**Защита сайта с помощью .htaccess и .htpasswd**

Автор: Голышев С.В. ([softtime.ru](http://www.softtime.ru/" \t "_blank))

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**

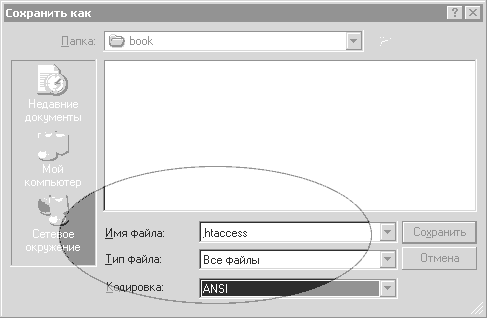
Для того чтобы проверить есть ли у Вас права на организацию защиты с помощью файлов .htaccess создайте текстовый файл с именем .htaccess (первым символом идет точка, расширение отсутствует).

**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

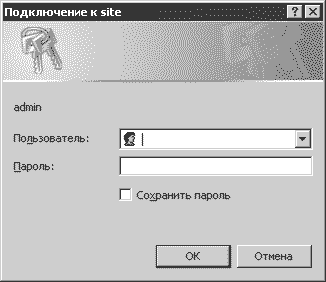
Затем, через FTP-доступ, перепишите файл .htaccess на сайт, в ту директорию, которую вы хотите защитить.

**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.

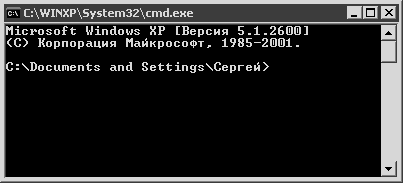
**Создание файла с паролями .htpasswd**

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**

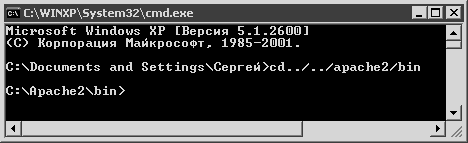
Если у Вас не установлен Apache, то утилиту htpasswd.exe можете скачать [здесь](http://www.php.su/articles/apache/010files/htpasswd.zip).

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 **ОК**. 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 с:/Apache2, тогда введите в командную строку команду: cd../../apache2/bin и нажмите ввод.



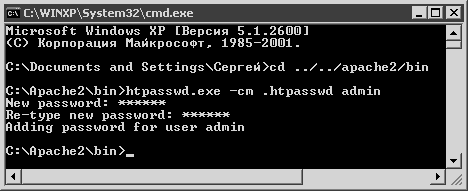
Вы перешли в директорию с:Apache2in. Теперь нужно дать команду на создание файла с паролем. Введите в командную строку следующее:

htpasswd -cm .htpasswd admin

* -cm — это ключи для утилиты. Ключ с — 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>

# htaccess запретить доступ

11.02.2012

[](http://lifeexample.ru/razrabotka-i-optimizacia-saita/htaccess-zapretit-dostup.html)

Здравствуй уважаемый читатель блога[LifeExample](http://lifeexample.ru/), сегодня, я хочу рассмотреть тему ограничения доступа к директориям сайта. Очень часто вебмастер сталкивается с вопросом: как с помощью файла **.htaccess** запретить доступ, к какой либо части сайта, или ко всему сайту.

Первой причиной такого желания может стать наличие админской части сайта и необходимость скрыть ее от чужих глаз и рук. Вторым поводом к запрету доступа может являться процесс внедрения изменений на вашем сайте с необходимостью перенаправления пользователей на временную страницу информации.

В этой статье я покажу как можно:

* С помощью **.htpasswd** и **.htaccess** запретить доступ к папке сайта.
* Закрыть доступ к сайту для всех, кроме одного IP адреса.
* С помощью **.htaccess** и **.htpasswd** настроить политику прав пользователей.
* Создать **.htpasswd** файл паролей, и где его лучше расположить.

В интернете море неисчерпаемых источников информаций на тему работы с файлом **.htaccess** . Тем не менее, чтобы выполнить вышеперечисленные задачи нужно пропустить через себя большой поток информации и понять, что действительно нужно, а без чего можно обойтись. Поэтому я постараюсь кратко и наглядно изложить суть задач и их решения.

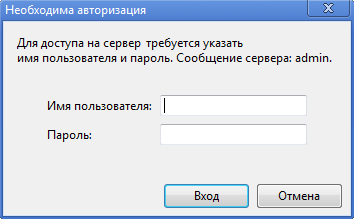
## Как с помощью .htpasswd и .htaccess запретить доступ к папке сайта.

Если ты читатель не знаешь, что это за файлы поясню:

**.htaccess** – файл дополнительной конфигурации веб-сервера.

**.htpasswd** – файл содержащий пары логин и пароль в MD5 хеше, используемый в механизме базовой аутентификации.

С помощью этих двух файлов можно запросто сделать так чтобы при попытке открыть сайт пользователям предлагалось ввести логин и пароль.



Только после того как пользователь правильно введет пару логин и пароль, соответствующую занесенной в файл **.htpasswd**, он будет допущен к просмотру скрытой области сайта.

Для того чтобы это осуществить достаточно создать файл **.htaccess** и поместить его в ту папку сайта к которой вы хотите запретить доступ.

1. Добавьте в созданный файл следующее содержание:

|  |  |
| --- | --- |
| 1 2 3 4 | AuthType Basic  AuthName "Please input login and password!"  AuthUserFile  /usr/host/mysite/.htpasswd  require valid-user |

* **AuthType** — может иметь значение Basic или Digest, в нашем случае нам подойдет Basic.
* **AuthName** — текст выводимый вместе с формой авторизации.
* **AuthUserFile** — путь к файлу с паролями (.htpasswd).
* **AuthGroupFile** — путь к файлу групп, если он существует.
* **Require** — определяет параметры аутентификации. (User,Group,Valid-user) Valid-user – все прошедшие проверку. User – пользователи имеющие права доступа. Group – группы пользователей имеющие права доступа.

2. Теперь создайте файл .htpasswd и в него поместите следующую строку:

|  |  |
| --- | --- |
| 1 | admin:$apr1$ejtOHg/4$cMColJLMZ39weAalo3V8k0 |

Строка эта содержит пару логин и пароль где:

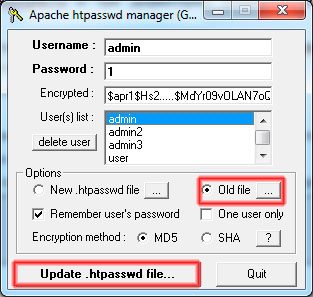
|  |  |
| --- | --- |
| 1 2 | Логин: admin Пароль: 1 |

Чтобы генерировать любые пароли для файла .htpasswd можно воспользоваться**утилитой для генерации паролей htpasswd.exe**

[Скачать утилиту для генерации паролей htpasswd.exe](http://lifeexample.ru/download_count.php?url=upload/htaccess-zapretit-dostup/htpasswd.zip) *( Скачали: 2110 чел. )*

Пользоваться утилитой для генерации паролей очень легко, нужно создать файл .htpasswd и в программе выбрать его с помощью кнопки “**Old file**“. Далее все интуитивно понятно, вводим новых пользователей и пароли к ним, затем жмем на “**Update**“. Кстати данная утилита генерации паролей для htpasswd, может не только создавать новые, но и редактировать старые пароли. Что очень удобно, в случае если вы забыли старый пароль. Еще стоит сказать что эта утилита является стандартной и поставляется вместе с сервером Apache.

На многих сайта и блогах, рассказывается, как пользоваться ей с помощью CMD, но нигде не встречал о том, как создавать пароли с ее графическим интерфейсом, а это может быть очень удобным для начинающих вебмастеров.



Теперь важный для новичков момент! Напротив директивы **AuthUserFile** в.htaccess должен быть прописан абсолютный путь до файла с паролями. Узнать путь можно многими способами, но самый легкий и точный, на мой взгляд, это выполнить в **index.php** строку кода:

|  |  |
| --- | --- |
| 1 | echo $\_SERVER['DOCUMENT\_ROOT']; |

Выполнение этой строчки кода приведет к выводу на страницу абсолютного путь до корня сайта, который нам и нужен.

**Внимание важное замечание!** Очень полезным будет расположить файл с паролями вне папки с содержимым сайта. Это позволит защитить его от злоумышленников.

Если у тебя читатель нету прав на создание файлов вне главного каталога сайта, не переживай, ты все равно сможешь его защитить если пропишешь в .htaccessстроки:

|  |  |
| --- | --- |
| 1 2 3 | <Files .htpasswd>     deny from all  </Files> |

Настало время проверить работоспособность базовой авторизации. Откроем сайт через любой браузер.

Теоретически после проделанных действий любой браузер выведет нам форму базовой аутентификации, если этого не произошло, нужно еще раз проверить указанный в **.htaccess** путь к **.htpasswd** и исправить его.

## Запретим доступ к одному файлу

Кроме запрета на директорию можно установить ограничение доступа к конкретному файлу, для этого нужно обрамить запись из **.htaccess** в тег <Files>

|  |  |
| --- | --- |
| 1 2 3 4 5 6 | <Files "passwd.txt"> AuthType Basic  AuthName "Please input login and password!"  AuthUserFile  /usr/host/mysite/.htpasswd  require valid-user  </Files> |

Хочу обратить внимание на то, что рассмотренный вид аутентификации, не должен являться основным средством защиты данных. Т.к. данные (логин и пароль) пересылаются между клиентом и сервером в открытом виде, и могут быть перехвачены злоумышленниками.

## Как с помощью .htaccess и .htpasswd настроить политику прав пользователей

Кроме простого запрета для всех пользователей с помощью инструмента базовой аутентификации сервера, можно сделать управление ролями. Конечно, высококлассной системы распределения ролей не получится но, тем не менее, что то похожее реализовать можно.

Допустим, у нас есть сайт http://lifeexample.ru и три директории отличающиеся степенью секретности.

1. http://lifeexample.ru/index.php – доступ имеют все пользователи из файла .htpasswd .
2. http://lifeexample.ru/admin/ – доступ имеют админы и cуперпользователи.
3. http://lifeexample.ru/admin/root/ – доступ имеют только cуперпользователи.

Чтобы реализовать разграничение пользователей, нужно объединить их в группы. Для этого в системе базовой аутентификации предусмотрен механизм работы с группами пользователей.

Создадим файл **.htgroup**, в него запишем группы и их пользователей в таком виде:

|  |  |
| --- | --- |
| 1 2 3 | Users: user user2 user3  Admins: admin admin2 admin3 user3 Root: root root2 root3 |

Замечу, что пользователь с логином **user3** состоит сразу в двух группах, и это значит, что он имеет доступ с админской части сайта. Сразу подправим файл**.htpasswd** и с помощью скачанной ранее программки для генерации паролей, занесем него всех пользователей из созданных групп.

Теперь в настройках .htaccess заменим строку:

|  |  |
| --- | --- |
| 1 | require valid-user |

на следующую:

|  |  |
| --- | --- |
| 1 | require group Users Admins Roots |

И расположим файл в главной директории сайта. Проверяем… И видим что ничего не работает. Это вполне объяснимо ведь мы не указали путь до файла с группами. Перед require добавим AuthGroupFile указав в ней путь:

|  |  |
| --- | --- |
| 1 | AuthGroupFile /usr/host/mysite/.htgroup |

Теперь можно радоваться, все заработало.

Т.к. файл .htaccess располагается в главной директории, вышеперечисленными действиями мы отрываем доступ к http://lifeexample.ru/index.php для всех имеющихся пользователей.

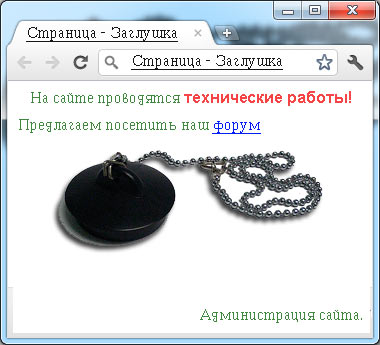
Следующей задачей будет отсеять обычных пользователей и оставить доступ только группам **Admins** и **Roots** к части http://lifeexample.ru/admin/. Сделать это проще чем кажется, нужно только скопировать **.htaccess** из главной директории, в папку “**/admin**” и подправить список допустимых групп, удалив из него группу**Users**.

Аналогичную операцию проделаем для папки “**/admin/root**” предназначенной только суперпользователям. Скопируем файл настроек и оставим там только группу**Roots**.

Еще один важный момент, если в **.htaccess** до всех этих операций содержались другие настройки, то при копировании этого файла в папки сайта, можно оставить только строки отвечающие за права доступа.

## Страница заглушка

В начале статьи я упомянул о том, что задача запрета доступа к сайту возникает при проведении технических работ. В этот период все пользователи должны перенаправляться на страницу “**Заглушку**”, в которой будет расположена информация о приводящихся работах.



Чтобы реализовать данную потребность нужно внести некоторые изменения в тот .htaccess, что мы уже сделали, дописав в самое его начало следующие правила:

|  |  |
| --- | --- |
| 1 2 3 4 5 | <ifModule mod\_rewrite.c> RewriteEngine On RewriteCond %{REMOTE\_ADDR} !=127.0.0.1 RewriteRule ^.\* zaglushka.php [L] </ifModule> |

Добавив эти строки и, **заменив** “127.0.0.1” на свой реальный IP адрес, только мы будем иметь доступ к содержимому сайта, а все остальные IP адреса, будут перенаправляться на страницу **zaglushka.php**. Кстати ее нужно будет создать и поместить в корневую директорию.

Может случиться так что модуль перенаправления **mod\_rewrite.c** будет отключен хостером. Тогда с эту задачу можно будет решить с помощью php:

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 | $SecretKey = '123456789'; $AdminCookie = 'Вход выполнен'; if ($\_COOKIE['AdminCookie'] != $AdminCookie && $\_SERVER['QUERY\_STRING'] != $SecretKey) {     require\_once 'zaglushka.html';     exit; } else {     setcookie('AdminCookie', $AdminCookie, time()+3600\*24\*365, '/');     if ($\_SERVER['QUERY\_STRING'] == $SecretKey) {         header('Location: /');         exit;     } } |

Для того чтобы получить доступ к сайту нужно ввести http://имя сайта/?123456789, если код не будет веден, пользователю будет выдана страница заглушка.

Конечно, можно обойтись и более коротким куском кода:

|  |  |
| --- | --- |
| 1 2 3 | if($\_SERVER['REMOTE\_ADDR'] != "мой.ip.адр.ес"){ header('Location: /zaglushka.html'); } |

Но первый вариант решения мне кажется, наиболее элегантным, поскольку с одного IP может входить сразу несколько пользователей, и один из них не должен иметь доступ к сайту. Для этого в первом способе используется секретный ключ.

С помощью данного скрипта можно реализовать хорошую систему аутентификации сайта с запретом доступа отдельных пользователей.

В отличии от базовой аутентификации с .htaccess и .htpasswd при которой кэширование параметров аутентификации (имя, пароль) осуществляет только в пределах одного сеанса, в этом случает обратившись к сайту единожды, и введя секретный код, известный только администратору, нам не придется его больше вводить в течении установленного времени.

Аналогичный механизм с использованием **cookes**и секретного ключа можно выполнить с помощью **.htaccess**:

|  |  |
| --- | --- |
| 1 2 3 4 5 6 | RewriteCond %{HTTP\_COOKIE} !hash276819937ee81772017638837ab49992heef78=true [NC] RewriteCond %{QUERY\_STRING} (123456789) RewriteRule .+ setcookie.php?%1 [L] RewriteCond %{HTTP\_COOKIE} !hash276819937ee81772017638837ab49992heef78=true [NC] RewriteRule .+ index.html [L] RewriteRule .+ index.php |

## Закрыть сайт для всех ip адресов

Иногда случается так, что нужно закрыть доступ к сайту кардинально без использования аутентификации, для всех, кроме себя конечно. Сделать это можно все тем же .htaccess

|  |  |
| --- | --- |
| 1 2 3 | order deny,allow deny from all allow from 127.0.0.1 *#[СВОЙ IP]* |

127.0.0.1 – заменить на свой.

## Закрыть сайт для списка ip адресов

|  |  |
| --- | --- |
| 1 2 3 4 | Order deny,allow Deny from all Allow from 127.0.0.1 *#[ запрет для одного IP]* Allow from 127.0     *#[ запрет со всех серверов, начинающихся на эти числа]* |

В этой статье мы рассмотрели тему: “**Как с помощью htaccess запретить доступ к части сайта**”, статья получилась объемная и хотелось бы подвести итоги. И так самое главное было понять, как работает механизм базовой аутентификации, и зачем он вобще нужен. Напомню, что данную аутентификацию лучше не использовать там где нужно реально защитить важную информацию. Если ты уважаемый читатель еще не разобрался с тем как настраивать файлы паролей и групп, предлагаю тебе скачать их в готовом виде:

[Скачать все файлы в одном архиве](http://lifeexample.ru/download_count.php?url=upload/htaccess-zapretit-dostup/htpasswd.zip) *( Скачали: 2110 чел. )*