активирован на этом узле
АМВ-4014	info	ha	Виртуальный IP-адрес снят с этого узла
АМВ-4020	info	ha	Установлено соединение с парным узлом
АМВ-4021	warning	ha	Соединение с парным узлом разорвано
АМВ-4022	info	ha	HA-демон запущен
АМВ-4030	info	ha	Запущена ручная синхронизация конфигурации
АМВ-4031	warning	ha	Пользователь запросил ручное переключение ролей
АМВ-4032	info	ha	Загружен CA-сертификат парного узла
АМВ-4033	info	ha	Удалён CA-сертификат парного узла
АМВ-4040	info	ha	Узел стал владельцем трафика (MASTER)
АМВ-4041	info	ha	Узел освободил владение трафиком (BACKUP/FAULT)

5xxx — SSL/TLS

Коды диапазона 5xxx связаны с сертификатами и проверкой подписи пакетов обновления.

Код	Уровень	Подсистема	Описание
AMB-5001	error	ssl	Проверка подписи пакета обновления завершилась ошибкой
AMB-5002	info	ssl	Добавлен доверенный ключ подписи обновлений
AMB-5003	warning	ssl	Отозван ключ подписи обновлений
AMB-5010	info	ssl	Загружен сертификат
AMB-5011	info	ssl	Применён CA-сертификат для SSL-инспекции
AMB-5020	error	ssl	Ошибка загрузки или перезагрузки TLS-сертификата

6xxx — Конфигурация

Коды диапазона 6xxx фиксируют операции с конфигурацией: импорт, применение, откат, резервные копии и историю изменений.

Код	Уровень	Подсистема	Описание
AMB-6001	info	config	Конфигурация импортирована
AMB-6002	info	config	Импортирована конфигурация F5 UCS
AMB-6003	info	config	Конфигурация возвращена к сохранённой версии
AMB-6004	info	config	Запущено применение конфигурации

Код	Уровень	Подсистема	Описание
AMB-6005	warning	config	Применение конфигурации заблокировано ограничениями лицензии
AMB-6006	warning	config	Применение конфигурации заблокировано проверкой безопасности
AMB-6007	error	config	Применение конфигурации завершилось ошибкой
AMB-6008	info	config	Конфигурация применена без таймера отката
AMB-6009	info	config	Конфигурация применена и ожидает подтверждения
AMB-6010	info	config	Применён шаблон топологии
AMB-6011	info	config	Запущена визуальная идентификация порта
AMB-6012	info	config	Обновлена секция конфигурации
AMB-6013	error	config	Применение конфигурации отклонено: число рабочих потоков превышает число RX-очереди сетевого адаптера
AMB-6014	warning	config	Конфигурация применена, но часть её датаплейн не использует

Код	Уровень	Подсистема	Описание
АМВ-6020	info	config	Применение конфигурации подтверждено пользователем
АМВ-6030	info	config	Резервная копия конфигурации выгружена
АМВ-6031	info	config	Резервная копия конфигурации загружена
АМВ-6040	info	config	Конфигурация восстановлена из снимка истории
АМВ-6041	info	config	Удалена запись истории конфигурации

7xxx — Оповещения

Коды диапазона 7xxx порождает подсистема оповещений при срабатывании и снятии правил.

Код	Уровень	Подсистема	Описание
АМВ-7001	warning	alert	Сработало правило оповещения
АМВ-7002	info	alert	Оповещение снято — условие правила больше не выполняется