

Документация, содержащая описание функциональных характеристик экземпляра программного обеспечения, предоставленного для проведения экспертной проверки

Программа для ЭВМ
«UBIC CORE»

Оглавление

ТЕРМИНЫ И СОКРАЩЕНИЯ	3
1. ОБЩИЕ СВЕДЕНИЯ.....	5
1.1 Наименование программы	5
1.2 Используемые языки программирования.....	5
1.3 Системные требования	5
2. ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ.....	6

ТЕРМИНЫ И СОКРАЩЕНИЯ

В настоящем документе используются следующие термины и сокращения:

Термин / сокращение	Расшифровка / определение
DPI	Deep Packet Inspection – технология анализа содержимого сетевых пакетов на уровне протоколов приложений
libpcap	Программная библиотека для чтения сетевых пакетов на уровне сетевого интерфейса
TLS	Transport Layer Security – протокол защищенного соединения, обеспечивающий шифрование передаваемых данных
SNI	Server Name Indication – поле в составе TLS-рукопожатия, указывающее имя запрашиваемого сервера
TCP	Transmission Control Protocol – протокол передачи данных с установлением соединения
RST	Reset – служебный флаг TCP-пакета, используемый для управления сетевым соединением
HTTP/HTTPS	Hypertext Transfer Protocol (Secure) – протокол передачи гипертекста; HTTPS – его защищенная версия с использованием TLS
DNS	Domain Name System – система преобразования доменных имен в IP-адреса
RADIUS	Remote Authentication Dial-In User Service – протокол аутентификации, авторизации и учета, применяемый операторами связи
RPC	Remote Procedure Call – удаленный вызов процедур; механизм взаимодействия программных компонентов
ZeroMQ	Библиотека для организации обмена сообщениями между процессами, в том числе в режимах PUSH/PULL и REQ/REP
Protocol Buffers	Формат сериализации структурированных данных
VLAN	Virtual Local Area Network – виртуальная локальная сеть; идентификатор VLAN используется совместно с IP-адресом для определения абонента
MD5	Алгоритм хеширования, применяемый для формирования контрольных сумм данных
Boost.Interprocess	Библиотека языка C++ для организации межпроцессного взаимодействия через разделяемую память
__SHID__	Именованный сегмент разделяемой памяти, используемый для хранения таблицы идентификации абонентов
__PSTATS__	Именованный сегмент разделяемой памяти, используемый для хранения счетчиков эксплуатационной статистики
pfletcher2	Компонент Программы, реализующий ядро системы: чтение пакетов, разбор протоколов HTTP/TLS и классификацию полезной нагрузки
dns-tracer	Компонент Программы, осуществляющий чтение и перенаправление DNS-ответов
radius-tracer	Компонент Программы, осуществляющий разбор данных RADIUS-аккаунтинга
identification	Компонент Программы, загружающий карту идентификации абонентов из СУБД MySQL в разделяемую память
catalog2	Компонент Программы, загружающий URL-каталог из СУБД MySQL в разделяемую память
pstats	Утилита управления сегментом эксплуатационной статистики
Ocat	Клиент командной строки для взаимодействия со служебным RPC-интерфейсом Программы

Термин / сокращение	Расшифровка / определение
su13lib	Вспомогательная библиотека сетевых утилит, применяемая компонентами Программы
linnaeus	Компонент Программы, реализующий работу с URL-каталогом
rosenthal	Компонент Программы, реализующий разбор поисковых запросов и паттернов

1. ОБЩИЕ СВЕДЕНИЯ

1.1 Наименование программы

Программа для ЭВМ «UBIC CORE» (далее – Программа, Система) – система анализа сетевых пакетов и умной маршрутизации, функционирующая на уровне сети провайдера или оператора связи.

1.2 Используемые языки программирования

- C++11 ((g++/clang), основной язык всех компонентов, сборка через CMake);
- Bash (вспомогательные скрипты, например `ipv6_resolver.sh`).

Конфигурация компонентов задается в формате JSON. Обмен служебными командами осуществляется в формате JSON с использованием транспортного протокола ZeroMQ. Выгрузка результатов выполняется в формате Protocol Buffers.

1.3 Системные требования

Программные требования:

Операционная система семейства Linux (рекомендуется Debian/Ubuntu/RHEL) с поддержкой `libpcap` для захвата сетевого трафика.

Для сборки и работы требуются библиотеки: `GFlags`, `ZeroMQ`, `libpcap`, `Threads`, `Protocol Buffers`, `Boost` (`Interprocess`), `MySQL Connector/C++`.

Для хранения исходных карт идентификации и URL-каталога используется СУБД `MySQL` (база `safeinet`).

Аппаратные требования:

Минимальные, необходимые для тестирования:

- 8 вычислительных ядер CPU, 16 ГБ ОЗУ, 50 ГБ SSD.

Производственные:

- вычислительный узел: 16 и более ядер ЦПУ, от 32 ГБ ОЗУ (сегмент идентификации `__SHID__` по умолчанию рассчитан на объем порядка 8 ГБ), SSD;
- один или несколько сетевых интерфейсов с пропускной способностью, соответствующей объему обрабатываемого трафика.

2. ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ

Программа обрабатывает сетевые пакеты в режиме, близком к режиму реального времени, идентифицирует абонента, классифицирует запрашиваемый контент, при необходимости осуществляет перенаправление (маршрутизацию) пакетов, а также ведет учет и выгрузку статистики во внешнюю систему аналитической обработки данных.

Программа построена как набор отдельных процессов, обменивающихся данными через разделяемую память (Boost.Interprocess) и очереди сообщений ZeroMQ, что обеспечивает высокую производительность и горизонтальную масштабируемость обработки.

Программа решает четыре основные задачи:

- идентификация абонента – по IP-адресу (или сочетанию IP-адреса и идентификатора VLAN) определяется пользователь и применимые к нему политики классификации пакетов;
- классификация трафика – для каждого HTTP/HTTPS-запроса определяется сайт, категория контента и соответствующий категории маршрут;
- учет и выгрузка статистики – ведется подсчет статистики и выгрузка «кликов» (фактов посещения URL) во внешнюю систему аналитической обработки данных.

Программа обеспечивает выполнение следующих функций:

- чтение пакетов посредством библиотеки libpcap в режимах online (получение данных с сетевого интерфейса), offline (обработка ранее сохраненного pcap-файла) и inmemory (обработка данных, размещенных в оперативной памяти);
- разбор протоколов HTTP (посредством zero-core парсера) и извлечение SNI из TLS-рукопожатия без расшифровки содержимого трафика;
- сборка TCP-сессий, в том числе при поступлении сегментов вне порядка;
- категоризация URL по иерархии каталогов (локальные переопределения, персональные категории, категории по расширению файла, основной каталог);
- разбор поисковых запросов наиболее популярных поисковых систем (Google, Yandex, Mail.ru, Bing, VK, Rambler) и фильтрация запросов по ключевым словам;
- реализация функций управления интернет-потреблением на основе заданных контрольных параметров;
- перенаправление пакетов в соответствии с определенной категорией контента;
- перенаправление DNS-ответов (порт 53) для абонентов в соответствии с политиками интернет-потребления абонентов;
- динамическое наполнение таблицы идентификации абонентов по данным RADIUS-аккаунтинга (порт 1813);
- загрузка статических карт идентификации и URL-каталога из СУБД MySQL в разделяемую память;
- сбор и просмотр эксплуатационной статистики функционирования Программы;
- сериализация результатов в формате Protocol Buffers и их выгрузка по ZeroMQ во внешнюю систему аналитической обработки данных;

- управление каталогами и правилами в реальном времени через служебный RPC-интерфейс.

Программа состоит из следующих компонентов:

- pfletcher2 – ядро системы: чтение пакетов, разбор HTTP/TLS, классификация пакетов, маршрутизация пакетов, выгрузка результатов;
- dns-tracer – перенаправление DNS-ответов в соответствии с политиками интернет-потребления абонентов;
- radius-tracer – разбор RADIUS-аккаунтинга и динамическое наполнение таблицы абонентов;
- identification – загрузчик карты «IP/логин → пользователь» из MySQL в разделяемую память __SHID__;
- catalog2 – загрузчик URL-каталога из MySQL в разделяемую память (дерево доменов);
- pstats – утилита управления сегментом статистики __PSTATS__;
- 0cat – CLI-клиент управляющего RPC-интерфейса pfletcher2 и dns-tracer.

Дополнительно используются библиотеки:

- su13lib (сетевые утилиты, MD5, потоки и lock-free очереди);
- linnaeus (движок URL-каталогов и управления интернет-потреблением);
- rosenthal (движок DACT для разбора поисковых запросов и паттернов).

Основные механизмы обработки трафика:

- разделяемая память (Boost.Interprocess) – именованные сегменты __SHID__ (идентификация абонентов), __PSTATS__ (счетчики статистики) и URL-каталог, к которым одновременно обращаются несколько процессов; обновление таблиц выполняется атомарной подменой без блокировки читателей;
- ZeroMQ – механизмы обмена сообщениями: PUSH/PULL (однаправленная передача данных для выгрузки результатов) и REQ/REP (запрос-ответ для служебного RPC-интерфейса управления каталогами и правилами).

Поддерживаемые режимы обработки трафика:

Режим	Описание
online	Чтение пакетов в реальном времени
offline	Обработка ранее записанного рсар-дампа
inmemory	Обработка пакетов из памяти (тестовые и отладочные сценарии)