

**Мясищев А.А.**

*Хмельницкий национальный университет, Украина*

## **Авторизация на web-сервере Arduino с помощью GET и POST запросов**

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 (шилдов), 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 технологией GET и POST запросов.

Рассмотрим web-сервер на Arduino, который управляет удаленно тремя светодиодами и получает данные с температурного датчика. На рисунке 1 показан сервер на Arduino mega с контроллером сети на ENC28J60.

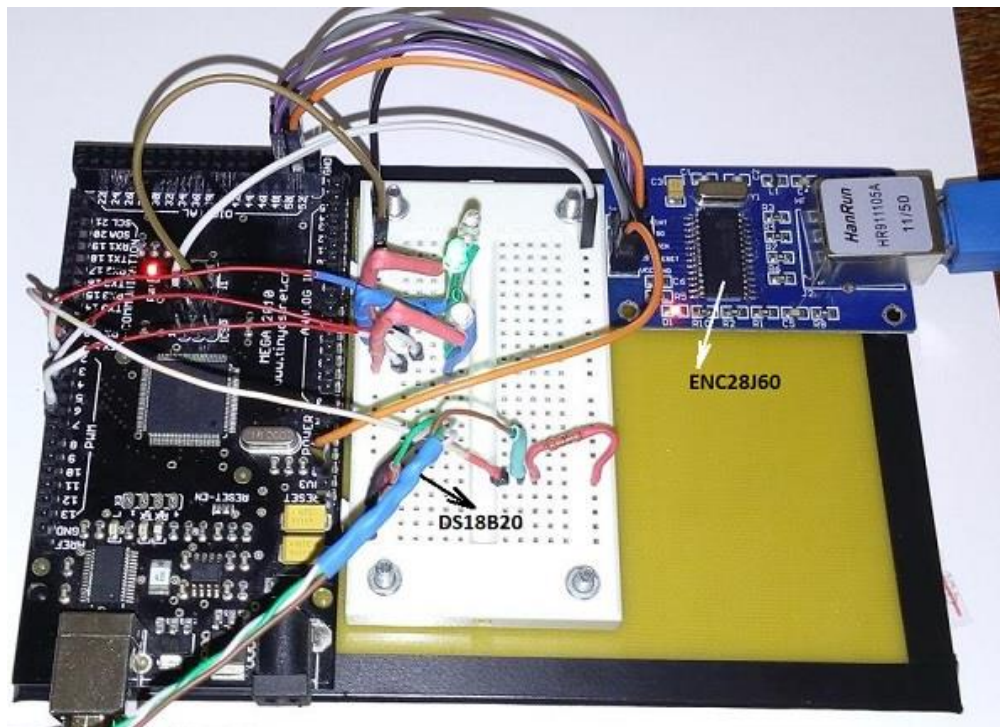


Рис.1. Макет исследуемого web-сервера на Arduino mega

Сервер на браузере клиента формирует форму, которая не должна отображать вводимые символы. Включение и выключение светодиодов выполняется вводом следующих кодов:

*Коды управления светодиодами:*

*Включить красный - 11361*

*Выключить красный - 11360*

*Включить зеленый - 11381*

*Выключить зеленый - 11380*

*Включить синий - 11401*

*Выключить синий - 11400*

Задача заключается в том, чтобы не удалось подсмотреть другому пользователю вводимые в форму коды управления. Шифрование данных по сети не рассматривается.

Рассмотрим GET запрос. Заголовок, посылаемый от браузера клиента серверу имеет вид:

*GET / HTTP/1.1*

*Accept: text/html, application/xhtml+xml, image/jxr, \*/\**

*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; WOW64; Trident/7.0; MALNJS; rv:11.0) like Gecko*

*Accept-Encoding: gzip, deflate*

*Host: 192.168.1.18:81*

*Connection: Keep-Alive*

В конце каждой строки заголовка следуют символы `\r\n`. В случае передачи параметров для включения красного и синего светодиодов заголовок будет следующим [1]:

*GET /?ron=11361&roff=&gon=&goff=&bon=11401&boff= 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; WOW64; Trident/7.0; MALNJS; rv:11.0) like Gecko*

*Accept-Encoding: gzip, deflate*

*Host: 192.168.1.18:81*

*Connection: Keep-Alive*

Таким образом при GET запросе передача параметров на управление передается в первой строке заголовка браузера. На рисунке 2 представлен интерфейс управления. Здесь видно, что в адресной строке в открытом виде передаются все кодовые параметры на управление устройствами. Поэтому коды

на управление можно легко подсмотреть с адресной строки. Программа сервера представлена в источнике[1].

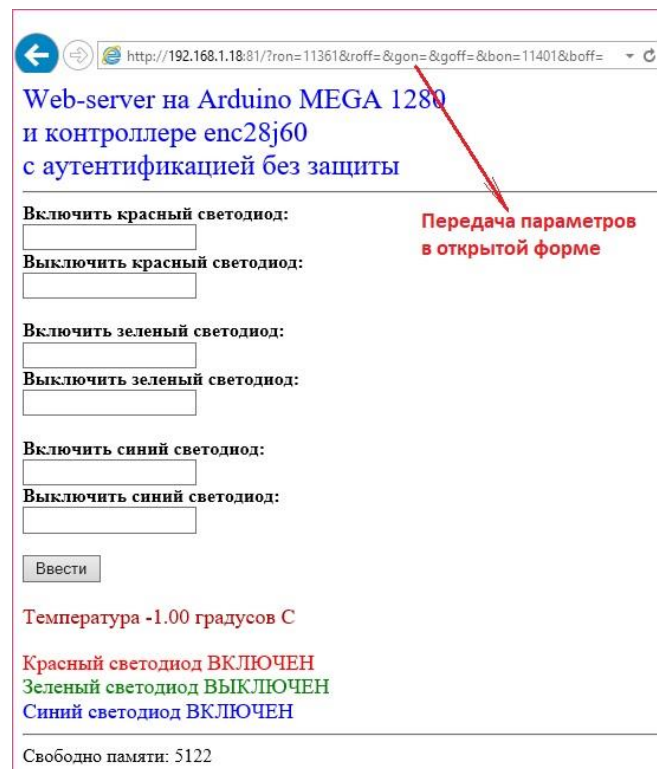


Рис.2. Интерфейс управления для GET запроса

Рассмотрим POST запрос. В этом случае параметры управления(коды) передаются не через адресную строку. Заголовок, который посылает браузер при POST запросе, имеет следующий вид:

*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*

*Content-Type: application/x-www-form-urlencoded*

*User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; MALNJS; rv:11.0) like Gecko*

*Accept-Encoding: gzip, deflate*

*Host: 192.168.1.18:81*

*Content-Length: 42*

*Connection: Keep-Alive*

*Cache-Control: no-cache*

<---здесь пустая строка--->

`ron=11361&roff=&gon=&goff=&bon=11401&boff=`

Параметры для включения красного и синего светодиодов передаются после основного заголовка через пустую строку. Первая строка заголовка указывает на передачу POST запроса.

На рисунке 3 представлен интерфейс управления для POST запроса.

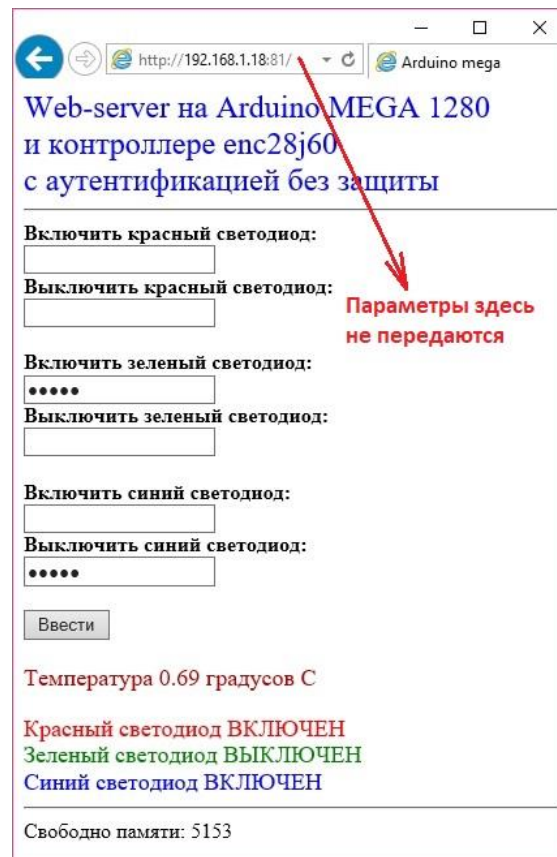


Рис.3. Интерфейс управления для POST запроса

Видно, что в адресной строке не передаются кодовые параметры на управление устройствами, например, при включении красного и синего светодиодов. Программа сервера для POST запроса представлена в источнике [2]. Для считывания кодового параметра используются операторы

```
while(client.available()) //Обработка запроса POST
```

```
{ post = client.read();
```

```
if (buffer.length() <= bufferMax) { buffer += post; } }
```

которые считывают часть заголовка, находящегося после пустой строки основной части заголовка. В массив `buffer` записываются управляющие

параметры (*ron=11361&roff=&gon=&goff=&bon=11401&boff=*). В зависимости от их значения операторами

```
if(buffer.indexOf("ron=11361") >= 0) {digitalWrite(3, HIGH);}
if(buffer.indexOf("roff=11360") >= 0) {digitalWrite(3, LOW);}
if(buffer.indexOf("gon=11381") >= 0) {digitalWrite(5, HIGH);}
if(buffer.indexOf("goff=11380") >= 0) {digitalWrite(5, LOW);}
if(buffer.indexOf("bon=11401") >= 0) {digitalWrite(7, HIGH);}
if(buffer.indexOf("boff=11400") >= 0) {digitalWrite(7, LOW);}
```

выполняются соответствующие действия(включение или выключение).

Из изложенного выше видно, что при POST запросе при вводе в форму кодов управления их значения подсмотреть невозможно(как и при вводе пароля при идентификации в системе через консоль). Однако данные передаются по сети в открытой форме, без кодирования. Поэтому используя программы - sniffer можно всегда перехватить переданные данные и выявить управляющие коды.

### Выводы

1. Авторизация на сервере Arduino с использованием метода запроса GET является нецелесообразной, так как при использовании форм ввода с полями для пароля коды открыто высвечиваются в адресной строке. Поэтому рядом находящийся пользователь их легко может подсмотреть.
2. Авторизация с использованием метода POST скрывает передаваемые коды, однако коды передаются по сети в открытой, в не кодированной форме, поэтому их можно перехватить с помощью программ снифферов.

### Литература

1. 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.

2. М'ясищев 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.
3. М'ясищев А.А. АВТОРИЗАЦІЯ НА WEB-СЕРВЕРЕ ARDUINO ДЛЯ УПРАВЛЕННЯ ЧЕРЕЗ ІНТЕРНЕТ/ ВІСНИК ХНУ. Технічні науки.- Хмельницький: ХНУ, 2018. - №2.-с. 183-191.
4. М'ясищев А.А. АВТОРИЗАЦІЯ НА WEB-СЕРВЕРЕ ARDUINO С ПОМОЩЬЮ HTTP BASIC ACCESS AUTHENTICATION. Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. - К.: ВІКНУ, 2018.-Вип №59. с. 126-133
5. М'ясищев А.А. WEB-SERVER НА ARDUINO ДЛЯ ГРАФІЧЕСКОГО ПРЕДСТАВЛЕННЯ ІНФОРМАЦІЇ С УДАЛЕННИХ ДАТЧИКОВ //ВІСНИК ХНУ. Технічні науки.-Хмельницький: ХНУ, 2018. - №2.-с. 146-153.
6. М'ясищев 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>
7. М'ясищев А.А. Создание надежного http-сервера для удаленного управления по TCP/IP сети на основе ATmega1280 и Wiznet W5100. Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. - К.: ВІКНУ, 2014.-Вип №46. с. 59-70