

Программный комплекс обработки и классификации пакетного трафика «TRACE TA»

Руководство пользователя

Версия документа: 1.41.0 (2025-10-03)

АННОТАЦИЯ

Настоящее руководство пользователя является основным документом для получения всей информации о программе, необходимой для её применения. В нём описаны назначение программы, функциональность, состав, принцип работы, условия применения и порядок действий пользователя.

Читателю необходимо иметь представление об устройстве и применении современных IBM-PC совместимых ЭВМ под управлением операционных систем семейства Linux, о технологиях разработки программного обеспечения на языке C++, о работе информационно-телекоммуникационных сетей, в первую очередь сети Интернет.

СОДЕРЖАНИЕ

1	Описание и работа.....	4
1.1	Наименование, обозначение	4
1.2	Назначение, функциональность	4
1.3	Состав	5
1.3.1	Логическая структура.....	5
1.3.2	Дистрибутив	5
1.4	Входные данные	6
1.5	Выходные данные.....	6
1.6	Выполнение программы	7
1.6.1	Общий порядок выполнения	7
1.6.2	Модель выполнения и основные объекты API.....	7
1.6.3	Включение и отключение обработчиков протоколов и атрибутов	9
1.6.4	Реакция на системные сигналы	10
1.6.5	Пользовательские признаки и обработчики протоколов	11
2	Использование	14
2.1	Условия применения	14
2.1.1	На этапе интеграции в пользовательское приложение	14
2.1.2	На этапе выполнения пользовательского приложения	14
2.2	Подготовка к работе	14
2.2.1	Получение (обновление) дистрибутива.....	14
2.2.2	Интеграция в пользовательское приложение	15
2.2.3	Развёртывание пользовательского приложения	15
2.3	Работа с программой.....	15
2.3.1	Основная функциональность.....	15
2.3.2	Справочная система «Прототека».....	15
	Приложение 1. Перечень поддерживаемых протоколов и сервисов.....	18
	Перечень терминов и сокращений	25

1 ОПИСАНИЕ И РАБОТА

1.1 Наименование, обозначение

Наименование: программный комплекс обработки и классификации пакетного трафика «TRACE TA».

Сокращённое наименование: ПК «TRACE TA».

Регистрация: свидетельство о государственной регистрации программного обеспечения № 2024662981 от 03.06.2024 (Роспатент), запись № 24378 от 18.10.2024 в реестре программного обеспечения (Минцифры).

1.2 Назначение, функциональность

ПК «TRACE TA» (TRACE – TRaffic Analysis and Classification Engine) представляет собой набор программных компонентов (SDK для C/C++), предназначенный для встраивания в пользовательские приложения с целью добавления функциональности анализа и классификации сетевого пакетного трафика.

Область применения – сетевые решения и решения для кибербезопасности: системы применения политик управления качеством и тарификации трафика (PCEF), межсетевые экраны (FW), программно-определяемые сети (SD-WAN), системы обнаружения и предотвращения вторжений и утечек (IPS/IDS/DLP) и др.

Основные функциональные возможности:

- глубокий анализ пакетов (DPI) до уровня L7 включительно сетевой модели OSI;
- обработка зашифрованного и обфусцированного трафика (анонимайзеры, туннели);
- классификация сетевых потоков – определение протоколов и сервисов, с которыми ассоциируется поток (более 500 наименований классов, см. Приложение 1);
- извлечение и обновление метаданных в режиме реального времени;
- идентификация типов услуг для некоторых сервисов (например, для ряда мессенджеров могут определяться текстовый чат, аудио- или видеозвонок, передача файлов);
- поддержка основных существующих групп методов классификации трафика: сигнатурные (по значениям полей заголовков пакетов, например, TCP/UDP-портам, IP-адресам, TLS SNI, TLS CN), эвристические (по более сложным правилам выделения и анализа байтовых последовательностей внутри пакетов и иных параметров потока), связывание потоков;
- возможность идентификации некоторых протоколов и сервисов по первому пакету (технология FPC/FPI и др. в терминах вендоров аналогичных решений);
- возможность обновления набора признаков и правил классификации, в том числе динамического – без прерывания выполнения пользовательского приложения (наборы признаков и правил постоянно актуализируются и могут быть предоставлены пользователю);

- возможность кастомизации – регистрации собственных классов трафика (протоколов, сервисов), добавления признаков в имеющиеся классы, разработки моделей, способов и алгоритмов классификации «на заказ» по запросу;
- возможность обработки трафика в одном направлении для повышения скорости классификации;
- возможность управления количеством запускаемых экземпляров для оптимизации производительности.

1.3 Состав

1.3.1 Логическая структура

ПК «TRACE TA» состоит из следующих основных компонентов:

- движок (engine) – отвечает за общие функции ПК «TRACE TA»: взаимодействие с пользовательским приложением, управление бандлом;
- бандл (bundle) – совокупность обработчиков отдельных протоколов и сервисов, реализует всю логику классификации потоков, извлечения и обогащения метаданных (с использованием ассета);
- ассет (asset) – набор признаков, используемый бандлом (вынесен для удобства в отдельный файл);
- «Прототека» – электронная справочная система, содержащая описание всех обработчиков протоколов и сервисов (особенности распознавания, метаданные и настройки).

1.3.2 Дистрибутив

Состав файлов:

- [wrdp_engine_api.h](#) – заголовочный файл, описывающий интерфейс (API) между движком и пользовательским приложением в части общих функций, типов, констант;
- [wrdp_bundle_api.h](#) – заголовочный файл, описывающий интерфейс (API) между движком и пользовательским приложением в части констант для конкретных протоколов и сервисов, реализованных в бандле;
- [libwrdpengine.so.X1.Y1.Z1](#) – динамическая библиотека движка, где X1.Y1.Z1 – три числа, описывающие версию движка;
- [libwrdpbundle.so.X2.Y2.Z2](#) – динамическая библиотека бандла, где X2.Y2.Z2 – три числа, описывающие версию бандла;
- [wrdpbundle.asset](#) – файл ассета;
- [prototeka_ГГГГ-ММ-ДД.zip](#) – архив с «Прототекой»;
- [documents_ГГГГ-ММ-ДД.zip](#) – архив с документацией (PDF).

1.4 Входные данные

На вход ПК «TRACE TA» подаются:

- управляющие воздействия (настройки);
- сетевые потоки (flow) – последовательности пакетов IPv4 или IPv6 с одним и тем же набором (5-tuple) значений «IP-адрес клиента, IP-адрес сервера, порт клиента, порт сервера, сетевой протокол» после удаления заголовков канального, сетевого и транспортного уровней (полезная нагрузка TCP- или UDP-дейтаграмм, payload). В некоторых случаях для распознавания потока требуется анализировать связанные с ним потоки, поэтому на вход рекомендуется подавать все IP-потоки одного абонента.

Для каждого входящего пакета (payload) должны также подаваться:

- 5-tuple;
- временная метка получения пакета (timestamp);
- направление передачи пакета (от клиента серверу, от сервера клиенту).

Перечисленные данные подаются на вход путём вызова функций через API между движком и пользовательским приложением.

1.5 Выходные данные

На выходе ПК «TRACE TA» формируются:

- результат классификации для каждого сетевого потока – путь классификации (classification path), который представляет собой кортеж из числовых идентификаторов всех обнаруженных в потоке протоколов и сервисов, упорядоченный слева направо от нижележащего к вышележащему уровню в смысле модели OSI (стек протоколов). Например, путь «3.81.205.199.54» интерпретируется как «base.ip4.tcp.ssl.google» и означает, что в потоке распознаны протоколы IPv4, TCP, SSL (TLS) и сервис Google (первый идентификатор «3», то есть «base» – служебный). Соответствие между числовыми идентификаторами и именами протоколов и сервисов описано в «Прототеке»;

- метаданные (атрибуты) – дополнительные данные, формируемые ПК «TRACE TA» в ходе анализа и классификации потоков. Могут быть двух видов: извлекаемые непосредственно из тела пакета (например, SNI – «example.test.ru») и вычисляемые (например, результат определения типа услуги – «Audio/Video»). В «Прототеке» для каждого протокола (сервиса) приведено описание метаданных, формируемых по этому протоколу (сервису);

- статистика обработки потоков в рамках сеанса выполнения программы (например, перечень всех обнаруженных протоколов и сервисов с указанием для каждого из них количества потоков, в которых он был обнаружен).

Перечисленные данные формируются путём вызова функций через API между движком и пользовательским приложением.

Кроме того, ПК «TRACE TA» ведёт журнал, куда сохраняются сообщения о ходе работы с учётом заданного уровня подробности (см. параметр движка [log_level](#) в файле [wrdp_engine_api.h](#)). Сообщения передаются в стандартный поток вывода (stdout) процесса пользовательского приложения и параллельно могут сохраняться в файл (см. параметр

движка `no_log_file` в файле `wrpd_engine_api.h`). Имя файла журнала имеет формат `<executable_file_name>.<date>.<time>.<network_name>.<pid>.log.txt`, где:

- `executable_file_name` – имя исполняемого файла пользовательского приложения;
- `date` – дата создания файла в формате ГГММДД;
- `time` – время создания файла в формате ЧЧММСС;
- `network_name` – сетевое имя ЭВМ, где запущено пользовательское приложение;
- `pid` – PID процесса, представляющего пользовательское приложение.

Файл журнала по умолчанию создаётся в каталоге `./<process_name>.logs`, где `process_name` – имя исполняемого файла процесса, представляющего пользовательское приложение. При создании файла проверяется наличие одноимённого файла (если есть, то сообщения пишутся в имеющийся), при этом файлы с разницей в 1 секунду (в поле `<time>`) считаются «одноимёнными». Путь по умолчанию может быть изменён (см. параметр движка `log_dir` в файле `wrpd_engine_api.h`). Внутри указанного каталога создаются подкаталоги с именами вида `<date>` (дата создания в формате ГГММДД). Каталоги и файлы старше 1 месяца удаляются.

1.6 Выполнение программы

1.6.1 Общий порядок выполнения

Пользовательское приложение с помощью API движка загружает бандл, подаёт ему на вход потоки (одновременно множество) и принимает обратно результаты их обработки. Бандл выполняет обработку и классификацию потоков, при необходимости опираясь на признаки в ассете.

Для каждого входного потока бандл получает от пользовательского приложения последовательно по одному пакету (со снятыми заголовками канального, сетевого и транспортного уровней), а по результатам его обработки ожидает новый пакет этого потока или сообщает пользовательскому приложению о том, что классификация завершена и поток можно закрыть. Когда очередной пакет распознан, он может быть использован для извлечения и/или вычисления метаданных (только для некоторых протоколов при задании соответствующих настроек).

В зависимости от того, какой протокол был обнаружен, бандл либо считает поток полностью распознанным, либо продолжает распознавание. Например, при обнаружении TLS и/или HTTP будет продолжен поиск вышележащего сервиса, который (в случае обнаружения) и будет считаться конечным в пути классификации. Пока бандл не примет решение о полном распознавании потока, он не сообщит пользовательскому приложению о том, что поток следует закрыть, поэтому критерий закрытия не полностью распознанного потока должен быть (при необходимости) реализован на стороне пользовательского приложения.

1.6.2 Модель выполнения и основные объекты API

Поддерживается многопроцессорность (обработка может выполняться несколькими ядрами) и многопоточность (обработка может выполняться в нескольких потоках внутри одного процесса, совместно использующих глобальные контексты). Управление потоками и процессами должно обеспечиваться пользовательским приложением.

Основные объекты представлены как непрозрачные структуры, доступ к которым осуществляется с помощью функций API. Объекты и взаимосвязь между ними показаны на рисунке 1.1.

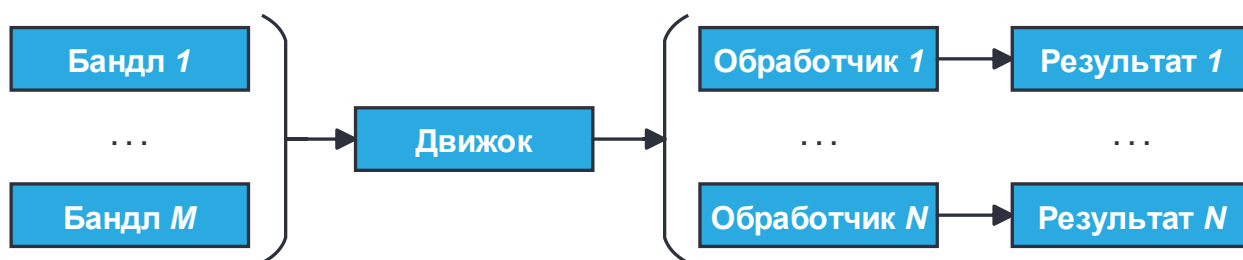


Рисунок 1.1 – Взаимосвязи основных объектов API

Структура `struct dpi_engine` описывает экземпляр движка, связанную с ним конфигурацию и ресурсы. Пользовательское приложение может создавать множество экземпляров движка.

Структура `struct dpi_bundle` описывает экземпляр бандла и связанную с ним конфигурацию. Движок может запускать несколько экземпляров бандла, но в любой момент времени активным может быть только один из них. Поток от пользовательского приложения подается движком только на активный экземпляр. Переключение активного бандла не требует перезапуска движка. Таким образом, можно обновлять бандл (и ассет) без перезапуска пользовательского приложения – путём обновления соответствующих файлов и последующего вызова и активации нового экземпляра бандла. При этом ранее поданные на вход потоки продолжают обрабатываться на старом экземпляре бандла, все новые потоки подаются на вновь подключённый экземпляр, а старый выгружается сразу после закрытия всех поданных на него потоков.

Структура `struct dpi_worker` описывает объект «обработчик» (worker). Это поток, в котором выполняется обработка входного трафика. Движок может создавать несколько обработчиков. Экземпляр бандла может запускаться на одном или нескольких обработчиках, при этом один входной сетевой поток (flow) должен подаваться только на один обработчик. Обработчику обычно соответствует поток выполнения (thread of execution) в ОС. Обработчик не должен использоваться более чем одним потоком выполнения одновременно.

Структура `struct dpi_flow` описывает объект «контекст потока» (flow context). Она предназначена для хранения всех данных, необходимых для обработки потока, таких как идентификатор потока (5-tuple, набор значений «IP-адрес источника, IP-адрес получателя, порт источника, порт получателя, сетевой протокол»), состояние потока (например, «создан», «к закрытию»). Жизненным циклом контекста потока (создание, обновление и освобождение контекста потока) управляет пользовательское приложение.

Структура `struct dpi_result` описывает объект «результат». В ней хранятся результаты классификации и метаданные.

1.6.3 Включение и отключение обработчиков протоколов и атрибутов

Распознавание протоколов/сервисов и извлечение/вычисление метаданных (атрибутов), формируемых по этим протоколам/сервисам, осуществляют отдельные компоненты в составе бандла – обработчики протоколов (signature).

Обработчики протоколов и атрибуты могут быть включены или отключены.

Если обработчик протокола включён, то соответствующий протокол/сервис может быть распознан, то есть его идентификатор может быть помещён в результат обработки потока (в путь классификации) для передачи в пользовательское приложение. Кроме того, этот обработчик может передать управление другим обработчикам (вышележащих протоколов/сервисов) для уточнения пути классификации. Также включённый обработчик протокола способен формировать атрибуты.

Если обработчик протокола отключён, то он реализует следующее поведение:

- BASE, IP4, IP6 – по-прежнему распознаются и сами эти протоколы, и зависимые (вышележащие) протоколы/сервисы (то есть при отключении поведение не меняется);
- TCP, UDP, SCTP – по-прежнему распознаются сами эти протоколы, но не распознаются зависимые (вышележащие) протоколы/сервисы. Например, если отключить обработчик TCP, то в потоке TCP не будет распознан ни протокол SSL, ни любые сервисы над SSL. В журнал выводятся сообщения уровня подробности WARNING с текстом *Absence of protocol "name"*, где *name* может принимать значения tcp, udp, sctp;
- все, кроме вышеперечисленных – не выполняются никакие реализованные в обработчике алгоритмы. В частности, не распознаются ни сами эти протоколы/сервисы, ни зависимые (вышележащие) протоколы/сервисы (например, если отключить обработчик SSL, то не будет распознан ни протокол SSL, ни любые сервисы над SSL), не извлекаются/вычисляются атрибуты протокола/сервиса.

Если атрибут включён, то значение атрибута может быть помещено в результат обработки для передачи в пользовательское приложение (для этого должен быть включён и обработчик протокола, к которому относится атрибут).

Если атрибут отключён, то значение атрибута не будет помещено в результат обработки для передачи в пользовательское приложение. При этом, если значение атрибута используется в каком-либо алгоритме, то для работы этого алгоритма оно будет извлечено/вычислено независимо от того, включён атрибут или отключён (например, если отключить атрибут **DPI_ATTR_SERVER_NAME** в обработчике SSL, то распознавание по SNI всё равно будет работать).

По умолчанию все обработчики протоколов и все их атрибуты отключены.

При необходимости состав включённых обработчиков протоколов и/или атрибутов может быть изменён. Включение/отключение обработчиков протоколов и атрибутов выполняется после загрузки бандла, но до его активации.

Включение/отключение любого обработчика протокола или атрибута производится независимо от остальных (в частности, включение обработчика протокола не включает его атрибуты, отключение обработчика протокола не отключает его атрибуты). А именно, после

вызова функции `dpi_bundle_create_from_file` формируется дерево обработчиков протоколов и относящихся к ним атрибутов, у каждого узла дерева есть свойство `enabled` (включён), и изменение свойства `enabled` у любого узла не меняет это свойство у дочерних и родительских узлов.

Для включения/отключения обработчиков протоколов используются функции:

- `dpi_bundle_signature_enable_all` – включает все обработчики;
- `dpi_bundle_signature_disable_all` – отключает все обработчики;
- `dpi_bundle_signature_enable` – включает заданный обработчик;
- `dpi_bundle_signature_disable` – отключает заданный обработчик;
- `dpi_bundle_signature_enable_by_family` – включает все обработчики, относящиеся

к заданному семейству;

- `dpi_bundle_signature_disable_by_family` – отключает все обработчики, относящи-

еся к заданному семейству;

- `dpi_bundle_signature_enable_by_tags` – включает все обработчики, имеющие за-

данный тег;

- `dpi_bundle_signature_disable_by_tags` – отключает все обработчики, имеющие за-

данный тег.

Для выяснения текущего состояния обработчика протокола (включён или отключён), используется функция `dpi_bundle_signature_is_enabled`.

Для включения/отключения атрибутов используются функции:

- `dpi_bundle_attr_register` – включает заданный атрибут заданного обработчика;
- `dpi_bundle_attr_unregister` – отключает заданный атрибут заданного обработчика.

1.6.4 Реакция на системные сигналы

Движок назначает свой обработчик системных сигналов SIGABRT, SIGBUS, SIGFPE, SIGILL, SIGSEGV при вызове функции `dpi_engine_create`.

Этот обработчик получает GNU backtrace и записывает его вывод (краткая трассировка) в журнал, затем вызывает процесс GDB для получения backtrace и записывает его вывод (расширенная трассировка) в журнал, после чего система формирует дамп памяти процесса и завершает его работу. Сообщения трассировки записываются на уровне подробности журнала ERR и выше (то есть любым, кроме NON).

Если пользовательское приложение назначит свои обработчики указанных сигналов после вызова `dpi_engine_create`, то движок не получит эти сигналы.

Назначенный движком обработчик никогда не снимается (предполагается, что движок никогда не выгружается). Это может привести к ошибке адресации, например, если библиотека движка была загружена динамически через `dlopen` и выгружена через `dlclose` без завершения работы основного процесса.

1.6.5 Пользовательские признаки и обработчики протоколов

Одним из основных методов распознавания протоколов/сервисов является поиск признаков типа «доменное имя» (хранятся в ассете) в обрабатываемых пакетах (см. в «Протоотеке» метод «По заголовкам и адресам»), что позволяет определить сервис над HTTP, TLS/SSL, QUIC. Обработчики протоколов ищут указанные признаки в следующих областях:

Обработчик	Область поиска
HTTP	Заголовок Host
SSL	Поля commonName и SAN в разделе Certificate сообщения ServerHello
	Поля с Server Name в разделе SNI сообщения ClientHello
QUIC	

Движок позволяет добавлять собственные (пользовательские) признаки типа «доменное имя» в имеющиеся обработчики протоколов, а также регистрировать пользовательские обработчики протоколов на базе пользовательских признаков указанного типа. Для этого применяется функция `dpi_bundle_signature_upper_add` (см. описание в заголовочном файле `wrdp_engine_api.h`).

Функцию следует вызывать до вызова `dpi_bundle_activate`. Функцию можно вызывать любое количество раз.

Один вызов функции добавляет один признак типа «доменное имя» (далее в этом пункте – просто «признак») в один обработчик протокола (далее в этом пункте – просто «протокол»). Если протокола нет, то он регистрируется. Повторный вызов для того же протокола добавляет очередной признак.

Аргумент `name` задаёт краткое название протокола (см. справочник полей в «Протоотеке»), значение должно быть уникальным (оно определяет протокол). Краткие названия и прочие параметры изначально зарегистрированных протоколов есть в «Протоотеке».

Аргумент `id` задаёт ID протокола. Для протокола с новым `name` значение `id` должно быть либо 0 (ID автоматически назначается бандлом из диапазона от 20000 до 29999 и возвращается функцией), либо уникальным из диапазона от 30000 до 39999; для уже зарегистрированного протокола (изначально или пользователем) `id` должен соответствовать известному `name`; в противном случае функция вернёт `DPI_BAD_PARAM`.

Аргументы `long_name`, `family` и `tags` для протокола с новым `name` задают соответственно название, семейство и теги протокола. Для уже зарегистрированного протокола (изначально или пользователем) значения этих аргументов игнорируются, то есть нельзя переопределить их с помощью функции `dpi_bundle_signature_upper_add` (но `family` и `tags` можно переопределить функциями `dpi_signature_family_set` и `dpi_signature_tags_set`). Значением `long_name` и `family` может быть пустая строка, значением `tags` может быть 0 (битовая маска из нулей – нет тегов).

Аргумент `pattern` задаёт добавляемый пользовательский признак (при нахождении которого поток будет распознан как `name`). Признак представляет собой подстроку доменного имени и может содержать только Punycode-символы и метасимволы (см. ниже).

Punycode-символы:

- буквы от «a» до «z» (без различия между прописными и строчными);
- цифры от «0» до «9»;
- дефис «-»;
- точка (не должно быть двух точек подряд).

Если в признаке нет метасимволов, то проверяется точное его совпадение с содержимым области поиска. Например, признак «book.com» сработает, только когда в области поиска содержится именно «book.com», других символов нет.

Метасимвол «звёздочка» («»)* может стоять в начале и конце признака. Этот метасимвол определяет, что соответственно перед и/или после содержимого признака возможно наличие других символов внутри области поиска. Варианты написания и интерпретации признака с метасимволом «*»:

Вариант	Пример признака	Пример области поиска
«*» в начале	*book.com	i364.book.com
«*» в конце	book.com*	book.com.tr
«*» в начале и конце	*book.com*	i364.book.com.tr

Метасимволы повторения означают повторение заданного количества любых символов. Метасимволы повторения не должны соприкасаться с метасимволом «*». Количество повторений должно быть в диапазоне от 0 до 64535. Варианты написания и интерпретации метасимволов повторения:

Вариант	Число повторений	Пример признака	Примеры области поиска
{n}	Ровно n раз	i{3}.book.com	i364.book.com
{m,n}	От m до n	i{2,3}.book.com	i36.book.com i364.book.com
{m,}	От m и более	i{5,}.book.com	i36456789s00.book.com
{,n}	Не более n	i{,3}.book.com	i.book.com i3.book.com i36.book.com i364.book.com

Поиск осуществляется одновременно по ассету и пользовательским признакам.

Ниже описаны ситуации с конфликтом признаков (значению в области поиска соответствует более чем один признак, добавленный в разные протоколы), когда возможно нежелательное поведение программы и снижение качества распознавания.

Первая ситуация – если значение в области поиска совпадает более чем с одним признаком (точное совпадение, без метасимволов). В ассете дублирования признаков нет, поэтому такое возможно в двух случаях:

- пользователь добавил признак, который уже есть в ассете – в этом случае сработает признак в ассете. Например, пусть в ассете признак «book.com» соответствует протоколу XXX, а пользователь через функцию [dpi_bundle_signature_upper_add](#) добавил признак «book.com» в протокол YYY – тогда, если в потоке найдено значение «book.com» (без символов слева или справа), то поток будет распознан как XXX;

– пользователь сам добавил один и тот же признак в разные протоколы – в этом случае может сработать любой из них (поведение считается не определённым, результат зависит от версии бандла – например, может сработать первый добавленный признак или случайно выбранный).

Вышеописанную ситуацию с дублированием должен исключить сам пользователь.

Вторая ситуация – если значению в области поиска соответствует более чем один признак, и хотя бы один из них с метасимволами (то есть не дублирование). В этом случае срабатывает признак с наибольшей длиной совпадения. Если длины равны, то признак выбирается случайным образом. Длина совпадения рассчитывается так: а) метасимволы «*» не считаются, б) метасимволы повторения заменяются соответствующими им символами из области поиска, в) обычные символы считаются один к одному. Например, встреченному (в области поиска) значению «fan.i364.book.com» могут соответствовать признаки:

- «*.i364.*» – длина совпадения 6 (считаем все символы без «*»);
- «*.i{1,4}.*» – длина совпадения 6 (заменяем {1,4} на 364 из встреченного значения);
- «fan.i364.book.com» – длина 17;
- «*fan.i364.book.com» – длина 17.

Чтобы минимизировать вероятность некорректного распознавания в такой ситуации, надо добавлять максимально конкретные признаки, и при наличии похожих признаков в разных протоколах – стараться обеспечить для них разную длину совпадения.

2 ИСПОЛЬЗОВАНИЕ

2.1 Условия применения

2.1.1 На этапе интеграции в пользовательское приложение

Требования к программным средствам:

- ОС Linux (протестировано на Debian 11);
- компилятор C++;
- пользовательское приложение, поддерживающее API движка ([wrdp_engine_api.h](#), [wrdp_bundle_api.h](#)), в исходных кодах для интеграции файлов ПК «TRACE TA» в проект и совместной компиляции.

Требования к аппаратным средствам на этом этапе зависят от ОС и компилятора.

2.1.2 На этапе выполнения пользовательского приложения

Требования к аппаратным и программным средствам:

- IBM PC-совместимая ЭВМ или виртуальная машина;
- процессор с архитектурой x86-64;
- ОС Linux (протестировано на Debian 11, Ubuntu 22);
- объём оперативной памяти порядка нескольких Гбайт: не менее 0,5 Гбайт на общие структуры, не менее 0,5 Кбайт на каждый сетевой поток;
- объём свободного дискового пространства для файлов не менее 200 Мбайт.

Прочие требования на этом этапе зависят от пользовательского приложения и сценария его применения. Например, для обработки реального трафика необходим сетевой интерфейс для захвата трафика пользовательским приложением.

2.2 Подготовка к работе

2.2.1 Получение (обновление) дистрибутива

В настоящее время дистрибутив предоставляется только по запросу.

Состав файлов – по п. 1.3.2. Порядок получения – по согласованию.

После получения дистрибутива необходимо для каждого из файлов [libwrdpengine.so.X1.Y1.Z1](#) и [libwrdpengine.so.X2.Y2.Z2](#) создать:

- файл [.so.X](#) – символическую ссылку на [.so.X.Y.Z](#). При установке программы из deb-пакетов этот файл создаётся автоматически утилитой `ldconfig` в ходе установки;
- файл [.so](#) – символическую ссылку на [.so.X](#). При установке программы из deb-пакетов этот файл создаётся автоматически утилитой `ldconfig` в ходе установки пакета `-dev` (пакета с соответствующим заголовочным файлом [.h](#)).

2.2.2 Интеграция в пользовательское приложение

Для добавления функциональности анализа и классификации сетевого пакетного трафика в пользовательское приложение необходимо:

- 1) добавить в исходный код пользовательского приложения вызовы функций движка в соответствии с API и требованиями к логике работы. Описание API движка приведено в п. 1.6 и в комментариях к исходному коду в файлах [wrdp_engine_api.h](#), [wrdp_bundle_api.h](#);
- 2) добавить в исходный код системы сборки пользовательского приложения требование к компоновщику использовать библиотеку `libwrdpengine` (если не используется динамическая загрузка библиотеки);
- 3) заголовочные файлы ([wrdp_engine_api.h](#), [wrdp_bundle_api.h](#)) поместить в каталог `\include\` (или иной каталог с файлами для включения в программу, заданный в настройках компилятора);
- 4) движок (файлы [libwrdpengine.so](#), [libwrdpengine.so.X1](#), [libwrdpengine.so.X1.Y1.Z1](#)) поместить в каталог, видимый компоновщиком динамических библиотек (можно путь к каталогу с этим файлом добавить в опции компоновщика);
- 5) выполнить компиляцию пользовательского приложения.

2.2.3 Развёртывание пользовательского приложения

До запуска пользовательского приложения необходимо:

- 1) движок (файлы [libwrdpengine.so](#), [libwrdpengine.so.X1](#), [libwrdpengine.so.X1.Y1.Z1](#)) поместить в каталог, видимый загрузчиком динамических библиотек (можно путь к каталогу с этим файлом добавить в системную переменную `LD_LIBRARY_PATH`);
- 2) бандл (файлы [libwrdpbundle.so](#), [libwrdpbundle.so.X2](#), [libwrdpbundle.so.X2.Y2.Z2](#)) поместить в каталог, путь к которому задан в параметрах функции создания экземпляра бандла [dpi_bundle_create_from_file](#) (в исходном коде пользовательского приложения);
- 3) ассет ([wrdpbundle.asset](#)) поместить в тот же каталог, где находится бандл.

2.3 Работа с программой

2.3.1 Основная функциональность

Функциональность ПК «TRACE TA» задействуется во время выполнения пользовательского приложения в соответствии с алгоритмом работы последнего.

2.3.2 Справочная система «Прототека»

Для использования «Прототеки» необходимо распаковать содержимое архива [prototeka_ГГГГ-ММ-ДД.zip](#) (будет извлечён каталог `\prototeka_ГГГГ-ММ-ДД\`) и открыть в веб-браузере файл [index.html](#) из этого каталога. На экране появится домашняя страница справочника (рисунок 2.1).

Справочник имеет типовой графический веб-интерфейс. В левой части экрана размещена панель навигации по страницам. В основной части экрана отображается текущая страница.

На домашней странице отображаются примечания к текущему и предыдущим выпускам «Прототеки» (журнал изменений).

Для просмотра основной таблицы протоколов и сервисов необходимо перейти на страницу *Обзор* → *Протоколы*.

Для просмотра описаний полей основной таблицы необходимо перейти на страницу *Справочники* → *Поля*.

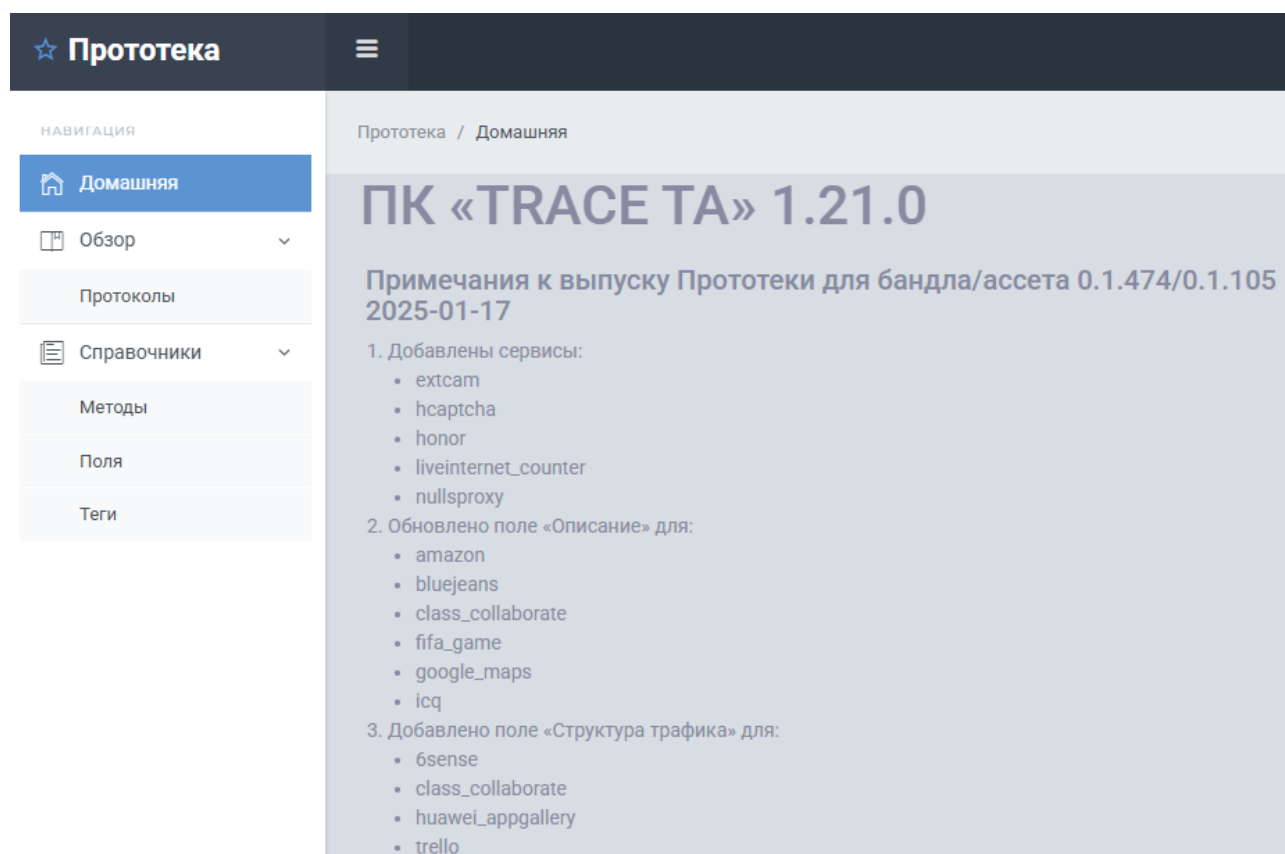


Рисунок 2.1 – Домашняя страница справочной системы «Прототека»

Для открытия страницы с карточкой (детализированным описанием) протокола необходимо дважды щёлкнуть ЛКМ в любом месте соответствующей строки основной таблицы.

Над основной таблицей справа есть поле ввода и радиокнопки «Поиск» (рисунок 2.2), предназначенные для фильтрации таблицы – отображаются только те записи, у которых в выбранной (с помощью радиокнопки) области поиска есть значение, указанное в поле ввода (регистр не имеет значения). Области поиска:

– «таблица» (выбрано по умолчанию) – только в полях карточки протокола, видимых в основной таблице. Примеры применения: найти протокол STUN или отобразить список протоколов и сервисов, относящихся к семейству Video Streaming;

- «структура трафика» – только в именах классов в поле «Структура трафика». Пример применения: отобразить список сервисов, в структуре трафика которых встречается протокол STUN;
- «остальное» – во всём остальном тексте карточки протокола. Пример применения: отобразить список протоколов и сервисов, распознаваемых методом «По порту».

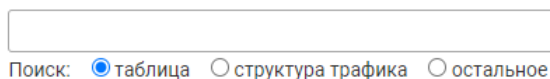


Рисунок 2.2 – Инструмент фильтрации таблицы по тексту в карточке протокола

Над основной таблицей также есть разворачивающиеся блоки «Семейства» и «Теги», в которых можно выбрать семейство и/или теги для дополнительной фильтрации таблицы.

Над основной таблицей слева отображается количество записей, удовлетворяющих всем применённым фильтрам (рисунок 2.3).

Там же есть кнопки «.XLSX» и «.TXT», позволяющие выгрузить такие записи в файл. XLSX-файл имеет формат Office Open XML, удобный для открытия в Microsoft Excel (2007 и новее) и аналогичных программах. TXT-файл имеет формат TSV (Tab Separated Values), кодировку UTF-8.



Рисунок 2.3 – Информация о результате фильтрации (пример)
и кнопки выгрузки отфильтрованных записей в файл

ПРИЛОЖЕНИЕ 1.

Перечень поддерживаемых протоколов и сервисов

Всего поддерживается 510 наименований классов, из них:

- 54 протокола;
- 455 сервисов;
- 1 служебный.

Их подробное описание приведено в «Прототеке».

Протоколы:

- | | |
|------------------------------|-----------------|
| – Apple HTTP Live Streaming; | – NNTPS; |
| – Apple Update; | – NSPI; |
| – BitTorrent; | – NTP; |
| – CAPWAP; | – POP3; |
| – CoAP; | – POP3S; |
| – DCE/RPC; | – PPTP; |
| – DHCP; | – QUIC; |
| – DHCPv6; | – RDP; |
| – DNS; | – RTCP; |
| – FTP (control); | – RTP; |
| – FTP (data); | – SCTP; |
| – HTTP; | – SIP; |
| – IMAP; | – SMTP; |
| – IMAPS; | – SMTPS; |
| – IPv4; | – SNMP; |
| – IPv6; | – SOCKS4; |
| – ISAKMP; | – SOCKS5; |
| – L2TP; | – SSDP; |
| – LLMNR; | – SSH; |
| – MAPI; | – SSL (TLS); |
| – mDNS; | – STUN; |
| – MQTT; | – syslog; |
| – MSNP; | – TCP; |
| – NBDS; | – Teredo; |
| – NBNS; | – UDP; |
| – NBSS; | – WireGuard; |
| – NNTP; | – WS-Discovery. |

Сервисы:

- | | |
|---------|---------------|
| – 1C; | – NullsProxy; |
| – 2ГИС; | – Nvidia; |

- 6sense;
- 24TB;
- Adobe Advertising Cloud;
- AccuWeather;
- Activision Blizzard;
- Adform;
- AdGuard;
- Adjust;
- Adobe;
- AdRiver;
- Adserving;
- Adult content;
- Advanced Hosting;
- AiData;
- Akamai Technologies;
- Alibaba Group;
- Alibaba;
- AliExpress;
- Amazon Ads;
- Amazon Chime;
- Amazon Prime Video;
- Amazon Web Services;
- Amazon;
- AmberData;
- AMG Radio;
- Amplitude;
- Angry Birds;
- any (Diginetica);
- AnyDesk;
- APNS;
- Apple;
- Apple Account;
- App Store;
- Apple Maps;
- Apple Music;
- Apple TV+;
- AppLovin;
- AppsFlyer;
- Archive.org;
- AstraLab;
- AtData;
- Okko;
- OneTrust;
- OpenStreetMap Foundation;
- OpenAI;
- OpenVPN;
- Opera;
- Oracle;
- Oracle Advertising;
- Oracle Cloud Infrastructure;
- OVHcloud;
- Ozon;
- Pangle;
- Patreon;
- PayPal;
- Peers.TV;
- Pinduoduo;
- Pinterest;
- Platforma;
- Playrix;
- PlayStation;
- Pokemon GO;
- Pornhub;
- PREMIER;
- Proton;
- Proton Mail;
- Proton VPN;
- Psiphon;
- PUBG: Battlegrounds;
- PubNub;
- Qiwi;
- Quran.Foundation;
- Radmin;
- Reddit;
- Roblox;
- RuMarket;
- RuStore;
- RUTUBE;
- Salesforce;
- Samsung;
- Sape;
- SberAds;

- Baidu;
- Between Exchange;
- Bidease;
- BidMachine
- Bigo;
- Bing;
- BlueJeans;
- Branch;
- Brawl Stars;
- Braze;
- Browsec VPN;
- ByteDance;
- Call of Duty;
- Candy Crush Saga;
- Canva;
- CDN77;
- CDNvideo;
- ChatGPT;
- Chi Gap;
- CHILL;
- CityAds Media;
- Clash of Clans;
- Clash Royale;
- Class for Web;
- Cloud.ru;
- Cloudflare;
- CloudPayments;
- Clubhouse;
- Comscore;
- Contentful;
- Counter-Strike;
- Criteo;
- CyberGhost VPN;
- Deezer;
- DepositFiles;
- DigiCert;
- Digital Turbine;
- DigitalOcean;
- DigitalOne;
- Discord;
- Disney+;
- SberDevices;
- Scaleway;
- Selectel;
- Sentry;
- Servers.ru;
- Sharethrough;
- Shazam;
- Shopify;
- Signal;
- Simpli.fi;
- Sina Weibo;
- Sinch;
- Skype;
- Slack;
- Snapchat;
- Snow;
- Softonic;
- Soloway;
- Solta;
- SoundCloud;
- SPB TV;
- Speedtest;
- Spotify;
- START;
- Steam;
- Streamerace;
- Stripe;
- Supercell;
- T-ID;
- Taboola;
- Taobao;
- Target RTB;
- Teads;
- Tealium;
- TeamViewer;
- Telegram;
- Teleport Media;
- Tencent Captcha;
- Tenor;
- The Settlers Online;
- The Trade Desk;

- Disqus;
- Dota 2;
- Douyin;
- Dropbox;
- EA Sports FC;
- eBay;
- EdgeЦентр;
- Edgio;
- Electronic Arts;
- Epic Games;
- Equativ;
- Evernote;
- ExtCam;
- Facebook;
- Facebook CDN;
- Facebook Video;
- FaceTime;
- Fandom;
- Fastly;
- FDC Servers;
- Federal.tv;
- Firebase Cloud Messaging;
- Firebase Crashlytics;
- First Touch Games;
- Flocktory;
- Fortnite;
- Free Fire;
- Gaijin Entertainment;
- Galaxy Store;
- Gcore;
- GeeTest;
- GeForce NOW;
- Genshin Impact;
- Getintent;
- GIPHY;
- Gismeteo;
- GitHub;
- Gmail;
- Gneздо;
- GoodGame.ru;
- Google;
- TikTok;
- Tilda Publishing;
- Tinder;
- Trello;
- Tripadvisor;
- TrustArc;
- Tumblr;
- TunnelBear VPN;
- Twilio;
- Twilio Segment;
- Twitch;
- Uber;
- Ubisoft;
- UCloud;
- Udemy;
- UFC Fight Pass;
- Unity;
- Unity Ads;
- UX Feedback;
- Viber;
- Video-mech.ru;
- Vigo;
- viju;
- Vimeo;
- VK;
- VK Analytics;
- VK ID;
- VK Видео;
- VK Музыка;
- VK Реклама;
- War Thunder;
- Wargaming.net;
- WARP;
- Wattpad;
- Webcaster Pro;
- Webex;
- Weborama;
- WeChat;
- WhatsApp;
- Wikimedia Foundation;
- Wikipedia;

- Google Account;
- Google Ads;
- Google Analytics;
- Google APIs;
- Google Chat;
- Google Cloud;
- Google Cloud Storage;
- Google Docs;
- Google Earth;
- Google Fonts;
- Google Maps;
- Google Meet;
- Google Play;
- GoTo Meeting;
- Guns of Icarus;
- HASSLE ONLINE;
- HBO;
- hCaptcha;
- HeadHunter;
- Heroes of the Storm;
- Hetzner Online;
- HeyTap;
- Hikvision;
- Hola VPN;
- Honkai: Star Rail;
- HONOR;
- Hotjar;
- Hotspot Shield;
- Huawei;
- Huawei AppGallery;
- HubSpot;
- Hulu;
- Hybrid;
- iCloud;
- ICQ;
- imo;
- InAppStory;
- Index Exchange;
- InMobi;
- Instagram;
- INVENTOS;
- Wildberries;
- Windows Update;
- Windscribe VPN;
- Wink;
- Wistia;
- WordPress;
- Workupload.com;
- World Of Tanks;
- Worldstream;
- X (Twitter);
- Xbox;
- Xiaomi;
- XNXX;
- XVideos;
- Yahoo Advertising;
- Yahoo;
- Yandex Ads;
- Yandex Cloud;
- Yandex Stream;
- Yappy;
- Yota;
- YouTube;
- YouTube Music;
- Zendesk;
- Zoom;
- Zynga;
- Авито;
- Альфа-Банк;
- Альфа-Групп;
- Амедиатека;
- Банк ВТБ;
- билайн;
- билайн тв;
- ВГТРК;
- Витрина ТВ;
- ВКонтакте;
- Газпром ID;
- Газпромбанк;
- Газпром-Медиа;
- ГИД Дата;
- Госуслуги;

- ironSource;
- iTunes;
- Java Update;
- Jivo;
- Joom;
- jQuery;
- jsDelivr;
- Kimberlite.io;
- Kinescope;
- King;
- KION;
- KLIPY;
- Kuaishou;
- Lamoda;
- Lazada;
- League of Legends;
- Liftoff;
- Line;
- LinkedIn;
- Liteapks.com;
- LiveInternet counter;
- LovePlanet;
- Mail.ru;
- Mapbox;
- MASTERTEL;
- MAX;
- MediaFire;
- MediaHills;
- Mediascope;
- MEGA;
- Melbicom;
- Messenger;
- Microsoft;
- Microsoft 365;
- Microsoft account;
- Microsoft Advertising;
- Microsoft OneDrive;
- Microsoft Outlook;
- Microsoft Teams;
- miHoYo;
- Mindbox;
- ГПМ Радио;
- ГПМ Реклама;
- Группа ВТБ;
- Группа компаний РБК;
- Дзен;
- Звезда;
- Звук;
- Иви;
- К2Тех;
- Карусель;
- Кинопоиск;
- Лаборатория Касперского;
- Лайм Эйч Ди;
- Магнит Маркет;
- Магнит;
- Матч ТВ;
- МегаФон;
- МЕДИА1;
- Микро-ИТ;
- Мир;
- МТС;
- МУЗ-ТВ;
- Национальная система платёжных карт;
- НТВ;
- Общественное телевидение России;
- Одноклассники;
- ОК Видео;
- ОК Музыка;
- Первый канал;
- Пикабу;
- Пятый канал;
- Радио Record;
- Райффайзен Банк;
- Рамблер/Почта;
- Рамблер/ТОП-100;
- Рамблер;
- РЕН ТВ;
- Сбер ID;
- Сбербанк;
- Сбербанк Онлайн;
- Смотрёшка;

- Minecraft;
- Miro;
- MIUI OS;
- Mixpanel;
- Mobile Legends: Bang Bang;
- Mortal Kombat;
- Movix;
- Mozilla Firefox;
- NashStore;
- Naver Blog;
- Naver Mail;
- Naver Maps;
- Naver Webtoon;
- Naver;
- Navitel;
- NDM Systems;
- NetEase Cloud Music;
- NetEase;
- Netflix;
- NGENIX;
- Nintendo;
- Совкомбанк;
- СПАС;
- СТС Медиа;
- Т-Банк;
- ТамТам;
- ТВ Центр;
- Триколор;
- Яндекс;
- Яндекс Go;
- Яндекс ID;
- Яндекс Диск;
- Яндекс Карты;
- Яндекс Маркет;
- Яндекс Мессенджер;
- Яндекс Метрика;
- Яндекс Музыка;
- Яндекс Навигатор;
- Яндекс Почта;
- Яндекс Путешествия;
- Яндекс Пэй.

Служебные:

- Базовый.

ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ

API	Application Programming Interface, программный интерфейс
CN	Common Name, общее имя (поле сертификата TLS)
DLP	Data Leak/Loss Prevention, предотвращение утечки/утери данных
DPI	Deep Packet Inspection, глубокий анализ пакетов
FPC	First Packet Classification, классификация по первому пакету
FPI	First Packet Identification, идентификация по первому пакету
FW	Firewall, межсетевой экран
IDS	Intrusion Detection System, система обнаружения вторжений
IPS	Intrusion Prevention System, система предотвращения вторжений
OSI	Open Systems Interconnection, взаимодействие открытых систем (модель)
PCEF	Policy and Charging Enforcement Function, система применения политик управления качеством и тарификации трафика
PDF	Portable Document Format, формат переносимых документов
PID	Process IDentifier, идентификатор процесса
SDK	Software Development Kit, комплект для разработки программного обеспечения
SD-WAN	Software-Defined Wide Area Network, программно-определяемая сеть
SNI	Server Name Indication, указатель имени сервера (поле сертификата TLS)
ассет	см. п. 1.3.1
бандл	см. п. 1.3.1
ГГММДД	год (2 цифры), месяц (2 цифры), день (2 цифры)
ГГГГММДД	год (4 цифры), месяц (2 цифры), день (2 цифры)
движок	см. п. 1.3.1
ЛКМ	левая кнопка мыши
ОС	операционная система
ПК	программный комплекс
ЧЧММСС	час (2 цифры), минута (2 цифры), секунда (2 цифры)
ЭВМ	электронно-вычислительная машина

ООО «ТехАргос»

127015, г. Москва, ул. Новодмитровская, 2Б, ДЦ «Дмитровский»

Телефон: +7(495) 411-90-37

Web: <https://t-argos.ru/>

E-mail: mail@t-argos.ru