```
Trek;ReportStart:=AutoStart;ReportText:= 'Суточный_автоматический_отчет';
      1:
ReportBegin:=HandBegin;ReportEnd:=HandEnd;ReportStep:=HandStep;ReportTrek:=Hand
Trek;ReportStart:=HandStart;ReportText:= 'Суточный_ручной_отчет';
   end_case;
   END_PROGRAM
```

Результаты работы: разработанная программа позволяет дистанционно отслеживать параметры объекта по каналам температуры, создавать несколько видов отчётов, архивировать данные с глубиной архива в 45 дней. Она предназначена для использования в технологическом процессе производства железо-бетонных конструкций.

Список использованных источников

1. SCADA назначение систем, [Электронный ресурс]. – Режим доступа: http://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:SCADA_%D0%BD%D0%B0%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC. – Дата доступа: 21.04.2020.
2. Система автоматического контроля и сбора информации (SCADA) [Электронный ресурс]. – Режим доступа: http://bourabai.kz/dbt/scada.htm. – Дата доступа: 21.04.2020.

УДК 004.057.4

ИССЛЕДОВАНИЕ ПРОТОКОЛА HTTP

Черненко Д.В., ст. преп., Гниденко А.К., ст. преп., Панкевич Д.С., студ.

Витебский государственный технологический университет,
г. Витебск, Республика Беларусь

Реферат. *В статье рассмотрен протокол HTTP, его исследование на примере сайтов vstu.by и sdo.vstu.by, а также четкая инструкция, по которой любой пользователь, повторив эти действия, сможет узнать, как отправляют и получает запросы/ответы любые сайты.*

Ключевые слова: сетевой протокол, HTTP, HTTPS, клиент-сервер, сеть.

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. Клиенты и серверы взаимодействуют, обмениваясь одиночными сообщениями, а не потоком данных. Сообщения, отправленные клиентом, обычно веб-браузером, называются запросами, а сообщения, отправленные сервером, называются ответами. HTTP-сообщения – это обмен данными между сервером и клиентом. Есть два типа сообщений: запросы, отправляемые клиентом, чтобы инициировать реакцию со стороны сервера, и ответы от сервера.

Сообщения HTTP состоят из текстовой информации в кодировке ASCII, записанной в несколько строк. В HTTP/1.1 и более ранних версиях они пересылались в качестве обычного текста. В HTTP/2 текстовое сообщение разделяется на фреймы, что позволяет выполнить оптимизацию и повысить производительность.

Веб разработчики не создают текстовые сообщения HTTP самостоятельно – это делает программа, браузер, прокси или веб-сервер. Они обеспечивают создание HTTP-сообщений через конфигурационные файлы (для прокси и серверов), APIs (для браузеров) или другие интерфейсы.

Так как HTTP это клиент-серверный протокол, соединение всегда устанавливается клиентом. Открыть соединение в HTTP – значит установить соединение через соответствующий транспорт, обычно TCP.

В случае с TCP, в качестве порта HTTP-сервера по умолчанию на компьютере используется порт 80, хотя другие также часто используются, например 8000 или 8080. URL загружаемой страницы содержит доменное имя и порт, который можно и не указывать, если он соответствует порту по умолчанию.

Для того чтобы самому понаблюдать за работой протокола, можно воспользоваться инструментами разработчика в браузере (например, Google Chrome). В нашем случае, будем рассматривать и исследовать сайты vstu.by и sdo.vstu.by. Для начала необходимо зайти в Developer tools, для этого нажимаем правой кнопкой мыши в активном окне браузера и выбираем пункт Inspect, также можно воспользоваться сочетанием клавиш Ctrl+Shift+I, или же просто нажав кнопку F12. Далее необходимо перейти в раздел Сеть(Network). Далее в разделе Network открываем раздел Doc, в разделе Doc открываем файл vstu.by и сразу увидим, на первый взгляд, много непонятного текста.

В активном открытом файле, в заголовке General (Основные заголовки), мы видим:

Request URL: https://vstu.by/ – адрес нашего сайта.

Request Method: GET – метод.

Status Code: 200 – статус, в нашем случае 200 (успешно).

Remote Address: 178.172.163.3:443 – адрес.

Referrer Policy: no-referrer-when-downgrade – заголовок ответа.

Проделав все те же действия с sdo.vstu.by (рис. 1) и полностью проанализировав два сайта, можно заметить, что sdo.vstu.by работает на протоколе HTTP/1.1, а vstu.by – на HTTPS (рис. 2). Так же можно заметить, что sdo.vstu.by работает на ASP.NET, а vstu.by на PHP/5.5.38. Необходимо подчеркнуть, что эти сайты работают на разных серверах, в первом случае, с sdo.vstu.by, Microsoft-IIS/8.5, во втором случае, с vstu.by, nginx/1.18.0. И это лишь малая часть отличий, которые можно отметить между этими, казалось бы, похожими сайтами.

HTTPS (HTTP Secure) является зашифрованной версией HTTP-протокола. Обычно он использует SSL или TLS для шифрования соединения между клиентом и сервером. Это безопасное соединение позволяет клиентам безопасно обмениваться конфиденциальными данными с сервером, например, для банковских операций или онлайн-покупок.

Рисунок 1 – Анализ работы сайта sdo.vstu.by

Рисунок 2 – Анализ работы сайта vstu.by

В заключении можно сказать, что HTTP – легкий в использовании, расширяемый протокол. Структура клиент-сервера, вместе со способностью к простому добавлению заголовков, позволяет HTTP продвигаться вместе с расширяющимися возможностями Сети. Хотя HTTP/2 добавляет некоторую сложность, встраивая HTTP сообщения во фреймы для улучшения производительности, базовая структура сообщений осталась с HTTP/1.0. Сессионный поток остается простым, позволяя исследовать и отлаживать с простым монитором HTTP-сообщений. Сообщения HTTP играют ключевую роль в использовании HTTP; они имеют простую структуру и хорошо расширяемы. Механизм фреймов в HTTP/2 добавляет еще один промежуточный уровень между синтаксисом HTTP/1.x и используемым им транспортным протоколом, не проводя фундаментальных изменений: создается надстройка над уже зарекомендовавшими себя методами.