



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«МИРЭА – Российский технологический университет»
РТУ МИРЭА**

Институт искусственного интеллекта
Базовая кафедра №252 – информационной безопасности

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Тема:

«Устройство и анализ алгоритмов по обходу DPI»

Студент группы ККСО-03-19

Николенко В.О.

(подпись)

Научный руководитель

Жанкевич А.О.

(подпись)

Работа представлена к защите

«__» _____ 2024 г.

Оценка:

«__»

Москва – 2024

1. СОДЕРЖАНИЕ

1. Содержание	2
2. Глоссарий	3
3. Введение	5
4. Глубокий Анализ Пакетов	6
4.1. Основные задачи DPI	6
4.2. Принцип работы DPI	7
4.3. Шаги анализа:	7
4.4. Виды DPI	7
4.5. Технические аспекты DPI	8
4.6. Подходы к обходу блокировок	9
4.7. Известные протоколы для доступа к заблокированным ресур-сам	10
4.7.1 Shadowsocks	10
4.7.2 VMess	11
4.7.3 VLESS	11
5. Заключение	13
6. Список литературы	14

2. ГЛОССАРИЙ

Давайте однозначно определим терминологию данной работы, с целью исключить недопонимания между её авторами и читателями.

DPI (англ. Deep Packet Inspection "глубокий анализ пакетов") - технология проверки сетевых пакетов по их содержимому с целью регулирования и фильтрации трафика, а также накопления статистических данных. В отличие от брандмауэров, DPI анализирует не только заголовки пакетов, но и полезную нагрузку, начиная со второго уровня модели OSI (канальный).

OSI (англ. Open Systems Interconnection "межсетевое взаимодействие открытых систем") - абстрактная модель представления одних и тех же данных, с которыми можно работать на разных уровнях, предлагаемых данной моделью: физическом, канальном, сетевом, транспортном, сеансовом, уровне представления, прикладном.

Физический Уровень - передача необработанных битов по физическим носителям (кабели, радиосигналы).

Примеры протоколов: Ethernet (физический аспект) USB, Bluetooth.

Канальный Уровень - организация надежной передачи данных между узлами одной сети; управление доступом к среде, обнаружение ошибок.

Примеры протоколов: Ethernet (кадры), Wi-Fi (IEEE 802.11), PPP (Point-to-Point Protocol).

Сетевой Уровень - маршрутизация и передача пакетов данных между различными сетями.

Примеры протоколов: IP (Internet Protocol), ICMP (Internet Control Message Protocol), IPSec (для обеспечения безопасности).

Транспортный Уровень - обеспечение надежной передачи данных, контроль ошибок, восстановление соединений (TCP/UDP).

Примеры протоколов: TCP (Transmission Control Protocol), UDP (User Datagram Protocol), SCTP (Stream Control Transmission Protocol).

Сеансовый Уровень - управление сессиями, установление, поддержание и завершение сеансов связи между приложениями.

Примеры протоколов: NetBIOS, PPTP (Point-to-Point Tunneling Protocol), RPC (Remote Procedure Call).

Уровень Представления - преобразование данных для представления их в удобном для приложений формате (шифрование, сжатие).

Примеры протоколов: SSL/TLS (для шифрования), JPEG, PNG (для изображения), ASCII, EBCDIC (кодировки данных).

Прикладной Уровень - взаимодействие с конечным пользователем через приложения (веб-браузеры, почтовые клиенты и т. д.).

Примеры протоколов: HTTP/HTTPS (веб-приложения), FTP (передача файлов), SMTP (электронная почта), DNS (система доменных имен).

Шифрование - процесс преобразования данных (*посредством таких операций как: линейных и нелинейных преобразований, наложения гаммы, и матричных преобразований*) в форму, непонятную без ключа. Используется для обеспечения конфиденциальности информации (например, в коммуникациях).

Дешифрование - обратный процесс к шифрованию, преобразующий зашифрованные данные обратно в исходный вид с помощью ключа.

Кодирование - преобразование данных в другой формат для удобства передачи или хранения (например, Base64 для передачи бинарных данных через текстовые протоколы). Кодирование не является средством защиты.

Хеширование - одностороннее преобразование данных в уникальную фиксированную строку (хеш), которое невозможно обратным образом восстановить. Используется для проверки целостности данных, паролей и цифровых подписей.

Прокси-сервер - промежуточный сервер в компьютерных сетях, выполняющий роль посредника между пользователем и целевым сервером, позволяющий клиентам как выполнять косвенные запросы к другим сетевым службам, так и получать ответы.

3. ВВЕДЕНИЕ

В наше время информационные технологии развиваются уже далеко не семимильными шагами, с каждым годом появляется все больше методов передачи, шифрования, кодирования, анализа трафика, такие технологии как: TCP, UDP, TSL, SSL, HTTP, HTTPS, различные подписи с использованием эллиптических кривых, RSA, различные способы распределения ключей между клиентом и сервером, множество протоколов аутентификации пользователя на каком-либо сервисе, все это представляет собой поток трафика.

Наша жизнь уже давно содержится в онлайн сервисах, друзья и персональные данные. Такое стечение обстоятельств побуждает экспертов по безопасности и правоохранительные органы искать новые способы расследования киберпреступлений и поиска доказательств, допустимых в суде. Также речь может идти и об ограничении доступа к некоторым ресурсам со стороны государства.

Онлайн-сервисы передают большие объемы данных по сетям связи в различных формах, среди которых наиболее распространены сетевые пакеты. Сетевые пакеты - это группы битов, которые включают данные, дополненные контрольной информацией, как правило, эти биты относятся к сетевому уровню (3 уровень) модели OSI. Они представляют собой наименьшую единицу данных в определенный момент времени, перехваченную и зарегистрированную в сетевом трафике, проходящем через сети с коммутацией пакетов, состоящем из управляющей информации (IP-адрес источника и назначения, коды обнаружения ошибок) и полезной нагрузки (само сообщение). Единица данных на канальном уровне (2 уровень) модели OSI называется кадром, который представляет собой группу битов, включающую данные с одним или несколькими адресами и другую служебную информацию протокола. На транспортном уровне (4 уровне) модели OSI эквивалент называется сегментом (или дейтаграммой).

В этой работе мы будем использовать термин "анализ пакетов независимо от того, является ли фактическое содержимое кадром, пакетом, датаграммой или сеансом, если не указано иное, то есть будем рассматривать любую информацию, передаваемую по сети.

4. ГЛУБОКИЙ АНАЛИЗ ПАКЕТОВ

Связь между сетевыми устройствами осуществляется с помощью протоколов, то есть механизмов идентификации и установления соединений, а также правил и соглашений форматирования, указанных для передачи данных. Сетевые данные можно анализировать, а сетевой трафик разделять по типу, используя специальное программное обеспечение, именно такую задачу решают программы по глубокому анализу трафика.

DPI (англ. Deep Packet Inspection, глубокий анализ пакетов) - это метод анализа сетевого трафика на более глубоком уровне, чем базовая маршрутизация и фильтрация на уровне заголовков пакетов. В отличие от простого фильтрации на уровне сети (сетевой уровень), которое оперирует заголовками IP-пакетов и портами, DPI выходит за рамки информации заголовка пакета и анализирует также полезную нагрузку пакета.

Также DPI является основообразующей системой для контроля доступа к запрещённым ресурсам на территории конкретного государства. Таким образом достигается либо замедление, либо полная блокировка конкретного ресурса. В данный момент, лидером в области ограничения доступа через национальных провайдеров к неудобным ресурсам является Китай.

4.1. Основные задачи DPI

1. Обеспечение безопасности.

DPI активно используется в системах защиты, таких как межсетевые экраны (firewall), системы обнаружения и предотвращения вторжений (IDS/IPS). Он позволяет выявлять и блокировать вредоносный трафик, скрытый под стандартными протоколами, такими как HTTP, SMTP и др.

2. Управление трафиком (QoS).

Операторы сетей и интернет-провайдеры применяют DPI для регулирования пропускной способности (Bandwidth Management) и для реализации политики качества обслуживания (QoS). DPI может различать трафик разных приложений и динамически изменять приоритет обработки, например, снижать приоритет P2P-программ в пользу VoIP или стримингового видео.

3. Анализ пользовательского поведения.

Некоторые интернет-провайдеры используют DPI для анализа трафика клиентов с целью таргетирования рекламы, анализа тенденций или контроля за соблюдением условий использования сети.

4. Обеспечение сетевой целостности.

DPI позволяет обнаруживать различные аномалии в трафике, такие как нетипичные паттерны данных, что помогает предотвращать возможные сетевые

сбои или атаки.

5. Расследование киберинцидентов.

DPI позволяет внедрить абсолютно любой функционал, включая логирование действий пользователей внутри локальной сети, сохранение каких-либо метаданных о всех сетевой активности внутри локальной сети, что сильно помогает в расследовании кибер-инцидентов.

4.2. Принцип работы DPI

DPI работает на 7 уровне модели OSI, анализируя заголовки и полезную нагрузку (payload) пакетов для извлечения и интерпретации их содержания. Это позволяет идентифицировать тип трафика и приложение, генерирующее этот трафик, даже если используются стандартные порты (например, обнаружить трафик торрент-клиента, работающего на порте HTTP).

4.3. Шаги анализа:

1. Захват пакета.

Сетевой трафик поступает на анализ через устройство или программу, выполняющую DPI.

2. Распаковка пакетов.

DPI извлекает полезную нагрузку из пакета. При необходимости декодируются протоколы и форматируется трафик для дальнейшего анализа.

3. Сопоставление с сигнатурами.

Содержимое пакетов сверяется с базой сигнатур для выявления известных приложений, протоколов и угроз (например, сигнатуры вирусов, нежелательных приложений, или специфичных атак).

4. Применение политики.

В зависимости от настроек политики, DPI может либо блокировать подозрительный трафик, либо допускать его в сеть, либо изменить приоритет этого трафика.

4.4. Виды DPI

1. Статический DPI.

Этот метод основан на сопоставлении данных пакетов с заранее известными сигнатурами или паттернами. Он позволяет обнаруживать известные приложения и угрозы, но неэффективен против новых или измененных атак, для которых сигнатуры не созданы.

2. Динамический DPI.

В отличие от статического, динамический анализ включает поведенческую

аналитику, оценивая аномальные паттерны трафика и его поведение. Это позволяет обнаруживать новые угрозы, даже если сигнатуры для них отсутствуют. Однако динамический анализ требует больше вычислительных ресурсов.

3. Полный DPI (Full DPI).

Этот тип анализа полностью сканирует не только заголовки, но и весь контент пакетов. Используется для самых точных и глубоких исследований трафика, однако является самым ресурсоёмким вариантом.

4. Инкрементальный DPI.

Позволяет анализировать трафик выборочно, например, только часть пакетов из потока или только определённые виды трафика. Это снижает нагрузку на сеть и ресурсы, но ограничивает детальность анализа.

4.5. Технические аспекты DPI

1. Пропускная способность.

DPI требует значительных вычислительных ресурсов для анализа больших объёмов данных на высоких скоростях. Чем выше скорость обработки трафика, тем сложнее обеспечивать точный анализ, особенно на сетях с высокой пропускной способностью (например, на магистральных сетях Tier 1).

2. Проблемы приватности.

Использование DPI вызывает беспокойство с точки зрения конфиденциальности, так как оно позволяет анализировать личные данные и взаимодействие пользователей в сети, потенциально нарушая приватность.

3. Шифрование.

Зачастую пакеты защищены сквозным шифрованием по протоколам: SSL/TLS, SSH, AES. DPI неэффективен против такого трафика. Для анализа зашифрованных данных DPI необходимо иметь возможность расшифровки или доступ к протоколам, таким как SSL/TLS. Для этого могут использоваться подходы типа Man-in-the-Middle (MITM) для SSL, но они сопровождаются значительными проблемами с приватностью и правовыми аспектами. В силу этого, анализ может проводиться по определенным паттернам или же по незашифрованным заголовкам, к примеру по ip, откуда пришел запрос, или по ip, куда идет запрос. Например, в процессе TLS-handshake сервер отправляет клиенту сертификат, содержащий доменное имя. DPI может использовать эту информацию для фильтрации или анализа, даже не расшифровывая трафик. Также по некоторым паттернам можно понять какой алгоритм шифрования использовался для конкретного запроса.

4.6. Подходы к обходу блокировок

Существует несколько методологий и подходов к обходу блокировок, предлагаю читателям перечислить их по ранжиру, от самого популярного к самым незнаменитым:

- 1) изменение характера трафика
 - шифрование трафика (VPN, SSH, TLS)
 - прокси-сервисы (HTTP/HTTPS-прокси, SOCKS5-прокси)
- 2) изменение сигнатур протоколов
 - замена сигнатур или же обфускация (Obfsproxy, Meek)
 - обфускация сессий (шумовой трафик)
- 3) обход блокировки на уровне IP и DNS
 - изменение DNS (DNS-over-HTTPS/DNS-over-TLS)
 - TOR (The Onion Router)
 - I2P (Invisible Internet Project)
- 4) туннелирование и псевдотуннелирование
 - ICMP - использование протокола ICMP (обычно для ping) для передачи данных.
 - DNS - использование DNS - запросов для передачи данных (например, через запросы к доменным именам).
 - GRE - использование протокола GRE для создания виртуальных частных сетей.
- 5) маскирование через разрешённые протоколы
 - VPN через разрешённые протоколы (OpenVPN over TCP / 443, WireGuard)
 - туннелирование через WebSocket
- 6) использование распределённых сетей и децентрализованных решений
 - decentralized VPN (dVPN)
 - технологии блокчейн (Orchid, Mysterium)

4.7. Известные протоколы для доступа к заблокированным ресурсам

4.7.1. Shadowsocks

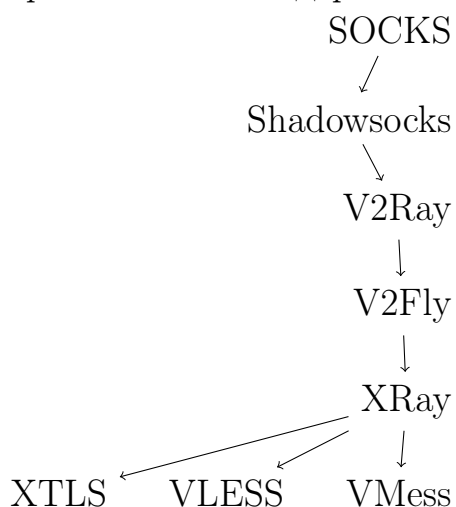
Shadowsocks - такой протокол который предполагает использование прокси-сервера. Сам же сервер использует шифрование для получения с клиента и отправки на него данных. Он был разработан для обхода интернет-цензуры и обеспечивает высокую скорость передачи данных.

Авторы взяли классический SOCKS-протокол, который передает все данные в открытом виде и потому очень легко определяется в DPI протоколах, применили поверх него шифрование с помощью разных алгоритмов, убрали излишний функционал (к примеру, была убрана авторизация в протоколе по логину и паролю, она стала проводиться по ключу шифрования), и добавили несколько других нововведений по обфускации трафика. И это сработало - долгое время Shadowsocks был излюбленным инструментом тысяч людей, позволяющим пробиваться через китайский firewall.

Особенности данного алгоритма:

- 1) Шифрование: Shadowsocks использует различные алгоритмы шифрования (например, AES-256-GCM), что делает трафик менее заметным для DPI.
- 2) Прокси: Это SOCKS5-прокси, который позволяет передавать трафик через сервер, скрывая реальный IP-адрес пользователя.
- 3) Обфускация: Некоторые реализации Shadowsocks включают обфускацию трафика, чтобы он выглядел как обычный HTTPS-трафик.

Хронологическое дерево алгоритмов можно представить в виде графа:



4.7.2. VMess

Непосредственно классических протоколов в V2Ray и XRay всего два не считая VLite: VMess, VLESS.

VMess - протокол, разработанный для использования с V2Ray. Это более сложная и мощная система, чем Shadowsocks. Протокол предлагает больше функций для обхода цензуры. Поддерживает определение "свой/чужой" по ID пользователя и опционально шифрование данных.

На данный момент VMess считается устаревшим, а при работе через TCP - небезопасным, однако вариант VMess-over-Websockets-over-TLS по-прежнему вполне себе жизнеспособен и может использоваться при отсутствии поддерживаемых в каком-либо клиенте альтернатив.

Особенности данного алгоритма:

- 1) Шифрование: VMess использует шифрование как для данных, так и для заголовков, что делает его более защищенным от анализа.
- 2) Динамическое изменение портов: Протокол может динамически изменять порты и другие параметры для затруднения распознавания.
- 3) Мультиплексирование: VMess поддерживает мультиплексирование соединений, что позволяет экономить ресурсы и улучшать производительность.

4.7.3. VLESS

VLESS — новая и улучшенная версия VMess, которая была разработана с акцентом на производительность и безопасность. Протокол предлагает упрощенный подход к шифрованию и аутентификации. В отличие от VMess он не предусматривает механизма шифрования (предполагается, что будет использоваться стандартное шифрование по типу TLS), а только проверку "свой/чужой" и "паддинг" данных (изменение размеров пакетов для затруднения узнавания паттернов трафика). В протоколе исправлен ряд уязвимостей старого VMess, и он активно развивается - например, автор планирует добавить поддержку компрессии алгоритмом Zstandard, не столько для производительности, сколько для затруднения анализа "снаружи".

Особенности данного алгоритма:

- 1) Безопасность: VLESS не использует шифрование заголовков, что делает его более легковесным и быстрым.
- 2) Поддержка различных транспортных протоколов: VLESS может работать через WebSocket, gRPC и другие транспортные протоколы, что помогает обойти блокировки.

- 3) Аутентификация: Использует различные методы аутентификации, включая UUID, что упрощает настройку.

5. ЗАКЛЮЧЕНИЕ

6. СПИСОК ЛИТЕРАТУРЫ

Научная статья №1