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

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 преобразовний)* 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.

Так как трафик это общение не только между машинами, но и между людьми...

4. ГЛУБОКИЙ АНАЛИЗ ПАКЕТОВ

Глубокий анализ пакетов/трафика - основообразующая система для контроля доступа к запрещённым ресурсам на территории конкретного государства. Таким образом достигается либо замедление, либо полная блокировка конкретного ресура. В данный момент, лидером в области ограничения доступа через национальных провайдеров к неугодным ресурсам является Китай.

4.1. Подходы к обходу блокировок

Существует несколько методологий и подходов к обходу блокировок, предлагаю 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)

4.2. Известные протоколы для доступа к заблокированным ресурсам

4.2.1. Shadowsocks

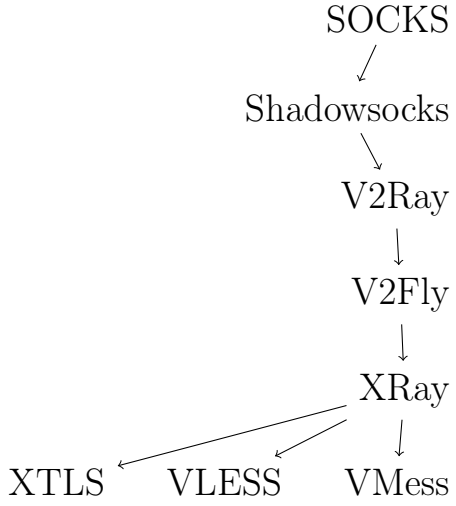
Shadowsocks - такой протокол который предполагает использование прокси-сервера. Сам же сервер использует шифрование для получения с клиента и отправки на него данных. Он был разработан для обхода интернет-цензуры и обеспечивает высокую скорость передачи данных.

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 протоколое 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:



4.2.2. 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. Поддерживат 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.

4.2.3. 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