Перенаправление на HTTPS

Протокол HSTS (HTTP Strict Transport Security) позволяет администратору сайта указать на необходимость обращения только по HTTPS и автоматизировать проброс на HTTPS при изначальном обращении по ссылке на HTTP.

Управление производится при помощи HTTP-заголовка Strict-Transport-Security, который выдаётся при обращении по HTTPS (при выдаче по HTTP заголовок игнорируется) и указывает браузеру на необходимость оставаться в зоне HTTPS даже при переходе по ссылкам "http://". Замена http:// на https:// будет автоматически выполняться при обращении к защищаемому ресурсу с внешних сайтов, а не только для внутренних ссылок.

Использование в Apache:

При помощи mod_headers устанавливаем для HTTPS-блока виртуального хоста заголовок Strict-Transport-Security (max-age - срок действия (1 год), includeSubdomains - распространять замену http:// на https:// для всех поддоменов; preload - занести в поддерживаемый браузером статический список). Дополнительно устанавливаем заголовок "X-Frame-Options: DENY" для запрета встраивания контента сайта в блоки iframe.

```
   LoadModule headers_module modules/mod_headers.so

   <VirtualHost 192.168.1.1:443>
      Header always set Strict-Transport-Security "max-age= 31536000;
includeSubdomains; preload"
      Header always set X-Frame-Options DENY
   </VirtualHost>
```

Для HTTP-блока хоста настраиваем редирект:

```
   <VirtualHost *:80>
     <IfModule mod_rewrite.c>
       RewriteEngine On
       RewriteCond %{HTTPS} off
       RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
     </IfModule>
    </VirtualHost>
```

Настройка в nginx:

Добавляем в блок server:

```
    add_header Strict-Transport-Security "max-age=63072000; includeSubdomains;
preload";
    add_header X-Frame-Options "DENY";
```

Настройка в Lighttpd:

```
   server.modules += ( "mod_setenv" )
   $HTTP["scheme"] == "https" {
       setenv.add-response-header  = ( "Strict-Transport-Security" => "max-
age=63072000; includeSubdomains; preload",  "X-Frame-Options" => "DENY")
   }
```