HTTP-широко распространённый протокол передачи данных, изначально предназначенный для передачи гипертекстовых документов (то есть документов, которые могут содержать ссылки, позволяющие организовать переход к другим документам).

Аббревиатура HTTP расшифровывается как HyperText Transfer Protocol, «протокол передачи гипертекста». В соответствии со спецификацией OSI, HTTP является протоколом прикладного (верхнего, 7-го) уровня. Актуальная на данный момент версия протокола, НТТР 1.1, описана в спецификации RFC 2616.

Протокол НТТР предполагает использование клиент-серверной структуры передачи данных. Клиентское приложение формирует запрос и отправляет его на сервер, после чего серверное программное обеспечение обрабатывает данный запрос, формирует ответ и передаёт его обратно клиенту. После этого клиентское приложение может продолжить отправлять другие запросы, которые будут обработаны аналогичным образом.

Задача, которая традиционно решается с помощью протокола НТТР - обмен данными между пользовательским приложением, осуществляющим доступ к веб-ресурсам (обычно это веб-браузер) и веб-сервером. На данный момент именно благодаря протоколу НТТР обеспечивается работа Всемирной паутины.

Также НТТР часто используется как протокол передачи информации для других протоколов прикладного уровня, таких как SOAP, XML-RPC и WebDAV. В таком случае говорят, что протокол НТТР используется как «транспорт».

АРІ многих программных продуктов также подразумевает использование НТТР для передачи данных JSON.

сами данные при этом могут иметь любой формат, например, XML или

Как правило, передача данных по протоколу НТТР осуществляется через ТСР/ІР- соединения. Серверное программное обеспечение при этом обычно использует ТСР- порт 80 (и, если порт не указан явно, то обычно клиентское программное обеспечение по умолчанию использует именно 80-й порт для открываемых HTTP-соединений), хотя может использовать и любой другой.

Как отправить НТТР-запрос?

Самый простой способ разобраться с протоколом НТТР это попробовать обратиться к какому-нибудь веб-ресурсу вручную. Представьте, что вы браузер, и у вас есть пользователь, который очень хочет прочитать статьи Анатолия Ализара.

Предположим, что он ввёл в адресной строке следующее:

http://alizar.habrahabr.ru/

Соответственно вам, как веб-браузеру, теперь необходимо подключиться к веб-серверу по адресу alizar.habrahabr.ru.

Для этого вы можете воспользоваться любой подходящей утилитой командной строки. Например, telnet:

telnet alizar.habrahabr.ru 80

Сразу уточню, что если вы вдруг передумаете, то нажмите Ctrl + «]», и затем ввод - это позволит вам закрыть НТТР-соединение. Помимо telnet можете попробовать пс (или ncat) - по вкусу.

После того, как вы подключитесь к серверу, нужно отправить НТТР-запрос. Это, кстати, очень легко - НТТР-запросы могут состоять всего из двух строчек.

Для того, чтобы сформировать НТТР-запрос, необходимо составить стартовую строку, а также задать по крайней мере один заголовок - это заголовок Host, который является обязательным, и должен присутствовать в каждом запросе. Дело в том, что преобразование доменного имени в IP-адрес осуществляется на стороне клиента, и, соответственно, когда Вы открываете ТСР-соединение, то удалённый сервер не обладает никакой информацией о том, какой именно адрес использовался для соединения: это мог быть, например, адрес alizar.habrahabr.ru, habrahabr.ru или m.habrahabr.ru отличаться. Однако фактически сетевое соединение во всех случаях открывается с узлом - и во всех этих случаях ответ может 212.24.43.44, и даже если первоначально при открытии соединения был задан не этот IP- адрес, а какое-либо доменное имя, то сервер об этом никак не информируется поэтому этот адрес необходимо передать в заголовке Host.

Стартовая (начальная) строка запроса для НТТР 1.1 составляется по следующей схеме: Метод URI HTТР/Версия

Чапример (такая стартовая строка может указывать на то, что запрашивается главная страница сайта):

GET / HTTP/1.1

Метод (в англоязычной тематической литературе используется слово method, а также иногда слово verb <<глагол>>>) представляет собой последовательность из любых символов, кроме управляющих и разделителей, и определяет операцию, которую нужно осуществить с указанным ресурсом. Спецификация НТТР 1.1 не ограничивает количество разных методов, которые могут быть использованы, однако в целях соответствия общим стандартам и сохранения совместимости с максимально широким спектром программного обеспечения как правило используются лишь некоторые, наиболее стандартные методы, смысл которых однозначно раскрыт в спецификации протокола.

URI (Uniform Resource Identifier, унифицированный идентификатор ресурса) - путь до конкретного ресурса (например, документа), над которым необходимо осуществить операцию (например, в случае использования метода GET подразумевается получение ресурса). Некоторые запросы могут не относиться к какому-либо ресурсу, в этом случае вместо URI в стартовую строку может быть добавлена звёздочка (астериск, символ «\*»).

Например, это может быть запрос, который относится к самому веб-серверу, а не какому- либо конкретному ресурсу. В этом случае стартовая строка может выглядеть так:

OPTIONS \* HTTP/1.1

Версия определяет, в соответствии с какой версией стандарта HTTP составлен запрос. Указывается как два числа, разделённых точкой (например 1.1).

Для того, чтобы обратиться к веб-странице по определённому адресу (в данном случае путь к ресурсу - это «/»), нам следует отправить следующий запрос:

GET / HTTP/1.1

Host: alizar.habrahabr.ru

При этом учитывайте, что для переноса строки следует использовать символ возврата каретки (Carriage Return), за которым следует символ перевода строки (Line Feed). После объявления последнего заголовка последовательность символов для переноса строки добавляется дважды.

Впрочем, в спецификации НТТР рекомендуется программировать НТТР-сервер таким бразом, чтобы при обработке запросов в качестве межстрочного разделителя воспринимался символ LF, а предшествующий символ CR, при наличии такового, Игнорировался. Соответственно, на практике большая часть серверов корректно обработает и такой запрос, где заголовки отделены символом LF, и он же дважды добавлен после объявления последнего заголовка.

Если вы хотите отправить запрос в точном соответствии со спецификацией, можете воспользоваться управляющими последовательностями \r и \n:

echo -en "GET / HTTP/1.1\r\nHost: alizar.habrahabr.ru\r\n\r\n" | ncat alizar.habrahabr.ru 80

Как прочитать ответ?

Стартовая строка ответа имеет следующую структуру:

HTТР/Версия Код состояния Пояснение

Версия протокола здесь задаётся так же, как в запросе.

Код состояния (Status Code) - три цифры (первая из которых указывает на класс состояния), которые определяют результат совершения запроса. Например, в случае, если был использован метод GET, и сервер предоставляет ресурс с указанным

идентификатором, то такое состояние задаётся с помощью кода 200. Если сервер сообщает о том, что такого ресурса не существует - 404. Если сервер сообщает о том, что не может

предоставить доступ к данному ресурсу по причине отсутствия необходимых привилегий у клиента, то используется код 403. Спецификация НТТР 1.1 определяет 40 различных кодов НТТР, а также допускается расширение протокола и использование дополнительных кодов состояний.

Пояснение к коду состояния (Reason Phrase) - текстовое (но не включающее символы CR и LF) пояснение к коду ответа, предназначено для упрощения чтения ответа человеком. Пояснение может не учитываться клиентским программным обеспечением, а также может отличаться от стандартного в некоторых реализациях серверного ПО.

После стартовой строки следуют заголовки, а также тело ответа. Например:

HTTP/1.1 200 OK

Server: nginx/1.2.1

Date: Sat, 08 Mar 2014 22:53:46 GMT

Content-Type: application/octet-stream

Content-Length: 7

Last-Modified: Sat, 08 Mar 2014 22:53:30 GMT

Connection: keep-alive

Accept-Ranges: bytes

Wisdom

Тело ответа следует через два переноса строки после последнего заголовка. Для определения окончания тела ответа используется значение заголовка Content-Length (B данном случае ответ содержит 7 восьмеричных байтов: слово «Wisdom» и символ переноса строки).

Но вот по тому запросу, который мы составили ранее, веб-сервер вернёт ответ не с кодом 200, а с кодом 302. Таким образом он сообщает клиенту о том, что обращаться к данному ресурсу на данный момент нужно по другому адресу.

Смотрите сами:

HTTP/1.1 302 Moved Temporarily

Server: nginx

Date: Sat, 08 Mar 2014 22:29:53 GMT

Content-Type: text/html

Content-Length: 154

Connection: keep-alive

Keep-Alive: timeout=25

Location: http://habrahabr.ru/users/alizar/

<html>

<head><title>302 Found</title> </head>

<body bgcolor="white">

<center><h1>302 Found</h1></center>

<hr><center>nginx</center>

</body>

</html>

В заголовке Location передан новый адрес. Теперь URI (идентификатор ресурса) изменился на /users/alizar/, а обращаться нужно на этот раз к серверу по адресу habrahabr.ru (впрочем, в данном случае это тот же самый сервер), и его же указывать в заголовке Host.

То есть:

GET /users/alizar/ HTTP/1.1

Host: habrahabr.ru

В ответ на этот запрос веб-сервер Хабрахабра уже выдаст ответ с кодом 200 и достаточно большой документ в формате HTML.

Если вы уже успели вжиться в роль, то можете теперь прочитать полученный от сервера HTML-код, взять карандаш и блокнот, и нарисовать профайл Ализара - в принципе, именно этим бы на вашем месте браузер сейчас и занялся.

А что с безопасностью?

Сам по себе протокол НТТР не предполагает использование шифрования для передачи информации. Тем не менее, для НТТР есть распространённое расширение, которое реализует упаковку передаваемых данных в криптографический протокол SSL или TLS. Название этого расширения - HTTPS (HyperText Transfer Protocol Secure). Для HTTPS- соединений обычно используется ТСР-порт 443. HTTPS широко используется для защиты информации от перехвата, а также, как правило, обеспечивает защиту от атак вида man-in- the-middle - в том случае, если сертификат проверяется на клиенте, и при этом приватный ключ сертификата не был скомпрометирован, пользователь не подтверждал использование неподписанного сертификата, и на компьютере пользователя не были внедрены сертификаты центра сертификации злоумышленника.

На данный момент HTTPS поддерживается всеми популярными веб-браузерами.