Защита сайта с помощью .htaccess и .htpasswd

Автор: Голышев С.В. (softtime.ru)

Защита сайта средствами самого сервера Apache является одним из самых простых и в тоже время достаточно надежных способов. В этом случае Вам не нужно досконально продумывать стратегию безопасности, осуществлять ее проектирование и реализацию в коде. К тому же, для того, чтобы создать хорошую систему защиты нужно обладать достаточной квалификацией в этом вопросе. Используя встроенную защиту WEB-сервера Apache, Вы значительно упрощаете себе задачу — все, что Вы должны сделать — это выполнить несложную последовательность действий и Ваш сайт будет в достаточной мере защищен. В данной статье будут подробно описаны шаги и действия, которые Вам необходимо совершить. А в конце статьи будут приведены примеры файлов `.htaccess`.

Базовая аутентификация

В данной статье будет рассмотрен самый простой и доступный способ защиты — базовая аутентификация.

Замечание

Аутентификация — процесс, с помощью которого проверяется, что некто является именно тем, за кого он себя выдает. Как правило, проверка включает в себя ввод имени и пароля.

Рассмотрим, как работает базовая аутентификация.
При обращении посетителя в защищаемую директорию, сервер Apache в ответ на запрос посылает заголовок с кодом 401 (401 authentication required header). Браузер посетителя принимает заголовок с кодом 401 и выводит окно с полями для ввода имени пользователя и пароля. После ввода имени и пароля эти данные отсылаются назад серверу, который проверяет имя пользователя на предмет нахождения в специальном списке, а пароль на правильность. Если все верно, то посетитель получает доступ к ресурсу. Вместе с заголовком браузеру посылается специальной имя, называемое областью действия. Браузер кэширует не только имя и пароль, чтобы передавать их при каждом запросе, но и область действия. Благодаря этому, ввод имени и пароля в защищаемой директории осуществляется только раз. В противном случае их необходимо было бы вводить при каждом запросе к защищаемой директории. Кэширование параметров аутентификации (имя, пароль, область действия), обычно осуществляет только в пределах одного сеанса.

Замечание

При базовой аутентификации имя пользователя и его пароль передаются в сеть в открытом виде в течении всего сеанса, когда посетитель работает с защищенной директорией. Хакер может перехватить эту информацию, используя сетевой анализатор пакетов. Данный вид аутентификации не должен использоваться там, где нужна реальная защита коммерческо-ценной информации.

Замечание

WEB-сервер Apache поддерживает еще один вид защиты — digest-аутентификацию. При digest-аутентификации пароль передается не в открытом виде, а в виде хеш-кода, вычисленному по алгоритму MD5. Поэтому пароль не может быть перехвачен при сканировании трафика. Но, к сожалению, для использования digest-аутентификации необходимо установить на сервер специальный модуль - mod_auth_digest. А это находится только в компетенции администрации сервера. Также, до недавнего времени, digest-аутентификация поддерживалась не всеми видами браузеров.

Защита сайта — это просто

Для того чтобы защитить сайт, нужно выполнить следующую последовательность действий: создать файл с паролями, переписать его на сервер, создать файл `.htaccess` и тоже переписать его на сервер. Для организации защиты понадобится.

1. WEB-сайт и FTP-доступ к нему.

2. Права на создание файлов `.htpaccess` и организацию защиты с помощью них.
3. Утилита генерации паролей `htpasswd.exe`

Проверка работы файла .htaccess на сервере

Для того чтобы проверить есть ли у Вас права на организацию защиты с помощью файлов `.htaccess` создайте текстовый файл с именем `.htaccess` (первым символом идет точка, расширение отсутствует).

Замечание

Удобно создавать файлы .htaccess с помощью встроенного редактора в оболочках Far, WindowsCommander, TotalCommander и т.п., а также в редакторе Блокнот.

Замечание

Чтобы блокнот не подставлял автоматически расширение txt, в диалоге сохранения в выпадающем списке "тип файла" следует выбрать опцию "Все файлы".

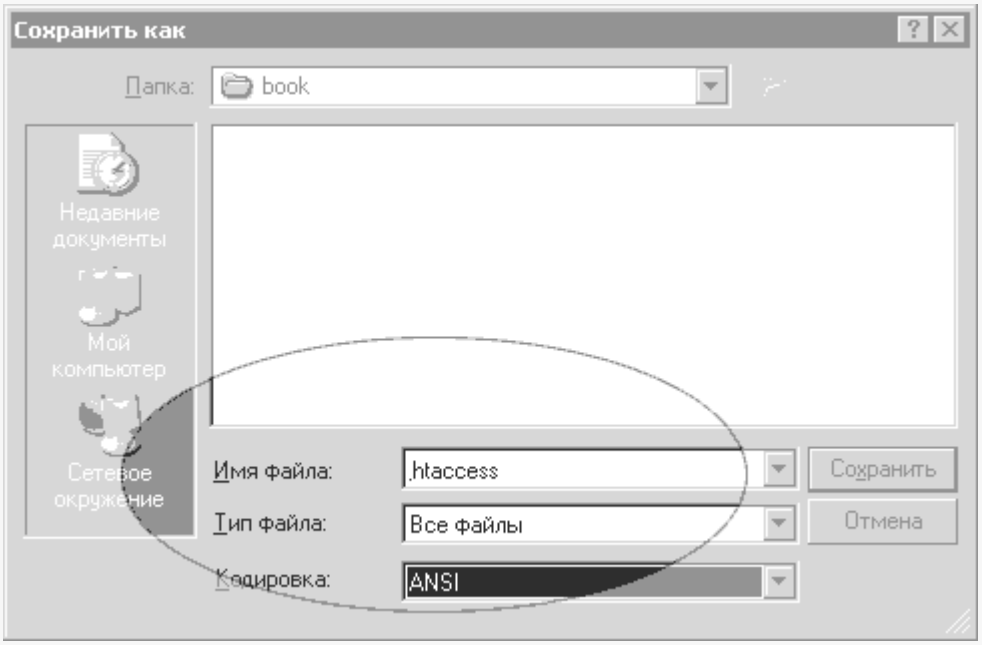


***Рис. 1**. Сохранение файлов `.htaccess` в блокноте*

Перед тем как сохранить файл, впишите в него следующие строки:

Проверка работы .htaccess

```
AuthType Basic
AuthName admin
require valid-user
```

Затем, через FTP-доступ, перепишите файл .htaccess на сайт, в ту директорию, которую вы хотите защитить.

Замечание

Действие файлов .htaccess распространяется не только на ту директорию, где лежит файл, но и на все поддиректрии, лежащие уровнем ниже.

Далее через браузер обратитесь к этой директории. Если Вы защищаете директорию admin и переписали туда файл .htaccess, то для проверки Вам следует вписать в адресную строку браузера следующий URL: http://www.mysite.ru/admin/.

Если после этого Вам открылся запрос на ввод логина и пароля, как на рисунке ниже, то тестирование прошло успешно и можно продолжать защиту директории.

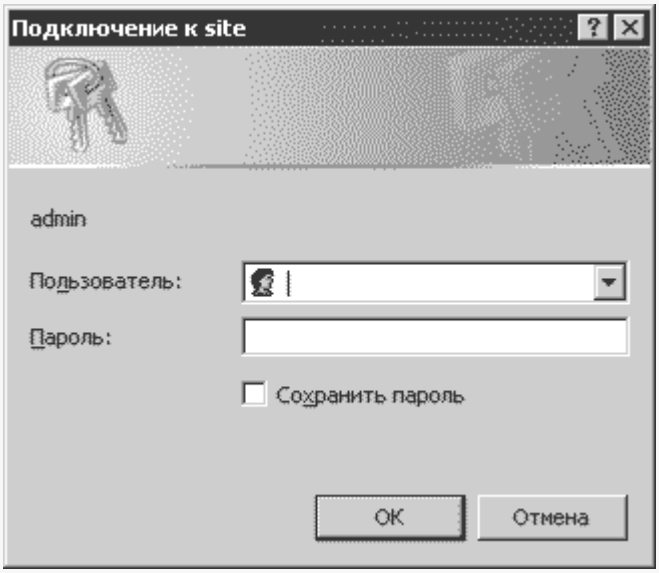


***Рис. 2** . Окно ввода логина и пароля*

Если вы все сделали правильно, но окошко ввода пароля не появилось, то это значит, что настройки сервера запрещают Вам использовать файлы `.htaccess` для защиты директорий. Для решения данного вопроса Вам следует связаться с администрацией сервера, либо использовать другой тип защиты. После того, как было выяснено, что файлы `.htaccess` работают, следует удалить с сайта только что написанный тестовый файл.

Замечание

Если по каким либо причинам Вы не можете удалить файл .htaccess, то создайте пустой файл .htaccess и замените им файл, лежащий на сервере.

Создание файла с паролями .htpasswd

Файл с паролями создается утилитой `htpasswd.exe`. Если у Вас на машине установлен WEB-сервер Apache, то данная утилита находится в директории с установленным **Apache**-ем в подкаталоге **bin**.

Замечание

Если у Вас не установлен Apache, то утилиту htpasswd.exe можете скачать здесь.

Для работы с утилитой htpasswd.exe необходим интерфейс работы с командной строкой. Интерфейсом работы с командной строкой обладают такие программы как Far, WindowsCommander и т.п. Здесь будет рассмотрена работа с командной строкой с помощью утилиты cmd, которая входит в поставку Windows 2000/XP и т.п.
Нажмите **"Пуск"->"Выполнить"**, введите в строку ввода **cmd** и нажмите **OK**. Вам откроется окно утилиты CMD.

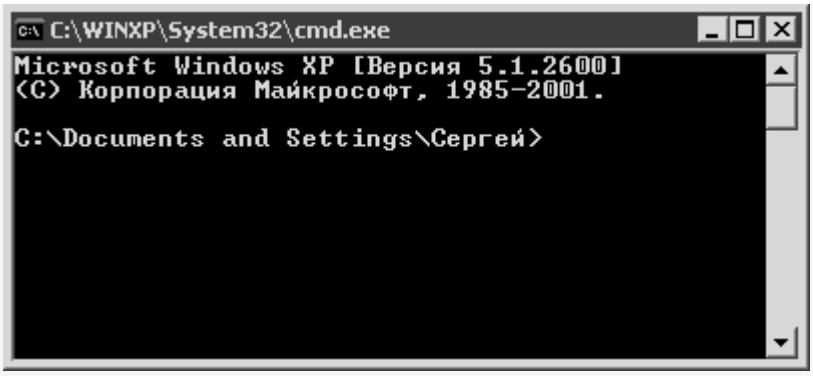
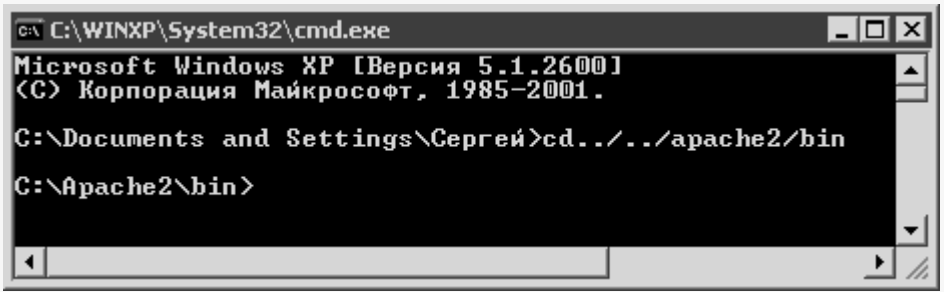


***Рис. 3**. Окно утилиты CMD*

Далее необходимо перейти в директорию, где находится утилита `htpasswd.exe`. Допустим, сервер Apache установлен в директории c:/Apache2, тогда введите в командную строку команду: cd../../apache2/bin и нажмите ввод.



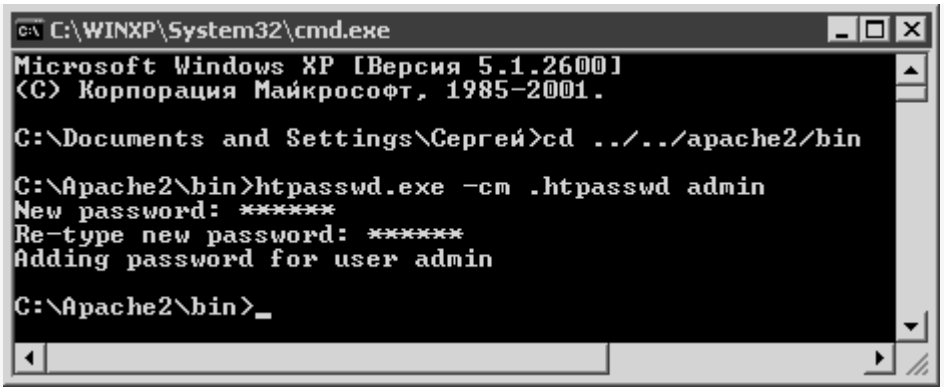
Вы перешли в директорию c:Apache2in. Теперь нужно дать команду на создание файла с паролем. Введите в командную строку следующее:

```
htpasswd -cm .htpasswd admin
```

- -cm — это ключи для утилиты. Ключ c — указывает, что необходимо создать новый файл с паролями. Если файл с таким именем уже существует, то он будет перезаписан. Ключ m — определяет шифрование по алгоритму MD5.
  .htpasswd — имя файла с паролями (можете использовать любое имя).
  admin — имя посетителя, которому будет разрешен доступ в закрытую область сайта.

В ответ, должен появится запрос на ввод пароля и его повтор. Если все правильно, то в завершении появится сообщение: Adding password for user admin. И в директории c:Apache2in появится файл .htpasswd, к котором будет находиться строка с именем пользователя и хеш-кодом его пароля. Для того, что бы в тот же файл .htpasswd добавить еще одного пользователя следует убрать ключ -c из команды запуска утилиты htpasswd.exe

```
htpasswd -m .htpasswd admin
```



Замечание

Если файл с паролями не был создан, то возможно, некоторые ключи утилиты не поддерживаются в Вашей операционной системе. Например, иногда не поддерживается ключ m. В этом случае, Вам нужно ввести htpasswd -c .htpasswd admin
Для того, чтобы посмотреть ключи и параметры работы утилиты введите htpasswd.exe /? Вам будет выдано описание интерфейса.

Итак, файл с паролями создан. Теперь Вам необходимо переписать его на сервер. Файлы с паролями очень желательно класть выше корневой директории сайта — туда, куда не будет доступа посетителям. Если это невозможно, то файлы с паролями следует обязательно защитить. Это можно сделать с помощью файлов .htaccess. Чтобы защитить файлы с паролями создайте файл со строками, представленными в следующем листинге.

Защита файлов .htpasswd

```
<Files .htpasswd>
   deny from all
</Files>
```

И положите его в ту директорию, где находится Ваш файл с паролями. Теперь посетители сайта не смогут получить к нему доступ.
Файл с паролем создан и защищен от несанкционированного доступа. Теперь необходимо создать файл .htaccess, который будет использоваться в защищаемой директории.

Создание файла .htaccess

Для защиты директории могут использоваться следующие директивы:

- AuthType — Тип используемой аутентификации. Для базовой аутентификации эта директива должна иметь значение: Basic
  AuthName — Имя области действия аутентификации. Текст, помогающий посетителю понять, куда он пытается получить доступ. Например, может быть написано: "Private zone. Only for administrator!"
  AuthUserFile — путь к файлу с паролями (.htpasswd).
  AuthGroupFile — путь к файлу групп, если он существует.
  Require — Одно или несколько требований, которые должны быть выполнены для получения доступа к закрытой области.

Пример файла .htaccess

```
AuthType Basic
AuthName "Private zone. Only for administrator!"
```

```
AuthGroupFile /usr/host/mysite/group
AuthUserFile  /usr/host/mysite/.htpasswd
require group admins
```

Следует более подробно описать директивы AuthUserFile и AuthGroupFile. В них прописываются абсолютные пути к соответствующим файлам от корня сервера.

Внимание!

Относительные пути работать не будут!

Путь от корня сервера, можно узнать, спросив у администрации сервера, либо можно попробовать выяснить его самим. Для этого выполните функцию phpinfo(). На экран будет выведена фиолетовая таблица. Значение абсолютного пути от корня сервера можно посмотреть в переменных: doc_root, open_basedir, DOCUMENT_ROOT.
Директива Require определяет кому разрешен доступ к закрытой области. Например,

- require valid-user — разрешен доступ всем прошедшим проверку
- require user admin alex mango — разрешен доступ только посетителям с именами admin, alex, mango. Естественно, они должны пройти аутентификацию.
- require group admins — разрешен доступ всем пользователям из группы admins

Файлы групп

Если к защищаемой области сайта должна иметь доступ большая группа людей, то удобно объединить людей в группы, и разрешать доступ, определяя принадлежность посетителя к группе.
Формат файла групп очень прост. Это текстовый файл, каждая строка, которой описывает отдельную группу. Первым в строке должно идти название группы с двоеточием. А затем через пробел перечисляются посетители, входящие в группу.

Пример файла групп

```
Admins: admin alex mango
Users: guest user max23
```

В группу Admins входят посетители с именами admin, alex, mango. А группу Users входят посетители с именами guest, user, max23.

Примеры файлов .htaccess

Доступ всем пользователям, прошедшим авторизацию

```
AuthType Basic
AuthName "Private zone. Only for administrator!"
AuthUserFile  /usr/host/mysite/.htpasswd
require valid-user
```

Доступ только пользователям admin и root

```
AuthType Basic
AuthName "Private zone. Only for administrator!"
AuthUserFile  /usr/host/mysite/.htpasswd
require user admin root
```

Доступ только пользователей из группы admins

```
AuthType Basic
AuthName "Private zone. Only for administrator!"
AuthUserFile  /usr/host/mysite/.htpasswd
AuthGroupFile /usr/host/mysite/group
require group admins
```

Запрет доступа только к файлу private.zip

```
<Files private.zip>
AuthType Basic
AuthName "Private zone. Only for administrator!"
AuthUserFile  /usr/host/mysite/.htpasswd
require valid-user
</Files>
```