# Некоторые идеи по безопасности

Ключ расширения : **security**



## Оглавление



# Безопасность

В данном руководстве представлен подход предложенный на сайте wiki.typo3.org под названием "Security - some ideas from DocTEAM".

Безопасность требуется любому сервеверу веб, и большому и маленькому. Первичный замысел этой главы состоит в обучении веб-мастеров, администраторов сайтов и мастеров сайтов безопасности веб страниц.

Предусмотрено три раздела:

- Безопасность общая
  - Безопасность общая применима к любой установке.
- Безопасность PHP
  - Возможная безопасность применимая к PHP.
- Безопасность TYPO3
  - Безопасность, как она применима к TYPO3.

## Безопасность общая

Существуют различные пути облегчения процесса сохранности вашего сайта. Давайте немного обсудим их, вдоль и поперек.

**Безопасность вашего сервера**

Сетевая топология имеет первостепенное значение в обеспечении безопасности и защиты от внешних вторжений. Существует еще несколько моментов которые вам следует рассмотреть.

Конфигурация файервола

- Отбрасывать все пакеты для не санкционированных портов или те, которые не нужны.
- Разрешить только порты которые вы будете использовать или через которые будет проходить трафик, которые обычно включают: порты 80 (http) и 443 (https). Ограничивая входящие соединения и выделяя только определенные порты вы можете избежать огромного количества успешных атак.

Конфигурация сервера

- Отключите все неиспользуемые службы. Службы, которые не используются, но запущены и выполняются, могут таить две проблемы.
  - Паразитное использование ресурсов. Чем больше вы запустите, тем больше памяти и ресурсов вам потребуется.
  - Возможно потенциально приоткрыть брешь в безопасности, или позволить атаку типа отказ в обслуживании(denial of service) (DoS).
- Ограничьте число людей с административными(superuser) привилигиями.
- Используйте безопасное или шифруемое соединение при конфигурировании или создании.
  - Используйте Secure Shell (SSH) или другое похожее защищенное соединение.

**Удалите страницы по-умолчанию**

Большинство из нас уже из видели: сайт, который не сконфигурирован.Если он скажет "Welcome to Apache" или "Welcome to IIS", то это сайты по-умолчанию, которые обеспечивают большинство веб серверов являются введением в стиль"Welcome Hackers". Если хакер увидит что вы не изменили сайт по-умолчанию, то это только облегчит его деструктивную задачу. А что если сайт по-умолчанию выставляет на показ имена пользователей, пароли, причем не зашифрованные, имеет символические ссылки на другие закрытые от всех остальных места и т.д.? А тогда, хакер знает как подобрать ключики.

Предотвратите это! Если у вас нет контента на сайте, сделайте что-нибудь из следующего:

1. Дезактивируйте ваш сервер, до тех пор пока вы удалите все компоненты по-умолчанию.
2. Удалите весь контент по-умолчанию.
3. Измените каталог корневой  / документов вашего сайта на новый каталог.

**Удалите информацию сервера**

По главному методу, если мне нужно сломать сайт, то я просто ищу цели, которые либо используют определенные веб серверы, с многочисленными проблемами или известными разработчиками, или использую определенные серверы, которые трудно настроить, и поэтому их количество очень велико.

Одним из простых путей является удаление банеров, раскрывающее используемую серверную технологию. Уверен, NetCraft это не понравится, но они не отвечают за ваше время падения (имеется в виду сервера), не так ли? Мне нравится поддерживать свои банеры неуловимыми насколько это возможно.

Если серфер(surfer) получает сообщение 404 - Page Not Found, то он увидит только, что у меня сервер **"Webserver - www.mysite.com"**

## Удаленные обработчики контента(Remove Content Handlers)

Развивая мысль удаления информации сервера, я также отмечу, что возможность изменения url вашим сервером является следующей фантастической идеей. В основе идея замены url состоит в том, что пользователь получает документ отличный по запросу, но тот же по ссылке.

К тому же эффекту приведет, если вы вдруг завтра проснетесь и узнаете, что у PHP, ASP, и т.д. появилось более 5,000 разработок по безопасности, но конечный пользователь увидит только документы html, а это заметно снизит потенциальные атаки.

Когда пользователь видит каталог, а не файл, то это стиль RealURL, из-за которого пользователь в точности не знает тип вернувшегося документа. Очень похожее впечатление создает simulateStaticDocuments о статичной веб странице, которая не может быть вскрыта из-за переполнения буфера в технологии программирования. Из-за подобных изменений, ваш сайт может быть потенциально просмотрен.

В TYPO3 существуют возможности использования подмен для облегчения навигации конечными пользователями и бороться / сопровождать установки безопасности.

- Смотри simulateStaticDocuments
- Смотри RealURL

## Открытые каталоги

***Что такое открытые каталоги?***

Открытые каталог – это каталог (folder) на вашем сайте, в котором нет html страницы по-умолчанию. В зависимости от вашего сервера, эти файлы могут отличаться.

- Общие документы по-умолчанию Apache
    - index.html index.htm index.php
- Общие документы по-умолчанию Microsoft IIS
    - default.htm default.asp index.htm

Существование документиов по-умолчанию предотвращает поиск листинга каталога исследователями, что значительно снижает потенциальный риск безопасности.

***Фиксация открытых каталогов***

Существует несколько путей достижения этого:

1. Сконфигурируйте ваш веб сервер таким образом, чтобы он не отображал индексы каталогов.
2. Поместите документ по-умолчанию в каждый каталог и под каталог на вашем сервере.

Не требуется чтобы ваши документы по-умолчанию были излишне сложными. Конечно, множество сайтов применяют перенаправление на другую страницу. Это очень просто и легко делается.

Приведем набор простых примеров:

**index.php**

```
<?php
Header("Location: /index.php");
?>
```

**index.html**

```
<HTML>
 <HEAD>
```

```
  <TITLE>Page Not Found</TITLE>
 </HEAD>
 <BODY>
  Sorry, that page was not found.
  <A HREF="/index.php">Click here to continue</A>
 </BODY>
</HTML>
```

Самый простой документ по-умолчанию – пустой документ. Запомните, цель состоит в предотвращении доступа к данным. Имея просто документы по-умолчанию во всех каталогах, нельзя предотвратить доступ к этим каталогам, в противоположность этому, предотвращается просмотр кем-либо всех файлов в этом каталоге. Веб исследователь может, в теории, познакомиться с файлами, которые могут существовать или нет в каталоге, но невидимы, и поэтому, такая задача радикально затруднена.

### Основные ссылки по безопасности

Приведем некоторые ссылки, которые помогут вам поддерживать безопасность вашего веб сервера, или, наконец, подадут вам идею чего еще можно было бы предпринять:

- http://www.w3.org/Security/Faq/wwwsf3.html
- http://httpd.apache.org/docs/misc/security_tips.html
- http://www.cert.org/security-improvement/modules/m11.html
- http://www.ciac.org/ciac/bulletins/j-042.shtml

## Безопасность PHP

**Заметим:** *Эта глава применима к любому PHP запущенному на веб сервере в производственном окружении. Если воспользоваться этими приемами, то разработка и отладка станут управляемыми и прогнозируемыми задачами.*

### Конфигурация PHP.INI

Эта глава даст вам беглый просмотр опций безопасности доступных в файле PHP.INI.

#### *Скрывайте сообщения об ошибках*

Если ваш скрипт неверен, то вы не можете пожелать чтобы подробности выводились для Joe Websurfer. А что если ошибка содержит чуствительные данные, такие как имя пользователя и пароль? Следующие пара опций запустят запись ошибок и предотвратят их отображение.

```
log_errors = On
display_errors = Off
```

Эти опции являются ночным кошмаром для программистов расширений или основных разработчиков, потому что они скрывают все обычные отладочные сообщения. И тем не менее, это позволяет вам сделать более привлекательными сообщения об ошибках, которые пользовательможет увидеть.

#### *Выключайте Register Globals*

### Для чего нужны Register Globals?

Переменные посылаемые в PHP (например методами GET/POST) доступны в скриптах PHP, после этого, как еще веб исследователь может контактировать с веб сайтом? В далекие времена для PHP (раньше PHP v4.20), все данные GET/POST автоматически пересылались и конвертировались в переменную.

Обновите ваш PHP.INI для включения:

```
register_globals = Off
```

Для демонстрации включения register globals, запросите url http://domain.tld/index.php?id=123

С включенными Register Globals, переменная **id** GET конвертируется и становится доступной в PHP как переменная **$id** следующим образом:

index.php

```
<?php
echo "You requested id #" . $id;
?>
```

Эта *возможность* разрешает очень легкий доступ к данным GET/POST.

**А в чем проблема?**

Проблема состоит в том, что если ваш скрипт PHP предусматривает посылку **id** как переменной метода POST, веб пользователь может потенциально переписать ее методом GET. В зависимости от целей вашего скрипта, обстоятельства могут усилить возможные риски.

Сходите по этой ссылке с целью более глубоко изучения проблемы:

- http://us2.php.net/register_globals

**Ссылки по безопасности PHP**

Возможно вам захочется дополнить этим соответсвующие ресурсы:

Для администраторов:

- http://www.onlamp.com/pub/a/php/2001/01/11/php_admin.html
- http://www.onlamp.com/pub/a/php/2001/03/29/php_admin.html
- http://us2.php.net/register_globals
- http://builder.com.com/5100-6371-5272345.html

Для разработчиков:

- http://www.devshed.com/c/a/PHP/PHP-Security-Mistakes/
- http://www.developer.com/lang/article.php/918141
- http://www.sklar.com/page/article/owasp-top-ten

## Безопасность TYPO3

TYPO3 был создан для удобства; и до настоящего времени, он является более *Content Management Framework* (CMF), чем *Content Management System* (CMS). Безопасность является внутренней частью начиная с самого начала и до сего дня.

**Безопасность после установки**

После успешной установки TYPO3, вы можете рассмотреть некоторые из этих приемов. Я настоятельно рекомендую их все, хотя ваши запросы могут отличаться.

Как только я успешно сконфигурировал TYPO3 инструментом установки(install tool), мне нет необходимости использовать его немедленно. Для начала я хотел бы познакомиться с начальной безопасностью.

- **Измените пароль утилиты установки**. Конечно, нам всем известно, что по-умолчанию это "joh316". Смотрите вперед и замените его чем-то очень секретным и трудным для гостей? Резюмируем, **нам всем известен пароль по-умолчанию**.

Теоретически, этого достаточно, для начала, но я более увлечен проблемой безопасности.

Я уверен в правоте следующего:

- Я не буду часто пользоваться инструментом установки, поэтому я затрудню его использование.
- У меню есть оболочка доступа к веб серверу (или возможность релактирования файлов прямо на сайте).
- Я планирую использовать инструмент установки либо с самого веб сервера, либо с другой моей машины в моей сети.

**Отключите инструмент установки(Install Tool)**

Во-первых, единственной причиной необходимости доступа к инструменту установки(Install Tool) могут быть только радикальные изменения на сайте. Временами мне нужен такой доступ для создания нового сайта, обновления программ, обновления “железа” или изменения базы данных. Так это очень важно, я хочу основательно испортить установку.

Редактируем **/webserverroot/typo3/install/index.php** и добавляем оператор *die()*.

Вот вам пример защищенного инсталляционного файла index.php:

```
<?php
/***************************************************************
*  Copyright notice
*
*  (c) 1999-2004 Kasper Skaarhoj (kasperYYYY@typo3.com)
*  All rights reserved
*
*  This script is part of the TYPO3 project. The TYPO3 project is
*  free software; you can redistribute it and/or modify
*  it under the terms of the GNU General Public License as published by
*  the Free Software Foundation; either version 2 of the License, or
*  (at your option) any later version.
*
*  The GNU General Public License can be found at
*  http://www.gnu.org/copyleft/gpl.html.
*  A copy is found in the textfile GPL.txt and important notices to the license
*  from the author is found in LICENSE.txt distributed with these scripts.
*
*
*  This script is distributed in the hope that it will be useful,
*  but WITHOUT ANY WARRANTY; without even the implied warranty of
*  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
*  GNU General Public License for more details.
*
*  This copyright notice MUST APPEAR in all copies of the script!
***************************************************************/
/**
 * Starter-script for install screen
 *
```

```
  * $Id: index.php,v 1.11 2004/09/13 22:57:22 typo3 Exp $
  *
  * @author      Kasper Skaarhoj <kasperYYYY@typo3.com>
  * @package TYPO3
  * @subpackage core
  */
 // ***************************************************************************
 // Insert some security here, if you don't trust the Install Tool Password:
 // ***************************************************************************
 die("Disabled.");
 // ***************************************************************************
 // Defining constants necessary for the install-script to invoke the installer
 // ***************************************************************************
 define('TYPO3_MOD_PATH', 'install/');
 $BACK_PATH='../';
 // Defining this variable and setting it non-false will invoke the install-screen
called from init.php
 define('TYPO3_enterInstallScript', '1');
 require ('../init.php');
 ?>
```

Для того чтобы запустить инструмент инсталляции, вам нужно будет удалить или закомментировать оператор die(). Для этого вам потребуется доступ на редактирование/запись(edit/write) на вашем сервере.

***Ограничьте доступ к инструменту установки(Install Tool)***

Веб серверы часто позволяют вам ограничить доступ к каталогам, в зависимости от определенных адресов IP, хостов и сетей. Вы можете также употреблять пароли, но внекоторых случаях пароли пересылаются обычным текстом, позволяя злоумышленникам этим воспользоваться.

С Apache, я обожаю использовать файл **.htaccess** в каждом каталоге с ограниченным доступом. Ваша конфигурация сервера Apache может потребовать некоторых модификаций для реализации этого. Если так, пожалуйста воспользуйтесь руководством Apache для этого.

Мой файл **.htaccess** выглядит так:

```
order deny,allow
deny from all
allow from 127.0.0.1
allow from 192.168.1.0
allow from 192.168.10.100
```

Таким способом включается запрет на доступ для всех, но разрешается доступ для localhost (127.0.0.1), моей локальной сети (192.168.1.0) и одного компьютера с ip 192.168.10.100. Этим предотвращается любой доступ снаружи для проникновения, поскольку, как вы видели, он запрещен!

При использовании Microsoft IIS, методика немного отличается, но с тем же результатом.

**Удалите Административную учетную запись по-умолчанию**

Простым удалением учетной записи администратора, вам удастся предотвратить массу непрятностей. По-умолчанию администратор осуществляет полный контроль над установкой TYPO3. Пользователь **admin** также имеет пароль по-умолчанию, который известен большинству из нас.

Некоторые рекомендуют просто изменить пароль администратора. А я имею другое мнение. Я создаю другого администратора с отличающимся именем. Например, можно создать любого из следующих: **superadmin**, **super** или **hostmaster**. Будьте находчивы и создайте трудности гостям.

Насчет паролей, у каждого свой метод, я предпочитаю *минимум* 8 символов с пятью буквами, 2-3

заглавные и двумя цифрами и одним специальным символом ( **!@#$%^&*().,** ).

После создания нового администратора или изменения его имени на что-то еще, вместе с паролем, конечно, удалите старого, если возможно.

### Удалите доступ к файлу localconf.php

Примечание: **суммируя рекомендации kasper.**

Что нам предстоит сделать для того чтобы парировать возможность удаленного пользователя посмотреть ваш файл localconf.php. Пока это доступно вам самому и КРАЙНЕ РЕДКО случается, пожалуйста будьте осведомлены, что в случае неверной конфигурации на этом шаге, ваш сайт станет недоступным.

Сначала, создадим безопасный каталог вне зоны нашего сайта.

Например в Linux, если ваш корневой каталог документов */var/www/localhost/htdocs*, то мы можем создать новый каталог */var/www/localhost/securedata*. Затем мы переместим наш

*/var/www/localhost/htdocs/**typo3conf/localconf.php*** в */var/www/localhost/securedata*. После перемещения новый файл *localconf.php* содержащий следующее:

```
<?php
require("/var/www/localhost/securedata/localconf.php");
?>
```

Целью этого является предотвращение нечаянно или намеренно получить доступ к вашим конфигурационным данным

Пример шагов в Linux

```
 Обновите пути, сохраните ваши данные и.д.
Примечание: Вы можете также выбрать произвольное имя для перемещаемого файла
localconf.php
 # cd /var/www/localhost
 # mkdir securedata
 # mv htdocs/typo3conf/localconf.php securedata
 # echo "<?php \
 > require(\"/var/www/localhost/securedata/localconf.php\"); \
 >?>" > \
 > htdocs/typo3conf/localconf.php
 # chown -R apache:apache securedata
 # chown -R apache:apache htdocs/typo3conf/localconf.php
```

Вам должно захотеться дважды, трижды, четырежды проверить на правильность эти установки.

### Измените каталог typo3

Следуйте рекомендациям, которые вы здесь найдете на смену каталога typo3 на что-нибудь другое.

- http://typo3.org/documentation/document-library/doc_core_inside/Changing_the_default-1/

### Ссылки по безопасности TYPO3

Некоторые ссылки по безопасности в Typo3:

- Глава 3.9 в Inside TYPO3 вся посвящена безопасности.

--Joshua Preston 21:16, 24 Apr 2005 (CEST)