

**Министерство науки и высшего образования
Российской Федерации
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Технологии Интернета

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ
В ДВУХ ЧАСТЯХ
Часть 2**

ПЕНЗА 2025

Министерство науки и высшего образования
Российской Федерации
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Пензенский государственный университет»

Технологии Интернета

ЛАБОРАТОРНЫЙ ПРАКТИКУМ
В ДВУХ ЧАСТЯХ
Часть 2

Составители:
И. В. Акифьев, Л. В. Коровина

Пенза
Издательство ПГУ
2025

УДК 004.9
Т38

Р е ц е н з е н т
кандидат технических наук, доцент,
доцент кафедры «Землеустройство и геодезия»
Пензенского государственного университета
архитектуры и строительства
Е. П. Тюкленкова

Технологии Интернета : лаб. практикум : в 2 ч. / сост.: И. В. Акифьев, Л. В. Коровина. – Пенза : Изд-во ПГУ, 2025. – Ч. 2. – 44 с.

Представлены теоретический материал и задания для проведения лабораторных работ. Рассмотрены вопросы использования сети Tor для обеспечения анонимности передачи информации в сети Интернет; проанализированы способы сканирования подключенных портов с целью выявления потенциальных опасностей; исследованы принципы обмена информацией по протоколу BitTorrent.

Издание подготовлено на кафедре «Информационное обеспечение управления и производства» ПГУ и предназначено для обучающихся по направлению подготовки 46.03.02 «Документоведение и архивоведение» очной и заочной форм обучения при изучении дисциплины «Технологии Интернета».

УДК 004.9

© Пензенский государственный
университет, 2025

СОДЕРЖАНИЕ

Лабораторная работа № 5. Сеть Tor. Настройка и работа с Tor Browser	4
Лабораторная работа № 6. Исследование удаленной системы для выявления уязвимостей. Обмен информацией по протоколу BitTorrent	21
Глоссарий	39
Список литературы	39

Лабораторная работа № 5

Сеть Tor. Настройка и работа с Tor Browser

Цель работы: освоить базовые навыки работы с сетью Tor; научиться использовать «мосты» для подключения; освоить и применить на практике теоретические знания об анонимном просмотре веб-страниц в сети Интернет.

Общие сведения

Браузер Tor (The Onion Router) – это бесплатный анонимный веб-обозреватель, который позволяет пользователям сохранять свою конфиденциальность и анонимность в Интернете. Он был разработан и поддерживается некоммерческой организацией «Tor Project» с целью защиты прав и свобод человека на свободу слова и информации.

Tor использует систему *прокси-серверов* для шифрования и анонимизации сетевого трафика, скрывая IP-адрес пользователя и не позволяя третьим лицам отслеживать его действия в сети. Прокси-серверы расположены по всему миру и организованы в так называемую «*луковую сеть*», где каждый сервер скрывает информацию о предыдущем и последующем узле. Таким образом, информация передается через несколько слоев шифрования, что делает практически невозможным отслеживание конечного адресата.

Отсюда следует еще одно преимущество использования Tor. Поскольку узел А (устройство пользователя) обращается к узлу Б (удаленный интернет-ресурс) не напрямую, а через промежуточные узлы, то это позволяет обойти «черный список» *брандмауэра* сети.

Рассмотрим конкретный пример. Предположим, в организации администратор заблокировал доступ сотрудников к социальной сети (например, «Одноклассники»). Сайт ok.ru и будет узлом Б, рабочий компьютер сотрудника – узлом А.

В таком случае пользователь запускает программу Tor и вводит адрес узла Б. Передаваемые им данные (в данном случае – адрес узла) будут зашифрованы и переданы первому узлу в цепочке – назовем его узел В, затем данные в том же зашифрованном виде будут переданы узлу Г и т.д. Так будет продолжаться, пока данные не получит последний узел цепочки (например, узел Т), который и передаст запрос пользователя конечному узлу – Б. Стоит отметить, что на последнем участке (от узла Т к узлу Б) данные будут незашифрованы, поскольку узел Б не поддерживает открытые ключи сети Tor (если бы это было так, то весь Интернет был бы анонимным).

На рис. 5.1 изображен процесс передачи данных между локальным и удаленным компьютерами через сеть Tor.

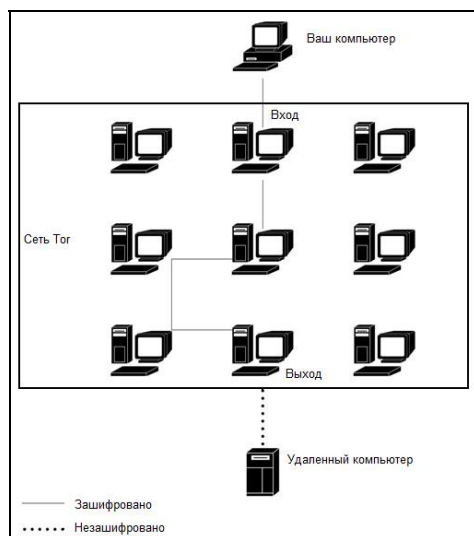


Рис. 5.1. Принцип работы сети Tor

Проанализировав рисунок, можно сделать следующие выводы:

- администратор локальной сети (или администратор провайдера) не сможет узнать, какие данные передает пользователь, поскольку данные передаются в зашифрованном виде;
- администратор сети не сможет узнать, какой узел посещает пользователь, поскольку вместо интересующего его узла (например, ok.ru, vk.com и т.п.) он формально будет обращаться к одному из узлов сети Tor – ничем не примечательному узлу с непонятным доменным именем;
- если администратор сети заблокировал доступ к интересующему узлу на брандмауэре, обойти это ограничение представится возможным, поскольку фактически компьютер подключается к совершенно другому узлу (к узлу цепочки Tor). Запрещать доступ к этому узлу нет смысла, так как при следующем подключении к Tor или при принудительной смене цепочки узел входа в Tor будет изменен;
- удаленный узел «увидит» только IP-адрес последнего узла цепочки, IP-адрес пользователя будет скрыт;
- теоретически перехват данных возможен на последнем участке пути – от последнего узла цепочки Tor до удаленного узла. Но для этого нужно отследить всю цепочку Tor, что технически сделать очень сложно, поскольку она может состоять из десятков узлов. Если же получить доступ к удаленному узлу, то все равно невозможно будет определить, кто есть кто, поскольку для этого нужно знать как минимум точку входа и точку выхода сети Tor.

Таким образом, браузер Tor особенно полезен для обхода цензуры и обеспечения свободы слова, а также для защиты от слежки. Он широко используется журналистами, активистами, правозащитниками и обычными пользователями, которые хотят сохранить свою анонимность и конфиденциальность.

Однако стоит отметить, что использование браузера может замедлить скорость загрузки веб-страниц из-за дополнительного уровня шифрования. Кроме того, использование Tor не обеспечивает абсолютную анонимность, так как некоторые данные все равно могут быть отслежены. Например, серьезный недочет Tor Browser – раскрытие типа операционной системы (ОС) пользователя.

Одним из основных принципов, обеспечивающих анонимизацию пользователей Tor Browser, является похожесть этих пользователей для внешнего наблюдателя в Интернете: они используют браузер одной и той же версии с максимально одинаковыми настройками, вплоть до размеров окон. Несмотря на то, что Tor Browser, запущенный под любой ОС, отправляет заголовок User-Agent¹, соответствующий Windows, механизм, позволяющий скрыть настоящую ОС от тестов, выполняемых средствами JavaScript², был утрачен несколько версий назад.

Если ранее тесты для всех ОС возвращали значение Win32, что соответствовало наиболее популярной ОС Windows, то теперь узнать тип операционной системы не составляет труда для любого сайта, ведь JavaScript включен в Tor Browser по умолчанию. Пример работы сайта webbrowserstools.com приведен на рис. 5.2.

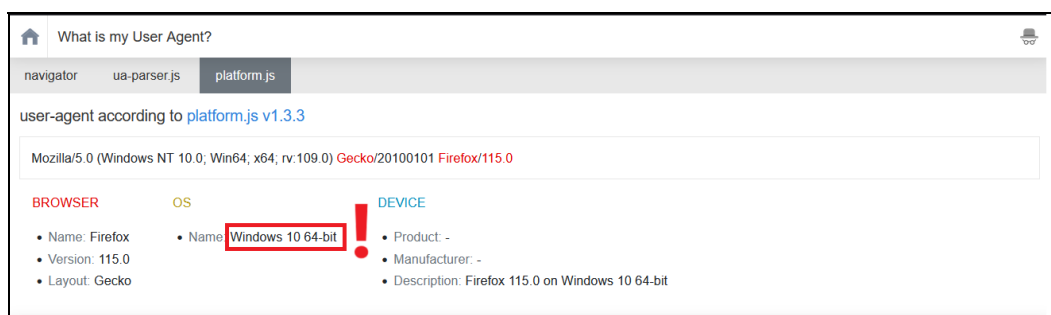


Рис. 5.2. Тест анонимности браузера

Это значительно снижает анонимность пользователей как Windows, так и Linux и выделяет их из общей массы среднестатистических пользователей. До версии Tor Browser 12 эту проблему можно было решить

¹ Идентификационная строка клиентского приложения; обычно используется для приложений, осуществляющих доступ к веб-сайтам браузеров, поисковых роботов и «пауков», мобильных телефонов и других устройств со встроенным доступом к веб-ресурсам.

² Язык программирования.

установкой дополнения User-Agent Switcher, как это предлагалось пользователям Tor Browser ранее. Однако, начиная с 12-й версии, данное дополнение перестало скрывать тип используемой ОС.

Задания

Задание 1. Скачайте и установите браузер Tor на компьютер.

Теоретический материал для выполнения задания

Программное обеспечение Tor следует скачивать только с официального сайта по адресу: <https://www.torproject.org/>

Данная ссылка, как правило, обычным способом не открывается. Для этого используется VPN Browsec (или аналогичный сервис VPN) для браузера, чтобы обойти блокировку. На сайте <https://browsec.com> справа сверху будет обведенная красным кнопка Download (рис. 5.3).

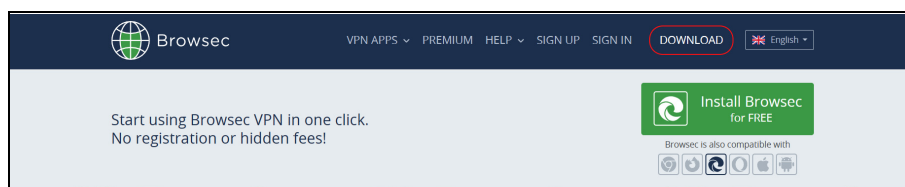


Рис. 5.3. Официальный сайт Browsec

Как видно из интерфейса сайта, расширение доступно для большинства браузеров. Следуя инструкциям, установите расширение для Вашего браузера и активируйте в панели задач (рис. 5.4, 5.5).

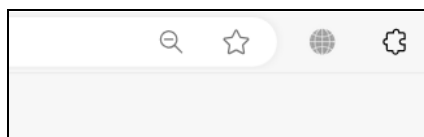


Рис. 5.4. Значок расширения в панели задач браузера Edge

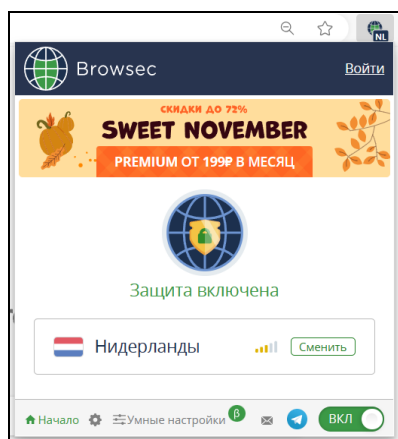


Рис. 5.5. Активация расширения

После установки и настройки VPN необходимо перейти по ссылке <https://www.torproject.org/> (рис. 5.6), нажать кнопку «Скачать Tor Browser»

(«Download Tor Browser», если язык автоматически не переключится на русский).

В открывшемся окне выбора установочного пакета для различных операционных систем выберите кнопку «Скачать для Windows».

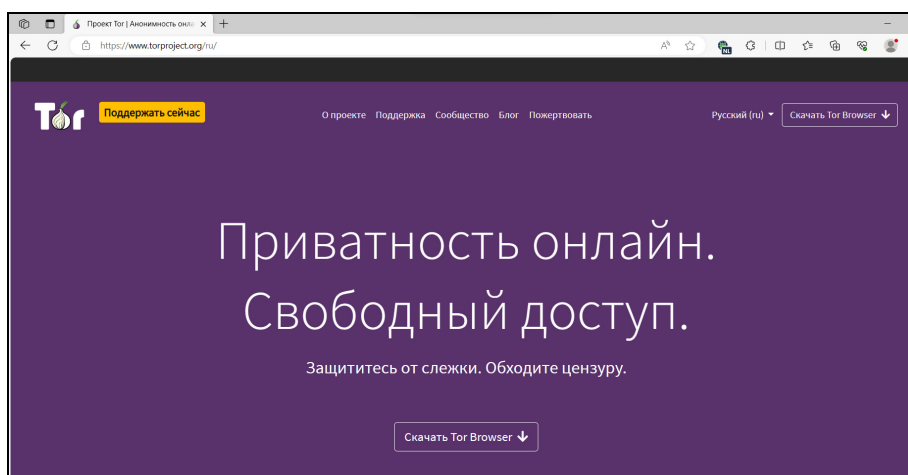


Рис. 5.6. Официальный сайт Tor

Примечание. Если скорость загрузки низкая – приостановите ее, отключите VPN и возобновите процесс загрузки файла.

Запустите загруженный файл, указав каталог, в который следует распаковать Tor. Пакет Tor Browser может работать в Windows (от XP до Windows 11), Linux, macOS, Android.

При первом запуске открывается окно предварительной конфигурации программы-маршрутизатора (рис. 5.7).

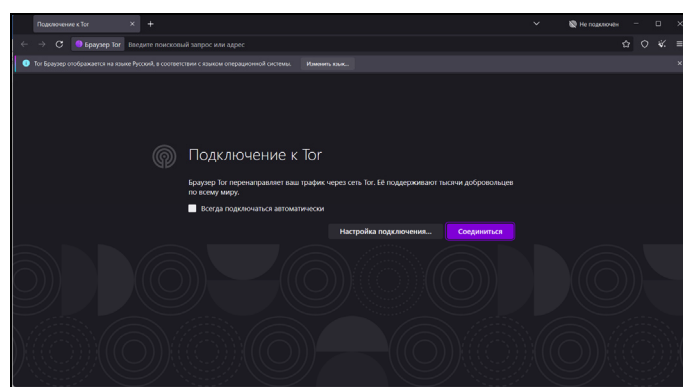


Рис. 5.7. Первое подключение к Tor Browser

Если процесс запуска завис на окне загрузки, то программа сама предложит вариант решения проблемы – подключиться к сети Tor через *мост* (рис. 5.8).

Оставьте выбранное значение «Автоматически» в выпадающем списке «Разблокировать Интернет» и нажмите кнопку «Попробовать мост».

Tor позволяет пользователям запрашивать мосты в индивидуальном порядке. В случае, если браузеру не удалось найти мост для подключения (очень долгая загрузка), нужно нажать кнопку «Отмена» и в окне, изображенном на рис. 5.8, перейти по гиперссылке «Настроить подключение». В открывшемся окне настроек в разделе «Подключение» в группе «Добавить новый мост» выберите пункт «Запросить мост у torproject.org» / «Запрос моста» (рис. 5.9).

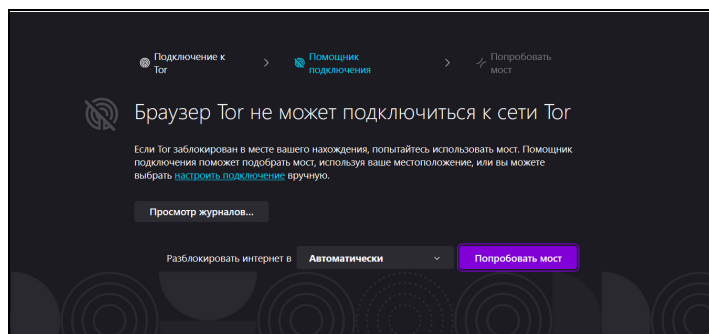


Рис. 5.8. Помощник подключения Tor

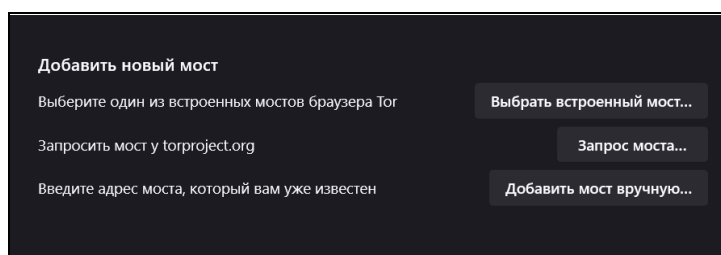


Рис. 5.9. Запрос моста

После ввода проверочного кода (капчи) Tor предоставит мост, который будет отображаться выше в группе «Ваши текущие мосты» со статусом «Подключен».

При установлении стабильного соединения с сетью Tor запускается стартовая страница браузера (рис. 5.10).

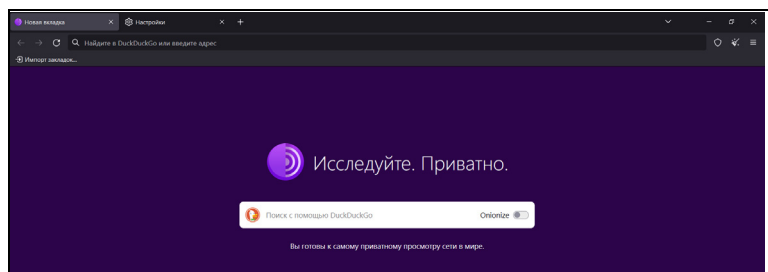


Рис. 5.10. Стартовая страница браузера

Необходимо обратить внимание на следующие командные кнопки:

– **Security Level** – стилизованное изображение щита, находящееся правее адресной строки, позволяющее открыть настройки уровня безопасности браузера, рассматриваемые далее.

– **New Identity** – стилизованное изображение метелки, позволяет перезапустить сессию работы с Tor Browser, чтобы оборвать свой «цифровой след» в Интернете. При нажатии на нее происходит построение нового Tor-адреса последнего узла цепочки, закрытие всех существующих окон браузера и очистка всех временных данных браузера, включая данные авторизации. Для пользователя это фактически равнозначно перезапуску всего браузера.

Задание 2. Проведите тест анонимности.

Теоретический материал для выполнения задания

Общеизвестно, что установить личность интернет-пользователя можно по его IP-адресу. Следует отметить, что IP-адрес является важным, но не единственным идентификатором пользователя в Интернете.

Чтобы убедиться, что браузер действительно работает через Tor, а значит, IP-адрес, с которым компьютер работает в Интернете, принадлежит одному из выходных узлов Tor, необходимо зайти на check.torproject.org – специальный сайт от проекта Tor. При корректной работе цвет луковицы будет зеленым (рис. 5.11).

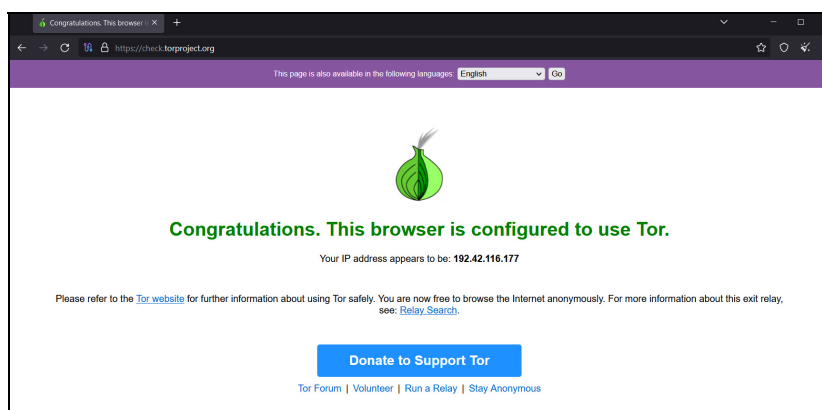


Рис. 5.11. Диалоговое окно, подтверждающее корректную работу Tor

В противном случае цвет луковицы может быть иным, например желто-оранжевым.

На официальном сайте check.torproject.org выполняется самая простая проверка анонимности, на прямое сокрытие IP-адреса. Есть множество более сложных техник, позволяющих установить если не истинный IP-адрес пользователя, то хотя бы то, что один и тот же пользователь использует разные подставные IP-адреса. Существует множество сайтов, использующих эти техники для проверки надежности анонимизации пользователя.

Задание 3. Выполните дополнительную настройку Tor, выбрав разные уровни безопасности, чтобы добиться еще большего уровня анонимности в сети.

Примечание. Повышение безопасности любой системы практически всегда приводит к снижению удобства ее использования. Некоторые сайты могут частично или полностью перестать работать. Это явление характерно для сайтов, перегруженных сложным функционалом.

Теоретический материал для выполнения задания

По умолчанию браузер настроен на наиболее комфортную работу пользователя. Настройки безопасности разрешают использование большинства современных Web-технологий ценой некоторого снижения уровня безопасности и повышения потребления ресурсов системы.

Исходя из приоритета безопасности и анонимности, рассмотрите дополнительные варианты настроек безопасности, предлагаемые Tor Browser. Для этого необходимо кликнуть на кнопке настроек безопасности правее адресной строки, далее выбирается команда «Настройки» (рис. 5.12).

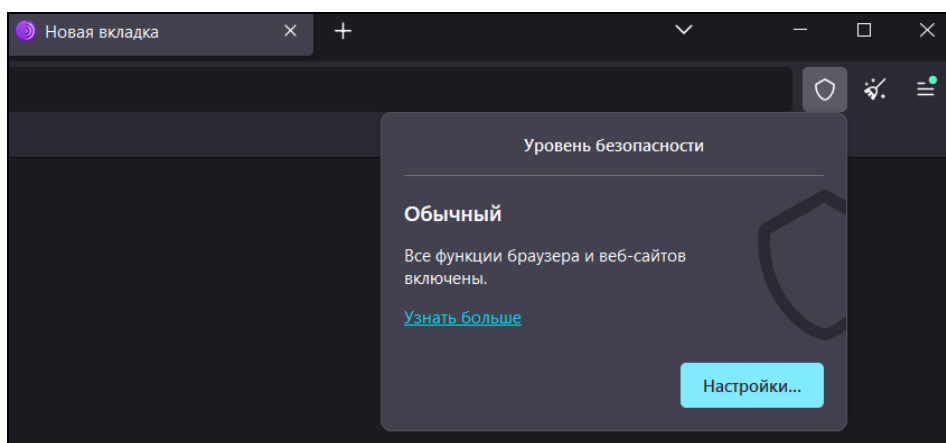


Рис. 5.12. Изменение уровня текущей безопасности

Для начала выберите средний (Safer) уровень безопасности (рис. 5.13).

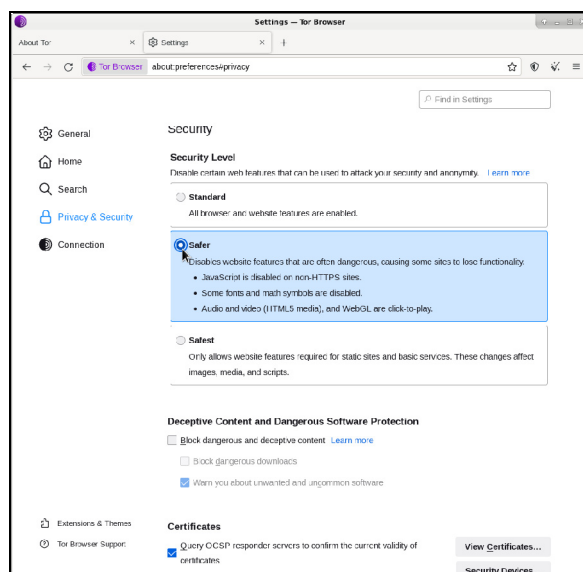


Рис. 5.13. Средний уровень безопасности

Особенности Safer уровня:

1. JavaScript выключен по умолчанию на сайтах, не поддерживающих HTTPS, защищающего соединение от прослушки и злонамеренного изменения данных. Таким образом, гарантируется, что загружаемые *скрипты*, несущие дополнительные риски безопасности, не были подменены «по дороге» злоумышленником.

2. HTML5 видео и аудио (например, на таких сайтах, как youtube.com) будет проигрываться только после явного клика мышкой по проигрывателю и активации через меню NoScript (тема рассмотрена далее).

3. Некоторые шрифты и отображение сложных математических формул и уравнений будут отключены. Это может затронуть, например Wikipedia.org – вместо формул со сложными дробями, математическими знаками интегралов и сумм будет представлена нечитаемая совокупность символов и ключевых слов языка разметки формул.

Если необходимо посещать сайты со множеством динамического контента, такие как социальные сети или видеохостинги, то этот режим будет являться разумным балансом между безопасностью и удобством использования.

Если же в приоритете безопасность и анонимность, то рекомендуется использовать максимальный уровень (Safest).

Отличия «наиболее безопасного» (Safest) режима от предыдущего уровня:

- JavaScript отключен по умолчанию для всех сайтов.
- Отключена поддержка векторных (SVG) рисунков.

Примечание. На этом уровне безопасности JavaScript выключен по умолчанию. Но даже после его включения многие сайты могут отображаться неправильно (например, ВК), а некоторые перестают работать (например, YouTube).

Задание 4. Добавьте кнопку дополнения NoScript на панель браузера, чтобы иметь возможность управлять настройками дополнения самостоятельно.

Теоретический материал для выполнения задания

Это дополнение может блокировать выполнение JavaScript (JS) кода и некоторых других потенциально опасных объектов, создающих интерактивность Web-страниц, и по совместительству – повышенное энергопотребление компьютера и множество потенциальных уязвимостей. К сожалению, современные сайты в большинстве своем очень сильно завязаны на возможность выполнять JS код, и многие из них (например, VK, Facebook) без JS не смогут работать. Именно это является причиной того, что по умолчанию в Tor Browser блокировка JS отключена, т.е. всем посещаемым страницам разрешается выполнять почти любой JS код.

Чтобы добавить кнопку дополнения NoScript на панель браузера нужно кликнуть правой кнопкой мыши справа или слева от адресной строки, например, на значке обновления страницы и в контекстном меню выбрать пункт «Customize...» («Персонализация...») (рис. 5.14).

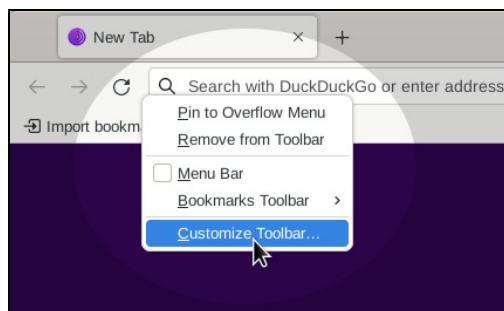


Рис. 5.14. Открытие настроек персонализации

Найти в списке доступных функций поисковую строку (рис. 5.15).

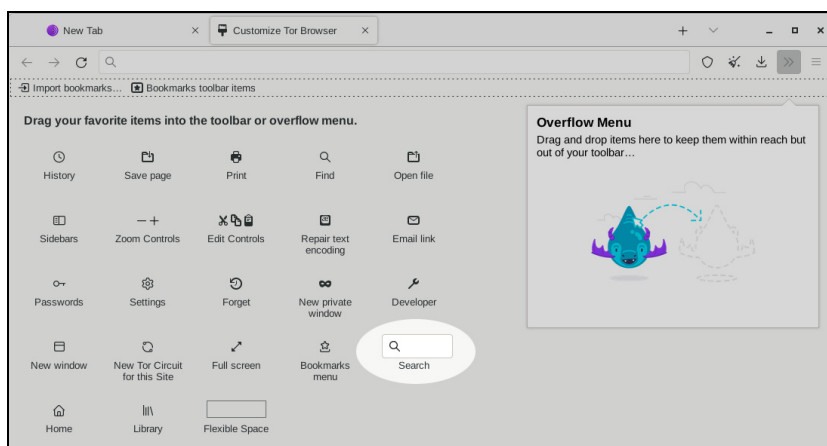


Рис. 5.15. Список доступных функций

Перетащить кнопку мышью на панель инструментов, например справа от адресной строки.

Вкладку настройки внешнего вида («Customize Tor Browser») можно закрыть либо добавить дополнительные элементы по своему усмотрению.

В главном меню зайти в раздел Add-on and themes (рис. 5.16).

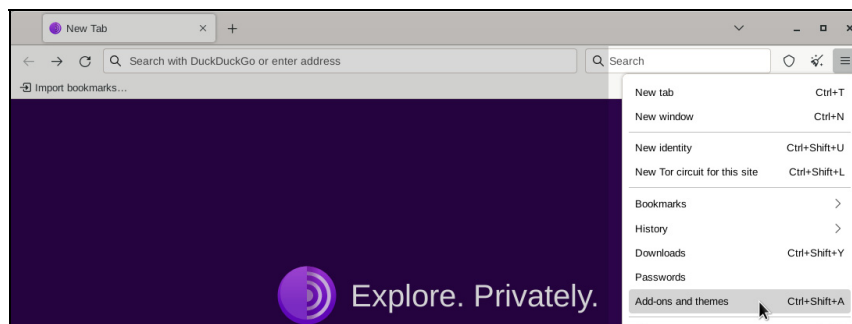


Рис. 5.16. Открытие настроек расширений

Кликнуть «Manage» в настройках NoScript, на вкладке «Details» выбрать «Show» (рис. 5.17).

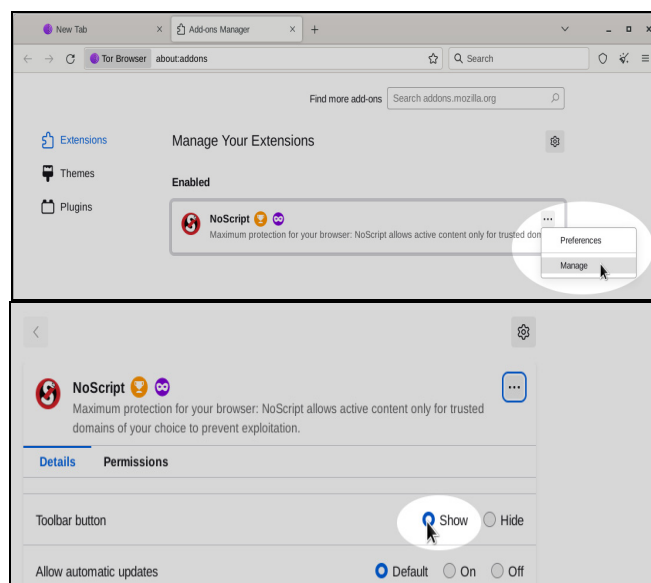


Рис. 5.17. Настройки расширения

NoScript и поисковая строка возвращаются на их обычное место на панель инструментов в правом верхнем углу сети Tor.

Задание 5. Задайте средний уровень безопасности (Safer) и зайдите на YouTube.com.

Теоретический материал для выполнения задания

На первый взгляд, проблем нет и все работает. Но при попытке просмотра видео появится либо черный фон со вращающимся значком загрузки в центре, либо запрещающий знак NoScript. Чтобы просмотреть видео, нужно кликнуть на значок NoScript справа от адресной строки. В раскрывшемся меню будет от одной до нескольких строк, каждая из которых состоит:

- из уровня разрешений – в левой части. Каждый из уровней обозначен отдельным значком:

-  – разрешения по умолчанию, зависят от заданного уровня безопасности;

-  – добавить сервер во *временный* список доверенных, которым разрешено выполнять активное содержимое. Разрешения сбросятся в значение по умолчанию после перезапуска браузера;

-  – добавить сервер в постоянный список доверенных, т.е. однажды заданные настройки разрешения будут сохранены и после перезапуска браузера. Не рекомендуется использовать этот пункт, так как можно забыть об этих разрешениях, но сайт будет продолжать их использовать;

– запретить все активное содержимое;
– задать настройки разрешений вручную;
– адреса сервера, для которого задаются настройки – в правой части.

В ситуации с YouTube следует начать с разрешения только самого YouTube, для чего нужно кликнуть по выделенному красным кружком значку в первой строке, левее надписи youtube.com (рис. 5.18).

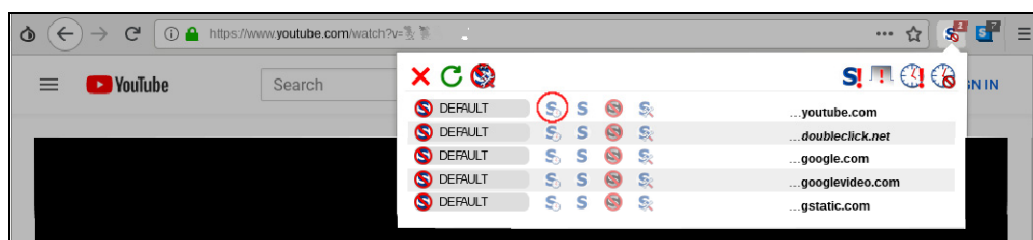


Рис. 5.18. Окно NoScript

После чего нужно закрыть меню NoScript, кликнув в любом месте на странице или на красный крест в верхнем левом углу. Страница автоматически обновится, а меню NoScript примет вид, как показано на рис. 5.19.

В случае с YouTube этого, скорее всего, будет достаточно для нормального воспроизведения видео. Если вдруг это не помогло, то можно постепенно добавлять разрешения для всех остальных серверов из списка NoScript.

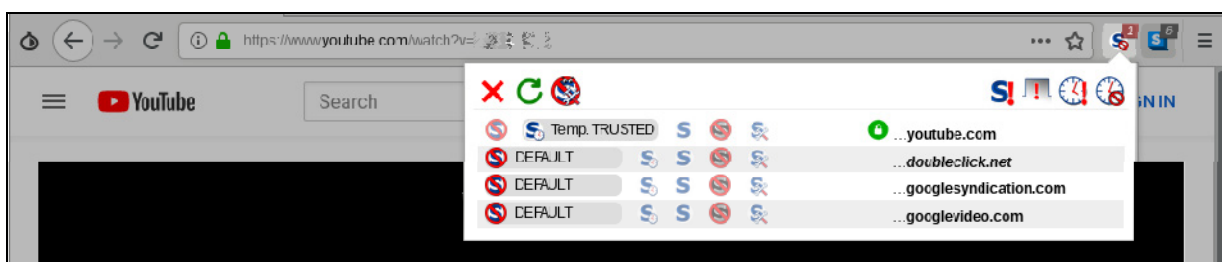


Рис. 5.19. Окно NoScript

При этом лучше избегать разрешения рекламных и аналитических сервисов, сервера которых часто содержат слова вроде «ads», «click», «spy», «banner», «metric», «counter», «top», «analytics» и др., связанные с рекламной деятельностью и Интернет-аналитикой.

Отдельно стоит остановиться на задании разрешений вручную. Использование его хотя и не обязательно, но предпочтительно, т.к. позволяет выбрать тот необходимый минимум, который действительно следует разрешить. Для примера можно использовать любой сайт (рис. 5.20).

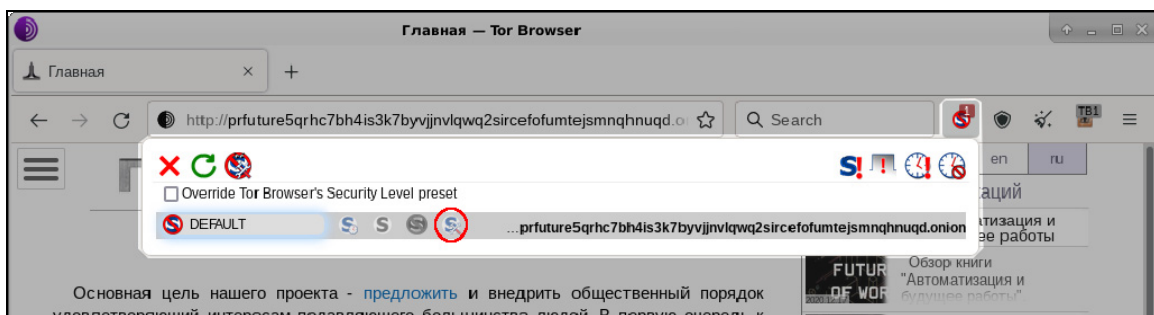


Рис. 5.20. Задание разрешений вручную

Если кликнуть на значок , то откроется подменю.

В нем есть список из флажков (checkbox), установка галочек на которых обозначает разрешение соответствующей технологии для данного сервера:

1. Script – разрешить непосредственно Java Script и другие языки скриптов. Чаще всего именно этот пункт нужно разрешить, чтобы страница «оживла».

2. Object – используется достаточно редко, рекомендуется разрешать, только если по-прежнему ничего не работает.

3. Media – различные мультимедиа технологии, в частности видео. Таким образом, для работы YouTube и других современных видеосервисов этот параметр должен быть разрешен.

4. Frame – используется нечасто, но не несет значимой опасности, потому разрешен по умолчанию.

5. Font – позволяет использовать шрифты, заданные создателем сайта. Без него страница может хуже выглядеть, некоторые элементы оформления могут не отображаться. Имейте в виду, что включение этой возможности вместе с разрешенными скриптами (script) позволяет сайту определить, какие шрифты установлены на компьютере, тем самым снизив ваш уровень анонимности.

6. WebGL – разрешается только при реально существующей необходимости.

7. Fetch, ping, other, noscript – используется некоторыми видами скриптов, можно разрешить если все обычное разрешение скриптов не помогло привести страницу в работоспособное состояние.

8. Unrestricted CSS – включает функционал браузера, позволяющий реализовать уязвимость PP0, при помощи которой злонамеренный веб-сайт может «подслушать» данные других приложений, запущенных на компьютере. Рекомендуется включать этот функционал только для действительно доверенных ресурсов.

Обратите внимание, что возможности, действительно запрашиваемые сайтом, выделяются красным фоном. В данном примере это возможность font. Все остальные пункты, не подсвеченные красным, в действительности сайтом не используются и разрешать их не имеет смысла.

Стоит также обратить внимание на значки в верхней строке меню NoScript.

Значки из левой части обозначают, соответственно: закрыть меню NoScript, обновить страницу и открыть настройки самого NoScript.

Первые три значка из правой части обозначают разрешение всего активного контента глобально (т.е. для всех сайтов), отмену ограничений для активной вкладки и добавление текущей страницы в список временно доверенных. Использование этих кнопок разрешает выполнение активного контента либо для всех сайтов вообще, либо для отдельных сайтов или вкладок. Потому не рекомендуется использовать эти кнопки, если важны анонимность и безопасность.

Кнопка, разрешающая все скрипты для отдельной вкладки, имеет вид . Ее можно использовать, например, когда временно необходимо привести в работоспособное состояние сайт, загружающий скрипты с пары десятков серверов. Разумеется, делать это вручную как неудобно, так и опасно, ведь некоторые из этих серверов могут использоваться и на других вкладках, на которых не нужно разрешать JavaScript. В таком случае правильней все-таки разрешить все скрипты для данной вкладки. После все временные разрешения для всего множества разрешенных серверов автоматически сбросятся.

Последняя, самая правая кнопка из верхнего ряда отменяет все выданные ранее временные разрешения для всех сайтов. Это значит, что после нажатия на нее все временные разрешения, добавленные через кнопку , вернутся в значение по умолчанию для выбранного уровня безопасности. Важно не забывать, что в случае с самым высоким уровнем (Safest), это обозначает запрет всех скриптов во всех вкладках и окнах браузера, следовательно, приостановку работы современных социальных сетей, видеохостингов и прочих сайтов.

Задание 6. Ограничьте автоматические подключения браузера к серверам.

Теоретический материал для выполнения задания

Даже если пользователь не проявляет какой-либо активности в Интернете и отключил JavaScript, способный выполнять запросы к серверам без его участия, браузер все равно будет время от времени автоматически создавать подключения к различным серверам в Интернете.

Например, он может выполнять проверку и загрузку обновлений самого браузера и установленных дополнений, обновлять различные служебные страницы. В некоторых случаях браузер может для удобства (так решили разработчики) создавать подключения к серверам в зависимости от действий пользователя, хотя он уверен, что не выполняет ничего такого, что должно привести к подключению к серверам в Интернете.

Отключение автоматического обновления Tor Browser

Начиная с Tor Browser версии 9 стало невозможным отключение автоматической проверки наличия обновлений через интерфейс настройки.

Для отключения автоматической загрузки и установки обновлений Tor Browser откройте настройки.

Введите «update» в поиск в правом верхнем углу вкладки настроек, после чего выберите опцию «Check for updates but let you choose to install them» (рис. 5.21).

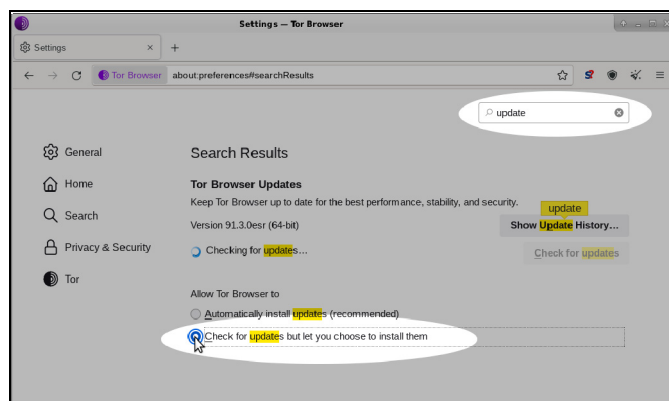


Рис. 5.21. Отключение автоматических обновлений

В таком режиме проверка обновлений выполняется автоматически, но скачивание происходит только после явного подтверждения пользователем.

Отключение проверки обновлений

Несмотря на невозможность выключить проверку обновлений через настройки браузера, это по-прежнему можно сделать, используя политики Firefox.

Примечание. Не забывайте, что, полностью отключив проверку обновлений, вы берете на себя ответственность за отслеживание выпуска новых версий браузера.

Чтобы сделать это, нужно сперва перейти в каталог, где находится исполняемый файл браузера, и создать там каталог с именем distribution, внутри которого создать файл policies.json следующего содержания:

```
{ "policies": { "DisableAppUpdate": true, "DisableTelemetry": true } }
```

После этого перезапустите браузер.

Например, если браузер распакован в /tmp, то нужным каталогом будет /tmp/tor-browser/Browser. В командной строке (Пуск / Стандартные / Командная строка) необходимые действия можно сделать так:

```
mkdir /tmp/tor-browser/Browser/distribution
```

```
cd /tmp/tor-browser/Browser/distribution
```

```
echo '{ "policies": { "DisableAppUpdate": true, "DisableTelemetry": true } }' > policies.json
```

Отключение автоматического обновления дополнений

Для этого выберите в меню пункт Add-on and themes.

В разделе Extensions откройте настройки и снимите галочку с пункта Update Add-ons Automatically.

После этого сбрасываем настройки автоматического обновления всех установленных дополнений в режим проверки вручную. Для этого нажимаем кнопку Reset All Add-ons to Update Automatically.

Отключение протокола OCSP

Когда пользователь посещает HTTPS-сайты (которых сегодня большинство), Tor Browser проверяет при помощи протокола OCSP действительность сертификата шифрования, используемого посещаемым сайтом. В ходе этой проверки он может устанавливать соединение с третьей стороной, обладающей информацией по действительности сертификата. А значит, о том, что посещен некоторый сайт, становится известно не только пользователю и серверу, на котором находится сайт, но и третьей стороне.

Безопасней всего будет отключить эту проверку, потому что случаи отзывов сертификатов достаточно редки, а HTTPS-сайты посещаются постоянно и их число с течением времени только увеличивается.

Открываем настройки, пишем «OCSP» в поиск в правом верхнем углу вкладки настроек, после чего снимаем галку с пункта Query OCSP responder server to (рис. 5.22).

Сброс зачек

Если пользователь запустил «закачку» большого объемного файла, но он не успел «докачаться» до того, как было произведено отключение от Интернета, то загрузка автоматически возобновится, как только у Tor Browser появится доступ в Интернет.

Если это не нужно, то следует открыть загрузки и удалить ненужные файлы, как показано на рис. 5.23.

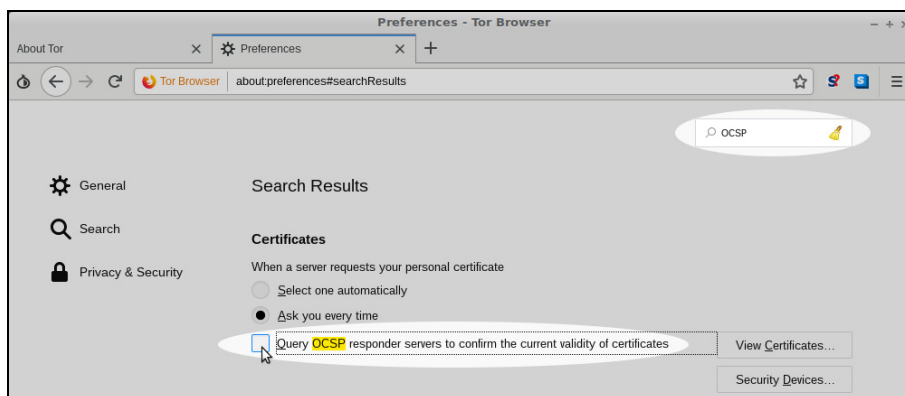


Рис. 5.22. Отключение протокола OCSP

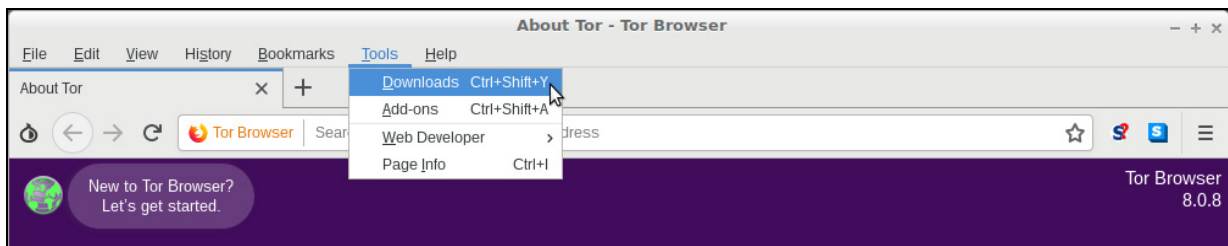


Рис. 5.23. Сброс зачек

Контрольные вопросы

1. С какой целью используется Tor?
2. Как работает сеть Tor?
3. Какие функции включает в себя Tor Browser?
4. Как настроить Tor Browser на компьютере?
5. Как проверить работоспособность Tor-соединения?
6. Что такое мосты в Tor и как их использовать?
7. Как обеспечить анонимность при использовании Tor?
8. Каковы особенности Safer уровня безопасности? Охарактеризуйте их.
9. Каковы особенности Safest уровня безопасности? Охарактеризуйте их.
10. Какие функции выполняет кнопка дополнения NoScript?
11. Каковы уровни разрешений NoScript? Охарактеризуйте их.
12. Как расширить список функций панели инструментов Tor?
13. Какие меры предосторожности следует принимать при работе с Tor?
14. Каковы основные преимущества и недостатки использования Tor?

Лабораторная работа № 6

Исследование удаленной системы для выявления уязвимостей. Обмен информацией по протоколу BitTorrent

Цель работы: научиться анализировать сетевые компьютеры и выявлять потенциальные опасности в их настройках. Получить практические навыки обмена информацией по протоколу BitTorrent.

Средства для выполнения работы:

- программные: сканер локальной сети Net View; сканер портов nmap;
- информационные: адрес сервера для сканирования;
- программы обмена информацией по протоколу BitTorrent.

Общие сведения

В Интернете применяют два основных протокола транспортного уровня: **TCP**, ориентированный на создание соединений, и **UDP** (User Datagram Protocol – пользовательский дейтаграммный протокол) – без установления соединения.

Протокол UDP позволяет приложениям отправлять *инкапсулированные* IP-дейтаграммы без установления соединений. Он практически представляет собой протокол IP с добавлением небольшого заголовка. Его отличие от использования IP заключается в указании портов источника и приемника.

Протокол TCP взаимодействует через межуровневые интерфейсы с нижележащим протоколом IP и вышележащими протоколами прикладного уровня или приложениями. Его задача заключается в передаче данных между любыми прикладными процессами, выполняющимися на любых узлах сети. Т.е. доставленный в компьютер-получатель средствами IP-протокола пакет должен быть направлен конкретному процессу-получателю.

Пакеты, поступающие на транспортный уровень, организуются операционной системой в виде множества очередей к точкам входа различных прикладных процессов. В терминологии TCP/IP такие системные очереди называют *портами*. Таким образом, адресом назначения, который используется протоколом TCP, является идентификатор (номер) порта прикладной службы. Номер порта в совокупности с номером сети и номером конечного узла однозначно определяют прикладной процесс в сети, который имеет название *сокет (socket)*.

В 1995 г. корпорация Netscape Communications представила систе-

му безопасности под названием SSL (Secure Sockets Layer – протокол защищенных сокетов). Протокол **SSL** используется очень широко (в том числе Internet Explorer).

SSL создает защищенное соединение между двумя сокетами, позволяющее:

- клиенту и серверу «договориться» об используемых параметрах;
- клиенту и серверу произвести взаимную аутентификацию;
- организовать тайное общение;
- обеспечить защиту целостности данных.

По сути дела, между прикладным и транспортным уровнями появляется новый уровень, принимающий запросы от браузера и отсылающий их по TCP для передачи серверу. После установки защищенного соединения основная задача SSL заключается в поддержке сжатия и шифрования. Если поверх SSL используется HTTP, этот вариант называется **HTTPS** (Secure HTTP – защищенный HTTP) несмотря на то, что это обычный протокол HTTP.

Область применения SSL не ограничивается исключительно веб-браузерами, но это наиболее распространенное применение. Существует несколько версий протокола SSL. Третья версия протокола SSL поддерживает множество разных алгоритмов и может обладать разными функциями, среди которых сжатие, тот или иной алгоритм шифрования, а также некоторые компоненты, связанные с ограничениями экспорта в криптографии.

SSL состоит из двух субпротоколов, один из которых предназначен для установления защищенного соединения, а второй – для использования этого соединения. SSL поддерживает разнообразные *криптографические алгоритмы*. Наиболее сильный из них использует для шифрации тройной стандарт шифрования **DES** (Data Encryption Standard – стандарт шифрования данных) с тремя отдельными ключами и функцию вычисления профиля сообщения **SHA-1** (Secure Hash Algorithm – надежный алгоритм хэширования) для обеспечения целостности данных. Такое сочетание алгоритмов работает медленно, поэтому применяется в основном в приложениях, в которых требуется высокий уровень защиты. В обычных приложениях для шифрации применяется 128-разрядный ключ, а для аутентификации – алгоритм **MD5** (Message Digest 5 – профиль сообщения 5). В качестве исходных данных алгоритму **RC4** передается 128-разрядный ключ, который разрастается во много раз при работе алгоритма. Это внутреннее число используется для создания ключевого потока. Последний суммируется по модулю 2 с открытым текстом, в результате чего получается обычный потоковый шифр.

Для реальной передачи данных используется второй субпротокол. Сообщения, поступающие от браузера, разбиваются на единицы данных размером до 16 Кбайт. Если сжатие включено, каждая из этих единиц независимо сжимается. Затем по двум нонсам вычисляется закрытый ключ, подготовительный ключ объединяется со сжатым текстом и результат хешируется по согласованному алгоритму (чаще всего **MD5**). *Хеш* добавляется к каждому фрагменту в виде **MAC** (Message Authentication Code – код аутентификации сообщения). Этот сжатый фрагмент вместе с **MAC** кодируется согласованным алгоритмом с симметричным ключом (обычно это суммирование по модулю 2 с ключевым потоком **RC4**). Наконец, присоединяется заголовок фрагмента, и фрагмент передается по TCP-соединению.

Назначение номеров портов прикладным процессам осуществляется либо централизованно, если эти процессы представляют собой популярные общедоступные службы (21 – служба удаленного доступа к файлам FTP, 23 – служба удаленного управления Telnet), либо локально для тех служб, которые еще не стали распространенными, чтобы закрепить за ними зарезервированные номера. Локальное присвоение номера порта заключается в том, что разработчик некоторого приложения просто связывает с ним любой произвольно выбранный числовой идентификатор, обращая внимание на то, чтобы он не входил в число зарезервированных. В дальнейшем все удаленные запросы к этому приложению от других приложений должны адресоваться с указанием назначенного ему номера порта.

Номера портов делят на три категории:

1. Номера от 0 до 1023 – зарезервированы для хорошо известных портов, которые ассоциированы со службами на постоянной основе (например, HTTP-серверы всегда принимают запросы к порту 80).

2. Порты с номерами от 1024 до 49 151 являются регистрируемыми и используются для различных целей.

3. Порты с номерами 49 151 до 65 535 представляют собой динамические и частные порты, с которыми не должна ассоциироваться ни одна служба.

Ниже приведены некоторые соответствия портов и служб/протоколов:

1. **TCP 22** – наиболее активно работающий сетевой порт. Используется программой Secure Shell (SSH), которая применяется для подключения компьютера с ОС Linux с утилитой IPTraf.

2. **UDP 138** – задействован службой NetBIOS Datagram Service. Данный порт используется несколькими службами Windows, в том числе Computer Browser, DFS, License, Messenger, Net Logon и Server.

3. В группу портов **TCP и UDP 135-139** входит несколько специфических портов, используемых многими приложениями Windows. По всей вероятности, некоторые из этих портов придется держать открытыми, что, к сожалению, открывает доступ к другим приложениям Windows.

4. **TCP 80** – используется для незашифрованного трафика HTTP (Web).

5. **UDP 137 и UDP 53** – эти порты используются службами преобразования имен Windows – в данном случае, NetBIOS и DNS.

6. **UDP 67 и UDP 68** – используются DHCP-сервером для назначения динамических IP- адресов.

7. **UDP 123** – зарезервирован для протокола Network Time Protocol (NTP) или, в случае Windows, Simple Network Time Protocol (SNTP). Этот протокол синхронизирует время между компьютером и NTP-сервером, например, контроллером домена (DC).

Сетевые порты могут дать важнейшую информацию о приложениях, которые обращаются к компьютерам по сети. Зная приложения, которые используют сеть, и соответствующие сетевые порты, можно составить точные правила для брандмауэра и настроить хост-компьютеры таким образом, чтобы они пропускали только полезный трафик.

Задача администратора состоит в том, чтобы выявить недостатки в функционировании сети и устранить их. Самый распространенный способ – *сканирование портов* – процесс обнаружения прослушивающих приложений путем активного опроса сетевых портов компьютера или другого сетевого устройства.

Результаты сканирования и сравнения сетевых отчетов с результатами хост-опроса портов позволяет составить ясную картину трафика, проходящего через сеть. Например, сканируя диапазон внешних IP-адресов, можно собрать ценные данные о взломщике, проникшем из Интернета. Поэтому следует чаще сканировать сеть и закрыть все необязательные сетевые порты. Построив профиль сети и разместив инструменты для распознавания сетевого трафика, можно более эффективно обнаруживать взломщиков, анализируя генерируемый ими сетевой трафик. Можно значительно уменьшить вероятность проникновения в систему, если отключить сетевую активность приложений (служб) или заблокировать неиспользуемые порты компьютера.

Каждый открытый порт – потенциальная лазейка для взломщиков, которые используют пробелы в хост-приложении или тайком обращаются к приложению с именем и паролем другого пользователя (либо применяют другой законный метод аутентификации). Можно обнаружить открытые порты с помощью стандартных приложений операционной системы (ОС). Например, в ОС Windows можно воспользоваться

утилитой Netstat. Она служит для отображения активных подключений TCP, портов, прослушиваемых компьютером, статистики Ethernet, таблицы маршрутизации IP, статистики **IPv4** (для протоколов IP, ICMP, TCP и UDP) и **IPv6** (для протоколов IPv6, ICMPv6, TCP через IPv6 и UDP через IPv6). Запущенная без параметров, команда *netstat* отображает подключения TCP. Результатом сканирования с параметром - *anap* является список с открытыми портами компьютера и название служб/протоколов, работающих на этом порту. Windows содержит ряд утилит для диагностики, отслеживания производительности и восстановления сетевых подключений.

Сканировать можно с помощью специальных программ, называемых *сетевыми сканерами*.

Сканер локальной сети NetView – программа-заменитель Сетевого окружения Windows – ведет лог со списком машин, адресами и описаниями и регулярно проверяет его на наличие выключенных машин, ведет лог активных сетевых соединений (имеет функцию черного и белого списков). Имеет быстрый поисковик файлов в распределенных ресурсах, сканер портов и диапазона IP-адресов, встроенный PortListener – монитор, отслеживающий соединения на заданные порты (полезен для обнаружения IP-адресов, с которых проводятся попытки установить соединения на троянские порты или сканирования портов), а также возможность посылать сообщения по сети. Может составлять посегментную карту сети (через traceroute), а также визуализировать сеть (картинки для компьютеров, линии, фоновая текстура). Позволяет открывать компьютеры как по имени, так и по IP-адресам.

Сканер портов Nmap – утилита для обследования сетей и аудита защиты. В ней поддерживается сканирование на основе запроса отклика (определение жизнеспособности узлов), много методов сканирования портов (определение сер-висов, доступных на узлах), определение версий (какие приложения/службы работают на порте) и анализ трафика TCP/IP (fingerprinting, идентификация типа устройства или ОС узла). Имеются гибкие возможности спецификации целевых устройств и портов, сканирование на предмет ловушек и замаскированных угроз, сканирование SunRPC и многое другое. Для большинства платформ Unix и Windows поддерживаются режимы командной строки и графического интерфейса.

BitTorrent

BitTorrent (дословно «поток битов») – P2P-протокол, предназначенный для обмена файлами через Интернет. BitTorrent был создан программистом Брэмом Коэном.

Терминология

Анонс (*announce*) – обращение клиента к трекеру. При каждом анонсе клиент передает на трекер информацию об объемах им скачанного и отданного, а трекер передает клиенту список адресов других клиентов. Обращение клиента к трекеру происходит через определенные интервалы времени, которые определяются настройками клиента и трекера.

Веб-сид – HTTP-сервер, который может использоваться как источник дан-ных, выступая в роли сида.

Доступность (*availability, distributed copies* – распространенные копии) – количество полных копий файла, доступных клиенту. Каждый сид добавляет 1,0 к этому числу; личеры увеличивают доступность в зависимости от количества скачанного, которого нет у других личеров. К примеру, если на раздаче есть один сид и два личера, скачавшие по 50 % файла (скачанные части равны между собой), то доступность равна 1,50.

Заглохший (*choked* – заглохший) – клиент, обмен данными с которым заглох. Либо его канал на выход забит полностью и он не может ничего передать (достиг `max_uploads`), либо он является сидом и ему ничего не нужно получать.

Заинтересованный (*interested*) – участник, желающий получить фрагменты файла, имеющиеся у другого участника. Например, если у клиента А нет каких-то частей, которые есть у клиента Б, считается, что клиент А заинтересован в обмене с клиентом Б.

Индекс (*index*) – это список torrent-файлов (обычно включающий описания и другую информацию), управляемый веб-сайтом (индексатором) и доступный для поиска. Индексирующий сайт также может быть и трекером.

Лич, иногда личер (*leech* – пиявка) – пир, не имеющий пока всех сегментов, то есть продолжающий скачивание. Термин часто употребляется и в негативном смысле, который он имеет в других файлообменных сетях: пользователь, который отдает гораздо меньше, чем скачивает.

Отравленный торрент – ситуация, когда часть пиров раздает поврежденные сегменты.

Пир (*peer* – соучастник) – клиент, участвующий в раздаче.

Поскрестись, **Scrape-запрос** (*scrape* – скрести, царапать) – процесс, аналогичный анонсу, но клиент запрашивает только статистику торрента, информацию о подключенных клиентах и возможности с ними связаться для обмена.

Пренебрегающий (*snubbed*) – клиент, подключенный к получателю, но не посылавший ему данные уже более 60 секунд.

Раздача (seeding) – процесс распространения файла по протоколу BitTorrent.

Рейтинг (share ratio) – отношение отданного к скачанному.

Рой (swarm) – совокупность всех пиров, участвующих в раздаче.

Сегмент, часть (part – часть) – все файлы для передачи делятся на небольшие куски – сегменты, которые затем передаются по сети в произвольном порядке для оптимизации обмена.

Сид, иногда сидер (*seeder* – сеятель) – пир, имеющий все сегменты распространяемого файла, то есть либо начальный распространитель файла, либо уже скачавший весь файл и оставшийся на раздаче.

Суперсид (Super seeding) – специальный режим раздачи в некоторых BitTorrent-клиентах, пытающийся минимизировать количество данных, которое отдаст раздающий до появления первого скачавшего. Суперсид предлагает каждому пиру скачать только один сегмент файла, которого еще нет у других пиров. Затем сид не дает этому пиру следующих сегментов, пока не получит от других пиров подтверждения, что они тоже получили этот сегмент. Таким образом, суперсид пытается избежать повторной отдачи одних и тех же сегментов и старается отдавать сегменты только тем пирам, которые активно передают их другим.

Трекер (англ. *tracker*) – специализированный сервер, работающий по протоколу HTTP. Трекер нужен для того, чтобы клиенты могли найти друг друга. Фактически на трекере хранятся IP-адреса, входящие порты клиентов и хэш-суммы, уникальным образом идентифицирующие объекты, участвующие в заках. По стандарту, имена файлов на трекере не хранятся, и узнать их по хэш-суммам нельзя. Однако на практике трекер часто помимо своей основной функции выполняет и функцию небольшого веб-сервера. Такой сервер хранит файлы мета-данных и описания распространяемых файлов, предоставляет статистику закачек по разным файлам, показывает текущее количество подключенных пиров и пр.

Хеш (hash) – строка буквенно-цифровых символов в torrent-файле, которую используют клиенты для проверки передаваемых данных. Каждая часть после получения сначала проверяется на совпадение хеша. Если проверка не удалась, данные отбрасываются и запрашиваются еще раз.

Passkey – аутентификатор пользователя на неанонимных трекерах. Содержится в скачиваемом torrent-файле. Таким образом, если кто-то получит доступ к torrent-файлу, он сможет работать с трекером от имени этого пользователя. Трекер может изменить passkey по запросу пользователя, но при этом необходимо будет перескачать все прошлые torrent-файлы (или вручную отредактировать их), чтобы иметь возможность и дальше раздавать скачанные файлы.

URL анонса (announce URL) – адрес трекера, к которому клиент делает анонс. Во многих клиентах называется «Tracker URL». Может включать «pass-key» – уникальный код, назначаемый трекером для аккаунта пользователя, помогающий идентифицировать его на трекере (добавляется к URL анонса в самом torrent-файле при скачивании).

Общие особенности BitTorrent

1. Отсутствие очередей на загрузку.
2. Файлы закладываются небольшими сегментами; чем менее доступен сегмент, тем чаще он будет передаваться. Таким образом, присутствие в сети «сидера» с полным файлом для загрузки необязательно – система распределяет сегменты между «пирами», чтобы в последующем они могли обмениваться недостающими сегментами.
3. Клиенты (*peers*) обмениваются сегментами непосредственно между собой, по принципу «ты – мне, я – тебе».
4. Закачанные сегменты становятся немедленно доступны другим клиентам.
5. Контролируется целостность каждого сегмента.
6. В качестве объекта загрузки могут выступать несколько файлов (например, содержимое каталога).

Протоколы и порты

Клиенты соединяются с трекером по протоколу TCP.

Клиенты соединяются друг с другом, используя протокол TCP.

Номера портов не фиксированы в спецификации протокола и могут изменяться при необходимости. Более того, в данный момент большинство трекеров используют обычный HTTP порт 80, а для клиентов рекомендуется выбрать случайный входящий порт.

DHT-сеть в BitTorrent-клиентах использует протокол UDP.

Примечание. DHT (Distributed hash table) – это протокол, позволяющий бит-торрентклиентам находить друг друга без использования трекера. Клиенты с поддержкой DHT образуют общую DHT сеть и помогают друг другу найти участников одних и тех же раздач. Поддержка DHT есть в клиентах Mainline, rTorrent, KTorrent, BitSpirit и BitComet. В Azureus есть собственная реализация DHT, то есть Azureus-клиенты образуют свою собственную отдельную DHT сеть.

PEX (Peer exchange) – это расширение БТ протокола для обмена списками участников. PEX реализуется как дополнительные сообщения между клиентами, уже соединенными между собой для обмена сегментами файла по обычному БТ протоколу.

В отличие от трекера и DHT, PEX может быть только вспомогательным средством получения пилов, так как он не может помочь подключиться к раздаче новым пирам. PEX есть в клиентах Azureus, BitComet, rTorrent и BitTornado, причем в каждом клиенте он реализован по-своему, поэтому PEX между собой могут пользоваться только одинаковые клиенты. Начиная с третьей версии Azureus (Vuze) может обмениваться PEX с uTorrent и BitTorrent.

Функции

И DHT, и PEX фактически выполняют основную функцию трекера – помогают участникам файлообмена узнать друг о друге. Они могут:

1. Помочь участникам быстрее друг друга найти.

Например, на раздаче есть пир X с недоступным портом. К раздаче подключается пир Z, который сам начать соединение к X не может, и вынужден ждать, пока X о нем узнает сам. X только что обращался к трекеру, и в следующий раз собирается это сделать через час.

Но вот пир Y в очередной раз обращается к трекеру и узнает про нового пира Z. При этом Y сам давно уже соединен и занимается файлообменом с X, по-этому он через PEX сообщает X адрес этого нового пира. Теперь X может начать соединение к Z.

2. Снизить нагрузку на трекер.

Некоторые клиенты, например Azureus, получая адреса пиров через DHT или PEX, реже обращаются за списком пиров на трекер.

3. Поддерживать участников вместе в периоды недоступности трекера.

Известно, что если трекер является единственным источником информации о пирах, то при его неработоспособности раздача постепенно останавливается. Клиенты помнят уже известные списки адресов других пиров, но постепенно эти списки устаревают – некоторые пользователи отключаются от раздачи, у некоторых меняется IP-адрес, а новые пользователи не могут подключиться к раздаче вообще.

PEX позволяет несколько замедлить процесс распада роя участников, а DHT позволяет полностью заменить трекер, то есть даже подключаться к раздаче новым участникам.

4. DHT позволяет раздавать сегменты без трекера.

Такая раздача называется trackerless. Торрент для нее создается без адреса трекера, и клиенты друг друга находят через DHT-сеть.

При участии в trackerless раздачах БТ клиенты приобретают определенное сходство с eMule, использующим сеть KAD.

Механизм работы DHT

Реализация распределенной сети в БТ-клиентах основана на варианте DHT, называемом Kademlia. DHT (Distributed hash table) означает децентрализованную распределенную систему для объединения большого количества постоянно исчезающих и появляющихся узлов и эффективной передачи сообщений между ними. На основе DHT-структур строят разные более сложные системы, такие как P2P-файлообмен, кооперативное веб-кеширование, DNS-сервисы и т.п.

DHT использует UDP-протокол. БТ-клиенты слушают тот же UDP номер порта, который они используют для входящих TCP-соединений. Если активно используется DHT, то открытие этого UDP-порта для доступа снаружи желательно, но не обязательно – DHT будет работать и так.

Каждый подключенный БТ-клиент является в DHT-сети отдельным узлом. У него есть свой уникальный ID (идентификатор), случайно выбираемый из того же 160-битного пространства, что и infohash'bi торрентов.

Каждый узел хранит таблицу маршрутизации, содержащую контактную информацию о многих «ближайших» к нему узлах и о нескольких более далеких. «Близость» двух узлов вычисляется из «сходства» их ID и не имеет никакого отношения к их географической близости. Когда узел хочет найти пиров для какой-то раздачи, он сравнивает infohash этой раздачи с ID известных ему узлов и затем посылает запрос тому узлу, чей ID наиболее похож на этот infohash. Тот узел возвращает ему адрес узла, чей ID еще ближе к infohash торрента.

Тогда наш узел посылает запрос тому новому узлу и получает от него адрес следующего узла, чей ID еще более похож на infohash торрента.

Таким образом, запросы от клиентов, участвующих в раздаче торрента с определенным infohash, постепенно стекаются к узлам, чьи ID наиболее похожи на этот infohash. Эти узлы помнят предыдущие запросы, и всем следующим запрашивающим узлам вернут адреса предыдущих пиров с той же раздачи.

Private key

На публичных (открытых) трекерах, где каждый желающий может скачать торрент и участвовать в раздаче, DHT и PEX служат на благо всех участников.

Частным (закрытым) трекерам в первую очередь важно, чтобы в раздачах могли участвовать только зарегистрированные пользователи и чтобы они соблюдали определенные правила. При первом обращении клиента частный трекер имеет возможность не допустить его к раздаче, просто не сообщая ему адреса других клиентов-участников. Поэтому для закрытого трекера важно, чтобы клиенты не получали эти адреса через DHT/PEX.

DHT и PEX появились в клиентах Azureus и BitComet примерно летом 2005 г. Администраторы многих частных трекеров были недовольны такой новой функциональностью и поэтому стали запрещать на трекере эти новые версии клиентов.

Тогда разработчики клиентов предложили новый ключ внутри торрент-файла: private. Если он равен 1, то клиент обязан для этого торрента автоматически отключать DHT/PEX независимо от желания пользователя. Такой торрент называют Secure Torrent.

Практически все современные частные трекеры сами принудительно вставляют private:1 во все торренты, выкладываемые на трекере, а также запрещают несколько устаревших версий клиентов, поддерживающих DHT или PEX, но еще не знающих про private key. Пользователи трекера просто не могут на раздачах использовать DHT/PEX.

Отметим, что присутствие private key изменяет infohash торрента, поэтому вырезать его из торрент-файла бесполезно – другие клиенты измененный торрент все равно не признают.

DHT и статистика

Примечание. Этот раздел касается только частных трекеров, на которых private key в торренты принудительно не вставляется, и на некоторых раздачах (в зависимости от того, вставил ли раздающий сам в торрент private key) можно использовать DHT и PEX.

Часто встречается мнение, что включенный в клиенте DHT влияет на учет статистики клиента трекером, например «раздавал через DHT, значит статистика шла мимо трекера». Это неверно.

Во-первых, DHT/PEX используется только для получения адресов пиров. Ни файлообмена, ни какого-либо учета статистики по ним не ведется. Клиент рапортует статистику скачанного и отданного только на трекер.

То есть «раздавал через DHT» фактически означает «о некоторых (или о всех) пирах получил информацию по DHT, и, вероятно, некоторые пиры тоже нашли меня через DHT».

Во-вторых, хотя клиенты обычно и знают, откуда ими получены адреса пиров, ни один клиент не разделяет трафик на «полученный/отданный DHT пирам» и «полученный/отданный пирам, полученным от трекера». Даже при желании это было бы клиенту сделать затруднительно – некоторые пиры могут быть получены и от трекера, и через DHT или PEX, и часто клиент не знает, как его адрес получил пир, сам начинающий к нему соединение.

Клиент рапортует трекеру суммарные данные об объемах им скачанного и отданного всем пирам, с которыми он общался, независимо от того, узнал клиент об отдельных пирах через трекер, DHT или PEX, или тот пир вообще начал соединение сам. То есть даже если из-за DHT/PEX на раздаче появятся «левые» пользователи (не обращающиеся к трекеру), клиент все равно сообщит на трекер все, что у них скачал и отдал.

Правильный учет статистики зависит только от состояния трекера: работает трекер – статистика учитывается, не работает – не учитывается. Только в случае длительно неработающего трекера DHT/PEX может играть косвенную роль, не давая постепенно затухнуть файлообмену на такой «раздаче без учета статистики».

Кроме того, протокол UDP используется UDP-трекерами (не поддерживается всеми клиентами и не является официальной частью протокола) и для соединения клиентов друг с другом через UDP NAT Traversal (используется только в клиенте BitComet и не является официальной частью протокола).

Файл метаданных

Для каждого распространяемого файла создается файл метаданных с расширением .torrent, который содержит следующую информацию:

- URL трекера;
- общую информацию о закачиваемом файле (имя, длину и пр.);
- контрольные суммы (точнее, хэш-суммы SHA1) сегментов закачиваемого файла.

Файлы метаданных могут распространяться через любые каналы связи – например, они (или ссылки на них) могут выкладываться на веб-серверах, размещаться на домашних страницах пользователей сети, рассылаться по электронной почте, публиковаться в блогах или новостных лентах RSS.

Клиент начинает загрузку, получив каким-либо образом файл с метаданными, в котором есть ссылка на трекер.

Подробнее про BitTorrent-трекер

BitTorrent-трекер – веб-сервер, осуществляющий координацию BitTorrent клиентов.

Координация клиентов – основная функция BitTorrent-трекера.

Каждый клиент периодически обращается к трекеру с запросом, в котором указаны:

- info_hash – уникальный хеш-торрент файла;
- port – TCP-порт, на котором клиент ждет соединений от других клиентов; количество данных, которыми клиент успел обменяться с другими клиентами, и некоторая другая информация.

Такое обращение представляет собой обычный GET HTTP запрос, в котором информация закодирована с помощью специального протокола Bencode.

Если в BitTorrent-клиенте открыто несколько активных торрент-файлов, то для каждого используется отдельный запрос, никак не связанный с другими. Интервалы между запросами по каждому отдельно взятому торренту зависят от настроек клиента и обычно составляют от 30 до 60 минут.

BitTorrent-трекер, используя info_hash, составляет списки IP-адресов и портов, участвующих в каждой отдельной раздаче клиентов. Каждому клиенту в ответ на очередной запрос трекер возвращает такой список, и клиент использует его для установления соединений с другими клиентами.

Роль трекера

Таким образом, трекер «связывает» клиентов друг с другом, но напрямую не участвует в обмене данными раздаваемых файлов. Более того, в принципе трекер может работать, не имея никакой информации об этих файлах, поскольку клиенты сообщают ему только `info_hash`.

Трекер считается «слабым» местом системы BitTorrent, поскольку при его отключении новые клиенты просто не могут друг друга «найти». При этом уже участвующие в раздаче клиенты могут некоторое время продолжать файлообмен, постепенно теряя тех, кто отключился или у кого поменялся динамический IP-адрес.

Для разрешения этой проблемы могут использоваться резервные трекеры или специальный бестрекерный протокол DHT.

Дополнительные функции

Большинство реализаций трекеров, кроме координации клиентов, также имеют дополнительные функции, доступные пользователям с браузерами через обычный веб-сервер, часто с другим адресом или портом, чем у интерфейса для запросов клиентов.

Так, практически все трекеры отображают статистику раздач, то есть количества сидов и пиров на каждой раздаче, общие объемы переданных между клиентами данных и другую доступную им информацию.

Часто трекеры используются для хранения торрент-файлов и их описаний.

Частные трекеры

Частный (англ. *private*) трекер – это трекер, ограничивающий доступ пользователям, обычно требованием регистрации учетной записи.

Одной из причин появления частных трекеров является асимметричность пользовательских каналов доступа к Интернету. Средний пользователь за время скачивания торрента успевает отдать другим пользователям незначительную по объему часть данных, при этом многие пользователи после получения нужного файла сразу закрывают торрент-клиент. В результате доступность торрента снижается, и на первоначальных распространителей файла ложится неоправданно большая нагрузка.

Как правило, частные трекеры записывают статистику каждого пользователя по количеству «скачанного» и «отданного» другим пользователям и требуют поддержания некоторого минимального отношения этих двух величин. Как результат, доступность и скорость скачивания торрентов на частном трекере обычно выше, чем на открытом.

Для опознания запросов от клиента (программы) пользователя трекер либо использует IP-адрес пользователя, либо уникальный для каждого пользователя `passkey`, добавляемый трекером в `announce URL`-торрент файла при скачивании пользователем.

Реализации трекеров

Существуют разные реализации трекеров, например как отдельный веб-сервер в виде модуля для стороннего HTTP-сервера (например, Apache) или в виде движка сайта, написанного, например, на PHP или JSP.

Работа без трекера

В новых версиях протокола были разработаны бестрекерные (англ. *trackerless*) системы, которые решают некоторые из предыдущих проблем. Отказ трекера в таких системах не приводит к автоматическому отказу всей сети.

Начиная с версии 4.2.0 официального клиента, в нем реализована функция бестрекерной работы, базирующаяся на протоколе Kademlia. В таких системах трекер доступен децентрализованно, на клиентах, в форме распределенной хеш-таблицы.

На данный момент еще не все клиенты используют совместимый друг с другом протокол. Совместимы между собой BitComet, pTorrent, KTorrent и официальный клиент BitTorrent. Azureus также имеет режим бестрекерной работы, но его реализация отличается от официальной, вследствие чего он не может работать через DHT с вышеперечисленными клиентами.

Принцип работы протокола BitTorrent

Перед началом загрузки клиент подсоединяется к трекеру, сообщает ему свой адрес и хэш-сумму запрашиваемого файла, на что в ответ клиент получает адреса других клиентов, скачивающих или раздающих этот же файл. Далее клиент периодически информирует трекер о своем прогрессе и получает обновленный список адресов.

Клиенты соединяются друг с другом и обмениваются сегментами файлов без непосредственного участия трекера. Для эффективной работы сети BitTorrent необходимо, чтобы как можно больше клиентов были способны принимать входящие соединения.

Неправильная настройка NAT или файрвола могут этому мешать.

При соединении клиенты сразу обмениваются информацией об имеющихся у них сегментах. Клиент, желающий скачать сегмент, посылает запрос, и если второй клиент готов отдавать, получает этот сегмент. После этого клиент проверяет контрольную сумму сегмента и оповещает всех присоединенных пиров о наличии у него этого сегмента.

Каждый клиент имеет возможность временно блокировать отдачу другому клиенту (англ. *choke*). Это делается для более эффективного использования канала отдачи. Кроме того, при выборе — кого разблокировать, предпочтение отдается пирам, которые сами передали этому клиенту много сегментов. Таким образом, пиры с хорошими скоростями отдачи поощряют друг друга по принципу «ты — мне, я — тебе».

Суперсид (англ. *super seeding*) – метод, реализованный в тех клиентах BitTorrent, авторы которых пытаются минимизировать объем данных до первого завершения загрузки пира. Метод был задуман Джоном Хофманом и впервые был осуществлен в клиенте «BitTornado» в середине 2003 г.

Суперсид является изменением в поведении сидеров и не может быть осуществлен без нарушения протокола BitTorrent. Тем не менее это не утверждается ни разработчиком протокола, Брэмом Коэном, ни в официальном клиенте.

Суперсид действует, когда имеется только один сидер. Фактически он заставляет пользователей делиться скачанными данными. Клиент, отдавший одному участнику какую-либо часть файла, прекращает передачу данных до тех пор, пока не обнаружит в сети вторую копию этой части. Поскольку многие участники не имеют возможности или не желают делать доступными для других скачанные данные, общая скорость отдачи падает. В обычном режиме же, в отличие от суперсид, каждый всегда получает запрошенную информацию, если она имеется, поэтому в нем средняя скорость передачи данных выше.

Суперсид эффективен при раздачах с большим количеством качающих. При небольшой группе людей (2–3 человека) их клиенты в силу разных причин могут отказаться устанавливать связь друг с другом. В этом случае включенный суперсид заставит клиентов ждать, пока данные станут доступны. Когда число качающих велико (10–20 человек), достаточно отдать каждую часть файла только один раз, тогда клиенты смогут обмениваться ею между собой и каждый из них получит файл целиком.

Задания

Задание 1. Исследуйте локальную сеть.

1. Определите локальный IP-адрес из командной строки (Пуск / Стандартные / Командная строка) при помощи утилиты **ipconfig**, а также определите адрес сети и маску подсети.

2. Просканируйте локальную сеть для определения присутствующих в ней узлов **nmap -sP адрес_сети/количество_бит_в_маске**

Например, для сети 172.16.72.0 и маски 255.255.255.0 это

nmap -sP 172.16.72.0/24

3. Запустите сканер локальной сети – NetView.

4. Разместите все найденные ранее узлы в одном хост-листе. Для этого сначала добавьте их вручную при помощи кнопки *Add host* на панели инструментов, а затем объедините (*Setting / Autosplit / Send all to Default* в меню). Чтобы не добавлять все узлы вручную, можно воспользоваться инструментом *Network scanner*, после завершения сканирования – выделить все узлы и выбрать *Add selected to hostlist*.

5. Просмотрите свойства узла в списке (контекстное меню узла / *Properties*).

6. Просмотрите активные подключения к компьютеру (*Tools / Net Watcher*).

7. Просканируйте открытые *TCP*-порты контролера домена:

– активизируйте инструмент «Сканер сети» (*Tools / Network Scanner*);

– введите в поле *IP diapason* – *<IP-адрес контролера домена>*;

– установите диапазон сканирования *TCP*-портов (*TCP Range*) с 1 по 65536 и запустите сканирование кнопкой *Start*.

8. Аналогично просканируйте открытые *UDP*-порты контролера домена.

9. Просмотрите статистику работы протокола *IP* (*IP logger*).

10. Найдите общедоступные ресурсы сети (*Tools / Resources scanner*).

Задание 2. Исследуйте удаленную систему и сохраните результаты исследования (*IP*-адрес компьютера, открытые порты с указанием сервисов, работающих на этих компьютерах) в файл с именем *scan.txt*.

1. Получите информацию об использовании сканера *Nmap*:

– откройте консоль (Пуск/Выполнить/cmd);

– перейдите в каталог со сканером;

– введите команду *nmap.exe*.

Если сканеру не передаются параметры, то он просто выдает справку об использовании. В общем случае синтаксис использования этого сканера следующий: *Nmap [-тип сканирования][-параметры] объект_сканирования*;

2. Найдите открытые порты вашего компьютера. Для этого введите команду **nmap** с параметром *127.0.0.1*.

На экран через некоторое время будет выдана информация об открытых портах из диапазона *1...1677*.

Пример выведенной информации представлен на рис. 6.1.

```
Interesting ports on 172.21.5.240:
(116 1657 ports scanned but not shown below are in state: closed)
PORT STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
554/tcp   open  rtsp
1720/tcp  open  H.323/Q.931
1723/tcp  open  pptp
1755/tcp  open  uns
2000/tcp  open  callbook
3389/tcp  open  ms-term-serv
5800/tcp  open  unc-http
5900/tcp  open  unc
MAC Address: 00:14:2A:6E:BE:2A (Elitegroup Computer System Co.)
Nmap finished: 1 IP address <1 host up> scanned in 23.625 seconds
```

Рис. 6.1. Пример вывода программы *nmap*

3. Просканируйте все порты (с 1 по 65536) любого другого узла локальной сети (не используйте адрес, принадлежащий вашему же узлу). Для этого введите команду **nmap** с параметром **-p1-65535**:

nmap -p1-65535 1Р-адрес_компьютера

4. Просканируйте тот же узел с попыткой определения версии сервисов, работающих на тех или иных портах:

nmap -sV 1Р-адрес_компьютера

5. Определите операционную систему узла:

nmap -O 1Р-адрес_компьютера

Результат детектирования можно найти в строке OS Details.

6. Просканируйте узел с обходом брандмауэра:

nmap -f 1Р-адрес_компьютера

7. Просканируйте узел с выводом отправленных и полученных пакетов:

nmap -packet-trace 1Р-адрес_компьютера.

8. Повторите пункты с 3-го по 7-й для сервера в сети Интернет в соответствии с вашим вариантом.

Варианты заданий

1. mail.pnzgu.ru
2. moodle.pnzgu.ru
3. elib.pnzgu.ru
4. gateway.pnzgu.ru
5. gitlab.pnzgu.ru
6. domoodle.pnzgu.ru
7. zabb.pnzgu.ru
8. directum.pnzgu.ru
9. online.pnzgu.ru
10. eios-lk.pnzgu.ru
11. olimp.pnzgu.ru
12. post.pnzgu.ru

Задание 3. Изучите принципы обмена информацией по протоколу BitTorrent.

1. Используя <http://ru.wikipedia.org/wiki/BitTorrent> и теоретическую часть, изучите принцип работы протокола BitTorrent, терминологию, связанную с ним. Выделите особенности работы с протоколом.

2. Загрузите и установите три программы-клиента обмена информацией по протоколу BitTorrent: Vuze (Azureus), Transmission и третью по собственному выбору.

3. Проведите регистрацию на неиндексируемом торрент-трекере (информация о раздачах которого недоступна без регистрации и не

индексируется поисковыми системами) по выбору (можно использовать, например, uniongang.club, exkinoray.fun, nmmclub.to и другие. Адреса можно найти на <http://www.toptracker.ru/>).

4. С помощью выше указанных программ-клиентов загрузите файл (небольшого размера). Сравните особенности работы каждой программы. Укажите достоинства и недостатки. Сделайте выбор в пользу использования той или иной программы, мотивируя свой выбор.

Примечание. После работы с программами-клиентами удалите их с компьютеров компьютерного класса.

5. Добавьте в отчет описание процесса регистрации на торрент-трекерах и работы с программами-клиентами.

Контрольные вопросы

1. Что понимается под термином «компьютерная атака»?
2. Что понимается под термином «хост»?
3. Что понимается под термином «порт»?
4. Как вы опишете принципы работы и возможности протокола TCP?
5. Как вы опишете принципы работы и возможности протокола UDP?
6. Как вы опишете принципы работы и возможности протокола SSL?
7. С какой целью реализуется процесс сканирования портов?
8. Какие известные порты вы знаете? Опишите их.
9. Каковы функции сканера локальной сети NetView? Опишите их.
10. Каково назначение утилиты Nmap?
11. Что понимается под терминами «анонс», «пир», «сид», «трекер»?
12. В чем заключаются принципы работы протокола BitTorrent?

Глоссарий

А

Анонс

Обращение клиента к трекеру.

Б

**Брандмауэр
(межсетевой экран)**

Система, которая выполняет роль защитной стены между глобальным Интернетом и устройством пользователя. Брандмауэр фильтрует всю входящую информацию и не пропускает вредоносный контент и вирусы.

В

Веб-сид

HTTP-сервер, который может использоваться как источник данных, выступая в роли сیدا.

Д

Доступность

Количество полных копий файла, доступных клиенту.

З

Заинтересованный

Участник, желающий получить фрагменты файла, имеющиеся у другого участника.

И

Индекс

Список torrent-файлов (обычно включающий описания и другую информацию), управляемый веб-сайтом (индексатором) и доступный для поиска.

Инкапсуляция

Включение всего пакета одного протокола (то есть его заголовки и данные) внутрь пакета другого протокола в качестве передаваемой информации.

К

**Криптографический
алгоритм**

Набор правил, по которым происходит шифровка и последующая расшифровка информации, передающейся в Интернете.

Л

Лич

Клиент, не имеющий всех сегментов и продолжающий скачивание файлов.

**Луковая
маршрутизация**

Технология анонимного обмена информацией через компьютерную сеть. Сообщения неоднократно шифруются и потом отсылаются через несколько сетевых узлов, называемых луковыми маршрутизаторами. Каждый маршрутизатор удаляет слой шифрования, чтобы открыть трассировочные инструкции и отослать сообщения на следующий маршрутизатор, где все повторится. Таким образом, промежуточные узлы не знают источник, пункт назначения и содержание сообщения.

М

Маска подсети

32-битное число, служащее битовой маской для разделения сетевой части (адреса подсети) и части хоста IP-адреса. Состоит из последовательности от 0 до 32 двоичных единиц, после которых остаток разрядов представляют двоичные нули. Их смешение недопустимо. Устройства в одной подсети имеют одинаковый адрес подсети и передают данные на канальном уровне.

О

Отравленный торрент

Ситуация, когда часть пиров раздает поврежденные сегменты.

П

Пир

Клиент, участвующий в раздаче.

Порт

Числовой идентификатор программы или процесса, которые обслуживают сетевые соединения на заданном сетевом адресе (IP-адресе).

Прокси-сервер

Промежуточный сервер (комплекс программ) в компьютерных сетях, выполняющий роль посредника между пользователем и целевым сервером, позволяющий клиентам как выполнять косвенные запросы (принимая и передавая их через прокси-сервер) к другим сетевым службам, так и получать ответы.

Р

Раздача

Процесс распространения файла по протоколу BitTorrent.

Рой

Совокупность всех пиров, участвующих в раздаче.

С

Сетевой мост

Сетевое устройство, которое соединяет две или более разные сети и позволяет им обмениваться данными друг с другом.

Сетевой сканер

Программное решение, позволяющее проводить диагностику и мониторинг информационной инфраструктуры в настоящем времени с целью оценки безопасности и выявления уязвимостей.

Сид

Пир, имеющий все сегменты распространяемого файла, то есть либо начальный распространитель файла, либо уже скачавший весь файл и оставшийся на раздаче.

<i>Сканирование портов</i>	Процесс обнаружения прослушивающих приложений путем активного опроса сетевых портов компьютера или другого сетевого устройства.
<i>Скрипт</i>	Набор команд, прописанных в коде, которые необходимы для выполнения задачи. Скрипты могут быть как небольшими, выполняя простые задачи, так и объемными, которые анализируют данные и представляют готовый результат на сайте.
<i>Сокет</i>	Виртуальная конструкция из IP-адреса и номера порта. Ее придумали для того, чтобы разработчикам было проще писать код, а программы могли передавать данные друг другу даже в пределах одного компьютера.
Т	
<i>Трекеры</i>	Серверы, которые отслеживают, какие пользователи скачивают и раздают определенные торренты, и помогают им устанавливать соединение друг с другом.
Х	
<i>Хеш</i>	Математический алгоритм, преобразовывающий произвольный массив данных в состоящую из букв и цифр строку фиксированной длины.
Ч	
<i>Частный трекер</i>	Это трекер, ограничивающий доступ пользователям, обычно с требованием регистрации учетной записи.

Список литературы

1. Дьяконов А. А., Гринева С. В. Рост количества пользователей российского сегмента сети Tor: выявление возможных причин // Инновационные направления развития в образовании, экономике, технике и технологиях : национальная с междунар. участием науч.-практ. конф., Ставрополь, 18–20 мая 2021 года / под общ. науч. ред. С. П. Бабенышева. Ставрополь : Ставролит, 2021. С. 361–365. URL: <https://www.elibrary.ru/item.asp?id=46441205>
2. Лушников Н. Д., Альтерман А. Д., Парфенова А. С., Ключек М. С. Tor как технология анонимности // Молодой исследователь: вызовы и перспективы : сб. ст. по материалам CXLVIII Междунар. науч.-практ. конф., Москва, 06.01.2020. М. : Интернаука, 2020. Т. 1. С. 454–457. URL: <https://www.elibrary.ru/item.asp?id=41812733>
3. Лыткин К. А. Использование протокола Bittorrent для обмена и синхронизации файлов между компьютерами // Тинчуринские чтения – 2022 «Энергетика и цифровая трансформация» : сб. ст. по материалам Междунар. молодежной науч. конф. Казань, 27–29 апреля 2022 г. : в 3 т. / Под общ. ред. Э. Ю. Абдуллазянова. Казань : Казан. гос. энергетич. ун-т, 2022. С. 83–86. URL: <https://www.elibrary.ru/item.asp?id=49225814>
4. Мостовой И. Л., Рябова Е. А., Малахов С. В., Мезенцева Е. М. Блокировка браузера Tor // Актуальные проблемы науки и техники : материалы I Междунар. науч.-техн. конф., Сарапул, 20–22 мая 2021 г. – Ижевск : Изд-во УИР ИжГТУ им. М. Т. Калашникова, 2021. С. 321–324. URL: <https://www.elibrary.ru/item.asp?id=47317144>
5. Новосельцева А. В., Ключев С. Г. Современные методы атак деанонимизации на сеть TOR // Прикаспийский журнал: управление и высокие технологии. 2020. № 1 (49). С. 155–161. URL: <https://www.elibrary.ru/item.asp?id=42883205>
6. Паниткин Д. В. Риск-модели процессов атак «сканирование портов» компьютерных систем: синтез и управление : дис. ... канд. техн. наук. Воронеж, 2008. 136 с.
7. Шаханова М. В., Солоненко Д. Ю., Шаханова Э. С. Разработка системы защиты от атаки сканирования сетевых портов // Международный журнал информационных технологий и энергоэффективности. 2023. Т. 8, № 5–1 (31). С. 67–74. URL: <https://www.elibrary.ru/item.asp?id=53802774>
8. Betka A., Toumi A., Terki A., Hamiane M. An efficient metaheuristic method based on the BitTorrent communication protocol (EM-BT) // Evolutionary Intelligence. 2023. Vol. 16, No. 4. P. 1115–1134. URL: <https://www.elibrary.ru/item.asp?id59683821>

Учебное издание

Технологии Интернета

Часть 2

Составители:

Акифьев Илья Владимирович,
Коровина Людмила Владимировна

Редактор *Л. Ю. Зимина*
Технический редактор *В. А. Гуцин*
Компьютерная верстка *В. А. Гуцина*

Подписано в печать 20.03.2025. Формат 60×84¹/₁₆.

Усл. печ. л. 2,96.

Заказ № 49. Тираж 30.

Издательство ПГУ.
440026, г. Пенза, ул. Красная, 40.
Тел.: (8412) 66-60-49, 66-67-77; e-mail: iic@pnzgu.ru

Вниманию авторов!

Издательство ПГУ выпускает учебную, научную и художественную литературу, презентационную и акцидентную продукцию, а также полноцветные юбилейные и мемориальные издания в соответствии с ГОСТ 7.60–2003.

Издательство ПГУ принимает к изданию рукописи, подготовленные с использованием текстового редактора Microsoft Word for Windows версий **2003 и выше**. Формат – А4, основной шрифт – Times New Roman, 14–16 pt через одинарный интервал (минимальный размер шрифта в таблицах и сносках – 12,5 pt). Тип файла в электронном виде – doc, docx.

Работа должна содержать индекс УДК, аннотацию.

Аннотация (ГОСТ 7.86–2003, ГОСТ 7.9–1995) включает характеристику основной темы, проблемы объекта, цели работы и ее результаты. В аннотации указывают, что нового несет в себе данный документ в сравнении с другими, родственными по тематике и целевому назначению. Аннотация может включать сведения о достоинствах произведения. Текст аннотации начинают фразой, в которой сформулирована главная тема документа. Заканчивается аннотация читательским адресом.

Рисунки и таблицы должны быть размещены в тексте после ссылки на них (растровые рисунки предоставляются в виде отдельных файлов в формате jpg, BMP с разрешением 300 dpi, векторные рисунки в формате Corel Draw с минимальной толщиной линии 0,75 pt. Рисунки должны быть доступны для правки!). Рисунки должны сопровождаться подрисуночными подписями, на все рисунки и таблицы в тексте должны быть ссылки.

Формулы в тексте выполняются только в редакторе формул **MathType версия 5.0** и выше. Символы греческого и русского алфавита должны быть набраны прямо, нежирно; латинского – курсивом, нежирно; обозначения векторов и матриц – прямо, жирно; цифры – прямо, нежирно. Наименования химических элементов набираются прямо, нежирно. Эти же требования необходимо соблюдать и в рисунках.

В списке литературы **нумерация источников** должна соответствовать очередности ссылок на них в тексте ([1], [2], ...). Номер источника указывается в квадратных скобках. Требования к оформлению списка литературы на русские и иностранные источники (ГОСТ Р 7.0.5–2008): для книг – фамилия и инициалы автора, название, город, издательство, год издания, том, количество страниц; для журнальных статей, сборников трудов – фамилия и инициалы автора, название статьи, полное название журнала или сборника, серия, год, том, номер, страницы; для материалов конференций – фамилия и инициалы автора, название статьи, название конференции, город, издательство, год, страницы.

К материалам **должна** прилагаться следующая информация: фамилия, имя, отчество, контактные телефоны.

Контакты Издательства ПГУ: т.: (8412) 66-60-49, 66-67-77, e-mail: iic@pnzgu.ru



