УДК 004.522 д.т.н., проф. **Мясищев А.А.** (ХмНУ)

АВТОРИЗАЦИЯ НА WEB-СЕРВЕРЕ ARDUINO С ПОМОЩЬЮ HTTP BASIC ACCESS AUTHENTICATION

В работе рассматривается возможность построения web-сервера для управления оборудованием и получением данных с различных датчиков через сеть Интернет. Указывается на то, что для удаленного управления необходимо проводить авторизацию. В противном случае оборудование будет не защищено от несанкционированного доступа пользователей сети Интернет. Отмечается, что управляющие web-сервера создаются на базе микроконтроллеров, которые имеют малые ресурсы и не в состоянии работать с протоколами https, ssl, tls. Поэтому эти сервера являются уязвимыми со стороны сетевых атак. В работе рассмотрено создание web-сервера на Arduino, который использует модернизированную HTTP basic authentication. Модернизация состоит в том, что для авторизации используется пароль из списка паролей, который выбирается пользователем на основании ключа, пересылаемого сервером. При каждом новом входе на сервер предыдущий пароль становиться недействительным. Представлен практический пример web-сервера на Arduino Mega, на котором установлены три светодиода, имитирующие включение-выключение 3-х силовых источников питания (например, электро-розеток), датчик температуры DS18B20 и датчик влажности и температуры DHT 11. Сервер тестировался с двумя контроллерами Ethernet: enc28j60 и w5500. Для этого использовались две библиотеки: UIPEthetnet и Ethernet2, которые показали одинаковые результаты работы.

Ключевые слова: Arduino, Ethernet Shield Arduino, ENC28J60, протокол https, ssl, tls, base64-encoded, basic authentication, web-server, микроконтроллер.

Постановка задачи. С появлением технологий удаленного управления оборудованием через сеть Интернет актуальной стала задача авторизации на управляющих серверах. Часто такими серверами являются web-сервера, построенные на основе микроконтроллеров среди которых наиболее распространены находящиеся в составе контроллеров Arduino. Контроллеры Arduino наиболее доступные, дешевые, имеют большое количество плат расширения (shield), бесплатную программную поддержку с большим количеством библиотек. Однако в отличие от современных компьютеров Arduino имеют скромные вычислительные ресурсы в связи с чем не в состоянии поддерживать полноценные сетевые протоколы, не говоря уже о программном обеспечении обеспечивающих шифрование данных в пакетах. Поэтому команды по управлению оборудованием посылаются с браузера клиента

на web-сервер Arduino без пароля идентификации. Таким образом удаленно может управлять устройствами любой пользователь, которому известен адрес сервера. Это может нанести вред правильному функционированию систем, которые могут располагаться в офисе, предприятии, учреждении, жилом доме.

Для надежной авторизации на серверах с популярными сетевыми операционными системами (Linux, FreeBSD, OpenBSD, Windows Server, ...) используется протокол https, данные которого «упаковываются» в криптографический протокол SSL или TLS, обеспечивая защиту этих данных. Работа этих протоколов выполняется с помощью специального программного обеспечения, которое не может быть установлено на сервера Arduino вследствие, в частности, малой памяти микроконтроллеров. Также и стек коммуникационных протоколов Ethernet Shield Arduino использует сокращенные, урезанные версии протоколов TCP/IP. Несмотря на указанные проблемы, рассмотрим возможность упрощенной удаленной авторизации на сервере Arduino. Для этого воспользуемся HTTP authentication, описанной в стандартах HTTP 1.0/1.1.

Изложение основного материала работы. Рассмотрим web-сервер на Arduino, который управляет удаленно тремя светодиодами (имитируют работу удаленного оборудования) и получает данные с температурных датчиков и датчика влажности. На рисунке 1 показан сервер на Arduino Mega с контроллером сети на ENC28J60(или W5500).

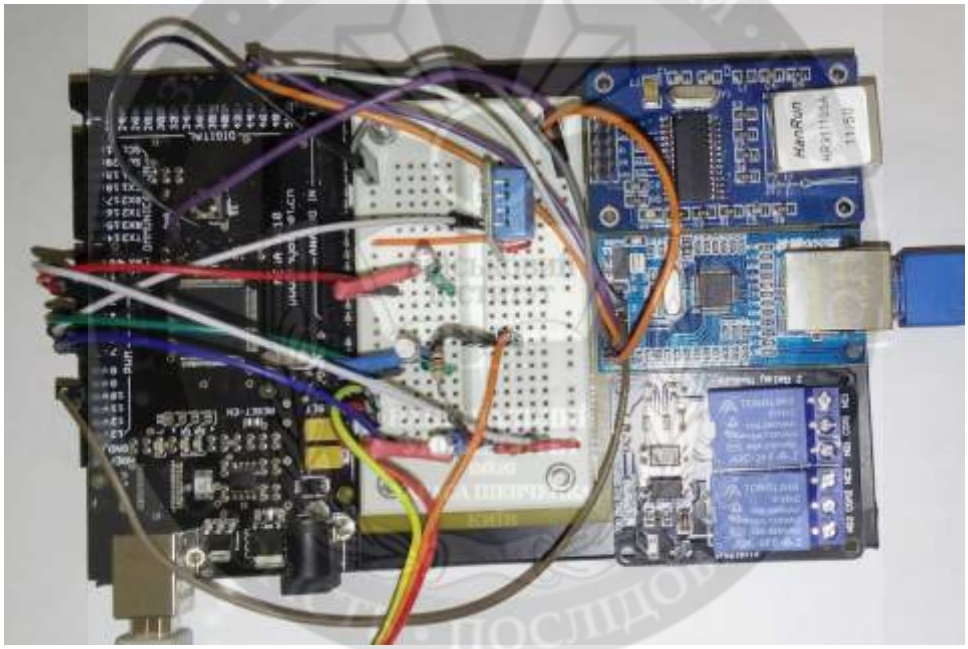

Рис. 1. Макет исследуемого web-сервера на Arduino Mega

В настоящее время протокол HTTP authentication активно применяется в корпоративной среде. Применительно к web-сайтам он работает следующим образом:

1. Сервер, при обращении неавторизованного клиента к защищенному ресурсу, отсылает HTTP статус “401 Unauthorized” и добавляет заголовок “WWW-Authenticate” с указанием схемы и параметров аутентификации.

2. Браузер, при получении такого ответа, автоматически показывает диалог ввода username и password. Пользователь вводит детали своей учетной записи.

3. Сервер аутентифицирует пользователя по данным из этого заголовка.

Весь процесс стандартизирован, поддерживается всеми браузерами и web-серверами. Существует несколько схем аутентификации, отличающихся по уровню безопасности. Ниже перечислены некоторые из них:

1. Basic - здесь username и password пользователя передаются в заголовке Authorization в незашифрованном виде (base64-encoded). Однако при использовании HTTPS (HTTP over SSL)

протокола, basic является относительно безопасной (рис. 2).

Рис. 2. Схема Basic аутентификации

2. Digest – challenge-response-схема, при которой сервер посылает уникальное значение nonce, а браузер передает MD5 хэш пароля пользователя, вычисленный с использованием указанного nonce. Более безопасная альтернатив Basic схемы при незащищенных соединениях, но подвержена man-in-the-middle attacks (с заменой схемы на basic). Кроме того, использование этой схемы не позволяет применить современные хэш-функции для хранения паролей пользователей на сервере.

3. NTLM (известная как Windows authentication) – также основана на challenge-response подходе, при котором пароль не передается в чистом виде. Эта схема не является стандартом HTTP, но поддерживается большинством браузеров и web-серверов. Преимущественно используется для аутентификации пользователей Windows Active Directory в web-приложениях. Уязвима к pass-the-hash-атакам.

Рассмотрим схему аутентификации basic, и впоследствии модифицируем ее для web-сервера Arduino. Согласно рис. 2 практически механизм авторизации выглядит следующим образом. Сначала при запросе на подключение к серверу Arduino браузер посылает заголовок без запроса авторизации:

GET / HTTP/1.1
Host: 192.168.1.18:81
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html, application/xhtml+xml, application/xml;q=0.9,image/webp,/*;q=0.8*
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/44.4.2403.3 Amigo/44.4.2403.3 MRCHROME SOC Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.6,en;q=0.4

Сервер Ардуино отвечает браузеру заголовком на необходимость запроса с авторизацией:

HTTP/1.0 401 Unauthorized
WWW-Authenticate: Basic realm="Arduino - HOME"

Браузер посылает серверу запрос, но уже с авторизацией:

GET / HTTP/1.1
Host: 192.168.1.18:81
Connection: keep-alive
Cache-Control: max-age=0
Authorization: Basic YWxleDoxMzY=
Accept: text/html, application/xhtml+xml, application/xml;q=0.9,image/webp,/*;q=0.8*

Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/44.4.2403.3 Amigo/44.4.2403.3 MRCHROME SOC Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.6,en;q=0.4

Сервер Arduino посылает ответ и после пустой строки html документ:

HTTP/1.0 200 OK
Content-Type: text/html
<Здесь идет передача браузеру html документа>

YWxleDoxMzY= -это закодированный в Base64 набор символов alex:136 (alex - это login, 136 - password, которые вводятся в выскакивающем в браузере окне авторизации). На рис. 3 представлены интерфейсы ввода логина и пароля и ответа сервера Arduino. Программа сервера Ардуино с описанной авторизацией представлена в источнике [1]. В этой программе работа сервера по представленной схеме реализуется с помощью операторов:

```
if (readString.lastIndexOf("YWxleDoxMzY=")>-1) {
  if (readString.lastIndexOf("GET /favicon.ico")>-1) {
    client.println("HTTP/1.0 404 Not Found");
  } else html_doc(client); }
else { client.println("HTTP/1.0 401 Unauthorized");
  client.println("WWW-Authenticate: Basic realm=\"Arduino - HOME\"");}
```

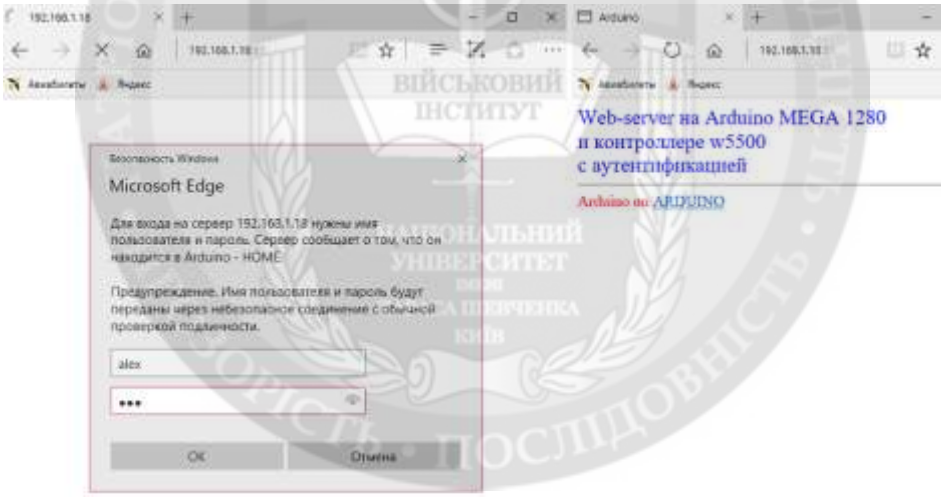

Рис. 3. Интерфейса ввода логина и пароля и ответа сервера Ардуино

Однако безопасность здесь при этой схеме весьма низкая. Пакеты по Интернет передаются в открытом виде, поэтому закодированный логин и пароль (YWxleDoxMzY=) можно легко выловить с помощью программ снифферов и раскодировать с помощью Base64 decode. Модернизируем программу, чтобы в большей степени защитить сервер Arduino для предотвращения несанкционированного управления им. Для этого создадим список паролей, которые известны пользователю сервера. Причем при каждом новом входе на сервер пароль доступа должен меняться на следующий по списку. Поэтому, если предыдущий пароль будет расшифрован, то он становиться недействительный и злоумышленник не сможет зайти по нему. В источнике [1] представлена программа web-сервера Ардуино, на котором установлены три светодиода, имитирующие включение-выключение 3-х силовых источников питания (например, электро-розеток), датчик температуры DS18B20(температура улицы) и датчик влажности и температуры DHT 11 (делает измерения в помещении), которая использует 3

пары логин: пароль:

1. alex:136; 2. alex:138; 3. alex:140

При каждом новом обращении к серверу пароль циклически меняется на следующий в списке. Для того, чтобы знать следующий номер пароля в браузере выводится параметр index. Его пользователь должен запомнить или определить с помощью открытого входа по адресу http://192.168.1.18:81/index. Таким образом, чем больше паролей в списке, тем более защищенным должен выглядеть сервер. На рис. 4 представлен интерфейс управления сервером на Arduino Mega.

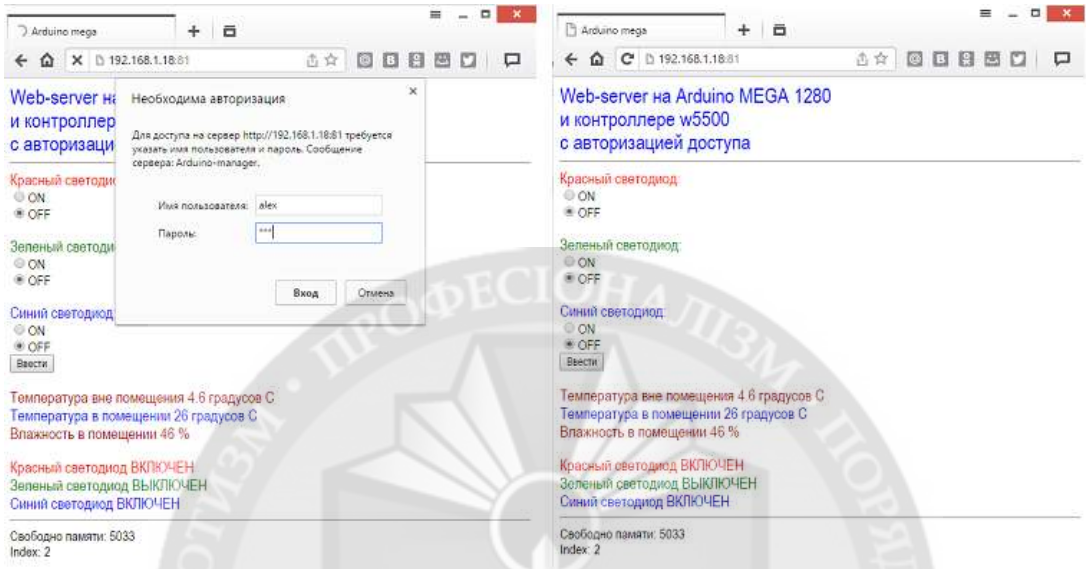

Рис. 4. Интерфейс управления web-сервером на Arduino Mega

В представленном интерфейсе Index:2 обозначает то, что при следующем входе на сервер необходимо для авторизации использовать пару alex:138

В программе работа сервера по представленной схеме для нескольких изменяющихся паролей реализуется с помощью операторов:

```
// 1-й пароль
      else if (readString.lastIndexOf("YWxleDoxMzY=")>-1 && passwd==1) {
        passwd=2; if (readString.lastIndexOf("GET /favicon.ico")>-1) {
          client.println("HTTP/1.0 404 Not Found"); }
        else { onoff(buffer); html_doc(client);} }
// 2-й пароль
       else if (readString.lastIndexOf("YWxleDoxMzg=")>-1 && passwd==2) {
        passwd=3; if (readString.lastIndexOf("GET /favicon.ico")>-1) {
          client.println("HTTP/1.0 404 Not Found"); }
        else {onoff(buffer); html_doc(client);} }
 // 3-й пароль   ...
```

Передача данных выполняется с помощью POST запроса. Данные запроса размещаются после пустой строки заголовка, который посылает браузер web-серверу Arduino:

```
POST / HTTP/1.1
Accept: text/html, application/xhtml+xml, image/jxr, */*
Referer: http://192.168.1.18:81/
Accept-Language: ru,en-US;q=0.8,en;q=0.5,uk;q=0.3
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/51.0.2704.79 Safari/537.36 Edge/14.14393
```

Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Host: 192.168.1.18:81
Content-Length: 23
Connection: Keep-Alive
Cache-Control: no-cache
Authorization: Basic YWxleDoxMzY=
< Пустая строка >
r=0&g=1&b=0&av=2018year

Последняя строка - это передача данных методом POST. Эти данные должны быть обработаны сервером для управления светодиодами (используется форма с радио кнопками).

Включение и выключение светодиодов определяется параметрами r, g и b. Если r=1 - красный светодиод включен, если r=0 - выключен. Аналогично для зеленого(g) и синего(b). Это выполняется операторами:

```
if(buffer.indexOf("r=1") >= 0) { digitalWrite(3, HIGH); }
if(buffer.indexOf("r=0") >= 0) { digitalWrite(3, LOW); }
if(buffer.indexOf("g=1") >= 0) { digitalWrite(5, HIGH); }
if(buffer.indexOf("g=0") >= 0) { digitalWrite(5, LOW); }
if(buffer.indexOf("b=1") >= 0) { digitalWrite(7, HIGH); }
if(buffer.indexOf("b=0") >= 0) { digitalWrite(7, LOW); }
```

После авторизированного входа на страничку управления работу с сервером удобно организовать так, чтобы после каждого изменения параметров светодиодов вход был не авторизированным. Для этого форма должна посылать серверу какой-то дополнительный код (набор символов), получив который сервер бы не запрашивал авторизацию. В этой программе посылается код 2018year с помощью оператора:

```
client.println("<input type='hidden' name='av' value='2018year'>");
```

Так как поле имеет тип hidden, браузер его не отобразит, но методом PUT будет передано значение *2018year* с параметром *av (r=0&g=1&b=0&av=2018year)*. С помощью операторов

```
else if (buffer.lastIndexOf("2018year")>-1 ) {
  if (readString.lastIndexOf("GET /favicon.ico")>-1) {
    client.println("HTTP/1.0 404 Not Found");  }
   else { onoff(); html_doc(client); } }
```

будет выполняться беспарольный вход на страничку управления. Однако для обеспечения лучшей защиты сервера Arduino из программы указанные выше операторы целесообразно убрать.

Выводы. 1. HTTP авторизация Basic передает username и password пользователя в незашифрованном виде (base64-encoded). И если не используется протокол HTTPS (HTTP over SSL), является относительно небезопасной.

2. Представленная в работе модернизация схемы авторизации Basic позволяет относительно безопасно управлять web-сервером Arduino, но требует использования нескольких паролей, каждый из которых сообщается сервером пользователю через специальный параметр index.

3. В среде разработки Arduino IDE 1.8.6 разработано программное обеспечение, реализующее web-сервер на Arduino Mega для управления удаленными объектами через Интернет. Работоспособность сервера опробована на макете, который представлен на рис. 1.

ЛИТЕРАТУРА:

1. Мясищев А.А. Web-server на Arduino MEGA и контроллере сети w5500 с авторизацией доступа для удаленного управления. [Electronic resource]. - Mode of access: https://sites.google.com/site/webstm32/web-server-avtorizaciej, 2018.

2. Мясищев А.А. GET и POST аутентификация на web-сервере Arduino. [Electronic resource]. - Mode of access: https://sites.google.com/site/webstm32/get-post-arduino, 2018.

3. Мясищев А.А. Web - server на Arduino mega с простой аутентификацией, использующей POST запрос. [Electronic resource]. - Mode of access: https://sites.google.com/site/webstm32/web-server--post-auten, 2018.

4. Kitsum. Авторизация на Web сервере микроконтроллера. [Electronic resource]. - Mode of access: https://it4it.club/topic/13-авторизация-на-web-сервере-микроконтроллера/, 2015

5. Гранкин С.С. Подключение Arduino к Интернету: настройка режима клиент-сервер, обработка GET и POST запросов. [Electronic resource]. – Mode of access: http://cxem.net/arduino/arduino176.php , 2016.

6. Выростков Д. Обзор способов и протоколов аутентификации в веб-приложениях. / DataArt Технологический консалтинг и разработка ПО. – [Electronic resource]. – Mode of access: https://habrahabr.ru/company/dataart/blog/262817, 2015.

7. Иго.Т. Arduino, датчики и сети для связи устройств: Пер. с англ. – 2-е изд. – СПб.: БХВ-Петербург, 2015. – 544 с. ил.

8. Иванов П.И. Управление блоком реле через браузер с помощью Arduino и Ethernet shield. [Electronic resource]. - Mode of access: http://www.psub.net/Publication/Details/60, 2015.

9. Скляров Д.В. Искусство защиты и взлома информации. – СПб.: БХВ-Петербург, 2004. – 288 с. ил.

10. Анисимов В.В. Криптографические методы защиты информации. [Electronic resource]. - Mode of access: https://sites.google.com/site/anisimovkhv/learning/kripto/lecture, 2012.

11. Малков А. Классификация механизмов аутентификации пользователей и их обзор. [Electronic resource]. - Mode of access: https://habrahabr.ru/post/177551/, 2013.

12. Гавриков В. Шифрование данных: криптозащита STM32. [Electronic resource]. – Mode of access: https://www.compel.ru/lib/ne/2016/10/2-shifrovanie-dannyih-kriptozashhita-stm32, 2016.

13. Шумель В.В., Рудикова Л.В. Общие принципы организации http-сервера на микроконтроллере. [Electronic resource]. – Mode of access: http://www.elib.bsu.by/bitstream/123456789/52523/1/41-45.pdf, 2013.

REFERENCES:

1. Myasishchev A.A. Web-server na Arduino MEGA i kontrollere seti w5500 s avtorizaciej dostupa dlya udalennogo upravleniya. [Electronic resource]. – Mode of access: https://sites.google.com/site/webstm32/web-server-avtorizaciej, 2018.

2. Myasishchev A.A. GET i POST autentifikaciya na web-servere Arduino. [Electronic resource]. - Mode of access: https://sites.google.com/site/webstm32/get-post-arduino, 2018.

3. Myasishchev A.A. Web - server na Arduino mega s prostoj autentifikaciej, ispol'zuyushchej POST zapros. [Electronic resource]. – Mode of access: https://sites.google.com/site/webstm32/web-server--post-auten, 2018.

4. Kitsum. Avtorizaciya na Web servere mikrokontrollera. [Electronic resource]. - Mode of access: https://it4it.club/topic/13-avtorizaciya-na-web-servere-mikrokontrollera/, 2015

5. Grankin S.S. Podklyuchenie Arduino k Internetu: nastrojka rezhima klient-server, obrabotka GET i POST zaprosov. [Electronic resource]. – Mode of access: http://cxem.net/arduino/arduino176.php , 2016.

6. Vyrostkov D. Obzor sposobov i protokolov autentifikacii v veb-prilozheniyah. / DataArt Tekhnologicheskij konsalting i razrabotka PO. – [Electronic resource]. – Mode of access: https://habrahabr.ru/company/dataart/blog/262817, 2015.

7. Igo.T. Arduino, datchiki i seti dlya svyazi ustrojstv: Per. s angl. – 2-e izd. – SPb.: BHV-Peterburg, 2015. - 544 s. il.

8. Ivanov P.I. Upravlenie blokom rele cherez brauzer s pomoshch'yu Arduino i Ethernet shield. [Electronic resource]. - Mode of access: http://www.psub.net/Publication/Details/60, 2015.

9. Sklyarov D.V. Iskusstvo zashchity i vzloma informacii. – SPb.: BHV-Peterburg, 2004. – 288 s. il.

10. Anisimov V.V. Kriptograficheskie metody zashchity informacii. [Electronic resource]. - Mode of access: https://sites.google.com/site/anisimovkhv/learning/kripto/lecture, 2012.

11. Malkov A. Klassifikaciya mekhanizmov autentifikacii pol'zovatelej i ih obzor. [Electronic resource]. - Mode of access: https://habrahabr.ru/post/177551/, 2013.

12. Gavrikov V. SHifrovanie dannyh: kriptozashchita STM32. [Electronic resource]. - Mode of access: https://www.compel.ru/lib/ne/2016/10/2-shifrovanie-dannyih-kriptozashhita-stm32, 2016.

13. SHumel' V.V., Rudikova L.V. OBSHCHIE PRINCIPY ORGANIZACII HTTP-SERVERA NA MIKROKONTROLLERE. [Electronic resource]. - Mode of access: http://www.elib.bsu.by/bitstream/123456789/52523/1/41-45.pdf, 2013.

д.т.н., проф. Мясіщев О.А.

АВТОРИЗАЦИЯ НА WEB-СЕРВЕРЕ ARDUINO С ПОМОЩЬЮ HTTP BASIC ACCESS AUTHENTICATION

У роботі розглядається можливість побудови web-сервера для управління обладнанням і отриманням даних з різних датчиків через мережу Інтернет. Вказується на те, що для віддаленого управління необхідно проводити авторизацію. В іншому випадку обладнання буде не захищене від несанкціонованого доступу користувачів мережі Інтернет. Відзначається, що керуючі web-сервера створюються на базі мікроконтролерів, які мають малі ресурси і не в змозі працювати з протоколами https, ssl, tls. Тому ці сервера є вразливими з боку мережевих атак. В роботі розглянуто створення web-сервера на Arduino, який використовує модернізовану HTTP basic authentication. Модернізація полягає в тому, що для авторизації використовується пароль зі списку паролів, який вибирається користувачем на підставі ключа, пересилається сервером. При кожному новому вході на сервер попередній пароль ставати недійсним. Представлений практичний приклад web-сервера на Arduino Mega, на якому встановлено три світлодіода, що імітують включення-виключення 3-х силових джерел живлення (наприклад електро-розеток), датчик температури DS18B20 і датчик вологості і температури DHT 11. Сервер тестувався з двома контролерами Ethernet : enc28j60 і w5500. Для цього використовувалися дві бібліотеки: UIPEthetnet і Ethernet2, які показали однакові результати роботи.

Ключові слова: Arduino, Ethernet Shield Arduino, ENC28J60, протокол https, ssl, tls, base64-encoded, basic authentication, web-server, мікроконтролер.

Prof. Myasischev A.A.

LOG ON TO THE ARDUINO WEB SERVER USING HTTP BASIC AUTHENTICATION

The paper considers the possibility of building a web server for controlling equipment and obtaining data from various sensors via the Internet. It is indicated that authorization is necessary for remote management. Otherwise, the equipment will not be protected from unauthorized access of Internet users. It is noted that the managing web servers are created on the basis of microcontrollers, which have small resources and are not able to work with the protocols https, ssl, tls. Therefore, these servers are vulnerable to network attacks. The paper considers the creation of a web server on Arduino, which uses modernized HTTP basic authentication. The upgrade consists in the fact that the password is used for authorization from the password list, which is selected by the user based on the key sent by the server. With each new login to the server, the previous password becomes invalid. A practical example of a web server is presented on the Arduino Mega, which has three LEDs simulating switching on and off of 3 power sources (for example, electric outlets), a temperature sensor DS18B20 and a humidity and temperature sensor DHT 11. The server was tested with two Ethernet controllers: enc28j60 and w5500. For this, two libraries were used: UIPEthetnet and Ethernet2, which showed the same results.

Keywords: Arduino, Ethernet Shield Arduino, ENC28J60, https protocol, ssl, tls, base64-encoded, basic authentication, web-server, microcontroller.