

Программа для ЭВМ
«UBIC Adblock»
Архитектура системы

Оглавление

1. ОБЩЕЕ ОПИСАНИЕ АРХИТЕКТУРЫ.....	3
2. КОМПОНЕНТЫ СИСТЕМЫ	3
2.1 Движок обработки и классификации пакетов.....	3
2.2 Модуль каталога рекламных доменов	3
2.3 adb-api – REST API интеграции и провизининга.....	3
2.4 Модуль управления комплексом.....	4
2.5 Подсистема идентификации абонентов.....	4
2.6 Прием пакетов (Packet Broker)	4
3. ПОТОК ДАННЫХ.....	5
4. ТЕХНОЛОГИЧЕСКИЙ СТЕК.....	5
5. СИСТЕМНЫЕ ТРЕБОВАНИЯ.....	5

1. ОБЩЕЕ ОПИСАНИЕ АРХИТЕКТУРЫ

Программа для ЭВМ «UBIC Adblock» построена как набор взаимодействующих модулей, устанавливаемых на сети оператора связи на площадке агрегации пакетов.

Основу решения составляют высокопроизводительный движок обработки и классификации пакетов (C++), самопополняемый каталог рекламных доменов и управляющий REST API идентификации абонентов и провизининга услуги (компонент adb-api).

Все модули строго взаимосвязаны в рамках единой инфраструктуры; ни один модуль не решает задачу самостоятельно. Модуль классификации получает все сетевые пакеты, но перенаправление либо модификацию рекламы выполняет только для абонентов с подключенной услугой – по абонентскому признаку, передаваемому модулем интеграции.

Комплекс работает на стандартных x86-серверах (архитектура 64 бита, IPv4/IPv6) и рассчитан на обработку пакетов до 10 / 40 / 100 Гбит/с в зависимости от условий внедрения. Отказоустойчивость обеспечивается на уровне, при котором сбой сервиса обработки, перенаправления либо модификации рекламных сообщений не нарушает основную услугу доступа абонента в Интернет.

2. КОМПОНЕНТЫ СИСТЕМЫ

2.1 Движок обработки и классификации пакетов

Обеспечивает захват сетевых пакетов (Core Packet Processing – разбор Ethernet/IP/TCP/UDP), идентификацию абонента, определение обращения к рекламному домену по заголовку HTTP Host и расширению SNI протокола TLS и, собственно, перенаправление либо модификацию пакетов. Для HTTP отправляется ответ-заглушка, для HTTPS соединение к рекламному домену разрывается (TCP RST). Управление модулем осуществляется через менеджмент-сервис. Режимы работы: пассивная копия (зеркало) и активная копия (установка «в разрыв»).

2.2 Модуль каталога рекламных доменов

Ведет каталог рекламных ресурсов сети Интернет. Каталог загружается из внешнего хранилища по расписанию и тиражируется на модули классификации. Ключевая уникальная функция – автоматический регулярный поиск новых рекламных доменов и их автоматическая классификация на базе анализа пользовательских пакетов (самопополнение каталога). Предусмотрены дополнительные интерфейсы для подключения других внешних источников данных.

2.3 adb-api – REST API интеграции и провизининга

Управляющий REST-интерфейс идентификации абонентов и провизининга услуги. Реализован на PHP (Mezzio/Laminas, архитектура CQRS). Предоставляет внешним системам оператора эндпоинты управления абонентами. Принятые настройки сохраняются в базах DB_CORE и DB_CATALOG и транслируются на модули классификации. Отчеты о статистике формируются подсистемой StatisticProcessing/ReportSender.

Сервис adb-api поставляется в составе дистрибутива в виде Docker-образа и запускается в контейнере. В качестве сервера приложений внутри контейнера используется RoadRunner (точка входа public/index.php, HTTP-воркер bin/http-worker.php); контейнер стартует командой `rr serve`. Конфигурация задается через переменные окружения (APP_MODE, API_KEY, MAINTENANCE, TZ, DB_CORE, DB_CATALOG). В комплект входят вспомогательные образы swagger-ui (публикация спецификации docs/swagger.yaml), mysql-dev и php-dev для локальной разработки и тестирования.

2.4 Модуль управления комплексом

Обеспечивает мониторинг работы комплекса (нагрузка и температура процессоров, наполнение баз данных) и использования, управление группами абонентов и групповыми политиками, ручное пополнение каталога рекламных доменов, ведение списка сайтов-исключений и взаимодействие с внешними системами мониторинга работоспособности.

2.5 Подсистема идентификации абонентов

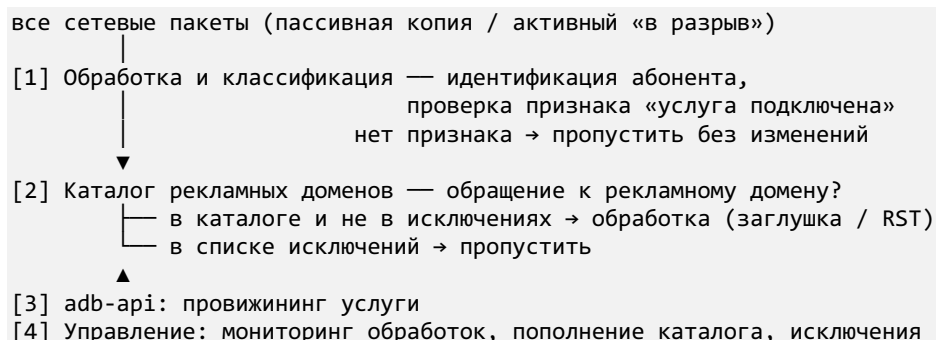
Идентифицирует абонента в общих пакетах во взаимодействии с AAA-серверами: получение копии AAA-пакетов (RADIUS/DIAMETER), таблиц статических IP серверов и таблиц NAT. Работает в двух режимах – распределенном (каждый модуль классификации идентифицирует самостоятельно) и централизованном (идентификация поступает из модуля управления). Модули классификации держат текущий срез идентификации в оперативной памяти.

2.6 Прием пакетов (Packet Broker)

Прием и распределение зеркалируемых пакетов может выполняться через Packet Broker (например, Gigamon), что позволяет подавать копию сетевых пакетов на модули классификации и переиспользовать оборудование в узлах внедрения.

3. ПОТОК ДАННЫХ

Обработка сетевых пакетов организована по следующей схеме:



4. ТЕХНОЛОГИЧЕСКИЙ СТЕК

Категория	Технологии
Движок классификации	C++11, CMake
Прием пакетов	Packet Broker (Gigamon), пассивная/активная копия
Управляющий API	PHP, Mezzio/Laminas, архитектура CQRS, REST (Swagger, ApiKeyAuth)
Развертывание API	Docker-контейнер, сервер приложений RoadRunner; конфигурация через ENV
Идентификация	AAA: RADIUS/DIAMETER, статические IP, таблицы NAT
Хранилища данных	СУБД: DB_CORE, DB_CATALOG

5. СИСТЕМНЫЕ ТРЕБОВАНИЯ

Программные требования: операционная система семейства Linux (Debian/Ubuntu/RHEL); компилятор C++11 и CMake для сборки движка классификации; среда Docker для запуска компонента adb-api (образ включает сервер приложений RoadRunner); СУБД для баз DB_CORE и DB_CATALOG.

Аппаратные требования: стандартные x86-серверы, архитектура 64 бита, поддержка IPv4/IPv6; производительность, обеспечивающая обработку пакетов до 10 / 40 / 100 Гбит/с на комплекс; сетевые интерфейсы или подключение через Packet Broker, соответствующие объему обрабатываемых пакетов. Отказоустойчивость – на уровне, исключающем нарушение основной услуги доступа в Интернет при сбое сервиса.