ОСОБЕННОСТИ ФУНКЦИОНИРОВАНИЯ ПРОТОКОЛОВ НА УРОВНЕ ВЗАИМОДЕЙСТВИЯ КЛИЕНТА И СЕРВЕРА

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Данильчук В.С.

Астровский И.И. – к.т.н., доцент

Веб-приложения - это особый тип программ, построенных по архитектуре "клиент-сервер". Особенность веб-приложения заключается в том, что само приложение находится и выполняется на сервере, в то время как клиентская часть реализует пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него. В работе рассматриваются особенности реализации протоколов функционирующих на уровне взаимодействия клиента и сервера.

Взаимодействие клиента и сервера основано на сетевом протоколе HTTP – протоколе прикладного уровня передачи данных. Протокол HTTP работает с 80 портом протокола TCP. Хотя средства, обеспечивающие поддержку HTTP, можно настроить на работу с любым другим портом, практически все браузеры по умолчанию пытаются сначала установить соединение через порт TCP с номером 80. Именно поэтому подавляющее большинство веб – серверов практически всегда вынуждены опрашивать порт 80. Однако одним из самых очевидных исключений является применение туннелирования протокола HTTP через протокол SSL (Secure Sockets Layer - уровень защищённых сокетов)[1].

Протокол SSL позволяет применить на транспортном уровне шифрование, благодаря которому злоумышленник, вклинившийся в сеанс связи клиента и сервера, уже не сможет увидеть передаваемые команды протокола HTTP в открытом тексте. Однако SSL лишь предоставляет протоколу HTTP “защитную оболочку” - не больше и не меньше. Он не расширяет и не вносит каких-либо существенных изменений в базовый механизм запроса и ответа HTTP. Конечно, в некотором смысле использование протокола SSL повышает безопасность, если и сервер, и клиент задействуют дополнительную возможность протокола, заключающуюся в применении сертификатов клиентов. В настоящее время на основании протокола SSL используется его модифицированная версия, названная TLS (Transport Layer Security - безопасность транспортного уровня). Протоколы SSL/TLS, как правило, используют порт TCP с номером 443 .

Основным объектом манипуляции в HTTP является ресурс, на который указывает URI (Uniform Resource Identifier) — унифицированный идентификатор ресурса в запросе клиента. Обычно такими ресурсами являются хранящиеся на сервере файлы, но ими могут быть и логические объекты. Особенностью протокола HTTP является возможность указать в запросе и ответе способ представления одного и того же ресурса по различным параметрам: формату, кодировке, языку и т.д. Именно благодаря возможности указания способа кодирования сообщения, клиент и сервер могут обмениваться двоичными данными, хотя данный протокол является текстовым.

В отличие от многих других протоколов, HTTP не сохраняет своего состояния. Это означает отсутствие сохранения промежуточного состояния между парами «запрос-ответ». Компоненты, использующие HTTP, могут самостоятельно осуществлять сохранение информации о состоянии, связанной с последними запросами и ответами. Браузер, посылающий запросы, может отслеживать задержки ответов. Сервер может хранить IP-адреса и заголовки запросов последних клиентов. Однако сам протокол не осведомлён о предыдущих запросах и ответах, в нём не предусмотрена внутренняя поддержка состояния, к нему не предъявляются такие требования.

Ключевой частью протокола HTTP являются HTTP cookie . HTTP cookie - это небольшой фрагмент данных, отправляемый сервером на браузер пользователя, который тот может сохранить и отсылать обратно с новым запросом к данному серверу. Это позволяет узнать с одного ли сервера пришли оба запроса, например для аутентификации пользователя [2].

Существует несколько типов протоколов аутентификации, которые можно встраивать в протокол HTTP:

- Базовая: имя пользователя и пароль перекодируются по алгоритму Base64.
- Дайджест: подобна базовой, но вместо паролей передаются дайджесты, что исключает возможность обратной расшифровки пароля.
- NTLM: протокол аутентификации, который реализуется в заголовках запросов и откликов протокола HTTP.

Все эти протоколы аутентификации функционируют поверх протокола HTTP (или SSL/TLS), а данные встраиваются непосредственно в поток данных, передаваемых в ходе обмена запросами и откликами [3].

Список использованных источников:
1. HTTP [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/HTTP
2. Скамбрэй, Д. Hacking exposed: Web Applications // Д. Скамбрэй – 2011.
3. Хол, П. Web Security Testing Cookbook // П. Хол, Б. Вальтер – 2008.