Принудительный вход через HTTPS

Плагины

forcessllogin

См. https://www.dokuwiki.org/plugin:forcessllogin, похоже, не отражает доступ SSL в URL , т.е. страница «Докувики», доступ к которой запрещен, не будет открыта по протоколу https, что затрудняет отладку и обеспечение безопасного входа в систему.

Апачи

Используя mod_rewrite Apache, можно принудительно использовать HTTPS для входа в DokuWiki, тем самым предотвращая передачу паролей в открытом виде.

Вам также может понадобиться, чтобы все запросы (а не только вход) использовали HTTPS. Для этого создайте файл .htaccess в корневом каталоге DokuWiki и вставьте следующий код.

.htaccess

```
RewriteCond %{HTTPS} !on
RewriteRule (.*) https://%{HTTP_HOST}/$1 [R,L]
```

Если вы хотите принудительно задать только какой-то определенный URL , сначала прочтите статью о перезаписи URL.

Перенаправление на защищенное соединение, которое ограничено определенным набором страниц (например, страницы входа), требует их распознавания на основе URL . Некоторые страницы (например, страницы «доступ запрещен», которые могут быть включены только в более новых версиях, например, 2014-05-05 «Ponder Stibbons» <ref> https://www.dokuwiki.org/plugin:ondeniedlogin </ref>) не включают такую метку и не могут быть отделены от остальных URL (к которым может потребоваться доступ без защищенного соединения для экономии ресурсов сервера).

Fix Me! Остальная часть параграфа обрабатывает только запросы с `?do=login` запросом GET, который не охватывает по крайней мере страницы «доступ запрещен»! Необходимо исследование правила перенаправления для всех запросов аутентификации по HTTP. Смотрите обсуждение для решения.

Далее предполагается, что вы уже настроили поддержку HTTPS для своей вики, сделав ее доступной по HTTP и HTTPS по одному и тому же адресу. По соображениям производительности только вход в систему и обновления профиля должны быть принудительно переведены на HTTPS, в то время как все «нормальные» действия вики будут продолжать работать по HTTP.

Поскольку вам необходимо настроить куки через HTTPS для работы на HTTP, вам необходимо сначала отключить опцию securecookie . Затем приступайте к настройке перенаправления в вашем `.htaccess`:

```
# Переключиться на безопасный режим при входе в систему, профиле и действиях администратора
RewriteEngine On
RewriteCond %{HTTPS} !on
RewriteCond %{QUERY_STRING} do=(log|profile|admin)
RewriteRule ^(.*) https://%{HTTP_HOST}/$1 [R,QSA,L,NE]

# Изменить обратно на небезопасный режим при показе действия
RewriteCond %{HTTPS} on
RewriteCond %{QUERY_STRING} !do=(log|profile|admin)
RewriteCond %{REQUEST_METHOD} GET
RewriteRule ^(.*) http://%{HTTP_HOST}/$1 [R,QSA,L]
```

Возможно, вы захотите изменить `${HTTP_HOST}` его на `${SERVER_NAME}`, где имя сервера совпадает с именем хоста в вашем SSL-сертификате.

Примечания:

- выше переключает обратно на не-SSL только на действии show. Это означает, что переключение может не произойти немедленно после входа, но гарантирует отсутствие предупреждений о «смешанном контенте» во время операции SSL.
- если у вас есть другие правила переписывания, например, те, которые используются при общем переписывании, поместите эти правила перед остальными.
- Если каталог установки DokuWiki не является корневым каталогом (например, http://example.com/wiki/) , вам необходимо добавить этот дополнительный путь к строкам 5 и 11 приведенного выше фрагмента, что будет выглядеть примерно так:`RewriteRule ^(.*) http://%{HTTP_HOST}/wiki/$1 [R,QSA,L]`

защищенный cookie

Обратите внимание: вам необходимо отключить « securecookie », `conf/dokuwiki.php` чтобы приведенный выше код работал. В противном случае ваши входы не будут успешно регистрироваться. Это происходит потому, что при включенном securecookie сеансовый cookie, созданный во время входа HTTPS, не может быть отправлен по HTTP, и сеанс теряется.

nginx

Такая настройка также возможна в nginx, но с небольшой доработкой fastcgi_params.

Во-первых, вам нужно иметь отдельные экземпляры сервера, для `http` и `https` каждый, чтобы поддерживать чистоту (и `rewrite` не запутаться и не попасть в ловушку redir loops). Это может выглядеть так. Каждый экземпляр имеет свое собственное правило перезаписи для переключения с http на https.

```
# Протестировано с nginx 0.8.5
```

```
# В http-контексте вашей конфигурации nginx
map $scheme $php_https { default off; https on; }

    server {
    server_name wiki.host.org
    root /path/to/dokuwiki;
    index doku.php;
    listen 80;
    #Enforce https for logins, admin
    if ($args ~* do=(log|admin|profile)) {
        rewrite ^ https://$host$request_uri? redirect;
    }
    include dokuwiki.conf;
    }

    server {
    server_name wiki.host.org;
    root /path/to/dokuwiki;
    index doku.php;
    listen 443 ssl;
    keepalive_requests    10;
    keepalive_timeout     60 60;
    ssl_certificate      /etc/ssl/certs/ssl-cert-snakeoil.pem;
    ssl_certificate_key  /etc/ssl/private/ssl-cert-snakeoil.key;
    #switch back to plain http for normal view

    if ($args ~* (do=show|^$)){
        rewrite ^ http://$host$request_uri? redirect;
    }
    include dokuwiki.conf;
    }
```

В dokuwiki.conf (тот же путь, что и у вашего nginx.conf) вы можете использовать фрагмент из вики nginx, но вам нужно добавить

```
fastcgi_param HTTPS $php_https;
```

в ваш fastcgi_params. Этот параметр и map директива в начале обязательны, поскольку Dokuwiki проверяет $_SERVER['HTTPS'] для работы.

Как и в случае с Apache, вам необходимо отключить Securecookie в вашем conf/dokuwiki.php.

HTTPS на основе PHP

Ниже полезно, если вы хотите принудительно использовать https-соединение ВСЕГДА (не только для входа) и не хотите полагаться на Apache или NGINX htaccess или другие специфичные для сервера директивы. Поместите следующие строки в начало файла шаблона в '…lib/tpl/template-name/main.php'

```php
<?php
if ($_SERVER[HTTPS]!="on") {
  //$strURIName=$_SERVER['SERVER_NAME'] . getenv("REQUEST_URI");
  // Функция 'getenv' не работает, если ваш API сервера - ASAPI (IIS).
  // Поэтому постарайтесь использовать не getenv('REMOTE_ADDR'), а
$_SERVER["REMOTE_ADDR"].
  $strURIName= $_SERVER['SERVER_NAME'] . $_SERVER['REQUEST_URI'];
  header ("Location: https://$strURIName");
  // Если это не работает для вас и вам нужно устранить неполадки в вашем PHP-коде,
  // раскомментируйте ниже, чтобы узнать о ваших конкретных переменных сервера
  /*
  echo "<b>_SERVER Variables from $_SERVER</b><br><br>";
  reset($_SERVER);
  while (list ($key, $val) = each ($_SERVER)) {
  print $key . " = " . $val . "<br>";
  }
  */
}
?>
```

Спасибо. Это спасло мой день!

Обсуждение

From:
https://wwoss.ru/ - **worldwide open-source software**

Permanent link:
https://wwoss.ru/doku.php?id=wiki:tips:httpslogin

Last update: **2024/08/09 04:01**